FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO


# Deep Homography for Endoscopic Capsule Frames Localisation

**Sara Garrido Gomes**

DISSERTATION FOR THE DEGREE OF MASTER IN BIOENGINEERING

MSC IN BIOENGINEERING - BIOMEDICAL ENGINEERING

Supervisor: António Manuel Trigueiros da Silva Cunha

Co-Supervisor: Hélder Filipe Pinto de Oliveira

July 25, 2019

# Deep Homography for Endoscopic Capsule Frames Localisation

**Sara Garrido Gomes**

MSc in Bioengineering - Biomedical Engineering

July 25, 2019

# Resumo

Cápsulas endoscópicas (VCE) são um avanço recente na tecnologia de endoscopia gastrointestinal, permitindo o exame do trato gastrointestinal, para deteção de várias patologias (como doença de Crohn, lesões neoplásticas, doença celíaca, etc). O uso de cápsulas permite a análise de todo o trato gastrointestinal, incluindo o intestino delgado, minimizando, em simultâneo, o desconforto para o paciente. Contudo, a endoscopia por cápsula traz-nos o problema de saber onde a câmara se encontra num dado momento: uma informação necessária para localizar as lesões encontradas e tomar as medidas adequadas. Completar visualmente esta tarefa é um processo desafiante, dada a falta de pontos de referência no trato gastrointestinal. Foi neste contexto que métodos baseados em *software* foram desenvolvidos, usando tanto técnicas clássicas de *multi-view stereo* como de *deep learning*. No entanto, estes sistemas falham pela sua falta de acuidade e impossibilidade de serem implementados num contexto clínico. Assim sendo, a presente dissertação pretende desenvolver uma técnica inovadora de localização de cápsulas endoscópicas, com base na estimativa de homografia entre imagens consecutivas de endoscopia do intestino delgado.

O sistema desenvolvido é constituido por três passos principais, usa duas imagens endoscópicas consecutivas como *input*, e tem como resultado o deslocamento, em milímetros, da cápsula entre elas. Primeiro, as imagens foram pré-processadas, com uma técnica de difusão anisotrópica, para as homogeneizar, reduzindo a iluminação. Os *frames* foram então passados à rede neuronal convolucional, treinada de forma não-supervisionada, para estimar a homografia entre as imagens. O passo final foi usar esta estimativa para calcular o deslocamento entre as duas imagens, e aplicar um passo de pós-processamento, para substituir valores absurdos pelo deslocamento médio do vídeo.

Para testar o sistema, um método de gerção de dados artificial foi usado, para obter pares de imagens de treino com homografia real associada. Estes pares foram passados à rede, obtendo uma estimativa da homografia com erro médio de canto (MACE) de 21 pixels. Apesar do estado da arte mostrar resultados superiores (MACE$=$ 2 pixels), técnicas anteriores usam redes supervisionadas, implicando o uso de *ground truth* para treinar a rede. O sistema desenvolvido prova ser mais útil em aplicações reais, onde não há *ground truth* disponível. O deslocamento e pós-processamento resultaram num deslocamento total estimado, correspondente ao comprimento do intestino delgado, entre $5,223$ e $6,734$ milímetros, nos vídeos testados. Estes valores estão dentro dos valores normais de comprimento intestinal.

É possível concluir que, apesar das técnicas não supervisionadas, como a desenvolvida, não terem ainda capacidade de ultrapassar os métodos com supervisão, a localização de cápuslas endoscópicas é um problema que deverá seguir no sentido de não necessitar de imagens com *ground truth* associado. A natureza da técnica e do órgão torna quase impossível obter *gorund truth* para as imagens, sendo apenas possível se os vídeos forem gravados, por exemplo, em modelos cirúrgicos. A presente dissertação prova que técnicas não supervisionadas são não só possíveis, como também eficientes, e merecedoras de mais investigação e desenvolvimento.

# Abstract

Video capsule endoscopy (VCE) is a recent advancement in gastrointestinal endoscopy technology, allowing the examination of the gastrointestinal tract, for the detection of various pathologies (such as Crohn's disease, neoplastic lesions, celiac disease, etc). The use of the capsules allows the analysis of the entire gastrointestinal tract, including the small bowel, while minimising the discomfort to the patient. However, capsule endoscopy brings us the issue of knowing where the camera is at a given time: a necessary information to locate the lesions found and take the necessary measures. This is a very challenging task to perform visually, due to the lack of landmarks within the gastrointestinal tract. It was in these conditions that software-based methods were developed, both using classical multi-view stereo and deep learning techniques. The present dissertation seeks to develop a novel capsule localisation system, based on homography estimation between consecutive VCE small bowel frames.

The developed system consists of three main steps, uses as input two consecutive images, and outputs the capsule displacement between them, in millimetres. First, the images were preprocessed using an anisotropic diffusion technique, to homogenise them, by reducing the illumination. The frames were then fed into a convolutional, trained in a completely unsupervised manner, to estimate the homography between them. The final step was to use the homography estimation to compute the displacement between the images, and perform a post-processing step to replace nonsensical displacement values by the average displacement of the video.

To test the system, an artificial data generation technique was used, to provide training pairs with a ground truth homography. These pairs were fed into the homography estimation network, achieving an homography estimation with mean average corner error (MACE) of 21 pixels. Although the state of the art shows better results (MACE= 2 pixels), it uses a supervised network, demanding the use of ground truth labels for the training of the network. The present method reveals itself as more useful in real world applications, where no ground truth is available. The displacement and post-processing provided a total displacement estimation, corresponding to the small bowel length, from $5,223$ to $6,734$ millimetres, in the tested VCE. These values are within the normal range for small bowel length.

It is possible to conclude that, although unsupervised techniques, as the one developed, are still not able to outperform its supervised counterparts, the endoscopic capsule localisation task should move towards not needing labelled images. The nature of the technique and of the organ makes ground truth labels very hard to obtain, only being possible when using surgical models to record the videos. The present work serves as proof that unsupervised techniques are not only possible, but effective, and deserving of further investigation and development.

# Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus orientadores, professores António Cunha e Hélder Oliveira, por todo o apoio dado durante o semestre. Os conselhos de trabalho foram tão importantes como as sugestões de descanso para o realizar desta dissertação. Pelo tempo que dispendeu para contribuir com o seu conhecimento prático para este projeto, estendo o obrigada à doutora Marta Salgado, do Centro Hospitalar do Porto.

