FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Monocular Depth Estimation for Three-Dimensional Reconstruction in Hysteroscopy Using Synthetic Data and Transfer Learning

DISSERTATION REPORT

Catarina Ferreira Guedes



Master in Bioengineering
Biomedical Engineering Branch

Supervisor: Prof. Dr. António Pedro Rodrigues Aguiar
Collaborator: Daniel Corona Oliveira Costa

October 31, 2026

© Catarina Ferreira Guedes, 2026

Abstract

Hysteroscopy provides direct visualisation of the uterine cavity, but conventional monocular imaging offers limited spatial information. Monocular depth estimation could recover three-dimensional information without additional hardware, supporting future improvements in the procedure. This work investigates its feasibility for 3D reconstruction in hysteroscopy using synthetic data and transfer learning, addressing the lack of reliable ground-truth depth in hysteroscopic images.

A synthetic hysteroscopic environment was developed to generate image sequences together with corresponding depth and camera information, allowing the behaviour of a pre-trained depth estimation model to be evaluated under controlled conditions. A self-supervised model that incorporates a geometric approach originally developed for gastrointestinal endoscopy was adapted to the hysteroscopic domain through fine-tuning on the generated data. Different training configurations were investigated, including learning rate, batch size, frame sampling and encoder freezing. Depth predictions were quantitatively evaluated on synthetic images and further analysed on real hysteroscopic images. The predicted depth was also used to reconstruct and fuse three-dimensional point clouds, providing an additional assessment of the spatial information recovered by the model. The needed computational time and resources were also addressed.

Some fine-tuning configuration showed improved metric for depth estimation on the synthetic test data. The errors were computed on synthetic test images after adjusting the scale of each image, independently, using the ratio between the median ground-truth depth and the median predicted depth. The configuration with the lowest mean AbsRel achieved a value of 0.26 and an RMSE of 2.96 mm, compared with 0.30 and 3.48 mm, respectively, for the pre-trained model. However, this model assumed a best performance at the first epoch, limiting the extent to which these improvements can be attributed to sustained fine-tuning. Conclusions for the real images depended on qualitative analysis with the addition of a reduced sensitivity to changes in image framing. The three-dimensional analysis revealed substantial frame-to-frame scale variation, demonstrating that improved depth estimation for individual images does not necessarily result in geometrically consistent reconstruction across a sequence. Overall, the results do not establish the viability of adapting a pre-trained monocular depth estimation model to hysteroscopy using synthetic data, but highlight sequence-level scale consistency and validation in the real domain as key challenges for future three-dimensional computer-assisted hysteroscopy.

Key-words: Learning-based depth estimation, fine-tuning, virtual uterine cavity, point-cloud fusion, image-guided intervention, hysteroscopy

Resumo

A histeroscopia permite a visualização direta da cavidade uterina, mas a imagem monocular convencional oferece informação espacial limitada. A estimativa da profundidade monocular pode recuperar informação tridimensional sem hardware adicional, possibilitando melhorias futuras no procedimento. Este trabalho investiga a viabilidade desta técnica para a reconstrução 3D em histeroscopia utilizando dados sintéticos e aprendizagem por transferência, abordando a falta de dados fiáveis de profundidade em imagens histeroscópicas.

Foi desenvolvido um ambiente histeroscópico sintético para gerar sequências de imagens juntamente com informações correspondentes de profundidade e câmara, permitindo que o comportamento de um modelo que estime a profundidade pré-treinado fosse avaliado sob condições controladas. Um modelo auto-supervisionado que incorpora uma abordagem geométrica originalmente desenvolvida para a endoscopia gastrointestinal foi adaptado ao domínio histeroscópico através de um *fine-tuning* nos dados gerados. Foram investigadas diferentes configurações de treino, incluindo taxa de aprendizagem, tamanho do lote, amostragem de fotogramas e congelamento do codificador. As previsões de profundidade foram avaliadas quantitativamente em imagens sintéticas e posteriormente analisadas em imagens histeroscópicas reais. A profundidade prevista foi também utilizada para reconstruir e fundir nuvens de pontos tridimensionais, fornecendo uma avaliação adicional da informação espacial recuperada pelo modelo. Recursos computacionais necessários foram considerados.

Algumas configurações de fine-tuning apresentaram melhorias nas métricas de estimação de profundidade nos dados sintéticos de teste. Os erros foram calculados em imagens sintéticas de teste após o ajuste independente da escala de cada imagem, utilizando a razão entre a mediana da profundidade de referência e a mediana da profundidade estimada. A configuração com o valor médio mais baixo de AbsRel atingiu um valor de 0,26 e um RMSE de 2,96 mm, em comparação com 0,30 e 3,48 mm, respetivamente, no caso do modelo pré-treinado. No entanto, este modelo assumiu melhor desempenho no primeiro epoch, limitando a extensão em que estas melhorias podem ser atribuídas a um processo de fine-tuning sustentado. As conclusões relativas às imagens reais basearam-se numa análise qualitativa, complementada por uma avaliação da menor sensibilidade a alterações no enquadramento das imagens. A análise tridimensional revelou uma variação substancial da escala entre frames, demonstrando que uma melhoria na estimação de profundidade para imagens individuais não resulta necessariamente numa reconstrução geometricamente consistente ao longo de uma sequência. De um modo geral, os resultados não permitem estabelecer a viabilidade da adaptação de um modelo pré-treinado de estimação monocular de profundidade à histeroscopia utilizando dados sintéticos, mas destacam a consistência da escala ao nível da sequência e a validação no domínio real como desafios fundamentais para o desenvolvimento futuro de sistemas tridimensionais de histeroscopia assistida por computador.

Palavras-chave: Estimação de profundidade baseada em aprendizagem, *fine-tuning*, cavidade uterina virtual, fusão de nuvens de pontos, intervenção guiada por imagem, histeroscopia

The United Nations Sustainable Development Goals

The proposed work contributes, directly or indirectly, to the achievement of several United Nations Sustainable Development Goals (SDGs) [1, 2]. These 17 goals define an agenda for promoting sustainable development and addressing major societal challenges, encompassing areas such as health, education, gender equality, innovation and more, providing a framework for assessing the broader impact of research and technological developments.

The SDGs with which this work is most closely aligned are presented below and analysed in terms of this work's contribution and the performance indicators and metrics that meet those goals.

- **SDG 3** - Ensure healthy lives and promote well-being for all at all ages:
  - **Goal: 3.4 - By 2030, reduce by one third premature mortality from non-communicable diseases through prevention and treatment and promote mental health and well-being.**

    • **Contribution:** The proposed framework supports the development of AI-assisted hysteroscopy by enabling accurate monocular depth estimation, followed by three-dimensional reconstruction of the uterine cavity. Enhanced spatial information and visualisation of the uterine anatomy may support more comprehensive inspection during hysteroscopy and provide a foundation for future developments, such as coverage assessment or AI-assisted detection and characterization of endometrial polyps, ultimately contributing to earlier diagnosis and more informed clinical decision-making.

    • **Performance indicators and metrics:** Reconstruction accuracy (e.g.: MAE, RMSE), qualitative assessment of the reconstructed anatomy and, in future work, the performance of coverage assessment methods and AI-assisted polyp detection approaches.

- **SDG 3** - Ensure healthy lives and promote well-being for all at all ages:
  - **Goal: 3.7 - By 2030, ensure universal access to sexual and reproductive health-care services, including for family planning, information and education, and the integration of reproductive health into national strategies and programmes.**

    • **Contribution:** By supporting the development of advanced technologies for three-dimensional reconstruction and visualisation of the uterine cavity, the proposed work may contribute to improving women's reproductive healthcare. More reliable three-dimensional visualisation could facilitate diagnosis in healthcare centres, including where there is limited expertise, while also providing a valuable tool for the education and training of gynaecologists. Furthermore, the reconstructed anatomy may serve as the basis for future tools aimed at improving the hysteroscopy.

• **Performance indicators and metrics:** Comparison of the performance between conventional 2D visualisation and AI-assisted 3D reconstruction, user studies involving clinicians or medical students, diagnostic support evaluation, and reconstruction quality metrics (MAE, RMSE and other depth estimation metrics).

- **SDG 3** - Ensure healthy lives and promote well-being for all at all ages:
  - **Goal: 3.8. - Achieve universal health coverage, including financial risk protection, access to quality essential health-care services and access to safe, effective, quality and affordable essential medicines and vaccines for all.**

    • **Contribution:** By improving the spatial information available during hysteroscopy, such methods have the potential to support more complete and consistent examination of the uterine cavity. In the long term, reliable AI-assisted three-dimensional reconstruction could contribute to the development of tools for standardized inspection, diagnostic support and, potentially, automatic lesion detection.

    • **Performance indicators and metrics:** Depth estimation and reconstruction performance (e.g., MAE, RMSE, Abs Rel), computational efficiency, qualitative evaluation by healthcare professionals and, in future work, assessment of applications built upon the reconstructed three-dimensional information.

- **SDG 9** - Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation:
  - **Goal: 9.5. - Enhance scientific research, upgrade the technological capabilities of industrial sectors in all countries, in particular developing countries, including, by 2030, encouraging innovation and substantially increasing the number of research and development workers per 1 million people and public and private research and development spending.**

    • **Contribution:** This work contributes to scientific research in medical image analysis by investigating the feasibility of three-dimensional reconstruction of the uterine cavity from monocular hysteroscopic images, extending concepts that have previously been explored in other endoscopic applications, such as the colon or the stomach. The reconstructed three-dimensional information provides a basis for future research on applications such as coverage assessment, computer-assisted diagnosis and image-guided interventions. The proposed framework promotes reproducible research while supporting innovation in medicine and image-guided interventions.

    • **Performance indicators and metrics:** Number of generated image sequences and corresponding ground-truth depth maps, reproducibility of the acquisition pipeline and description of the methodology and algorithm performance, contributing to research and innovation and being available for future reuse by the research community.

Acknowledgements

I do not know exactly how to start. "Start with the beginning", that is how it is said. But even so, I do not know.

I spent a lot of my childhood days with my grandparents and my great-aunt so I will start there: thank you for everything, for always supporting my decisions, despite not always knowing what on Earth this degree was actually about.

Grandfather and great-aunt, you will not be here to see the next phases of my life, but I know that you are smiling. I would like to thank my parents for always trusting me and giving me freedom, and for always being there to listen and help. Sister, for all the hugs and affection, as always, and I am sorry I was not home as often. To my aunts and uncle, with the usual "give me a call if you need to" (or some variation of that) and your immediate, caring replies. To my friends, who have supported me through every stage and as I have grown, whilst I have watched you grow too and feel proud of you. I do not know how to thank you all for the times you have been a shoulder to lean on and a listening, understanding ear. Cunha, for being all this and more, for seeing me, for broadening my horizons, for seeking out opinions different from your own and for always trying to be better. To more adventures.

I also thank my supervisor, Prof. António Pedro Aguiar, and colaborator, Daniel Corona, for the time you have dedicated, for the guidance you have given me over this period, and for always being understanding. Special thanks to investigator Tiago Marques for the contribution while my work was in progress.

I will always cherish the memory of these five years at FEUP: the people I worked with in groups and the lecturers and how they prepared me and instilled in me resilience, flexibility and resourcefulness. I will hold dear in my heart those unexpected moments, the library staff, the ladies at the library bar and the canteen and the long corridor that usually filled me with enthusiasm.

And, for the end, I will finish by mentioning the little Catarina: sempre chegaste aonde esta te esperava.

Catarina Ferreira Guedes

Statement on the Use of Generative Artificial Intelligence

Generative Artificial Intelligence (AI) tools, namely ChatGPT, Claude and Google Gemini, were used during the development of this dissertation exclusively as supporting tools for language-related and technical tasks. On all those occasions, there was a critical mindset and extra care was taken.

Those tools were used to refine and improve the clarity and grammar of some passages; suggest and refine search queries for the literature review; provide guidance on the use of the Blender's "Shader Editor" during the development of realistic materials and textures; provide the expected workflow for the scripts in Unity; assist in troubleshooting and understanding specific implementation issues encountered during the development.

Methodological decisions, experimental design, interpretation of results and the overall structure and reasoning presented throughout this dissertation did not come from Generative AI.

In addition, no patient images or other clinical data were ever submitted to AI tools, ensuring compliance with confidentiality and ethical principles throughout the development of this work.

Contents

1 Introduction **1**
1.1 Medical Background . . . 1
1.2 Motivation . . . 2
1.3 Objectives . . . 7
1.4 Outline . . . 8

2 Literature Review **9**
2.1 Literature Search . . . 9
2.2 Hysteroscopy State-of-Art . . . 10
2.2.1 Future Development . . . 11
2.2.2 Considerations . . . 12
2.3 Depth Estimation . . . 12
2.3.1 How to Have Depth Estimation . . . 12
2.3.2 Depth Estimation Evaluation Metrics . . . 19
2.3.3 Self-supervision Approach and Related Metrics . . . 21
2.4 Data for Endoscopic Depth Estimation . . . 22
2.4.1 Data Sources . . . 22
2.4.2 How Synthetic Endoscopic Datasets are Generated . . . 27
2.4.3 Validity of Training on Synthetic Data . . . 31
2.4.4 Validation on Real Data when Training on Synthetic Data . . . 33
2.4.5 Considerations . . . 35
2.5 3D Reconstruction . . . 37
2.5.1 From Depth and Camera Motion to a 3D Model . . . 37
2.5.2 Scale Ambiguity and Scale Consistency . . . 39
2.5.3 Evaluation of 3D reconstruction . . . 39
2.5.4 Other uses of Depth Estimation . . . 43
2.5.5 Practical and Clinical Deployment Concerns . . . 43
2.5.6 Considerations . . . 44
2.6 Adapting Pre-trained Weights and Fine-tuning . . . 44
2.6.1 Considerations . . . 51
2.7 Literature Review Discussion . . . 53

3 Methodology **54**
3.1 Dataset . . . 55
3.1.1 Dataset Establishment . . . 55
3.1.2 Synthetic Dataset Generation . . . 59
3.1.3 Synthetic Data Acquisition . . . 69
3.1.4 Collected Data in Unity . . . 78
3.1.5 Remaining Data . . . 79

3.1.6 Assessment of the Synthetic Rendered Data . . . 79
3.2 Depth Prediction Pipeline . . . 80
3.2.1 Prediction Method Decision . . . 80
3.2.2 The Endo-SfMLearner Framework . . . 84
3.2.3 Proposed Approach . . . 89
3.2.4 Interface and Dynamics . . . 97
3.2.5 Per-frame Scale Factor . . . 98
3.2.6 Per-frame Depth Representation in Point Clouds . . . 99
3.2.7 How the Depth Estimation Results are Measured . . . 99
3.2.8 Expected Results for Depth Estimation . . . 100
3.3 3D Reconstruction Pipeline . . . 101
3.3.1 3D reconstruction Fusion Procedure in Endo-SfMLearner . . . 101
3.3.2 3D reconstruction Fusion Procedure in this Work . . . 102
3.3.3 Global Scale Fator . . . 103
3.3.4 How the Fused Point Cloud Reconstruction Results are Measured . . . 103
3.3.5 Expected Outcomes for the Point Cloud Fusion Procedure . . . 104
3.4 Time and Computational Requirements for Clinical Applicability . . . 105
3.4.1 Used Harware . . . 105
3.4.2 What is and is Not Instrumented . . . 105
3.4.3 What is Recorded . . . 107
3.4.4 Time and Computational Requirements Expected Results . . . 107

4 Results and Discussion **109**
4.1 Depth Prediction Results . . . 109
4.1.1 Analysis of the Global and Component Losses and the Validation Metric 109
4.1.2 Effect of Fine-tuning and Training Configuration . . . 117
4.1.3 Comparison with the Previous Endoscopic Depth Estimation . . . 127
4.2 Analysis of Generalisation to Real Images . . . 128
4.2.1 Analysis of the Obtained Real Results . . . 128
4.2.2 Analysis of the Effect of Image Cropping . . . 132
4.2.3 Analysis for a More Diverse Real Images . . . 134
4.2.4 Summary . . . 136
4.3 Fused Point-Cloud Analysis . . . 136
4.4 Time and Computational Performance for Clinical Applicability . . . 141

5 Conclusions and Future Work **146**

References **150**

A Interviews **159**

B Blender Shading Characteristics and Defined Material **161**

C Unity Implementation and Data Acquisition **162**
C.1 Endoscope Movement . . . 162
C.2 Data Acquisition . . . 163
C.2.1 Ground-truth Depth Rendering . . . 163
C.2.2 Capture Rig and Frame Recording . . . 164
C.2.3 Ground-truth Depth Validation . . . 166
C.2.4 Trajectory Constraints for Parallax . . . 167
C.3 Unity Inspector Configuration . . . 170

D Model Scale - Illustration of the Strategy for Measuring the Meshes **174**

E Specific Parameter Configuration for Each Recording **175**

F Endo-SfMLearner License **177**

G File Organisation Before and After Adaptation to the Endo-SfMLearner Requirements **178**

H Implementation of the Fused Point-Cloud Generation **179**
H.1 Camera-Frame Geometry . . . 179
H.2 World Placement . . . 180
H.3 Prediction Scale . . . 181
H.4 The Fusing Loop . . . 182
H.5 Verification . . . 183

I Scale Aligment Analysis **184**

List of Figures

1.1 Illustration of a frozen step in the hysteroscopic procedure with a rigid hysteroscope. . . . 1
1.2 Frames for various positions along the intrauterine navigation by hysteroscopy. 1
1.3 Uterus and its constituents and layers. . . . 2
1.4 Hysteroscopy Procedures Market with a Growing Prediction until 2030 4
3.1 Overview of the methodology for the approach in the present work. . . . 54
3.2 Workflow of the implemented methodology, showing the steps performed with each software. . . . 55
3.3 Established setup for the phantom data acquisition. . . . 58
3.4 Overview of the synthetic dataset generation process. . . . 60
3.5 Scheme for some of the uterus models from the UterUS dataset. . . . 62
3.6 Chosen anatomical structures for the simulated environment, obtained from the UterUS dataset. . . . 63
3.7 UV mapping of the surface of the model 13_18, in Blender. . . . 64
3.8 Purple reflections on the intrauterine walls due to unrealistic lighting. . . . 67
3.9 Example of frames from the results after the Unity set up side by side with the real images from CHUSJ. . . . 68
3.10 Overview of the synthetic data acquisition process. . . . 70
3.11 Examples of the performed trajectories in the simulated Unity navigation for the collection of data. . . . 73
3.12 Relevant information on the Unity configuration of the automated helical trajectory and its point representation within the Game Scene. . . . 74
3.13 Some of the collected images from articles where a polyp is present in the intrauterine cavity. . . . 76
3.14 Architecture of the depth estimation module 85
3.15 Architecture of the pose estimation module 86
3.16 How the two networks relate 86
3.17 Overview of the proposed depth estimation approach. . . . 90
3.18 Main processing steps implemented in `preprocessing.py`. . . . 93
3.19 Processing steps applied to the real video data and images. . . . 94
3.20 Illustration of the dynamics for the Endo-SfMLearner algoritm training, validation and testing. . . . 97
4.1 Training loss for three configurations for the learning rate 10^{-5}: batch 4 and no skipping; batch 8 and skipping; batch 8 and no skipping. . . . 109
4.2 Validation AbsRel for three of the trained configurations for the learning rate 10^{-5}: batch 4 and no skipping; batch 8 and skipping; batch 8 and no skipping. 110
4.3 Training loss and validation AbsRel for the configuration with learning rate 5×10^{-5} and batch size 4. . . . 111

4.4 Training loss and validation AbsRel for the configuration with learning rate 5×10^{-5}, batch size 4 and frame skipping of 3. 111
4.5 Training loss and validation AbsRel for the configuration with learning rate 5×10^{-5}, batch size 8 and frame skipping of 3. 112
4.6 Training loss and validation AbsRel for the configurations with frozen layers, all with learning rate 10^{-5}, batch size 4 and frame skipping of 3. 113
4.7 Training loss components for the configuration with learning rate 10^{-5} and batch size 4. 114
4.8 Training loss components for the configuration with learning rate 10^{-5} and batch size 8. 114
4.9 Training loss components for the configuration with learning rate 5×10^{-5} and batch size 4. 114
4.10 Training loss components for the configuration with learning rate 5×10^{-5}, batch size 4 and frame skipping 3. 115
4.11 Training loss components for the configuration with learning rate 5×10^{-5}, batch size 8 and frame skipping 3. 115
4.12 Analysis of the depth predictions for the models of learning rate 5×10^{-5} and skipping, with comparison to the GT and one configuration for learning rate of 10^{-5}. 121
4.13 Analysis of the depth prediction through a 3D visualisation of depth for a single frame across all runs and including the GT. 126
4.14 Depth prediction for two real images from the model with configuration learning rate 5×10^{-5}, batch 4 and skip 3. 128
4.15 Analysis for real images. 129
4.16 Depth prediction for real images were the full squared image is compared to the center-cropped version of it. 133
4.17 Depth prediction for real images with polyps. 135
4.18 Fused ground-truth and predicted point clouds for scene_15 and scene_17. . . 137
4.19 Examples of the spatial overlap between the ground-truth and fine-tuned predicted point clouds and the ground-truth and without fine-tuned predicted point clouds for scene 15. 138
4.20 Trajectory of the scene under analysis, scene_15. 139
4.21 Trajectory of the scene under analysis, scene_17. 139
4.22 Preprocessing time per step. 142
4.23 Per-frame cost by step for the configuration with learning rate 10^{-5}, batch size 4 and frame skipping of 3, and the configuration with learning rate 5×10^{-5}, batch size 4 and no frame skipping. The whisker on each bar reaches the ninety-fifth percentile and the percentage above it is that step's share of the total. 143

4.24 Per-frame cost by step for the configurations with learning rate 10^{-5}, batch size 4 and freezing of the first, second and third encoder layers, respectively, both using frame skipping of 3. The whisker on each bar reaches the ninety-fifth percentile and the percentage above it is that step's share of the total. 143
4.25 Processing time for the point-cloud fusion implementation for one recording. . . 144
B1 Blender's "Shader Editor" construction and connection of nodes in order to have the desired effect, a similar visual to the intrauterine tissue. 161
B2 visualisation of the defined material shading created in Blender 161
C1 Camera Inspector settings for the object's characteristics. 171
C2 Camera Inspector section connected to the script responsible for the motion of the camera. 171
C3 Camera Inspector settings in Unity responsible for determining the placement of the waypoints. 172
C4 Spot light Inspector settings. 173
D1 Central cross-sections perpendicular to each principal axis for the four uterine cavity meshes (13_09, 13_18, 22_12 and 22_59), after axis alignment. 174
E1 Illustration of the considered curvature orientations for the uterus structures. . 175
G1 Organisation of the folders used during the preprocessing stage. 178
I1 Analysis of the depth prediction through per-frame depth converted into point clouds for a single frame across all runs and including the GT. 184

List of Tables

2.1 Summary of hysteroscopy features and advantages. 11
2.2 Comparison of learning possibilities for depth estimation. 14
2.3 Monocular depth estimation methods developed for endoscopy. 16
2.4 Data sources and augmentation of the surveyed depth methods 25
2.5 Synthetic data generation approaches in the reviewed literature. 31
2.6 Representative clinical application areas of endoscopic depth estimation 43
2.7 Depth estimation training strategies across the reviewed literature for endoscopy. 46
2.8 Training configuration of the reviewed methods for endoscopy that do not adapt pre-trained weights. 48
2.9 Training configuration of the reviewed methods that adapt pre-trained weights to the endoscopic domain. 50
3.1 Overview of existing datasets for depth estimation, disease classification, polyp recognition, segmentation and pose tracking. 57
3.2 Comparison of the Imaging Resources considered for the 3D model construction. 61
3.3 Comparison between the dimensions of the meshes measured in Unity and the corresponding real-world dimensions 69
3.4 Depth values obtained from the Unity Recorder AOV for a cube positioned at different known distances from the camera. 70
3.5 Uterine structures, considered orientation and number of frames used for each synthetic recording. 79
3.6 Partition of the synthetic dataset. 95
3.7 Level of freeze for the algorithm training layers. 96
4.1 Depth prediction results for models trained with all layers unfrozen and without frame skipping for the metrics AbsRel, SqRel and RMSE. 117
4.2 Depth prediction results for models trained with all layers unfrozen and without frame skipping for the metrics $\text{RMSE}_{\log}$, Log10, MAE. 118
4.3 Depth prediction results for models trained with all layers unfrozen and without frame skipping for the metrics δ_1, δ_2, δ_3. 118
4.4 Depth prediction results for models that revealed a best model on an epoch different than the first for the metrci AbsRel, SqRel, RMSE. 120
4.5 Depth prediction results for models that revealed a best model on an epoch different than the first for the metrics $\text{RMSE}_{\log}$, Log10, MAE. 120
4.6 Depth prediction results for models that revealed a best model on an epoch different than the first for the metrics δ_1, δ_2, δ_3. 120
4.7 Depth prediction results for the original not finetuned synthetic hysteroscopy sequence for the metrics AbsRel, SqRel and RMSE. 121
4.8 Depth prediction results for the original not finetuned synthetic hysteroscopy sequence for the metrics $\text{RMSE}_{\log}$, Log10, MAE. 122

4.9 Depth prediction results for the original not finetuned synthetic hysteroscopy sequence for the metrics δ_1, δ_2, δ_3. 122
4.10 Effect of frame skipping on depth prediction performance for models trained with all layers unfrozen for the metrics AbsRel, SqRel and RMSE. 123
4.11 Effect of frame skipping on depth prediction performance for models trained with all layers unfrozen for the metrics $\text{RMSE}_{\log}$, Log10, MAE. 123
4.12 Effect of frame skipping on depth prediction performance for models trained with all layers unfrozen for the metrics δ_1, δ_2, δ_3. 123
4.13 Effect of layer freezing on depth prediction performance for models trained with a learning rate of 1×10^{-5}, batch size 4, and frame skipping of 3. 124
4.14 Effect of layer freezing on depth prediction performance for models trained with a learning rate of 1×10^{-5}, batch size 4, and frame skipping of 3 for the metrics $\text{RMSE}_{\log}$, Log10, MAE. 124
4.15 Effect of layer freezing on depth prediction performance for models trained with a learning rate of 1×10^{-5}, batch size 4, and frame skipping of 3. 125
4.16 Scale-invariant shape descriptors of the predicted depth maps on the real hysteroscopy frames, given as the range observed across the analysed frames. . . . 131
4.17 Comparison of the full real image versus the crop agreement. 133
4.18 Point-cloud reconstruction metrics results 136
4.19 Scale consistency of the predicted depth, without and with fine-tuning, for the synthetic 15 and 17 scenes. 140
4.20 Memory usage of the different stages of the depth prediction pipeline. 141
A1 Registration of the performed interviews. 160
E1 Parameters used for the generation of the synthetic recordings. 176
I1 Predicted depth range for frame 20 of scene_15 across training configurations in terms of the per-pixel minimum and maximum of the predicted depth map before the scaling factor, including their difference, all in the network units. . . 185

Nomenclature

2D	Two-dimensional
3D	Three-dimensional
AbsRel	Absolute Relative Error
AI	Artificial Intelligence
AOVs	Arbitrary Output Variables
AR	Augmented Reality
ATE	Absolute Trajectory Error
b	Batch
CAGR	Compound Annual Growth Rate
cGAN	conditional Generative Adversarial Network
CHUSJ	Centro Hospitalar Universitário São João, Porto, Portugal
CNN	Convolutional Neural Network
CR	Computed Radiography
CT	Computed Tomography
FPFH	Fast Point Feature Histograms
FPS	Frames Per Second
FRD	Fréchet Radiomic Distance
GANs	Generative Adversarial Networks
GPU	Graphics Processing Unit
GRS	Global Rating Scale
GT	Ground Truth
HDRP	High Definition Render Pipeline
LR	Learning Rate
MAE	Mean Absolute Error
NeRFs	Neural Radiance Fields
NN	Neural Network
PT	Positron Emission Tomography
RGB	Red, Green, Blue
RMSE	Root Mean Squared Error
$RMSE_{log}$	Root Mean Squared Error of Logarithms
RPE	Relative Pose Error
RT-STRUCT	Radiotherapy Structure Set
RTA	Relative Threshold Accuracy
SfM	Structure from Motion
SfS	Shape from Shading
SLAM	Simultaneous Localisation and Mapping
SqRel	Squared Relative Error
SSIM	Structural Similarity Index Measure
std	Standard Deviation

TPU	Tensor Processing Unit
TSDFs	Truncated Signed Distance Functions
TVUS	Transvaginal Ultrasound
UU	Unity Units
VAEs	Variational Autoencoders
VR	Virtual Reality

Chapter 1

Introduction

The purpose of this chapter is to provide an overview of the topic explored in this dissertation, situating it within its respective field and highlighting the motivation for its development. The objectives of the dissertation are also outlined and described.

1.1 Medical Background

Hysteroscopy consists of a minimally invasive procedure performed by qualified gynecologists to diagnose and treat intrauterine lesions. In this procedure, a hysteroscope with a camera attached explores the uterine cavity with the help of image software [3, 4]. An illustration for a specific position using a rigid hysteroscope is presented in Figure 1.1.

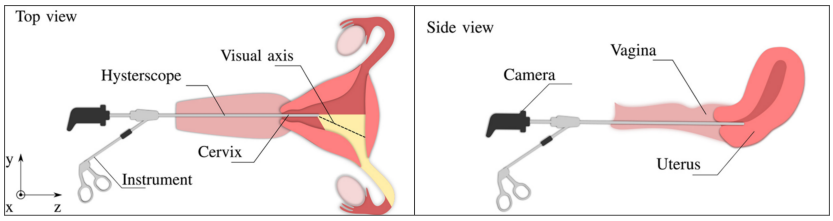


Figure 1.1: Illustration of a frozen step in the hysteroscopic procedure with a rigid hysteroscope, showing both top and side views (image source from Poliakov et al. [5]).

This figure highlights the complexity due to the pear-shaped, non-uniform three-dimensional (3D) structure of the uterus that influences hysteroscopic navigation [6]. To illustrate the navigation process and the complexity of the intrauterine environment, hysteroscopic views acquired from "Centro Hospitalar Universitário São João" (CHUSJ) in Porto, Portugal, are presented in Figure 1.2. Two frames were selected from each video, with the selection aiming to represent the visual diversity observed across the available recordings.

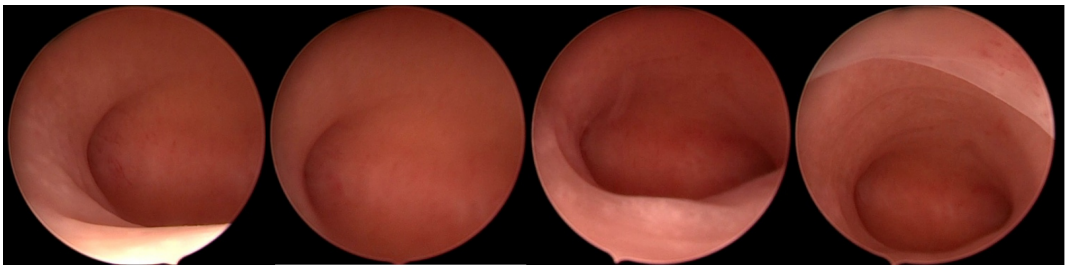


Figure 1.2: Frames for various positions along the intrauterine navigation by hysteroscopy (real images from CHUSJ).

The uterus is a hollow organ that matures with puberty [7] and plays a central role in the menstrual cycle, fertility and pregnancy, serving as the site of embryo implantation and

fetal development [8]. Despite its anatomically defined location in the female lower abdomen between the bladder and rectum [7] and its constituent anatomical structures, depicted in Figure 1.3, the uterus exhibits considerable variability both between individuals, with differences in orientation, tilt and curvature, and within the same individual in terms of size, shape and morphology, depending on the reproductive stage and exposure to ovarian steroid hormones [9–11].

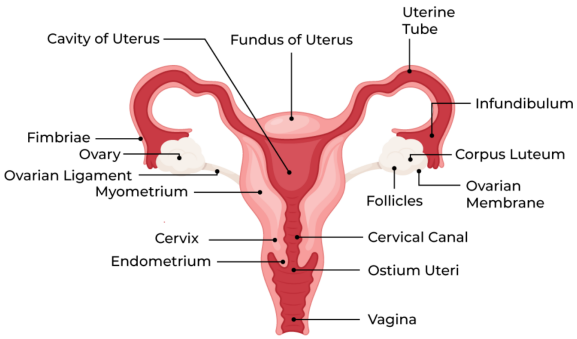


Figure 1.3: Uterus and its constituent structures and layers (image source from [12]).

Regarding uterine pathologies, endometrial polyps are common ones. Others include fibroids, endometrial polyps, endometriosis and pelvic inflammatory disease. They typically present with abnormal bleeding, pelvic pain and fertility-related symptoms [8].

Endometrial polyps develop within the inner endometrial layer, a vascularized inner lining that undergoes cyclic changes in thickness and composition throughout the menstrual cycle [9, 11, 13]. They consist of localized overgrowths of endometrial cells, thus containing the endometrium characteristic components in variable quantities: glands, stroma, and blood vessels [9, 14]. The cause for their formation is not known [14], but hormonal factors appear to play a significant role in their development [15, 16]. They may present different colors according to vascularization and different sizes. In addition, they may exist as single or agglomerations and can be classified morphologically as pedunculated or sessile according to elongated or large flat bases, respectively [14, 17].

Diagnostic approaches include transvaginal ultrasound and hysteroscopy, which enables direct visualisation and histopathological confirmation [18]. Their treatment will be according to the history of the patients and to the surgeon's preference and expertise. In order to remove them, high-speed rotational movements are applied with an instrument tip until the polyp base is completely separated from the endometrium, which will be left undamaged due to how it is performed [19] unless adverse outcomes arise [17].

1.2 Motivation

Choosing the Improvement of a Medical Procedure

The ongoing improvement of medical procedures is a key focus of medical innovation. It can have benefits such as reducing facilities' cost, enhancing efficiency and economic value, avoiding procedure repetition, reducing work time by optimization, reducing risks and improving

procedures security and outcomes [20]. For instance, in hysteroscopy, with a reduced time for the procedure to happen, there is less opportunity for uterine perforation. In addition, by reducing time, complications have less opportunity to develop or happen, namely the saline medium intravasion, as the longer it lasts, the greater the risk of leakage, bleeding and infection [21].

One possible way to achieve this is through the development of advanced computer-assisted technologies that enhance the 3D visualisation, in this case, of the uterine cavity, and provide richer spatial information during procedures. These technologies have the potential to support clinicians, facilitate the assessment of the examined regions and provide a foundation for future applications, including the use of detection algoritms for pathologies and physician training.

Choosing Hysteroscopy

Regarding the procedure that was considered to be explored, it brings advantages considering it is used to both diagnose and treat multiple intrauterine pathologies and it has advantages over alternative procedures when detecting and managing the same abnormalities and lesions [22–26].

Advantages include improved image quality and definition, which are also supported by the saline solution pump that helps maintain visual field clear and increases contrast [4, 27]. With it, it becomes possible to diagnose and operate in a single intervention, at the same time presenting accuracy and feasibility [3]. The procedure is cost-effective, as softwares are durable, reusable and do not consume much and only the reamining instruments, scissors and tweezers, require periodic replacement due to wear [17]. By using thin instruments and a saline solution pump with continuous inflow and outflow, pressure is reduced while ensuring adequate uterine distension. Thus, it is possible to perform the procedure without anesthesia or cervix dilation, unless necessary, and two other tools are avoided [4].

This reality has allowed for the procedure to be possible to happen in an *in-office* or ambulatory setting if a surgery room is not needed [3]. In addition, it is associated with a low complication rate [17], and the patient may not even need recovery time [4].

Potential complications of hysteroscopy include pain as the primary reason for failure [4]. Bleeding during the procedure, pelvic inflammatory disease, infection [17] and uterine perforation are the most common risks, and gas embolism is the rarest. Other possible ones are the risk of intravasation of the saline medium [17], which presents a low risk [4], but the longer the hysteroscopy time, the greater the risk of leakage, which can be monitored by checking the intraoperative fluid balance.

It has become the gold standard for diagnosis and treatment, and an analysis of the market size reflects it. Figure 1.4 illustrates the projected growth trend of the hysteroscopy procedures market in the US from a report from 2024. It shows a Compound Annual Growth Rate (CAGR) of 5.2% from 2025 to 2030. This reinforces the importance of focusing on it and optimizing its outcomes.

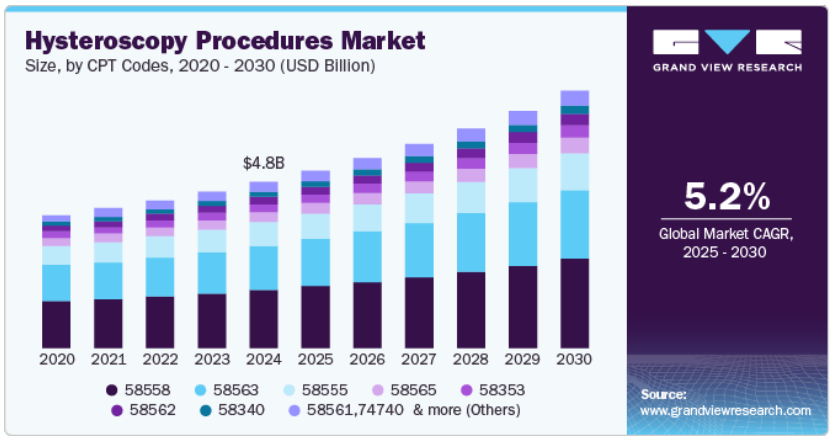


Figure 1.4: Hysteroscopy Procedures Market with a Growing Prediction until 2030 (image source from [28]).

Motivation Validation through Consultation

Interviews were conducted to obtain diverse perspectives, understand the needs of critical users of the proposed product and ultimately get a validation of the motivation of this work.

Only two were possible, as only these two participants responded to the invitation. A portuguese gynecologist from Barreiro and a portuguese gastroenterologist and investigator were consulted. While the first interviewee justified the motivation by highlighting the difficulty of detecting polyps and emphasizing the importance of minimizing procedure duration, the second focused on the relevance of innovation and procedural optimization, highlighting that they are always welcome in the medical field. More information on the first interview, one's position, workplace and key insights of the discussion is presented in Appendix A. The topics discussed during the second interview are not reported, as no information relevant to the scope of this work was obtained. While these opinions provide valuable insights, they should be interpreted within their context, as a gathering of additional perspectives would ensure a more comprehensive understanding.

Clinical Relevance of the Proposed Work

Hysteroscopy is currently regarded as the gold standard for the diagnosis and treatment of intrauterine abnormalities. Nevertheless, conventional hysteroscopy provides only a monocular two-dimensional view of the uterine cavity [4]. Since the image carries no direct depth information, clinicians must infer spatial analysis and lesion morphology from camera motion and personal experience. Consequently, the procedure depends on the operator's real-time judgment, including navigation and operative decision-making [29]. Navigation is also related to the extent to which the uterine cavity has been visualised. In the study conducted by Curlin et al. [30], investigators determined whether the cavity had been visualised based on their own judgment, without requiring any additional predefined criteria.

There is a need for systematic documentation of hysteroscopic findings. Structured reporting has been proposed as a way to improve consistency, completeness and communication

across clinicians and centres [27, 31]. Unstructured narrative reporting may fail to document unexamined or uninvolved regions adequately [31]. Furthermore, as reported by Toscano et al. [32], an incomplete examination is clinically relevant.

Thus, a coverage mapping system, defined as a visual representation of the inspected and uninspected areas of the uterine cavity, could help address the dependence on the operator's judgment by reducing ambiguity regarding whether the cavity has been fully explored. By providing a visual representation of the areas examined and those that remain unexamined, a visual extension of the procedure, such a system could make the completeness of the examination more explicit and help identify and document areas that could not be adequately visualised. For example, if a region containing a lesion could not be inspected, the coverage map could clearly indicate that region, providing a visual record of the incomplete examination. It could also offer a common visual reference to support traceability and communication between clinicians. The article by De Silva et al. [27] recommends capturing digital images of the uterine cavity or cervix whenever possible, giving a panoramic view of the cavity as an example. A coverage mapping system could potentially contribute to this objective by providing a broader visual representation of the examination. In addition, an international consensus on structured reporting promotes standardized data collection, highlighting its value for clinical care, research and education, and arguing that it improves diagnostic reproducibility and supports multicentre registries [31]. Data derived from coverage mapping, such as the regions examined and the progression of the examination, could contribute to this goal by providing structured information about uterine cavity exploration.

In this way, the development of a coverage mapping tool addresses a relevant problem in clinical workflow and documentation by supporting structured completeness assessment and standardized documentation. It should be noted, however, that the literature supports the need for structured completeness criteria and standardized documentation but does not establish coverage mapping as an existing clinical requirement. It is therefore presented here as a motivated proposal. The main value of such a system would be to support systematic inspection, helping clinicians assess whether the entire uterine cavity has been examined according to more standardized criteria, improve examination traceability, and provide a shared visual representation of the areas already inspected during the procedure. This aligns with the direction of recent consensus work in hysteroscopy, which emphasizes standardized terminology, structured sections and consistent reporting of uterine cavity findings to improve reproducibility and research utility.

At the same time, by improving documentation and supporting more systematic assessment of examination completeness, such a system could ultimately contribute to improvements in hysteroscopic practice, with potential benefits for disease diagnosis, follow-up and monitoring.

Besides its role in documentation, coverage mapping could also be useful when a decision is made to monitor a lesion or other abnormality rather than remove it. During a subsequent follow-up examination, a visual record of the previously identified region and a real-time notion of the seen regions during navigation could help the clinician locate it more easily and direct attention to the relevant area.

In addition, recording examination completeness and providing a visual representation of inspected and uninspected regions may play a role in training, certification and the development of hysteroscopists' psychomotor skills, three topics mentioned by Catena et al. [31] and De Silva et al. [27]. Such a representation could support the objective evaluation of procedural completeness and reporting accuracy.

Finally, De Silva et al. [27] identifies the assessment of the safety, acceptability and feasibility of hysteroscopy according to the angle of the distal optical lens as a research consideration. Coverage mapping could potentially contribute to this type of research by providing information about the regions visualised during the examination and supporting the assessment of how the viewing angle affects cavity visualisation.

Building such a map requires knowing the three-dimensional geometry of the cavity surface, which a monocular image sequence does not provide directly. Recovering this geometry is the step addressed in this dissertation. Beyond coverage assessment, three-dimensional information obtained during or after hysteroscopy could also help clinicians assess lesion size, location and anatomical context and support pre-operative planning, documentation and follow-up [33]. This could benefit patients through better monitoring and disease prevention, while promoting their involvement in their own care and informed decision-making.

Advances in computer vision, particularly in depth estimation for monocular collected data, have brought new possibilities for recovering 3D information from these images and for developing a wide range of applications based on the resulting reconstruction. Examples of works focusing on that theme are the works from Ozyoruk et al. [34], İncetan et al. [35] and Rau et al. [36]. Depth estimation is one way of technological advancement in hysteroscopy, deserving further investigation and exploration regarding its cost, reproducibility, potential applications and clinical value.

Motivation Final Remarks

A clinical problem that remains open in hysteroscopy is the lack of an objective record of how much of the uterine cavity was actually inspected, since completeness has been left to the operator's own judgment, without predefined criteria [30]. Ultimately, this work aims to contribute to the development of a depth estimation approach for hysteroscopic images, providing a basis for a future coverage mapping of a three-dimensional reconstruction of the uterine cavity to support the hysteroscopic procedure. This work builds on existing research into depth estimation in endoscopy and its use for subsequent 3D reconstruction, one example being the approach developed by Ozyoruk et al. [34]. Such 3D information may provide a basis for assessing the completeness of uterine exploration, supporting structured reporting [31], and serve as a foundation for future computer-assisted applications, including computer-aided diagnosis systems. Coverage mapping itself is not implemented or validated in this dissertation, and its clinical value remains to be demonstrated.

1.3 Objectives

The main objective of this dissertation is to investigate the feasibility of adapting a monocular depth estimation framework to the hysteroscopic domain in order to reconstruct the three-dimensional geometry of the uterine cavity from conventional monocular endoscopic images. By providing depth information that is not directly available during standard hysteroscopy, the proposed approach aims to improve the three-dimensional visualisation of the uterine cavity and establish the basis for future applications such as coverage assessment, computer-assisted navigation and computer-aided diagnosis, such as with polyp detection. This will ultimately assist healthcare professionals with the final aim of optimizing the hysteroscopy procedure.

To achieve this objective, the following goals are defined:

- Develop a simulated synthetic hysteroscopic environment that resembles the procedure circumstances capable of generating RGB image sequences together with accurate ground-truth depth maps, camera poses and intrinsic parameters;
- Develop a depth map estimation model for 3D monocular hysteroscopic images by adapting a model originally developed for gastrointestinal endoscopy to the hysteroscopic domain using the generated synthetic dataset;
- Evaluate the quality of the estimated depth maps and the resulting three-dimensional reconstructions using quantitative metrics on the synthetic dataset, and perform a qualitative evaluation of the reconstructed point clouds and depth estimation results on real hysteroscopic images;
- Assess the applicability of a monocular depth estimation approach previously proposed for gastrointestinal endoscopic image sequences to the intrauterine endoscopy, the hysteroscopy, identifying its strengths, limitations and potential for domain adaptation;

The work is guided by the following research hypotheses:

- A monocular depth estimation model originally trained for gastrointestinal endoscopy can be successfully adapted to hysteroscopy through transfer learning using synthetic data;
- Fine-tuning the model with a synthetic hysteroscopic dataset improves the accuracy of depth estimation and the quality of the reconstructed three-dimensional geometry when compared with the original pre-trained model;
- A sufficiently realistic synthetic environment provides an alternative and first step for training and evaluating monocular depth estimation algorithms in the absence of real hysteroscopic datasets with ground-truth depth;
- The depth maps estimated by the adapted monocular depth estimation model enable the reconstruction of a coherent three-dimensional representation of the uterine cavity from monocular hysteroscopic image sequences.

1.4 Outline

Beyond this introductory chapter, the dissertation is organised into four more chapters. Chapter 2 reviews the clinical and technical context of hysteroscopy, monocular depth estimation, synthetic data and 3D reconstruction. Chapter 3 presents the methodology, including the synthetic dataset development, model adaptation through transfer learning and evaluation procedures. Chapter 4 presents and discusses the experimental results, including depth estimation, generalisation to real images, 3D reconstruction and computational performance. Chapter 5 summarises the main findings, evaluates the proposed objectives, and discusses the limitations and future work.

Chapter 2

Literature Review

This chapter presents a comprehensive review of the literature relevant to the scope of this dissertation, aiming to establish the clinical and technical foundations that support the proposed work.

The clinical context of hysteroscopy is first introduced in order to deepen the subject, understand the environment and have the needed context to establish a baseline for the synthetic data generation and 3D reconstruction.

Then, depth estimation is reviewed, including the different approaches used to perform it, such as geometric and learning-based methods, with the review subsequently focusing on its application to endoscopy and on the metrics used for evaluation. Data sources are then analysed, with a greater focus on synthetic data, particularly in terms of how they are generated and how they are evaluated and validated. Finally, 3D reconstruction is addressed, covering what it consists of, how it is evaluated, and the main aspects that need to be considered. At the end, a section explores the gap and need addressed by the work.

Whenever considered necessary, reflections and considerations resulting from the review were included at the end of each part.

2.1 Literature Search

The literature review was conducted through the search of scientific publications and technical reports.

Bibliographic databases were consulted, primarily Scopus and Google Scholar. In addition, free-text searches in Google Search were used for more specific topics and clarifications, which led to relevant sources indexed in IEEE Xplore, PubMed and ResearchGate. The search was conducted up to May, 2026. It did not serve only to address the expected topics but also to support investigation, validation, comparison, argumentation and methodological insights. Among the main search queries were included the following, which represent a querie version for "Scopus". Some of them were produced with some help in adjustment from the generative AI ChatGPT. Some of the used queries were:

- `TITLE-ABS-KEY (review AND virtual AND hysteroscopy)`
- `TITLE-ABS-KEY (("3D Reconstruction" OR "3D Depth" OR "dense reconstruction") AND image AND (Hysteroscop* OR Endoscop*) AND (monocular OR "monocular vision"))`
- `TITLE-ABS-KEY ((endoscop* OR hysteroscop*) AND (self-supervised OR unsupervised OR SLAM OR SfM OR "Structure from Motion")`

```
AND "depth estimation")
```

- `TITLE-ABS-KEY ((endoscopy) AND ("map quality" OR "map coverage" OR "reconstruction quality"))`

The collected articles focused on peer-reviewed publications. The inclusion criteria prioritized recent studies and works that contributed meaningfully in the already mentioned ways of inclusion. For some documents, the bibliographical references were also reviewed whenever validation was required or when additional relevant information and comparisons were expected.

Articles were excluded when, simultaneously, the document was not available and the abstract did not provide sufficient information, also when the content did not contribute to the research objectives.

2.2 Hysteroscopy State-of-Art

As mentioned in the previous Chapter, hysteroscopy is a minimally invasive procedure [3] that allows direct inspection and treatment of intrauterine lesions. This endoscopic examination revolutionised the gynecological healthcare [3, 6]. Information regarding procedural advancements and the factors that justify its status is summarized in Table 2.1.

Table 2.1: Summary of hysteroscopy features and advantages.

Aspect	Description	Clinical / Technical Relevance
Imaging	Real-time high-quality visualisation with light source; can be rigid, fibre-optic or flexible; uses a single camera (standard); stereoscopic rising solution for accurate structures that uses dual camera for 3D but at high computational cost [37]	Enables panoramic view and improved diagnosis; enables the assessment of polyp characteristics, including number, size, location, surface texture, mucosal coating, superficial vascularization; 3D imaging limited by cost and availability
Diagnostic and therapeutic value	Enables diagnosis and treatment in a single intervention; often preceded by ultrasound [3, 16, 19]; detects and treats abnormalities such as polyps, fibroids, myomas, placental remnants, congenital malformations [3]	High clinical utility for multiple intrauterine pathologies; reduces number of procedures, time and costs; low complication rate [17]; possibly not need for recovery time [4]
Patient comfort	No speculum needed for dilation; no tenaculum required due to instrument endings [4, 38]	Less discomfort; increased feasibility in outpatient settings
Instrumentation	Thin, long, sterilized hysteroscope inserted through cervix; may include bipolar loop or alligator ending [14]; reusable; software is durable and do not consume much; only scissors and tweezers require periodic replacement due to wear (from the interview with a Gynecologist)	Enables minimally invasive access and interaction with lesions; no need for anesthesia or cervical dilation unless required [4, 5]; increased feasibility in outpatient settings; cost-effective [17]
Fluid management	Saline pump provides continuous inflow/outflow, maintaining low intrauterine pressure and adequate distension; isotonic fluid reduces overload risk [27]	Supports visualisation quality; reduces discomfort

2.2.1 Future Development

Future development should address challenges, such as the limited forceps grip due to the small hysteroscope diameter (interview with a Gynecologist), and aim at the ongoing reducing of instrument size and integration of 3D hysteroscopy for improved evaluation.

Yang et al. [39] also highlight that endoscopes are often either uncalibrated or dynamically adjusted during procedures. If this is the case for the hysteroscope, a suggestion would be for this parameters or procedure to be standarised so that it is not an additional factor to tackle with depth estimation strategies.

Other promising directions will be the treatment applying AI for lesion detection and classification, and exploring endoscopic robotics for finer dissection and two-handed maneuvers are promising directions [3].

The ideas pursued in this project align in a way with these proposed innovations, aiming to help for current limitations and advance the precision, safety and effectiveness of hysteroscopic

interventions.

2.2.2 Considerations

Analysing the hysteroscopy state-of-art is essential to set the baseline of this work. For instance, as the use of dual camera and SLAM reconstruction arises high computational costs to the software, alternatives can be considered for the single camera that is the standard. In turn, the use of a single camera and collection of 2D images sets the baseline for the 3D reconstruction to be applied and to be proposed to defy the computational resources implied in the rising solution of a dual-camera for 3D depth.

In addition, understanding intrauterine navigation provides important context for the images that are acquired. For instance, the images are typically of high quality due to the characteristics of the hysteroscope and the continuous saline infusion. Furthermore, because the procedure does not require the use of a speculum and generally causes less discomfort to the patient, clinicians can focus on performing a systematic examination to ensure that the entire intrauterine cavity is inspected, but it is still important that the examination does not take that much time. The distension of uterus provided by the saline pump also influences the overall appearance and consistency of the acquired images.

These considerations are also critical when addressing image quality and frame characteristics for the development of lesion detection algorithms [3].

2.3 Depth Estimation

A camera captures a 3D scene by projecting it onto a 2D image plane with a defined height and width and three RGB colour channels. The resulting image contains visual cues that provide information about the spatial arrangement of the scene, such as perspective, occlusion, texture gradients, shading and familiar object size, which are cues that allow the human eye to infer depth. However, those cues have no meaning for the 2D image, as the depth coordinate of each observed point is not explicitly available [40, 41].

Depth estimation addresses this problem by inferring the distance of the observed scene points relative to the camera from the available visual information. The result is commonly represented as a depth map, in which each pixel is associated with a depth value. This information can subsequently be used to determine distances between points, estimate the size and shape of objects and their proximity and recover the three-dimensional structure of a scene. Such information is relevant in applications including autonomous driving and parking assistance, robotic navigation, augmented and virtual reality and medical image-guided procedures [39, 40].

2.3.1 How to Have Depth Estimation

Depth information can be obtained through different approaches, depending on the characteristics of the environment and the available hardware.

Active depth-sensing systems, such as structured light and laser-based scanning, emit energy into the environment and measure its interaction with the scene. These approaches can provide direct measurements of distance and, if conditions are appropriate, high metric accuracy [41, 42]. As it does not depend exclusively on visual characteristics, active sensing can also be advantageous compared to methods that only involve images. However, these systems require specific hardware and may not be suitable when space, weight, cost or integration constraints are restrictive [40].

Passive vision-based approaches explore strategies such as shape-based, multi-view geometry-based and learning-based methods. The former infer surface geometry from variations in image intensity under assumptions about illumination and reflectance, Shape from Shading (SfS) being an example. The second recover depth by the possible geometric relationships between multiple images of the same scene that are available from different viewpoints. Stereo vision and Structure from Motion (SfM) are examples of those methods [39].

Stereo vision uses two cameras with a known relative position and estimates depth from the disparity between corresponding image points. Once a physical point has been identified in both images, that were taken at the same time, its position in three-dimensional space can be recovered through triangulation. The accuracy of this process depends, among other factors, on the ability to establish reliable correspondences between the two views and on the camera baseline [39, 40].

In turn, SfM performs triangulation using corresponding observations acquired from different viewpoints of a moving camera. The fundamental idea behind SfM is that the camera's movement creates different viewpoints of the same scene, making it possible to simultaneously estimate both motion and 3D structure [41]. SfM may obtain spatial information from two or more cameras, or, in contrast, from a single one, gathering information from consecutive frames of the image sequence [34, 43]. Therefore, unlike in stereo vision, where the relative configuration between cameras is fixed and known, SfM recovers the camera motion from the image sequence itself.

Simultaneous Localisation and Mapping (SLAM) builds upon related geometric principles but tackles a broader problem, as it simultaneously estimates the camera trajectory and constructs or maintains a representation of the surrounding environment. Thus, while SfM primarily focuses on recovering scene structure and camera motion from multiple views, SLAM maintains a map while simultaneously estimating the camera pose within that map. It can also use a single or two or more cameras, exploiting the relative motion of the camera and correspondences between views [44, 45].

Finally, learning-based methods use machine learning, particularly deep neural networks, to learn relationships between image appearance and depth or scene geometry.

The development of deep learning has significantly expanded the use of monocular depth estimation [33]. Investigation has been exploring learning-based depth estimation methods either with supervised, weak-supervised, self-supervised or unsupervised approaches, resumed in Table 2.2.

Table 2.2: Comparison of learning possibilities for depth estimation, according to Niu et al. [33].