Obrigada a todos aqueles que cruzaram o meu caminho nestes últimos cinco anos. Obrigada aos Reis pela companhia, aos Templários pela ajuda, aos Abades pelo exemplo, aos Mosqueteiros pela diversão e aos Velhinhos pela experiência. Obrigada a todos que ainda por cá vão ficar, e boa sorte.

A todos os amigos, velhos e novos, cá e lá. Obrigada pelo gozo, pela descontração e por serem a melhor fonte de energia no final de um longo, longo dia. Terei sempre o privilégio de dizer que vos conheci e espero nunca perder o direito de vos chamar amigos.

E como Roma não se construiu num dia, queria terminar com aqueles que estão lá desde o primeiro. Obrigada à minha família que no meio de brincadeira e discussões, riso e lágrimas, sempre me forçou a dar o meu melhor, seja qual for o desafio a enfrentar.

Sara Garrido Gomes

*"I think we've outgrown full-time education...*
*Time to test our talents in the real world, d'you reckon?"*

J.K. Rowling

# Contents

# List of Figures

# List of Tables

# Abbreviations and Acronyms

| | |
|---|---|
| Adam | Adaptive Moment Estimation |
| CE | Capsule Endoscopy |
| CLAHE | Contrast-Limited Adaptive Histogram Equalisation |
| CNN | Convolutional Neural Network |
| DFT | Direct Fourier Transform |
| DLT | Direct Linear Transform |
| DoF | Degrees of Freedom |
| DoG | Difference of Gaussians |
| DS | Differentiable Sampling |
| DWT | Direct Wavelet Transform |
| FFNN | Feed Forward Neural Network |
| GIT | Gastrointestinal Tract |
| kNN | k-Nearest Neighbours |
| LMS | Least Median of Squares |
| LoG | Laplacian of Gaussian |
| LSTM | Long Short Term Memory |
| MACE | Mean Average Corner Error |
| MOPS | Multi-scale Oriented Patches |
| MVS | Multi-View Stereo |
| NN | Neural Network |
| OGIB | Obscure Gastrointestinal Bleeding |
| PM diffusion | Perona-Malik diffusion |
| PSGG | Parameterised Sampling Grid Generator |
| RANSAC | Random Sample Consensus |
| RCNN | Recurrent Convolutional Neural Network |
| ReLU | Rectified Linear Unit |
| RMSE | Root Mean Squared Error |
| RNN | Recurrent Neural Network |
| SfS | Shape from Shading |
| SIFT | Scale-Invariant Feature Transform |
| SURF | Speeded Up Robust Features |
| TDLT | Tensor Direct Linear Transform |
| VCE | Video Capsule Endoscopy |

# Chapter 1

# Introduction

## 1.1 Context

Gastrointestinal endoscopy is a medical technique used to examine the gastrointestinal tract (GIT), for the detection of pathologies such as Crohn's disease, neoplastic lesions, celiac disease and obscure gastrointestinal bleeding (OGIB). This procedure is performed since the 1960s, using flexible endoscopes [57]. However, in 2000, capsule endoscopy (CE) was introduced, and it has been widely used in clinical practice ever since [27]. These capsules are vitamin-sized, and capture the interior of the intestinal tract through the use of a camera and small light installed within [57].

The use of the capsule proves to be more advantageous than the traditional method, because it can reach the entire small bowel, a region of the GIT that was out of bounds to the traditional methods [57], and it provides a more comfortable experience for the patient.

The capsules provide an 8 to 10 hour video, that is transmitted through a recorder (radio-frequency transmitter), and can be later analysed by a doctor. This analysis can be tedious and prone to mistakes, due to the high amount of data and its poor quality. It should take from 45 to 90 minutes [16].

When performing capsule endoscopy, one key aspect is the ability to know where the camera is, in order to understand the location of the lesions found, and ensure the efficacy of further interventions, such as tissue biopsies [57]. The doctor may perform this localisation visually, but due to the nature of the organ, this task becomes very difficult, since few landmarks can be used for guidance. The low quality of the images can also be a factor: a low frame rate means that there may be no overlap in consecutive frames, and the existence of air bubbles and other artefacts in the image obstructs the view of the small bowel walls.

There are, however, some solutions to track the position of an endoscopic capsule within the human body, using additional hardware, both within the capsule and externally attached to the patient. These techniques use electromagnetic waves or the strength of the magnetic field to determine the position of the capsule [57].

In terms of electromagnetic waves, the most explored option is the radio frequency waves, already widely used in the tracking of objects. However, when applied to the GIT, some limitations

are imposed, since low-frequency waves are not capable of reaching the precision required for this application [57]. Additionally, we can consider the use of continuous X-ray images to track the capsule in real time. This method is useful for capsules that have an integrated steering system, making it possible for the physician to direct the capsule in the desired direction. It has, however, the disadvantage to be a solely visual method, without providing any actual parameters of position and orientation of the capsule at a given moment [57].

The magnetic field strength methods are explored not only by those interested in the localisation of the capsule, but also those intending to include an actuator in it, in order to guide the capsule within the GIT [57]. Although they can provide a high accuracy, these systems always have the need to take extra space within the capsule (making it bigger) and can interfere with the actuation system, if implemented [57].

Although these are only a few examples of the physical methods, their common trait is the need to use external equipment to perform the localisation [16]. This means that the patient will have to endure not only the discomfort of the procedure itself (swallow the capsule and carry the transmitter for the designated time period), but they will have to make additional sacrifices, such as swallowing a bigger capsule, carry a second device for localisation or even remain in the hospital while the exam is taking place (in case of continuous X-ray).

## 1.2 Motivation

Given the downsides that the current methods of capsule localisation present to the patients, the development of new techniques becomes imperative. That is why completely software-based methods are starting to be explored. The software techniques use either a topographic video segmentation approach or a motion estimation approach [16]. Topographic video segmentation is a process through which a series of frames is divided into consecutive segments corresponding to different parts of the GIT. When applying a motion estimation approach, the main goal is to compute the displacement and rotation of the capsule between two consecutive frames, and thus be able to recreate its trajectory through the body. These methods are mainly feature based, using mostly colour and texture [16]. However, as mentioned before, the quality of the images obtained through capsule endoscopy is low, making it very difficult to detect distinct features, and even harder to detect common features between two consecutive images (necessary for motion estimation). Even so, these techniques present advantages when compared to the physical methods, but they still are not able to reach a satisfying degree of accuracy [16], [23].

Additionally, there are some works that present deep learning techniques for the visual odometry of the capsule, dispensing the use of feature-based methods, as seen, for example in Turan et al. [61]. The use of these techniques seems to improve upon the works seen before, showing great potential for future exploration. Deep learning has also been used for other applications that require the computation of homography between two images [38], providing techniques that could it be useful for this particular application. However, these systems require a large amount of labelled endoscopic images for training and testing, which can be very difficult to obtain.

## 1.3 Objectives

Given the limitations of the methods presented and the ever-growing exploration of deep learning techniques for image analysis, the main objective of the present dissertation is to use machine learning techniques, particularly deep learning, to estimate capsule progression between consecutive video capsule endoscopy (VCE) frames in the small intestine in a completely unsupervised manner.

The system should be based on deep learning techniques, given the potential shown in other works, and used to compute visual odometry on the videos produced by endoscopic capsules. The use of these techniques should be able to improve the accuracy and ease of use of other existing solutions, and create a tool suitable for clinical utilisation in the future. Ideally, the system should not require supervision, meaning that it will not need to be trained using a labelled database.

The computation of the localisation of the capsule should rely only on the images the capsule itself provides, avoiding the implementation of any additional hardware, both external or within the capsule, in order to maximise patient comfort and maintain the capsule specifications, concerning size and battery life.

Furthermore, the work will be preceded by the collection of the state-of-the-art techniques used not only in the endoscopic capsule area, but also in the general field of visual odometry. This will allow a comprehensive understanding of the existing techniques, enabling the creation of an innovative and efficient new method.

## 1.4 Contributions

The present dissertation provided the following contributions to its field of study:

- Creation of an unsupervised system of homography estimation for VCE images;
- Assessment of the impact of elapsed time between frames in the estimation of the homography matrix;
- Computation of per frame displacement, based on homography matrix between frames;
- Computation of total capsule displacement in any given VCE frame.

This thesis also resulted in the submission and acceptance of a conference paper:

- S. Gomes, M. T. Valério, M. Salgado, H. P. Oliveira, and A. Cunha. Unsupervised Neural Network for Homography Estimation in Capsule Endoscopy Frames, 2019 (Accepted)

## 1.5 Document Structure

The present dissertation is comprised of 5 chapters. First, Chapter 1 will present a small introduction to the problem at hand, focusing mainly on the motivations for the work and the objectives it intends to fulfil.

On Chapter 2, the main background knowledge required for the interpretation of the dissertation will be presented. The theoretical and mathematical basis for the work are explained both regarding homography estimation, and the functioning of neural networks. Additionally, some of the main methods used for multi-view stereo are presented, including classical approaches of feature detection and matching, as well as several deep learning systems.

Chapter 3 will move to the system developed, exploring the problem at hand and the steps needed to solve it, from the proposed pre-processing techniques and data generation method, to the architecture of the used neural networks and the process to convert this homography into displacement values.

Moving on to Chapter 4, the results of each step of the framework are presented, along with some discussion about their meaning and possible future work to solve the problems that arose in the process. A deeper analysis of the dataset used will also be conducted.

Finally, Chapter 5, the final considerations about the work are made, discussing if the main objectives were met, and which steps should be taken in the future to improve upon the system developed.

# Chapter 2

# Literature Review

In this Section, the main techniques applied to problems similar to the one at hand will be presented, along with some theoretical knowledge needed to fully understand how these techniques work. Section 2.1 will begin by exploring the anatomy of the small bowel and some relevant characteristics of the GIT. Section 2.2 will address the main classical methods for multi-view stereo, focusing on feature-based methods, and addressing their main steps and important considerations. Next, in Section 2.3, a new generation of methods for multi-view stereo, that use deep learning techniques, is addressed, followed, in Section 2.4, by deep learning techniques applied to the visual odometry of endoscopic capsules. Lastly, Section 2.5 will explore the main databases available for problems concerning endoscopic image analysis.

## 2.1 Background on Small Bowel Anatomy

The digestive system plays an integral role in human function, since it is the main responsible for nutrient breakdown and absorption. These tasks are performed by two groups of organs, all belonging to the digestive system: the alimentary canal organs, and accessory digestive organs (see Figure 2.1). The accessory digestive organs include the tongue, liver, and pancreas, and their function is to augment the function of the alimentary canal organs, providing secretions or mechanical action essential to the digestive activity. On the other hand, we can consider the GIT, or alimentary canal, in which the absorption of nutrients will take place. The GIT is, in essence, a tube, spanning from the mouth to the anus, constituted by: the pharynx, oesophagus, stomach, small intestine, and large intestine. In total, the alimentary canal is around 7.5 meters long, although it can vary greatly [39].

Of all the organs belonging to the digestive system, the small intestine, or small bowel, could be considered the most important one: it is where the majority of digestion occurs, and where nutrients can be absorbed into the bloodstream [39]. The small intestine alone constitutes about two thirds of the GIT, making it approximately 3 to 4 meters long, although this number is highly variable from person to person [13, 7].

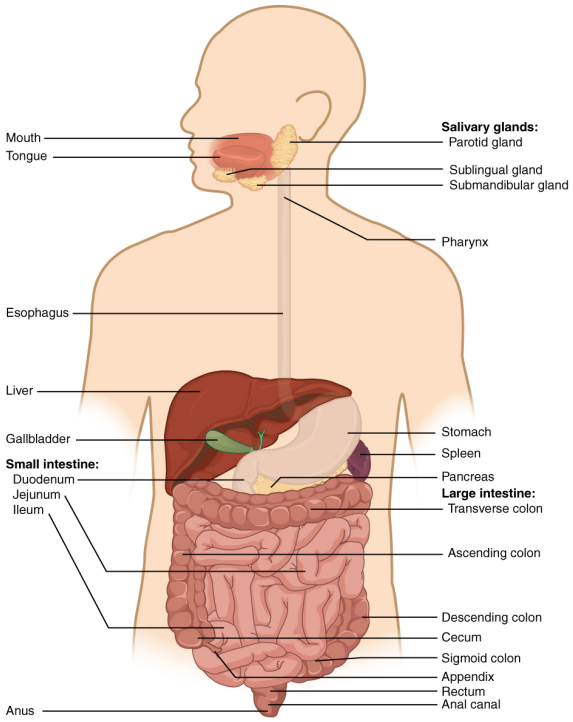



Figure 2.1: Schematic representation of the digestive system, along with its main anatomical structures.