Approach	**Training Information**	**Advantages**	**Drawbacks**
Supervised Learning	Images paired with dense ground-truth depth maps. The network learns by comparing its predictions with the available depth annotations	Can achieve high accuracy when accurate and representative ground-truth data are available	Requires dense and accurate depth annotations, which can be expensive or technically difficult to acquire
Weakly-Supervised Learning	Partial annotations or indirect information, such as intrinsics or correlations between images	Reduces the need for complete ground-truth annotations and can exploit information that is easier to acquire	The incomplete or indirect supervisory signal may provide less precise information about the true depth
Self-Supervised Learning	Unlabelled image sequences are used to construct the supervisory signal. In monocular approaches, depth and camera motion are estimated to reconstruct one frame from another	Does not require dense ground-truth depth and can exploit the temporal information available in image sequences	Relies on assumptions such as photometric or geometric consistency; Moving objects, non-rigid deformation, illumination changes, specularities and pose errors can affect the results
Unsupervised Domain Adaptation	A model trained on a source domain is adapted to an unlabelled target domain	Can improve model applicability to a new domain without requiring target-domain annotations	Performance depends on the similarity between domains and the effectiveness of the adaptation strategy

The mentioned methods and approaches are not necessarily mutually exclusive. For instance, SfM and SLAM are not inherently deep-learning methods. Their classical implementations rely on geometric operations such as the vaguely mentioned feature matching, triangulation, pose estimation and optimisation through geometry [41]. However, deep learning can be incorporated into these frameworks to replace or complement individual components. For example, neural networks can be trained to estimate depth, optical flow, image features, camera motion or other intermediate representations that are subsequently used within a geometric reconstruction pipeline. Therefore, the use of architectures such as ResNet or DispNet makes the corresponding component a deep-learning component, while the underlying reprojection, triangulation or camera-motion formulation may still be fundamentally geometric. The resulting depth and pose estimates can subsequently be used for three-dimensional reconstruction, as mentioned [34, 40].

Analysing from the perspective of the number of cameras, monocular depth can, then, be estimated by exploiting constraints across multiple observations or learned relationships between image appearance and scene geometry. In this case, when only a single camera is available, the depth estimation is more challenging because the image formation process does not directly provide the geometric correspondence between two simultaneously observed views. Several three-dimensional configurations can produce similar two-dimensional projections. Different objects at different distances may project to similar image regions, while changes in

illumination, texture, scale or viewpoint can produce similar visual patterns. Consequently, monocular depth estimation must rely on learned or explicitly modelled relationships between image appearance and scene geometry [33].

Yet, monocular depth estimation has a fundamental limitation: scale ambiguity. In monocular geometric formulations, depth and camera translation can be scaled simultaneously without changing the resulting image projection. Therefore, a system may recover the relative structure of a scene while remaining uncertain about its absolute scale [33].

Some representative approaches to monocular depth estimation for endoscopy based on the review by Niu et al. [33], are presented in Table 2.3.

Table 2.3: Monocular depth estimation methods developed for endoscopy. LoRA means "Low-Rank Adaptation".

Method	**Superv.**	**Architecture**	**Anatomy**	**Purpose**	**Pose**
Mahmood and Durr, 2018 [46]	Unsup.	CNN with a conditional random field	Gastrointestinal tract	Transfer from synthetic renderings to real endoscopic images	×
Endo-SfM (EndoSLAM), 2021[a],[b] [34]	Self-sup.	DispNet with a brightness-aware photometric loss	Colon, stomach, small intestine; ex-vivo porcine	Provide a dataset and a depth-plus-odometry method for endoscopy	✓ Reported at sequence scale: ATE, translational and rotational RPE per organ sequence
AF-SfMLearner, 2022 [47]	Self-sup.	Appearance flow, on SfMLearner	Porcine abdomen (SCARED), gastrointestinal (EndoSLAM), heart phantom (Hamlyn), ex-vivo porcine torso (SERV-CT)	Survive brightness inconsistency between frames	✓ Reported at 5-frame scale: ATE averaged over 5-frame snippets, each aligned separately, so sequence-level drift is not exposed
MonoLoT, 2024 [48]	Self-sup.	Loss framework (point-matching via frozen LoFTR + batch image shuffle) on top of Monodepth2/Lite-mono/MonoViT backbones	Colon (C3VD, SimCol), upper gastrointestinal tract: pharynx, middle-upper body, antrum	Recover depth in low-texture mucosa for robotic visual servoing	(✓) Estimated, not reported: seven depth metrics only. Trajectory ground truth for the robotic demonstration comes from an external electromagnetic tracker
LGIN, 2024 [49]	Unsup.	Hybrid CNN and pyramid vision transformer, dual head	Colon: clinical colonoscopy from 80 patients, plus synthetic	Estimate dense depth and odometry simultaneously	✓ Reported: ATE, translational and rotational RPE on synthetic data; SSIM and PSNR of warped frames on clinical data, which has no ground truth

Method	Superv.	Architecture	Anatomy	Purpose	Pose
IID-SfMLearner, 2024 [50]	Unsup.	ResNet18 encoder; pose subnetwork after Monodepth2	Porcine abdomen (SCARED), abdomen and heart phantom (Hamlyn)	Decompose into albedo and shading so illumination change does not corrupt the loss	✓ Reported: ATE for three sequences of the SCARED dataset: 0.0462 (448 frames) or 0.0559 (406 frames) or 0.0638 (693 frames) appear in the results
EndoDAC, 2024 [51]	Self-sup.	Foundation model with dynamic vector-based LoRA	Porcine abdomen (SCARED), surgical (Hamlyn)	Adapt a large pre-trained depth model to endoscopy at low cost	✓ Reported: ATE on two sequences. The pose network additionally estimates the camera intrinsics
Surgical-DINO, 2024[c] [52]	Supervised	DINOv2 with LoRA adapters	Porcine cadaver abdomen (SCARED, da Vinci Xi)	Adapt a general visual foundation model to surgical depth	× no pose network exists in the architecture
SfMDiffusion, 2024 [53]	Self-sup.	Conditional diffusion model	Colon	Apply generative diffusion to self-supervised endoscopic depth	✓ Reported: Camera Rotation Accuracy and Camera centre Accuracy

[a]The work by Ozyoruk et al. [34] declares to present an unsupervised algoritm.

[b]Ozyoruk et al. [34] either present a depth estimation and 3D reconstruction pipeline and algoritm and a dataset named EndoSLAM

[c]Tabulated by Niu et al. [33] as self-supervised. The publication states that scale-invariant depth loss and gradient loss serve as the supervision constraints, and reports no photometric objective.

Methods developed for conventional imaging environments face several challenges when directly applied to endoscopic scenes, either due to ethical concerns, the scarcity of labelled training data due to lack of time for annotations or impossibility of including sensors or other cameras, or either due to the variability according to patients, tool properties and situations, such as tissue deformation, specular reflections, changes in insufflation, physiological motion, rapid changes in viewing angles, the lighting on the tissue following the camera's movement and other imaging artefacts and the repetitive texture of some anatomical surfaces. Furthermore, biological tissue is non-rigid and can deform during the procedure, violating the rigid-scene assumption commonly adopted by multi-view geometric methods [33]. In addition, these depth estimation methods should comply with real-time in order to be feasible and useful [40, 54].

When specifying to hysteroscopy, besides the monocular and low depth perception [4] that is shared with endoscopy, the procedure brings specific information for the acquisition of data. This is, the hysteroscope must remain sufficiently thin to be inserted into the uterine and to be comfortable for the patient, leaving limited space for integrating additional optical or depth-sensing components alongside the imaging system. Consequently, clinical hysteroscopic systems commonly rely on monocular imaging, and stereo image acquisition becomes expensive or impractical [39]. Furthermore, obtaining dense and accurate metric depth ground truth during real clinical procedures is challenging. These characteristics make learning-based monocular depth estimation an attractive alternative for recovering three-dimensional information from hysteroscopic image sequences [33].

Besides claiming that learning-based monocular depth estimation an attractive alternative, Niu et al. [33] further declare that their review highlights self-supervised methods due to the photometric and geometric consistency they present, suggesting them as a central driver in the future of endoscopic settings.

Conceptually, in a monocular self-supervised approach, consecutive frames provide multiple observations of the same scene from slightly different viewpoints. Given a target image I_t and a source image I_s, a depth network predicts a depth map D_t for the target image, while a pose network estimates the relative transformation $T_{t\to s}$ between the two camera positions. The predicted depth and estimated camera motion can be used to project pixels from the target image into three-dimensional space and then reproject them into the source image. This projection relies on the camera intrinsic matrix K, given by

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where f_x and f_y denote the focal lengths in the horizontal and vertical directions, respectively, measured in pixels, and c_x and c_y denote the horizontal and vertical coordinates of the principal point, also measured in pixels. The zero entries correspond to the assumption of zero skew between the image axes, while the bottom-right entry is a homogeneous-coordinate scaling factor and is dimensionless. The intrinsic camera matrix describes the internal geometric properties of the camera and relates three-dimensional points expressed in the camera coor-

dinate system to their corresponding pixel coordinates in the image plane. More specifically, for a 3D point $\mathbf{P}_c = (X, Y, Z)^\top$ expressed in the camera coordinate system, its corresponding pixel coordinates (u, v) are given by

$$u = f_x \frac{X}{Z} + c_x, \qquad v = f_y \frac{Y}{Z} + c_y. \tag{2}$$

This relation defines the perspective projection from the three-dimensional camera coordinate system onto the image plane.

The resulting reprojection maps the source image information to the target viewpoint, generating a reconstructed target image $\hat{I}_t$. The network parameters are then optimized by minimizing a photometric reconstruction loss between the original target image I_t and the reconstructed image $\hat{I}_t$. Therefore, the image sequence itself provides the supervisory signal required for training, without requiring a ground-truth depth map for every frame.

The effectiveness of this approach depends on several assumptions regarding the scene and the image formation process. In particular, photometric consistency assumes that the appearance of a surface point remains sufficiently similar between consecutive frames. This assumption can be violated by changes in illumination, shadows, reflections and other appearance variations. The geometric formulation also generally assumes that the scene is static or approximately rigid and that the observed changes between frames are predominantly caused by camera motion [33].

2.3.2 Depth Estimation Evaluation Metrics

The evaluation metrics that recur throughout the depth estimation literature either obtain the depth error, where a lower result is better, or the depth accuracy, where a higher result is what is wanted [33, 40, 41].

The error metrics are

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i \in \mathcal{V}} (d_i - d_i^*)^2}, \tag{3}$$

$$\text{AbsRel} = \frac{1}{N} \sum_{i \in \mathcal{V}} \frac{|d_i - d_i^*|}{d_i^*}, \tag{4}$$

$$\text{SqRel} = \frac{1}{N} \sum_{i \in \mathcal{V}} \frac{(d_i - d_i^*)^2}{d_i^*}, \tag{5}$$

$$\text{RMSE}_{\log} = \sqrt{\frac{1}{N} \sum_{i \in \mathcal{V}} (\ln d_i - \ln d_i^*)^2}, \tag{6}$$

$$\log_{10} = \frac{1}{N} \sum_{i \in \mathcal{V}} |\log_{10} d_i - \log_{10} d_i^*|, \tag{7}$$

$$\text{MAE} = \frac{1}{N} \sum_{i \in \mathcal{V}} |d_i - d_i^*|, \tag{8}$$

where d_i denotes the predicted depth at pixel i and d_i^* the corresponding ground truth. Metrics are computed only over the set $\mathcal{V}$ of valid pixels, with $N = |\mathcal{V}|$. A pixel is excluded when the ground truth is missing, when it falls outside a depth range fixed in advance, or when it lies outside the region the imaging system actually covers. The metrics that involve the logaritm have the constraints related to logaritmic behaviour.

They differ in how the error is normalised and in how heavily large errors are weighted, which is why several are reported together rather than one being chosen.

RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error) both measure the absolute difference between the predicted and reference depth values and are expressed in the same units as depth, such as millimetres or metres. However, they differ in the way errors are penalized. MAE assigns the same weight to all errors and therefore provides a more direct measure of the average absolute discrepancy. RMSE squares the individual errors before averaging them, giving greater weight to large errors. Consequently, RMSE is more sensitive to outliers and is particularly useful when large depth errors should be strongly penalized, whereas MAE is more robust to isolated large errors and can provide a clearer representation of the typical prediction error. Since both metrics are expressed in absolute depth units, they can also be more strongly affected by errors in distant regions when the same relative error corresponds to a larger absolute depth difference.

AbsRel (Absolute Relative Error) and SqRel (Squared Relative Error) normalize the depth error by the ground-truth depth, making them sensitive to errors relative to the distance of the observed surface. For example, an absolute error of 1 mm at a depth of 5 mm and an error of 10 mm at 50 mm produce the same relative error in AbsRel. This makes relative metrics particularly useful when the scene contains a wide range of depths, as they account for the scale of the error with respect to the actual distance.

MAE, AbsRel and log-based absolute errors measure the average magnitude of the prediction error, whereas RMSE and $\text{RMSE}_{\log}$ square the individual errors before averaging and subsequently take the square root. Squared-error metrics therefore assign greater weight to large deviations and are more sensitive to outliers. For example, a method that performs well over most pixels but produces substantial errors around specular highlights or occlusion boundaries may be penalized more strongly by RMSE than by MAE. Consequently, two methods may receive different rankings depending on the evaluation metric, with such differences providing information about the distribution and severity of their errors.

Logarithmic error metrics evaluate discrepancies in logarithmic space rather than directly in metric depth. Consequently, they are more closely related to relative or multiplicative errors than to absolute distance errors. For example, overestimation from 5 mm to 10 mm and from 50 mm to 100 mm corresponds to the same factor of two and therefore produces the same absolute logarithmic error. This property makes logarithmic metrics useful when relative accuracy across different depth ranges is important. The difference between $\log_{10}$ error and RMSElog lies in how the logarithmic errors are aggregated. The former typically uses the mean absolute logarithmic error, whereas RMSElog computes the root mean square of the logarithmic errors and, therefore, gives greater weight to larger relative deviations.

The error metrics above summarise the whole distribution into one number, which hides how the error is distributed across pixels.

In turn, for depth accuracy, where a higher result means a better one, threshold accuracy, or relative threshold accuracy (RTA) reports the fraction of pixels whose prediction lies within a multiplicative tolerance of the truth [33]. These metrics, commonly denoted as δ_1, δ_2 and δ_3 are obtained from

$$\delta_j = \frac{1}{N}\left|\left\{ i \in \mathcal{V} \ : \ \max\left(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}\right) < 1.25^{\,j} \right\}\right| \qquad j \in \{1,2,3\}. \tag{9}$$

The ratio is taken symmetrically so that estimation under or over are penalised equally, and the thresholds are conventionally 1.25 (δ_1), 1.25^2 (δ_2) and 1.25^3 (δ_3) [33]. Thus, δ_1 is the strictest criterion, while δ_3 allows substantially larger relative deviations. Unlike error-based metrics such as AbsRel, these metrics measure the percentage of predictions that fall within a specified accuracy range rather than the magnitude of the error itself.

This means that RTA and AbsRel can provide different and complementary assessments of the same predictions. AbsRel describes the magnitude of the relative error across all pixels, so it can be strongly influenced by a smaller number of pixels with large error. In contrast, δ metrics only determine whether each prediction falls within the specified tolerance once a prediction exceeds the threshold, the magnitude of its error no longer affects the metric. Values δ_1, δ_2, and δ_3 indicate the proportion of predictions satisfying increasingly relaxed accuracy requirements.

It is important to mention that, for meaningful comparisons between methods, the evaluation protocol should clearly specify the depth range, and any post-processing performed on the predicted depth maps, including scale alignment, and the applied criteria, such as the strategy for which image regions are considered during evaluation [33].

In addition, although these metrics provide useful information about pixel-level agreement between predicted and reference depths, they do not fully capture other relevant aspects of performance, such as temporal consistency, accuracy at the boundaries of lesions or surgical instruments, robustness to artefacts or reliability in clinical decisions involving critical size thresholds [33, 40].

2.3.3 Self-supervision Approach and Related Metrics

Self-supervised monocular depth estimation introduces additional considerations that affect the interpretation and comparability of the metrics described above. In particular, the scale ambiguity inherent to monocular depth estimation and the joint estimation of camera motion must be considered when evaluating these methods [40].

Because monocular depth estimation recovers scene depth only up to an unknown scale factor [40], a raw prediction cannot be meaningfully compared with metric ground-truth depth without first resolving this scale ambiguity. A common approach is median scaling [33, 54], in which the predicted depth map is multiplied by a scale factor that matches its median to that of the ground-truth depth. For intance, Zhou et al. [55] mentions that their way of tackling

the scale ambiguity is using a single calculated global scale.

The scaled prediction, $\hat{d}_i$, substitutes d_i throughout Equations ((3)) to ((9)) and, according to Zhou et al. [55] and as used by Shao et al. [47], is given by Equation (10), where variables d_i and d_i^* have the same meaning as in the previous section of the metrics: d_i is the predicted depth at pixel i and d_i^* the corresponding ground-truth depth.

$$\hat{d}_i = d_i \frac{\text{median}\,(d^*)}{\text{median}\,(d)}. \tag{10}$$

This procedure removes the global scale difference between the prediction and the ground truth. Consequently, the resulting metrics evaluate the relative geometric structure of the predicted depth rather than the ability of the model to recover an absolute metric scale. For example, two predictions that differ only by a global scale factor will produce identical results after median scaling.

Pose

In monocular self-supervised formulations, depth and camera motion are jointly estimated since the photometric objective cannot be evaluated without both, and the motion is therefore part of what is being assessed. The relative camera transformation is required to reproject one image into another and construct the photometric training signal.

To meet the need for camera motion data collection, camera pose is also estimated. Two commonly used trajectory metrics are the absolute trajectory error (ATE) and relative pose error (RPE) [34]. ATE is particularly relevant when the complete estimated trajectory is used to fuse individual depth maps into a common three-dimensional model, as global trajectory consistency directly affects the alignment of reconstructed points. RPE, in contrast, is useful for assessing the accuracy of local camera motion between frames and can therefore provide information about short-term motion estimation and drift [56].

2.4 Data for Endoscopic Depth Estimation

The development of depth estimation methods depends not only on the model and learning strategy, but also on the data available for training and evaluation.

2.4.1 Data Sources

The four sources described below are the ones used through literature. These data sources provide different trade-offs between clinical realism, experimental control and availability of ground truth.

- **In-vivo clinical video**

  In-vivo data consists of video recorded during real clinical procedures. It will provide the most faithful representation of the environment, presenting the actual illumination, tissue appearance, motion, fluid, smoke and instrument occlusions. Besides ethical concerns

and availability, its main limitation is the lack of accurate ground-truth depth because, as mentioned, additional depth-sensing equipment cannot normally be integrated into a clinical endoscope, making dense, registered depth measurements unavailable. Obtaining reliable ground truth for in-vivo scenes remains one of the main challenges in endoscopic depth estimation [33].

- **Ex-vivo animal tissue**

  Ex-vivo datasets typically use animal tissue, particularly porcine organs or cadavers. They provide realistic tissue appearance and optical properties while allowing additional equipment to be used to obtain ground-truth geometry. For example, SCARED provides metric depth obtained using structured-light projection [50, 51], and ex-vivo specimens are therefore widely used for quantitative validation [41]. However, the absence of physiological motion, perfusion and other properties of living tissue limits their representativeness of the clinical setting.

- **Phantoms**

  Phantoms are manufactured anatomical or geometric replicas, such as silicone, rubber or other artificial models, with the advantage of being privacy-friendly. Their known and repeatable geometry allows ground-truth measurements to be obtained under controlled conditions, while the same camera trajectory can be reproduced across experiments [47, 48]. However, their optical and geometric properties may differ substantially from biological tissue. Consequently, results obtained with simple or anatomically unrealistic phantoms may overestimate the performance of a reconstruction method [41].

- **Synthetic renderings**

  Synthetic data are artificially generated data designed to reproduce relevant characteristics of real-world environments that are usually accompanied by structured annotations (e.g., object labels, bounding boxes, or segmentation masks), which are crucial for supervised training pipelines [57]. In endoscopy, they can be obtained by rendering a virtual camera moving through a three-dimensional anatomical model in a virtual environment, using software such as Blender or Unity. Since the scene geometry, dense depth and camera pose are directly available from the simulation, synthetic data can provide complete ground truth for depth estimation and, consequently, for geometric evaluation.

Synthetic data has evolved into a strategy for training, testing and validating machine learning systems, particularly in domains where data acquisition is costly, limited or constrained by privacy regulations [58].

Despite these advantages, synthetic data also present limitations that need to be considered during their generation and use. Synthetic samples may appear visually convincing while still failing to reproduce the detailed variability and rare conditions present in real data, which can limit realism and model generalisation. Furthermore, increasing visual fidelity and variability may require additional computational resources, creating a trade-off between

realism, accuracy and computational efficiency. The configuration of the generation process can also substantially affect the quality and diversity of the resulting dataset [59].

These limitations can be addressed through several complementary strategies. Synthetic data generation should be designed to provide sufficient variability in relevant conditions, while generated data should be evaluated against the target real domain in terms of both diversity and usefulness. Approaches can be used to assess whether models trained on synthetic data generalise to real images. Detailed documentation of the data sources, assumptions and generation parameters can further support reproducibility and facilitate iterative refinement of the generation process [59].

In endoscopic applications, these considerations are particularly relevant because of the discrepancy between synthetic and clinical images, the sim-to-real domain gap: differences in texture, illumination and dynamic tissue behaviour may reduce performance when models trained on synthetic data are applied to clinical images [33].

The generation of synthetic endoscopic datasets and the main considerations regarding their domain gap, anatomical realism and validity for training are discussed in the following Chapter.

Data augmentation can be applied to the different types of datasets by introducing controlled variations such as geometric transformations, noise, illumination changes or viewpoint variations. Although augmentation can increase data diversity and improve robustness, it remains constrained by the characteristics of the original data and therefore cannot replace the acquisition or generation of fundamentally different anatomical or procedural conditions [59, 60].

Different approaches have consequently been adopted in the literature to generate or acquire suitable training and evaluation data. The characteristics of these data directly influence the type of supervision that can be used, the reliability of the resulting evaluation, and the extent to which the trained models can be applied to the target environment.

Table 2.4 presents the data source used by the studies included in Table 2.3.

Table 2.4: Data used by the endoscopy monocular depth estimation methods of Table 2.3 according to the respective data sources and datasets, together with the ground-truth source and the augmentation each paper describes. Where in-vivo data are used, supervision is instead obtained from synthetic, ex-vivo, phantom or self-supervised sources, depending on the method.

Method	**Data Source**	**Datasets**	**Ground-truth source**	**Augmentation**
Mahmood and Durr [46]	Synthetic, then real	Rendered endoscopic views; real endoscopy	Rendering	None described
EndoSLAM [34]	Ex-vivo animal, with synthetic parts	Eight ex-vivo porcine gastrointestinal organs	Optical scanner; camera trajectory from a robotic and a manual scanner	At training time: horizontal flip and random zoom-crop, both propagating the intrinsics. No photometric augmentation. The blur, fisheye, defocus and frame-rate functions released with the dataset are offline tools for robustness testing and are not part of the training pipeline
AF-SfMLearner [47]	Ex-vivo animal; phantom; in-vivo human	SCARED (ex-vivo porcine abdomen), EndoSLAM (ex-vivo porcine), SERV-CT (ex-vivo porcine torso), Hamlyn (phantom heart and in-vivo surgery)	Projector for SCARED; stereo matching for Hamlyn	None described
MonoLoT [48]	Phantom; synthetic; real	C3VD (colon phantom, clinical colonoscope), SimCol (Unity engine, 33 videos of the lower GI tract), upper-GI (3 704 real images, 11 landmarks), EndoSLAM, EndoMapper; soft-rubber upper-GI phantom for the robotic trial	Rendering for SimCol; registered phantom geometry for C3VD; electromagnetic tracker for the robotic trajectory	Batch image shuffle, vertical and horizontal, and random patch swapping between half-batches (introduced by the paper as a contribution rather than as a preprocessing step)

Method	Data Source	Datasets	Ground-truth source	Augmentation
LGIN Fan et al. [49]	In-vivo human; synthetic	Clinical colonoscopy from 80 patients; synthetic colonoscopic sequences	Rendering for the synthetic data. None for the clinical data, which is why SSIM and PSNR of warped frames are reported there instead of depth error	Not stated
IID-SfMLearner Li et al. [50]	Ex-vivo animal; phantom	SCARED (9 scenes, ex-vivo porcine abdomen), Hamlyn (porcine abdomen, heart phantom)	Projector for SCARED, in millimetres; stereo matching software for Hamlyn	Random adjustment of brightness, contrast, saturation and hue, to prevent overfitting to a particular patient or camera
EndoDAC Cui et al. [51]	Ex-vivo animal; in-vivo human	SCARED, Hamlyn	Projector for SCARED	Inherited by reference from prior work; not specified in the paper
Surgical-DINO Cui et al. [52]	Ex-vivo animal	SCARED (22 950 frames, fresh porcine cadaver abdomen, da Vinci Xi), Hamlyn	Projector for SCARED. Used as training supervision, not only for evaluation	Resize and upsample
SfMDiffusion Li et al. [53]	Various datasets (objects)	Colonoscopic	—	None described

2.4.2 How Synthetic Endoscopic Datasets are Generated

Across the reviewed literature, a pipeline recurs in terms of how synthetic datasets are generated, assembled from four stages: patient-derived geometry, texture transplanted from real imagery, rendering through a game engine and explicit modelling of the endoscopic camera.

1. Geometry from patient imaging

In what regards the anatomical surface structure, studies that use synthetic-based datasets obtain it through segmented organs from imaging, more specifically, Computed Tomography (CT).

For the VR-Caps environment proposed in 2021 by İncetan et al. [35], the three-dimensional organ models are reconstructed from DICOM CT volumes drawn from public archives, using the open-source reconstruction software InVesalius and refining the mesh in Blender to remove bone, fat and skin so that only the colon, small intestine and stomach remain. VR-caps aimed at providing a virtual simulation platform for capsule endoscopy, enabling the representation of different tissue conditions, organ types and capsule endoscope configurations, including monocular, stereo, dual-camera and 360° setups.

In 2024, Jeong et al. [61] adapted the same strategy for the colon, whose surface model is extracted from segmented CT colonography before being imported into the simulator environment, Unity, with the aim to obtain depth estimation through deep learning directly from endoscopic images. In an article published in 2026, Liu et al. [54] use synthetic colonoscopy data also from CT-based colon models, their data being used to pretrain a scale-consistent reconstruction framework.

The recurring premise is that it makes sense that the geometry, which is the part of the scene that determines the depth map, would be real and patient-derived, while only its appearance is synthesised.

2. Texture

After obtaining a basic representation of the anatomical surface, the internal appearance of the model still differs substantially from real endoscopic imagery in terms of texture, illumination, surface reflections, colour variation and optical artefacts. The anatomical surface consists only in the structure; photorealism is not obtained from the anatomical geometry alone. Instead, it must be deliberately constructed through the appearance and rendering process.

In this literature, Blender is used as a mesh-modelling and editing tool, whereas the realism of the final endoscopic appearance is achieved through additional processing.

In VR-Caps, for example, the authors construct a continuous mucosal texture from clinical images and separately extract and overlay vascular patterns. Pathological features are also incorporated, with the resulting textures mapped onto the anatomical mesh through UV mapping [35].

EndoSLAM follows a different approach by applying optical and imaging effects, including specular reflections, lens distortion, chromatic aberration and field-of-view adjustments.

Additional augmentation is applied to the generated frames through resizing, Gaussian blur, fisheye distortion, defocus blur and frame-rate variation [34].

The second strategy declines to author the appearance at all and learns it in order to address the domain gap, as explored next.

3. The domain gap and how it is addressed

No author in this literature claims that synthetic images are interchangeable with real ones. The gap between simulated and real data is commonly acknowledged, highlighting the importance of generating sufficiently realistic and representative synthetic data and carefully considering how well it reflects real endoscopic conditions [34].

The poor transfer of models trained exclusively on simulated data has been attributed to the visual discrepancy between simulated and real endoscopic images [61]. At the field level, the same limitation is identified in the literature: although rendering synthetic views from CT-derived anatomy enables the generation of large quantities of densely annotated data, it introduces a sim-to-real gap and limit generalisation to real endoscopic images in regards of, for example, differences in texture, illumination and dynamic elements such as bleeding or smoke [33].

In 2019, Rau et al. [36] proposed an alternative to closing the gap at the image level. A conditional generative adversarial network, pix2pix, was trained to map endoscopic images directly to depth, and unlabelled real frames were passed through the model alongside the synthetic training data. Because those frames carry no depth label, their predictions are excluded from a supervised term, but they remain subject to the adversarial loss, so the discriminator must still judge them. Adaptation therefore occurs implicitly, within the training of the depth network itself: the synthetic data supply the depth supervision, while the unlabelled real images supply information about the target domain through the adversarial signal. Rather than training a separate network to transform synthetic images towards the appearance of real endoscopic ones, the cGAN incorporates the characteristics of the real domain directly into the training process.

Another approach, in 2024, by Jeong et al. [61] closes the gap at the image level instead. Rather than generating new simulation data, the colon models and depth maps of EndoSLAM were reused, and a dilated-attention cycle-consistent adversarial network was trained to translate the simulated images into realistic ones, reducing the Fréchet Inception Distance to real endoscopic images from 21.62 to 8.36. A residual U-Net was then trained on the translated images paired with the original simulated depth, so that supervision is inherited from the simulator while appearance is inherited from real data.

In the VR-caps work, the response was to close the gap at the rendering stage, by modelling illumination, camera artefacts and scene debris faithfully enough to reduce the extent of adaptation required. In their work, the mucosal texture is assembled by stitching real endoscopic frames, taken from different patients and classified by region of the gastrointestinal tract, and projecting them onto the unwrapped UV map of the model in Blender to produce continuous, non-blurry mucosal walls. Vein networks are extracted from the same real images

by conversion to grayscale, median filtering and contrast-limited adaptive histogram equalisation, and applied over the base texture with randomised position, rotation and size drawn from empirically determined distributions [35]. Pathological appearance is produced from real images of the corresponding disease [35].

In 2024, the work by Walkner et al. [62] aimed to broaden the synthetic data distribution through style translation and colour augmentation. The mentioned style translation refers to modifying the visual appearance of synthetic images to introduce characteristics that are more representative of real endoscopic images while preserving the underlying structural information. Rather than applying only conventional transformations such as changes in brightness or contrast, style translation can increase the visual diversity of the synthetic domain by introducing variations in texture, colour, illumination and other appearance-related characteristics observed in clinical data. This allows the synthetic dataset to cover a broader range of visual conditions and potentially better approximate the distribution of real endoscopic images, thereby improving the model's ability to generalise to the clinical domain. The authors started from 20 000 synthetic images and generated 100 000 photo-realistic variants and evaluated their ability to support depth prediction on clinical images using a U-Net. The use of middle-level augmentation improved threshold accuracy on an unseen style from 62.4% to 68.0% and reduced RMSE from 19.82 mm to 10.89 mm, demonstrating improved generalisation to an unseen synthetic style.

Liu et al. [54] uses a strategy that keeps the synthetic data as a pretraining stage only and transfers it through the camera model rather than through appearance: a depth network pre-trained on synthetic colonoscopy is applied to real images to produce pseudo-depth labels, which an intrinsics-guided alignment module converts into a scale space defined by the known camera intrinsics of the target endoscope, so that subsequent learning on real sequences inherits a physically interpretable scale from the simulator without inheriting its appearance.

A further consideration is whether the simulated anatomy is treated as rigid or deformable. Since insufflation and instrument interaction can modify the geometry during endoscopic procedures, a static model represents only a particular anatomical state. Regarding this matter, the authors of VR-Caps, İncetan et al. [35], address this aspect because of its importance in creating a virtual environment that closely approximates real scenarios. They implement a finite element model integrated with Unity's physics engine through an interface to the SOFA framework. As the finite element method is computationally expensive, they also provide a more efficient elastic approximation, in which contact forces attenuate with the inverse square of distance and a damped spring restores vertices towards their resting positions. This approximation is intended to reproduce a deformation with a similar appearance to that obtained with the physically accurate model, while reducing computational cost.

Notably, adaptation is also performed in the opposite direction, translating real images into the simulated domain so that a model trained on synthetic data can be applied to them unchanged [61].

To conclude, there is, in short, no single established methodology. The common element is only the recognition that synthetic supervision must be paired with some explicit mechanism

for handling appearance.

4. Rendering Engine

Unity is the dominant engine, chosen for its maturity as a configurable simulation platform rather than for graphics alone [63]. It provides user-friendly tools for creating interactive and controllable virtual environments, including the possibility of simulating camera motion and recording image sequences.

In addition, synthetic data can be generated together with ground-truth annotations directly from the simulated scene, pixel-perfect depth and pose for every frame [34], since the geometry, camera parameters (such as camera pose) and object states are known within the virtual environment. For the generated ground truth to be quantitatively meaningful, the virtual scene must be defined in a metric scale, with a calibrated camera and consistent coordinate systems for the depth and camera pose. This ensures that the rendered depth maps and poses correspond to the same geometric reference frame, allowing the ground truth to be interpreted and evaluated in physical units.

Unity is the used software behind the synthetic portion of the EndoSLAM dataset, provided by Ozyoruk et al. [34] in a work that also uses the dataset to train a model, the Endo-SfMLearner. It is also behind the simulated colonoscopy images used for sim-to-real depth transfer in the work by Jeong et al. [61], and it is identified in survey form as one of the standard routes by which synthetic endoscopic datasets are produced [33].

VR-Caps integrates Unity's High Definition Render Pipeline (HDRP) to overcome the limitations of the default rendering pipeline, the Universal Render Pipeline (URP), in producing high-quality and realistic visuals. According to the authors, HDRP enables more realistic material differentiation and consistent interaction with illumination across the scene, while its shaders and post-processing effects can reproduce endoscopic characteristics [35].

Liu et al. [54] also configure the virtual camera with the calibrated intrinsics of a real colonoscope, so that the rendered imaging geometry matches the intended instrument.

Table 2.5 presents these design choices across literature.

Table 2.5: Synthetic data generation approaches in the reviewed literature.

Work	Geometry	Appearance	Engine	Domain gap
Rau et al. [36]	CT / phantom	Rendered	A simulator (not clarified)	cGAN, implicit DA
VR-caps, 2021, İncetan et al. [35]	CT (public archive)	Real frames (Kvasir)	Unity, HDRP	Realism by design
EndoSLAM, 2021, Ozyoruk et al. [34]	CT / porcine ex-vivo	Rendered	Unity	Provided as testbed
Jeong et al. [61]	CT colonography	Rendered	Unity	CycleGAN sim-to-real
Walkner et al. [62]	Surgical scene models	GAN style transfer	N/a (benefit from existent dataset)	Style augmentation
Liu et al. [54]	CT colon models	Rendered	N/a (methodology not clear)	Intrinsics-guided pseudo-labels

It is important to note that, while Table 2.4 focuses specifically on endoscopy-specific monocular depth estimation methods, Table 2.5 broadens the scope to representative approaches for synthetic endoscopic data generation, including works that are not themselves depth estimation methods. For instance, VR-caps work (İncetan et al. [35]) is not included in Table 2.4 as it is not a comparator for the depth estimation model. It does not propose an architecture, or a training procedure. Their environment is meant for supporting tasks.

2.4.3 Validity of Training on Synthetic Data

The use of synthetic data requires careful consideration to determine whether the resulting dataset is sufficiently representative and suitable for its intended purpose. A model may, for example, achieve high performance on a synthetic test split simply because it shares similar characteristics with the training data, without demonstrating that the dataset adequately represents the variability of the target real-world domain.

The following topics are ways of exploring this question and validity that were found in literature.

If synthetic data can teach the task at all

Walkner et al. [62] study, also cited above, trains on eight models, validates on one and tests on a tenth, reaching a root mean squared error below 10 mm and a δ_1 of 77.15% [62]. This establishes that the task is learnable from the data, that synthetic should not be regarded as inherently inadequate, but nothing more.

If it covers the actual clinical situation

A failure mode worth testing for explicitly is absence rather than inaccuracy. In the study by Walkner et al. [62], the predictions degrade specifically on scenes where a puncture of the

abdominal wall and an event the generator never produced are not correctly represented in the simuation set. The way to answer this topic is by understanding what a real procedure contains, list everything and checking that list.

If its variation generalises

If several appearance configurations are used during simulation to generate the dataset, one configuration can be excluded entirely from training and reserved for testing. This provides a way of assessing whether the variation represented in the training data is sufficient to generalise to an unseen appearance. Data augmentation can also influence this generalisation and should therefore be considered.

This was investigated by Walkner et al. [62], where authors generated images in five styles, trained the model on four of them and tested it on those four, achieving a δ_1 of 67.14% and an RMSE of 10.86 mm. The same model, when tested on the fifth style, a style never seen before, yielded a δ_1 of 62.43% and an RMSE of 19.82 mm. This substantial degradation indicates that the model did not generalise equally well to an appearance that was not represented during training, highlighting the effect of the visual domain gap.

The authors further investigated data augmentation, including variations in colour and image transformations such as cropping and adjustments to brightness, contrast and saturation. With moderate augmentation, performance on the unseen style improved to a δ_1 of 68.01% and an RMSE of 10.89 mm. In particular, the RMSE decreased by approximately 45% compared with the non-augmented model on the unseen style. This suggests that increasing the visual diversity of the synthetic training data can substantially improve generalisation to appearances not encountered during training.

Overall, these results suggest that the original appearance configuration separation for testing may not provide sufficient visual diversity for robust generalisation across different styles. The improvement obtained with moderate augmentation indicates that expanding the range of appearances represented during training can help compensate for this limitation. However, one cannot conclude that augmentation eliminates the domain gap, also because these experiments were performed on synthetic styles rather than real clinical images and therefore do not establish that the same degree of generalisation would be achieved in the clinical domain.

Understand how generation choices matter

The changes in the simulated environment, if adjustable, can be measured and its influence can be evaluated. The İncetan et al. [35] environment grades each optical effect (depth of field, vignetting, chromatic aberration and more) across five intensity levels explicitly so that a network's sensitivity to them can be quantified. This converts the design of the generator into an experiment: rather than asserting that an effect was worth simulating, one measures what its presence or absence does.

Closeness between synthetic and real data

The distance between the synthetic and real image distributions can be measured directly, independently of any depth model. The Fréchet Inception Distance (FID) is the measure used, which compares the distributions of feature representations extracted from two sets of images, and it is sensitive enough to rank generation strategies. FID relies on features extracted by InceptionV3 and evaluates the similarity of image distributions rather than the usefulness of the data for a particular task, meaning that a low FID does not necessarily imply accurate geometry or improved depth estimation. It is a property of the data, not of a trained model, which is what makes it useful in this regard. A lower FID generally indicates greater similarity between the distributions [64].

However, the obtained value should not be interpreted as an absolute measure of realism or compared directly across studies, since it depends on the image sets and evaluation conditions used. Thus, FID values obtained from different studies should not be considered directly comparable, since differences in the real and synthetic datasets, anatomy, number of images and acquisition conditions can substantially affect the result. In addition, FID is sensitive to the number of images and estimates based on a small number of images can be unstable and tend to produce higher values. The metric may also be strongly influenced by differences in field of view, vignetting or camera framing between the two domains. Consequently, a high FID may reflect discrepancies in these global image characteristics rather than differences in tissue appearance or texture alone [64]. Therefore, a real-to-real comparison using the same sample size can provide a useful baseline for interpreting the magnitude of the synthetic-to-real difference.

The Jeong et al. [61] colonoscopy pipeline falls from 21.62 for the raw renderings to 10.68 after a standard translation network and 8.36 after an improved one.

Result for real images

Real images lack of ground truth, making direct quantitative evaluation difficult. In some cases, an alternative reference or pseudo-ground truth allows quantitative comparison. For example, a depth estimator trained exclusively on synthetic renderings showed a 78.7% improvement in structural similarity on real porcine colon images after domain adaptation [46].

More commonly, however, performance on real images must be assessed through indirect or qualitative criteria, such as anatomical plausibility, spatial and temporal consistency, reprojection or clinically measurable quantities, as discussed in the following Subsection (subsection 2.4.4). Consequently, results obtained on synthetic test data demonstrate performance under the simulated conditions, whereas claims about performance in the real clinical domain must be interpreted according to the type and strength of evidence available.

2.4.4 Validation on Real Data when Training on Synthetic Data

The methodological consequence of the absence of dense depth ground truth in real images is stated most plainly in a study that trained a depth network entirely on synthetic laparoscopic

data and then applied it to real cholecystectomy video, evaluating the results for the real images qualitatively [62].

Anatomical spatial plausibility

One way of assessing depth predictions on real images is to verify whether they are consistent with the expected spatial arrangement of the anatomy. Although the true depth of each pixel is unknown, the relative position of anatomical structures can often be inferred from the scene, for instance, a structure visibly located in the foreground should generally be predicted as closer to the camera than a structure located behind it. This allows qualitative expectations to be defined before inspecting the prediction and subsequently checked against the estimated depth.

This approach was used by Walkner et al. [62] when evaluating predictions on clinical laparoscopic images. The liver, located in the foreground, was expected to correspond to a relatively shallow depth, while the abdominal wall in the background was expected to have greater depth. The predicted depth followed this relationship. Similarly, the ligament was expected to appear at a plausible intermediate depth and to remain distinguishable from the abdominal wall behind it, which was also observed in the predictions.

Such checks provide evidence that the estimated depth is spatially consistent with the observed anatomy, but they do not establish its metric accuracy. Importantly, the expected spatial relationships should be defined before examining the predictions, rather than inferred from them afterwards. This makes the assessment more objective and allows predictions that contradict the expected anatomical relationships to be identified, while still recognising that this remains a qualitative form of validation in the absence of ground truth.

Spatial plausibility in clinical measurement

A further possibility is to evaluate the clinical usefulness of the estimated geometry by comparing a physical measurement derived from the image with an independently known reference. For instance, the estimated size of a polyp can be compared with its actual dimensions measured after resection. This approach evaluates whether the estimated depth provides sufficiently accurate geometry for a clinically relevant task, rather than directly validating the depth map itself. Niu et al. [33] study deployed a system that reported a relative accuracy of 89.9% against a reference measurement, compared with 54.7% for endoscopists. However, this result does not demonstrate that the individual depth estimates are metrically accurate; instead, it shows that the information derived from the estimated geometry can be useful for a clinically relevant measurement.

Spatial and temporal consistency

Two further self-checks require no labels at all. An image can be divided into parts and each part predicted separately, with agreement between the parts and the whole indicating that the prediction is driven by scene content rather than by framing [62]. In addition, sucessive frames

can be compared to understand if the previsions are stable and coherent in time. Both will detect instability, but will not return a quantitative error nor be able to tell if it is a consistent one.

Photometric self-consistency - Reprojection

As depth and pose are estimated together, when applicable, one frame can be warped into another using both and compared with the frame actually recorded. The structural similarity index and peak signal-to-noise ratio between the original and the warped image are then reported as the quantitative result on clinical data, which is the protocol adopted for the clinical portion of one recent colonoscopy study precisely because no ground truth exists for it [49]. However, it is a measurement of internal consistency because a depth and pose pair can be both innacurate but the analysis may reproject correctly and score well.

Controlled perturbation

A more informative test asks not whether the prediction is right but what it is sensitive to. This would be met by adjusting the RGB values of the clinical images so that their mean and standard deviation match those of the synthetic training set, and then comparing the before and after. It isolates how much of the discrepancy is attributable to colour alone. In the study by Walkner et al. [62], the difference proved minimal, which supports the conclusion that the colour component of the domain gap can be absorbed by simple augmentation.

Substituted ground truth

Another possibility is to use the ground truth result for other studies as reference, although this does not necessarily provide true ground truth. In Li et al. [50], for example, the depth provided with the Hamlyn dataset was obtained using stereo matching software. Evaluation against such data therefore measures agreement with another estimation method rather than accuracy with respect to a direct measurement.

Summary

None of the six measures depth accuracy on a patient. Plausibility and the two consistency checks detect instability; controlled perturbation diagnoses sensitivity; substituted ground truth measures either inter-algorithm agreement or pose alone; and spatial plausibility in clinical measurement validates the application. A results chapter should therefore say which of these it is reporting and should not present a structural similarity figure computed on warped frames as though it were a depth error.

2.4.5 Considerations

The review of synthetic data generation clarifies and establishes a number of points that are particularly relevant when adapting these approaches to endoscopy and, more specifically,

hysteroscopy.

When the anatomical geometry is derived from medical imaging, its patient-derived origin is an important advantage, as it provides a physically grounded basis for the resulting depth labels, since the ground-truth depth is determined by the geometry alone of the reconstructed surface. In turn, appearance influences what the network observes, not directly determining the geometric aspect. This means that inaccuracies in the segmentation or reconstruction of that surface directly affect the validity of the depth labels and it is important to mention the results in performing segmentation.

When thinking about the anatomical sources used in the reviewed literature, the gastrointestinal tract is the most commonly represented anatomy among the reviewed endoscopic datasets and the imaging technique from where models are gathered is usually CT colonography. For the uterine cavity, however, the standard imaging context is different, as transvaginal ultrasound is more routinely used to characterise the uterus than CT. This way, exposion to the ionising radiation that a CT-based pipeline would imply is also avoided. Using ultrasound-derived anatomy would therefore provide a more appropriate modality for this application while preserving the methodological principle identified in the literature.

With regards to the domain-adaptation approaches that incorporate real images into the synthetic data generation or adaptation process, they introduce an additional consideration. Although real images can provide valuable information about clinical appearance and help reduce the sim-to-real gap, their use may also introduce privacy and regulatory constraints, particularly when patient-derived images are involved. Consequently, the benefit of incorporating real images must be considered alongside the requirements associated with their acquisition, storage and processing.

As mentioned in the validity of the simulated environment, producing realistic synthetic data therefore requires an understanding of the anatomical structures and of the conditions under which they are observed. In this work, this understanding is supported by the prior analysis of hysteroscopy and the analysis of the real videos from CHUSJ. These considerations provide the basis for selecting and configuring the elements represented in the simulated environment.

A further consideration applies to photometric evaluation based on consecutive endoscopic frames. Since consecutive frames may be highly similar, a high SSIM (Structural Similarity Index Measure) between a reprojected frame and its target does not necessarily indicate that the estimated depth and pose produced an accurate reprojection. Consequently, photometric metrics computed on consecutive frames should be interpreted as measures of reconstruction consistency rather than direct evidence of depth accuracy, particularly when camera motion between frames is small, or consecutive frames for analysis should not be as similar, if applicable.

The remaining aspects identified in the literature, including approaches for reducing the sim-to-real domain gap, complementary strategies for validating models trained on synthetic data and methods for evaluating predictions on real images are addressed in the methodology chapter.

Nevertheless, even when multiple complementary validation approaches are adopted, the absence of direct empirical validation may remain a source of uncertainty and skepticism among practitioners and regulators.

2.5 3D Reconstruction

Depth estimation is not the final objective of this work, but rather one of the means through which three-dimensional information can be recovered from hysteroscopic images. A depth map describes the distance of the observed surface from the camera for a single frame. However, an individual depth map corresponds to a single frame in the coordinates that the camera was when acquired it. It provides only a partial observation of the scene. Recovering the third dimension from endoscopic video is described in the literature not as a technical refinement in itself but as the foundation for a range of downstream capabilities, transforming the endoscope from a purely visual device into a quantitative sensing platform [33].

A further objective of monocular depth estimation is consequently to integrate depth information across multiple frames to obtain a three-dimensional reconstruction. The main objective of three-dimensional reconstruction is therefore to combine the information obtained from multiple observations into a common spatial representation of the environment. In this context, depth estimation provides the geometric information for each frame, while camera motion provides the information required to position the individual observations relative to one another. The resulting model can subsequently be used to provide additional information about the spatial organisation, shape and relative position of observed structures, assess completeness of an examination, or support other spatially informed tasks [33, 39, 65].

3D reconstruction inherits the limitations that already existed at the time of depth estimation, which is why the properties of a depth estimate that matter in this scenario are not only its per-pixel accuracy but its consistency across frames. Thus, difficulties such as establishing reliable correspondences between observations due to smooth or weakly textured surfaces, repetitive structures and specular reflections, biological tissues being non-rigid with the possibility to deform during a procedure, and changes in illumination due to the light source tipically moving together with the camera, also affect the 3D reconstruction, either directly compromising the estimation of depth and camera motion or, even when the individual depth maps appear accurate, through errors in camera pose or inconsistencies between consecutive depth estimates, which can cause the individual observations to be incorrectly positioned during fusion. A depth estimation method that performs well when evaluated frame by frame may not necessarily produce a geometrically consistent three-dimensional reconstruction [39, 54, 66].

2.5.1 From Depth and Camera Motion to a 3D Model

Once a depth map and the corresponding camera pose are available, the observed pixels can be transformed into three-dimensional points [67]. For a pixel (u, v) with predicted depth $D_t(u, v)$, and camera intrinsic matrix $\mathbf{K}$, its position in the coordinate system of frame t can

be obtained through back-projection,

$$\mathbf{X}_t = D_t(u, v)\mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{11}$$

The resulting three-dimensional point is initially expressed in the camera coordinate system of the frame from which it was obtained. If $\mathbf{T}_{t\to w}$ represents the transformation from the camera coordinate system at frame t to a common world coordinate system, the point can be transformed,

$$\mathbf{X}_w = \mathbf{T}_{t\to w}\mathbf{X}_t. \tag{12}$$

Repeating this process for the frames of the sequence produces a set of three-dimensional points expressed in a common coordinate system. The individual point sets can subsequently be fused into a single representation, according to

$$\mathcal{P} = \bigcup_{t=1}^{N} \mathcal{P}_t, \tag{13}$$

where $\mathcal{P}_t$ represents the three-dimensional points reconstructed from frame t and N is the number of frames [67].

The most direct result of this process is a point cloud. In a point cloud, the observed surface is represented by a collection of three-dimensional points [41].

Point clouds are particularly useful because they preserve the spatial information obtained from the individual observations without requiring an explicit surface representation. Nevertheless, they can also be converted into other representations when a continuous surface is required. A mesh, for example, represents the reconstructed surface through vertices, edges and polygons and can be generated from a point cloud using surface reconstruction methods. Alternatively, volumetric approaches such as truncated signed distance functions (TSDFs) represent the scene within a discretised volume, where each voxel stores information about its distance to the nearest surface. A surface mesh can subsequently be extracted from a volumetric representation [39, 68]. Other representations, including surfels and implicit neural representations, have also been investigated for three-dimensional endoscopic reconstruction [39, 68].

The choice of representation depends on the intended application. Point clouds provide a relatively direct representation of the reconstructed observations and are convenient for geometric analysis and coverage assessment. Meshes provide an explicit continuous surface, while volumetric representations such as TSDFs are well suited to incremental fusion of observations [39].

According to the analysis in Yang et al. [39], the probability of using point clouds in rigid scenarios is 81%, in instrument scenes is 25% and in non-rigid scenarios is 38%, while mesh is chosen by 19% of articles with non-rigid scenarios and TSDFs by 17% in the same scenario.

As disadvantages, Yang et al. [39] note the presence of noise and holes for point clouds, high complexity for meshes and a moderate accuracy for TSDFs.

In turn, Neural Radiance Fields (NeRFs), an example of implicit neural representations, represents a scene using a neural network that learns a continuous function relating spatial coordinates and viewing direction to scene properties such as density and radiance. This enables the scene to be represented continuously and allows novel views to be synthesised from the learned representation. This means that, in contrast to the previous options, it provides a continuous learned representation that can simultaneously capture scene geometry and appearance, making them particularly attractive for complex or deformable environments. However, neural representations introduce different challenges, such as a poor generalisation [39].

2.5.2 Scale Ambiguity and Scale Consistency

A fundamental issue in monocular reconstruction is the scale ambiguity already described in Section 2.3.3. A monocular image sequence does not, by itself, determine the absolute metric scale of the reconstructed scene, as depth and camera translation may be multiplied by any common positive factor without altering a single image projection [33, 54].

The ambiguity is therefore a single unknown for the whole sequence, and a property of the projection rather than a shortcoming of any method. Assigning an independent factor to each frame resolves nothing, since it introduces one unknown per frame. It is justified only if each frame carries its own scale, which is precisely the failure that scale consistency describes rather than the ambiguity that motivates the correction [33, 34].

That failure has a direct consequence for reconstruction. If consecutive frames observing the same anatomical surface are predicted at substantially different scales, their point clouds have different spatial dimensions; when transformed and fused using the estimated camera poses, the corresponding surfaces appear displaced from one another instead of coinciding, even where each individual depth map is accurate in relative terms [69].

It follows that scale ambiguity and scale inconsistency are distinct problems, and that only the first is resolved by rescaling. A single factor applied uniformly to every frame supplies the missing unknown from outside the video, such as from a reference model or from known camera intrinsics, and the reconstruction becomes metric. This means that this factor itself is a result, reporting how far the prediction stood from the true scale [54]. For instance, Ozyoruk et al. [34] refer in their article that they apply geometry consistency loss in their algoritm for scale-inconsistency between consecutive frames.

2.5.3 Evaluation of 3D reconstruction

Depth metrics and reconstruction metrics are not the same measurement, and a model can score well on the first and fail at the second. For example, errors in camera pose can misalign otherwise accurate point clouds, while inconsistent depth scales can prevent observations from different frames from being correctly fused. In addition, a reconstruction may be geometrically

accurate where points are available but fail to cover a substantial part of the anatomical surface.

A review of quantitative evaluation in laparoscopic reconstruction involved forty-four studies and found that, whatever the reconstruction method, assessment reduces to the same operation, which consists in the reconstruction error is the Euclidean distance between each reconstructed surface point and the closest point on the reference surface, aggregated most commonly as mean absolute error or root mean squared error, even though mean relative error, standard deviation, RMSE log, absolute relative and squared relative error are also used [41]. Across the surveyed methods, machine-learning approaches combined with stereo imaging, and stereoscopy combined with structured light, achieved the best accuracy, in the submillimetre range [41].

The critical qualification is that these geometric evaluations require a ground-truth reference model [41]. Placing physical markers is an option, but, as Yang et al. [39] also mentions, not in this scenario, as it is not easy to place them manually in an endoscope environment. The ground-truth reference model is major challenge in clinical endoscopy, where the true three-dimensional geometry of the observed anatomy is generally not directly available during image acquisition.

To overcome the lack of direct references, several works have relied on external anatomical models and registration techniques. For example, in Cui et al. [70], a point cloud of the nasal endoscope is reconstructed using SfM and MVS (Multi-View Stereo) and subsequently registered with a CT reconstruction model using fast point feature histograms (FPFH). Similarly, in Qiu and Ren [71], a point cloud generated using SLAM is registered with a CT-based model to support navigation in minimally invasive surgery. Other works have investigated the registration of endoscopic reconstructions with CT or preoperative anatomical models, including approaches based on point-to-plane ICP (Iterative Closest Point) for registering reconstructed point clouds with kidney models [72], as well as methods combining CT reconstruction and non-rigid registration [73].

However, these approaches do not provide a direct ground-truth measurement of the intraoperative anatomy, particularly in the presence of deformation and non-rigid motion.

An alternative line of research has focused on extending the observable field through image mosaicing and stitching rather than explicitly reconstructing the complete three-dimensional anatomy. Although mosaicing can substantially increase the visual coverage of the anatomy, it primarily provides an extended two-dimensional representation and therefore does not directly provide the three-dimensional geometric reference required for quantitative point-cloud evaluation. Several works have investigated image stitching based on improved pose estimation, feature matching and registration techniques [39, 74].

Several approaches have been specifically designed to handle non-rigid deformation between preoperative models and intraoperative images. These include particle-filter-based estimation of the optimal similarity transformation of patient-specific models [75]. Other methods have also been proposed to estimate dense deformation fields from sparse feature correspondences and compensate for non-rigid motion during image stitching [76]. Other approaches have combined multispectral endoscopy and video stitching to construct panoramic views of anatomical

regions [39, 74].

More recently, implicit neural representations have also been investigated for endoscopic and surgical reconstruction, particularly in dynamic and deformable environments. Wang et al. [77] proposed a dynamic neural radiance field for representing deformable surgical scenes, using tool masks and depth information to guide the reconstruction. Other approaches have employed multiple neural fields to separately model surface dynamics, shape and texture, while Yang et al. [78] represented the spatial and temporal dimensions of the surgical scene using a 4D representation [39]. These approaches demonstrate the potential of neural representations for modelling complex and dynamic surgical environments. However, some of these methods still depend on accurate geometric supervision, such as disparity or depth information, highlighting that the absence of reliable ground-truth geometry remains an important challenge even for more recent reconstruction paradigms.

Despite these advances, clinical endoscopy remains limited by the absence of a directly observable ground-truth geometry. CT-based registration provides an external reference but requires correspondence estimation and may not accurately represent the intraoperative anatomy under deformation. Mosaicing increases visual coverage but does not directly provide a three-dimensional geometric reference, while neural representations introduce alternative ways of modelling dynamic scenes but may still require geometric supervision [35, 39, 74].

Surface and coverage metrics

Before any surface distance can be computed, the reconstructed points and the reference must be expressed in a common coordinate system, and the procedure used to achieve this is part of the measurement. A rigid alignment, over $SE(3)$, corrects an arbitrary choice of world frame while leaving the size of the reconstruction untouched. A similarity alignment, over Sim(3), additionally estimates a scale factor; it is the standard choice when reconstructions are obtained from structure from motion and are therefore defined only up to scale, and it is typically initialised from a coarse alignment and refined by minimising the squared distance between corresponding points [79].