The small intestine can be further divided into 3 regions (see Figure 2.1): the duodenum, jejunum, and ileum [37]. The duodenum is the shortest portion, and begins at the pyloric sphincter, which connects the stomach and the small bowel. The jejunum connects to both the duodenum and the ileum, but no clear demarcation exists between the different structures. Finally, the ileum, the longest part of the small bowel, will connect to the large intestine at the ileocecal sphincter [39].

The interior of the small intestine (as well as the remaining GIT) is covered by the mucous membrane, or mucosa, which is responsible for both secreting fluid into the organ, and, through a thin layer of muscle, pulling the small intestine into folds. These folds, along with hairlike projections of the mucosa onto the interior of the organ (villi or microvilli), are responsible for the $200m^2$ of surface area of the small bowel, in spite of its small 2.5 centimetres diameter [13]. The existence of such structures allow for an optimised nutrient absorption throughout the entire organ [39].

The digestive activity that takes place within the small bowel is not limited to the chemical action of its secretions. In addition, intestinal muscles are responsible for peristaltic movements, where rings of muscle alternately contract and relax, at a rate of 8 to 12 times per minute, depending on the location. The main goal of these movements is not to push the gastrointestinal content forward, but rather combining it with the existing secretions, and pushing it against the mucosa, ensuring proper digestion and absorption [39].

The small bowel is a key component of the digestive system, without which the human body

can not function. With such a role, it is of the utmost importance that small bowel abnormalities are found and treated as fast and efficiently as possible. However, the very nature of the organ, with various kinds of folds and secretions, along with its position in the abdomen and natural movement, provide a natural barrier to diagnosis methods applied to the remaining digestive organs.

## 2.2 Classic Multi-view Stereo

Given a 3D object in the real world, that can be represented by a set of points, its projection onto an image plane produces a set of 2D points, that depend on the position of the camera or viewpoint [51]. Multi-view stereo (MVS) is the term used for a group of techniques that use stereo correspondence to extract 3D geometry from 2D images, using multiple images [10], allowing the reconstruction of the scene. The techniques used for such task are evolving rapidly throughout the years, enabling high-quality results [26, 49]. These algorithms generally consist of the following steps [46]:

- Feature detection and matching (section 2.2.1);
- Structure and motion recovery (section 2.2.2);
- Stereo mapping;
- Modelling.

In this work, the focus will be in the first two steps presented, since stereo mapping and modelling focus on the 3D reconstruction of the observed scene, a discipline that exceeds the goals of the present dissertation.

### 2.2.1 Feature Detection and Matching

The first crucial step, as seen above, is the feature detection and matching. In this step, we first compute local features (specific patterns which are unique from their immediately close pixels) of each image. These features may be corners, edges, regions, among others, and will then be converted into numerical descriptors, representing unique and compact summarisation of these local features.

#### 2.2.1.1 Feature Detectors and Descriptors

An ideal local feature, and consequently an ideal feature detector, must have some key qualities, in order to assure the effectiveness of the entire system. The features should have distinctiveness, locality, quantity, accuracy, efficiency, repeatability, invariance, and robustness. The most important quality, generally speaking, is repeatability: given two frames of the same object with different viewpoints, a high percentage of the detected features from the overlapped visible part should be found in both frames. However, the different qualities depend on each other, and often compromises need to be made [47].

Table 2.1: Summary of the performance of dominant features detection algorithms, as seen on Salahat and Qasaimeh [47].

| Features Detector | Category | Invariance | | | Qualities | | | |
|---|---|---|---|---|---|---|---|---|
| | | Rotation | Scale | Affine | Repeatability | Localisation | Robustness | Efficiency |
| Harris | Corner based | X | | | +++ | +++ | +++ | ++ |
| Hessian | Blob (interest point) | X | | | ++ | ++ | ++ | + |
| SUSAN | Corner based | X | | | ++ | ++ | ++ | +++ |
| Harris-Laplace | Corner based | X | X | | +++ | +++ | ++ | + |
| Hessian-Laplace | Blob (interest point) | X | X | | +++ | +++ | +++ | + |
| DoG | Blob (interest point) | X | X | | ++ | ++ | ++ | ++ |
| Salient Regions | Blob (interest region) | X | X | X | + | + | ++ | + |
| SURF | Blob (interest point) | X | X | | ++ | +++ | ++ | +++ |
| SIFT | Blob (interest point) | X | X | | ++ | +++ | +++ | ++ |
| MSER | Blob (interest region) | X | X | X | +++ | +++ | ++ | +++ |

The most important local features are edges, corners, and regions. Edges refer to pixel patterns at which the intensities abruptly change (strong gradient magnitude). Corners are points at which two or more edges intersect in the local neighbourhood. Finally, regions are a closed set of connected points with a similar homogeneity criterion (usually, an intensity value) [47].

In Table 2.1, a summary of the most used feature detectors is exposed, as seen on Salahat and Qasaimeh [47]. Some of the algorithms presented in the table have been further improved (like the Harris or the SIFT algorithms), giving origin to new derivatives of the original methods. In the next sections, we will focus on the Harris, Speeded Up Robust Features (SURF), Scale Invariant Feature Transform (SIFT) and Multi-scale Oriented Patches (MOPS) algorithms.

### Harris Corner Detector

The Harris corner detector is an algorithm that provides good repeatability, and robustness to changes in lighting and rotation of the image. The Harris definition of a corner relies on the matrix of the second order derivatives of the image intensities. The second-order derivatives of images, taken at all points in the image, can be thought of as forming new "second-derivative images" or, when combined, a new Hessian image. For the Harris corner detector, we consider the autocorrelation matrix of the second derivative images over a small window around each point. The corners will exist where this autocorrelation matrix has two large eigenvalues, meaning that there is texture (edges) going in at least two separate directions centred around said point [4].

### Scale Invariant Feature Transform

Contrary to the Harris corner detection, SIFT is, as the name indicates, scale invariant. This

algorithm starts by constructing a scale space, meaning that it creates internal representations of the original image to ensure scale invariance [14].

Then, we can apply a difference of Gaussians (DoG) to the image. This DoG is used as an approximation of a Laplacian of Gaussian (LoG) that, although it is very efficient in feature detection, it is computationally very expensive. The DoG will be applied over several scalings of the image (scale-space), and the features selected will be the stable local minima and maxima of the DoG over the scale space, making the algorithm scale invariant [14].

To find the orientation of the key points, the gradient orientation histogram is computed in a region around the key point, where each pixel in the region is weighted by the gradient magnitude and a Gaussian with a standard deviation of 1.5 times the scale of the key point. The highest peak in this histogram will be the dominant orientation. After the orientation is decided, the feature description is computed by using a set of orientation histograms on 4 by 4 regions. Each histogram exploits 8 bins to represent 8 directions, so a vector of $4 \times 4 \times 8 = 128$ elements is employed to describe each key point. This vector is normalised to be invariant to the illumination change, obtaining the final SIFT descriptor [28].

**Speeded Up Robust Features**

SURF is, in essence, a speeded up and more robust version of the SIFT algorithm, and consists of two main steps. First, it detects local interest points by adopting a fast approximation of the Hessian matrix that exploits integral images. The extracted interest points are then the local maxima of the Hessian matrix determinant [55].

The second step is the feature description. The SURF descriptor captures the intensity content distribution around the points detected before. The first order Haar wavelet responses are computed with the use of integral images. In order to achieve rotation invariance, a dominant orientation is determined by selecting the direction that maximises the sum of the Haar-wavelet responses in a sliding window around the neighbourhood of each interest point [55].

**Multi-Scale Oriented Patches**

The MOPS algorithm is a feature descriptor method. In the original work [6], the first step applied was the detection of interest points using a multi-scale Harris corner detector, as described above. Then, the number of interest points is suppressed (in order to reduce computational expenses in the matching phase) based on the corner strength, and only those that are maximum in a neighbourhood of radius $r$ pixels are retained. The value of $r$ should start at zero and be increased until the desired number of interest points is reached.

For the description of the interest points detected, given an oriented interest point, we sample an 8 by 8 patch of pixels around the location of the point, using a spacing of 5 pixels between samples. After sampling, the descriptor vector is normalised so that the mean is 0 and the standard deviation is 1, making the features invariant to changes in intensity. Finally, a Haar wavelet transform is performed in the descriptor patch to form a 64-dimensional descriptor vector containing the wavelet coefficients [6].

#### 2.2.1.2 Feature Matching

After the computation of the features, it is important to perform an adequate feature matching, in other words, find correspondence points in different images, in order to understand how they overlap and relate spatially [36]. In this Section, different feature matching algorithms will be discussed.

**Exhaustive Search**

The exhaustive search, also called brute-force algorithm, is a basic k-nearest neighbour (kNN) search method consisting in computing the distances between a feature descriptor in an image and each of the feature descriptors in a second image. Then, the k-nearest neighbours are determined using a sorting algorithm [11].

Although this may be a very simple algorithm, it is very demanding in terms of computation time, since the distance between every possible pair of features in the images must be computed. For that reason, the methods developed generally seek to reduce the computational time, by decreasing the number of distances that must be calculated [11].

**KD-tree Search**

The KD-tree is a form of balanced binary search tree, that is less computationally expensive than the exhaustive search. At the first level of the tree, the data is split into two halves, through the median of the dimension with the greatest variance in the data set (in this case, the set of feature descriptors from one image). By comparing our base feature descriptor from another image (or query vector) with the partitioning value, we can determine to which side of the data the query vector belongs to. Each of the halves of the data is then recursively split in the same way to create a fully balanced binary tree. At the bottom of the tree, each node corresponds to a single feature descriptor, which will be the first candidate for the nearest neighbour [5, 52].

This method allows a reduced number of distances calculated, when compared to the exhaustive search, providing a faster computation time.

### 2.2.2 Structure and Motion Recovery

One of the main steps in MVS is stereo mapping (mapping between images of the scene). For this, it is required to compute the homography between images. The planar homography is a non-singular linear relationship between points on planes: images of points on a plane in one view are related to corresponding image points in another view by a planar homography using a homogeneous representation [2].

**Perspective Projection**

Let us analyse a situation where a planar object $\pi$, with normal vector $n$ is observed from two different viewpoints (see Figure 2.2). Assuming two camera frames, defined as $F$ and $F^*$,

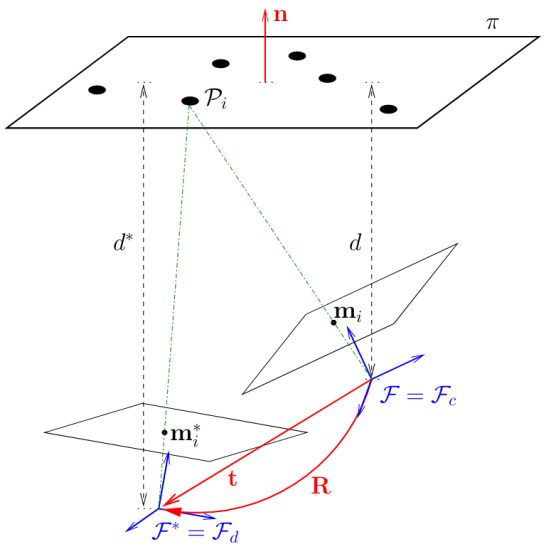



Figure 2.2: Homography induced by a plane, as seen on Malis and Vargas [30].

and a set of 3D points with Cartesian coordinates $P(X,Y,Z)$ belonging to the object observed, we can convert the 3D point coordinates from $F$ to $F^*$ by knowing the rotation and translation that occurred between the two frames. The matrix $[R|t]$ is called the extrinsics matrix, describing the rotation, $R$, of the camera and its translation, $t$ [10].

The image acquisition of such object implies the projection of the 3D points $P$ on a plane, so that the depth of each projection is the same, when related to its corresponding camera frame [30]. The normalised projective coordinates will be $m = (x,y,1)$, for frame $F$, and $m^* = (x^*,y^*,1)$, for frame $F^*$. To obtain the homogeneous image coordinates, $p = (u,v,1)$, in pixels, a simple transformation can be applied:

$$p = Km \tag{2.1}$$

$$K = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.2}$$

In Equation 2.1, $K$ (Equation 2.2) is the intrinsics matrix, that takes into account certain parameters of the camera itself, such as vertical and horizontal focal lengths $f_x$ and $f_y$, the principal point, $(c_x,c_y)$ and the skew, $s$. Generally, some assumptions can be made: we can say that $f_x = f_y$, assume that the centre of the image is the principal point, and make the skew equal to zero [9].