Having $\mathcal{P}$ as the reconstructed point cloud and $\mathcal{R}$ as the reference surface, the distance from a reconstructed point p to the closest point on the reference surface can be defined as

$$e_p = \min_{r \in \mathcal{R}} \|p - r\|_2 , \qquad p \in \mathcal{P} \tag{14}$$

The resulting point-to-surface distances can be aggregated using, for example, the mean absolute error and root mean squared error, as shown in

$$\mathrm{MAE}_{3D} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} e_p, \tag{15}$$

$$\mathrm{RMSE}_{3D} = \sqrt{\frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} e_p^2}. \tag{16}$$

MAE for this three-dimension describes the average magnitude of the geometric error,

whereas RMSE gives greater weight to larger deviations. Both metrics are expressed in the units of the reconstructed scene, such as millimetres. Reporting both can, therefore, distinguish a reconstruction with a generally small error from one containing a smaller number of substantial geometric failures [41].

In addition, coverage is particularly relevant in endoscopy because a two-dimensional video only records the surfaces visible from the current viewpoint. Regions that are never brought into the field of view remain unobserved and cannot be identified from the video alone. A three-dimensional reconstruction can instead provide a spatial representation against which observed and unobserved regions can be identified. This principle has been investigated in endoscopic reconstruction and coverage assessment, including simulation-based studies in which the known geometry provides a reference for evaluating the reconstructed coverage [33, 35, 65].

Its value should be justified based on the spatial resolution of the reference model, the expected reconstruction accuracy or the requirements of the intended application. The sampling density of the reference surface should also be controlled, since a non-uniform or sparse reference representation may introduce a bias in the estimated coverage.

Considering the work by Knapitsch et al. [79], besides the point-wise error metrics such as MAE and RMSE, the quality of a reconstructed point cloud can be evaluated in terms of precision and recall, which provide complementary information on the coverage of the region.

Trajectory consistency

The camera trajectory is another important component of reconstruction because it determines the position of each frame in the common coordinate system. Knapitsch et al. [79] presented in 2017 a benchmark for image-based 3D reconstruction in all of its applications and reinforces this by mentioning the camera trajectory information as an advantage that video has over image collections.

Therefore, the pose metrics introduced in Section 2.3.3, namely the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE), are also relevant when assessing the reconstruction pipeline. ATE evaluates the global consistency of the estimated trajectory with respect to the reference trajectory, whereas RPE evaluates the accuracy of relative camera motion over a specified temporal interval [56].

On comparing reported values

The endoscopic reconstruction Yang et al. [39] survey from 2024 mentions that the data used by each method is essentially self-designed and that the evaluation process and metrics are not unified. In turn, Göbel et al. [41] find in their review, also from 2024, that the wide variation in reported error is attributable in large part to differences in the validation scenario. Authors of the latter propose a standardised evaluation protocol to enable more meaningful comparisons. For example, using publicly available datasets or ex-vivo porcine organs with independently acquired ground truth, and reporting consistent metrics such as reconstruction RMSE.

2.5.4 Other uses of Depth Estimation

Although the reconstruction of the anatomical environment is the main focus of this section, following the broader path from depth estimation to three-dimensional mapping, analysis of surface morphology and spatial relationships between anatomical structures, 3D reconstruction also has other potential applications. Niu et al. [33] summarizes the application in terms of its clinical goal and key technology, as shown in Table 2.6.

Table 2.6: Representative clinical application areas of endoscopic depth estimation, their clinical objectives and enabling technologies, according to Niu et al. [33].

Application Area	**Clinical Goal**	**Key Technology**
Surgical Navigation	Support the avoidance of anatomically critical structures, improve the completeness of surgical procedures, register pre-operative and intra-operative information	SLAM, SfM, AR
Polyp Detection	Improve the identification of adenomatous lesions and decrease the likelihood of polyps being overlooked, particularly when they present as flat lesions	3D Visualisation, Mucosal Mapping
Quantitative Metrology	Enable objective estimation of lesion dimensions to support risk assessment and the planning of subsequent surveillance	Scale-Aware 3D Reconstruction
Invasion Depth Assessment	Determine whether early cancerous lesions are confined to the mucosa or extend into the submucosal layer, supporting the selection of an appropriate treatment	Video-based AI with Geometric Features
Procedural Quality	Verify that the mucosal surface has been adequately examined and identify areas that may have been insufficiently inspected	Real-time 3D Coverage Mapping
Risk Prediction	Estimate the likelihood of procedure-related complications	Machine Learning with Geometric Features

2.5.5 Practical and Clinical Deployment Concerns

Göbel et al. [41], in an article from 2024, argue that, beyond reconstruction accuracy, methods should also be assessed according to practical criteria such as computational cost, real-time capability, the need for additional hardware, compatibility with existing clinical equipment and overall implementation cost. These aspects are particularly relevant for clinical deployment, where a method that achieves high accuracy but requires specialised hardware or excessive computational resources may have limited practical applicability.

Similarly, in the same year, the article by Yang et al. [39] also highlights that need through the analysis on real-time performance as an important consideration for endoscopic depth estimation, declaring that monocular approaches are generally computationally simpler and can achieve high frame rates on modern GPUs (Graphics Processing Units), while stereo methods

require additional computations, such as cost-volume construction. It mentions a work on deep neural networks, saying that architectural and implementation optimisations have also made real-time stereo estimation feasible. Thus, when reflecting of real-time performance, also explore computational costs. However, it contradicts in a way with the previous article, as it seems that research is moving in the direction of those practical criteria.

2.5.6 Considerations

The motivation for not adopting a supervised approach becomes clear when considering that the objective extends beyond depth estimation to three-dimensional mapping. The reconstruction and subsequent coverage mapping require both the depth of the observed scene and the relative motion of the camera between consecutive frames. A supervised depth model would provide depth estimates, but would not for itself provide the camera pose required to place these estimates consistently within a common three-dimensional coordinate system. An additional pose estimation method would therefore be necessary, introducing a separate component whose predictions would need to be combined with those of the depth network and potentially leading to scale inconsistencies between the two estimates. In contrast, monocular self-supervised Structure-from-Motion formulations, for example, jointly estimate depth and relative camera motion, using their geometric consistency to generate the training signal.

This is precisely why, in the context of this work, evaluating only depth-related metrics such as MAE, RMSE, AbsRel and δ would not be sufficient. Pose evaluation complements depth evaluation and is directly related to the final objective of three-dimensional reconstruction.

Regarding the assessment aspect, and the need to have a GT, this is precisely where a synthetic environment offers a significant advantage: a simulated cavity can be generated from known three-dimensional geometry, providing an exact reference surface against which the reconstructed point cloud can be quantitatively compared.

With reference to the evaluation metric values reported in the literature, the review by Göbel et al. [41] includes a graph presenting the distribution of MAE and RMSE values reported for different methods. However, the exact values are difficult to determine from the graph, as the results are distributed over relatively broad intervals of approximately 2 mm. The only exact value is the one the review mentions, a MAE value of 0.15 mm for SfM. However, this value is distinguished from AI-based SfM, suggesting that the reported method does not correspond to the type of deep learning-based Structure-from-Motion approach considered in this work, which combines DispNet and ResNet. Therefore, the reported results are not directly comparable. Furthermore, the analysis concerns the laparoscopic setting, which introduces an additional limitation to the direct comparison of these results with those obtained in hysteroscopic reconstruction.

2.6 Adapting Pre-trained Weights and Fine-tuning

Regarding the training strategies employed, the literature does not follow the same methodology. Table 2.7 presents the variations identified across the reviewed studies. Table 2.8

and Table 2.9 explore the configurations for training either of the algoritms that do not use pre-trained weights and of the ones that use.

Table 2.7: Depth estimation training strategies across the reviewed literature for endoscopy. Details for the configuration are given in Tables 2.8 and 2.9.

Method	**Starting point**	**Adaptation and freezing**	**Objective**	**How the adaptation is assessed**
Mahmood and Durr [46]	None; trained solely on synthetic renderings	Reverse adaptation: the real image is transformed into a synthetic-like representation at inference, so the network itself is never adapted	Depth and topographical reconstruction	Measured before and after adaptation, which quantifies the domain gap itself
Rau et al., 2019 [36]	None; pix2pix trained from scratch	Implicit adaptation: unlabelled real frames pass through the generator, excluded from the supervised L_1 term but retained in the adversarial loss	Depth for colonoscopy, polyp measurement and navigation	Compared with and without the unlabelled real images in training
AF-SfMLearner, 2022 [47]	SfMLearner	Two training stages: the optical-flow network is trained first, then frozen while depth, pose and appearance-flow networks train together	Depth and ego-motion under brightness inconsistency	Transfer to two further datasets without fine-tuning, presented as evidence of generalisation
Surgical-DINO, 2024 [52]	DINOv2	Freezes the image encoder entirely; optimises only inserted low-rank adaptation layers and a depth decoder. Freezing is justified by memory and compute cost, not by overfitting	Surgical depth, towards 3D reconstruction, navigation and AR	Compares three regimes: zero-shot, decoder-only fine-tuning and low-rank adaptation, reporting that fine-tuning a decoder head alone is not sufficient to transfer visual features to geometric relations
EndoDAC, 2024 [51]	Depth Anything	Dynamic vector-based low-rank adaptation with convolutional neck blocks; justified by reduced memory and convenient deployment	Endoscopic depth for navigation, surface reconstruction and AR	Superior performance reported with fewer training epochs and without knowledge of the camera intrinsics

Method	Starting point	Adaptation and freezing	Objective	How the adaptation is assessed
IID-SfMLearner, 2024[b] [50]	ResNet18 pre-trained on ImageNet	No freezing; the pre-trained encoder trains together with the rest	Depth robust to illumination change, towards reconstruction and AR	Equalises the initialisation across compared methods, stating that all but one began from an ImageNet pre-trained encoder
Jeong et al., 2024 [61]	No model inherited; the data of EndoSLAM is reused	Adaptation at the image level: a dilated-attention CycleGAN translates simulated images towards real ones, after which a residual U-Net trains from scratch	Depth from monocular endoscopy	Compares a standard translation network against their own, and supervised against self-supervised depth networks
Walkner et al., 2024 [62]	None; U-Net trained from scratch	No adaptation to the clinical domain at all	Laparoscopic depth	Entire appearance style excluded from training, which measures the dataset rather than the model
BodySLAM, 2026 [66]	Zoe, a metric depth foundation model	Declines fine-tuning explicitly, to preserve the generalisation that motivated the choice of model	Surgical visual SLAM; a 3D map	The absence of adaptation is presented as the contribution, validated across three surgical domains
Liu et al., 2026 [54]	A pre-trained foundation backbone	No hard freeze; layer-wise rates hold the encoder nearly fixed. Two steps: supervised pretraining on synthetic data, then joint depth and pose on real data	Scale-consistent 3D reconstruction of the colon	Compares supervised and self-supervised transfer baselines with 95% confidence intervals, and ablates the first training step against the second

Table 2.8: Training configuration of the reviewed methods for endoscopy that do not adapt pre-trained weights.

Method	**Training data**	**Optimiser and learning rate**	**Epochs**	**Overfitting**	**Reported result**
Mahmood and Durr [46]	200 000 rendered images with ground-truth depth	SGD, momentum 0.9; Learning rate initialized at 0.00001, decrease by 20% every 20 epochs	300	Epochs with least log10 error were selected to avoid it	Structural similarity on a real porcine colon improved by 78.7%
Rau et al., 2019 [36]	More than 16 000 rendered images with depth, plus unlabelled real frames	Adam, 2×10^{-4} for both generator and discriminator, batch 20; inference under 0.04 s per image on a Titan V	300; reduced to 40 for the phantom experiment	Explicit, and anatomical: the epoch count was cut "as the network starts to over-fit to the shape of the patient CTC"	–
AF-SfMLearner, 2022 [47]	SCARED, split 15 351 / 1 705 / 551	Adam, $\beta_1 = 0.9$, $\beta_2 = 0.99$, batch 12; initial 1×10^{-4}, multiplied by 0.1 after 10 epochs, so ending at 1×10^{-5}	20 + 20, in two stages	Not discussed; regularisation enters through smoothness and auxiliary loss terms	AbsRel 0.059 on SCARED; point-cloud accuracy 3.174

Method	**Training data**	**Optimiser and learning rate**	**Epochs**	**Overfitting**	**Reported result**
IID-SfMLearner, 2024 [50]	SCARED, split 15 351 / 1 705 / 551, following Shao et al. [47]; also Hamlyn	Adam, $\beta_1 = 0.9$, $\beta_2 = 0.999$, batch 8; initial 1×10^{-4}, multiplied by 0.1 after 10 epochs, so ending at 1×10^{-5}; RTX 3090	30	Explicit: brightness, contrast, saturation and hue are randomised "to prevent overfitting and ensure that the training data distribution remains unbiased across specific patients or cameras"	AbsRel 0.171 under transfer
Jeong et al., 2024 [61]	21 850 frames sampled from EndoSLAM, of which only 1 034 were used for training, matched with an equal number of real images	Translation network: Adam, 2×10^{-4}, $\beta_1 = 0.5$, $\beta_2 = 0.99$, batch 16. Depth networks: Adam, 2×10^{-4}, batch 16	300 for the translation network, of which the checkpoint at 220 was used; 100 for the depth networks	Not verified; the checkpoint was selected at the minimum generator loss rather than at the end of training	FID 21.62 $\rightarrow$ 10.68 $\rightarrow$ 8.36
Walkner et al., 2024 [62]	20 000 unique images from ten models, translated into five styles for 100 000 in total; eight models train, one validates, one tests	Static, dynamic and cyclic schedulers compared; the dynamic schedule gave the best average accuracy and RMSE. Absolute rate not stated	Not stated	Addressed by augmentation at three intensities and measured by holding out an unseen style; moderate augmentation reduced the unseen-style error by approximately 45%	Seen style RMSE 10.86 mm; unseen 19.82 mm; with augmentation 10.89 mm
BodySLAM, 2026 [66]	No training on the target domain	Not applicable	Not applicable	Avoided by not adapting: the generalisation of the pre-trained model is the stated reason for leaving it unchanged	–

Table 2.9: Training configuration of the reviewed methods that adapt pre-trained weights to the endoscopic domain, as according to Niu et al. [33] and the original publications. Reported results are not comparable between rows: the reference data, the units and the alignment protocol differ in every case.

Method	**Training data**	**Optimiser and learning rate**	**Epochs**	**Overfitting**	**Reported result**
Surgical-DINO, 2024 [52]	split into 15351/1705/551, following Shao et al. [47]	AdamW, initial 1×10^{-5}, weight decay 1×10^{-4}, batch 8	50	Not discussed; the encoder is frozen for reasons of memory and deployment cost rather than of overfitting	AbsRel 0.053, δ 0.975[a]
EndoDAC, 2024 [51]	15351/1705/551, following Shao et al. [47]; 1.6 million trainable parameters, or 1.6% of the total	Two AdamW optimisers, for the depth and pose networks separately, both at initial 1×10^{-4}, batch 8	20	Not discussed; the small trainable fraction is justified by memory and deployment	AbsRel 0.051, δ 0.981[a]; ATE 0.0741 and 0.0512
Liu et al., 2026 [54]	SimCol 25 942 / 2 882 / 9 009; C3VD 6 504 / 1 730 / 1 781; clinical 5 980 / 1 256 / 1 290	Step 1: AdamW, layer-wise, 3.22×10^{-6} for the pre-trained encoder against 1.61×10^{-4} elsewhere, weight decay 0.1, batch 8. Step 2: Adam, 1×10^{-4}, halved every 10 epochs, batch 8	Step 2: 20, with the encoder warmed up over the first two. Step 1 not stated	Addressed structurally, by holding the encoder nearly fixed rather than by regularisation	AbsRel 0.0953; MLS 0.2084; Sim(3) scale 0.73–0.87; mucosal coverage 56.2%

From the methods presented in Table 2.7, two close the gap in image space: Jeong et al. [61] translates synthetic images towards real ones before training, and Mahmood et al. [46] translates real images towards synthetic ones at inference, which allows clinically relevant information to be preserved while bridging the source and target domains in the reverse direction. One closes it in the training objective, the work by Rau et al. [36], admitting unlabelled real frames to the adversarial loss while excluding them from the supervised term. AF-SfMLearner [47] and IID-SfMLearner [50] both inherit only an ImageNet pre-trained encoder, which is universal practice rather than domain transfer. And two apply no adaptation at all: Walkner et al. [62], which trains from synthetic data and is applied to clinical video unchanged, and BodySLAM [66], which declines adaptation as a stated design choice. They are explored in greater detail in Table 2.8, allowing a narrow focus on fine-tuning methods and allowing the different procedures that adapted pre-trained weights to be analysed as a whole.

The three further explored in Table 2.9 begin from a model learned on natural images and restrict what is updated, whether by, for instance, a freeze or by low-rank adaptation. It is important to note that none adapts an endoscopic model to a different anatomy.

When analysing Table 2.9, it is understood that, to adapt to pre-trained weights, algoritms used a few hundred of image sequences. Two of them used the same amount of some of the algoritms that trained from the start, as they adopted the same SCARED split of 15 351 training frames, but, aside from those, the three from Table 2.9 used less than the others from Table 2.8.

For the learning rate, the interval was from values at 10^{-6} to values at 10^{-4}, while Table 2.8 shows that 10^{-4} is more common for this parameter when training does not adapt from pre-trained weights. Liu et al. [54] used 3.22×10^{-6} for the pre-trained foundation backbone, this is, for the pre-trained weights; 1.61×10^{-4} was only used for the remaining layers. The batch size was the same for the three methods.

2.6.1 Considerations

Adapted methodologies give rise to several considerations, also regarding the training parameters.

Regarding the amount of data, the reviewed studies generally had access to datasets specifically available for training and adaptation. In the present work, however, the amount of data available for training is constrained, which may limit the number of samples that can be used for adaptation.

Learning rate is another relevant consideration. During pre-training, models typically use relatively larger learning rates, whereas adapting a pre-trained model generally requires lower values to avoid substantially modifying the knowledge already encoded in the pre-trained parameters. For example, values in the order of 10^{-5} to 5×10^{-5} have been suggested for fine-tuning in some settings [80], fine-tuning being one specific form of adapting a pre-trained model, in which its pre-trained parameters are further optimised for a new task or domain. This makes sense, as what is wanted is learned representations are to be preserved while still allowing adaptation to the target domain.

Batch size is also subject to computational constraints. Small batch sizes, such as 8 or 16, require less memory and may therefore be more suitable for machines with limited computational resources, whereas larger batch sizes can increase training efficiency when high-end GPUs or TPUs (Tensor Processing Units) are available. This may constitute a constraint in the present work, as access to high-end computational resources is limited.

The number of epochs is similarly relevant, as longer training may allow the model to adapt more extensively to the target data but can also increase the risk of overfitting. A larger initial number of epochs combined with early stopping can therefore be considered as a strategy for determining when further training no longer provides improvements [81].

When it comes to the results shown in this section, they are obtained under different training protocols. Therefore, when analysing the results as a whole, this must be taken into consideration, as metrics such as AbsRel may not be directly comparable. One example is that a fine-tuned model and a zero-shot model evaluated on the same frames cannot be considered to be operating under the same experimental conditions. Even when both models are evaluated on identical test data, the fine-tuned model has been adapted to the target domain, whereas the zero-shot model has not. Consequently, differences in their results may reflect not only the underlying methods, but also the different training regimes. Niu et al. [33] addresses this issue by explicitly identifying the training regime of each method and by separating in-domain results from transfer results into distinct blocks. This also motivates the reason why the results reported in the tables above are not analysed extensively as a direct performance ranking.

In addition to the previous considerations, freezing parameters changes the training regime being evaluated. For example, freezing the encoder preserves the features learned during the original training, whereas fully fine-tuning the model allows all parameters to adapt to the new data. Thus, a frozen-encoder model may achieve a higher error on the target domain than a fully fine-tuned model, while potentially retaining more general features that transfer better to other domains. Consequently, the two strategies should not be compared solely on the basis of their in-domain error. A meaningful comparison should also consider performance on data outside the domain used for fine-tuning, or explicitly acknowledge that the comparison is limited to in-domain performance.

Additionally, adapting pre-trained weights, for example through fine-tuning, may improve the reported depth errors without necessarily improving the model's ability to estimate depth at the correct absolute scale.

Finally, another consideration applies specifically to the present work and should not be overlooked. The published fine-tuning results cited above adapt within endoscopy, and usually within a single anatomical region, not involving a change that would bring changes in anatomy, imaging geometry and image origin. This means that literature does not provide a direct precedent for a domain shift as proposed in this work.

Wang et al. [40] argued that the data-driven nature of deep learning models often requires them to be retrained on specific datasets before deployment, and that reducing the need for such retraining across patients with the same or similar anatomical regions could substantially accelerate the adoption of artificial intelligence (AI) in medicine. This may support further

exploration of adaptation from pre-trained weights, including fine-tuning, as a means of reducing the need for training models from scratch. However, their argument does not necessarily imply that such adaptation can be expected to remain effective under large domain shifts. The scenarios considered may instead involve more limited changes, such as the differences between patients within the same anatomical region, which are already considerable, or between closely related anatomical sites.

2.7 Literature Review Discussion

The literature reviewed in this chapter establishes the main foundations for monocular depth estimation and three-dimensional reconstruction in endoscopic environments.

Given the constraints of conventional hysteroscopy, particularly the limited space available for additional sensing hardware, monocular approaches are especially relevant. Among the reviewed methods, self-supervised monocular approaches that involve Structure from Motion (SfM) are particularly suited to this application, as they can estimate depth and camera motion from single-camera image sequences, providing the information required for subsequent 3D reconstruction.

For the dataset, the reviewed literature is predominantly focused on gastrointestinal and abdominal anatomies, and no monocular depth estimation method or synthetic dataset specifically developed for the uterine cavity was identified. When the dataset is synthetically made, there are some concerns, such as how does it generalise to real images, and strategies, such as data augmentation with the real images.

Although transfer learning has been explored to adapt pre-trained models to new domains, no directly comparable adaptation to the uterine cavity procedure was identified. These gaps motivate the development of a dedicated synthetic hysteroscopic dataset and the investigation of whether a pre-trained endoscopic depth estimation framework can be adapted to hysteroscopy, which, despite being an endoscopic procedure, presents distinct imaging characteristics.

The review also highlights that frame-level depth accuracy alone does not fully describe reconstruction performance. Common depth metrics depend on scale alignment, and per-frame scaling can conceal inconsistencies between consecutive predictions that become relevant when depth maps are fused into a common 3D coordinate system. Camera-pose accuracy and consistency are likewise important for maintaining the reconstructed geometry.

Therefore, the present work investigates not only monocular depth estimation but also the subsequent fusion of depth predictions into a three-dimensional representation, providing a basis for assessing whether the adapted model can produce geometrically consistent reconstructions of the uterine cavity and be able to obtain a coverage mapping or auxiliate in other computed-assisted tasks.

Chapter 3

Methodology

The present work explores three-dimensional reconstruction in hysteroscopic imaging, with the broader aim of investigating its potential to support future improvements in hysteroscopic procedures and related clinical applications.

The proposed approach combines monocular depth estimation with camera pose estimation by fine-tuning an existing strategy developed for endoscopic environments, so that a coverage mapping of the uterine cavity is possible. This formulation allows the estimated depth maps and camera poses to be progressively integrated to reconstruct the observed uterine cavity, providing a basis for future coverage mapping, which was beyond the scope of the present work. In addition, the reconstruction of the cavity can be analysed and potentially used for other purposes, such as lesion detection.

An overview of the methodology is provided in Figure 3.1.

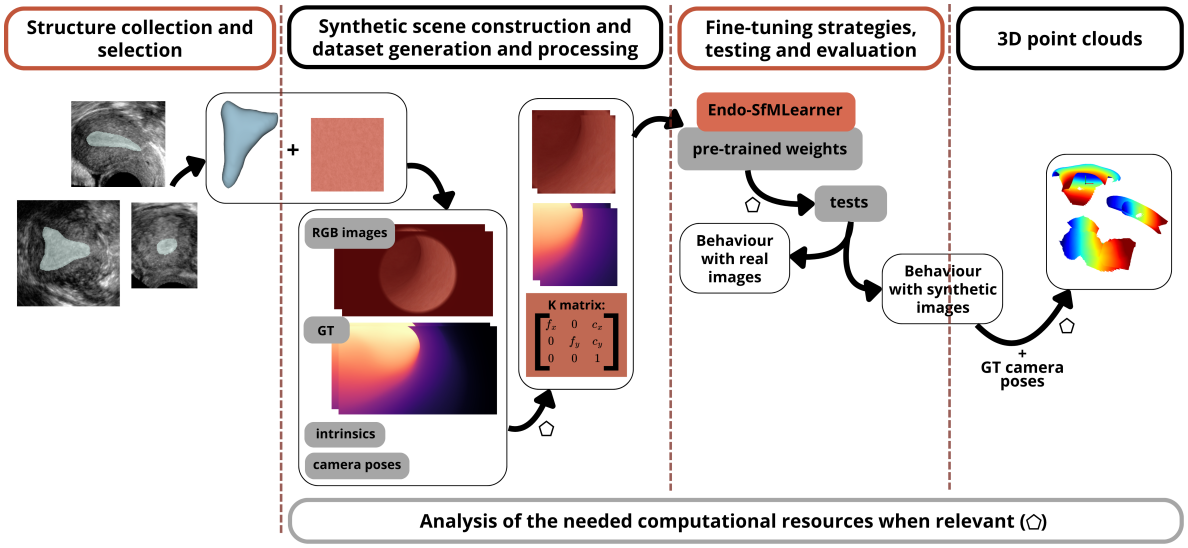


Figure 3.1: Overview of the methodology for the approach in the present work.

A controlled simulated hysteroscopic environment is used, in which both the underlying three-dimensional structure of the scene, derived from ultrasound segmentation, and the visual observations can be controlled and recorded. This provides access to ground-truth depth and camera poses, enabling the quantitative assessment of the estimated depth obtained through the self-supervised strategy, as well as the resulting three-dimensional reconstruction and point cloud generation. Following this controlled evaluation, the behaviour of the proposed approach on real hysteroscopic images is also explored to assess its potential to generalise beyond the synthetic domain, while acknowledging the limitations imposed by the quality and characteristics of the available real-world data. In addition, the computational performance of the

processing steps that would be relevant to clinical practice is analysed to assess the feasibility of their use in a real-time setting.

Figure 3.2 shows the workflow in terms of main steps and the corresponding used softwares.

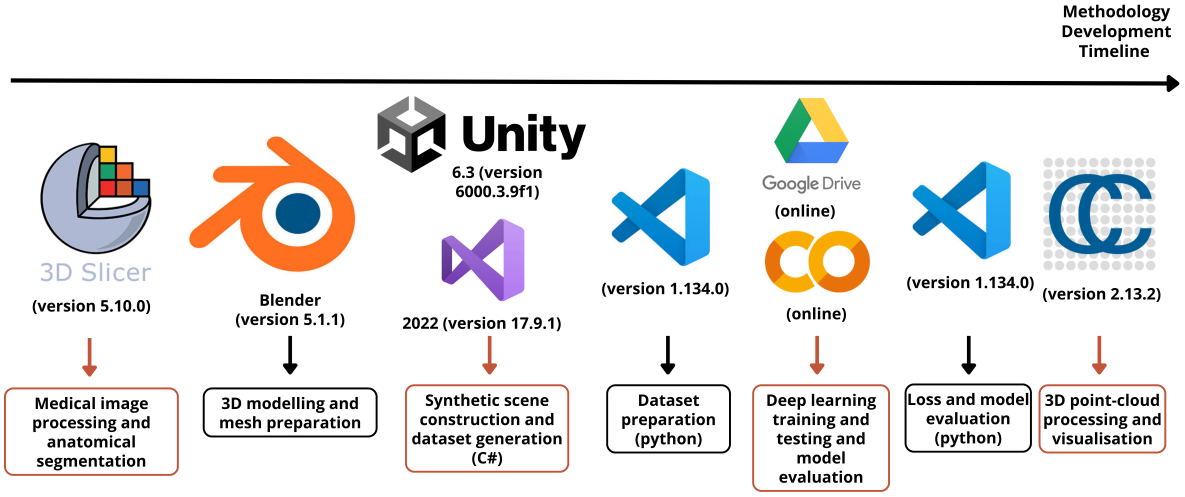


Figure 3.2: Workflow of the implemented methodology, showing the steps performed with each software.

The following sections describe the construction of the simulated environment and dataset, the selection and adaptation of the depth estimation approach, the training and evaluation procedures, and the subsequent reconstruction of the scene from the estimated depth and camera motion. Practical considerations such as time and computational resources are also explored.

3.1 Dataset

The first step in this work consisted in the establishment of the dataset.

3.1.1 Dataset Establishment

Having ground truth (GT) is important in this context because it allows the results to be evaluated quantitatively rather than only qualitatively, even when the depth estimation algorithm itself is not supervised. In other words, although GT is not required by the algorithm during its training or operation, it is needed to provide a reference against which the estimated depth can be compared.

As mentioned, real in-vivo images do not have GT for depth estimation, nor is it generally feasible to obtain or annotate it directly. Its acquisition would require the involvement of auxiliary imaging techniques, such as multi-view data fusion [33], while, in contrast, tasks such as lesion detection can obtain their GT by annotations of experienced physicians that identify the regions of interest [82]. Furthermore, the quality of a dataset should not be assessed solely based on the accuracy of its annotations. It should also be considered in terms of whether the data adequately represent clinically relevant variability and whether the findings obtained from them are expected to remain applicable beyond controlled experimental conditions [33].

If the images were acquired using a binocular setup, obtaining depth information would be more feasible, as stereo endoscopes can recover 3D spatial information from the relative positions of their lenses and their known intrinsic parameters. In such systems, depth estimation primarily relies on determining the disparity between corresponding points in the two views through stereo matching and subsequently converting this disparity into depth [33]. However, the hysteroscopic images considered in this work are monocular.

As identified in the literature review presented in the previous chapter, a particularly suitable approach would be to record ex-vivo real intrauterine tissue while preserving, as closely as possible, the anatomical configuration of the uterus. Ideally, an experienced professional would perform the navigation using a physical instrument designed to reproduce the characteristics and motion of a hysteroscope, while a camera system simultaneously captured the scene and provided ground-truth depth information. The resulting depth data could then be appropriately preprocessed to enable their use for training and quantitative evaluation. Although this approach would not fully reproduce the tissue deformation that occurs during a real procedure, the use of ex-vivo tissue would provide a more controlled and similar environment. The work by Ozyoruk et al. [34] followed a similar strategy but only collected data from gastrointestinal organs, none from the uterus.

The remainder of the literature reflects the same limitation, with no identified dataset providing comparable ground-truth depth data specifically for the intrauterine cavity, as summarised in Table 3.1.

Table 3.1: Overview of existing datasets for depth estimation, disease classification, polyp recognition, segmentation and pose tracking based on Ozyoruk et al. [34].

Dataset	**Modality**	**Findings**	**Organs**	**Tasks**	**Size**
Kvasir-SEG	Capsule endoscopy	Polyps	Colon	Segmentation	1000
Kvasir	Standard endoscopy	Z-line, pylorus, cecum, esophagitis, polyps, ulcerative colitis, dyed lifted polyps and dyed resection margins	Colon	Disease detection	6000
Kvasir	Standard endoscopy	Lifted polyps and dyed resection margins	Colon	Segmentation	2000
Hamlyn Centre Datasets	Laparoscopy	Polyp	Colon	Segmentation	7894
	Laparoscopy	None	Kidney	Disparity	40,000
	Laparoscopy	Polyp	Colon	Polyp recognition	2000
	Laparoscopy	None	Liver, ureter, kidney, abdomen	Localisation, tissue deformation, tracking	No size information
KID Dataset	Capsule endoscopy	Angioectasias, aphthae, chylous cysts and polypoid, vascular and inflammatory lesions	Small bowel and colon	Classification	2448
NBI-InFrames	Standard endoscopy	Angioectasias, aphthae, chylous cysts and polypoid	Larynx	Classification	720
EndoAbs	Standard endoscopy	–	Liver, kidney, spleen	Classification	120
ASU-MAYO Clinic	Standard endoscopy	Polyp	Colon	Segmentation	22,701
ROBUST-MIS Challenge	Laparoscopy	Rectal cancer	Abdomen	Segmentation	10,040

Another possible alternative would be to use synthetic data or physical phantoms. These approaches are also represented in the literature, including in the work of Ozyoruk et al. [34], which considered all three types of data acquisition.

As shown in Table 3.1, other datasets have likewise relied on synthetic environments, physical phantoms, or a combination of these approaches. In the context of the present work, the development of a suitable physical phantom would itself require an initial synthetic model to define and validate its anatomical geometry. The use of a synthetic environment therefore provides a natural first step towards such a solution.

Moreover, a sufficiently detailed virtual environment can reproduce not only the anatomical geometry but also relevant visual characteristics, including surface textures, illumination, and material appearance, allowing the simulated images to approximate the visual conditions encountered during hysteroscopy while retaining direct access to the underlying three-dimensional structure. Physical phantoms, in contrast, introduce additional sources of discrepancy between the simulated or physical model and biological tissue, as their optical, geometric and mechanical properties may differ from those of real biological tissues due to the silicone printing. These differences would be relevant in an imaging-based application, where surface appearance and light interaction with the tissue directly affect the acquired images.

Furthermore, a previous work developed in association with SYSTEC, such as the present one, provides an additional practical limitation to the use of a physical phantom in the context of this project. The study reported that the available RGB-D camera, selected based on its availability and its capability to simultaneously provide RGB and depth information, had to record from an approximate distance of 40 cm, due to the minimum functional range of the Intel RealSense D435i in the considered configuration, as illustrated in Figure 3.3. Such a configuration would therefore not reproduce the close-range navigation and camera movements characteristic of hysteroscopy, limiting its suitability for the specific objective of this work.

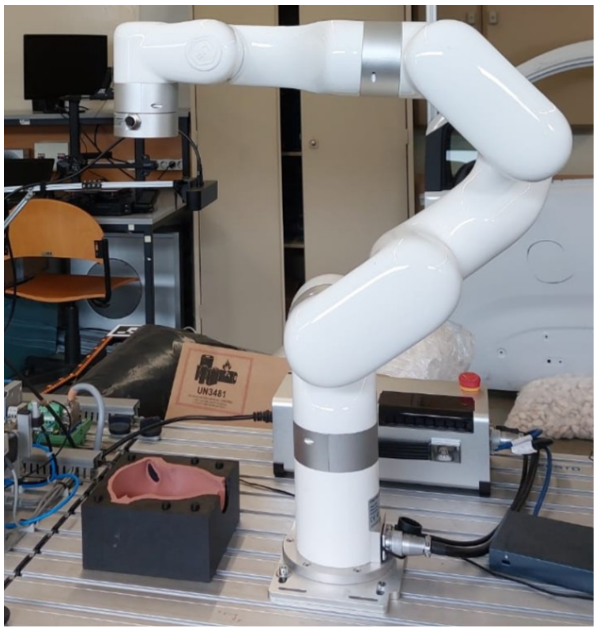


Figure 3.3: Established setup for the phantom data acquisition in the work by Fernandes [37].

Considering these constraints, a synthetic environment was selected aiming to replicate the real environment and its characteristics as closely as possible. In the present work, synthetic data are therefore used for quantitative evaluation, as the known three-dimensional geometry provides the reference required to assess both depth estimation and the resulting reconstruction. It is important to notice that results obtained in the rendered environment establish that the the method works, or not, where its assumptions hold. The reported figures in Chapter 4 should be read in that light.

For this work, two videos from different patients at Centro Hospitalar Universitário São João (CHUSJ) were also made available. These videos do not capture the clinically relevant variability discussed in the previous chapter, and only a limited subset of the available frames represents the anatomical region of interest. The implications of these limitations, as well as the possibility of using these videos either than in the reconstruction generalisation assessment, are further discussed.

Additionally, a few images illustrating the presence of pathology, less than fifteen, were collected from the works by Zhang et al. [83], Takahashi et al. [84], and Yang et al. [85], and could also be used for testing. Their results are of particular interest since they contain a salient and distinct anatomical structure such as a polyp. They cannot be used for training, not only because of their limited number, but also because they lack the temporal continuity required by the monocular depth estimation strategy. In this case, the evaluation also cannot be based on quantitative measures, since information such as the dimensions of the polyps and their distance from the camera across all their points is not available.

The authors of the articles whose those images were collected state that their work is licensed under CC BY 4.0, with the corresponding link: creativecommons.org/licenses/by/4.0/. They allow use, sharing, adaptation, distribution and reproduction in any medium or format, as long as it is given the appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. In this caso, the changes are the merging of images to illustrate appearance diversity and the crop, resizing and use for testing the 3D reconstruction.

Therefore, although the collected in-vivo clinical images cannot be used for training, they still provide significant value as far as possible for the development of the simulated environment. They are used to understand and reproduce relevant visual features, such as the details of the intrauterine tissue and camera movement, while also enabling a qualitative analysis, albeit on a limited amount of data, of how the fine-tuned approach behaves on data representing the target clinical domain.

3.1.2 Synthetic Dataset Generation

The dataset generation started with the acquisition of the uterus underlying structure, followed by its adaptation to the uterine cavity and the reproduction of relevant real-world conditions, such as camera reflections, together with the recording of the required data. The process is described in Figure 3.4.

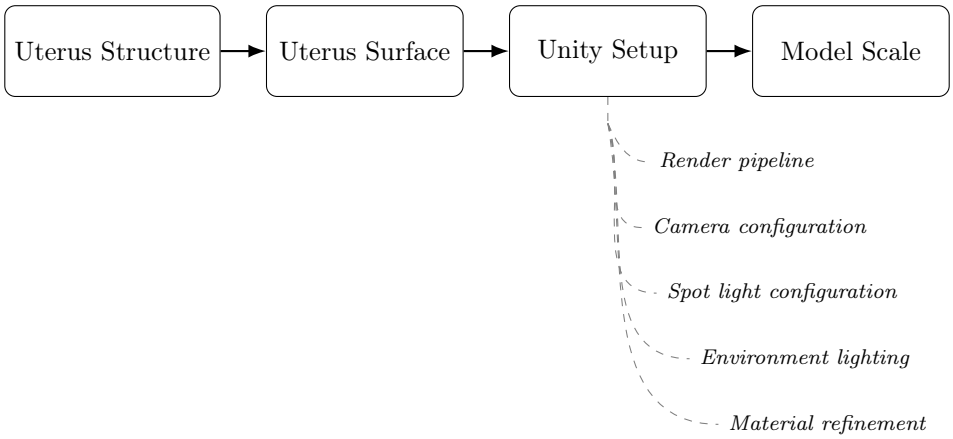


Figure 3.4: Overview of the synthetic dataset generation process.

Uterus Structure

As discussed in the previous chapter, an anatomical basis for the target organ is required and, in the literature, this is commonly achieved using CT images, as in the work of Jeong et al. [61], where the three-dimensional model of the colon to be used in simulation was generated from CT images from where the colon was segmented. This approach is appropriate for their application. For the uterine cavity, however, the standard imaging modality is transvaginal ultrasound (TVUS), which is also commonly used as an initial imaging examination before hysteroscopy. Therefore, ultrasound-derived anatomy provides a more appropriate modality for this application while preserving the methodological principle identified in the literature.

Two archives were analysed, the "NLM Visible Human Project" [86] and the "Cancer Imaging Archive" [87], together with the dataset from the work by Boneš et al. [88], named UterUS. Some of their characteristics are presented in Table 3.2.

Table 3.2: Comparison of the Imaging Resources considered for the 3D model construction. CT means "Computed Tomography", PT means "Positron Emission Tomography", MRI means "Magnetic Resonance Imaging", CR means "Computed Radiography", US means "Ultrasound", RT-STRUCT means "Radiotherapy Structure Set".

Resource	**Uterus?**	**Image Modality**	**Patient Condition**	**Segmentation**	**Segmentation Details**
NLM Visible Human Project [86]	Not specified, only that collects from one female cadaver	MRI, CT and anatomical cross-section images	No information	No	–
Cancer Imaging Archive [87] - option A	Yes	CT, PT, MRI, CR	Pathological	No	–
Cancer Imaging Archive [87] - option B	Possibly, it specifies "organs from the whole body"	MRI, CT, RT-STRUCT	Healthy	One contains expert contours, but frame by frame	–
UterUS dataset [88]	Yes	US	Mixed (authors excluded as criteria history of pregnancy, endometrial polyps, unicornuate uterusendometrial polyps, presence of acquired uterine abnormalities such as myomas)	Yes	Annotations and analysis (average Dice coefficient of 0.863 ± 0.043 on external data)

As the Table 3.2 shows, the UterUS dataset was the only one with performed and available segmentation results for the region of interest. It is under the license CC BY-NC-SA 4.0. In addition, it presents quantitative results for the obtained segmentations, which means that there was a GT to compare with, confirmed analysing their files. Those GT manual annotations were the ones used for this project. However, while considered as the GT, they may still contain inaccuracies arising from human interpretation. Nevertheless, in the context of this project, such inaccuracies are not expected to have a substantial influence on the results, given the intended use and the characteristics of the data.

The "annotations" file was read into 3D Slicer 5.10.0, where 101 of the 141 annotated volumes, which were NIfTI Files, were possible to analyse in the three axis and in a 3D view,

after defining a threshold for the relevant region.

All volumes imported into 3D Slicer were visually and qualitatively assessed. Examples of the models included in the dataset are presented in Figure 3.5. The figure provides a visual organisation of some of the available models according to their main observed characteristics, facilitating their identification and selection for subsequent analyses. They were collected from similar angles and organized by their intrauterine 3D structure (low to high) and their observed curvature (low to high). These were not the only observed characteristics: additional anatomical features can also be identified in the models, such as the openings of the Fallopian tubes.

For each model, the corresponding model identifier is displayed in the bottom-right corner of that model's image and their group classification in the up right. The latter is provided by the dataset authors and divides into "General Population - G" (women from 18-28 who have never been pregnant), "Unexplained Infertility - I" (at least 6 months of unexplained infertility despite normal semen analysis, tubal patency and ovulation), "Recurrent Miscarriages - M" (2 or more spontaneous miscarriages with normal karyotype and absence of antiphospholipid syndrome or thrombophilia) and "Recurrent Implantation Failure - RIF" (2 or more implantation failures).

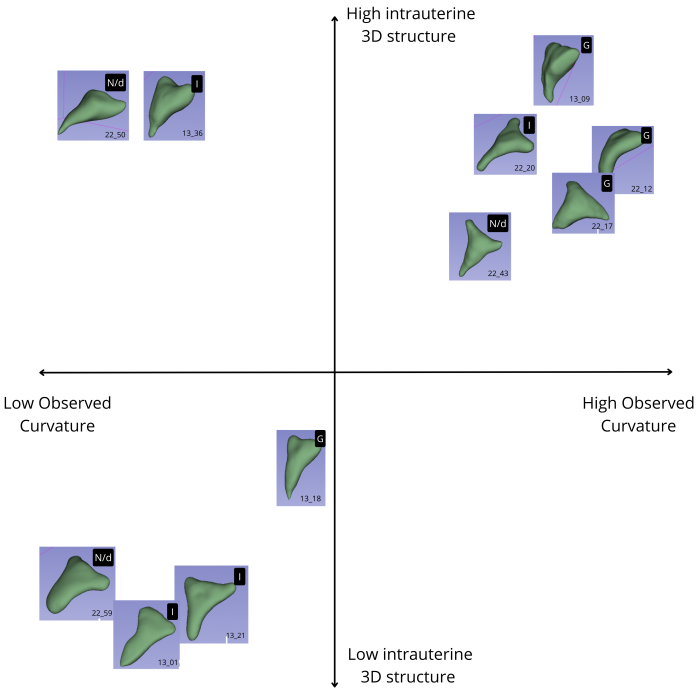


Figure 3.5: Scheme for some of the uterus models from the UterUS dataset [88], organized by their intrauterine 3D structure (low to high) and their observed curvature (low to high). Each model has its identification and group: G (women from 18-28 who have never been pregnant), I (at least 6 months of unexplained infertility despite normal semen analysis, tubal patency and ovulation) and N/d (not defined). Images were acquired from the 3D Slicer software.

For the data collection process, the objective was to obtain data that are both representative of the clinical application and sufficiently diverse, at the same time avoiding excessive repetition, which could lead to overfitting and limit the ability of the model to generalize to

previously unseen configurations.

Although aspects such as illumination, colour, reflections and more will be considered in greater detail in a later stage to help in this, the variability was already taken into account during the selection of the anatomical models. Consequently, rather than relying on a single uterine model, four different models were selected. This also allowed for the test to be applied to a different anatomical structure of the ones used for the training data. The selection of the structures was based on two main criteria:

- **Anatomical differences between the models**

  The models were selected according to the differences highlighted in Figure 3.5. These included variations in the shape and configuration of the uterine cavity, particularly in the appearance and positioning of the openings towards the Fallopian tubes, as well as differences in curvature and overall three-dimensional structure. This variability was intended to prevent the training data from being dominated by a single anatomical configuration.

- **Visibility and appearance within the simulation environment**

  The models were also assessed based on how their structures appeared from the simulated hysteroscope viewpoint. In particular, models were preferred when the openings towards the Fallopian tubes were not immediately visible from the initial camera position, at the entrance of the intrauterine cavity. If these structures were visible from the beginning, the field of view would be larger and images would not have that many traits, even when the camera was moving. Conversely, when the path towards these structures was longer for the camera to be able to se them, it was possible to collect a larger number of frames in which the observed scene progressively changed. This resulted in greater variation in the visual information available across consecutive frames, while maintaining a realistic progression of the camera through the uterine cavity.

The chosen models were the ones shown in Figure 3.6 and exported in the STL file format.

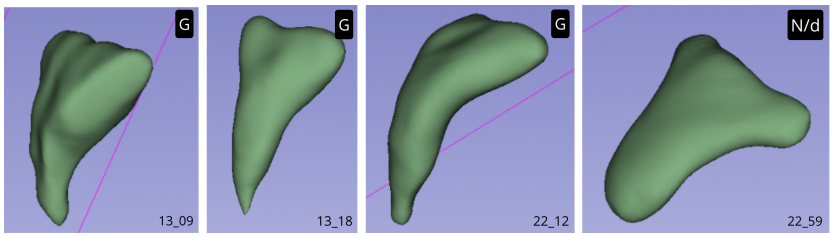


Figure 3.6: Chosen anatomical structures for the simulated environment, obtained from the UterUS dataset by Boneš et al. [88]. Images were acquired from the 3D Slicer software.

Uterus Surface

After obtaining and chosing the base structures for the uterus models, they were firstly imported into Blender 5.1.1 and then passed as FBX files to Unity 6.3 (version 6000.3.9f1).

Blender was used to prepare the anatomical models and define the properties of the surfaces of interest for the integration into Unity. This way, the resulting models and properties would be imported into Unity without requiring repetitions on the procedure at the same time that Blender provides more direct and precise control over mesh-level properties, such as surface orientation, normal vectors, textures and material properties.

In this case, two steps were particularly important: "Shade Auto Smooth" and "Flip Normals". The first improved how the lighting behaves and the visual continuity of the surface, giving a more natural appearance, while the second induced the interior surface to have a correct orientation, which prevents the interior of the model from appearing invisible in Unity.

The surface was also prepared for subsequent use in Unity through UV mapping, as shown in Figure 3.7, using "UV (Smart Project)". This process defined the surface appropriately, which allowed the application of materials and textures in Unity. This way, the material properties, such as colour and texture, only needed to be created and defined once and then assigned in Unity to the models, not requiring for them to be individually recreated in Blender.



Figure 3.7: UV mapping of the surface of the model 13_18, in Blender.

Finally, regarding the material, also created in Blender, the definition of its appearance was guided by the hysteroscopy videos provided by CHUSJ. These recordings served as a visual reference for reproducing the appearance of the uterine tissue in the simulated environment as faithfully as possible. This reference was particularly relevant for establishing the visual characteristics that would later contribute to the realism and variability of the simulated hysteroscopic images, while still recognising that the clinical recordings themselves exhibited limited visual variability.

Characteristics in the material option such as the material smoothness, roughness and metallic aspect were adjusted as closely as possible. The combination of the chosen values for them already influenced the lighting and the reflections on the surface, which was more visible when the setup in Unity was complete. For the visual texture, obtained in the material shading, ChatGPT assistance was used to obtain guidance on implementing procedural effects in Blender's "Shader Editor", particularly to generate subtle lighter reddish spots and irregular vein-like patterns distributed across the surface. The prompt consisted of this exactly written

request without any image from the CHUSJ patients. AI assistance was sought due to the lack of prior experience with Blender's "Shader Editor" and the large design space it offers, as various combinations of connected shader nodes can produce different visual effects, either when using the same set of components or as there are different options for each node that produce a different effect (e.g., textures can be of Noise, Voronoi, Brick, Wave or Sky textures), or when using the same set of components. These suggestions served only as implementation guidance and learning, while the final shader parameters were manually refined based on visual comparison with the reference videos.

The defined material properties in Blender's "Shader Editor" and the resulted texture material are included in Appendix B. It is important to note that the colors that were chosen for part of the configuration were selected by analysing the variety of colours observed across the real hysteroscopic RGB images using a pixel-colour inspection tool. This approach directly addresses the issue raised in Section 2.4.4, concerning the similarity between the colour characteristics of the real and synthetic images. The requirement described in that section goes further, mentioning the correspondence between real and synthetic colours quantitatively, based on their statistical distributions.

The created pattern was not directly passed through the material in the FBX file of the model. To reproduce the same appearance in Unity, the procedural texture was first baked into a 2048 x 2048 PNG image in Blender and then assigned to the material's "Base Color" property in Unity, with the "Wrap Mode" of the PNG set to "Repeat", so that the image would repeat through all of the surface.

The baking process required the use of Blender's UV Editing workspace to generate the texture map. An image was referenced in the "Shader Editor" through an "Image Texture" node, a box that needed to be sellected during the process, allowing the material to be rendered according to the model's UV coordinates. Since the baked texture depends on the UV mapping of the object, a new object, a plane, was created and assigned the same material so that a complete square representation of the texture was possible, not according to the highlighted regions in Figure 3.7. This texture also contributed to the adjustment of the material's base colour behind the texture in Unity, which was a bright and almost impercetible shade of pink.

To enable independent editing of the material properties in Unity, the material embedded in the imported FBX file had to be extracted (or duplicated) as a standalone material asset. Once extracted, this material could be assigned to the model through the "Materials" section and edited independently of the original imported asset.

Unity Setup

Several aspects of the Unity environment had to be explored, adjusted and defined in order to obtain the intended simulation environment.

Unity Render Pipeline

The data acquisition was performed using the 17.3.0 version of the High Definition Render Pipeline (HDRP), following the approach adopted in the VR-Caps work [35]. This choice was also motivated by the capability demonstrated in that work to record depth maps simultaneously with the image sequences through the Arbitrary Output Variables (AOVs), available in the Unity Recorder. This provided a potential means of obtaining both the RGB images and their corresponding depth information during the same recording process.

Unity camera configuration

In order to simulate a real hysteroscopic camera, a virtual monocular camera was created as an independent object in the Unity hierarchy. It was positioned inside the uterine model, at the inferior region of the models, and oriented towards the uterine cavity in the direction of the Fallopian tubes. This initial placement allowed the simulated navigation to follow the entering the uterine cavity movement.

The camera was connected to C# scripts, developed in Visual Studio 2022, version 17.9.1. One of the scripts was specifically responsible for controlling the camera movement within the Unity environment, while the remaining were dedicated to data acquisition and to defining or maintaining the properties required for the acquisition trajectories. Those are explored below, in Subsection 3.1.3.

`EndoscopeController.cs` is the one who enables camera translation through keyboard inputs and orientation that allows rotation using mouse movements, reproducing the free navigation typically performed during a hysteroscopy. This allows to control parameters such as the move speed and the mouse sensitivity.

In addition, the parameter `maxDegreesPerFrame` was used to limit the maximum angular displacement applied to the endoscope between consecutive frames. This was particularly important during data acquisition because the recording frame rate could temporarily decrease, causing mouse movements to accumulate between successive frames. Without this limitation, the accumulated mouse input could result in a disproportionately large rotation being applied in a single frame, producing abrupt camera movements and potentially unrealistic transitions between consecutive images.

The complete implementation of the script and the possible variable changes in the Unity interface for the camera are provided in Appendix C.

The principal altered from default parameters are the near and far clipping planes, defined as 0.01 and 60, respectively, a focal length of 28.97056 mm in the "Physical Camera" mode and a field of view of 45° in the camera projection section.

The virtual camera does not model lens distortion, no radial or tangential distortion coefficients are present. Thus, the corresponding distortion parameters can be considered zero, this is k_1=k_2=p_1=p_2=0.

Regarding the camera's intrinsic parameters, which were subsequently obtained, the focal lengths were f_x=f_y=1545.097 pixels, while the principal point was located at

(c_x,c_y)=(960,540) pixels. The latter corresponds exactly to the centre of the (1920×1080) image. The equality between f_x and f_y indicates equal focal scaling along the two image axes, while the centred principal point and absence of a skew term characterise the resulting intrinsic geometry.

Overall, the camera configuration provided a controlled and fully defined imaging geometry for the simulation.

Spot light configuration

To reproduce the illumination provided by a hysteroscope, a "Spot Light" was added as a child object of the camera. This ensured that the light source remained aligned with the camera throughout the navigation. The spot light parameters, including the inner and outer cone angles and emission intensity, range and colour were adjusted to approximate the illumination observed in real hysteroscopy videos. These parameters were later involved in the changes during acquisition so that simulations were executed under a range of variable conditions.

Due to the spot light configuration choice, only the central circular region of the rendered image is illuminated, visually reproducing the circular field of view characteristic of a real hysteroscope, with the endoscopic image displayed within the circular region and a uniform background outside it.

The final camera and illumination parameters are included in Appendix C.

Environment lighting

All default light sources in the Unity scene were removed so that the camera was the only source of light. Additionally, a "Global Volume" was created and, in its inspector, a "Visual Environment" override was added, with the sky type set to "None". This configuration eliminated unrealistic lighting and the purple reflections on the uterine walls shown in Figure 3.8.

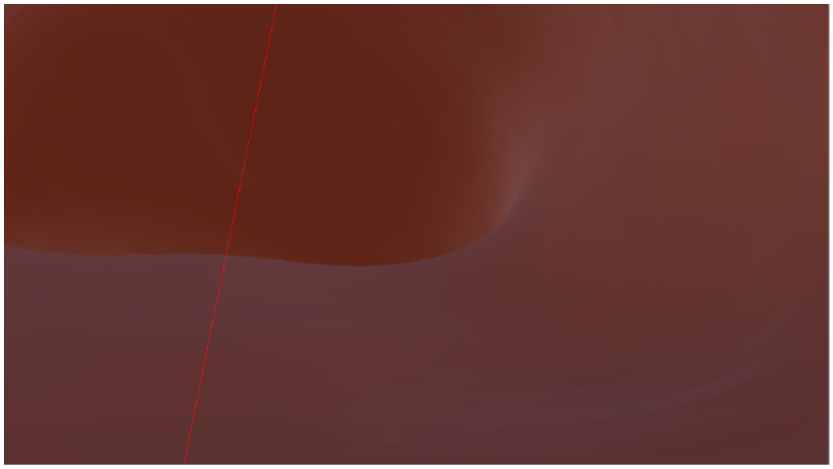


Figure 3.8: Purple reflections on the intrauterine walls due to unrealistic lighting. These reflections were corrected.

Visual Appearance and Material Refinement

After establishing the anatomical configurations, materials, camera, lighting, and other details described above, additional aspects were investigated with the aim of improving the visual realism of the simulated tissue, particularly with regard to how light interacts with and is reflected by the surface.

The "Emission" property of the material was explored. As the basis for the recordings, the final setup was defined by incorporating emission into the material, using a colour that was also present in the real videos, but a darker shade of the ones used for the texture, together with an emission intensity value of 3.

In turn, the "Coat Mask" parameter was lit and adjusted in an attempt to reproduce the moist appearance and subtle specular reflections, but these modifications did not produce a difference in the visual effect under the selected rendering configuration.

Additionally, alternatives in the material type were considered but the visuals remained the same as the established one, "Standard", except for the "Subsurface Scattering", which produced an excessively fluorescent appearance on the surface, not representative of the intended uterine tissue appearance.

Based on the defined settings and the previously established configurations, the final image acquisition setup was defined. Figure 3.9 shows frames of the real images and the visual results of the synthetic data generated as reproducing the relevant visual characteristics of the real data.

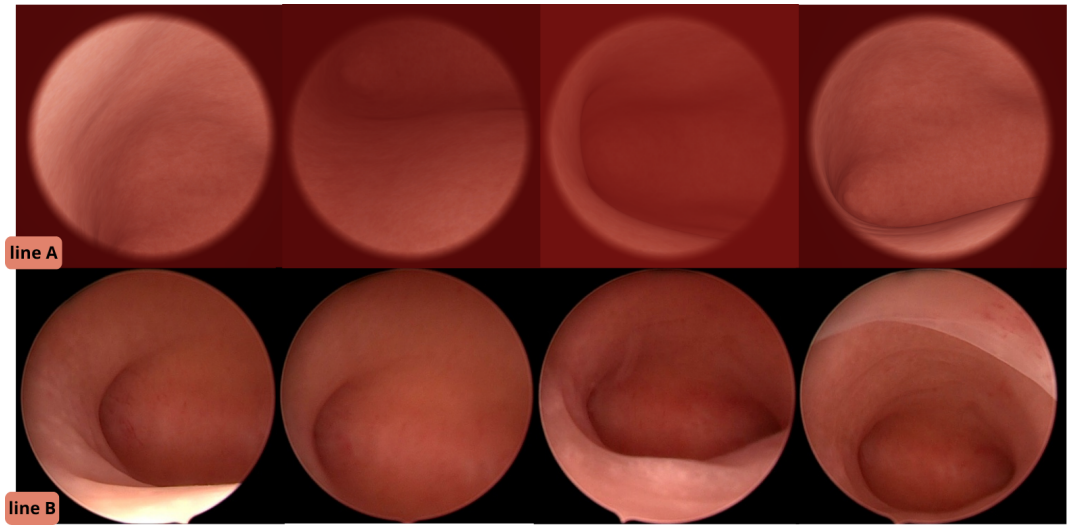


Figure 3.9: Example of frames from the results after the Unity set up (line A) side by side with the real images from CHUSJ (line B). This figure should not be interpreted as implying that the figure above reproduces the figure below. The ones below are provided solely for general comparison, without excluding any from Figure 1.2, in order to avoid introducing bias into the visual perception.

Model scale

The real physical dimensions of the structures were obtained from the UterUS study, where one voxel corresponded to 0.5 mm. An analysis was performed to verify whether the final mesh, from which the structural distances were measured, preserved these physical dimensions, since scaling or other transformations applied in Blender or Unity could have altered the original scale. The dimensions are presented in Table 3.3. This analysis consisted on slicing the model to obtain the dimensions, as shown in Appendix D.

Table 3.3: Comparison between the dimensions of the meshes measured in Unity units (UU) and the corresponding real-world dimensions (mm) obtained from the segmented ultrasound volumes.

Uterus structure	**Mesh Dimensions (UU)**	**Uterus Structure Dimensions (mm)**
13_09	$41.41 \times 29.00 \times 16.54$	$41.56 \times 28.49 \times 16.11$
13_18	$44.46 \times 26.52 \times 11.17$	$44.22 \times 26.49 \times 10.96$
22_12	$36.64 \times 24.43 \times 12.22$	$36.82 \times 24.10 \times 12.77$
22_59	$30.61 \times 31.49 \times 8.44$	$30.86 \times 31.16 \times 8.23$

The conversion factor between Unity Units (UU) and millimetres was estimated as

$$k = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathrm{mm}_i}{\mathrm{UU}_i} = 0.9952, \qquad \sigma_k = \frac{s}{\sqrt{n}} = 0.0056 \tag{17}$$

which means that

$$1\ \mathrm{UU} \approx 0.9952 \pm 0.0056\ \mathrm{mm}. \tag{18}$$

The estimated conversion factor is sufficiently close to unity to establish the metric scale of the simulated environment. Accordingly, throughout the subsequent analyses, 1 UU was considered equivalent to 1 mm.

Still, no conversion is applied during the processing pipeline. Depth is exported directly in scene units, and the networks are trained and evaluated using these same, with the possibility for the resulting metrics to be interpreted in millimetres. The point clouds are also back-projected and fused in the inherited units. Since the scene was scaled such that one scene unit corresponds to around 1 mm, these scene-unit depth values can be interpreted as millimetres in physical terms.

3.1.3 Synthetic Data Acquisition

The synthetic dataset was acquired in the Unity environment using the anatomical models described previously. The acquisition procedure was designed to reproduce, as closely as possible within the simulated environment, the type of navigation performed during diagnostic hysteroscopy, while maintaining control over the camera trajectory and ensuring the availability of the information required for the monocular depth estimation and 3D reconstruction.

For each captured frame, four types of information were recorded: the RGB image, the depth map, the camera pose and the camera intrinsic parameters.

The flow of this subsection is according to Figure 3.10.

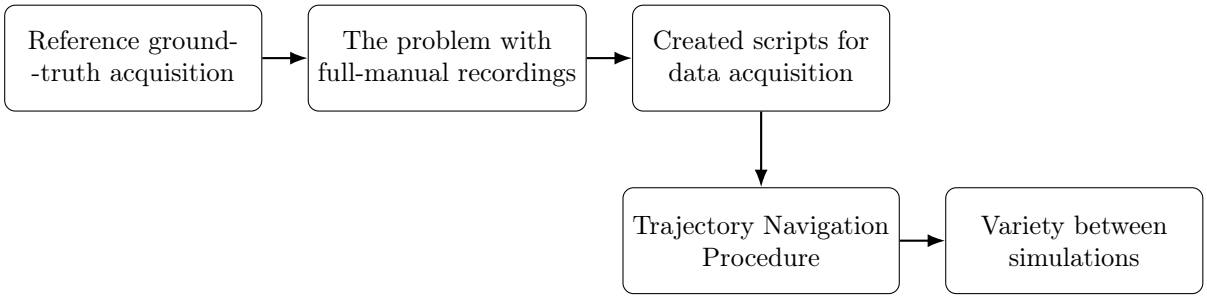


Figure 3.10: Overview of the synthetic data acquisition process.

Reference ground-truth acquisition

It was stated above that one of the reasons for selecting HDRP as the Unity rendering pipeline was that İncetan et al. [35] obtained depth maps using the Unity Recorder through an AOV image sequence, with the "depth" option selected. When analysing, this option did not provide sufficient information about how the depth values were represented or whether they corresponded to a linear physical distance. Therefore, an initial validation was performed to determine the meaning of the exported values.

For this purpose, a cube was created and positioned at different known distances from the camera, according to the axis coordinates. The minimum and maximum values obtained from the corresponding depth maps are presented in Table 3.4.

Table 3.4: Depth values obtained from the Unity Recorder AOV for a cube positioned at different known distances from the camera.

Real distance from the cube	**Minimum depth (units of measure)**	**Maximum depth (units of measure)**
1 m	0.5263672	0.65527344
2 m	0.0000000	0.21533203
5 m	0.0000000	0.068725586
10 m	0.0000000	0.032043457
20 m	0.0000000	0.015380859

As shown, results did not correspond to the expected distance, either when considering Euclidian distance or when considering other possible option, the Z-depth, which represented only the component of the distance along the camera's optical axis. They showed no pattern, which indicated that the exported AOV values did not provide the required representation of physical depth. The observed behaviour was therefore interpreted as being related to the depth representation used internally by the rendering pipeline.

Thus, the depth maps generated through the AOV system were not ultimately used as the ground-truth depth data. Instead, they were collected through a C# script.

The problem with full-manual recordings

The camera trajectory was initially controlled manually, with the movement from the defined keyboard keys and mouse in `EndoscopeController.cs` and collection of data through the existing scripts, explored below.

However, when stopping the movement to look around, both keyboard keys and mouse working together was difficulted or, sometimes, not even possible. Thus, the corresponding sequences only involved rotational motion and no translation. This means that, when looking at consecutive frames, the training signal of the algoritm would not understand depth.

To address the lack of translational motion, a scripted camera trajectory was therefore introduced. The objective was to ensure that the camera continuously underwent translation while changing its viewing direction, rather than simply rotating around a fixed position. This script is explored below.

Created scripts for data acquisition

The acquisition pipeline was implemented through several custom Unity scripts, as explained below. The full scripts are presented in Appendix C. To implement this functionality, generative AI Claude, was used as an assisting tool, namely due to Unity-specific commands and technical specifications that were outside previous experience. It would create based on prompts with the required functionalities and the intended behaviour of each component and the resulting code was reviewed, tested and modified when necessary.

- `EuclideanDepthPassHDRP.cs`

  To generate the ground-truth depth, a custom rendering pass was implemented for the High Definition Render Pipeline (HDRP), using `EuclideanDepthPassHDRP.cs` together with `EuclideanDepthHDRP.shader`. This approach was adopted because, as discussed previously, the depth information directly available from the Unity rendering pipeline does not correspond to the Euclidean distance between the camera and each visible surface point.

  A shader is determining how information is processed during rendering, being responsible for processing the depth information for each rendered pixel and computing the Euclidean distance. To get access to the camera and transmit the information at the right time, it needed to be integrated into the rendering pipeline, which was performed with the connection through a mechanism such as a Custom Pass. In turn, it needed an element in the Unity scene in order for it to be registered and inserted into the HDRP rendering pipeline, which asked for the need of the `EndoscopeRigSetup.cs` script.

- `EndoscopeRigSetup.cs`

  The `EuclideanDepthPassHDRP.cs` costum pass was hosted by a specific `CustomPassVolume`, created and configured by `EndoscopeRigSetup.cs`. This way, the global `CustomPassVolume` defined the rendering context to execute `EuclideanDepthPassHDRP.cs`.

Cameras other than the one for recordings were avoided because obtaining the modalities from the same rendering camera ensures their spatial alignment by construction.

- `DepthValidationScene.cs`

  The generated depth maps were validated independently of the acquisition process. This script used a planar wall positioned perpendicular to the camera's optical axis to verify that the generated depth represented Euclidean distance rather than the camera-space Z coordinate. If depth corresponded directly to Z, all pixels of the planar surface would have the same value. In contrast, Euclidean depth increases towards the image borders according to the distance from the camera centre. This validation was also used to verify the consistency of the camera intrinsic parameters used in the acquisition pipeline.

- `EndoscopeScriptedPath.cs`

  This script was created to avoid purely rotational image pairs, which were happening with a manual trajectory, and to be more suitable for the self-supervised depth estimation framework. This was intended to ensure that consecutive frames contained sufficient parallax for monocular depth estimation based on view synthesis and for subsequent three-dimensional reconstruction.

  It involves a waypoint-based approach, consisting of the connection with created points in the Unity game interface, manually distributed in the anatomical structure. The script reads the places and progressively changes its viewing direction towards them. This allowed different regions of the uterine cavity to be observed while maintaining controlled translational movement. This combination of scripted camera control and manually defined waypoints provided a compromise between reproducibility and flexibility, allowing the acquisition trajectory to be adapted to the anatomical differences between models while avoiding the limitations encountered during fully manual acquisition.

- `EndoscopeManualOrbit.cs`

  In addition to a scripted path, obtained in `EndoscopeScriptedPath.cs`, a script was created to allow manual trajectories while, at the same time, avoiding purely rotational image pairs. This was different from teh previous manual manoeuvres, which did not avoid it. This script introduced a small lateral displacement during manual camera motion, while the scripted trajectory maintained translational motion during changes in viewing direction.

- `EndoscopeRecorder.cs`

  This script synchronised the acquisition of all modalities using a common frame counter, ensuring that RGB frame i, the depth frame i and the corresponding pose recorded in the CSV file referred to the same camera state. During capture, the engine's frame rate is pinned, so the interval is constant and the trajectory is reproducible. RGB images were exported as PNG files, depth maps as 32-bit floating-point EXR files, camera poses as CSV data and intrinsic parameters as a JSON file.

Trajectory Navigation Procedure

Initially, the camera followed a trajectory from the cervical canal towards the uterine cavity, progressively advancing while changing its viewing direction to observe the surrounding tissue. The navigation continued until reaching regions in which the openings of the Fallopian tubes became visible, looking at the left and right. The camera advanced while observing the tissue in front of it rather than explicitly rotating backwards to inspect previously traversed regions in order for it to to remain consistent with the navigation behaviour observed in hysteroscopic videos. This provided a controlled and reproducible acquisition path.

During the acquisition process, it was decided that the trajectory could vary between videos, which were called scenes. Although the overall direction of navigation was maintained, the initial camera position and the regions observed were modified whenever the geometry of the anatomical model allowed it. In some sequences, the camera was even positioned further inside the cavity and subsequently moved backwards, while maintaining the forward-facing viewing direction. However, for some cases, backward navigation resulted in the camera leaving the interior of the anatomical model due to the difficulty of simultaneously controlling the backward navigation and its top-to-bottom movement. When that mistake occured and the outside was seen inside the defined circle, such sequences were excluded from the final dataset.

Examples of the trajectories are presented in Figure 3.11.

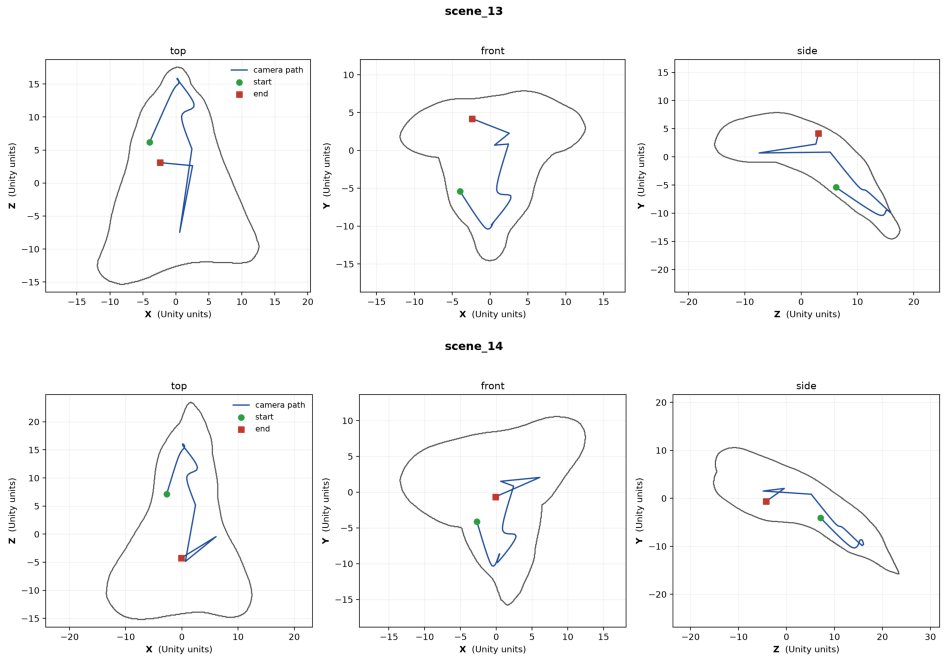


Figure 3.11: Examples of the performed trajectories in the simulated Unity navigation for the collection of data.

Due to what was explained in ***The problem with full-manual recordings***, and as explained in ***Created scripts for data acquisition***, the script `EndoscopeScriptedPath.cs` was used for trajectory.

An initial attempt consisted of initiating the recording and the camera would move along a predefined helical trajectory implemented through the script and by creating trajectory segments in the Unity interface. Figure 3.12 illustrates the resulting trajectory configuration in the Unity interface.

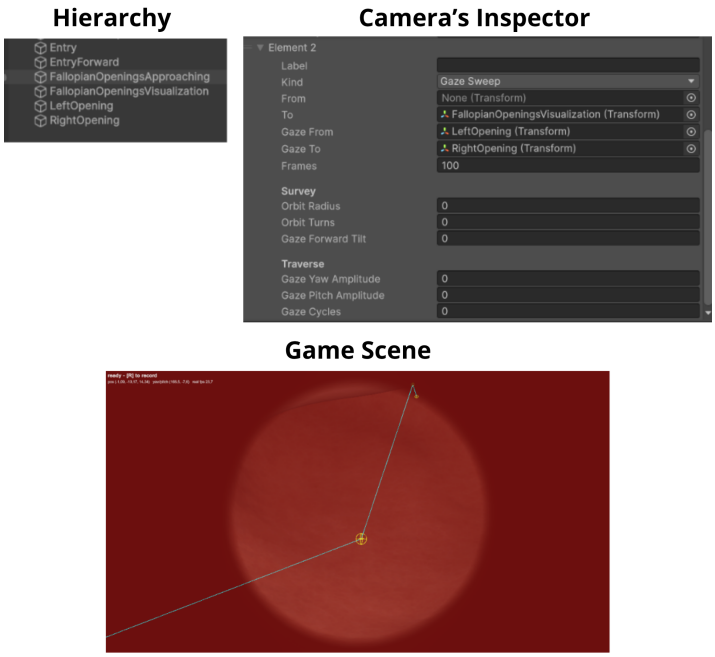


Figure 3.12: Relevant information on the Unity configuration of the automated helical trajectory and its point representation within the Game Scene.

However, this did not provide the expected result, as viewing direction remained unchanged. To overcome this limitation, the idea of adopting a hybrid approach emerged, in which the camera could still be manually controlled to vary its viewing direction while a scripted movement simultaneously ensured the required translational displacement. This approach was intended to combine the flexibility of manual navigation with the controlled motion provided by the script, thereby avoiding the predominantly rotational movements observed previously. The script that allowed this strategy was `EndoscopeManualOrbit.cs`.

Nevertheless, the manual acquisition presented practical limitations. In particular, when a movement was requested, the response was sometimes delayed because the system was simultaneously processing and saving a large amount of information. This means that the acquisition of approximately 100 frames could take around 9 minutes.

In addition, sometimes an error occurred and the central circular view dissapeared and the whole frame remained in one color, requiring additional camera manoeuvres to restore the normal view, or the simulation could occasionally freeze during navigation, preventing continuous camera control. The former was subsequently corrected, in the preprocessing preparation for depth estimation. The latter led to several sequences to contain insufficient variation in viewpoint between consecutive frames and, since it could negatively affect the self-supervised learning process, several sequences had to be repeated to obtain sufficient viewpoint variation.

To improve reproducibility, the `EndoscopeScriptedPath.cs` script was reconsidered. Since

the objective was to observe different regions of the anatomical model throughout the trajectory, the approach was adapted so that the camera would be progressively directed towards predefined locations. This led to the introduction of waypoints, which constrained the camera's viewing direction towards selected regions while maintaining the controlled aspect. Several waypoint locations were defined along the cavity, with their positions selected to expose different anatomical regions. The order of the waypoints was then specified in the camera inspector. These configurations are illustrated in Appendix C.

Particular attention was given to the orientation of each waypoint, since the camera only looked at those waypoints that had their own forward direction turned to meet the camera's forward direction. In addition, the way that the camera viewed the point for each waypoint was as if the camera positioned itself in their foward direction to look at the waypoint's exact place. This was important, as, if done wrong, this would allow orientations that caused the camera to observe outside of the model. This was particularly relevant in regions with pronounced changes in cavity geometry.

Ending up with these two possibilities for recording made it possible to alternate between them, either recording through manual or automated acquisition.

Variety between simulations

In order to increase data diversity, an important consideration identified previously was taken into account: the simulations should be performed under a range of variable conditions rather than relying on a single fixed configuration. This is particularly relevant as the synthetic data are intended to represent a clinical scenario and avoid limited generalisation to real procedures.

As discussed in Subsection 2.4.3, particularly in the topic ***If it covers the actual clinical situation***, the variability expected during real hysteroscopic procedures was considered when defining the simulation conditions between recordings, which were named as "scenes". Several factors may contribute to differences between procedures and between consecutive images, including:

- **Tissue appearance:** the observed colour may vary, for example, between more pinkish and more orange tones, as observed in the real hysteroscopic videos;
- **Illumination and surface reflections:** the illumination produced by the hysteroscope can generate stronger reflections on surfaces located closer to the camera, with the appearance of these reflections depending on the relative position and orientation of the camera and tissue;
- **Camera and illumination characteristics:** the illumination conditions may not remain identical throughout different procedures, and different hysteroscopes or camera systems may also introduce variations in image appearance;
- **Anatomical variability:** uterine anatomy differs between individuals, making the inclusion of different anatomical configurations important for avoiding an excessively specific representation of a single cavity geometry;

- **Navigation variability:** camera motion is not necessarily identical between procedures, as different clinicians may navigate the hysteroscope differently, and even the same clinician may perform different movements depending on the procedure and the region being examined;
- **Pathological and lesion-related anatomical variability:** the internal appearance and geometry of the uterine cavity may differ substantially in the presence of pathologies, lesions or other abnormal anatomical conditions. Such conditions can introduce additional structures or protrusions and modify the appearance and spatial distribution of the surrounding tissue, as exemplified in Figure 3.13. Consequently, anatomical diversity based only on healthy uterine configurations does not fully represent the range of structures that may be encountered during hysteroscopy.

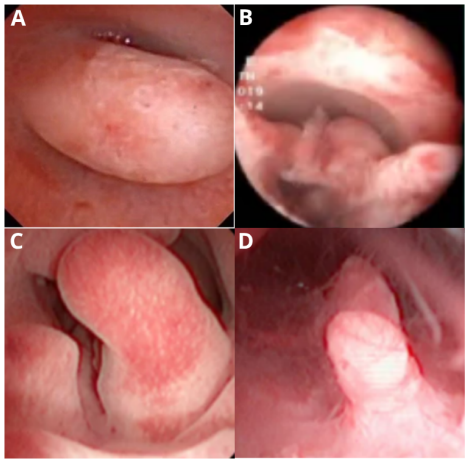


Figure 3.13: Some of the collected images from articles where a polyp is present in the intrauterine cavity. A - from Takahashi et al. [84], B - from Zhang et al. [83], C, D - from Yang et al. [85], licensed under CC BY 4.0, with the corresponding link connection: creativecommons.org/licenses/by/4.0/.

Based on these considerations, several sources of variability were introduced during the data acquisition process. To provide a basis for considering which aspects of the simulated environment could reasonably vary in the present application, the work by Walkner et al. [62] was considered. The authors introduced variations in image colour and applied image transformations such as cropping and modifications to brightness, contrast and saturation. Although the specific transformations used in that work were not directly replicated, they reinforce the considered variations:

1. **Camera position and viewing direction:** the camera position and orientation were varied between scenes, either only one of them or simultaneously, to provide different viewpoints of the same anatomical structures;
2. **Anatomical diversity:** different uterine models were selected according to their structural differences, as illustrated in Figure 3.5. Some models from the UterUS dataset

were excluded when it was considered that they resulted in highly repetitive anatomical image sequence collection, prioritising variation in the structures represented rather than simply increasing the number of samples;

3. **Anatomical model orientation:** some models were rotated to change their spatial orientation, changing the direction of their curvature. This allowed different relative configurations between the camera and the anatomy to be obtained;

4. **Spotlight intensity and colour:** the intensity and colour of the spotlight were varied between acquisitions, using a grey-based light as the initial configuration, in order to represent differences in illumination conditions, meeting pinkish and more orange tones;

5. **Material base colour:** the base colour of the anatomical material behing the applied texture was modified between acquisitions to represent variations in tissue appearance, particularly between the pinkish and more orange tones observed in the real hysteroscopic videos;

6. **Material smoothness:** the smoothness of the material was varied to modify the way light interacted with the anatomical surfaces and, consequently, the appearance of surface reflections;

7. **Material emission:** the colour and intensity of the material emission were varied between acquisitions to introduce differences in the background involving light;

8. **Excluded sequences and models:** sequences and models that resulted in excessive repetition or insufficient variation were not included in the final dataset. This was particularly relevant for models with highly similar anatomical configurations or acquisitions that did not provide sufficiently different viewpoints;

9. **Camera trajectory:** both manual and scripted camera trajectories were used. Manual navigation provided greater variability in camera movement, while scripted trajectories provided more controlled and reproducible motion. Using both approaches allowed different movement patterns to be represented while reducing reliance on a single acquisition strategy.

Finally, the self-supervised approach also applies basic data augmentation techniques, namely image flipping and cropping. Although these augmentations are relatively limited in terms of the variability they introduce, they still contribute to increasing the diversity of the training data.

To conclude, the present dataset focuses on variability in anatomical configuration, acquisition conditions, illumination, material properties and camera trajectories, while pathological and lesion-specific variability is identified as a limitation and potential direction for future work.

3.1.4 Collected Data in Unity

The objective of the simulated synthetic environment was to provide a controlled basis for generating image sequences and their corresponding ground-truth depth maps, together with additional information such as the camera trajectory, which is essential for the subsequent reconstruction process.

For each simulated sequence, the following data were collected, with the aim of preserving as much information as possible, including by retaining a high image resolution:

- **RGB image sequence**: Exported as PNG images with a resolution of 1920×1080. These images constitute the input to the proposed 3D reconstruction pipeline, where a specific amount will be used for training and validation and the remaining for test. They correspond to the data that will also be available in the intended clinical application;
- **Depth map sequence**: Exported as an EXR file (32-bit floating-point) with a resolution of 1920×1080. The channel corresponds to the Euclidian distance, where pixels with -1 values are pixels with no geometry. One depth map was generated for each RGB frame and serves as the ground truth for training and evaluating the depth estimation algorithm;
- **Camera intrinsics**: Stored in JSON format, with one file generated for each video sequence, not by each image sequence. This file registers the image resolution, focal lengths (f_x and f_y) principal point coordinates (c_x and c_y), which define the intrinsic calibration matrix of the virtual camera. The field of view and the near and far clipping plane were also stored to facilitate reproducibility of the acquisition setup. The intrinsics enable the projection of depth maps into 3D space, allowing the generation of point clouds to visually evaluate results or to obtain 3D meshes if necessary for the following steps for polyp detection methods;
- **Camera poses**: Stored in CSV format. Each row corresponds to one RGB frame and contains the camera position (x, y, z) together with its orientation represented as a quaternion (q_x, q_y, q_z, q_w). In addition, the seconds since the video started and the constant interval from the previous frame are also collected ($videoTime$ and $deltaTime$, respectively). Poses are what places each frame's points in a common coordinate system, so they are essential for the creatiom of point clouds.

The collected simulated sequences, named as "scenes" consisted of a total of 17, and the uterine structure used as basis for each and the number of obtained frames are presented in Table 3.5. The variation of the uterine structure included its positioning according to its curvature. For each scene, parameters were varied to obtain the most diverse dataset as possible. Values such as the spot light intensity, color, the material smoothness, coat mask and emission intensity were varied. More detailed information on the curvature positioning and the considere metrics for each sequence are presented in Appendix E.

The collected data were subsequently processed to make them suitable for use as a dataset for the selected self-supervised learning method, as explained in subsection 3.2.3.

Table 3.5: Uterine structures, considered orientation and number of frames used for each synthetic recording.

Scene	Uterine structure (curvature)	Number of frames
scene_1	22_12 (up)	201
scene_2	22_12 (down)	99
scene_3	22_12 (down)	97
scene_4	13_18 (down)	193
scene_5	13_18 (down)	109
scene_6	13_18 (down)	106
scene_7	13_09 (down)	179
scene_8	13_09 (up)	58
scene_9	13_09 (down)	127
scene_10	22_12 (down)	70
scene_11	22_12 (down)	36
scene_12	22_12 (down)	74
scene_13	22_12 (down)	146
scene_14	13_18 (down)	144
scene_15	22_59 (down)	203
scene_16	13_18 (down)	154
scene_17	22_59 (down)	152
Total		**2148**

3.1.5 Remaining Data

For the real data obtained from the hospital and from the articles Zhang et al. [83], Takahashi et al. [84], and Yang et al. [85], these data were also incorporated into the dataset as scenes 19 to 22. Scenes_19 and 20 correspond to the first and second videos, respectively, while scenes_21 and 22 comprise small regions extracted from the real images to meet what was mentioned in ***Spatial and Temporal Consistency***, discussed in subsection 2.4.4. Scene_22 additionally includes images containing pathology, ensuring that the evaluation on real data encompasses this type of visual diversity.

The real data were preprocessed as required for the frames to be correctly read by the algorithm. For example, the video data were extracted frame by frame before other processing, whereas the polyp images only required resizing. The specific preprocessing steps applied to each type of data are described below in ***Data preprocessing***, included in Subsection 3.2.3.

3.1.6 Assessment of the Synthetic Rendered Data

FID, explored in the literature, was initially considered for this evaluation because it measures the similarity between image distributions. However, as also discussed, FID is sensitive to the number of images, and estimates based on a small number of samples can be unstable and tend to produce higher values. Since the size of the dataset was already known at this stage of the methodology, the suitability of FID for the present data was further investigated in order to determine whether a sufficient number of samples was available. According to information available from online sources, FID requires 10 000 to 100 000 samples to obtain reliable estimates of the underlying mean and covariance [89, 90].

As the dataset generated in the present work and the set of real data are both substantially

smaller than these values, FID was not considered sufficiently reliable for this evaluation and was therefore not used.

Konz et al. [89] proposed a new metric to meet the usual availability of smaller datasets for medical image, the Fréchet Radiomic Distance (FRD). However, authors state that this metric is only prepared for images from radiology, not being able to be used in this work.

No other quantitative metric was considered appropriate for assessing the similarity of the rendered data at this stage. The generated sequences were therefore only assessed visually, as mentioned, with particular attention to the variation in anatomical appearance, illumination, texture and scene configuration across the simulated sequences.

3.2 Depth Prediction Pipeline

This section sets out why a self-supervised method derived from the structure-from-motion formulation was adopted, and why one particular implementation of it was chosen.

3.2.1 Prediction Method Decision

Building on the literature review, which includes a 2025 overview of the field, Table 2.3 was created to present the most recent works on monocular depth estimation specifically applied to endoscopy.

Reflecting on the conclusions drawn from the literature review, SfM and SLAM-based approaches were considered the most appropriate directions for the present work. Among learning-based approaches, supervised learning was not preferred, since it requires reference depth that is not available in the intended clinical setting. In addition, estimating the camera pose was considered important, as it is needed for obtaining the coverage map. Since this information was not available as a measurement for all works, a method capable of estimating pose as part of its formulation was considered preferable to one that only estimated depth. In addition, several studies in the literature adopted SfMLearner as a starting point and sought to improve, adapt or further investigate its formulation. Taken together, these considerations motivated the selection of an SfM-based approach for this work.

On the learning-based approaches

Regarding the properties and characteristics of SfMLearner-based approaches, the self-supervised formulation further supported this choice. The chronology of Table 2.3 also supports this decision independently. Among the endoscopy-specific methods surveyed, the great majority of those published since 2021 are self-supervised or described by their authors as unsupervised. The two exceptions use supervised learning with reference depth acquired using a projector on cadaveric tissue, an acquisition setup that has no direct clinical counterpart.

The weakly supervised regime does not provide an additional degree of freedom that is particularly useful in the present setting. Camera intrinsics are a fixed property of the imaging system rather than a per-sample annotation, and the self-supervised methods considered

already incorporate them into the reprojection process. What weak supervision would add beyond this is additional correspondence information, bringing the problem back towards the matching-based formulation that motivated the consideration of learning-based approaches together with the geometric route.

Unsupervised learning should also be distinguished from self-supervised learning in this context. The unsupervised works were of domain adaptation, which transfers a model trained on a source domain to an unlabelled target domain. It is therefore not an alternative to the other learning regimes, but rather a complementary strategy, since it presupposes a model trained elsewhere using one of those regimes. Moreover, it does not itself provide the camera motion required for the subsequent reconstruction.

The self-supervised formulation has two particularly relevant consequences for this work. First, it can be trained directly on real hysteroscopic video, which does not provide reference depth and is unlikely to do so in the intended clinical setting. Thus, the same training formulation used here with synthetic rendered data could, in principle, also be applied to clinical recordings without requiring a change in the type of supervision. This continuity between synthetic training and potential real-data adaptation is relevant. Second, the formulation estimates depth and camera motion jointly, and when pose data, which is needed for the mentioned application, is not available from the hardware, the estimated motion is required to transform the points reconstructed from individual frames into a common coordinate system. A method producing depth alone would not provide all the information required to obtain the coverage mapping and enable scene reconstruction.

On the geometric approaches

Finally, the use of SfM provides a connection between the learning-based and geometric components of the problem, as the learning-based component incorporates concepts traditionally associated with geometric reconstruction, including feature correspondence, image motion, camera-pose estimation and triangulation, into the training objective. In this way, geometric constraints are used to guide the learning process while the network learns to predict depth and motion from image information, combining the strengths of both learning-based and geometric approaches.

A possible reason why fewer of the methods identified in the literature adopt a full SLAM formulation is that their primary objective is not necessarily depth estimation itself, but rather camera localisation and the reconstruction and maintenance of a scene map. In this sense, SLAM moves one step beyond depth estimation: depth can be used as an intermediate component for recovering the scene structure, without necessarily being produced as an explicitly optimised or evaluated dense depth map for every frame. A SLAM system may therefore reconstruct a 3D structure without providing the type of dense, per-frame depth estimation required by the proposed pipeline.

This distinction is relevant to the present work. The objective here is not only to recover the three-dimensional structure of the scene, but also to obtain dense depth information that can subsequently be used in, for instance, point-cloud generation and coverage analysis or,

going further, supporting future polyp detection.

On the SfM approaches

An important distinction should first be made regarding SfMDiffusion. Although its name mentions the structure-from-motion formulation, it is not directly equivalent to the SfM-Learner family considered here. Its purpose is specifically to apply generative diffusion to self-supervised endoscopic depth estimation.

Besides that one, three recent methods in the surveyed literature derive more directly from the SfMLearner structure and were therefore considered: IID-SfMLearner, AF-SfMLearner and Endo-SfMLearner.

- **IID-SfMLearner, by Li et al. [50]**

  This work, from 2024, decomposes the image into reflectance and shading so that illumination changes do not corrupt the photometric objective, which is directly relevant to the conditions of hysteroscopy. Their formulation in how to obtain the loss combines decomposing-synthesis, albedo, mapping-synthesis and edge-aware smoothness terms.

  However, the evaluation protocol used by this method presents a limitation in the context of the present work. Trajectory error is calculated over five-frame snippets and subsequently averaged over the complete sequence. Thus, the evaluation primarily measures the local consistency of the estimated motion rather than the ability to maintain an accurate trajectory over a long sequence. This could mean that it may fail to capture errors that accumulate over time, including trajectory drift and global scale or positional deviations. This would mean, for this application, that it could produce a trajectory that progressively deviates from the reference over the complete sequence even if achieves favourable results under a short-term trajectory evaluation, and accurate long-term camera motion is required to obtain a consistent global reconstruction of the uterine cavity.

  Consequently, a previous algoritm was considered and analysed.

- **AF-SfMLearner, by Shao et al. [47]**

  This work from 2022 addresses the illumination problem through an appearance-flow formulation, while retaining a decoder design based on Monodepth2 and explicitly estimating ego-motion.

  The main limitation with its selection was a practical one. On the hardware available for this work, AF-SfMLearner could not be trained with a batch size larger than four because higher values resulted in GPU out-of-memory errors. The fact that the article mentions that they performed it in a NVIDIA TITAN GPU explains the computational power needed. This difference limited the range of configurations that could realistically be explored in the present work. Moreover, the batch size reported as optimal by the authors was 12, making it difficult to determine whether comparable training conditions

could be reproduced on the available hardware. Consequently, using this framework would have restricted the experimental space available for the adaptation study.

- **Endo-SfMLearner, by Ozyoruk et al. [34]**

  Then, the Endo-SfMLearner, a work from 2021, was considered.

  It produces the information required by the proposed pipeline: depth and camera motion are estimated jointly, and both outputs are reported. Moreover, its trajectory evaluation considers the scale of a complete organ sequence rather than only short frame snippets, making it more aligned with the requirements of global reconstruction.

  Then, the method provides publicly available code, which makes its adaptation feasible in the present work. Pre-trained weights are also available, which is particularly relevant to the fine-tuning strategy explored subsequently.

  The method is also specifically adapted to the characteristics of endoscopic imagery. Endo-SfMLearner itself extends a general-purpose framework, SC-SfMLearner, to the endoscopic domain. Therefore, the adaptation from natural images to endoscopic imagery has already been addressed and evaluated by the original authors, and the adaptation required here is mostly a change in anatomy.

  Furthermore, the authors explicitly identify a limitation of generic self-supervised approaches, as their assumption of photometric consistency between frames can be violated in endoscopic video because the camera and light source move together. Consequently, the same anatomy may exhibit substantially different appearance as the camera changes its position, particularly for tissue close to the lens. Endo-SfMLearner addresses this through a brightness-aware photometric loss, introduced specifically to account for these illumination variations, together with a spatial attention block that directs the network towards more distinguishable and highly textured regions. This is their way of trying to meet the illumination problem that is also mentioned in AF-SfMLearner and IID-SfMLearner.

  These characteristics address several conditions shared by hysteroscopy and the gastrointestinal endoscopy for which the method was developed, including a light source travelling with the instrument, specular reflections from wet tissue, and regions of mucosa with limited texture. The authors also report improved recovery of polyp boundaries and details, which could be relevant if the depth estimation moved in the direction of pathologies detection.

  The dataset employed by the authors focuses of the gastrointestinal tract (colon, stomach and small intestine) and the method is trained with one part of the dataset and explored those two anatomies, not remaining on a single anatomy. The authors further state that, although specifically developed and optimised for endoscopic images, the method also shows promise for laparoscopic images because of their similar texture characteristics. Both considerations do not provide evidence of performance in the uterine cavity, and no result reported in that work directly concerns hysteroscopy. Nevertheless, it provides

some support for the possibility of transferring the learned formulation across related imaging contexts.

This transfer should, however, not be assumed to be straightforward. The gastrointestinal sequences used to develop the method predominantly concern tubular lumens traversed along their longitudinal axis, whereas the uterine cavity is a relatively enclosed chamber with a different spatial organisation. Consequently, the visual and geometric priors acquired from the original training data are not necessarily the same as those required for navigation within the uterine cavity. This anatomical difference is therefore precisely what motivates the fine-tuning performed in the present work.

At the same time, Endo-SfMLearner incorporates a geometry consistency loss from the framework on which it is based. This term was introduced to encourage consistency between consecutive depth predictions in response to scale inconsistencies caused by changes in the distance between the camera and the organ wall. In the authors' own comparison, the two methods incorporating geometry consistency achieved considerably better performance than the previous evaluated approaches. While this could be a limitation when considering tissue deformation, it could also be a suitable approach if its predictions remain within an acceptable error range.

Taken together, these considerations led to the selection of Endo-SfMLearner for the present work. It provides the required depth and camera-motion outputs, explicitly addresses illumination variations characteristic of endoscopic imaging, offers publicly available code and pre-trained weights, and was developed specifically for endoscopic imagery. In addition, it involves a geometry consistency formulation, which could be relevant to subsequent reconstruction. The license for its use is included in this report in Appendix F.

3.2.2 The Endo-SfMLearner Framework

Endo-SfMLearner estimates depth and camera motion jointly from monocular video.

The method relies on two convolutional neural networks (CNNs): a depth network predicts a dense depth map D_t for a target frame I_t, while a pose network estimates the relative rigid transformation $\mathbf{P}_{t\to s}$ between the target frame and an adjacent source frame I_s.

Both networks are built on residual encoders. The depth network pairs a ResNet18 encoder with a five-level decoder that concatenates the matching encoder feature map at every level but the finest. The pose network uses an encoder of the same design but taking two concatenated frames as input, with an endoscopic spatial attention block (ESAB) inserted after the stem [34].

The use of residual architectures is particularly relevant to the network design. Increasing the depth of a conventional convolutional network can make optimisation progressively more difficult because of degradation during training. Residual architectures address this problem by learning corrections to an identity mapping instead of the complete transformation directly, facilitating the training of deeper networks while maintaining their representational capacity

[40]. This is advantageous for endoscopic images, where the networks must represent both fine anatomical structures and larger-scale spatial patterns.

The two networks are jointly used to establish geometric correspondences between consecutive frames. Given the camera intrinsics matrix $\mathbf{K}$, the estimated depth and relative pose define a mapping from the target frame to the source frame. Specifically, each pixel p_t in the target frame is projected into the source frame according to

$$p_s \sim \mathbf{K}, \mathbf{P}_{t \to s}, D_t(p_t), \mathbf{K}^{-1} p_t, \tag{19}$$

where $D_t(p_t)$ provides the depth associated with the target pixel.

Sampling the source image at the resulting coordinates produces a synthesised target image $\hat{I}_t$. The discrepancy between $\hat{I}_t$ and the recorded target image I_t then provides the supervisory signal used during training.

This means that the method does not require an annotated depth map. What happens instead is that the relationship between consecutive frames provides the supervision and the pose network supplies the relative motion required to establish the correspondence between them.

The schemes in Figure 3.14 and 3.15 illustrate the explained framework.

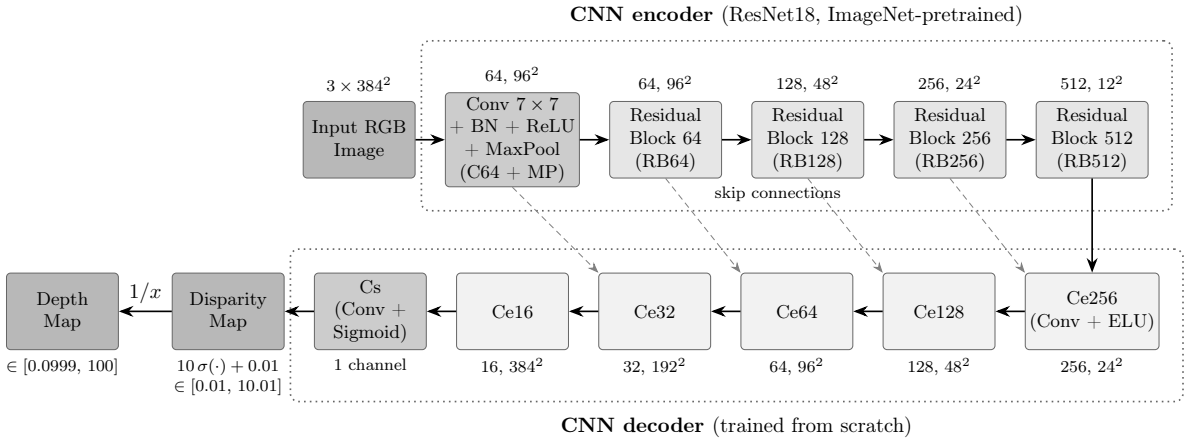


Figure 3.14: Architecture of the Endo-SfMLearner depth estimation module, in the block nomenclature of Ozyoruk et al. [34].

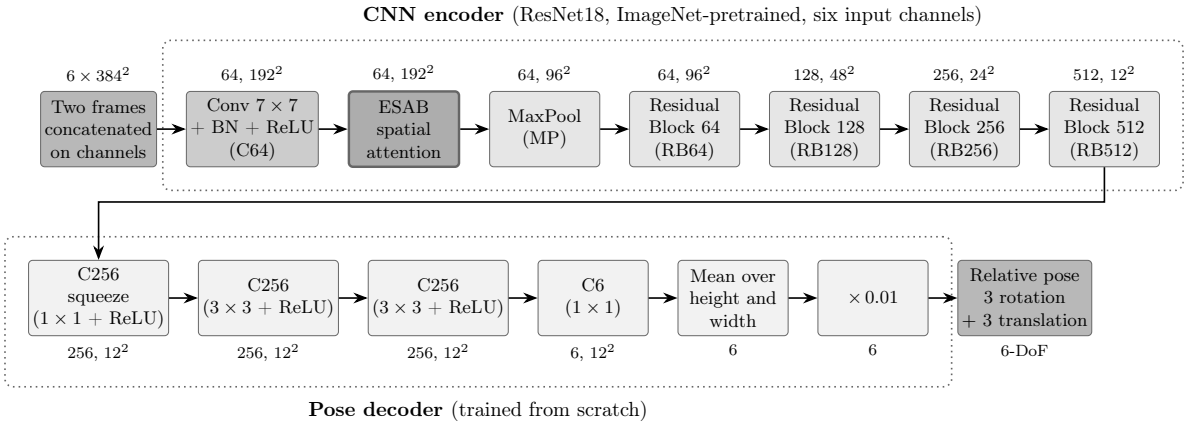


Figure 3.15: Architecture of the Endo-SfMLearner pose estimation module, in the block nomenclature of Ozyoruk et al. [34].

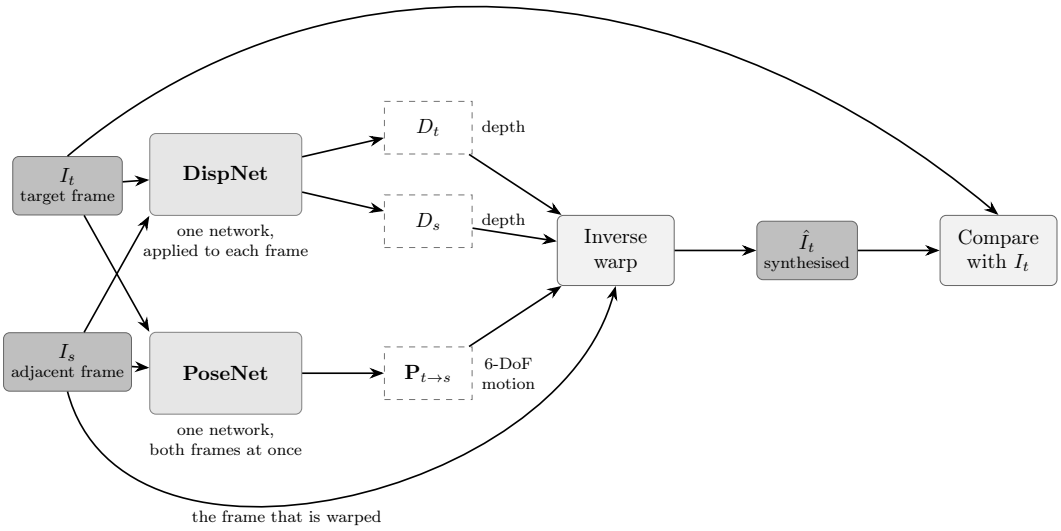


Figure 3.16: How the depth and pose networks relate.

As antecipated in Figure 3.14, the depth network does not directly predict metric depth. Instead, it predicts disparity, which is converted to depth through its reciprocal.

Consequently, the resulting depth is defined only up to an unknown scale, as expected for monocular self-supervised estimation. The predicted depth must be aligned with the ground truth before the evaluation metrics are computed, and this is addressed through median scaling, inherited from the framework on which Endo-SfMLearner is based. The corresponding expression is shown in

$$\hat{D} = D \cdot \frac{\text{median}(D^*)}{\text{median}(D)}, \tag{20}$$

where D denotes the predicted depth, D^* the ground-truth depth, and $\hat{D}$ the scaled prediction. The same scaling procedure is adopted in the present work, providing consistency with the published evaluation protocol.

Moreover, because depth is obtained as the reciprocal of disparity, the resulting depth resolution is not uniform: the representation provides finer resolution at shorter distances and progressively coarser resolution at larger distances. This is relevant to the present application, since distant regions of the uterine cavity may therefore be represented with lower depth precision.

Regarding the training objective, it combines three complementary terms, this way answering the limitation of relying solely on the photometric reconstruction error.

- **Brightness-aware photometric loss**

  A conventional photometric loss assumes that corresponding surface points maintain approximately the same appearance between consecutive frames.

  When thinking about endoscopic scenarios, the light source is attached to the camera, which means that motion changes not only the viewpoint but also the illumination on the tissue, so a region may undergo substantial intensity changes even when its position in the scene remains unchanged. Additionally, small changes in camera position can produce large changes in illumination when, for instance, a tissue close to the lens reflects it with a very bright result. So, the conventional photometric loss assumption is not what happens for endoscopic data and can add error.

  Thus, Endo-SfMLearner incorporates a brightness-aware formulation of the photometric loss, together with a structural similarity component, to reduce the sensitivity of the reconstruction error to these illumination changes.

- **Edge-aware smoothness**

  This term arises from the fact that the photometric reconstruction term alone provides weak constraints in regions with little or repetitive texture, which happens in endoscopy, as large portions of the mucosal surface may contain limited distinctive texture.

  This term is applied to the predicted depth. The regularisation encourages neighbouring pixels to have similar depth values while reducing its influence near image gradients. This way, the predicted depth can remain smooth in homogeneous regions without unnecessarily smoothing across anatomical boundaries.

- **Geometry consistency**

  The geometry consistency term introduces an additional constraint between the depth predictions of consecutive frames.

  Due to the predicted depth being compared with the depth obtained by projecting the target-frame prediction into the source view, the depth estimates of consecutive frames are encouraged to remain geometrically compatible. This makes the contrast with the possibility of them being treated as independent predictions.

Authors also mention that this term is a means of adressing the scale-inconsistency from one frame to another. Their experimental comparison further reports that methods incorporating geometry consistency achieved considerably better performance than the remaining approaches considered in their study [34]. This is also important when considering that this work focuses on reconstruction of the scene in a common coordinate system, as the geometry consistencyworks as an additional mechanism for maintaining compatibility.

Thus, the loss is given by

$$\mathcal{L} = \alpha\mathcal{L}_{\text{bp}} + \beta\mathcal{L}_{\text{s}} + \gamma\mathcal{L}_{\text{GC}}, \tag{21}$$

where $\mathcal{L}_{\text{bp}}$ is the brightness-aware photometric loss, $\mathcal{L}_{\text{s}}$ the smoothness loss, $\mathcal{L}_{\text{GC}}$ the geometry consistency loss and α, β, and γ are the weights assigned to the respective loss functions, which are not necessarily required to sum to one. The overall objective of the system is to minimize this weighted sum.

Reported Training, Optimisation and Comparative Evaluation

Both networks are trained together by a single gradient step on the summed objective. The optimiser, Adam, starts with an initial learning rate of 10^{-4} and the customary decay rates of 0.9 and 0.999. The released configuration weights the photometric term at 1, the smoothness term at 0.1 and the geometry consistency term at 0.5, and places the parameters of the two networks in separate groups that nonetheless share one learning rate, so no layer-wise schedule is applied by default.

Training runs for 200 epochs over randomly shuffled batches of four images, with validation after every epoch. The training set corresponded to 2039 images for training and 509 for validation, all of them being the colon synthetic part of the EndoSLAM dataset that was generated in Unity [34].

It is reported that $Endo - SfMLearner$ results for frames from the stomach, small intestine and colon are compared with their own algoritm without brightness loss integration ($Endo_w/o_b_1$) and without attention integration($Endo_w/o_a_1$), and also with others from literature: $SC - SfMLearner$, $Monodepth2$, published pre-trained Monodepth2 ($Monodepth2_{pre}$), $SfMLearner$ and published pre-trained SfMLearner ($SfMLearner_{pre}$).

The mean±std in non-clarified units of measure for Endo-SfMLearner was of 0.2966±0.0622 on stomach, 0.1785±0.0265 on the small intestine and 0.2722±0.0615 on the colon. These are the lowest when compared to the alternatives and the authors highligh the high performance of the first two.

It is stated that the comparison between their work alternatives ($Endo - SfMLearner$, $Endo_w/o_b_1$ and $Endo_w/o_a_1$) suggests that the main benefit of incorporating the ESAB block into PoseNet, alongside DispNet, is its ability to improve sensitivity to image texture. In contrast, the brightness-aware photometric loss primarily encourages the network to account for spatial variations in illumination across the image.

With regards to the comparison with the literature, those were trained with the same data and the optimal parameter set and, while both decisions were intended to support a fair and unbiased assessment, the second decision is more questionable, as using the optimal hyperparameters by their work may introduce a potential source of bias. These settings may give their method an advantage in the comparison. An alternative approach would be to establish the best-performing hyperparameter for the other methods. This would ensure that differences in performance are less likely to result from differences in hyperparameter optimization rather than from the methods themselves.

Scene rigidity assumptions and limitations

The Endo-SfMLearner framework also assumes that changes in the images are primarily caused by camera motion rather than by deformation of the observed scene. Thus, the observed anatomy is assumed to remain static, or sufficiently close to static for residual deformation to be neglected [33].

This becomes a limitation of the algoritm, as peristaltic motion is a characterist of the gastrointestinal tract. Generalisation to real-world scenarios remains a potential challenge in endoscopic depth estimation [34].

When considering the uterus, it consists of a muscular organ, so it may undergo contractions. For the uterine cavity, it is mechanically distended during the procedure, allowing intrauterine pressure and cavity distension to be controlled at a constant average [4]. This could provide a comparatively stable scene geometry during image acquisition at the same time that changes in distension pressure can alter the overall configuration of the cavity during a procedure. In addition, direct contact with the hysteroscope, bleeding and tissue manipulation during resection can introduce local deformation.

In this work, the rendered sequences also contain a static anatomical mesh that does not represent possible deformation. This makes it more adjusted to the algoritm (may contribute to reconstruction and depth-estimation errors) but gives it less realism, remaining a relevant limitation for future transfer to clinical hysteroscopic video. This is important to notice because successful performance on the synthetic sequences cannot, by itself, establish robustness and generalisation for clinical scenarios.

3.2.3 Proposed Approach

After identifying the algorithms from the literature that were relevant to the objectives of this work, they were evaluated against the requirements of the proposed approach. Several were subsequently discarded due to specific limitations or incompatibilities, leaving Endo-SfMLearner as the selected framework.

The proposed approach consists of fine-tuning the model originally trained by the authors using the generated and collected data for the uterine cavity. The rationale behind this decision is discussed below, followed by a clarification of the implementation adopted in this work. The flow of this section is illustrated in Figure 3.17.

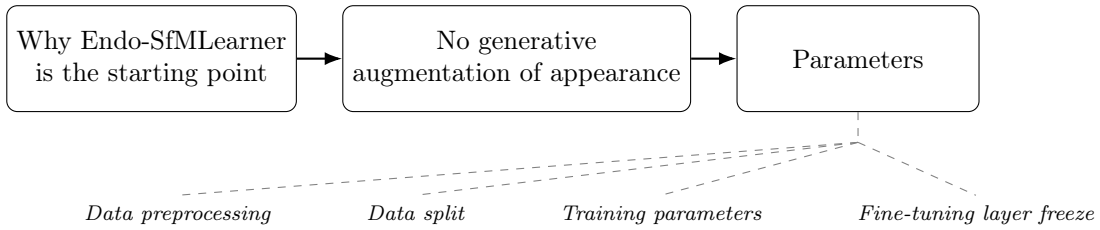


Figure 3.17: Overview of the proposed depth estimation approach.

Why Endo-SfMLearner is the starting point

The choice of Endo-SfMLearner as the starting framework is motivated by several complementary considerations.

First, its formulation is closely aligned with the requirements of the present work. It estimates both monocular depth and camera motion from endoscopic video, allowing the predicted 3D structure to be expressed in a common coordinate system and subsequently used for scene reconstruction. This way, a estimated version of the poses can be used for the 3D reconstruction whenever needed, for instance, if the clinical scenario does not allow for the direct collection of poses from the hysteroscopic navigation. Starting from a framework that already provides it avoids introducing a separate pose-estimation from a different work and preserves the self-supervised formulation on which the reconstruction pipeline relies.

Second, Endo-SfMLearner constructed on a previous algoritm that was meant for generic natural-image depth estimation and specifically developed it for endoscopic imaging. As mentioned, although its training data primarily represent the gastrointestinal domain, it is argued that the framework could transfer to other applications in mind, such as laparoscopic. The transition from gastrointestinal endoscopy to hysteroscopy, a endoscopy but not the same as for the gastrointestinal tract, represents a shift within the endoscopic domain, despite it being a broader endoscopic domain, with a change in anatomy and scene characteristics. The differences between the two settings are consequently not interpreted a priori as evidence that the method will fail to generalise. Instead, they provide a motivation for investigating whether fine-tuning can adapt the existing representations to the uterine cavity and how much adaptation is required.

In addition, the amount and diversity of data being as already explored, namely having less variety, not including pathologies and lesions and resulting of difficulties and time consumed in recording the scenes, also favour a transfer-learning strategy. The obtained dataset is not what would be desirable for training a deep monocular depth and pose estimator entirely from scratch. Starting from pre-trained representations reduces the amount of task-specific data that must be used to learn generic visual and geometric features, allowing the available data to be concentrated on adapting those representations to the target anatomy and imaging conditions.

A further motivation is the availability of pre-trained weights for the Endo-SfMLearner framework. The use of these weights makes transfer learning directly applicable rather than requiring the complete optimisation of both networks from random initialisation, which could be

done with the available EndoSLAM dataset. This is particularly relevant given the computational cost of training two convolutional networks jointly. Training from scratch would require the model to learn both general feature representations and the task-specific relationship between appearance, depth and camera motion, increasing the computational requirements and time needed.

The choice of Endo-SfMLearner should therefore not be understood as an assumption that the pre-trained model will transfer directly to hysteroscopy. Rather, it provides a controlled starting point from which the effect of domain adaptation can be investigated. Thus, while 3D reconstruction in the utery cavity can be explored, the performance obtained before and after fine-tuning can be measured in order to explore the suitability of the framework for hysteroscopy and the extent to which it can be adapted to another similar domain, but always having in mind that the dataset is synthetic, which can also influence results.

No generative augmentation of appearance.