**The Homography Matrix**

With $p^* = (u^*,v^*,1)$ being a vector containing the homogeneous coordinates of an image point in

frame $F^*$, and $p = (u, v, 1)$ the vector of the homogeneous coordinates of an image point in frame $F$, the projective homography matrix transforms one vector into the other, up to a scale factor [30]:

$$\alpha p = Gp^* \tag{2.3}$$

This homography can be measured from image information, as seen on the following sections. Additionally, this matrix is related to the transformation elements, $R$ and $t$, as well as to the object plane, as follows [30]:

$$G = \gamma K(R + tn^T)K^{-1} \tag{2.4}$$

Assuming that an estimation of $G$ and $K$ is achieved, based on image information, the homography in the Euclidean space can be computed:

$$H = K^{-1}GK \tag{2.5}$$

The Euclidean homography matrix performs a similar transformation as the projective homography, but regarding the projective coordinates:

$$\alpha m = Hm^* \tag{2.6}$$

#### 2.2.2.1 Direct Linear Transform

The direct linear transform (DLT) is a simple algorithm used to compute the homography matrix given a sufficient set of point correspondences.

Given two corresponding points represented with homogeneous coordinates, they can be written as [9]:

$$\alpha \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = G \begin{pmatrix} u^* \\ v^* \\ 1 \end{pmatrix} \tag{2.7}$$

Where $\alpha$ is any constant (different from zero), $(u, v, 1)^T$ represents $p$, $(u^*, v^*1)^T$ represents $p*$, and $G = \begin{pmatrix} g_1 & g_2 & g_3 \\ g_4 & g_5 & g_6 \\ g_7 & g_8 & g_9 \end{pmatrix}$.

If we take Equation 2.7 and develop it, and divide the first and second rows of the equation by the third row, we obtain:

$$-g_1u^* - g_2v^* - g_3 + (g_7u^* + g_8v^* + g_9)u = 0 \tag{2.8}$$

$$-g_4u^* - g_5v^* - g_6 + (g_7u^* + g_8v^* + g_9)u = 0 \tag{2.9}$$

Since each point correspondence provides two equations, 4 correspondences are enough to solve for the 8 degrees of freedom of $G$ [2]. The only restriction is that no 3 points should be colinear. Theoretically, more than 4 correspondences could be used and, if the points were exact, there would always be a single homogeneous solution for $G$. However, in the real world the points used are not exact, and, consequently, there is not an exact solution, meaning that it becomes necessary to solve for a vector $g$ that minimises a suitable cost function (algebraic distance, geometric distance, reprojection error). Additionally, a process of normalisation can be applied to the DLT algorithm to ensure that the solution always converges to the correct result, despite not having exact data concerning the points [2].

#### 2.2.2.2 Random Sample Consensus

The DLT algorithm is only robust when faced with noise in the measurement of the points it uses, in other words, when the data is not exact. However, there are situations where the errors do not lay in the points themselves, but in the correspondences: two features in the images don't correspond to the same real-world feature. Random sample consensus (RANSAC) is the most commonly used robust estimation method for homographies [9].

When using RANSAC, for a set number of iterations, a random set of 4 correspondences is selected and a homography, $G$, is calculated. Each other correspondence is then classified as an inlier or outlier, depending on its concurrence with $G$. At the end of all the iterations, the iteration with the highest number of inliers is selected, and $G$ is recomputed considering all the correspondence points classified as inliers in that iteration. The decision of classifying each correspondence as an inlier or outlier is, usually, made by setting a threshold for the distance between $p$ and $Gp*$ [2].

#### 2.2.2.3 Least Median of Squares Regression

The lack of robustness of the DLT algorithms derives from the use of the sum of squared differences between $p$ and $Gp*$ as the cost function. One way to deal with this, besides RANSAC, is the use of the least median of squares (LMS), replacing the sum with the median of the squared errors. LMS works well if there are less than 50% outliers in the set, and it does not require the setting of thresholds or *a priori* knowledge of how much error to expect (contrary to what happens in RANSAC) [9].

#### 2.2.2.4 Homography Decomposition

After the estimation of the projective homography matrix, $G$, based on image information, it is, in some applications, necessary to decompose the matrix into its factors ($R$, $t$ and $n$), to better analyse the camera movement. The first step should be to obtain the corresponding Euclidean homography

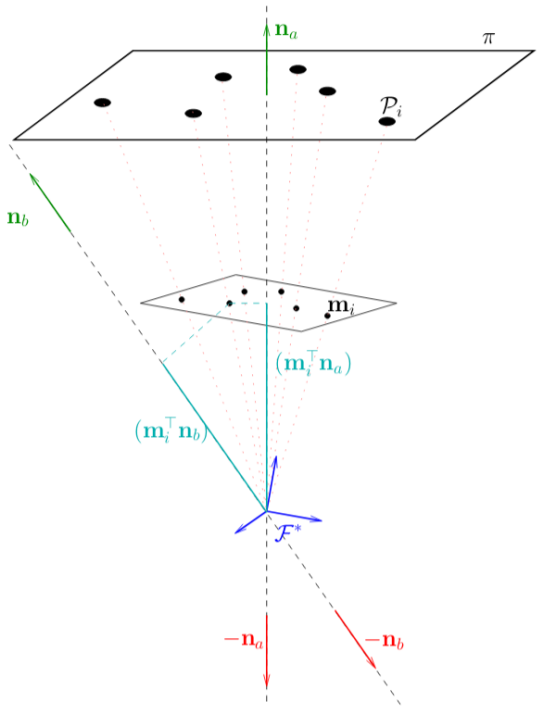



Figure 2.3: Geometrical representation of the homography constraint that determines that all observed points must be in front of the camera, as seen on Malis and Vargas [30].

matrix, H, according to Equation 2.5. From there, the decomposition can be made by resolving the following expression:

$$H = R + tn^T \tag{2.10}$$

This system can be solved, and 4 different solutions will be obtained. To decrease the number of solutions, we can apply a simple rule: for all the image points to be visible, they must be in front of the camera [30]. In mathematical terms, this means that:

$$m^T(RN) > 0 \tag{2.11}$$

Where, $m = K^{-1}p$. This rule is represented geometrically in Figure 2.3.