When considering the literature on the adaptation of pre-trained weights and fine-tuning, one aspect is the use of generative appearance augmentation in synthetic images to address the sim-to-real gap. In this strategy, a generative adversarial network is trained to translate rendered frames into the style of real endoscopic images, or to synthesise several distinct appearances while preserving the underlying geometry of the images. This strategy was not adopted in the present work for several reasons.

First, the learned appearance transfer requires a set of real images to translate towards, and the only real hysteroscopic material available for this work (the two videos from CHUSJ) is reserved for testing to evaluate the 3D reconstruction in real images, which is fundamental. Training a translation model on it would place the evaluation domain inside the training pipeline, and any subsequent statement about performance on real video could be circular and the possible attributed credit with it generalising to the real images would be relatively weak, with the potential source of the result lying in the chosen methodology.

In addition, when considering the introduction of those images in the learned appearance transfer, for instance, with a split of the real images set, they could not be divided into the appearance transfer process and test and still remain a good sample for both, either in terms of size and appearance. On the one hand, there is the barrier of the amount of frames being reduced, which would prevent a good number of samples to evaluate in the test. On the other hand, the variety of appearance is limited, which could mean the same problem explored in the previous paragraph and the reason for the learned appearance would be lost, as its purpose is to widen the range of appearances the model is trained on.

To conclude, this does not mean that appearance diversity is not wanted or needed, but that the present work has this limitation for the dataset to be used and the consequence of an appearance diversity form real images provoked to the synthetic ones was not possible to be explored neither how it could affect the generalisation to in-vivo clinical frames.

Parameters

Endo-SfMLearner was applied to a total of 2420 frames, which is consistent with the thousands of samples from the algoritms in the literature that adapted from pre-trained weights (Table 2.9), even though those are larger datasets, with more than 15000 frames. Nevertheless, the amount used here is comparable to the number of frames with which the original authors of Endo-SfMLearner themselves considered to train the framework from the beginning rather than merely to adapt it, which were 2039 images for training and 509 for validation. In addition, the number of training instances presented to the network are increased by the geometric augmentation in the algoritm (horizontal flipping and random scale-and-crop operations), even though these do not introduce new scene content. Thus, the amount of data was considered sufficient to support fine-tuning, while not being regarded as sufficient to justify training the network from scratch.

Then, all frames were subjected to the preprocessing procedure.

Data preprocessing

The preprocessing stage was performed using a CPU-based environment, with Visual Studio Code version 1.134.0 and Python 3.12, together with the required Python libraries. The organisation of the folders used throughout the stage of the synthetic data processing is included in Appendix G.

All images were converted to a resolution of 384×384 pixels, corresponding to the resolution expected by the model. During this process, one of the problems reported in the original data, the possibility of frames consisting almost entirely of a single colour, making them unsuitable for training, was addressed by analysing the pixels and giving the frames a tolerance of five colours to identify and discard these cases. Invalid values in the depth maps were also removed before the data were passed to the model.

In addition, the intrinsic parameters stored in the dataset need to correspond to the image that is actually provided to the model. Therefore, these parameters were adjusted according to the final processed image rather than being kept according to the original Unity resolution.

The preprocessing was implemented in the `preprocessing.py` script, which performs the operations required to convert the original synthetic recordings into the format used by the model. An overview of the script is shown in Figure 3.18.

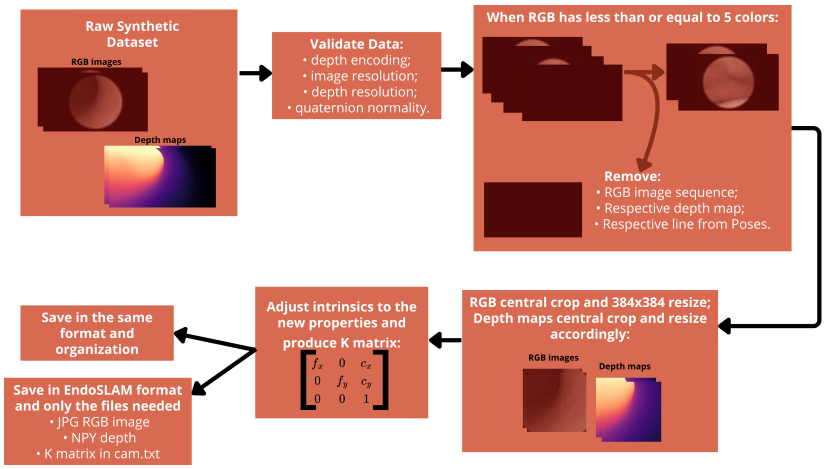


Figure 3.18: Main processing steps implemented in `preprocessing.py`.

A circular black mask was considered but not applied to the images, despite the circular field of view of the hysteroscope being black. The first reason is because the mask would not change what was seen, as the circular centre was already were the information was. The second is due to the fact that introducing an artificial black region around the image would add information that is not part of the actual scene and could therefore affect the model's interpretation of the image. This region was not particularly prominent or a substantial part of the image, as can be seen in the cropped and resized examples in Figure 3.18, but it could still affect the model's input. For instance, the model understanding that the black was a distance that corresponded to more depth, or the region aroung those black pixels could be influenced by them. This way, the original color, which is also darker than the content, remained with the error being inherent to the image and not produced by the mask.

Regarding the real data, it was subsequently included in the dataset as scenes 19 to 22, as already mentioned. For the videos, the frames were extracted directly from the corresponding video files. It should be noted that a frame decoded from an `.mp4` file has already undergone lossy compression, chroma subsampling, and potentially other processing applied by the recording device before the video was stored. Therefore, these frames do not represent the same type of raw image that would be obtained directly from the camera sensor.

Furthermore, the intrinsic parameters of the real camera at the source resolution of 1920×1080 were not available.

In the test video, the circular field of view was also positioned approximately 23 pixels to the left, which further limited the possibility of accurately reproducing the original camera geometry.

The extraction of the video frames for scene_19 and scene_20 was performed using the `extract_frames_videos.py` script, whose processing sequence is illustrated in Figure 3.19.

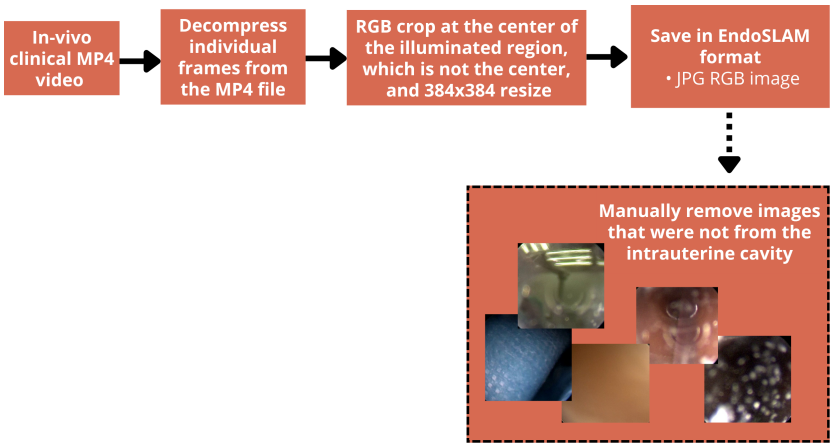


Figure 3.19: Processing steps applied to the real video data and images.

Then, the same video frames were centrally cropped using a 384×384 region so that these images could also undergo depth estimation, meeting the ***Spatial and Temporal Consistency*** requirement discussed in subsection 2.4.4, as previously mentioned.

For the real images containing polyps, introduced in scene_22, the preprocessing consisted of collecting and separating the individual images, which were then further cropped to obtain the same number of pixels in width and height, and subsequently resized to the same 384×384 resolution used for the other data so that they could be processed by the model.

Data split

From those 2420 frames, 272 were real, consisting of all of the frames that could be used from the clinical videos. From the remaining 2148 frames, 1458 were attributed to training, 335 to validation and 355 to testing.

This split was performed manually, as the algorithm does not randomly divide frames according to predefined percentages. Since training relies on temporal continuity between frames, the scenes had to be used as the unit of splitting, ensuring that frames from the same scene remained within the same subset rather than being distributed across training, validation and testing. At the same time, the split was designed to approximate the commonly used 70/20/10 proportion for training, validation and testing.

The split corresponded to approximately 68%/16%/17% of the synthetic data for training, validation and testing, respectively. The dvision of scenes is presented in Table 3.6.

Table 3.6: Partition of the synthetic dataset having the scenes as unit of splitting, with the frame count of every scene given in parentheses.

Split	**Scenes**	**Scene (frames)**	**Total frames**
Training	12	scene_1 (201), scene_3 (97), scene_4 (193), scene_6 (106), scene_7 (179), scene_8 (58), scene_10 (70), scene_11 (36), scene_12 (74), scene_13 (146), scene_14 (144), scene_16 (154)	1458
Validation	3	scene_2 (99), scene_5 (109), scene_9 (127)	335
Test	2	scene_15 (203), scene_17 (152)	355
Total	**17**		**2148**

The training and validation scenes are listed in `train.txt` and `val.txt` at the root of the dataset, which the data loaders read directly. Scenes 15 and 17 are exclusive for the test and were chosen due to the being associated with the uterus structure 22_59, which is not the structure of any other scene, as explored in Subsection 3.1.4.

Every scene was captured with the same intrinsics ($f_x = f_y = 824.05$, a horizontal field of view of 26.23°) with one exception, noted below.

Training Parameters

At the first interaction, the needed number of epochs was explored. Then, the tested variables were the learning rate, the batch size and the frame stride. Later, the possibility of freezing layers in fine-tuning was explored.

The number of epochs was determined by analysing the training loss and validation AbsRel curves, with particular attention to the point at which the training loss stabilised and further training no longer resulted in a meaningful improvement in validation performance. The validation AbsRel correspond to the value of the metric. No validation loss was calculated and given during the training, as the algoritm only collected it while training in the version `--without-gt`, and this was not the used mode. This analysis is presented in Subsection 4.1.1.

Learning rates of 10^{-5} and 5×10^{-5} were used, one and two orders below the 10^{-4} at which the framework was originally trained, on the reasoning that adaptation values would be below the pre-trained representation. These orders are also common for the algoritms from literature that adjusted to pre-trained weights, as explored in Subsection 2.6. Regarding the batch size, values of 4 and 8 were primarily used due to GPU memory limitations.

With regard to the frame stride, since adjacent frames in the slowly moving camera sequence provided limited parallax, increasing the frame stride was also explored, as it can enlarge the effective baseline and strengthen the depth signal, although at the cost of greater appearance variation between frames. Strides of value of three and four were trained and compared besides the version with no stride. These were the selected values to introduce a

relatively large temporal separation between consecutive training frames while avoiding excessively large inter-frame motion.

The loader's minimum-sequence-length check does not account for the selected stride, meaning that scenes with fewer than $2k + 1$ frames would contribute no samples without warning. The tested strides of 3 and 4 require at least 7 and 9 frames, respectively. This does not affect the present dataset, whose shortest scene contains 36 frames.

Fine-tuning layer freeze

Fine-tuning a network that was pre-trained on a different domain raises the question of how much of it should be allowed to change. By updating every weight, the model adapts, but this option also allows for it to discard representations that were expensive to learn and that the new domain does not provide enough data to relearn [91].

The freeze of layers was considered and possible to apply in the Endo-SfMLearner algoritm, which would be through the ResNet-18 encoder and evaluating how much of the pre-trained encoder should be allowed to change. And, since it is shared by both networks, freezing a given layer would affect both the depth and pose branches, which is consistent with their joint training, as freezing only one branch while allowing the other to adapt could introduce an imbalance in how the shared representation is updated.

For this, the training script was extended with a `--freeze` argument taking one level of four of them. Each level corresponds to a contiguous group of stages of the ResNet-18 encoder in the order since the input, as according to Table 3.7 and Figures 3.14 and 3.15.

Table 3.7: Level of freeze for the algorithm training layers.

Level	**Stages held fixed**
0	none; every weight is updated
1	`conv1`, `bn1`, `layer1`
2	`conv1`, `bn1`, `layer1`, `layer2`
3	the whole encoder (`conv1` to `layer4`)

This way, when freezing is selected during training, `requires_grad` is set to `False` for the identified layers, and the frozen layers were selected as contiguous prefixes of the network. To also avoid the module to keep updating the mean and variance of those normalisation layers, the frozen modules were placed in evaluation mode at the start of every training epoch. Each level was trained under otherwise identical conditions, with the same seed, the same learning rate, the same batch size and the same data split, so that the level is the only variable.

There is no clear indication of how freezing these layers affects the network. Analysing their impact can therefore help assess whether this strategy is necessary and provide insight into the relative importance of the encoder within the overall architecture.

Results and comparisons are presented in the next chapter.

3.2.4 Interface and Dynamics

Figure 3.20 illustrates the dynamics involved in training, validation and test with the Endo-SfmLearner and the generated dataset.

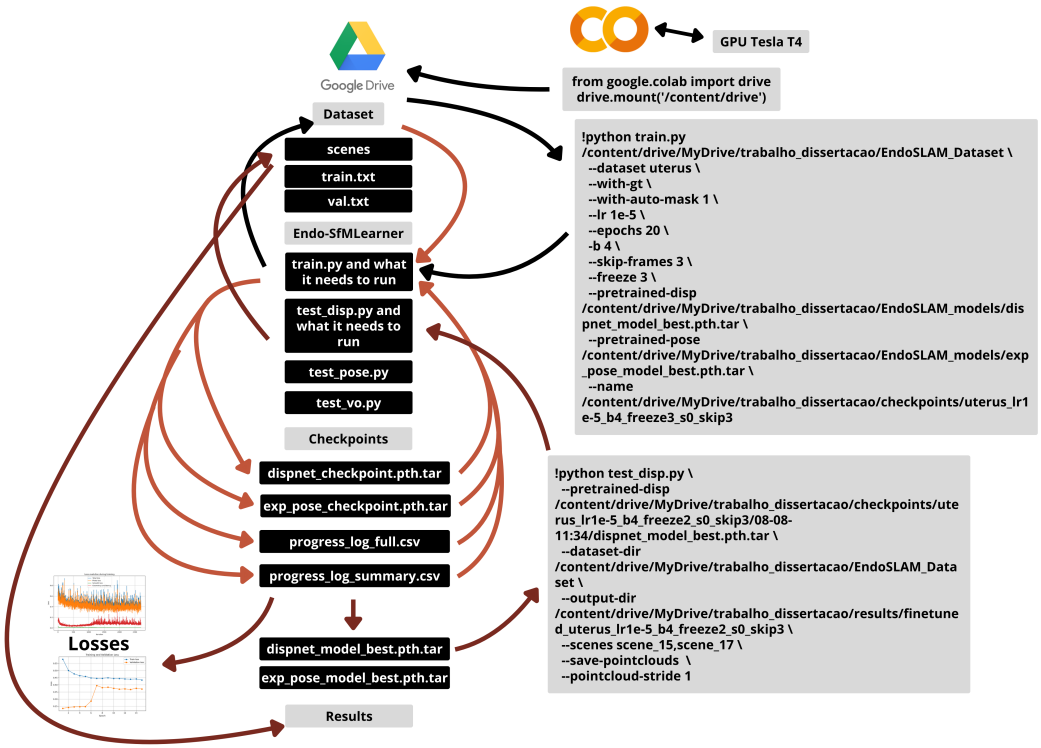


Figure 3.20: Illustration of the dynamics for the Endo-SfMLearner algoritm training, validation and testing.

Training and inference were performed in Google Colab, with the data accessed through Google Drive. This environment was used to provide access to GPU acceleration, which was not available on the local machine. The experiments therefore used a NVIDIA Tesla T4 GPU with CUDA 12.8 and PyTorch 2.11.0. The depth-estimation metrics were also computed within the same environment as the training and inference runs, avoiding the need to download the generated predictions.

Several modifications were required to the original Endo-SfMLearner scripts. These changes were introduced either to accommodate differences in the data loader and software versions, to implement the frame-stride experiments in `train.py`, or to add outputs required by this work, such as depth maps per frame and computational time and cost measurements for the clinical-applicability analysis described in a later subsection.

Once the adapted framework was established, training was performed using only the image data and the intrinsics matrix.

The released EndoSLAM checkpoint provides the initial weights for both the encoder and decoder. These pre-trained weights are then adapted to the synthetic uterine sequences through the same self-supervised training objective. Training was performed using only the

image data and the intrinsics matrix.

The depth network produces a dense depth map for each frame, while the pose network estimates the relative six-degree-of-freedom motion between frames. At the end of each epoch, a checkpoint is saved to the Google Drive volume attached to the notebook, together with the best-performing depth and pose models. Two types of training logs are also retained: a per-epoch summary and a per-iteration record of the individual loss components. The latter allows the photometric, smoothness and geometry-consistency terms to be examined separately rather than only through their weighted sum, which is useful when assessing the contribution of the individual components.

The dataset was split so that the validation frames originate from recordings that were not seen during training, and all sequences had characteristics that varied between them, as already described, so both training and validation had the possible diversity. These are two considerations that end up trying to limit the risk of overfitting. In addition, the learning rate was kept substantially below that used for the original training, limiting the extent to which the pre-trained representation could be displaced by the relatively small adaptation dataset. Validation metrics, computed after every epoch, also help monitoring overfitting by the possibility to identify a divergence between training behaviour and validation performance, which may indicate that the model is increasingly fitting the training sequences rather than improving its generalisation. However, none can guarantee its absence.

After training, predictions are generated using `test_disp.py` and stored as one depth array per frame. The scenes included in testing are specified explicitly through the command line, for example `-scenes scene_15,scene_17`.

3.2.5 Per-frame Scale Factor

As a self-supervised monocular network recovers depth only up to a global multiplicative factor, as already explored in the literature, a scaling factor needs to be applied.

For the depth estimation results, it was performed for each frame,

$$r_i \;=\; \frac{\operatorname{median}\big(D_i^{\mathrm{gt}}\big)}{\operatorname{median}\big(D_i^{\mathrm{pred}}\big)}, \tag{22}$$

computed over valid pixels, considering ground-truth depths within $[0.1, 50]$ scene units and finite, positive predictions.

Before this calculation, the ground-truth depth was converted from ray length to the view-space z coordinate to ensure consistency with the network output. Scale consistency across each sequence was assessed through the dispersion of r_i, expressed as the standard deviation-to-mean ratio, as well as through its range, $\max_i(r_i)/\min_i(r_i)$. These measures were also computed for the 30-frame subsample used by the fusion scale estimator, allowing the consistency of the scale estimated for fusion to be assessed separately.

It is important to mention that there were experimental limitations, namely the computational resources available and time, that did not allow to repeat trainings with other epochs or after some of the analysis.

3.2.6 Per-frame Depth Representation in Point Clouds

Point clouds from the predicted NPY files depth were generated to auxiliate the analysis as an extra, if needed.

These individual point clouds are initially expressed in the coordinate system of their respective camera, so they cannot be directly combined to represent the complete scene. For this, a camera trajectory is always required to relate the frames spatially and reconstruct their geometry in a common reference system, so no fused point clouds could be created for the real images.

3.2.7 How the Depth Estimation Results are Measured

Depth is evaluated in the `aggregate_results.py` script per frame against the rendered ground truth when the ground truth lies within a valid range, the pixels corresponding to the outside of the circular image not included. The evaluation metrics were the absolute relative error, squared relative error, root mean squared error and its logarithmic form, mean absolute error, the mean absolute error in $\log_{10}$ space, and the relative threshold accuracies $\delta < 1.25$, $\delta < 1.25^2$ and $\delta < 1.25^3$.

These metrics are accompanied by a standard deviation error that corresponds to the frame-to-frame dispersion. Consecutive frames are independent in terms of depth prediction, but statistically they are not independent observations. This is, they are temporally correlated and one frame can be very similar to the next, which affects this analysis, so the reported standard deviation should not be interpreted as a standard error of the mean or as an indication of the statistical uncertainty of the mean.

With regards to the real frames, no depth ground truth exists, and none can be obtained without an instrument that was not part of the acquisition, so none of the above applies to it. The real frames are therefore assessed qualitatively, and the one quantitative statement that can honestly be made about them is a check of scale and right order of size and proportion. This does not give an error at any point, but a form of verification that the synthetic evaluation cannot provide on its own, since a model can be accurate on rendered frames and still not be able to generalise.

As images were cropped, a brief analysis on the comparison of the appearance for the image and its cropped version was performed. AbsRel, RMSE and δ_1 were evaluated for each frame, the last two with per-frame median scaling to isolate the structure, the wide prediction used as the reference.

To conclude, although the procedure above follows the convention used by the method this work builds on, the results are not directly comparable with those originally published.

First, the units of the metrics for which units are required are not specified, and it is not assumed that the two evaluations are based on identically defined reference quantities. Both studies start from a synthetic Unity environment but, while in this work it was possible to explore the mesh that is exported from the Unity, understand its dimensions and compare them with reported values for the uterus, the Endo-SfMLearner authors do not make that

same Unity mesh available.

Second, even if the reference quantities were defined consistently, the difference in anatomical domain would still affect the interpretation of the comparison, as the two evaluations concern different anatomical structures.

When considering that fine-tuning was performed, other works that also adapt pre-trained weights could provide a more relevant basis for comparison. However, their evaluations still concern different anatomical structures and therefore do not allow for a direct comparison.

3.2.8 Expected Results for Depth Estimation

Some hypotheses were established regarding depth estimation.

Regarding the component losses, the photometric loss is expected to exhibit a more irregular evolution, with more pronounced fluctuations throughout training. Its value is also not necessarily expected to approach zero, given that it represents the photometric discrepancy between the target and reconstructed images.

Then, regarding the depth predictions, a first consideration concerns the fact that depth is obtained from the reciprocal of disparity, which does not provide uniform depth resolution. At larger disparities, corresponding to shorter distances, small changes in disparity produce relatively small changes in depth. Conversely, at smaller disparities, corresponding to greater distances, the same change in disparity produces progressively larger changes in the estimated depth. Consequently, uncertainty or errors in disparity are amplified at greater distances, potentially resulting in lower metric depth precision for more distant regions of the uterine cavity. Thus, the more distant regions of the uterine cavity are expected to present less depth precision.

Subsequently, the adaptation to uterine anatomy is expected to improve the predictions relative to applying the pre-trained model without adaptation. The pre-trained network has learned visual and geometric patterns from gastrointestinal endoscopy, where the cameramoves through tubular structures with the aspect of the tissue for each organ. Although the uterine cavity differs anatomically, similarities in visual appearance and local geometry are still present, particularly in the predominance of tissue surfaces and the gradual variation of distance from the camera. Therefore, it is expected that synthetic uterine sequences used for fine-tuning provide an opportunity to adapt the learned representation to the geometry of the target anatomy.

Nevertheless, some ambiguities are expected to remain, for instance, in the sequence of frames dominated by a relatively uniform tissue surface, as the visual information available to the network may provide limited evidence about how depth varies across the scene. Consequently, similar predictions across such frames may not necessarily indicate that the network has learned the underlying geometry, making it difficult to draw a clear conclusion from these cases alone. The adaptation is therefore expected to improve the representation of uterine geometry, but not necessarily to reproduce the full anatomical structure with the accuracy of the rendered ground truth.

The frame stride is expected to have a better performance than the no skipping versions,

as increasing the frame separation could, in principle, provide larger camera displacement and promove a learning with stronger geometric cues.

In addition, the similarity between the synthetic sequences also introduces a potential risk of overfitting. Although the scene geometry and visual appearance were varied, the training data do not reproduce the full variability of real hysteroscopic images. Good performance on the synthetic validation set may consequently reflect adaptation to the characteristics of the simulated domain rather than a complete acquisition of the underlying geometric cues.

This may imply that real images will not have the same performance as for the synthetic sequences. Nevertheless, it is expected that, for the real images, the most prominent structures in the scene will be captured reasonably well, also due to the previous training by the authors of Endo-SfMLearner, whereas regions with more complex, more challenging to interpret or less familiar visual characteristics may be reconstructed less accurately. Moreover, the depth estimated by the network is not necessarily expected to match the true metric depth. In particular, regions at greater distances may be predicted as being closer than they actually are, due to the mentioned detail of the disparity-to-depth conversion and since the synthetic images used for fine-tuning did not include the full range of distances present in real uterine structures.

3.3 3D Reconstruction Pipeline

The 3D reconstruction stage was included to obtain a geometric representation of the uterine cavity through their fusion into a representation of the complete scene.

3.3.1 3D reconstruction Fusion Procedure in Endo-SfMLearner

The reconstruction procedure of the work by Ozyoruk et al. [34] differs from this requirement. Their reconstruction pipeline assembles frames into a single stitched panorama and subsequently generates a point cloud from this representation.

The "README" describing the reconstruction procedure reports four main stages: stitching multiple frames into a single panorama using the method of Brown and Lowe, inpainting damaged regions using OpenCV and manual editing in GIMP, obtaining depth using the aforementioned methods and finally generating the point cloud.

The final point-cloud generation implemented in the released `depth.m` script does not use the camera model or camera motion. The image is converted to grayscale and the pixel intensity is directly assigned to the Z coordinate over a regular grid.

Thus, the resulting coordinates are defined by a regular X–Y grid, while the Z coordinate is obtained from the image intensity. No camera intrinsic matrix is used, no pixel is back-projected through a camera model, and no camera pose is applied during this step. The subsequent scripts convert the coordinates to the required format and generate the final PLY file, after which alignment with the scanned reference is performed using ICP in CloudCompare.

This procedure is therefore fundamentally different from the multi-view reconstruction

required in the present work. Rather than placing observations from multiple calibrated views into a common three-dimensional reference frame, it produces a surface representation from a stitched image. Consequently, the surface errors reported by the authors cannot be directly compared with those obtained from the reconstruction developed here.

The treatment of scale is also different. In the Endo-SfMLearner's reconstruction, the scale of the final representation is not obtained solely from the reconstruction process. Instead, the generated point cloud is subsequently aligned to the scanned reference using ICP. Therefore, the scale ambiguity inherent to monocular depth estimation is handled through the external alignment procedure rather than through the multi-view reconstruction itself. This is particularly relevant to the present work, where scale consistency between consecutive depth predictions is required before the frames can be fused into a coherent scene representation.

For these reasons, this reconstruction procedure was not adopted as the reconstruction methodology for the present work.

3.3.2 3D reconstruction Fusion Procedure in this Work

A fusion procedure was developed to combine the frame-wise point clouds using the corresponding camera poses. The corresponding script is included in Appendix H. The procedure consists of four main steps: back-projection of the depth maps, conversion to the required axis convention, voxel filtering and scale estimation.

For the scale estimation, a single scale factor is computed for the entire sequence from the ratio between the medians of the predicted and ground-truth depths over a sample of frames. With a single factor and not a scaling for each frame, the inter-frame scale variation that the reconstruction is intended to expose is possible to be present. In addition to the sequence-level scale factor, the scale estimated for each sampled frame is retained. Their range and relative spread are then reported as indicators of scale consistency. A threshold of 15 % is used only to flag potentially relevant variation in the output. It is a reporting threshold rather than a criterion for accepting or rejecting a reconstruction; the underlying scale distribution is reported separately.

The resulting point clouds were inspected both through generated views from selected perspectives and interactively in CloudCompare v2.13.2.

For this reconstruction, the poses recorded during data acquisition in Unity were used. These poses were processed by the same `preprocessing.py` script used to convert the recordings into the dataset format required for training, with the processing performed through Visual Studio Code 1.134.0. Using the recorded poses rather than the poses estimated by `test_vo.py` was a deliberate methodological choice. This way, it was possible to isolate the effect of depth estimation when comparing the predicted reconstruction with the ground-truth geometry, rather than introducing an additional source of variation.

This also provides a consistency check for the pose transformations used in the reconstruction. Since the recorded poses correspond to the same Unity coordinate system in which the ground-truth geometry was generated, the point clouds can be transformed into that common reference frame without an additional registration step. Incorrect application of the poses,

for example, through an incorrect quaternion ordering, sign convention, or coordinate-system convention, would cause the individual frames to be placed inconsistently, not forming the expected continuous surface. Thus, the successful spatial composition of the frames also provides a practical verification of the pose transformation used in the fusion procedure.

Thus, as poses are not obtained as an output of the Endo-SfMLearner method, ATE and RPE were not calculated for these poses, as these metrics would evaluate the trajectory estimated by `test_vo.py`. In that case, the estimated trajectory can be exported in KITTI format and evaluated using a script available by the Endo-SfMLearner authors, `kitti_eval/eval_odom.py`, which reports the ATE and RPE.

While pose estimation is not independently introduced as a second variable during point-cloud fusion, it is still part of the Endo-SfMLearner formulation and is therefore present in the predicted depth results and available to use in a clinical environment as the only option to obtain camera poses from the hysteroscopic navigation system.

In addition, no surface mesh was generated in this work. The proposed coverage-map strategy can be implemented directly from the reconstructed point clouds, and a mesh could be generated from the same points if a later application were to require a continuous surface representation.

Finally, both reconstruction alternatives were also analysed in terms of their computational time, memory requirements and overall processing cost. These aspects are considered in a different subsection below, where cost is treated not necessarily as a fixed quantity but according to how it scales with the amount of data and the number of frames being processed.

3.3.3 Global Scale Fator

Fusing a sequence into a common coordinate system requires a single factor for the whole trajectory,

$$s \;=\; \frac{\operatorname{median}\big(D^{\mathrm{gt}}\big)}{\operatorname{median}\big(D^{\mathrm{pred}}\big)}. \tag{23}$$

If the network's implicit scale varies between frames, no single factor aligns them all. This was assessed on the two synthetic sequences for which both a ground-truth depth map and a ground-truth camera trajectory are available (scenes 15 and 17). The analysis was performed both for the pre-trained version and a fine-tuned one.

3.3.4 How the Fused Point Cloud Reconstruction Results are Measured

The visual results of the point-cloud reconstruction were generated in Visual Studio Code 1.134.0 on the local computer, as this environment provided colour rendering and allowed the use of Open3D, which was not available in Google Colab.

To analyse results, the reconstruction with the fused ground-truth is used and the accuracy RMSE, completness RMSE, Chamfer and Hausdorff distances are computed.

The first two differ, as the first measures the distance to the closest real point for every predicted point, exploring if the obtained points are in the surface, and the second measures

the distance to the closest predicted point for each true point, exploring if the reconstruction is complete, or regions are missing.

The Chamfer and Hausdorff distances are reported to assess the geometric agreement between the predicted and ground-truth point clouds. The Chamfer distance provides an overall measure of spatial discrepancy between the two point sets, in this work corresponding to the half-sum of the accuracy and completeness distances, whereas the Hausdorff distance captures the largest such discrepancy, and is given as the symmetric maximum and in both directions separately. These measures are suitable for when the geometric results differ in density or contain a different number of points.

As the Hausdorff distance is a maximum, thus being determined by the single worst point in the cloud, one stray back-projection can dominate it entirely, so it is accompanied by the 50^{th}, 90^{th}, 95^{th} and 99^{th} percentiles of the same distances, as they describe the same distribution without refering to its extreme.

As mentioned, time and computational resources considerations were also part of what was measured, being collected through the Google Colab setup so that the same computational resources of the other components remained for this evaluation. Those considerations are explored in another subsection below.

3.3.5 Expected Outcomes for the Point Cloud Fusion Procedure

With regards to the the ground-truth reconstruction, since the camera poses recorded in Unity are used to transform the frame-wise point clouds into the common reference system, it is expected to preserve the correct spatial organisation of the uterine cavity. This also provides a controlled reference against which the effect of the predicted depth can be assessed.

For the predicted reconstruction, it is expected to provide a complementary assessment of the depth predictions rather than simply reproduce the conclusions obtained from the per-frame depth metrics. The quality of the reconstructed point clouds is not expected to directly follow the depth metrics. Poor depth metrics do not necessarily imply that a meaningful point-cloud reconstruction cannot be obtained, as some errors may have more impact when the predictions are integrated across frames. Conversely, errors that are not captured by the per-frame evaluation may become apparent during reconstruction. In particular, an estimate whose scale drifts from frame to frame is independently corrected before being evaluated, and a metric computed in this way cannot detect scale drift across the sequence. However, an agreement between improvements in the depth metrics and reconstruction quality would therefore provide stronger evidence that the adaptation benefits the intended downstream application.

At the same time, the predicted reconstruction is not expected to reproduce the uterine surface with the same structural accuracy or level of detail as the ground truth. Consequently, the results are not expected to be sufficiently accurate for alternative applications requiring detailed surface inspection, such as precise polyp localisation or anatomical measurements. Nevertheless, it may provide sufficient spatial information for the intended coverage map application, the application that motivates the methodological steps undertaken in this work but

remains beyond its scope, as, for this application, the overall distribution of the reconstructed surface is more relevant than fine anatomical detail.

3.4 Time and Computational Requirements for Clinical Applicability

Given the intended clinical application of a coverage mapping, which would provide a spatial representation of the uterine cavity and track the progress of the procedure, the complete pipeline must also be feasible within the computational constraints of the clinical setting. This includes meeting the available computational resources and operating without introducing a significant burden to the procedure.

In particular, the additional information and support provided to the procedure by the pipeline, ultimately intended to support the procedure through a coverage mapping, although this step is already not in the scope of this work, should assist the clinician without interfering with the intervention. This is, excessive computational load, system freezing or the need for the clinician to pause or substantially slow down navigation would limit the practical value of such a system. This is particularly relevant in hysteroscopy, where procedures are generally short and maintaining efficient navigation is desirable.

Therefore, an evaluation on the needed time and the computational cost for processing various steps in the proposed pipeline are considered. This way, a factor identified in the literature as relevant but being underexplored, the computational resources, is also taken into account in the present work.

3.4.1 Used Harware

The computational measurements were performed using the hardware required by each stage of the pipeline. GPU acceleration was used for the neural-network inference stages, whereas preprocessing was performed on the computer's CPU, "AMD RADEON(TM) GRAPHICS", using NumPy and CPU-based Open3D routines, without GPU acceleration and, in fact, these main operations are already CPU-based numerical computations. The reported times therefore represent the computational cost of the implemented CPU-based reconstruction, while further acceleration may be possible through GPU-based implementations.

Thus, comparisons between the neural network and remaining stages should consequently be interpreted in light of the hardware on which each was executed, at the same time that the inference measurements should be interpreted as an estimate of the computational cost on the hardware used in this work, rather than as a general limit on the performance achievable with a dedicated accelerator.

3.4.2 What is and is Not Instrumented

A shared instrumentation procedure was used across the different stages so that their measurements could be compared directly.

What steps are relevant for evaluation

As the ultimate goal is to enable coverage mapping, which was neither implemented nor evaluated in this work, it is important to assess the computational requirements of the main pipeline stages that provide the basis for this application. These include dataset preprocessing, depth estimation during training and inference and 3D scene reconstruction. Although camera pose estimation is also necessary for the intended application, it was not implemented or evaluated in this work.

Dataset preprocessing is relevant because, in a clinical setting, the images acquired during the hysteroscopic procedure would need to be prepared for depth estimation. For each input frame, the preprocessing stage includes frame reading, the required image operations, such as cropping and resizing and the writing to the Endo-SLAM format.

Depth prediction is then performed through `test_disp.py`. An analysis on the step for the depth prediction is essential because it provides the depth information required for subsequent coverage mapping. This stage includes the network forward pass, in which the input image is processed by the encoder and decoder to produce the disparity and corresponding depth, as well as the writing of the predicted depth and, when enabled, the corresponding per-frame point cloud.

The 3D fusion reconstruction stage was also considered, as computing coverage involves back-projecting each depth map, transforming it with the estimated pose and accumulating the result in a common frame, which is precisely what the fusion step does. What the 3D fusion includes that is not needed for the clinical application is the obtained serialised .PLY cloud and a dense reconstruction, as a live implementation would maintain an incremental voxel occupancy structure rather than fusing all frames and downsampling at the end, which bounds its memory independently of sequence length.

If pose estimation were performed as part of theis work's pipeline, its computational cost would also need to be included in the corresponding real-time requirements.

Input Resolution and Computational Cost

The input frames are resized to 384×384 pixels before being provided to the depth estimation network. This resolution is part of the defined preprocessing pipeline and is therefore not considered an optimisation of the method itself. Nevertheless, using a lower input resolution naturally reduces the number of pixels processed during depth estimation, when using the `test_disp.py` script, and when obtaining the point clouds, and contributing to a lower computational cost.

What did not make sense to evaluate

When considering the training, training cost is excluded from this analysis, as training is performed before clinical use and therefore does not form part of the computational workload during a clinical examination.

Considerations of the procedure to evaluate

Operations such as final point-cloud fusion, sequence-level output generation or sequence-level scale estimation are performed only once per recording, whereas other operations are repeated for every frame. These one-off operations are reported separately from recurring per-frame costs to avoid making the computational requirement depend artificially on the duration of the recording.

GPU operations are synchronised before timing. Since CUDA operations are executed asynchronously, synchronisation is required to ensure that each separately timed stage has actually finished before its execution time is recorded. Thus, while it is necessary, it does not represent a measure that would be required during clinical operation.

The initial executions are excluded from the recurring-cost measurements. The first executions may include one-off costs associated with context creation, algorithm selection, memory allocation, and other initialisation procedures. These costs are not representative of the sustained processing rate and are therefore treated separately.

3.4.3 What is Recorded

For each operation, the processing time is recorded, including its median, 95th percentile, total duration, and contribution to the total processing time. Memory usage is also recorded for the process as a whole and, when a GPU is used, for the memory allocated and reserved by the accelerator.

In turn, the total processing time per frame is calculated by first summing all operations performed for that frame and then calculating the summary statistics over these per-frame totals. This approach is preferred to simply adding the median duration of each operation, since operations performed on the same frame may be correlated and their individual summary statistics should not be treated as independent quantities.

3.4.4 Time and Computational Requirements Expected Results

When considering the expected results, the first one is that no relevant differences are encounteres throught the different configurations for training, as the depth prediction being evaluated happens when these parameter changes are not influencing the processing. Instead, differences in time are expected in the training when a batch size is different or layers are frozen, for example. Reduced differences are still expected due to execution variability.

Then, the preprocessing stage is expected to require little computational time, since it is limited mainly to operations such as cropping, resizing and formatting the frames for the network input. Similarly, depth estimation is expected to be suitable for frame-wise processing, as each frame can be passed independently through the trained network without requiring the reconstruction accumulated from previous frames. Its computational cost is therefore expected to remain relatively stable throughout a recording, apart from initialisation and other one-off operations.

The 3D reconstruction is expected to represent a larger computational cost. Unlike depth estimation, it requires the predicted depth to be back-projected into 3D, transformed using the camera pose and progressively combined with information from other frames. In particular, operations involving the accumulated point cloud may become more expensive as the recording progresses. Consequently, the reconstruction is not necessarily expected to meet the per-frame budget in its complete form. However, this does not necessarily mean the approach is unsuitable for clinical use, since the coverage map, the intended future application, which is beyond the scope of the present work, does not need to be updated at the same frequency as the video itself.

A clinically useful implementation could therefore use different temporal requirements for different outputs. Depth estimation and other operations required continuously during navigation would ideally keep pace with the video, while the point-cloud fusion and, then, coverage map updates could be performed less frequently or asynchronously. If possible, these steps may also benefit from optimisation.

The expected computational requirements also differ between stages. Neural network inference is the component most directly dependent on GPU acceleration, whereas the point-cloud reconstruction and evaluation procedures used here rely primarily on numerical operations implemented in NumPy and related libraries.

Regarding the deployment, the computational requirements are expected to suggest that access to a GPU would be beneficial, particularly for the depth estimation stage. However, the results are not expected to require a high-end or specialised accelerator, and a conventional consumer-grade GPU, such as a NVIDIA GPU, is expected to be sufficient. In turn, adaptations to the software currently used for hysteroscopy may be required, particularly to integrate the proposed processing pipeline with the existing imaging and navigation system.

Chapter 4

Results and Discussion

This chapter presents and discusses the results obtained throughout the development and evaluation of the proposed approach.

4.1 Depth Prediction Results

The performance of the depth prediction results of the proposed approach is assessed from complementary perspectives, combining an analysis of the training losses with quantitative and qualitative evaluations of the predicted depth.

4.1.1 Analysis of the Global and Component Losses and the Validation Metric

The training losses are obtained throughout the training process and are accompanied by the validation results, with the AbsRel metric. No validation loss is obtained, as already explained in Subsection 3.2.3. The global training loss and the validation AbsRel were collected and analysed from the first experiments in order to understand the behaviour of the training process. Examining the evolution of these curves provides an indication of whether the number of epochs initially selected to test is sufficient for the training loss to reach a stable region and whether the validation performance has already reached a plateau.

Figure 4.1 presents this initial analysis on the training loss for some of the first runs and parameter configurations tested.

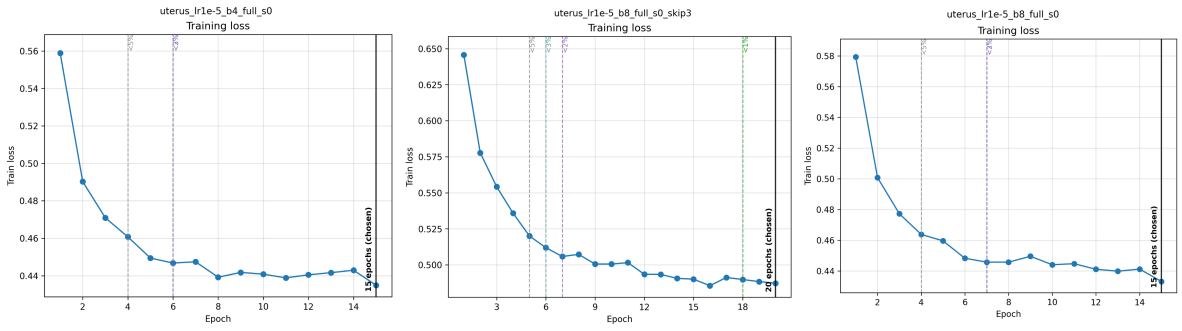


Figure 4.1: Training loss for three configurations for the learning rate 10^{-5}: batch 4 and no skipping; batch 8 and skipping; batch 8 and no skipping.

When considering more than one model, a common behaviour can be observed. The overall trend for the training loss curve is downward and is followed by a period of relative stability. The curves therefore suggest that the optimisation reaches a plateau rather than continuing to change substantially during the later epochs.

Based on this behaviour, the number of epochs was selected, but not solely as an arbitrary fixed value early in the graph. Instead, the number of epochs was both selected according to a stabilization point and according to a margin ensure that the model was not stopped immediately when the loss first entered a stable region. This stabilisation point was defined as the point after which the training loss varied by less than 2% within a three-epoch window.

For example, in the run with learning rate 10^5 and batch size 4, the training loss first satisfies this criterion at epoch 6, with a variation of 0.60% over the considered window. Training was nevertheless continued until epoch 15 to provide an additional margin after this point. During the last five epochs, the maximum variation between consecutive epochs was 1.81%, while the variation considering the complete five-epoch window was 1.34%.

The configuration of learning rate 10^5 and batch size 8 shows a similar behaviour. The training loss reaches the stabilisation criterion at approximately epoch 7, with a variation of 0.56%. Training was continued until epoch 15, providing an additional margin after this point. During the last five epochs, the variation considering the complete window was 2.46%, while the largest variation between consecutive epochs was 1.83%.

For the run with learning rate 10^5, batch size 8 and frame skipping 3, the training loss reaches the stabilisation criterion at approximately epoch 18, with a variation of 0.59%. Training was continued until epoch 20. During the last five epochs, the variation over the complete window was 0.58%, while the largest variation between consecutive epochs was 1.18%.

When analysing results, the validation curves were also explored, their behaviour presented in Figure 4.2.

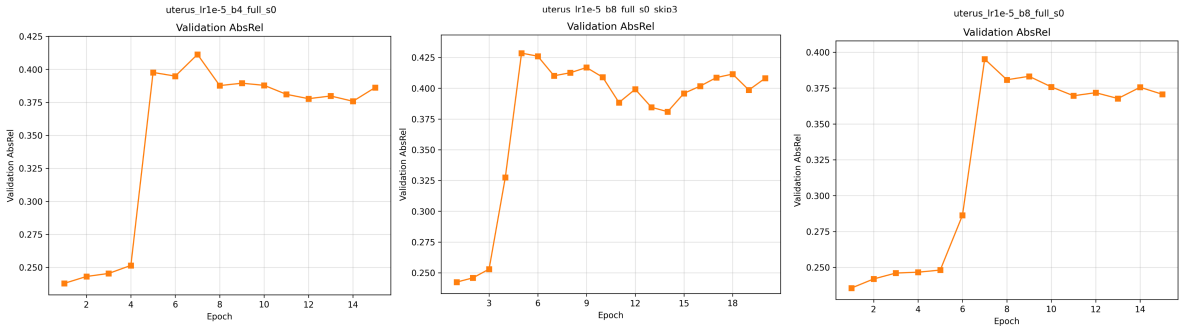


Figure 4.2: Validation AbsRel for three of the trained configurations for the learning rate 10^{-5}: batch 4 and no skipping; batch 8 and skipping; batch 8 and no skipping.

An initial transient behaviour can be observed, which could suggest either some degree of overfitting or the need for the first epochs to warm up before reaching a more stable training behaviour. If this initial transient were interpreted as overfitting, the subsequent behaviour would not be adequately accounted for, even with the relatively narrow range at the latter stages.

However, these observations were taken into account, and further analysis revealed that, in these cases, the epoch selected for the best model was the first one. This led to an investigation of the model-selection procedure, which showed that the validation AbsRel was the only metric considered by the algorithm when selecting the best model, with the model corresponding to

the lowest validation error being retained. In these cases, this resulted in the initial epoch being selected as the best model. This selection criterion may, in fact, be appropriate in this context, since, if the observed behaviour is indeed caused by overfitting, as considered the most likely explanation, selecting a later model would potentially retain a model that had already started to overfit. The fact that the best model corresponds to an early epoch may therefore indicate that the training data are contributing to the observed overfitting behaviour.

In turn, a higher learning rate was analysed. Figures 4.4 and 4.5 present the obtained behavior.

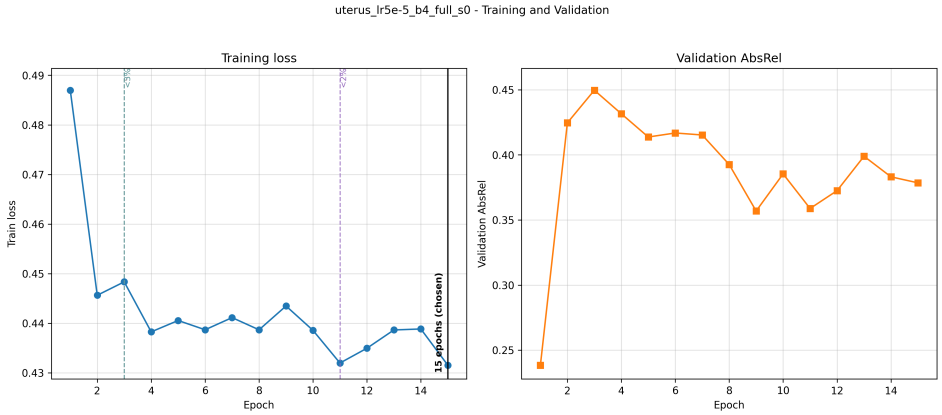


Figure 4.3: Training loss and validation AbsRel for the configuration with learning rate 5×10^{-5} and batch size 4.

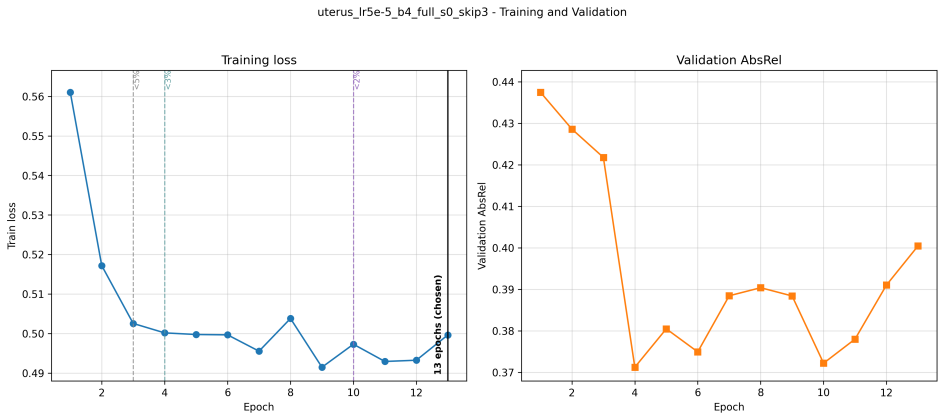


Figure 4.4: Training loss and validation AbsRel for the configuration with learning rate 5×10^{-5}, batch size 4 and frame skipping of 3.

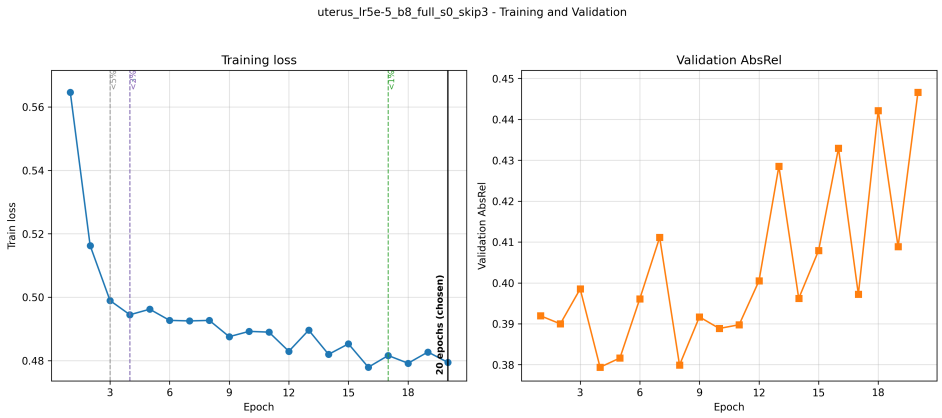


Figure 4.5: Training loss and validation AbsRel for the configuration with learning rate 5×10^{-5}, batch size 8 and frame skipping of 3.

While the first showed a similar behaviour to the previous analysis, the others differed.

For the alternative of the same configuration but with frame skipping (Figure 4.4), the validation AbsRel curve initially adjusted and had a minimal fluctuation until more relevante ones started. The training loss curve had already plateaud at epoch 4, which was the one considered for the best model. For this configuration, it is concluded that the best model is competitive and relevant for the depth estimation and point-cloud analysis.

The latter configuration presented a validation curve with later additional fluctuations while the training loss continues to decrease. This is compatible with overfitting, but it may also reflect unstable optimisation caused by the higher learning rate and its interaction with the batch size and frame sampling strategy. The epoch considered for the best model had not learned from that stage, but the previous epoch of the one considered revealed a higher value of the validation error, which may also indicate that the best model for this configuration could have suffered from overfitting.

Consequently, this learning rate seemed to be preferable to the learning rate of 10^{-5}. The trainings for the configurations with frozen layers had already been performed and the validation curves, presented in Figure 4.6, showed that, for them, the best model was also for the epoch 1. This is according to what was expected at this point, as they were performed for the learning rate of 10^{-5}. However, an analysis on the encoder and the possible level of freeze was already possible by comparing the results for those attempts with the one with no frozen layers.

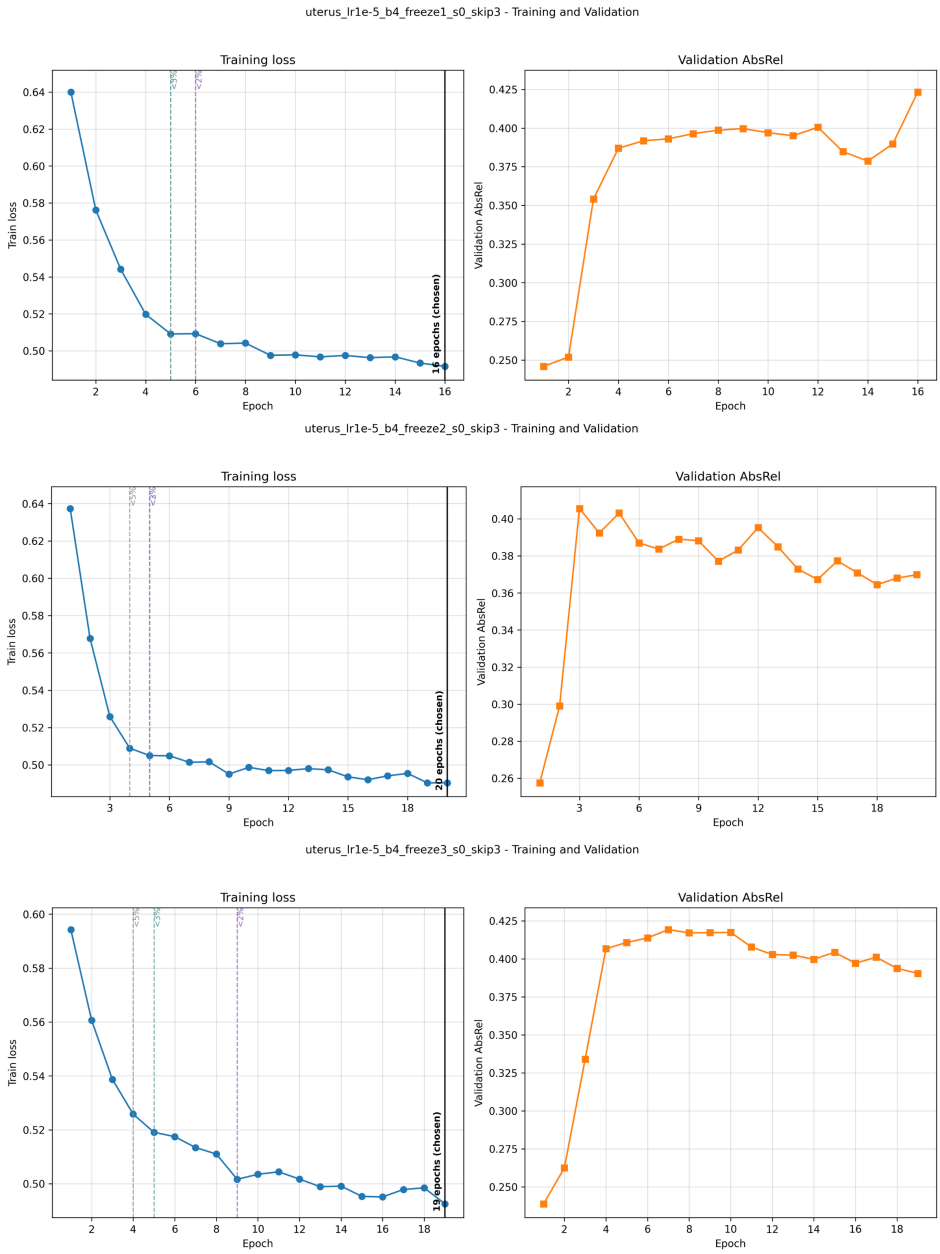


Figure 4.6: Training loss and validation AbsRel for the configurations with frozen layers, all with learning rate 10^{-5}, batch size 4 and frame skipping of 3.

To conclude, the loss curves also show that a lower training loss does not necessarily correspond to a better validation result, and a decreasing training loss alone does not indicate that the model is improving in terms of generalisation, as exemplified in Figure 4.5. These instabilities detected in the configurations were taken into account for the further analysis of the results.

In addition to the global loss, the individual loss components were analysed. As previously described, the global objective combines the brightness-aware photometric loss, the geometry-consistency loss and the smoothness loss. The training curves for these components are shown in Figures 4.7 to 4.11.

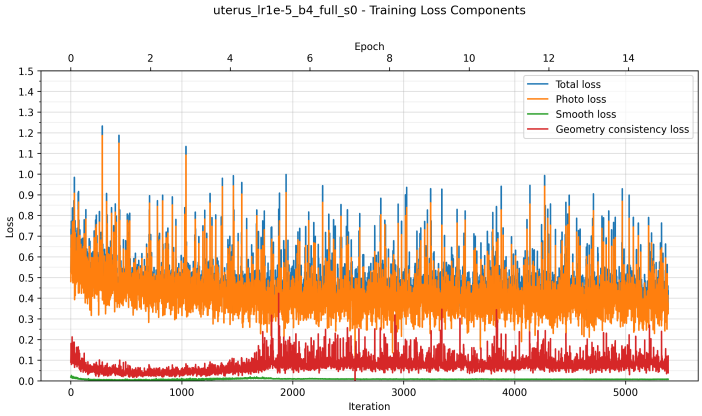


Figure 4.7: Training loss components for the configuration with learning rate 10^{-5} and batch size 4.

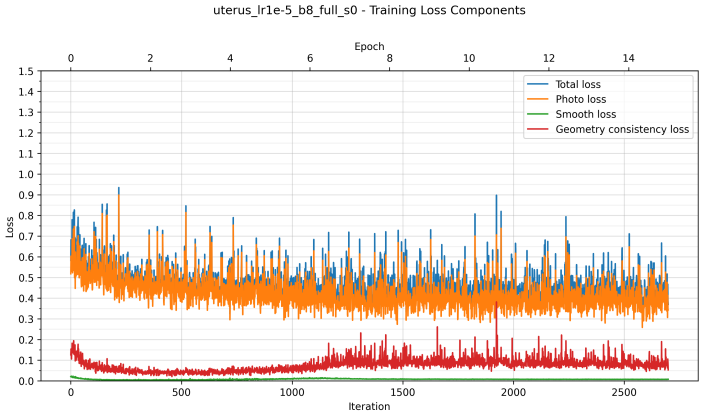


Figure 4.8: Training loss components for the configuration with learning rate 10^{-5} and batch size 8.

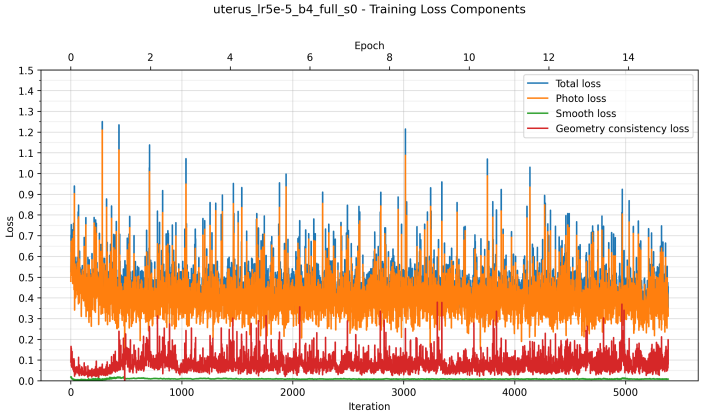


Figure 4.9: Training loss components for the configuration with learning rate 5×10^{-5} and batch size 4.

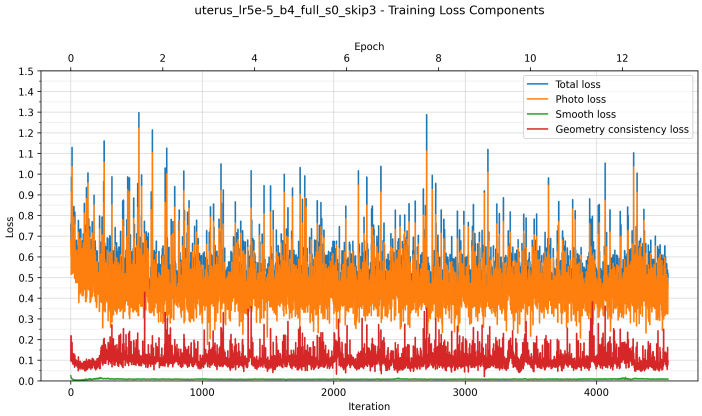


Figure 4.10: Training loss components for the configuration with learning rate 5×10^{-5}, batch size 4 and frame skipping 3.

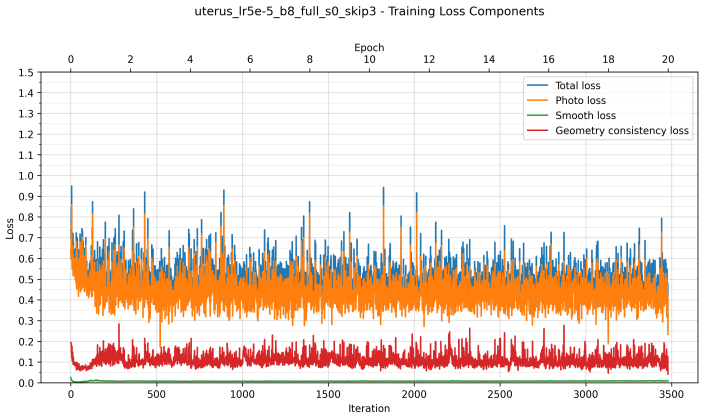


Figure 4.11: Training loss components for the configuration with learning rate 5×10^{-5}, batch size 8 and frame skipping 3.

Across the experiments, the photometric component represents the largest numerical contribution to the global loss, while the smoothness component remains considerably smaller and the geometry-consistency loss remains at an intermediate but much lower magnitude. For example, in Figure 4.7, the photometric loss has a mean value of 0.42 and a range of 1.19, whereas the geometry-consistency loss has a mean value of approximately 0.08. A similar behaviour is observed for the batch 8 configuration, for which the photometric loss has a mean value of 0.42 and the geometry-consistency loss a mean value of 0.07. For the 5×10^{-5} with a batch of 4 configuration, the corresponding mean values are 0.40 and 0.08, respectively.

The large variation of the photometric loss is also visible in the curves. For the three examples, the difference between the minimum and maximum photometric loss corresponds to approximately 151.5% to 300.6% of its mean value. In contrast, the smoothness loss remains close to the bottom of the plots and shows substantially smaller variations. The global training behaviour is therefore largely driven by the photometric component.

This dominance is expected from the formulation of the objective, as the three components are combined as a weighted sum rather than contributing equally by definition, and the presented behaviour is a reflection of the fact that this term is assigned a higher weight when the

predicted and interpolated depth maps are inconsistent at a given pixel due, for instance, to changes in illumination and by variations in the distance between the camera and the observed tissue, which are common in the dataset.

Endo-SfMLearner incorporates a geometry consistency constraint to make the depth predictions more compatible across consecutive frames. The geometry consistency loss compares depth predictions after accounting for the estimated relative motion and is intended to encourage consistency between consecutive depth maps. This term is presenting a low value for all the runs, indicating that this particular constraint is being satisfied during training. However, this does not imply that the final predictions are perfectly consistent in scale over an entire sequence.

The smoothness loss addresses a different limitation. Endoscopic images can contain regions with limited texture, where the photometric reconstruction error provides weaker information for determining local depth structure. The smoothness term is regularising the predicted depth and uses image gradients so that depth is encouraged to vary smoothly within homogeneous regions while allowing discontinuities near image edges. This provides an additional constraint where photometric information alone may be insufficient.

The component losses also provide some indication of differences between the training configurations and the component losses for the epoch considered as best model.

The range of the photometric loss varied considerably across the experiments, with the smallest observed range being 0.40 for the 10^{-5}, batch 8 and a frame skipping of 4 configuration and the largest being 1.44 for the 10^{-5}, batch 4, freeze 3, frame skipping of 3 configuration. The smoothness loss remained relatively similar across the different runs. For the geometry consistency loss, larger mean values were observed for some configurations, particularly those involving three frozen encoder layers. This suggests that these configurations experienced greater difficulty in satisfying the geometric consistency constraint during training, although the loss values alone are not sufficient to predict the final depth performance.

For the two configurations that obtained the best model from an epoch different than 1, the batch 8 one presented less irregularities for the photometric loss and similar geometry consistency behaviours.

Differences can also be observed in how quickly the loss components settle. Some configurations, such as the 10^{-5}, batch 8 and batch 4 runs without frame skipping, take longer to reach a stable pattern, whereas other configurations settle earlier. This provides an additional indication that the training dynamics are affected by the choice of parameters, namely the batch size, learning rate and frame sampling. For these configurations, the model selected as the best one corresponds to an early stage of training, corresponding to a higher photometric loss and a lower geometry consistency error than would be observed at later epochs. If the validation error observed during this initial stage were not taken into account, a later epoch could instead be selected, potentially corresponding to a model with a less favourable geometric consistency. Therefore, the observed behaviour does not allow the later epochs to be directly attributed to overfitting, as the relationship between the different loss components and the validation metric is not necessarily consistent throughout training. Further experiments

would be required to determine whether the observed behaviour is indeed related to overfitting or instead reflects differences in the training dynamics induced by the selected configurations.

Overall, the loss analysis was useful not only for deciding whether the selected number of epochs was sufficient, but also for understanding how the different training configurations behaved during optimisation. The training curves indicate that the model generally reaches a stable region within the tested training periods, while the higher learning rate produces less predictable validation behaviour. The component losses further show that the photometric term accounts for most of the numerical variation in the objective, while geometry consistency and smoothness provide additional constraints that are important for the particular difficulties of monocular endoscopic depth estimation. In addition, further training could potentially lead to stabilisation, allowing for the identification of a better set of training parameters.

The loss curves cannot, however, be directly compared with the results reported for Endo-SfMLearner, since the corresponding reuslts are not reported in that work. The comparison with the literature is therefore restricted to the subsequent purpose of the loss components rather than their numerical values.

4.1.2 Effect of Fine-tuning and Training Configuration

The effect of fine-tuning and the adopted training configurations were evaluated across the different experimental runs. The quantitative evaluation considered a set of the metrics reported in the literature, combining complementary measures used across different studies.

The reported values are presented as mean ± standard deviation (std), with the std representing the frame-to-frame dispersion of each evaluated metric across the sequence.

The RMSE, SqRel and MAE values are expressed in Unity units (UU), which can be interpreted as milimeters according to the ***Model scale*** analysis in Subsection 3.1.2. These evaluation metrics are performed after a scaling factor. Appendix I explores the need to use it after it had been mentioned in the previous chapters.

Analysis of Models with no Frame Skipping that Presented the Best Performances

First, Tables 4.1, 4.2 and 4.3 present the three configurations with no frame skipping and with all encoder layers unfrozen that obtained the best results for the AbsRel and RMSE metrics. The learning rate is represented with "LR" and the batch with "b".

Table 4.1: Depth prediction results for models trained with all layers unfrozen and without frame skipping for the metrics AbsRel, SqRel and RMSE.

LR	Batch	Skip	AbsRel ↓	SqRel ↓ (mm)	RMSE ↓ (mm)
1×10^{-5}	4	0	0.28 ± 0.51	2.43 ± 8.21	3.00 ± 2.02
1×10^{-5}	8	0	0.28 ± 0.50	2.40 ± 8.09	3.00 ± 2.00
5×10^{-5}	4	0	0.26 ± 0.47	2.10 ± 7.27	2.96 ± 1.84

Table 4.2: Depth prediction results for models trained with all layers unfrozen and without frame skipping for the metrics $\text{RMSE}_{\log}$, Log10, MAE.

LR	Batch	Skip	$\text{RMSE}_{\log} \downarrow$	Log10 $\downarrow$	MAE $\downarrow$ (mm)
1×10^{-5}	4	0	0.26 ± 0.22	0.08 ± 0.06	2.28 ± 1.48
1×10^{-5}	8	0	0.26 ± 0.22	0.08 ± 0.06	2.29 ± 1.47
5×10^{-5}	4	0	0.25 ± 0.21	0.08 ± 0.06	2.28 ± 1.35

Table 4.3: Depth prediction results for models trained with all layers unfrozen and without frame skipping for the metrics δ_1, δ_2, δ_3.

LR	Batch	Skip	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
1×10^{-5}	4	0	0.74 ± 0.21	0.89 ± 0.12	0.96 ± 0.08
1×10^{-5}	8	0	0.74 ± 0.21	0.89 ± 0.12	0.96 ± 0.08
5×10^{-5}	4	0	0.73 ± 0.21	0.91 ± 0.11	0.97 ± 0.07

Regarding the lower values of AbsRel, which evaluates the mean error relative to the ground-truth depth, these results suggest that, on average, the predicted depths are closer to the ground truth in relative terms. SqRel also evaluates the error relative to the ground-truth depth, but gives greater weight to larger errors due to the squared error term, so its values being higher and showing a comparatively large frame-to-frame dispersion may therefore indicate that, although the average relative error remains moderate, some pixels or frames contain substantially larger deviations from the ground truth.

These substantial frame-to-frame dispersion results, noticeable for the three configurations mainly for the SqRel and $\text{RMSE}_{\log}$, show that, even if the global mean values suggest a reasonably good overall performance, the predictions do not perform uniformly across the sequence, with some frames exhibiting considerably larger errors than others.

RMSE quantifies the absolute differences between predicted and ground-truth depth and is particularly sensitive to larger errors, as these are squared before averaging. The RMSE values obtained are approximately 3 mm according to the established metric scale. This value should be interpreted in relation to the spatial scale of the uterine structure, whose total dimensions range in the three directions approximately from 8 mm to 31 mm, as well as to the fact that each frame may present a different distance between the camera and the observed surface, frequently smaller than these overall dimensions. Therefore, an error of approximately 3 mm can represent a non-negligible proportion of the observed depth in some frames.

Nevertheless, the relevance of this error should also be considered in the context of the intended application of the depth estimation, the coverage mapping. Given the overall model dimensions of approximately $30 \times 31 \times 8$ mm, as established in the ***Model scale*** analysis in Subsection 3.1.2, a depth error of this magnitude does not necessarily imply a substantial change in whether a given region is classified as observed or unobserved. However, such errors may still reduce the spatial precision of the reconstructed surface. If errors of a similar magnitude are consistently present across multiple frames, they may accumulate or contribute

to misalignment between observations, potentially affecting the accuracy and reliability of the resulting coverage map.

In turn, MAE gives equal linear weight to all absolute errors and can be interpreted as a measure of the typical absolute deviation between the prediction and the ground truth. As expected from their definitions, the MAE values are lower than the corresponding RMSE values, since larger errors have a greater influence on RMSE.

$\text{RMSE}_{\log}$ and Log10 evaluate the differences in the logarithmic domain, making them more focused on relative depth discrepancies and less directly influenced by the absolute magnitude of the depth values. The obtained $\text{RMSE}_{\log}$ values, approximately 0.25-0.26, and Log10 values, approximately 0.08-0.09, indicate relatively limited logarithmic deviations between the predicted and ground-truth depths. The lower magnitude of Log10 is also expected from the different formulation and scale of this metric.

The δ metrics, in turn, quantify the proportion of predictions for which the ratio between predicted and ground-truth depth remains below progressively less restrictive thresholds. For the configuration with a learning rate of 1×10^{-5} and batch size 4, the obtained values indicate that approximately 73.8% of the predictions satisfy the strictest threshold, increasing to approximately 88.5% and 96.1% when the threshold becomes progressively more permissive. The corresponding configuration with batch size 8 obtained very similar values, also suggesting that changing the batch size produced only a limited effect on the overall depth prediction performance.

When comparing the two configurations with a learning rate of 10^{-5}, differing only in batch size, the results are very similar. The batch size of 8 provides slightly lower mean values for AbsRel, SqRel and RMSE, as well as slightly lower $\text{RMSE}_{\log}$ and MAE values, while the differences in the δ metrics are also very small. The different batch size is concluded to not provide a substantial improvement in the evaluated depth predictions under these experimental conditions. This result should not be interpreted as evidence that one batch size is generally preferable, since the effect of batch size may depend on the learning rate, dataset characteristics and training dynamics.

A noticeable difference is observed when comparing the two configurations with batch size 4 and different learning rates. Increasing the learning rate from 10^{-5} to 5×10^{-5} reduced the mean of AbsRel from 0.28 to 0.26, mean of SqRel from 2.43 mm to 2.10 mm and mean of RMSE from 3.00 mm to 2.96 mm. Improvements were also observed for $\text{RMSE}_{\log}$, Log10, MAE, δ_2 and δ_3, but δ_1 decreased. This difference illustrates the complementary nature of the metrics. While AbsRel, RMSE and the logarithmic measures describe the magnitude of the errors across the predictions, δ_1 measures specifically the proportion of pixels satisfying the strictest relative-error threshold. Consequently, a configuration may reduce the magnitude of errors overall while simultaneously changing the distribution of errors around the δ_1 threshold, resulting in a lower δ_1 value. The increase in δ_2 and δ_3 suggests, nevertheless, that a larger proportion of the predictions falls within the less restrictive relative-error thresholds.

Among these three configurations, the model trained with a learning rate of 5×10^{-5} and a batch size of 4 achieved the lowest mean AbsRel (0.26). It also obtained the lowest mean

values for the other main error-based metrics within this group, including an RMSE of 2.96 mm, although its δ_1 value was slightly lower than those of the configurations trained with a learning rate of 10^{-5}. Therefore, this configuration is highlighted based primarily on its mean AbsRel, rather than as the best-performing configuration across all evaluation metrics.

However, all three configurations correspond to models that were trained only until the first epoch. On the one hand, this makes the analysis of the configurations where the validation error started higher and the epoch for the best model was not the first particularly relevant, as well as the comparison with their resulting evaluation metrics. On the other hand, the favourable performance at such an early stage raises the possibility that part of the observed performance was already present in the original pre-trained model, rather than being exclusively attributable to the fine-tuning process.

Comparison Against Model Performance Beyond the First Epoch

First, the obtained metrics for the configurations where the best model was not on epoch 1 is presented in Tables 4.4 to 4.6.

Table 4.4: Depth prediction results for models that revealed a best model on an epoch different than the first for the metrci AbsRel, SqRel, RMSE.

LR	Batch	Skip	AbsRel ↓	SqRel ↓ (mm)	RMSE ↓ (mm)
5×10^{-5}	4	3	0.38 ± 0.64	4.04 ± 13.20	4.18 ± 2.53
5×10^{-5}	8	3	0.41 ± 0.72	5.07 ± 16.97	4.60 ± 2.57

Table 4.5: Depth prediction results for models that revealed a best model on an epoch different than the first for the metrics RMSE_{log}, Log10, MAE.

LR	Batch	Skip	RMSE_{log} ↓	Log10 ↓	MAE ↓ (mm)
5×10^{-5}	4	3	0.33 ± 0.24	0.11 ± 0.07	3.39 ± 1.97
5×10^{-5}	8	3	0.35 ± 0.24	0.12 ± 0.07	3.51 ± 1.94

Table 4.6: Depth prediction results for models that revealed a best model on an epoch different than the first for the metrics δ_1, δ_2, δ_3.

LR	Batch	Skip	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
5×10^{-5}	4	3	0.58 ± 0.23	0.85 ± 0.18	0.93 ± 0.11
5×10^{-5}	8	3	0.60 ± 0.19	0.83 ± 0.17	0.92 ± 0.11

Comparing these two configurations, the one with batch 4 presented the best performance for all metrics except for the δ_1. However, both present worse results than the previous three configurations. This seems to suggest that the adjustment enabled through fine-tuning did not improve the results and may even have had a detrimental effect. This could raise questions regarding the usefulness of the synthetic dataset for adapting the model to the target domain, although this interpretation should be considered cautiously, as the observed degradation

may also result from other aspects of the training configuration or dataset characteristics. Additionally, the frame-to-frame dispersion presented in SqRel is very significant for both, which motivated the analysis of the visual frames, presented in Figure 4.12.

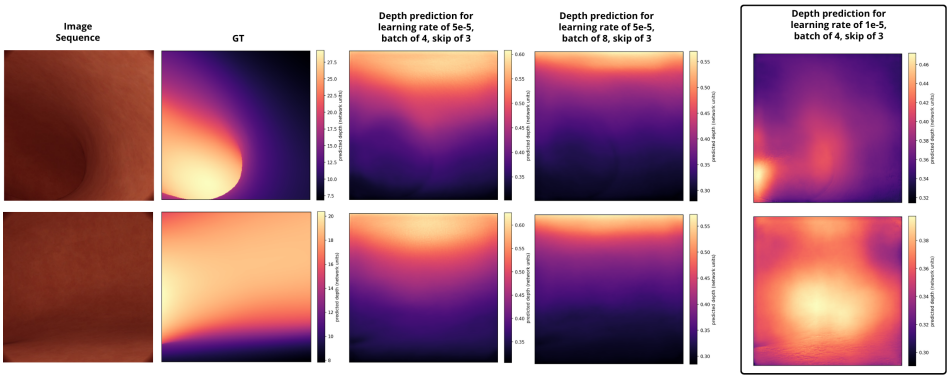


Figure 4.12: Analysis of the depth predictions for the models of learning rate 5×10^{-5} and skipping, with comparison to the GT and one configuration for learning rate of 10^{-5}.

These results suggest that the models trained with a learning rate of 5×10^{-5} and a frame skip of 3 did not adequately adapt their depth predictions to the specific spatial cues present in the images. An additional configuration, of learning rate of 10^{-5} is included to illustrate that the results were not always without the network to understand the geometry and cues. The 5×10^{-5} predictions exhibit a similar gradient across frames, with greater depths consistently assigned to the upper regions and smaller depths to the lower regions. This behaviour may indicate that the models relied on a simplified spatial pattern rather than capturing the image-specific geometric cues. In contrast, the configurations that achieved the best quantitative results produced depth maps that appeared more responsive to the spatial structure of the individual images.

The last observations motivated the evaluation of the original pre-trained model, also performed as part of this work's objectives. For this case, if a favourable depth prediction was already obtained at such an early stage of fine-tuning, part of the observed performance could potentially be attributed to the representation already having been learned, not needing subsequent adaptation to the hysteroscopic domain.

Comparison Against the Non-Fine-Tuned Model

The results for the original model are presented in Tables 4.7, 4.8 and 4.9.

Table 4.7: Depth prediction results for the original not finetuned synthetic hysteroscopy sequence for the metrics AbsRel, SqRel and RMSE.

Dataset	AbsRel ↓	SqRel ↓ (mm)	RMSE ↓ (mm)
Without fine-tuning	0.30 ± 0.48	2.37 ± 7.51	3.48 ± 1.82

Table 4.8: Depth prediction results for the original not finetuned synthetic hysteroscopy sequence for the metrics $RMSE_{log}$, Log10, MAE.

Dataset	$RMSE_{log} \downarrow$	Log10 $\downarrow$	MAE $\downarrow$ (mm)
Without fine-tuning	0.29 ± 0.20	0.10 ± 0.05	2.79 ± 1.44

Table 4.9: Depth prediction results for the original not finetuned synthetic hysteroscopy sequence for the metrics δ_1, δ_2, δ_3.

Dataset	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
Without fine-tuning	0.64 ± 0.19	0.89 ± 0.10	0.97 ± 0.07

When analysing the tables, the original model performance is better than for the configurations of learning rate of 5×10^{-5} and a frame skip of 3, revealing that the training on these configurations is not preferable when compared to the direct application of the Endo-SfMLearner. However, when compared with the corresponding configurations without frame skipping, its performance remains worse, as the configuration with the lowest mean AbsRel achieved an AbsRel of 0.2647 and an RMSE of 2.9557 mm, compared with 0.2984 and 3.4828 mm, respectively, for the original model. Therefore, the hypothesis that the configurations for which the first epoch was selected as the best model did not require further fine-tuning may not fully explain the observed behaviour.

Overall, the comparison with the original model suggests that some fine-tuned configurations achieved better depth estimation results on the synthetic hysteroscopic dataset. However, since the best model for these configurations was selected at the first epoch, it cannot be concluded that further fine-tuning would necessarily maintain or improve this advantage. Therefore, the observed improvement should be interpreted cautiously, as it may reflect the early stage of the training process rather than a consistent benefit of fine-tuning.

Analysis of the Skipping Effect

Following the initial comparison of learning rates and batch sizes, frame skipping was investigated to assess the effect of the temporal separation between frames used during training. The comparison of frame skipping for learning rate 5×10^{-5} was not included, as this would compare two different scenarios: one where the best model corresponded to a plateaud training loss, and one where it was still changing and converging. The results are presented in Tables 4.10 to 4.12.

Table 4.10: Effect of frame skipping on depth prediction performance for models trained with all layers unfrozen for the metrics AbsRel, SqRel and RMSE.

LR	Batch	Skip	AbsRel ↓	SqRel ↓ (mm)	RMSE ↓ (mm)
1×10^{-5}	4	0	0.28 ± 0.51	2.43 ± 8.21	3.00 ± 2.02
1×10^{-5}	4	3	0.28 ± 0.52	2.45 ± 8.53	2.95 ± 2.04
1×10^{-5}	8	0	0.28 ± 0.50	2.40 ± 8.09	3.00 ± 2.00
1×10^{-5}	8	3	0.28 ± 0.51	2.44 ± 8.48	2.95 ± 2.03
1×10^{-5}	8	4	0.28 ± 0.50	2.40 ± 8.17	2.98 ± 2.01

Table 4.11: Effect of frame skipping on depth prediction performance for models trained with all layers unfrozen for the metrics $\text{RMSE}_{\log}$, Log10, MAE.

LR	Batch	Skip	$\text{RMSE}_{\log}$ ↓	Log10 ↓	MAE ↓ (mm)
1×10^{-5}	4	0	0.26 ± 0.22	0.08 ± 0.06	2.28 ± 1.48
1×10^{-5}	4	3	0.26 ± 0.22	0.08 ± 0.06	2.22 ± 1.48
1×10^{-5}	8	0	0.26 ± 0.22	0.08 ± 0.06	2.29 ± 1.47
1×10^{-5}	8	3	0.26 ± 0.22	0.08 ± 0.06	2.22 ± 1.46
1×10^{-5}	8	4	0.26 ± 0.22	0.08 ± 0.06	2.24 ± 1.46

Table 4.12: Effect of frame skipping on depth prediction performance for models trained with all layers unfrozen for the metrics δ_1, δ_2, δ_3.

LR	Batch	Skip	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
1×10^{-5}	4	0	0.74 ± 0.21	0.89 ± 0.12	0.96 ± 0.08
1×10^{-5}	4	3	0.74 ± 0.21	0.89 ± 0.12	0.96 ± 0.08
1×10^{-5}	8	0	0.74 ± 0.21	0.89 ± 0.12	0.96 ± 0.08
1×10^{-5}	8	3	0.74 ± 0.21	0.89 ± 0.12	0.96 ± 0.08
1×10^{-5}	8	4	0.74 ± 0.21	0.89 ± 0.12	0.96 ± 0.07

Increasing the frame separation could, in principle, provide larger camera displacement and consequently stronger geometric cues for self-supervised learning. However, larger displacements can also increase the difference in appearance between the frames, making the photometric reconstruction task more difficult. Therefore, the selected frame-skipping values aimed to increase the available inter-frame motion while remaining within a range compatible with the observed camera trajectory.

For the configurations trained with a learning rate of 10^{-5} and batch size of 8, both frame skipping of 3 and 4 produced relatively small changes compared with the configuration without skipping. Frame skipping of 4 mostly performed better than the corresponding configuration without skipping. However, compared with frame skipping of 3, the differences were generally small and the ranking depended on the metric: the configuration with skipping of 4 was slightly better for SqRel and δ_3, while skipping of 3 obtained slightly better values for several of the remaining metrics, such as the MAE. These results do not indicate a clear advantage of

increasing the frame separation from 3 to 4.

Then, frame skipping of 3 was selected for further investigation with other parameter combinations. The differences observed for the learning rate of 10^{-5} and batch size of 8 and for the configuration with learning rate of 10^{-5} and batch size of 4 were similar. They both slightly improved results from the versions with no skipping and presented similar values for the obtained metrics.

Overall, the experiments indicate that moderate frame separation can be compatible with the training process, but increasing the separation does not necessarily improve the depth predictions and may become detrimental when combined with a configuration that is already sensitive to training instability.

Analysis of the Freezing Effect

The effect of freezing encoder layers was subsequently investigated using the same learning rate, batch size and frame-skipping configuration, the one with learning rate 5×10^{-5} aside, one of the two that best performed. This experiment was designed to isolate the effect of layer freezing while keeping the remaining training conditions fixed. It was not reproducted for other configurations mainly due to computational resources reaching the limit. Results are presented in Tables 4.13, 4.14 and 4.15.

Table 4.13: Effect of layer freezing on depth prediction performance for models trained with a learning rate of 1×10^{-5}, batch size 4, and frame skipping of 3.

Freeze State	LR	Batch	Skip	AbsRel ↓	SqRel ↓ (mm)	RMSE ↓ (mm)
No Freeze	1×10^{-5}	4	3	0.28 ± 0.52	2.45 ± 8.53	2.95 ± 2.04
Freeze 1	1×10^{-5}	4	3	0.28 ± 0.51	2.47 ± 8.42	3.04 ± 2.08
Freeze 2	1×10^{-5}	4	3	0.28 ± 0.52	2.51 ± 8.64	3.04 ± 2.10
Freeze 3	1×10^{-5}	4	3	0.28 ± 0.48	2.32 ± 7.63	3.16 ± 1.90

Table 4.14: Effect of layer freezing on depth prediction performance for models trained with a learning rate of 1×10^{-5}, batch size 4, and frame skipping of 3 for the metrics $\text{RMSE}_{\log}$, Log10, MAE.

Freeze State	LR	Batch	Skip	$\text{RMSE}_{\log}$ ↓	Log10 ↓	MAE ↓ (mm)
No Freeze	1×10^{-5}	4	3	0.26 ± 0.22	0.08 ± 0.06	2.22 ± 1.48
Freeze 1	1×10^{-5}	4	3	0.26 ± 0.22	0.08 ± 0.06	2.31 ± 1.53
Freeze 2	1×10^{-5}	4	3	0.26 ± 0.22	0.08 ± 0.06	2.29 ± 1.52
Freeze 3	1×10^{-5}	4	3	0.27 ± 0.21	0.09 ± 0.05	2.44 ± 1.39

Table 4.15: Effect of layer freezing on depth prediction performance for models trained with a learning rate of 1×10^{-5}, batch size 4, and frame skipping of 3.

Freeze State	LR	Batch	Skip	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$
No Freeze	1×10^{-5}	4	3	0.74 ± 0.21	0.89 ± 0.12	0.96 ± 0.08
Freeze 1	1×10^{-5}	4	3	0.73 ± 0.21	0.89 ± 0.12	0.96 ± 0.08
Freeze 2	1×10^{-5}	4	3	0.72 ± 0.22	0.90 ± 0.12	0.96 ± 0.08
Freeze 3	1×10^{-5}	4	3	0.72 ± 0.21	0.90 ± 0.11	0.96 ± 0.07

Freezing the first encoder layer produced a small deterioration across the evaluated metrics relative to the configuration without freezing. Freezing two layers did not improve the main error metrics either, while freezing three layers produced a slightly higher δ_3 but resulted in a further general deterioration, mainly seen in metrics such as MAE and δ_1.

The fact that the configuration without frozen layers obtained the lowest values for the main error metrics may indicate that restricting encoder adaptation was not advantageous under the tested conditions. This is, it suggests that some adaptation of the learned features was required to account for the characteristics of hysteroscopic images, even though it is not clear which specific features are being influenced by the encoder, only that the encoder stages change the spatial resolution and number of feature channels.

At the same time, the relatively small differences between some configurations indicate that encoder adaptation may not have been the dominant factor determining the final depth performance. From a practical perspective, this observation could become relevant if computational efficiency were prioritized and a small reduction in accuracy were acceptable. However, the present results do not allow this trade-off to be generalized, particularly because a dataset with a larger variety and more similarity could lead to different effects of freezing.

3D Visual Analysis

While quantitative metrics provide an objective assessment of the depth predictions, visual inspection offers a complementary perspective, also important, particularly by allowing differences in the spatial structure and appearance of the predicted depth maps to be examined.

A qualitative analysis of the predictions is presented in Figure 4.13, which presents a visualisation of the same real image side-by-side with the 3D representation of the corresponding GT and the results it obtained without fine-tuning and in each model configuration.

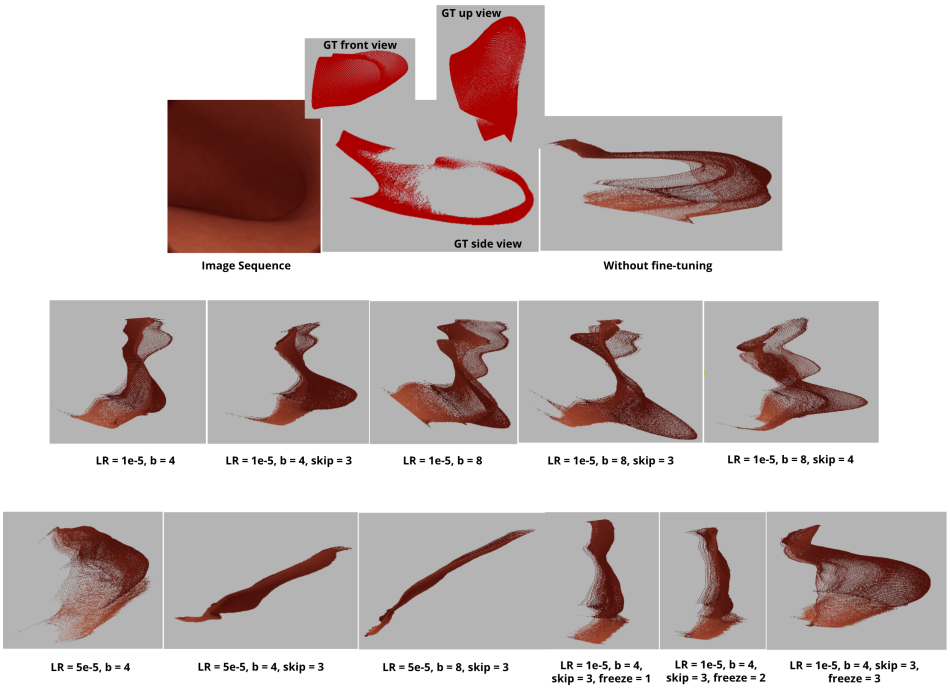


Figure 4.13: Analysis of the depth prediction through a 3D visualisation of depth for a single frame across all runs and including the GT.

Keeping the input fixed makes it possible to distinguish differences caused by the training configuration from differences caused by the anatomical content of the image.

Since the predicted depths are shown in the network output units and no scale alignment is applied at this stage, the depth ranges visible in these representations should not be interpreted as the actual metric ranges or as directly corresponding to the values used in the quantitative evaluation. For this reason, numerical depth ranges are not reported here. A more detailed analysis of the predicted depth ranges and their scale alignment is provided in Appendix I, where these values are examined to further clarify the scale differences between configurations.

Analysing the point clouds, the predictions seem, in general, to capture the region that corresponds to more depth and the remaining regions, as they are, for most of the configurations, in similar regions and being represented consistently.

However, it is noticeable that not all of them perceive the range of depth for that specific region. The depth predictions that appear to be closest to the ground truth are those from the non-fine-tuned model and the configuration with the first three encoder layers frozen. This is consistent with the latter configuration being the one that underwent the least modification of the encoder during fine-tuning, and therefore remaining closest to the original pre-trained model.

The configurations with a learning rate of 5×10^{-5} are the most notable exception, as they do not appear to correctly capture the region associated with greater depth. This behaviour is consistent with the instability previously observed for this configuration in the quantitative evaluation.

Overall, fine-tuning all parameters may have resulted in depth predictions with a narrower

range, suggesting that the training process may have adjusted the network towards producing flatter depth representations. However, it is important to refer that this is an illustrative comparison rather than as a general assessment. It only provides a more detailed qualitative and frame-level examination of the predictions.

It should be noted that the quantitative evaluation is performed after per-frame scale alignment, which compensates for the inherent scale ambiguity of monocular depth estimation. While this allows the spatial structure of the predictions to be compared under a common scale, it also removes differences that would exist in the absolute scale predicted by each model for individual frames. Consequently, a configuration may obtain favourable depth metrics after scaling while still exhibiting inconsistent scale across a sequence. This limitation is particularly relevant for the intended three-dimensional reconstruction, where a consistent scale between frames is required for reliable point-cloud fusion. Therefore, the quantitative depth metrics should be interpreted primarily as an assessment of per-frame depth accuracy after scale alignment, rather than as a measure of absolute or sequence-level scale consistency.

Depth Evaluation Metrics Final Remarks

An additional reflection is that no individual metric fully characterizes the model's performance, and that the different metrics should therefore be considered jointly.

To conclude, considering the results as a whole, the learning rate produced the most noticeable change among the parameters investigated, particularly when comparing 10^{-5} with 5×10^{-5} under the same batch size. The batch size and frame skipping produced differences, but minor, while encoder freezing resulted in a worsening of performance. No single configuration was unequivocally optimal and defined as the best. Furthermore, all these observations should be interpreted within the scope of the synthetic dataset used in this work.

4.1.3 Comparison with the Previous Endoscopic Depth Estimation

As referred, results of this work cannot be compared with the obtained ones for the Endo-SfMLearner. First, as no reference units are given, it is only known that those are the Unity units in their scale, while the ones in this work, also corresponding to Unity units, were studied in terms of the scale they represent, and it was concluded that one Unity unit meant mm. Second, their work only studied the colon, small intestine and stomach, and results for different domains should not be compared. The comparison with the Endo-SfMLearner work only supports the methodological motivation of this work rather than serving as a direct ranking of the two models.

In addition, as previously mentioned, other works that also adapt pre-trained weights could serve as a basis for comparison if they had been developed for the same anatomical structure. However, to the best of the authors' knowledge, no such work was identified.

4.2 Analysis of Generalisation to Real Images

The evaluation on real hysteroscopic images is not accompanied by ground-truth data, reference measurements or additional imaging modalities, such as ultrasound or other imaging techniques, that could provide a reference for the actual dimensions.

Therefore, the analysis of depth estimation for these images cannot be quantitative. These results cannot be used to determine whether a particular predicted depth value is correct. Their purpose is instead to assess whether the model produces spatially meaningful responses outside the synthetic domain.

The qualitative evaluation on real hysteroscopic images provides a complementary assessment to the quantitative evaluation performed on the synthetic dataset, while also allowing the generalisation of the model to real-world data to be investigated.

4.2.1 Analysis of the Obtained Real Results

Starting with the obtained results, Figure 4.14 shows how the real images were predicted by learning rate of 5×10^{-5} and frame skipping.

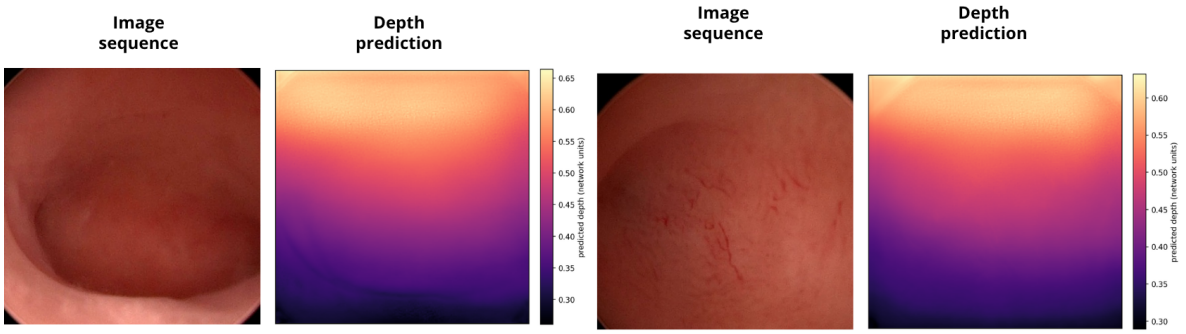


Figure 4.14: Depth prediction for two real images from the model with configuration learning rate 5×10^{-5}, batch 4 and skip 3.

Results showed that, for the real scenario, the models performed similarly to the synthetic one, as expected, not being able to understand the information on the frames.

Consequently, a further analysis was performed for one of the algorithms that achieved the best metric values, and presented in Figure 4.15.

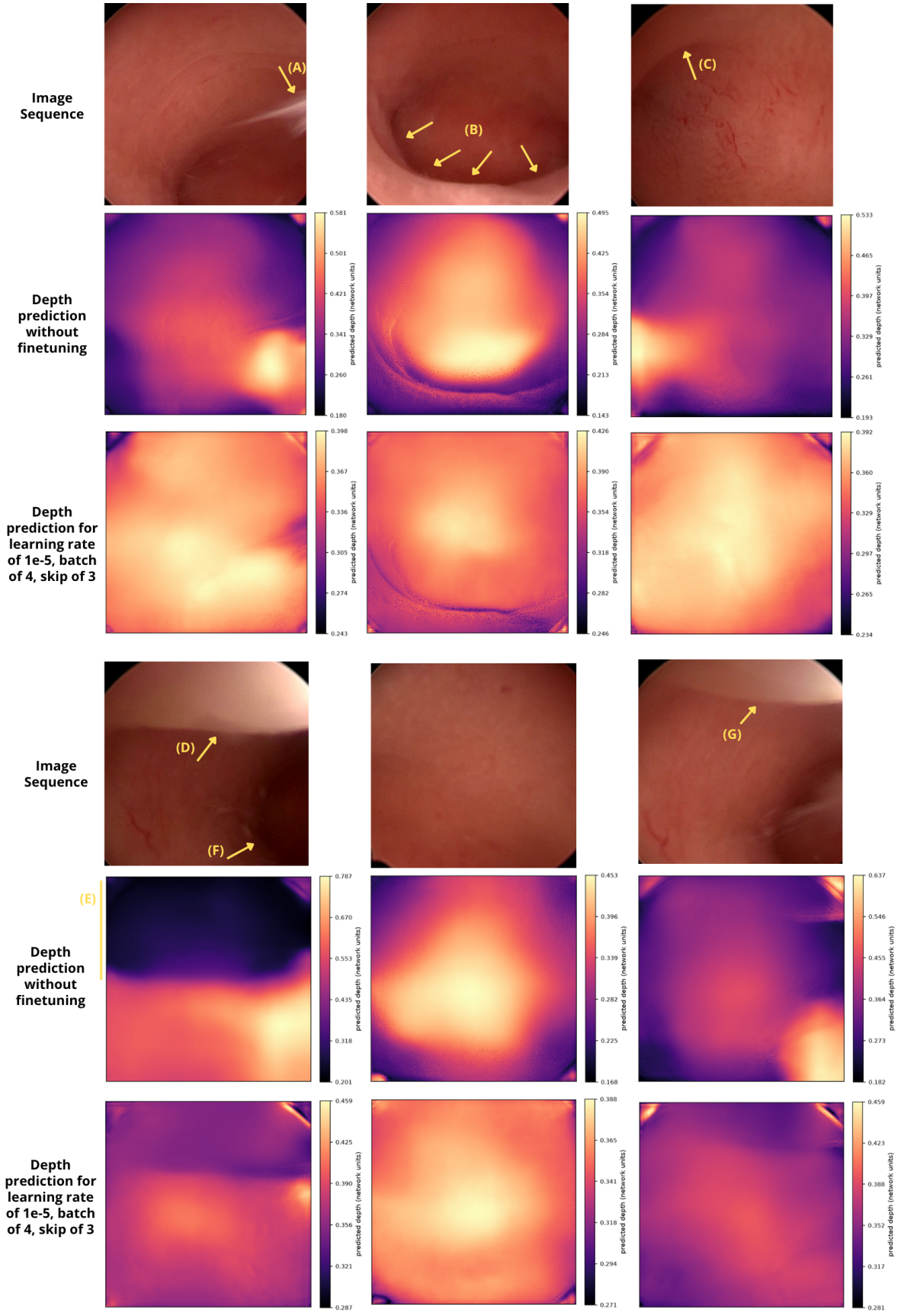


Figure 4.15: Analysis for real images.

Looking at the colour ranges used to represent how depth is perceived, these ranges vary between images. The fine-tuned prediction does not exceed 0.450 network units, whereas the prediction without fine-tuning reaches 0.580. It also does not estimate values below 0.230, whereas the other configuration reaches values as low as 0.143. This does not in itself imply anything since there is no expected range or known distinction between the expected maximum and minimum depths, and this change in depth distribution is expected in the absence of a common metric reference and prevents the predictions from being interpreted directly as absolute depth measurements. However, from the images, the prediction without fine-tuning appears to have a visual representation more consistent with the depth ordering inferred visually. This is, based on the analysis inferred visually, having that range of amplitude seems to improve the prediction.

Considering deeper versus closer regions, considered by inferring visually, the first image does not appear to be very well understood, since the model without fine-tuning assumes a smaller region of proximity than it should, while the fine-tuned model does not seem to identify exactly where this region is, although this region is represented more clearly than the rest. The artefact present in the middle, indicated by the letter A, is detected by the prediction after fine-tuning but not by the pre-trained model. Since this artefact is present in a real scenario, it would be important for this artefact to be detected, and this can therefore be considered a positive point for fine-tuning.

Continuing to consider the depths perceived by visual inference, the middle image appears to predict depth in a way consistent with what is inferred, although the model without fine-tuning identifies the deeper region better, located towards the back and below, indicated by the letter B. For the third image, the fine-tuned model predicts greater depth closer to the centre, which is not according to what can be inferred visually. Therefore, the model without fine-tuning appears to be more consistent with the visual inferred depth ordering, although it does not correctly perceive the depth throughout the entire arc, indicated by the letter C.

For the second set of images, the upper region closer to the camera, which is also affected by reflection, indicated by the letter D, appears to be consistent with what can be inferred visually and perceived as being closer. The model without fine-tuning estimates approximately 0.201, while the fine-tuned model estimates approximately 0.335. However, when analysing and inferring depth of the actual image, the model without fine-tuning appears to perceive a greater height for this region than the fine-tuned model, indicated by the letter E, while it also appears to be more consistent to where the image is visually inferred as deeper, indicated by the letter F. In the case of the third image in this row, with a similar region indicated by the letter G, the fine-tuned model is now the one that appears to be more consistent with what can be visually inferred, whereas the deeper region appears to be more consistently predicted to what can be visually inferred by the model without fine-tuning. For the central image, both models appear to be more consistent with what can be inferred visually for where the image is deeper, with the model without fine-tuning predicting 0.453 and the fine-tuned model predicting approximately 0.388.

Overall, the models appear to receive the visually inferred visual cues from the images

and infer depth from them, although they sometimes fail to do so correctly. The comparison between regions D and G does not show a consistent understanding of both regions, which may suggest some degree of randomness or simply that the model was not able to extract the relevant cues adequately. However, this analysis also appears to suggest that the model is not relying as strongly on colour, but rather on cues such as geometry, among others. Otherwise, in these cases, it would have consistently interpreted the lighter colour of D as indicating a closer region.

Therefore, depth prediction with the model without fine-tuning seems to be more consistent with the depth order visually inferred from the RBG images, while small cues, such as the artefact A, seem to be more consistent with the depth order visually inferred from the RGB images with the fine-tuned model, even though images containing such an artefact were not used during training.

Furthermore, the occurrence of these errors does not imply specifically that they are related to the change from synthetic dataset to real domain, but rather to errors inherent to the models' results. Therefore, these results do not appear to indicate a lack of generalisation simply because the input is no longer generated by the synthetic environment. Instead, they suggest that the model has learned some meaningful information that can be transferred to real hysteroscopic images.

Since median scaling is not applicable to these images, as it requires the ratio between the median ground-truth depth and the median predicted depth, a possible alternative analysis was performed, as presented in Table 4.16. The first row reports the ground-truth relief across the two synthetic test scenes, scene_15 and scene_17, included only as an order-of-magnitude reference for a plausible uterine surface.

Table 4.16: Scale-invariant shape descriptors of the predicted depth maps on the real hysteroscopy frames, given as the range observed across the analysed frames.

Configuration	**max/min**	**(max−min)/median**	**std/median**
Synthetic ground truth (used as reference)	2.5–4.4	0.72–0.95	0.18–0.28
Without fine-tuning	2.7–3.9	0.82–1.31	0.17–0.40
LR = 5e-5, b = 4	1.8–2.5	0.50–0.69	0.09–0.13
LR = 1e-5, b = 4, skip = 3	1.4–1.7	0.42–0.48	0.04–0.05

Comparing with the synthetic reference, where the std/median value is 0.18–0.28, the fine-tuned model returns an almost flat depth field the std/median of 0.04-0.05 against 0.17-0.40 for the pre-trained model, and a max/min ratio below 1.7 against 2.7-3.9. This indicates that fine-tuning did not merely rescale the output but collapsed its dynamic range, and suggests that this behaviour is a property of the fine-tuned model.

Thus, when considering the purpose meant for the depth estimation, the coverage mapping, the version with no finetuning seems to be more adequate.

4.2.2 Analysis of the Effect of Image Cropping

An additional experiment was performed by comparing the prediction obtained from a complete real frame with the prediction obtained from a cropped region of the same frame, for the same configuration explored a for the full images and the pre-trained model with no fine-tuning.

This comparison was intended to determine whether the estimated depth of a local anatomical structure remained qualitatively consistent when the surrounding image context was removed. This was inspired by what was mentioned in the literature review in ***Spatial and temporal consistency***, in Subsection 2.4.4. This does not evaluate metric accuracy, it just provides an additional qualitative assessment of the stability of the learned depth representation.

This analysis is relevant because monocular depth estimation can depend on contextual visual cues. If the same anatomical region receives substantially different predictions when isolated from its original context, this would indicate that the estimation is strongly dependent on the surrounding image. Conversely, similar relative depth patterns would suggest that the model is able to preserve local spatial information, inferring depth locally, independently of the complete field of view.

Visual examples are presented in Figure 4.16, while a metric analysis is presented in Table 4.17, where only one run is analysed. All metrics, except for AbsRel, were obtained after applying median scaling using as reference the complete square image.

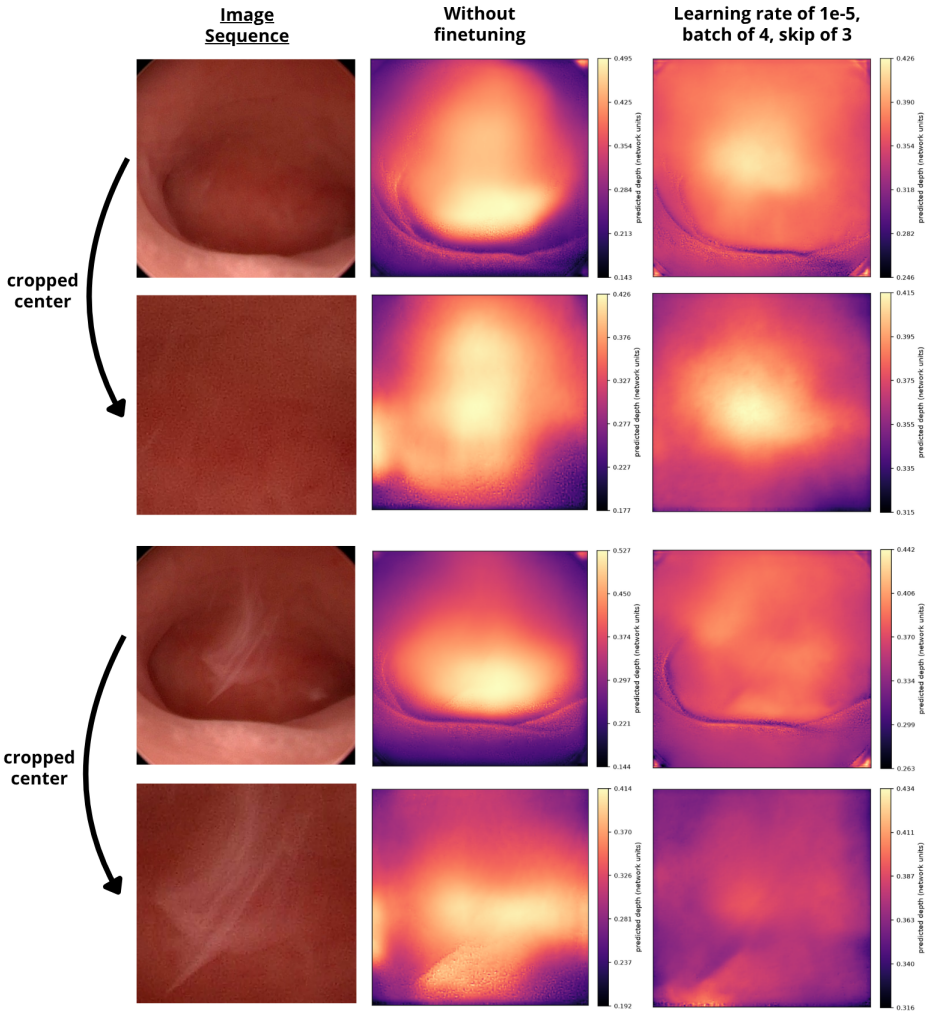


Figure 4.16: Depth prediction for real images were the full squared image is compared to the center-cropped version of it.

Table 4.17: Comparison of the full real image versus the crop agreement, per sequence. Metrics with the excetion of AbsRel use per-frame median scaling having the image that was cropped as reference.

Model	**Sequence**	**Frames**	**AbsRel**	**RMSE**	δ_1	**Wide/tight scale factor**
Pre-trained	scene 19	167	0.08	0.04	0.94	1.15
Pre-trained	scene 20	95	0.09	0.05	0.92	1.17
Fine-tuned	scene 19	167	0.02	0.01	1.00	1.03
Fine-tuned	scene 20	117	0.02	0.01	1.00	1.04

For the pre-trained model, the two predictions of that shared region differ, after per-frame median scaling, by a mean relative error of 0.08 on scene 19 and 0.09 on scene 20 ($\delta_1 = 0.94$ and 0.92). The ratio of medians between the two framings is 1.15 and 1.17. This means that the same surface is predicted, on average, about 15 to 17% further away when it is seen inside the full image, with the full endoscope aperture and surrounding wall, than when it fills the tight crop. This analysis indicates that the pre-trained model is measurably framing-dependent, which could be due to some domain sensitivity.

For the fine-tuned model, the two predictions of the shared region are almost identical: AbsRel 0.024 and 0.023 with median scaling ($\delta_1 = 1.00$ and 1.00, i.e. essentially every pixel within a factor of 1.25), and a median ratio of 1.03 and 1.04. The residual structural disagreement is about three to four times smaller, only a 3.5% residual offset attributable to framing, and the amount by which the median ratio departs from one is four to five times smaller than for the pre-trained model. Thus, these metrics suggest that there is not a frame dependency and, if so, fine-tuning would have changed the depth prediction in a way that matters.

Thus, the behaviour is consistent across the two recordings, where, for the fine-tuned model, scene 19 and scene 20 agree in AbsRel and 0.01 in the scale factor, and the pre-trained model shows the same direction in both sequences, with more variation between them (0.08 against 0.09, 1.15 against 1.17). However, for scene 20, only 95 of the 117 frames had a pre-trained wide-crop prediction, so that row is computed over fewer frames than the fine-tuned one.

It is important to note that framing invariance and scale consistency are distinct properties, as a model can predict the same region consistently across framings and still let its scale drift as the camera moves through a sequence.

In addition, the experiment measures agreement between two framings, cropped and not cropped, not correctness, although the larger disagreement of the pre-trained model shows they are at least not trivially identical, both predictions could share the same error.

Results suggest that the fine-tuned network appears to have acquired a local depth-from-appearance mapping and not a global idea, which would justify what was analysed from the images that the regions with more depth were not understood that well. This could be benefitial for this work, as a reconstruction pipeline would need that kind of behaviour.

4.2.3 Analysis for a More Diverse Real Images

The used model aims to learn to infer geometry rather than to recognise pathologies or lesions. As they are also not included in the training data, it cannot be stated that the method would perform equally well in a more diverse dataset and real images. To provide some insight into this aspect, polyps were selected for reconstruction after training, with the results presented in Figure 4.17. This extends the evaluation to a more diverse set of real hysteroscopic images.

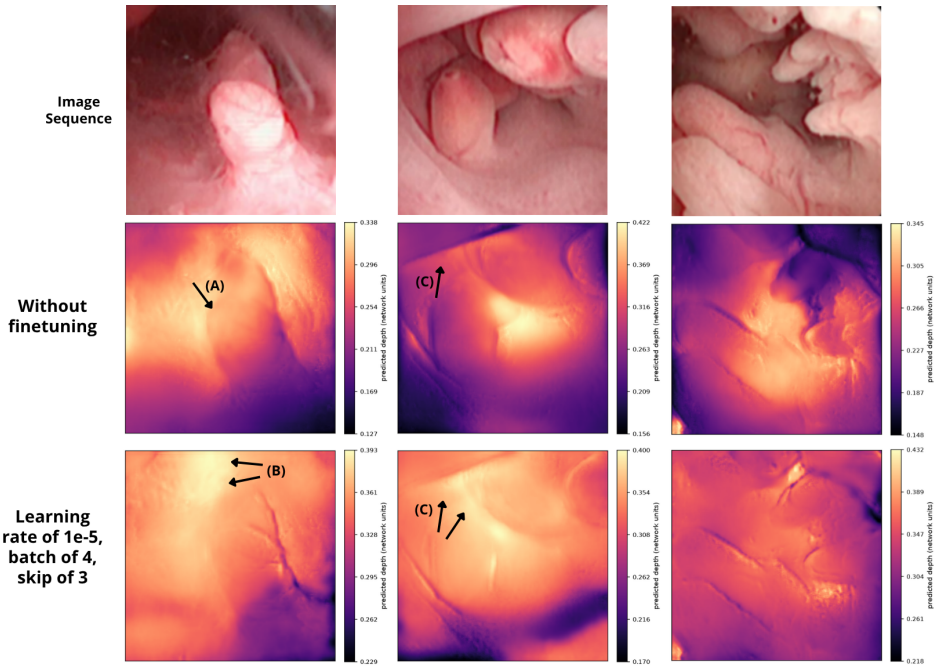


Figure 4.17: Depth prediction for real images with polyps.