## 2.3 Deep Learning Techniques for Multi-view Stereo

Deep learning allows computational models that are composed of multiple processing layers - Neural Networks (NN) - to learn representations of data with multiple levels of abstraction. These methods have been used to improve the state-of-the-art in areas like speech recognition, visual object recognition, object detection and even domains such as drug discovery and genomics [24].

One of the areas where deep learning was applied successfully was in MVS, with several approaches being applied to perform homography, depth and ego-motion estimation. These less classical methods do not need to follow the traditional pipeline, shown in Section 2.2, having no need to establish frame-to-frame feature correspondence.

In this Section, a deeper understanding of neural networks will be given (Section 2.3.1), as well as an overview on some of the different types of networks being used today in many applications (Section 2.3.2). A deeper analysis will be made regarding deep learning usage for MVS applications (Section 2.3.4).

### 2.3.1 Feed-forward and Recurrent Neural Networks

Generally, neural networks (NN) can be divided into feed-forward (FFNN) or recurrent (RNN) neural networks [19]. While FFNN only allow data to flow in one direction - input to output - by connecting each layer only to the followig layers, RNN include connections in both directions, allowing the output of a certain layer to affect its input. This means that RNN have "memory", making them suitable for tasks where each output affects the following outputs (for example, the weather seen today can help make assumptions about the weather tomorrow), by incorporating sequential information [63].

The most basic form of a FFNN is a single-layer perceptron, where each neuron receives the inputs (one or more), computes the pre-determined activation function - linear or non-linear -, and outputs a predicted value. In this case, the neurons are organised into a single layer, which is also the output layer [48]. In the case of a multilayer perceptron, the neurons are organised into more than one layer, which are interconnected. The neurons in the intermediate layers - hidden layers - will compute the activation functions, and the result will serve as input for the next. The final layer will once again provide the final predicted value [48]. Examples of both types of network are represented in Figure 2.4.

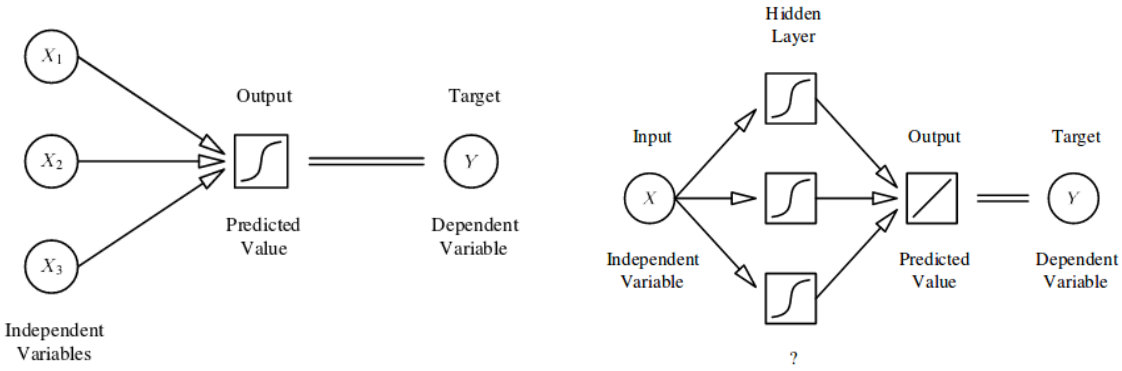



Figure 2.4: Schematic representation a single-layer perceptron (left), and a multilayer perceptron, with one hidden layer (right), from Sarle [48].

Each neuron in a NN will assign a certain weight to each input received. Each of the weights must be optimised in order to minimise the distance between the predicted value and the target. The optimisation process can be done through back-propagation algorithms, which will initialise the weights randomly, and proceed to compute the predicted value, and the associated error. With the errors known, the gradient associated with each weight can be computed (starting with the final layers, and working towards the first layers), and the weights updated based on those values.

From this general model, several types of networks were born, by including or removing connections between layers, introducing convolutional layers, and removing altogether the target values (originating unsupervised learning techniques).

### 2.3.2 Convolutional Neural Networks

Convolutional neural networks (CNN) are neural networks that include a feature extraction component, where a series of convolutions and pooling operations (or layers) are applied to the image in order to produce a feature map. After this component, some fully connected layers will receive the extracted features to perform the classification necessary. Once again, this process can be performed in several different ways, originating different types of CNN.

#### 2.3.2.1 GoogLeNet

GoogLeNet was proposed by Szegedy et al. [56], in order to improve neural network performance in computer vision tasks. This architecture implemented for the first time the so called inception modules.

An inception module applies different sized filters to the same input ($1 \times 1$ convolution, $3 \times 3$ convolution, $5 \times 5$ convolution, $3 \times 3$ pooling), and concatenates the results of each filter, to serve as input for the next layer. This allows the network to perform multi-level feature extraction, extracting both general and local features of the image at the same time. However, this architecture leads to an ever increasing number of outputs from layer to layer (when several inception modules are stacked), leading to an excessive computational effort. To avoid this, $1 \times 1$ convolutions are added as dimension reductions, giving us the final inception module architecture, as seen in Figure 2.5 [56].

The other main advantage of this architecture is that it allows the increase of the number of neurons in each layer, as well as the number of layers, without increasing computational complexity to a point where current technology cannot cope [56].

GoogLeNet is an inception network, since it is based on the stacking of several inception modules, with a specific architecture. It starts by performing two pairs of convolution and max pooling layers, followed by 9 inception modules, alternated with 3 max pooling layers. The final layers are similar to a classic neural network, with a dropout layer, a fully connected layer and a softmax layer [56].