The estimated depths remained within similar ranges. For the first image, the model without fine-tuning appears to be more consistent with the depth order inferred visually for the proximity (letter A), whereas the fine-tuned model appears to be more consistent with the depth order inferred visually for when regions are farther away (letter B). Both models perceive the presence of the two polyps, although subtly, as concluded from a visual inferring, without assuming that their entire structure is close and instead mainly identifying their bases as visually inferred. In the second image, the fine-tuned model appears to perceive the path of the camera (letter C) more consistently with the depth order inferred visually, whereas the other configuration appears to define the location of the polyps more consistently with the depth order inferred visually. In the third image, both models capture the contours of the polyps, which can be inferred visually, more consistently with the visual inferred RGB image for the model without fine-tuning. Moreover, the model without fine-tuning seems to be more consistent with the depth ordering that can be visually inferred from the RGB images in terms of the perception of depth differences, but perceives one of the polyps as being closer when a visual analysis suggests it is not.

This analysis ultimately provides some insight into the localisation and perception of polyps in the reconstruction, although this is not the focus of the work, as it is concerned with coverage mapping. Such information could be useful if the reconstruction were subsequently used to support lesion detection. In the present context, however, the results aim to explore adaption even to real images that differ more substantially from the training data, providing a greater degree of diversity than that represented in the training dataset. Overall, considering that the analysis was qualitative and based on the visual inference from the RGB images, the predictions suggest that the model might be able to retain some of the learned spatial cues when faced

with these more diverse real images, although this transfer is only partially consistent across the analysed cases.

4.2.4 Summary

This subsection provides an indication of how the adapted model behaves outside the synthetic domain. The results, dependent on what can be visually inferred and the consistent degree of the depth predictions that is also dependent on what can be visually inferred, suggest that the adapted model is able to produce depth predictions that reflect some spatial cues present in real hysteroscopic images, indicating a degree of transfer from the synthetic environment to the real domain. Nevertheless, the models that achieved the best performance were obtained at a stage when the training loss curves had not yet stabilised. In addition, the variation observed between images and configurations also highlight the difficulty of assessing real-world performance in the absence of a corresponding depth reference. A visually plausible depth map may still contain errors in absolute scale or local geometry, which cannot be quantified from the available real images. Differences between the synthetic and real results are therefore expected in a transfer-learning setting and should be interpreted as an indication of the challenges involved in transferring the learned depth representations to real hysteroscopic images, rather than as a direct measure of model performance in the real domain.

4.3 Fused Point-Cloud Analysis

The depth predictions were further evaluated at the sequence level through point-cloud fusion. This analysis was performed for scenes 15 and 17, for which ground-truth depth and camera poses were available. This was performed for only one of the models, the one with configuration of learning rate 10^{-5}, batch of 4 and skip of 3, as it presented one of the best performances. This provides an evaluation of the reconstructed geometry that cannot be obtained from the per-frame depth metrics alone. Results are presented in Table 4.18.

Table 4.18: Point-cloud reconstruction metrics for scenes 15 and 17, without fine-tuning and with fine-tuning (LR = 1e-5, b = 4, skip = 3).

Scene	**Model**	**Accuracy RMSE ↓ (mm)**	**Completeness RMSE ↓ (mm)**	**Chamfer (half-sum) ↓ (mm)**	**Hausdorff ↓ (mm)**
15	Without fine-tuning	2.14	0.67	1.02	6.78
15	Fine-tuned	2.17	1.81	1.53	6.87
17	Without fine-tuning	2.01	2.15	1.50	8.63
17	Fine-tuned	1.55	3.71	1.82	11.05

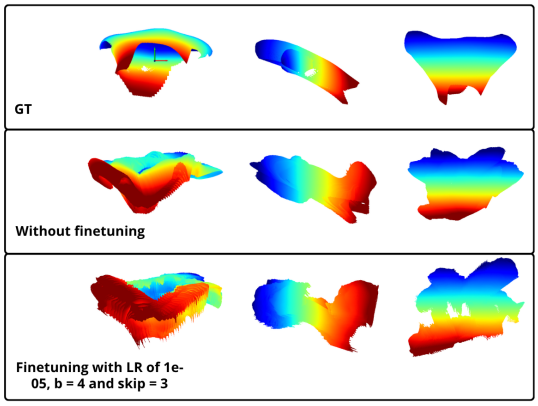
These results illustrate the different information provided by the two directional errors. Accuracy describes how closely the predicted points lie to the reference surface, whereas com-

pleteness measures how well the reference surface is represented by the prediction, as a reconstruction can contain a relatively accurate subset of the surface while still failing to represent other regions.

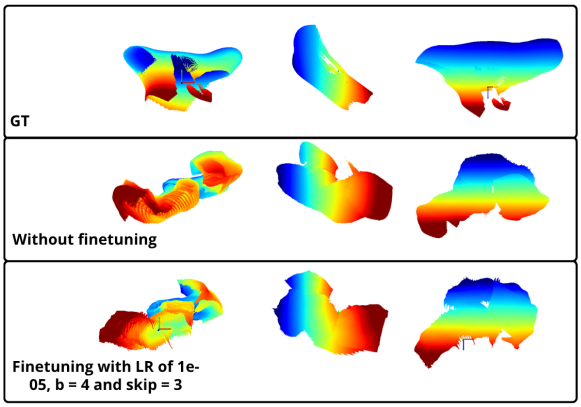
With one Unity unit taken as one millimetre, the reconstructed surface lies within a few millimetres of the ground truth, with an accuracy RMSE of around 1.5 to 2.2 mm and a Chamfer half-sum of around 1.0 to 1.8 mm. As they are obtained for a cavity around 30 mm across, the predicted local geometry was concluded to be reasonable.

Regarding the fine-tuning, it does not seem to improve the reconstruction and worsens its completeness when compared to the version before the fine-tuning. For scene_15, the accuracy RMSE is very similar, but higher, while the completeness nearly triples. In scene_17, the fine-tuned points that exist are tighter, but the surface is less complete. In addition, the Chamfer half-sum, which balances accuracy and completeness, rises with fine-tuning in both scenes, and the Hausdorff distance, set by the single worst point, is essentially unchanged for scene 15 but markedly larger for scene 17. However, the better completeness of the model without fine-tuning does not imply that its reconstruction is more similar to the GT.

It order to explore how the pointclouds are represented and overlapped with GT, visual versions of the fused point clouds are presented in Figures 4.18 and 4.19.

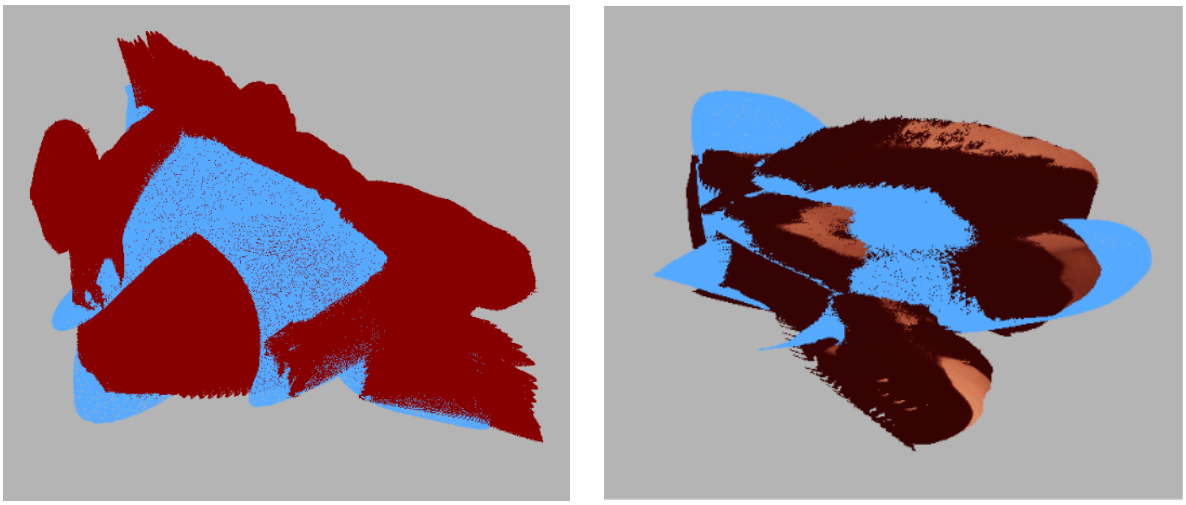


(a) Scene_15



(b) Scene_17

Figure 4.18: Fused ground-truth and predicted point clouds for scene_15 (a) and scene_17 (b).



(a) GT and without fine-tuning prediction (b) GT and fine-tuned prediction

Figure 4.19: Examples of the spatial overlap between (a) the ground-truth and fine-tuned predicted point clouds and (b) the ground-truth and without fine-tuned predicted point clouds for scene 15.

The predicted point clouds do not get a very similar structure to the uterus cavity and do not reproduce the reference surfaces with the same coherence. For instance, analysing Figure 4.18, for scene_15 the first view may appear to correspond to the correct format, but the left closer region shown in red is upwards and, while the third view shows a region with two paths in the darkest blue, those are not compatible with the ostia openings, at the same time that there is a third and fourth path to the left and right in the light blue region. For the other scene, the region that is more similar is the second, the side view, but the red region is still to the left. For both scenes, local deviations and a greater dispersion of the points can be observed.

When comparing the predictions after and before fine-tuning shown in Figure 4.18, scene_15 does not show a relevant difference and scene_17 present a larger, more spread out cloud for the before. These results also ilustrate that what was mentioned about the better completeness of the model without fine-tuning does not actually mean a more approximate result.

In addition to not reproducing the true surface, the predicted clouds show a spread that follows the path of the camera rather than the shape of the cavity, as illustrated by Figures 4.20 and 4.21. The clouds elongate in the direction the camera travelled, extending beyond the region occupied by the ground-truth cloud. The trajectory determines where each observation is placed, but what is happening is a shaping of the reconstructed geometry.

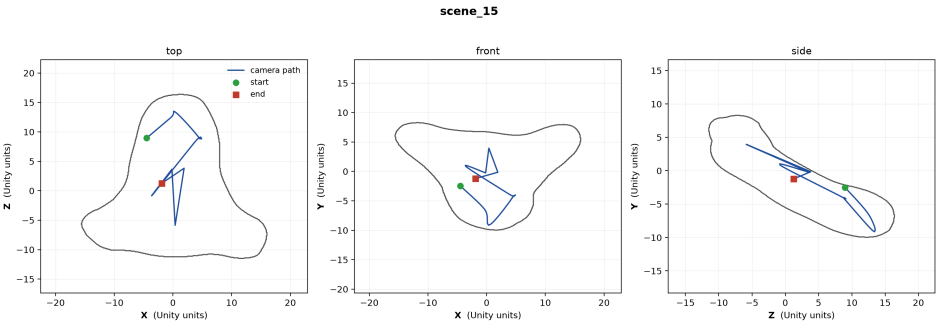


Figure 4.20: Trajectory of the scene under analysis, scene_15.

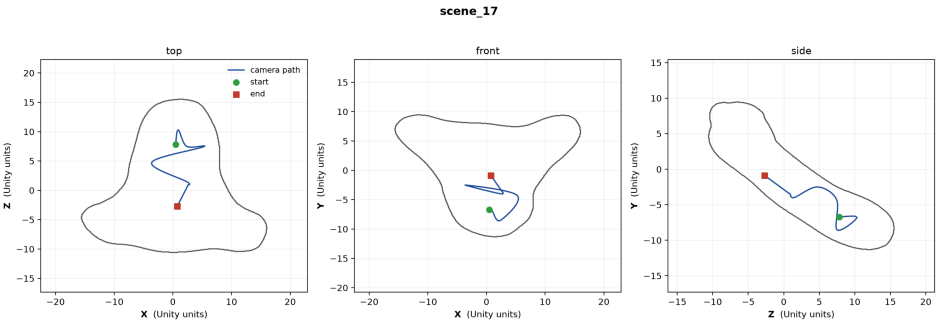


Figure 4.21: Trajectory of the scene under analysis, scene_17.

The possibility that the camera poses were recorded or transformed incorrectly, for instance, through a wrong quaternion ordering or coordinate-system handedness, was considered. However, the ground-truth depth maps fused into a coherent surface under exactly the same trajectories, and it was thought that an incorrect pose ordering or convention would have affected those frames as well. These ground-truth point clouds end up acting as a control for this analysis.

Therefore, the discrepancies observed in the predicted reconstructions are suspected to be either associated with the predicted depth or with the different step in the procedure, the global scale factor, needed to obtain the reconstruction of the complete sequence.

As for the fused point clouds no individual scaling is performed, a different behaviour than with the analysis for the per frame is expected, as when it was that scenario, the evaluation was relative. Applying a different factor to every frame for the fused point clouds would place the individual predictions at different metric scales and prevent surfaces observed from different viewpoints from coinciding.

The variation of these factors across a sequence provides a measure of the scale consistency of the predictions. The results are presented in Table 4.19.

Table 4.19: Scale consistency of the predicted depth, without and with fine-tuning (LR = 1×10^{-5}, batch size = 4, skip = 3), for the synthetic 15 and 17 scenes.

	Scene 15		**Scene 17**	
Metric	**No Fine-tuning**	**Fine-tuning**	**No Fine-tuning**	**Fine-tuning**
AbsRel, per-frame scaling	0.19	0.16	0.44	0.43
AbsRel, single global scaling	0.32	0.35	0.48	0.46
Drift penalty (global − per-frame)	0.13	0.19	0.04	0.04
Per-frame scale dispersion (std/mean)	30.0%	31.2%	26.8%	26.1%
Per-frame scale factor range	×8.3	×8.7	×3.0	×2.6
Fused-cloud collapse	×1.3	×1.4	×1.4	×1.5
Fused-cloud collapse, GT	×11.7		×11.1	

Without fine-tuning, the relative dispersion of the scale factors was 30.0% for scene 15 and 26.8% for scene 17. The corresponding ranges between the minimum and maximum scale factors were factors of 8.3 and 3.0, respectively. Fine-tuning did not substantially reduce this variation, as the dispersion changed to 31.2% and 26.1%, while the scale-factor ranges became factors of 8.7 and 2.6. Thus, the predictions remained subject to considerable frame-to-frame scale variation after adaptation.

Furthermore, analysing values from the independent per-frame scaling and with a single global scale, the AbsRel of scene 15 increased from 0.19 to 0.32 without fine-tuning and from 0.161 to 0.349 after fine-tuning. For scene 17, the corresponding changes were from 0.44 to 0.48 and from 0.43 to 0.46. These differences correspond to the introduced additional error. This effect is particularly clear in scene 15, where fine-tuning improves the per-frame AbsRel but produces a larger penalty when a single scale is imposed on the sequence.

The fused-cloud collapse values provide a geometric interpretation of the same phenomenon. When the ground-truth depth maps are fused, the resulting cloud collapses by factors of ×11.7 for scene 15 and ×11.1 for scene 17, reflecting the agreement between observations of the same surface from different viewpoints. The predicted clouds, in contrast, show collapse factors of only ×1.3 and ×1.4 without fine-tuning, changing to ×1.4 and ×1.5 after fine-tuning. This means that the predicted frames remain more spatially dispersed instead of forming a thin coherent surface, as the previous analysis already suggested.

All of these indicate that the advantage of fine-tuning seen in the depth estimation metrics

does not carry over to the fused reconstruction. Whether the pre-trained version is in fact preferable for this application cannot be concluded, as only two scenes are being analysed. To conclude something similar, more studies need to be done, such as examining more sequences and training configurations.

This result is relevant to the geometry-consistency component of the original Endo-SfMLearner formulation, which is intended to constrain the geometric relationship between consecutive predictions. The persistence of scale variation after fine-tuning does not contradict the purpose of this component. A consistency term constrains the predictions during training, but does not guarantee identical scale across all frames, particularly when the model is transferred to a different anatomical domain.

Overall, the fused point cloud analysis shows that the limitations of this work cannot be explained solely by the accuracy of individual depth predictions, as predictions from different frames must be also geometrically compatible when transformed into the same coordinate system. In this work, the observed frame-to-frame scale variation appears to be an important factor limiting this compatibility, and a model intended for three-dimensional hysteroscopic reconstruction must maintain a consistent geometric scale throughout the sequence and a future coverage system would be extremely influenced by this.

4.4 Time and Computational Performance for Clinical Applicability

To explore time and computational performance in order to infer clinical applicability, the time requirements and computational costs of the stages that would also be performed in a clinical application were analysed. The only exception was the estimation of the pose, which was not included in this analysis since it was not part of this work.

The analysed stages were the preprocessing stage, depth prediction through `test_disp.py` and 3D point-cloud fusion.

The memory usage results for these stages are presented in Table 4.20. Measurements were obtained using a NVIDIA Tesla T4 GPU.

Table 4.20: Memory usage of the different stages of the depth prediction pipeline. The total corresponds to the sum of the peak resident memory values of the individual stages, not representing the peak memory usage of the complete pipeline when the stages are executed sequentially.

Memory metric	Preprocessing	Depth estimation	Fused point clouds	Total
Peak resident memory (MB)	251	1629	390	2270
Growth over baseline (MB)	15	615	136	766
Peak GPU allocated (MB)	–	393	–	–
Peak GPU reserved (MB)	–	428	–	–

The peak memory usage of the complete pipeline is determined by the depth estimation stage, which reached 1629 MB of resident memory, corresponding to an increase of 615 MB over the baseline. Although the depth estimation stage therefore represents the main memory

requirement of the pipeline, the measured GPU allocation was considerably lower, with a peak of 393 MB and a peak reserved memory of 428 MB. However, these values should not be interpreted as the minimum GPU memory required to run the system, since additional memory is required by the operating system, the GPU runtime, and other processes. Nevertheless, the measured GPU memory consumption indicates that GPU memory is unlikely to constitute a major hardware constraint for the proposed approach.

The memory requirement of the depth estimation stage is also dependent on the selected batch size and input resolution. A trade-off between memory consumption and computational requirements may be acceptable without substantially compromising prediction performance, which is relevant for a future clinical implementation. For the resolution, it could lower the memory footprint at the expense of prediction detail or throughput. In this work, the considered size of the images was 384×384 pixels, which were then presented to the algoritm in JPG format.

In comparison, preprocessing and point-cloud fusion reached peak resident memory values of 251 MB and 390 MB, respectively. The relatively low memory requirement of preprocessing is consistent with the expected results and the operations performed at this stage. Point-cloud fusion requires more resident memory due to the accumulated point-cloud data.

It should also be noted that the total presented in Table 4.20 is not representative of the peak memory requirement of the complete pipeline. It corresponds to the sum of the individual peak resident memory values and provides an upper-bound cumulative figure rather than a measurement of simultaneous memory usage. Since the stages are executed sequentially, their individual peaks do not necessarily occur at the same time. The actual peak memory requirement of the sequential pipeline is consequently determined by the stage with the highest instantaneous resident memory usage, namely depth estimation.

In turn, time requirements are explored. Figure 4.22 presents the time distribution of each relevant task for clinical applicability within the preprocessing stage. The equivalent time distribution for each task for the depth estimation is presented in Figures 4.23 to 4.24, exploring this aspect across different runs. Finally, Figure 4.25 explores the time required for the fusion of the point clouds.

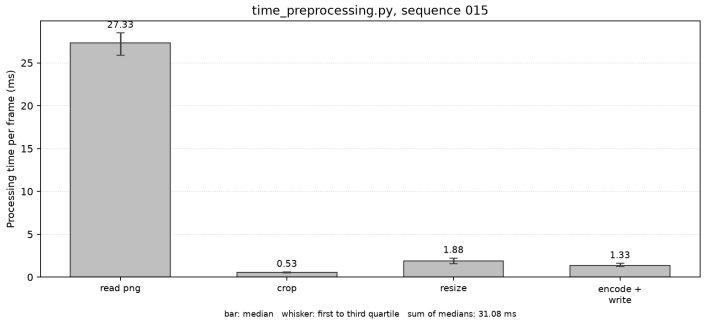


Figure 4.22: Preprocessing time per step.

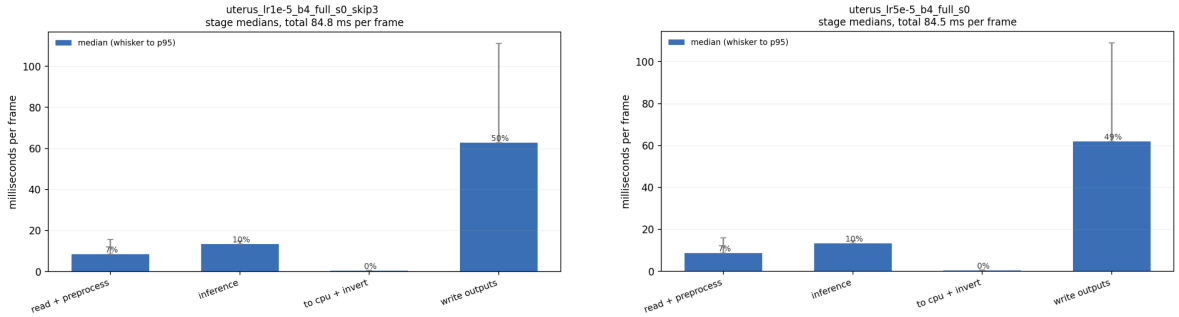


Figure 4.23: Per-frame cost by step for the configuration with learning rate 10^{-5}, batch size 4 and frame skipping of 3, and the configuration with learning rate 5×10^{-5}, batch size 4 and no frame skipping. The whisker on each bar reaches the ninety-fifth percentile and the percentage above it is that step's share of the total.

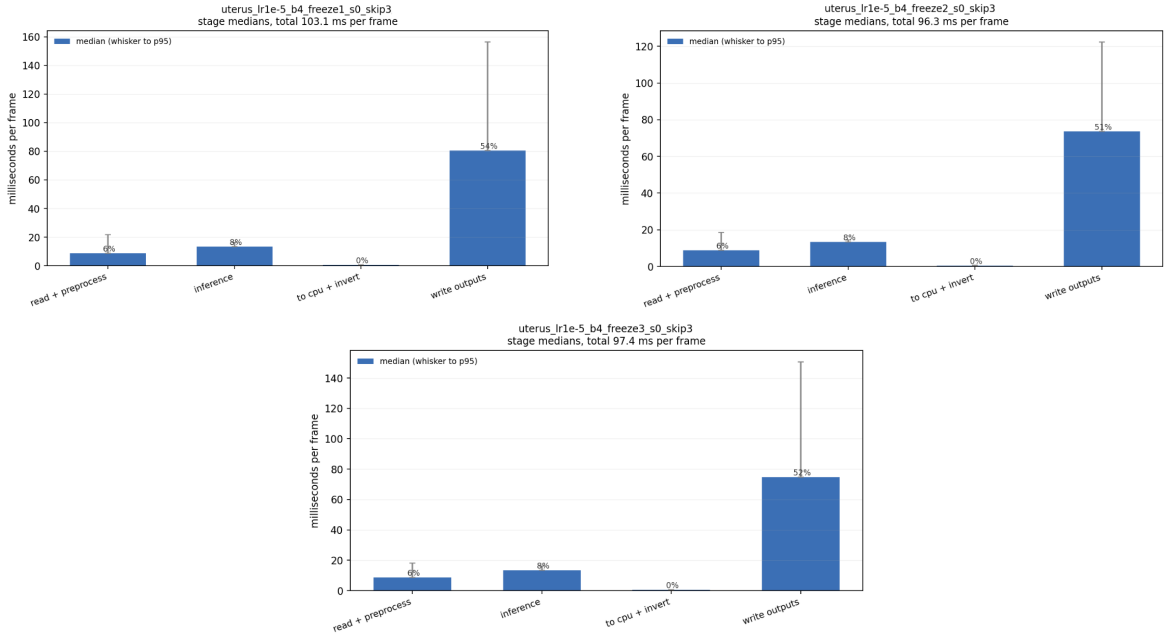


Figure 4.24: Per-frame cost by step for the configurations with learning rate 10^{-5}, batch size 4 and freezing of the first, second and third encoder layers, respectively, both using frame skipping of 3. The whisker on each bar reaches the ninety-fifth percentile and the percentage above it is that step's share of the total.

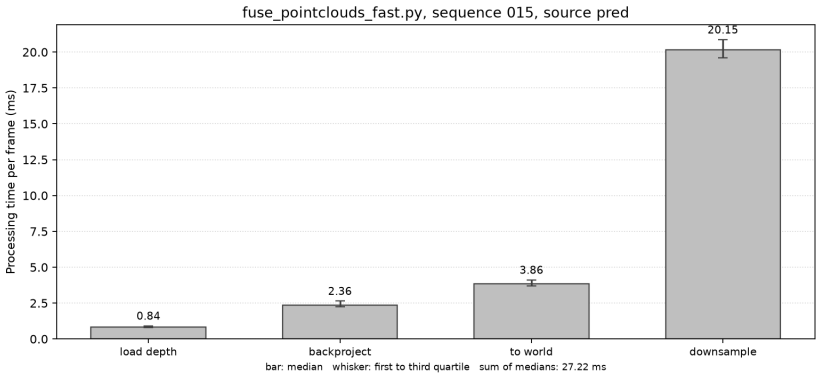


Figure 4.25: Processing time for the point-cloud fusion implementation for one recording.

Considering the complete stages, point-cloud fusion has the lowest measured per-frame processing time, at approximately 27.22 ms, followed by preprocessing, with a median time of 31.08 ms per frame. Depth estimation represents the most computationally demanding stage, with the evaluated configurations requiring approximately 84 to 103 ms per frame. Given these results, this set of processing times may therefore not be ideal in practice in terms of efficiency and real-time constraints.

Regarding real-time processing, there is no threshold for the operation to be in real time for this application, but it can be considered that the processing should match the image acquisition frame rate used in clinical practice. At an acquisition rate of 30 frames per second, this corresponds to approximately 33.3 ms available for the processing of each frame.

Nevertheless, the stage-level measurements alone do not fully describe where the computational cost originates and a more detailed analysis of the individual operations would allow to identify potential optimisation focuses.

For preprocessing, the measured cost is largely associated with input operations, with approximately 27.33 ms per frame. This value should not necessarily be considered representative of a future clinical implementation, since the format and mechanism through which images are provided by the hysteroscopic acquisition or navigation system may differ. For example, receiving frames directly from an acquisition buffer or using a different image representation could reduce the cost associated with file reading. The measured preprocessing time should therefore be interpreted primarily as a characteristic of the current data-processing implementation rather than as an intrinsic requirement of the proposed method.

A similar distinction is relevant for depth estimation, as the step that took more time was the outputs. The latter operations are required in this work, but in a clinical environment data handling and the writing of intermediate outputs could instead remain in memory and be directly passed to subsequent processing stages. Within the measured stage, neural-network inference itself required approximately 13 ms per frame, suggesting the possibility of real-time, also because the network inference is the component that would most directly need to follow the acquisition frame rate.

For the point-cloud fusion, the main computational cost identified was for the downsampling operation. One possible optimisation would be to investigate the use of a larger voxel

size, which could reduce the number of points retained during downsampling and consequently decrease the processing time. This would require a balance between computational efficiency and the preservation of geometric detail. The existing implementation already supports frame subsampling through the stride parameter, which could therefore be investigated as a trade-off between computational cost and reconstruction completeness.

In addition, reconstruction and coverage do not necessarily need to be performed for every acquired frame. For example, the coverage map could be updated at a lower frequency while depth prediction continues at the video frame rate, without affecting the procedure.

The fact that different depth-estimation runs were performed allows to explore the variability of the computational cost. Since frame skipping is applied during training and does not modify the architecture used during inference, no systematic difference in per-frame inference time would be expected between models trained with different skip values. Similarly, freezing encoder layers during fine-tuning does not remove those layers from the inference graph and should therefore not intrinsically reduce the forward-pass prediction time. And results are as expected. The freeze configurations involved more time for the outputs, but they are more likely to reflect execution variability, data loading, data conversion, and similar.

To conclude, the current timing results should be interpreted as a description of the present implementation rather than a definitive statement about real-time feasibility, but they indicate a possible clinical use.

Chapter 5

Conclusions and Future Work

This dissertation investigated the feasibility of adapting a monocular depth estimation framework originally developed for gastrointestinal endoscopy to hysteroscopy, with the aim of recovering three-dimensional information from conventional monocular images of the uterine cavity. This information is a prerequisite for coverage mapping, a future application that could make the completeness of the examination explicit and support structured reporting [31].

To address the lack of hysteroscopic data with ground-truth depth, a synthetic environment was developed in which the uterine geometry and camera acquisition could be controlled while providing depth and camera information for quantitative evaluation. No direct evaluation of the synthetic dataset's diversity and realism was performed.

The developed dataset enabled the adaptation of Endo-SfMLearner to the hysteroscopic domain through transfer learning. The developed dataset enabled the adaptation of Endo-SfMLearner to the hysteroscopic domain through fine-tuning. On the synthetic test data, some configurations achieved improved depth estimation metrics compared with the original pre-trained model. The configuration with the lowest mean AbsRel achieved a value of 0.26 and an RMSE of 2.96 mm, compared with 0.30 and 3.48 mm, respectively, for the original model. Its δ_1 value was 0.73, compared with 0.64 for the original model. However, the selected checkpoint corresponded to the first epoch, so these improvements cannot be confidently attributed to sustained fine-tuning. It is also important to note that the reported errors for the depth estimation results were computed on synthetic test images after independently adjusting the scale of each image using the ratio between the median ground-truth depth and the median predicted depth. These results do not provide sufficient evidence to establish that fine-tuning with synthetic hysteroscopic data is necessary or consistently beneficial for adapting a monocular depth estimation model developed for another endoscopic anatomy to hysteroscopy. The experiments with different training configurations also showed that their influence was not uniform. Learning rate had the most noticeable effect, while batch size, frame skipping and encoder freezing did not provide a consistent advantage under the tested conditions. These observations are specific to the dataset and experimental conditions used, and different anatomical and visual variability could affect the relative performance of the configurations.

The analysis further showed that frame-wise depth accuracy and sequence-level three-dimensional reconstruction should not be considered equivalent, as the fused point clouds still presented local geometric discrepancies and, more importantly, substantial variation in the scale factors required to align individual frames. The reconstruction itself uses a single scale factor for the sequence, rather than independently correcting each frame. Consequently, the observed variation indicates that the predicted depth does not maintain a sufficiently consistent

scale across the sequence for fully coherent reconstruction. This also demonstrates a limitation of frame-wise metrics based on independent scale alignment: errors in scale consistency may remain hidden when frames are evaluated separately but become evident when their predictions are integrated into a common three-dimensional representation.

Evaluation on real hysteroscopic images suggested that the fine-tuned configurations selected at the first epoch understood mainly the representation of local visual cues, artefacts and image details, rather than providing a clear improvement in the overall depth structure. In contrast, the non-fine-tuned model appeared to provide a more consistent depth representation, which may be more favourable for the intended coverage mapping application. In turn, the comparison of two image framings for the same image showed reduced dependence on the surrounding visual contexts, either with and without fine-tuning. However, the absence of ground-truth depth in the real data means that these observations cannot be interpreted as quantitative evidence of depth accuracy. The real-domain results should therefore be regarded as an analysis on geometry, consistency and transfer beyond the synthetic environment rather than as validation of the estimated metric depth and it can be concluded that the adaptation to real-world scenarios remains to be demonstrated.

The fused point-cloud results did not reproduce the surface, with local differences from the reference. This is, the reconstruction across sequences has not been consistent. For the analysed sequence, the reconstruction, performed with a global scale factor applied to entire scenes, achieved an accuracy RMSE of 1.97 mm and a completeness RMSE of 1.17 mm. These results support the use of point-cloud reconstruction as a complementary evaluation of depth estimation, since sequence-level geometric behaviour reveals aspects that are not captured by frame-wise depth metrics. Nevertheless, the remaining scale inconsistency and geometric discrepancies prevent the reconstruction from being considered sufficiently reliable for coverage mapping and additional applications.

The main limitations of the present work are related to the available data and the transfer from simulation to real hysteroscopy. The synthetic dataset provides controlled geometry and ground truth, but its anatomical and visual variability is limited, with no pathological variability or a strategy for increasing real-image appearance, and the simulated anatomy is static, whereas real hysteroscopy may involve changes in tissue deformation and other appearance variations. Furthermore, the fact that most of the real images used for evaluation were also used as sources for the generation of the synthetic data may have introduced a source-related bias, which may have reduced the independence of the real-image evaluation and potentially limited the extent to which the observed results can be interpreted as generalisation to real images. In addition, the conclusions regarding real images remain qualitative and based on consistency properties.

The computational analysis showed that the current implementation does not constitute a real-time end-to-end pipeline, but also indicated that this limitation is not primarily imposed by neural-network inference. The depth network required approximately 13 ms per frame on the tested GPU, whereas the complete measured processing stage was substantially slower due to additional operations, including output handling and writing intermediate results. This

suggests that optimisation of the implementation, particularly by avoiding unnecessary disk operations and reducing the amount of data processed or stored, could considerably reduce the overall processing time. The point-cloud reconstruction may also benefit from further optimisation if it is required to operate continuously.

Future work should focus first on increasing the anatomical and visual diversity in order to have representation of pathological conditions, tissue deformation, anatomical geometrical plausibility and more, to improve the realism of the simulated environment.

If conducted with a synthetic dataset, it would be meaningful to study it more extensively to make it as similar as possible to the target domain, including the study of reprojection-based consistency as an additional assessment of whether the predicted depth and camera motion are geometrically compatible with the observed images. It would also be relevant to train the algorithms using a more artificial version of the dataset alongside the the more realistic one, in order to investigate whether there is a significant difference between the results obtained with different levels of realism. A phantom could potentially complement this evaluation, at least to assess the compatibility of the movements, camera behaviour, field of view and other relevant characteristics. When using ex-vivo tissue, some of these aspects are already fulfilled but further work would still be necessary to address tissue deformation and other factors.

It will also be important to perform further training experiments, explore the effect of the number of epochs and investigate other algorithms for comparison. In this context, future work in this area should also consider the computational requirements of the proposed approaches, with particular attention to whether they can be trained and applied using less powerful GPUs.

A further priority is the improvement of sequence-level scale consistency. Since three-dimensional reconstruction requires a common scale across frames, future approaches could investigate temporal constraints, additional scale-consistency losses or alternative strategies for estimating and maintaining a global sequence scale. The depth and camera-motion predictions should also be evaluated jointly on real sequences, since inaccuracies in pose estimation can affect the reconstructed geometry. Finally, the computational pipeline should be redesigned for streaming operation, avoiding intermediate disk storage and allowing reconstruction-related outputs to be updated at an appropriate frequency rather than necessarily at every acquired frame.

Overall, this work investigated the feasibility of adapting monocular depth estimation to hysteroscopy using synthetic data. The main conclusions are that the depth estimation through fine-tuning per image improved in some configurations on synthetic data, even thought it did not consistently translate into improved performance across the different stages of the proposed pipeline, also that the adaptation to real-world scenarios remains to be demonstrated, and that a consistent reconstruction across sequences has not been demonstrated. The work highlightings that an improvement at one level, such as frame-wise depth, does not necessarily imply an equivalent improvement at the next, such as the point-cloud completeness. The findings show that reliable three-dimensional reconstruction requires not only accurate depth estimation, but also consistent geometric behaviour across sequences and validation in the real domain. Consequently, sequence-level geometric consistency and real-domain validation

remain essential challenges for the future development and clinical adoption of computer-assisted hysteroscopy systems.

References

[1] "THE 17 GOALS | sustainable development," Accessed: Jul. 19, 2026. [Online]. Available: https://sdgs.un.org/goals

[2] "Engenharia para o desenvolvimento sustentavel: Resumo - UNESCO digital library," Accessed: Jul. 19, 2026. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000375634_por

[3] S. G. Vitale et al., *Hysteroscopy: Where did we start, and where are we now? the compelling story of what many considered the "cinderella" of gynecological endoscopy*, Aug. 16, 2024. DOI: 10.1007/s00404-024-07677-x. [Online]. Available: https://link.springer.com/10.1007/s00404-024-07677-x

[4] M. H. Emanuel, *New developments in hysteroscopy*, Jun. 2013. DOI: 10.1016/j.bpobgyn.2012.11.005. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1521693413000059

[5] V. Poliakov, K. Niu, D. Tsetserukou, and E. V. Poorten, *An in-office hysteroscopy vr/haptic simulation platform for training in spatial navigation and passage of the cervical canal*, 2022. DOI: 10.1109/TMRB.2022.3188438.

[6] E. Boneš, M. Gergolet, C. Bohak, Ž. Lesar, and M. Marolt, *Automatic segmentation and alignment of uterine shapes from 3D ultrasound data*, Aug. 2024. DOI: 10.1016/j.compbiomed.2024.108794. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0010482524008795

[7] Thompson Louisa. "The uterus - structure - location - vasculature - TeachMeAnatomy," Accessed: Dec. 26, 2025. [Online]. Available: https://teachmeanatomy.info/pelvis/female-reproductive-tract/uterus/

[8] Cleveland Clinic. "Uterus: Anatomy, function, size, position & conditions," Accessed: Dec. 26, 2025. [Online]. Available: https://my.clevelandclinic.org/health/body/22467-uterus

[9] T. M. Hoagland. "Uterus anatomy: Overview, gross anatomy, natural variants," Accessed: Dec. 26, 2025. [Online]. Available: https://emedicine.medscape.com/article/1949215-overview?form=fpf#a4

[10] Refaey, Mohamed. "Uterus | radiology reference article | radiopaedia.org," Radiopaedia, Accessed: Dec. 26, 2025. [Online]. Available: https://radiopaedia.org/articles/uterus

[11] E. R. Sokol. "Clinical anatomy of the uterus, fallopian tubes, and ovaries | GLOWM," Accessed: Dec. 26, 2025. [Online]. Available: http://www.glowm.com/section-view/heading/clinical-anatomy-of-the-uterus-fallopian-tubes-and-ovaries/item/1

[12] GeeksforGeeks. "Uterus diagram," Accessed: Nov. 7, 2025. [Online]. Available: https://www.geeksforgeeks.org/biology/uterus-diagram/

[13] Chiswo, Ruvimbo. "Myometrium," Kenhub, Accessed: Dec. 26, 2025. [Online]. Available: https://www.kenhub.com/en/library/anatomy/myometrium

[14] A. Di Spiezio Sardo, G. Calagna, M. Guida, A. Perino, and C. Nappi, *Hysteroscopy and treatment of uterine polyps*, Oct. 2015. DOI: 10.1016/j.bpobgyn.2015.06.005. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S152169341500108X

[15] C. Mayo. "Uterine polyps-uterine polyps - symptoms & causes," Accessed: Dec. 26, 2025. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/uterine-polyps/symptoms-causes/syc-20378709

[16] A. Al-Harthy and M. Arafa, *Clinicopathological study of endometrial polyps: Experience from a tertiary care center*, Aug. 14, 2025. DOI: 10.3897/folmed.67.e153005. [Online]. Available: https://foliamedica.bg/article/153005/

[17] N. P. Nijkang, L. Anderson, R. Markham, and F. Manconi, *Endometrial polyps: Pathogenesis, sequelae and treatment*, Jan. 2019. DOI: 10.1177/2050312119848247. [Online]. Available: https://journals.sagepub.com/doi/10.1177/2050312119848247

[18] H. Monira and M. W. S. "Endometrial polyp," Accessed: Dec. 26, 2025. [Online]. Available: https://www.pathologyoutlines.com/topic/uterusendopolyp.html

[19] V. Tanos et al., *The management of polyps in female reproductive organs*, Jul. 2017. DOI: 10.1016/j.ijsu.2017.05.012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1743919117303941

[20] E. Rahou, V. Borodin, D. Perţicaş, A. Florea, M. Hajjaj, and A. Elassfouri, "Transforming healthcare professions: The role of artificial intelligence and data analytics in driving innovation and efficiency," presented at the Lecture Notes in Networks and Systems, vol. 1587 LNNS, 2026, 177–186. DOI: 10.1007/978-3-032-01970-7_18. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105020891157&doi=10.1007%2f978-3-032-01970-7_18&partnerID=40&md5=3bf47853970c760b62927074a03e6706

[21] P. Tarneja, V. Tarneja, and B. Duggal, *Complications of hysteroscopic surgery*, Oct. 2002. DOI: 10.1016/S0377-1237(02)80090-9. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377123702800909

[22] C. Cleveland. "What is a d&c?" Cleveland Clinic, Accessed: Nov. 11, 2025. [Online]. Available: https://my.clevelandclinic.org/health/procedures/dilation-and-curettage

[23] S. Liu et al., *Comparison of prognosis of patients with endometrial cancer after hysteroscopy versus dilatation and curettage: A multicenter retrospective study*, 2023. DOI: 10.3389/fmed.2022.1097133. [Online]. Available: https://www.scopus.com/pages/publications/85146890025?inward

[24] A. J. Will and K. E. Sanchack, "Endometrial biopsy," in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2025. Accessed: Nov. 11, 2025. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK541135/

[25] C. Clinic. "What is an endometrial biopsy?" Cleveland Clinic, Accessed: Nov. 11, 2025. [Online]. Available: https://my.clevelandclinic.org/health/diagnostics/15676-endometrial-biopsy

[26] "Office hysteroscopy or endometrial biopsy | elite gynecology miami," Accessed: Nov. 11, 2025. [Online]. Available: https://www.toplinemd.com/elite-gynecology-miami/office-hysteroscopy-or-endometrial-biopsy/

[27] P. M. De Silva, P. P. Smith, N. A. M. Cooper, and T. J. Clark, *Outpatient hysteroscopy: (green-top guideline no. 59)*, Dec. 2024. DOI: 10.1111/1471-0528.17907.

[28] "Hysteroscopy procedures market size | industry report, 2030," Accessed: Jan. 7, 2026. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/hysteroscopy-procedures-market-report

[29] R. Watrowski et al., *Artificial intelligence, deep learning, and computer vision in hysteroscopy: A systematic review*, 2026. DOI: 10.3390/diagnostics16121899. [Online]. Available: https://www.mdpi.com/2075-4418/16/12/1899

[30] H. Curlin, A. Cholkeri-Singh, J. G. G. Leal, and T. Anderson, *Hysteroscopic access and uterine cavity evaluation 12 months after endometrial ablation with the cerene cryotherapy device*, Mar. 1, 2022. DOI: 10.1016/j.jmig.2021.11.016. [Online]. Available: https://doi.org/10.1016/j.jmig.2021.11.016

[31] U. Catena et al., *Structured reporting of diagnostic hysteroscopy findings in women with reproductive disorders: An ESGE–ESHRE–IFFS international consensus*, Sep. 15, 2026. DOI: 10.52054/FVVO.2026.547. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC13575364/

[32] F. P. Toscano, M. D'Angelo, L. Riello, M. Ardovino, and A. Mollo, *Clinical factors associated with pain during outpatient hysteroscopy in a standardized vaginoscopic setting: A prospective observational study*, 2026. DOI: 10.3390/jcm15166335. [Online]. Available: https://www.mdpi.com/2077-0383/15/16/6335

[33] K. Niu et al., *Endoscopic depth estimation based on deep learning: A survey*, 2026. DOI: 10.1016/j.neucom.2026.133958. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092523122601355X

[34] K. B. Ozyoruk et al., *EndoSLAM dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos*, Jul. 1, 2021. DOI: 10.1016/j.media.2021.102058. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841521001043

[35] K. İncetan et al., *VR-caps: A virtual environment for capsule endoscopy*, May 2021. DOI: 10.1016/j.media.2021.101990. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1361841521000360

[36] A. Rau et al., *Implicit domain adaptation with conditional generative adversarial networks for depth prediction in endoscopy*, Jul. 1, 2019. DOI: 10.1007/s11548-019-01962-w. [Online]. Available: https://doi.org/10.1007/s11548-019-01962-w

[37] A. F. Fernandes, "3d uterine cavity reconstruction for computer-assisted hysteroscopy," M.S. thesis, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2025. [Online]. Available: https://repositorio-aberto.up.pt/bitstream/10216/168647/2/734972.pdf

[38] C. Clinic. "Speculum: Design, purpose, types, exams & what to expect," Cleveland Clinic, Accessed: Nov. 11, 2025. [Online]. Available: https://my.clevelandclinic.org/health/drugs/24238-speculum

[39] Z. Yang, J. Dai, and J. Pan, *3d reconstruction from endoscopy images: A survey*, Jun. 2024. DOI: 10.1016/j.compbiomed.2024.108546. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0010482524006309

[40] X. Wang, B. Yang, M. Wei, L. Liu, J. Zhang, and Y. Nie, *Deep learning for endoscopic depth estimation: A review*, Dec. 2025. DOI: 10.1016/j.displa.2025.103086. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0141938225001234

[41] B. Göbel, A. Reiterer, and K. Möller, *Quantitative evaluation of camera-based 3D reconstruction in laparoscopy: A review*, 2024. DOI: 10.1016/j.ifacol.2024.11.109. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896324022365

[42] C. E. Schroeder, D. A. Wilson, T. Radman, H. Scharfman, and P. Lakatos, *Dynamics of active sensing and perceptual selection*, Apr. 2010. DOI: 10.1016/j.conb.2010.02.010. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC2963579/

[43] The MathWorks, Inc. "What is structure from motion? - MATLAB & simulink," Accessed: Aug. 16, 2026. [Online]. Available: https://www.mathworks.com/help/vision/ug/what-is-structure-from-motion.html

[44] H. Song, C. Liu, and H. Dai, "BundledSLAM: An accurate visual SLAM system using multiple cameras," in *2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, ISSN: 2689-6621, vol. 7, Mar. 2024, 106–111. DOI: 10.1109/IAEAC59436.2024.10503743. Accessed: Aug. 16, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/10503743/citations

[45] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, *MonoSLAM: Real-time single camera SLAM*, Jun. 2007. DOI: 10.1109/TPAMI.2007.1049. [Online]. Available: https://ieeexplore.ieee.org/document/4160954

[46] F. Mahmood, R. Chen, and N. J. Durr, *Unsupervised reverse domain adaptation for synthetic medical images via adversarial training*, Dec. 2018. DOI: 10.1109/TMI.2018.2842767. [Online]. Available: https://ieeexplore.ieee.org/document/8370747/

[47] S. Shao et al., *Self-supervised monocular depth and ego-motion estimation in endoscopy: Appearance flow to the rescue*, Apr. 2022. DOI: 10.1016/j.media.2021.102338. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1361841521003832

[48] Q. He, G. Feng, S. Bano, D. Stoyanov, and S. Zuo, *MonoLoT: Self-supervised monocular depth estimation in low-texture scenes for automatic robotic endoscopy*, Oct. 2024. DOI: 10.1109/JBHI.2024.3423791. [Online]. Available: https://ieeexplore.ieee.org/document/10587075/

[49] W. Fan, W. Jiang, H. Fang, H. Shi, J. Chen, and X. Luo, "Simultaneous monocular endoscopic dense depth and odometry estimation using local-global integration networks," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, M. G. Linguraru et al., Eds., Cham: Springer Nature Switzerland, 2024, 564–574, ISBN: 978-3-031-72089-5.

[50] B. Li, B. Liu, M. Zhu, X. Luo, and F. Zhou, *Image intrinsic-based unsupervised monocular depth estimation in endoscopy*, Dec. 2025. DOI: 10.1109/JBHI.2024.3400804. [Online]. Available: https://ieeexplore.ieee.org/document/10530343/

[51] B. Cui, M. Hoque, L. Bai, and A. Wang, "Endodac: Efficient adapting foundation model for self-supervised depth estimation from any endoscopic camera," in Oct. 2024, 208-218, ISBN: 978-3-031-72088-8. DOI: 10.1007/978-3-031-72089-5_20.

[52] B. Cui, M. Islam, L. Bai, and H. Ren, *Surgical-DINO: Adapter learning of foundation models for depth estimation in endoscopic surgery*, Jun. 1, 2024. DOI: 10.1007/s11548-024-03083-5. [Online]. Available: https://doi.org/10.1007/s11548-024-03083-5

[53] Y. Li et al., *SfMDiffusion: Self-supervised monocular depth estimation in endoscopy based on diffusion models*, 2025.

[54] Y. Liu, D. Yu, Y. Xiao, T. He, and L. Liu, *Scale-consistent 3D reconstruction in monocular colonoscopy via camera-intrinsics-guided learning*, Mar. 2026. DOI: 10.1016/j.compmedimag.2026.102734. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0895611126000376

[55] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1851–1860. DOI: 10.1109/CVPR.2017.700.

[56] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[57] D. D. Tully. "Synthetic data creation for visual AI training," Scenegraph Studios, Accessed: Dec. 31, 2025. [Online]. Available: https://scenegraphstudios.com/ai/synthetic-data-creation-for-visual-ai-training/

[58] A. M. "From VR twins to diffusion dreams: The rise of synthetic data in AI," Medium, Accessed: Dec. 31, 2025. [Online]. Available: https://medium.com/@abhishek.gmit/from-vr-twins-to-diffusion-dreams-the-rise-of-synthetic-data-in-ai-2e623418e162

[59] LIGHTLY. "The ultimate guide to synthetic data generation," Accessed: Dec. 31, 2025. [Online]. Available: https://www.lightly.ai/blog/synthetic-data

[60] Eleks, Inc. "Synthetic data generation explained: A detailed overview," ELEKS: Enterprise Software Development, Technology Consulting, Accessed: Dec. 31, 2025. [Online]. Available: https://eleks.com/types-of-software-development/synthetic-data-generation/

[61] B. H. Jeong, H. K. Kim, and Y. D. Son, *Depth estimation from monocular endoscopy using simulation and image transfer approach*, Oct. 2024. DOI: 10.1016/j.compbiomed.2024.109038. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0010482524011235

[62] H. Walkner, L. Krames, and W. Nahm, *Synthetic data in supervised monocular depth estimation of laparoscopic liver images*, Dec. 1, 2024. DOI: 10.1515/cdbme-2024-2162. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/cdbme-2024-2162/html

[63] A. Juliani et al., *Unity: A general platform for intelligent agents*, Sep. 2018. DOI: 10.48550/arXiv.1809.02627.

[64] Y. Yang et al., *A geometry-aware deep network for depth estimation in monocular endoscopy*, 2023. DOI: https://doi.org/10.1016/j.engappai.2023.105989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197623001732

[65] A. Richter, T. Steinmann, J.-C. Rosenthal, and S. J. Rupitsch, *Advances in real-time 3D reconstruction for medical endoscopy*, 2024. DOI: 10.3390/jimaging10050120.

[66] G. Manni, C. Lauretti, F. Prata, R. Papalia, L. Zollo, and P. Soda, *BodySLAM: A generalized monocular visual SLAM framework for surgical applications*, Oct. 2026. DOI: 10.1016/j.imavis.2026.106094. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0262885626002015

[67] R. A. Newcombe et al., "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136. DOI: 10.1109/ISMAR.2011.6092378.

[68] J. Zhong, H. Ren, Q. Chen, and H. Zhang, *A review of deep learning-based localization, mapping and 3D reconstruction for endoscopy*, Jun. 2025. DOI: 10.1007/s12213-024-00181-0. [Online]. Available: https://link.springer.com/10.1007/s12213-024-00181-0

[69] J. Bian et al., "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/6364d3f0f495b6ab9dcf8d3b5c6e0b01-Paper.pdf

[70] Z. Cui, Y. He, P. Zhang, Y. Hu, H. Jin, and S. Liu, "Virtual reality navigation system of nasal endoscopy with real surface texture information," in *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, Xining, China: IEEE, Jul. 15, 2021, 135–140, ISBN: 978-1-6654-3678-6. DOI: 10.1109/RCAR52367.2021.9517378. Accessed: Aug. 15, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/9517378/

[71] L. Qiu and H. Ren, *Endoscope navigation with SLAM-based registration to computed tomography for transoral surgery*, Jun. 2020. DOI: 10.1007/s41315-020-00127-2. [Online]. Available: https://link.springer.com/10.1007/s41315-020-00127-2

[72] M. Koeda et al., "Position and orientation registration of intra-abdominal point cloud generated from stereo endoscopic images and organ 3D model using open3d," in *Human-Computer Interaction. Technological Innovation*, M. Kurosu, Ed., Cham: Springer International Publishing, 2022, 52–65, ISBN: 978-3-031-05409-9.

[73] S. Zhang, L. Zhao, S. Huang, R. Ma, B. Hu, and Q. Hao, "3d reconstruction of deformable colon structures based on preoperative model and deep neural network," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China: IEEE, May 30, 2021, 1875–1881, ISBN: 978-1-7281-9077-8. DOI: 10.1109/ICRA48506.2021.9561772. Accessed: Aug. 15, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/9561772/

[74] O. Zenteno et al., *Optical biopsy mapping on endoscopic image mosaics with a marker-free probe*, Apr. 2022. DOI: 10.1016/j.compbiomed.2022.105234. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0010482522000269

[75] M. A. Kokko, D. W. Van Citters, J. D. Seigne, and R. J. Halter, *A particle filter approach to dynamic kidney pose estimation in robotic surgical exposure*, Jun. 1, 2022. DOI: 10.1007/s11548-022-02638-8. [Online]. Available: https://doi.org/10.1007/s11548-022-02638-8

[76] H. Zhou and J. Jayender, *Real-time nonrigid mosaicking of laparoscopy images*, Jun. 2021. DOI: 10.1109/TMI.2021.3065030. [Online]. Available: https://ieeexplore.ieee.org/document/9374480/

[77] Y. Wang, Y. Long, S. H. Fan, and Q. Dou, *Neural rendering for stereo 3D reconstruction of deformable tissues in robotic surgery*, 2022. arXiv: 2206.15255 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2206.15255

[78] C. Yang, K. Wang, Y. Wang, X. Yang, and W. Shen, *Neural lerplane representations for fast 4D reconstruction of deformable tissues*, 2023. arXiv: 2305.19906 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2305.19906

[79] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, *Tanks and temples: Benchmarking large-scale scene reconstruction*, Aug. 31, 2017. DOI: 10.1145/3072959.3073599. [Online]. Available: https://dl.acm.org/doi/10.1145/3072959.3073599

[80] M. Brenndoerfer. "Fine-tuning learning rates: LLRD, warmup & decay strategies." Section: Interactive, Accessed: Aug. 16, 2026. [Online]. Available: https://mbrenndoerfer.com/writing/fine-tuning-learning-rates-llrd-warmup-decay-transformers

[81] GeeksforGeeks. "How to choose batch size and number of epochs when fitting a model?" GeeksforGeeks. Section: Machine Learning, Accessed: Aug. 16, 2026. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/how-to-choose-batch-size-and-number-of-epochs-when-fitting-a-model/

[82] M. Mascarenhas et al., *Artificial intelligence and hysteroscopy: A multicentric study on automated classification of pleomorphic lesions*, Aug. 3, 2025. DOI: 10.3390/cancers17152559. [Online]. Available: https://www.mdpi.com/2072-6694/17/15/2559

[83] Y. Zhang et al., *Deep learning model for classifying endometrial lesions*, Jan. 6, 2021. DOI: 10.1186/s12967-020-02660-x. [Online]. Available: https://doi.org/10.1186/s12967-020-02660-x

[84] Y. Takahashi et al., *Automated system for diagnosing endometrial cancer by adopting deep-learning technology in hysteroscopy*, Publisher: Public Library of Science, Mar. 31, 2021. DOI: 10.1371/journal.pone.0248526. [Online]. Available: https://doi.org/10.1371/journal.pone.0248526

[85] J.-H. Yang, C.-D. Chen, S.-U. Chen, Y.-s. Yang, and M.-J. Chen, *Factors influencing the recurrence potential of benign endometrial polyps after hysteroscopic polypectomy*, Dec. 2015. DOI: 10.1371/journal.pone.0144857.

[86] U.S. National Library of Medicine, *Visible Human Project: Getting the Data*, https://www.nlm.nih.gov/research/visible/getting_data.html, Jul. 11, 2025.

[87] B. J. Erickson, D. Mutch, L. Lippmann, and R. Jarosz, *The Cancer Genome Atlas Uterine Corpus Endometrial Carcinoma Collection (TCGA-UCEC)*, The Cancer Imaging Archive, 2016. DOI: 10.7937/K9/TCIA.2016.GKJ0ZWAC.

[88] E. Boneš, M. Gergolet, C. Bohak, Ž. Lesar, and M. Marolt, *Automatic Segmentation and Alignment of Uterine Shapes from 3D Ultrasound Data*, 2024. DOI: 10.1016/j.compbiomed.2024.108794.

[89] N. Konz et al., *Fréchet radiomic distance (FRD): A versatile metric for comparing medical imaging datasets*, May 2026. DOI: 10.1016/j.media.2026.103943. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC13224392/

[90] N. Young. "Frechet inception distance (FID): What it is and how it works," Accessed: Aug. 25, 2026. [Online]. Available: https://www.teachfloor.com/blog/frechet-inception-distance

[91] X. Ma et al., *Frozen weights as prior for parameter-efficient fine-tuning*, 2025. DOI: 10.1109/ACCESS.2025.3529807.

Appendix A

Interviews

In order to understand and validate the project's motivation and need, along with the selection of key stakeholders and help in understanding of their benefits and costs, a search for relevant contacts was carried.

The most important people to contact were the gynecologists, as they would give an insight in each of the mentioned aspects and be the ones to actually determinate the validation of the motivation and need.

The search involved the look for physicians in the Gynaecology-Obstetrics area, with specific knowledge and practice of hysperoscopy. The investigation involved identifying professionals from CUF and other clinics located in Porto, Madeira, Viseu, England and Switzerland. For those involved in teaching practical classes, a systematic search was conducted using the Medicine course plan of the Faculty of Medicine at the University of Porto. In addition, contact with professionals from other areas or in another scope was sought whenever the need arose.

The method of contact varied between LinkedIn, e-mail, and text message, depending on the available resources. Some physicians were impossible to contact in any way, some did not accept the LinkedIn connection and some did not answer.

The interviews were conducted with a Portuguese gynaecologist from Barreiro and a Portuguese gastroenterologist and researcher. The latter interview focused on an article co-authored by the interviewee, while the former, due to being aimed at gathering information, is summarised in Table A1.

Table A1: Registration of the performed interviews.

Interviewee, Occupation, Scope	**Key Insights**
Gynecologist, at CUF, "Clínica CUF Quinta da Lomba", Barreiro, Portugal	• Scratching the endometrium with a hysteroscope is not serious, it is a minor issue, which seems to indicate that there will be no harm if there are more false positives (which would be with the aim of circumventing the probability of algorithm error affecting the patient's diagnosis (in the case of FN)); • For polyps, care must be taken because their association with the endometrium is continuous, and even more so in the case of menopause, when the tissue layer is thinner; • The following positive aspects of the procedure were highlighted: good size for patient comfort and entry through the canal, improvement in the functioning of the irrigation pumps, and the imaging equipment (it was emphasised that the excellent image definition makes all the difference); • When asked directly about improvement in the imaging software, it was simply stated that it is already good and has excellent definition; • Viewing polyps comes with experience, especially the smaller ones; • Highlighted as a difficulty: the forceps not having a more effective grip, also due to the size of the hysteroscope being very small; • A second procedure (and more risks) will be necessary if it is not possible to remove the lesions in a timely manner, which is mainly due to the difficulty of grasping them with forceps. The work to be done would not help in this regard to avoid further surgery; • It is important that the surgery is not prolonged, that it takes a reasonable amount of time, because the longer it takes, the greater the risk of saline solution intravasation. The work to be done could help to ensure that it is done in a timely manner; • Other risks: haemorrhage, perforation or false passages, infection (minimal risk), embolisation (rare). There is no proportionality in their occurrence with longer duration, but more interaction during surgery, associated with its prolongation, may cause them; • Hysteroscopy can take place in the operating theatre or in the labour room. In the first case, the entire team is involved. In the second, only the gynaecological surgeon and a nurse who operates the pump. This reinforces the idea that the main stakeholder at this stage is the surgeon; • Equipment cost: 2 Hysteroscopes and irrigation pump: 50,000 €; Forceps and scissors (wear easily): recurring costs, typically several hundred euros; Imaging system: highest-cost component of the setup; • There is no standard imaging software for hysteroscopy in Portugal; it varies greatly, so sales in our country alone would have to be attempted across the various types used.

Appendix B

Blender Shading Characteristics and Defined Material

This appendix presents the Blender's "Shader Editor" produced to obtain a uterus interior as close as possible to the real visualisation. This is presented in Figure B1 and the result for the material is presented in Figure B2.

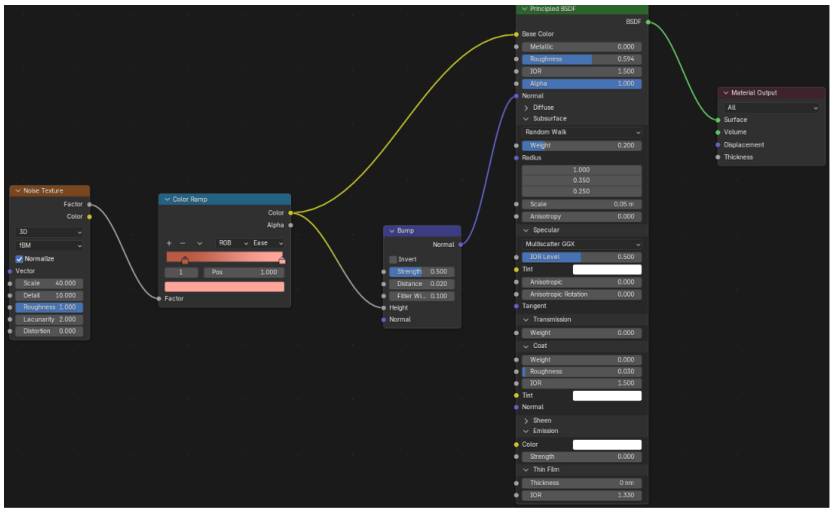


Figure B1: Blender's "Shader Editor" construction and connection of nodes in order to have the desired effect, a similar visual to the intrauterine tissue.

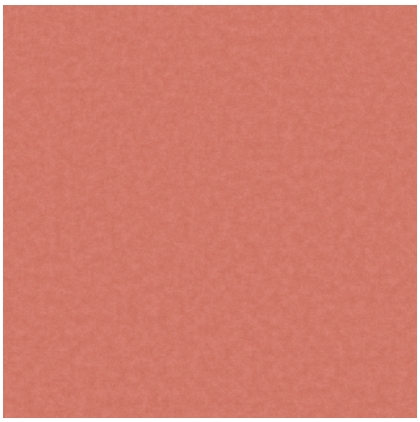


Figure B2: visualisation of the defined material shading created in Blender, corresponding to the construction in Figure B1.

The "Color Ramp" colors in Figure B1 are according to the variety of colours observed across the real hysteroscopic RGB images using a pixel-colour inspection tool.

Appendix C

Unity Implementation and Data Acquisition

This appendix explores the Unity scripts used to move the virtual endoscope and to acquire the synthetic dataset, together with the Inspector settings used during acquisition. The scripts were implemented with the assistance of a generative AI tool for the Unity-specific API, and the resulting code was reviewed, tested and modified as needed.

C.1 Endoscope Movement

The `EndoscopeController.cs` moves the virtual endoscope through the uterine model. Translation is driven by the keyboard (WASD, a vertical pair, and a boost key) and the viewing direction by the mouse. The script also places the camera at a predefined start point when the simulation begins and restores that pose on a reset key. Its parameters, namely speeds, mouse sensitivity, pitch limits and the per-frame rotation cap, are exposed in the Inspector, so the movement configuration can be changed without editing the source.

MovementDelta

All motion advances by video time rather than by wall-clock time, as this option would have the clamp saturating on every frame, so a frame with a key held always moved a fixed amount and a frame without one moved nothing, turning translation into a sequence of teleports separated by stationary frames. Returning `Time.deltaTime` instead, together with `Time.captureFramerate` to 1/fps during recording, allows a reproducible trajectory, consistent with the `deltaTime` column written to the pose file.

```
// Time.captureFramerate pins Time.deltaTime to exactly 1/fps while recording,
// so this is also what makes the trajectory reproducible and consistent with
// the deltaTime column written to the pose CSV.
float MovementDelta()
{
    return Time.deltaTime;
}
```

Listing C1: `MovementDelta`: the movement time base.

HandleRotation

Mouse movement is converted to yaw and pitch and then clamped to a maximum number of degrees per frame. The clamp is applied after the sensitivity and in degrees, as a pixel-based clamp would need re-tuning every time the sensitivity changed.

```
void HandleRotation()
{
    Vector2 mouseDelta = Mouse.current.delta.ReadValue();

    float deltaYaw = mouseDelta.x * mouseSensitivity;
    float deltaPitch = -mouseDelta.y * mouseSensitivity;

    if (maxDegreesPerFrame > 0f)
    {
        Vector2 rot = Vector2.ClampMagnitude(
            new Vector2(deltaYaw, deltaPitch), maxDegreesPerFrame);
        deltaYaw = rot.x;
        deltaPitch = rot.y;
    }

    yaw += deltaYaw;
    pitch = Mathf.Clamp(pitch + deltaPitch, minPitch, maxPitch);

    transform.rotation = Quaternion.Euler(pitch, yaw, 0f);
}
```

Listing C2: `HandleRotation` Function: mouse input to a rotation, capped per frame.

`HandleMovement` (standard WASD translation scaled by `MovementDelta`), `Start` and `ResetToStart` are omitted.

C.2 Data Acquisition

For the data acquisition, scripts render the ground-truth depth, write each captured frame to disk and constrain the camera trajectory so that the acquired sequences carry usable parallax.

C.2.1 Ground-truth Depth Rendering

`EuclideanDepthPassHDRP.cs` is a High Definition Render Pipeline (HDRP) Custom Pass that converts the camera depth buffer into the true Euclidean distance from the optical centre to each visible surface point, in scene units. The resulting depth is written to a dedicated 32-bit floating-point render texture, with no exposure, colour curve, or tone mapping applied. Pixels with no geometry receive a configurable sentinel value (-1 by default). The pass can also optionally write the world-space surface normals to a second texture. Its `Setup` method follows the standard Custom Pass pattern by resolving the shader, creating the material, and allocating the required render targets.

The `Execute` function performs the full-screen rendering of the depth and, when enabled, the normal map. Although the Custom Pass is hosted in a global Custom Pass Volume, the execution is restricted to the game camera. This is necessary because a global volume is also evaluated by other cameras used by the Unity editor, including the Scene view, preview cameras, and material-thumbnail cameras. Since the pass redirects the render target to its dedicated texture, allowing it to execute for these cameras could cause their rendering to be written to the depth texture instead of their normal render targets. The camera type check therefore ensures that the pass only affects the camera used for data capture.

```
protected override void Execute(CustomPassContext ctx)
{
if (ctx.hdCamera.camera.cameraType != CameraType.Game)
return;

‘‘‘
material.SetFloat(s_InvalidDepthValue, invalidDepthValue);

CoreUtils.SetRenderTarget(ctx.cmd, rtHandle, ClearFlag.Color, Color.clear);
CoreUtils.DrawFullScreen(ctx.cmd, material, shaderPassId: 0);

if (normalHandle != null)
{
    CoreUtils.SetRenderTarget(ctx.cmd, normalHandle, ClearFlag.Color, Color.clear);
    CoreUtils.DrawFullScreen(ctx.cmd, material, shaderPassId: 1);
}

// Restore the camera render target so as not to contaminate what HDRP
// draws next in this frame.
CoreUtils.SetRenderTarget(ctx.cmd, ctx.cameraColorBuffer, ctx.cameraDepthBuffer);
‘‘‘

}
```

Listing C3: `Execute` function. The symbol "’ represents omitted code.

C.2.2 Capture Rig and Frame Recording

The `EndoscopeRigSetup.cs` script is an editor tool (`Tools > Endoscope > Build Capture Rig`) that builds or repairs the capture rig in the open scene. A single camera is used so that the RGB image, and depth map originate from the same render, so they are pixel-aligned by construction. To enforce this, the script untags and disables every other camera it finds before configuring the capture camera.

```
foreach (var other in Object.FindObjectsByType<Camera>(FindObjectsSortMode.None))
{
if (other == capture)
continue;

‘‘‘
if (other.CompareTag("MainCamera"))
    other.gameObject.tag = "Untagged";

if (other.enabled)
    other.enabled = false;
‘‘‘

}

capture.gameObject.tag = "MainCamera";
capture.enabled = true;
```

```
capture.nearClipPlane = 0.01f;
```

Listing C4: Enforcing a single active, MainCamera-tagged camera before the rig is wired. The symbol "' represents omitted code.

The rest of `BuildRig` adds to the camera `EndoscopeController`, `EndoscopeRecorder`, `EndoscopeScriptedPath` and `EndoscopeManualOrbit` if they are not already present. It also creates or reuses the Custom Pass Volume that hosts the Euclidean Depth Pass and wires it into the recorder. The capture resolution is decided solely by the recorder's Inspector fields, so that a value set there is not silently reverted when the rig is rebuilt.

`EndoscopeRecorder.cs` performs the frame-by-frame capture. For each sequence `<NNN>`, it writes `rgb/<NNNNNNN>.png`, the Euclidean depth as `depth/<NNNNNNN>.exr`, `poses_<NNN>.csv`, and `intrinsics_<NNN>.json`. All of these are written by this component using the same frame counter, ensuring that the RGB image, depth map and line i of the pose file correspond to the same simulated instant. `BeginRecording`, which creates the folders and opens the CSV and `SavePose`, which writes one CSV row containing the position and quaternion per frame, are standard file-handling operations and are omitted. `SaveRGB` is also omitted, as it mirrors `SaveDepth` but writes a PNG.

CaptureLoop

Two aspects of the capture process are not part of the default frame handling. First, a fixed number of warm-up frames is discarded before recording starts. Second, the camera is rendered explicitly with `cam.Render()` before any data are read back. This explicit rendering forces the Custom Pass to execute and fill the depth texture for the current frame, ensuring that the RGB image and depth map are obtained from the same render.

```
IEnumerator CaptureLoop()
{
while (true)
{
yield return new WaitForEndOfFrame();

‘‘‘
    if (!recording)
        continue;

    if (warmupCounter < warmupFrames)
    {
        warmupCounter++;
        continue;
    }

    // Explicit render: at this point the Custom Pass runs and fills the
    // depth RTHandle, so RGB and depth come from the same render.
    cam.targetTexture = rgbRT;
    cam.Render();
    cam.targetTexture = null;

```

```
    if (recordRGB)     SaveRGB();
    if (recordDepth)   SaveDepth();
    if (recordNormals) SaveNormals();
    SavePose();

    frameIndex++;
}
‘‘‘

}
```

Listing C5: `CaptureLoop`: the per-frame capture strategy. The symbol "’ represents omitted code.

SaveDepth

The depth texture is read back and written as a compressed float32 EXR without normalisation or encoding. The stored value is therefore the distance in scene units exactly as rendered. No linearisation or rescaling is required on the Python side.

```
void SaveDepth()
{
EuclideanDepthPass pass = GetEuclideanDepthPass();
if (pass == null)
return;

‘‘‘
if (!ReadbackInto(pass.GetDepthTexture(), depthTex,
                  GraphicsFormat.R32_SFloat, "depth"))
    return;

// float32 + ZIP: without ZIP a 1920x1080 frame takes up 8.3 MB.
byte[] exr = depthTex.EncodeToEXR(
    Texture2D.EXRFlags.OutputAsFloat | Texture2D.EXRFlags.CompressZIP);

File.WriteAllBytes(
    Path.Combine(depthFolder, $"{frameIndex:D7}.exr"), exr);
‘‘‘

}
```

Listing C6: `SaveDepth`: readback and raw float32 encoding of the Euclidean depth map. The symbol "’ represents omitted code.

C.2.3 Ground-truth Depth Validation

`DepthValidationScene.cs` is an editor tool that builds a minimal scene consisting of a single flat wall perpendicular to the optical axis and positioned at a known distance. Its purpose is to confirm unambiguously that the exported depth represents Euclidean distance rather than the camera-space Z coordinate.

If the exported value represented camera-space Z, every pixel of the wall would have the same value because the wall is perpendicular to the optical axis and therefore lies at a constant Z. In contrast, for Euclidean distance, the value increases towards the image borders according to

$$D(u, v) = D_{\text{centre}} \sqrt{1 + x^2 + y^2}.$$

This behaviour depends only on the camera intrinsics, also providing a verification of f_x and f_y. The wall is implemented as a flattened cube rather than a quad so that backface culling cannot make it invisible from the camera side.

```
var wall = GameObject.CreatePrimitive(PrimitiveType.Cube);
wall.transform.localScale = new Vector3(400f, 400f, 0.2f);
wall.transform.position = new Vector3(0f, 0f, WallDistance + 0.1f);

var volGO = new GameObject("Euclidean Depth Volume");
var volume = volGO.AddComponent<CustomPassVolume>();
volume.isGlobal = true;
volume.injectionPoint = CustomPassInjectionPoint.BeforeTransparent;

var pass = volume.AddPassOfType(typeof(EuclideanDepthPass)) as EuclideanDepthPass;
pass.euclideanDepthShader = Shader.Find("Hidden/EuclideanDepthHDRP");
recorder.euclideanDepthVolume = volume;
```

Listing C7: The flat wall and the Euclidean Depth Pass wiring in the validation scene.

After running the tool, the scene is played for approximately one second, and the resulting depth map is checked against the theoretical formula using a Python script.

C.2.4 Trajectory Constraints for Parallax

A photometric depth loss learns depth only from parallax: under pure rotation, the loss is flat with respect to depth. Hand-driven sequences can contain many rotation-only frames because the operator may stop translating while looking around. Two scripts are therefore used to reduce this limitation.

`EndoscopeScriptedPath.cs` produces a deterministic trajectory, advancing by exactly one step per captured frame so that N steps produce N frames and identical waypoints reproduce the same path. A trajectory consists of a list of segments of one of four kinds (Survey, Traverse, GazeSweep and Waypoints), each configured in the Inspector.

Evaluate

`Evaluate` is a pure function mapping a frame index to a camera pose. It identifies the segment corresponding to the given index and dispatches the evaluation to the geometry associated with that segment type. The dispatch itself is ordinary control flow and is shown here only to provide the call context for the following listing.

```
public bool Evaluate(int index, out Vector3 position, out Quaternion rotation)
{
```

```
int cursor = 0;
for (int i = 0; i < segments.Length; i++)
{
var seg = segments[i];
int frames = Mathf.Max(2, seg.frames);

‘‘‘
    if (index >= cursor + frames) { cursor += frames; continue; }

    float t = (index - cursor) / (float)(frames - 1);

    if (seg.kind == SegmentKind.Waypoints)
        return EvaluateWaypoints(seg, t, out position, out rotation);

    Vector3 start = SegmentStart(i);
    Vector3 axis = (seg.to.position - start).normalized;
    Basis(axis, out Vector3 right, out Vector3 up);

    if (seg.kind == SegmentKind.Survey)
        EvaluateSurvey(seg, start, seg.to.position, t, axis, right, up,
                       out position, out rotation);
    // Traverse and GazeSweep follow the same pattern, omitted here.

    return true;
}
position = transform.position; rotation = transform.rotation;
return false;
‘‘‘

}
```

Listing C8: `Evaluate`: frame index to camera pose, dispatched by segment kind. The symbol ”’ represents omitted code.

EvaluateSurvey

This is the concrete mechanism used to promote parallax throughout the trajectory. The camera orbits the segment's axis while simultaneously progressing from start to end, so that the optical centre translates on every frame even while the gaze sweeps between the wall and the lumen. A pan in place would produce changing images without providing the translational parallax required for depth estimation.

```
void EvaluateSurvey(Segment seg, Vector3 start, Vector3 end, float t,
Vector3 axis, Vector3 right, Vector3 up,
out Vector3 position, out Quaternion rotation)
{
float theta = 2f * Mathf.PI * seg.orbitTurns * t;
Vector3 radial = right * Mathf.Cos(theta) + up * Mathf.Sin(theta);

‘‘‘
// The orbit translates the optical centre on every frame.
position = Vector3.Lerp(start, end, t) + radial * seg.orbitRadius;

```

```
// The tilt sweeps rather than sitting at one angle, so a single segment
// shows both the near wall and the far lumen.
float tiltDeg = seg.gazeForwardTilt
    + seg.gazeTiltAmplitude * Mathf.Sin(2f * Mathf.PI * seg.gazeTiltCycles * t);
float tilt = tiltDeg * Mathf.Deg2Rad;
Vector3 gaze = radial * Mathf.Cos(tilt) + axis * Mathf.Sin(tilt);

rotation = Quaternion.LookRotation(gaze.normalized, axis);
'''

}
```

Listing C9: `EvaluateSurvey`: the helical orbit-and-sweep pattern. The symbol "' represents omitted code.

There are other secundary functions present: `EvaluateTraverse`, `EvaluateGazeSweep`, `EvaluateWaypoints`, the segment field definitions, the driver loop, and the editor-only helpers (presets, path preview, dry-run validation and gizmo drawing).

`EndoscopeManualOrbit.cs` preserves manual driving, which provides the ability to decide what gets filmed, while adding a small automatic lateral wobble so that even a stationary hand-driven frame has a baseline. The offset is lateral because a point at depth Z moves by approximately $f_x t/Z$ pixels under sideways translation, producing substantially more image displacement than a forward translation of the same magnitude.

LateUpdate and Offset

The non-obvious part is that the previous frame's offset is subtracted before the new one is added, so that it never accumulates. This composes correctly with `EndoscopeController`, which applies movement as a delta, but not with `EndoscopeScriptedPath`, which sets an absolute position every frame. A guard therefore disables the wobble when a scripted path is driving the camera. The wobble itself follows a Lissajous figure whose non-integer frequency ratio prevents successive cycles from repeatedly tracing the same circle.

```
void LateUpdate()
{
// Undo the previous frame's offset first, so it never accumulates.
if (appliedOffset != Vector3.zero)
{
transform.position -= appliedOffset;
appliedOffset = Vector3.zero;
}

'''
if (recorder == null || ScriptedPathIsDriving)
    return;

if (onlyWhileRecording && !recorder.IsRecording)
{
    step = 0;
    return;
}
```

```
19
20 transform.position += Offset(step);
21 appliedOffset = Offset(step);
22 step++;
23 ‘‘‘
24
25 }
26
27 // Lateral offset in the camera’s own frame.
28 Vector3 Offset(int index)
29 {
30 float theta = 2f * Mathf.PI * index / Mathf.Max(4, framesPerRevolution);
31 return transform.right * (radius * Mathf.Cos(theta))
32 + transform.up * (radius * Mathf.Sin(lissajousRatio * theta));
33 }
```

Listing C10: `LateUpdate` and `Offset`: apply and un-apply the lateral wobble each frame. The symbol ”’ represents omitted code.

Other functions are present, such as the `ScriptedPathIsDriving`, `Start`, `OnDisable` and the `EstimateParallax`.

C.3 Unity Inspector Configuration

This section includes the settings for the objects’ Inspectors in Unity, either the interface characteristics and the parameters exposed by the scripts above that are set through the interface. The scripts were made that way so that the acquisition configuration is reproducible without editing source code. The settings are shown in Figures C1 to **??**.

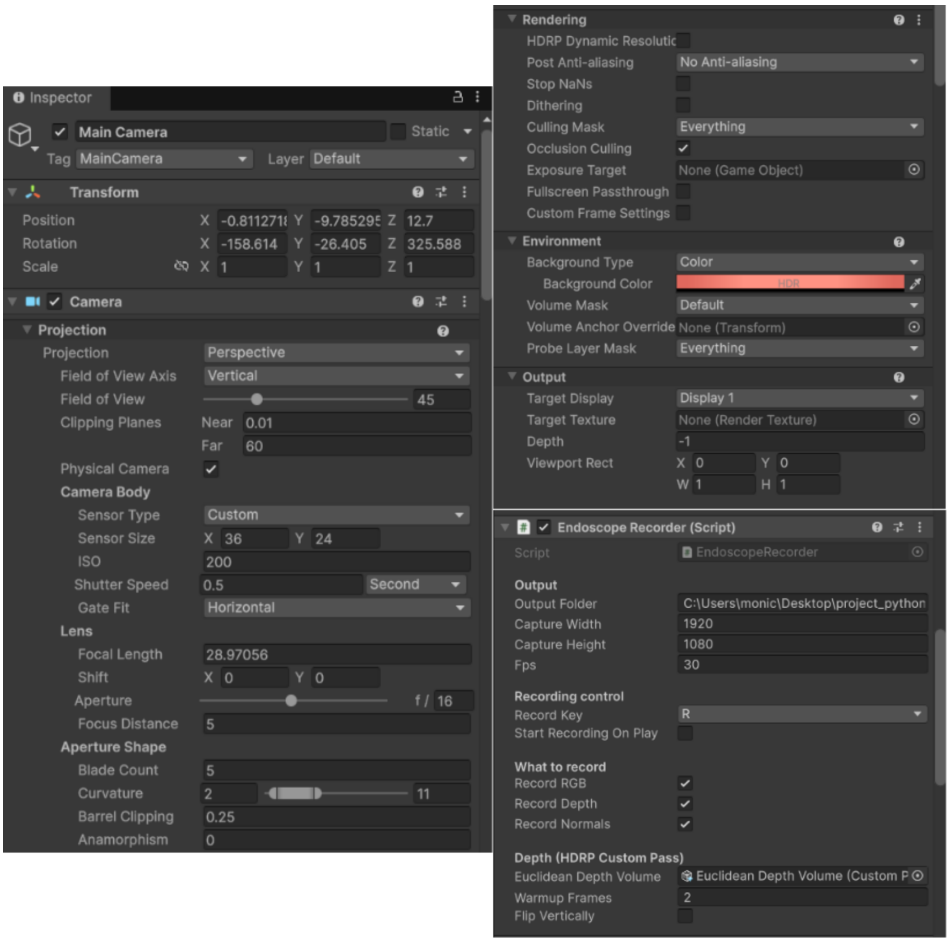


Figure C1: Camera Inspector settings for the object's characteristics.

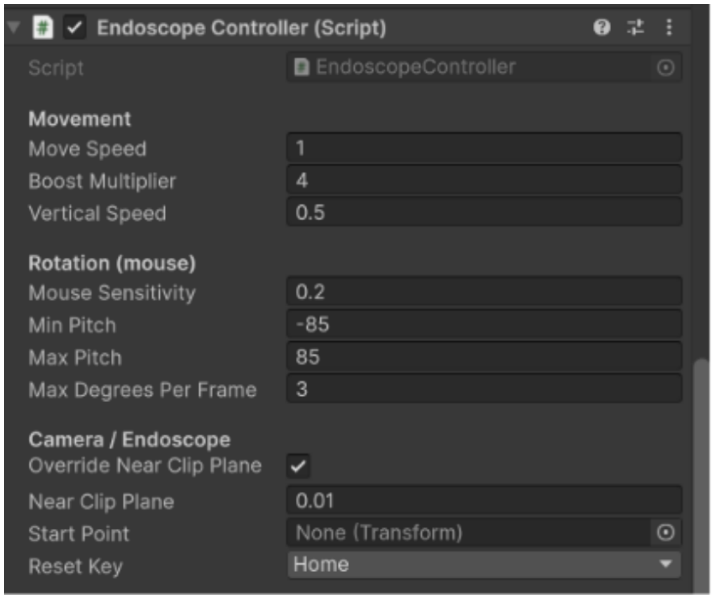


Figure C2: Camera Inspector section connected to the script responsible for the motion of the camera.

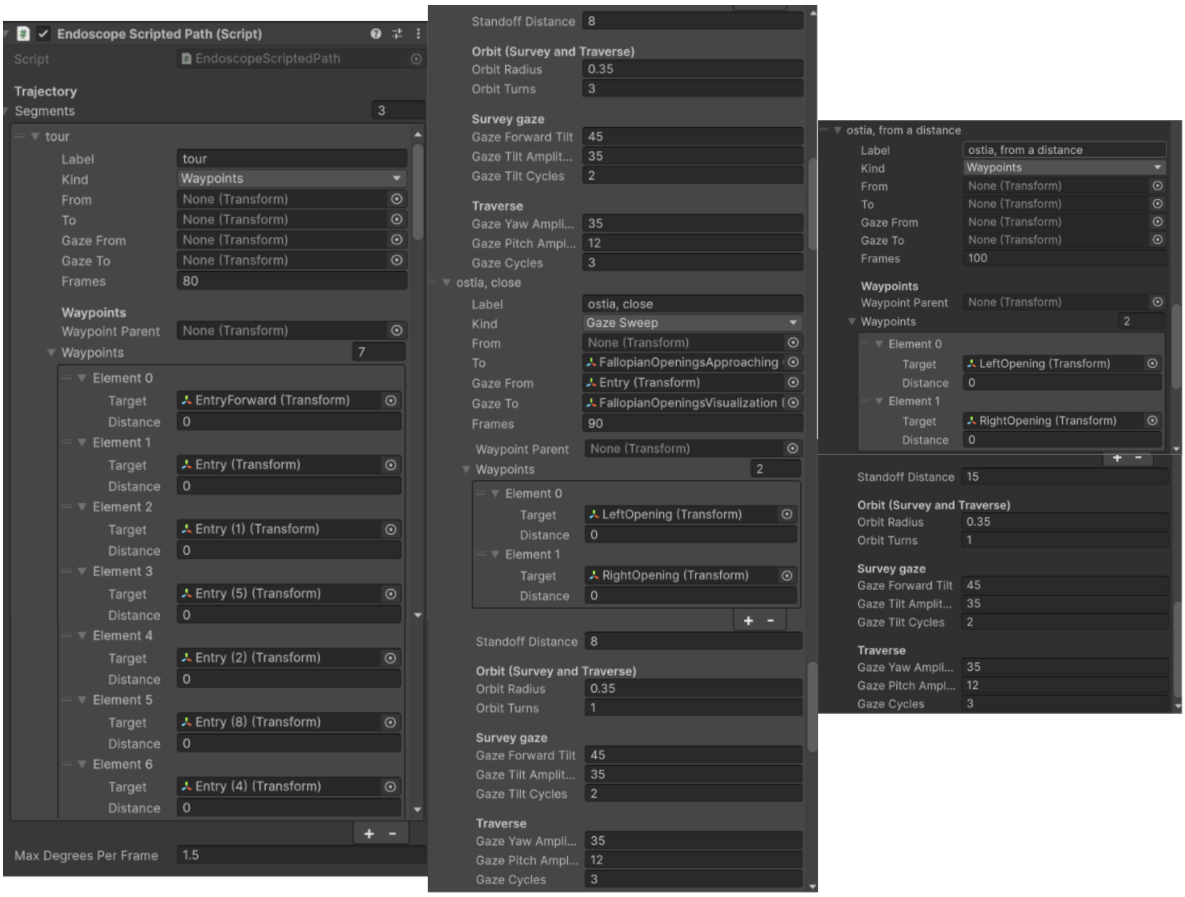


Figure C3: Camera Inspector settings in Unity responsible for determining the placement of the waypoints. The names of the waypoints may not be according to the order due to the change of their place from one recorded scene to another.

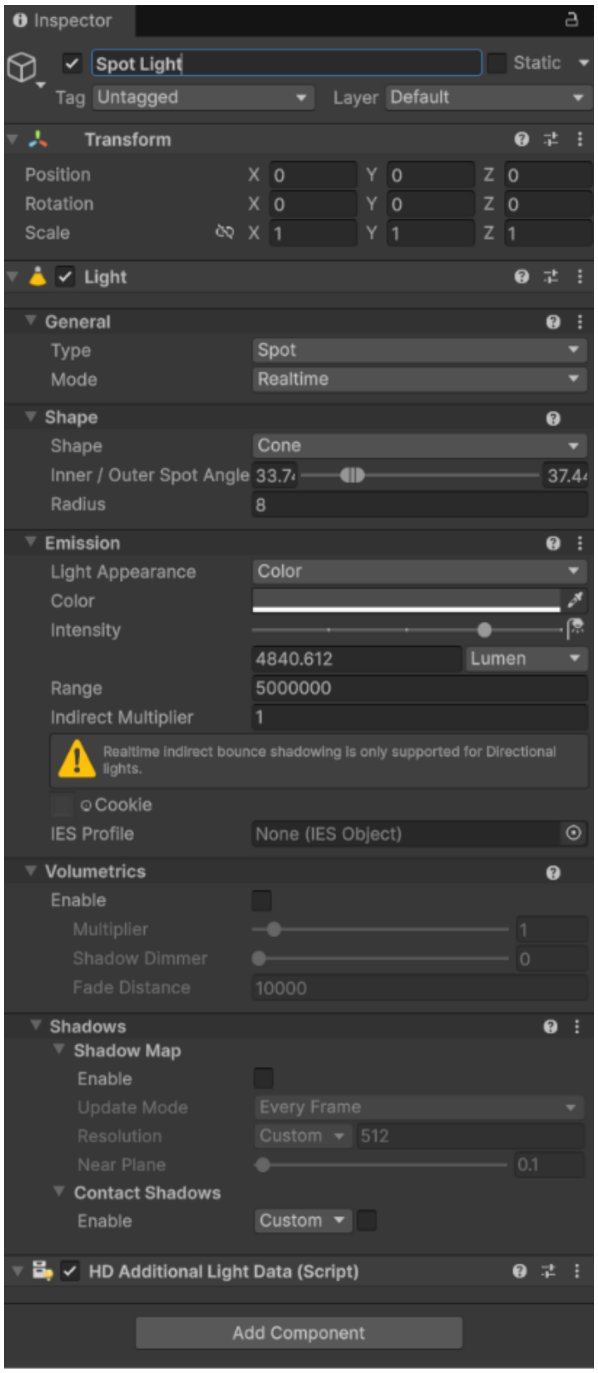


Figure C4: Spot light Inspector settings.

Appendix D

Model Scale - Illustration of the Strategy for Measuring the Meshes

This appendix is related to the model scale analysis to understand the dimensions of the depth predictions in terms of physical measurements. Figure D1 presents the illustration of the strategy to obtain the dimensions of the uterine meshes from Unity, so that those could be compared to the ones from the segment origin, which allowed to obtain the real dimensions, in mm.

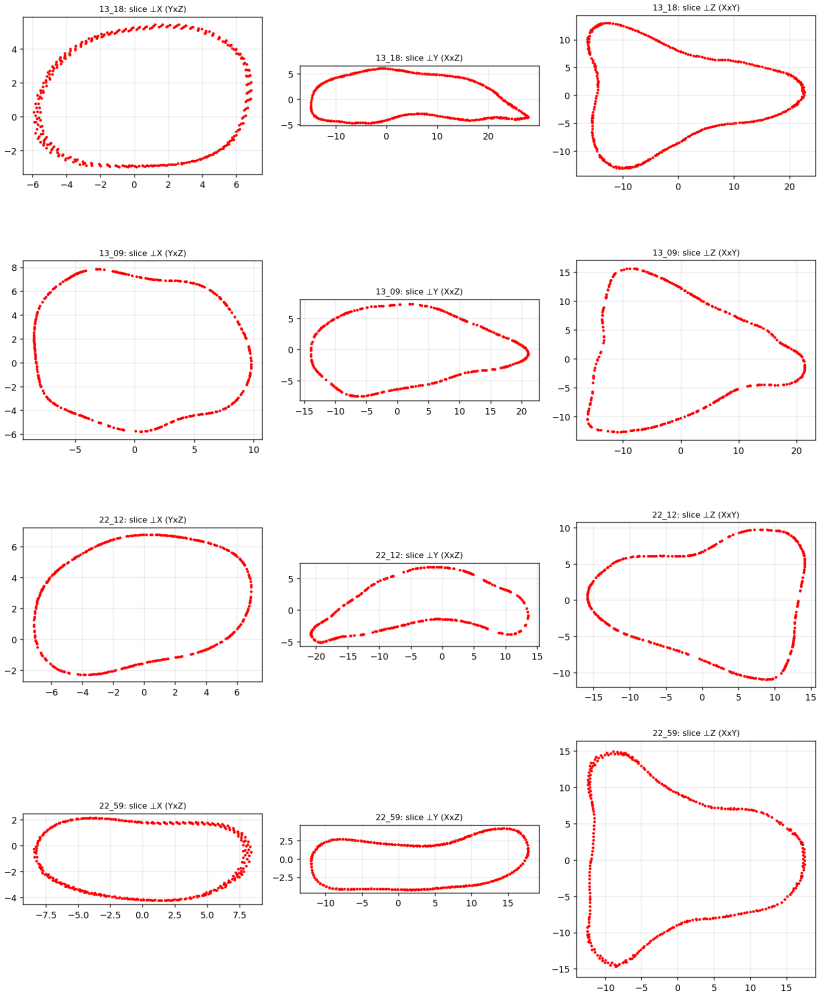


Figure D1: Central cross-sections perpendicular to each principal axis for the four uterine cavity meshes (13_09, 13_18, 22_12 and 22_59), after axis alignment.

Appendix E

Specific Parameter Configuration for Each Recording

This appendix presents the configurations for the recording of the synthetic sequences in Unity. Curvature was changed so that the structure models could involve more diversity. Between recordings, configurations for the camera, spot light and material were changed so that the dataset could be as more diverse as possible. These are presented in Figure E1 and Table E1.

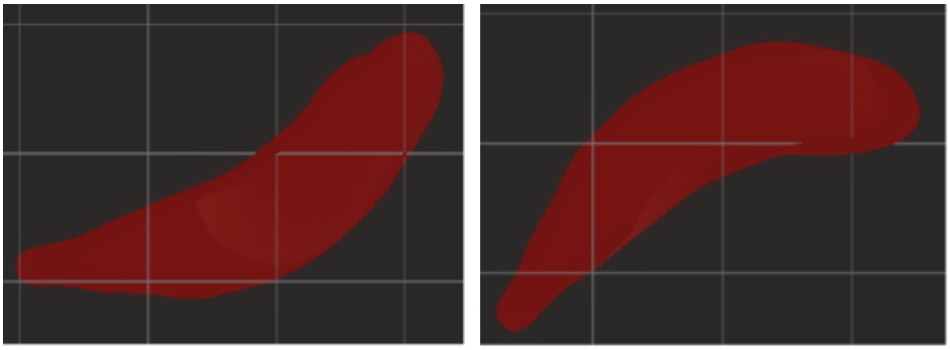


Figure E1: Illustration of the considered curvature orientations for the uterus structures.

Table E1: Parameters used for the generation of the synthetic recordings. Unless otherwise specified, the default configuration was: spot light emission colour = `4C4C4C`, spot light range = 50000, spot light angle = 33.7–37.4°, material base colour = `#FFE6EB` (the most common value across scenes), material metallic property = 0, and material coat mask = 0. "Hybrid" means manual motion designed to preserve parallax, with the trajectory changed between recordings. "Waypoints" means waypoint-based motion using different waypoint layouts between recordings.

Scene	**Movement**	**Camera start**	**Spot intensity**	**Spot angle**	**Emission (intensity/color)**	**Smoothness**	**Other changes**
scene_1	Hybrid	closer to the entry	2235.432	33.7–37.4°	3 / `4A120D`	0.08	–
scene_2	Hybrid	closer to the entry	9218.796	33.7–37.4°	3 / `#4A120D`	0.454	–
scene_3	Hybrid	closer to the entry	2235.432	33.7–37.4°	5 / `#4A120D`	0.661	Coat mask = 1
scene_4	Hybrid	closer to the entry	2235.432	33.7–37.4°	3 / `#4A120D`	0.661	–
scene_5	Hybrid	closer to the middle	2235.432	33.7–37.4°	3 / `#531E0D`	0.661	–
scene_6	Hybrid	closer to the middle	5291.2	33.7–37.4°	off / off	0.661	Spot range = 500
scene_7	Hybrid	closer to the entry	5291.2	33.7–37.4°	3 / `#531E0D`	0	Spot range = 500
scene_8	Hybrid	closer to the entry	2008.216	11.67–14.34°	3 / `#4A120D`	0	Spot range = 500
scene_9	Hybrid	closer to the entry	2235.432	33.7–37.4°	3 / `#4A120D`	0.08	–
scene_10	Hybrid	closer to the entry and surface limit	2724.951	33.7–37.4°	3 / `#4A120D`	0.08	–
scene_11	Waypoints	closer to the entry	2724.951	33.7–37.4°	3 / `#4A120D`	0.08	–
scene_12	Waypoints	closer to the entry	2724.951	33.7–37.4°	3 / `#4A120D`	0.08	–
scene_13	Waypoints	closer to the entry	2724.951	33.7–37.4°	3 / `#531E0D`	0.08	–
scene_14	Waypoints	closer to the entry	3843.784	33.7–37.4°	3 / `#531E0D`	0.08	Spot light colour = `#333333`
scene_15	Waypoints	closer to the entry	1564.177	33.7–37.4°	0.9 / `#4A120D`	0.08	Base colour = `#FFE1BE`
scene_16	Waypoints	closer to the entry	2235.432	33.7–37.4°	0.2 / `#531E0D`	0.192	Base colour = `#FFE1BE`
scene_17	Waypoints	closer to the entry	4840.612	33.7–37.4°	0.2 / `#5C1A14`	0.192	Base colour = `#FFE0E8`

Appendix F

Endo-SfMLearner License

The Endo-SfMLearner code used in this work is distributed under the MIT License, Copyright © 2022 DeepMIA Lab.

The permission notice is as follows: "Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

Appendix G

File Organisation Before and After Adaptation to the Endo-SfMLearner Requirements

This appendix documents the information on the organisation of the folders either as data was collected from Unity and as the Endo-SfMLearner algoritm required. This is presented in Figure G1.

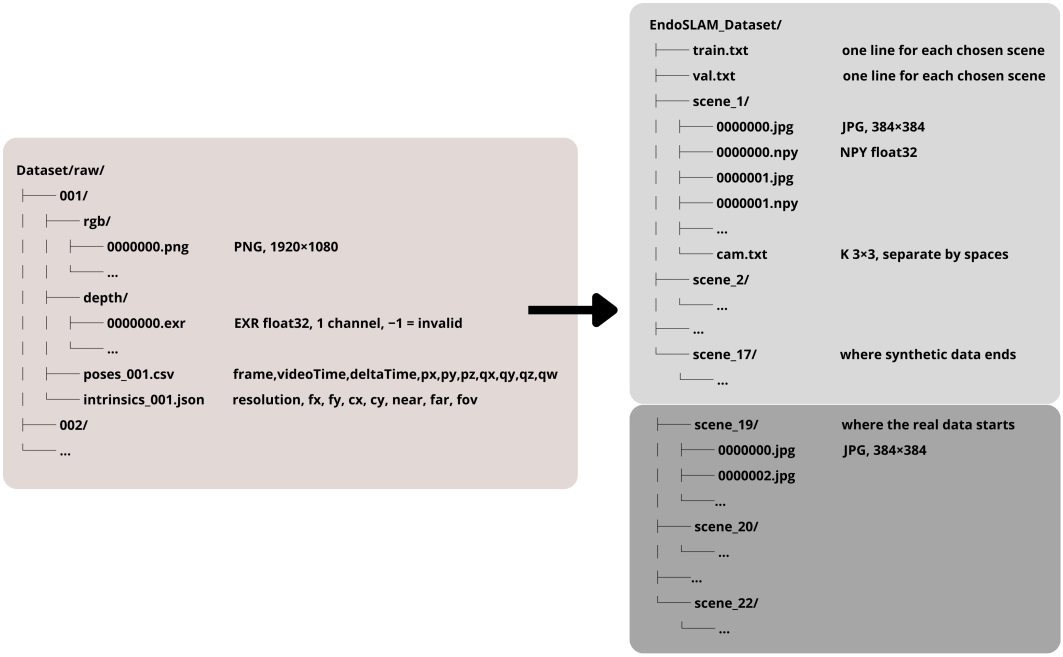


Figure G1: Organisation of the folders used during the preprocessing stage.

Appendix H

Implementation of the Fused Point-Cloud Generation

The obtained fused point clouds are generated using `fuse_pointclouds_fast.py`. For each frame, the depth map is back-projected into the camera coordinate system, transformed into the common world coordinate system using the corresponding camera pose, and accumulated with the points from the remaining frames. A voxel-based downsampling step is then applied to reduce the accumulated representation to one point per occupied voxel. Colour information is not loaded, as it is not required for the geometric measurements performed in this work. To improve computational efficiency, the per-pixel back-projection grid is computed once for each image resolution and camera configuration rather than recomputed for every frame. Argument parsing, path resolution, and PLY file writing are omitted for conciseness. The script was developed with the assistance of a generative AI tool and subsequently reviewed, tested, and modified.

H.1 Camera-Frame Geometry

The relationship between image coordinates and the camera coordinate system requires an explicit treatment of the vertical axis. In particular, the sign of the camera-space Y coordinate is not directly assumed because Unity uses a left-handed coordinate system, whereas image rows increase downwards. This convention is therefore determined empirically by `verify()`, which back-projects pairs of consecutive frames using each candidate axis convention and evaluates how well the resulting points from the same rigid surface coincide in world space. The convention producing the smallest residual is selected. For sequence 001, the selected convention resulted in a median residual of approximately 0.07 scene units, about eight times smaller than that obtained with the next-best alternative. The resulting signs are stored in `AXIS_CONVENTION` and used throughout the subsequent back-projection and fusion steps.

backproject_ cached

This function converts a depth map into 3D points expressed in the camera coordinate system. The ground-truth depth generated by the Unity rendering pipeline represents the Euclidean length of the viewing ray, whereas the monocular network predicts the camera-space Z component. These quantities are not interchangeable: using Euclidean depth directly as Z would incorrectly place off-axis points farther along the camera axis and could, for example, deform a planar surface into a curved shape. For Euclidean depth, the ray length is therefore converted to its corresponding Z component by dividing by

$$\sqrt{1 + x^2 + y^2}. \tag{24}$$

The normalized image coordinates, axis signs and this conversion factor are stored in a grid that is computed once for each image resolution and camera configuration.

```
_GRID = {}

def backproject_cached(depth, K, convention, depth_kind='euclidean', mask_fov=False):
h, w = depth.shape
key = (h, w, K, convention['x_sign'], convention['y_sign'])
if key not in _GRID:
fx, fy, cx, cy = K
ys, xs = np.mgrid[0:h, 0:w]
x, y = (xs - cx) / fx, (ys - cy) / fy
_GRID[key] = dict(sx=convention['x_sign'] * x,
sy=convention['y_sign'] * y,
ray=np.sqrt(1.0 + x**2 + y**2))
g = _GRID[key]

'''
valid = np.isfinite(depth) & (depth > 0) & (depth < MAX_VALID_DEPTH)
z = depth if depth_kind == 'z' else depth / g['ray']
return np.stack([(g['sx'] * z)[valid],
                 (g['sy'] * z)[valid], z[valid]], -1), valid
'''
```

Listing H1: `backproject_cached`: depth map to camera-frame points. The symbol "' represents omitted code.

H.2 World Placement

load_poses

Camera poses recorded during dataset generation are indexed by the original frame number rather than by their position in the preprocessed sequence. This distinction is necessary because preprocessing removes invalid or otherwise discarded frames, meaning that the i-th retained frame does not necessarily correspond to frame i in the original recording. Therefore, the pose files are loaded into a dictionary indexed by the original frame number, ensuring that each depth map is paired with the pose of the corresponding captured frame.

```
def load_poses(raw_sequence_dir):
path = glob.glob(os.path.join(raw_sequence_dir, "poses*.csv"))[0]
table = np.genfromtxt(path, delimiter=',', names=True)
poses = {}
for r in table:
R = quat_to_R(r['qx'], r['qy'], r['qz'], r['qw'])
t = np.array([r['px'], r['py'], r['pz']])
poses[int(r['frame'])] = np.hstack([R, t.reshape(3, 1)])
return poses
```

Listing H2: `load_poses`: recorder poses as 3×4 matrices.

to_world

The available pose sources are converted to a common 3×4 camera-to-world representation when loaded. This provides a single interface for the fusion procedure, avoiding separate transformation branches depending on whether the poses originate from the Unity recorder or from the estimated pose source.

```
def to_world(points_cam, pose):
R, t = pose[:, :3], pose[:, 3]
return points_cam @ R.T + t
```

Listing H3: `to_world`: camera to world coordinates.

H.3 Prediction Scale

estimate_scale

Monocular depth predictions are inherently defined only up to an unknown scale. In the case of the point-cloud fusion, a single scale factor is estimated for the entire sequence rather than independently for each frame. Applying a different factor to each frame would allow each depth map to adopt its own scale, potentially introducing inconsistencies between overlapping surfaces and preventing them from aligning correctly in the fused point cloud.

The sequence-level scale factor is estimated from the median depths over a subset of frames. The variation of the corresponding per-frame scale factors is also used as a measure of scale consistency: a small dispersion indicates that a common scale is appropriate across the sequence, whereas a large dispersion indicates greater frame-to-frame scale variation, which can contribute to thickening or misalignment in the fused point cloud.

```
def estimate_scale(entries, sequence, K, sample=30):
picked = entries[::max(1, len(entries) // sample)][:sample]
ratios, gt_all, pred_all = [], [], []
for frame, pred_path in picked:
gt = euclidean_to_z(load_gt(sequence, frame), K)
pred = np.load(pred_path).astype(np.float32)
v = (gt > 0) & (gt < MAX_VALID_DEPTH) & (pred > 0) & fov_mask(*gt.shape)
ratios.append(np.median(gt[v]) / np.median(pred[v]))
gt_all.append(gt[v]); pred_all.append(pred[v])

'''
scale = float(np.median(np.concatenate(gt_all))
              / np.median(np.concatenate(pred_all)))
spread = np.std(ratios) / np.mean(ratios)
return scale, spread
'''
```

Listing H4: `estimate_scale`: one factor per sequence, from pooled median depths. The symbol ”’ represents omitted code.

H.4 The Fusing Loop

fuse_fast

Each frame is downsampled before being accumulated to limit the number of points handled during the subsequent fusion step, since a full-resolution depth map contains approximately one million pixels. After all frames have been transformed into the common world coordinate system, a second voxel-based downsampling step is applied to the combined point set. The reduction obtained in this second step also provides a useful diagnostic of inter-frame alignment: when consecutive views are correctly aligned, observations of the same surface occupy common voxels and are consequently merged. For fusions using estimated poses, no additional rescaling is applied because the predicted depth and predicted poses are already expressed in the common scale learned by the network.

```
def fuse_fast(source, sequence, output_path, poses_file=None, voxel=VOXEL_SIZE):
K, entries, kind, loader, mask_fov, raw_dir = load_source(source, sequence)
poses = (load_poses_kitti(poses_file, [f for f, _ in entries])
if poses_file else load_poses(raw_dir))


scale = estimate_scale(entries, sequence, K)[0] if source == 'pred' else 1.0

clouds = []
for frame, path in entries:
    depth = loader(path).astype(np.float32) * scale
    pts, _ = backproject_cached(depth, K, AXIS_CONVENTION, kind, mask_fov)
    world = to_world(pts, poses[frame])
    clouds.append(voxel_downsample(world, None, voxel)[0])

points = np.concatenate(clouds)
before = len(points)
points, _ = voxel_downsample(points, None, voxel)
print("%.1fx collapse across frames" % (before / len(points)))
save_ply(output_path, points)
```

Listing H5: `fuse_fast`: per-frame downsample, then one cross-frame merge.

voxel_downsample

This function retains one point for each occupied voxel, keeping the first point encountered within each voxel. To reduce the computational cost of identifying occupied voxels, the three integer voxel indices are packed into a single `int64` key. This avoids the substantially higher cost of applying `np.unique` directly along the rows of a two-dimensional array, which is particularly relevant because the operation is performed once for every frame.

```
def voxel_downsample(points, colours, voxel):
keys = np.floor(points / voxel).astype(np.int64)
keys -= keys.min(0)
dims = keys.max(0) + 1
packed = (keys[:, 0] * dims[1] + keys[:, 1]) * dims[2] + keys[:, 2]
_, keep = np.unique(packed, return_index=True)
```

```
keep.sort()
return points[keep], (colours[keep] if colours is not None else None)
```

Listing H6: `voxel_downsample`: one point per voxel via a packed integer key.

H.5 Verification

check_ against_ original

The cached back-projection grid is verified against a direct implementation in which the normalized coordinates and conversion factors are recomputed for each frame. Both implementations are applied to a sample of frames, and the largest coordinate difference between the resulting point sets is recorded. The difference is not exactly zero because the two implementations involve slightly different floating-point operations, but it remains around 10^{-4} scene units, substantially below the 0.05-unit voxel size. This confirms that caching the back-projection grid does not introduce a meaningful numerical change in the generated point coordinates.

```
def check_against_original(source, sequence, frames=8):
K, entries, kind, loader, mask_fov, _ = load_source(source, sequence)
worst = 0.0
for frame, path in entries[::len(entries) // frames][:frames]:
depth = loader(path).astype(np.float32)
a, va = backproject(depth, K, AXIS_CONVENTION, kind, mask_fov)
b, vb = backproject_cached(depth, K, AXIS_CONVENTION, kind, mask_fov)
assert a.shape == b.shape and np.array_equal(va, vb)
worst = max(worst, np.abs(a - b).max())
print("largest difference %.3e scene units" % worst)
```

Listing H7: `check_against_original`: cached grid versus direct recomputation.

The resulting point cloud is stored in scene units.

Appendix I

Scale Aligment Analysis

This appendix explores the depth estimation results in terms of the need for the scaling factor to be applied to them, either per frame, when each configuration is evaluated and compared with the GT of that frame, or global, when there is the aim to obtain fused point clouds of entire sequences.

This study is presented for one case scenario, included in Figure I1. It presents a visualisation of the same real image side-by-side with the 3D representation of the corresponding GT and the results it obtained without fine-tuning and in each model configuration.

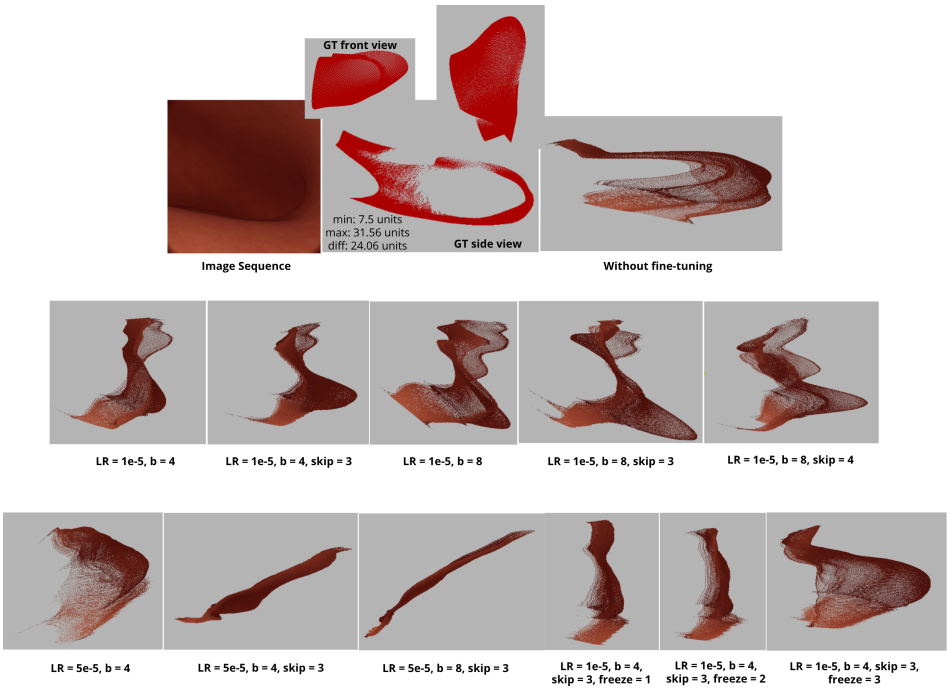


Figure I1: Analysis of the depth prediction through per-frame depth converted into point clouds for a single frame across all runs and including the GT, presented in Unity units. "LR" means "learning rate" and "b" means batch.

Even among the configurations that capture the general spatial structure of the scene, differences can be observed in the magnitude and distribution of the predicted depth values, presented numerically in Table I1.

Table I1: Predicted depth range for frame 20 of scene_15 across training configurations in terms of the per-pixel minimum and maximum of the predicted depth map before the scaling factor, including their difference, all in the network units.

Configuration	**Min (network units)**	**Max (network units)**	**Max – Min (network units)**
Without fine-tuning	18.18	58.54	40.36
LR = 1e-5, b = 4	29.83	48.24	18.41
LR = 1e-5, b = 4, skip = 3	25.37	47.18	21.81
LR = 1e-5, b = 8	29.57	50.57	21.00
LR = 1e-5, b = 8, skip = 3	24.79	49.83	25.04
LR = 1e-5, b = 8, skip = 4	24.31	48.30	23.99
LR = 5e-5, b = 4	23.64	56.59	32.95
LR = 5e-5, b = 4, skip = 3	29.44	61.16	31.72
LR = 5e-5, b = 8, skip = 3	27.60	57.56	29.96
LR = 1e-5, b = 4, skip = 3, freeze = 1	32.75	42.17	9.42
LR = 1e-5, b = 4, skip = 3, freeze = 2	33.57	45.23	11.66
LR = 1e-5, b = 4, skip = 3, freeze = 3	28.03	52.46	24.43

Values cannot be compared directly, as the ranges for the predictions are expressed in network units while the ground-truth one is expressing in Unity units. A model may learn where depth changes occur without accurately reproducing the magnitude of those changes. This means that the predicted depth of any configuration is defined only up to an unknown factor, and a scale alignment is needed.

This distinction is particularly important in the present application, since preserving the relative relief of the scene is relevant for subsequent 3D reconstruction.

It is important to notice that this analysis considers only a single frame, so it should not be regarded as representative of the overall behaviour observed in the quantitative evaluation presented previously. It only provides a more detailed qualitative and frame-level examination of the predictions.