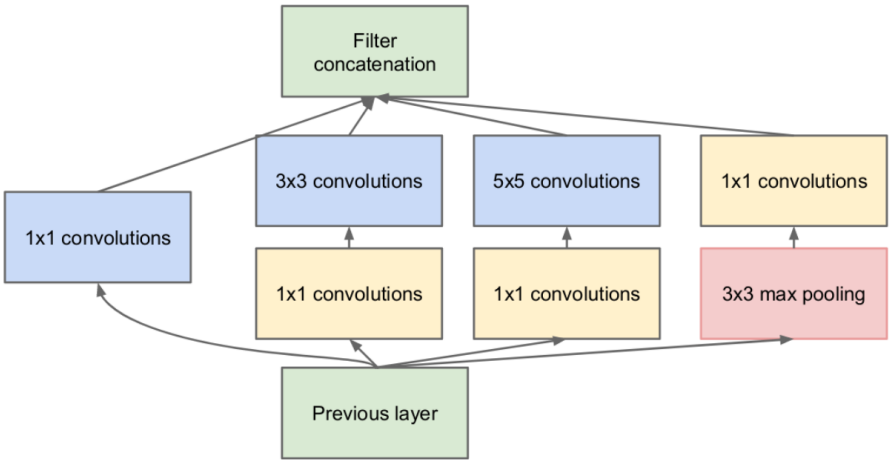



Figure 2.5: Inception module with dimension reductions, as seen on [56].

#### 2.3.2.2 DispNet

DispNet was first proposed in Mayer et al. [32] and applied to compute disparity between two images. This work explored the possibilities of applying different convolutional network architectures to optical flow applications.

This network consists of two parts. The first, a contracting part, contains convolutional layers and performs a downsampling of 64, allowing the network to estimate large displacements between the two images. This section is comprised of 10 convolutional layers. The second part, expanding part, gradually upsamples feature maps, by performing a series of up-convolutional and convolutional layers. Here, there are 15 layers (5 upconvolutional layers, and 10 convolutional and loss layers) that alternate amongst themselves. Figure 2.6 shows a schematic representation of the network.

This network allows disparity estimation between images, being able to run 1000 times faster, and produce better results, than other networks used for the same effect.

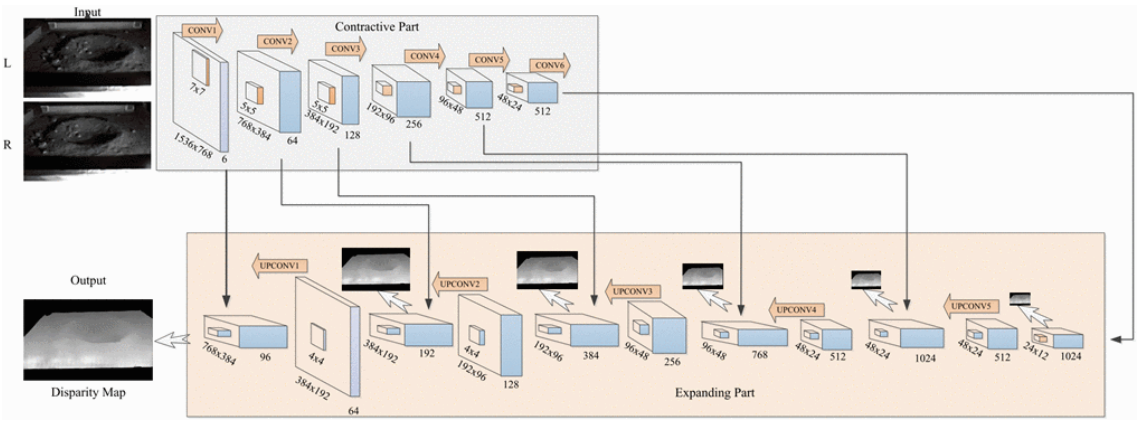



Figure 2.6: Schematic representation of the DispNet architecture, as seen on [20].

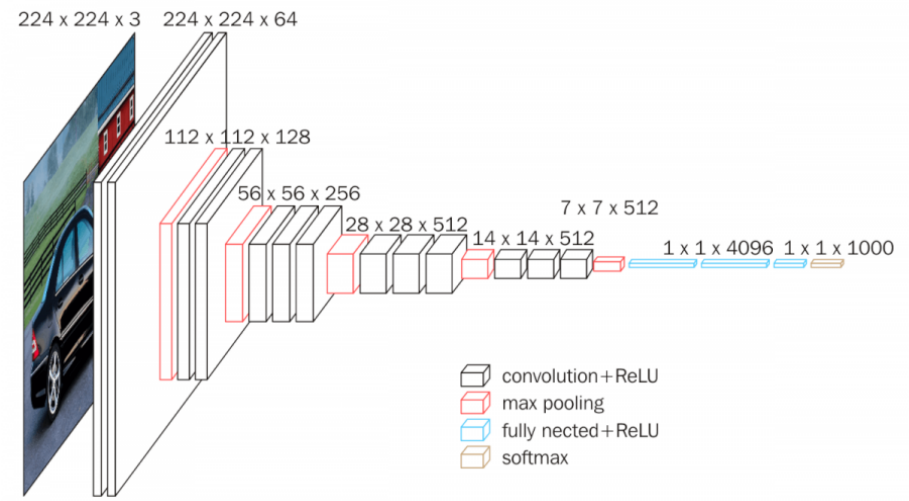



Figure 2.7: Schematic representation of the VGG Net architecture, as seen on Acharya et al. [1].

#### 2.3.2.3 VGG Net

VGG Net is a family of CNN architectures, that consist in the application of a greater number of convolutional layers (very deep networks) with small filters ($3 \times 3$). These networks were developed by Simonyan and Zisserman [53], in order to study the impact of an increased depth in the CNN performance. The filter size was chosen to be the smallest size possible, while keeping the notion of direction (left/right, up/down). Additionally, max-pooling layers are alternated with the convolutions. After the convolutional stage, three fully connected layers are included (Figure 2.7).

The several networks that belong to the VGG family are differentiated by the number of convolutional layers. For example, VGG-11 has 11 weight layers (8 convolutions and 3 fully connected), while VGG-19 has 16 convolutions and 3 fully connected layers.

This architecture has the advantage, because it uses small filters, of being able to increase the depth of the network, without increasing too much the number of parameters that need to be trained.

### 2.3.3 Long Short Term Memory

RNN are a type of neural network that as a temporal dimension, making them apt to be used in tasks such as speech and handwriting recognition, where a pattern is established between the desired outputs. As is the case with CNN, several variations of RNN have been developed. One of the most prominent is the Long Short Term Memory (LSTM).

For a standard RNN, the hidden state of a neuron at a given timestep is a function of the input at the timestep (affected by the weights), and the hidden state at the previous timestep, which is also multiplied by its own weight matrix, known as transition matrix. Since each hidden state depends on the previous one, all states will be affected by all that came before, with lesser and lesser extent. These networks can be trained through back-propagation algorithms, just like FFNN.

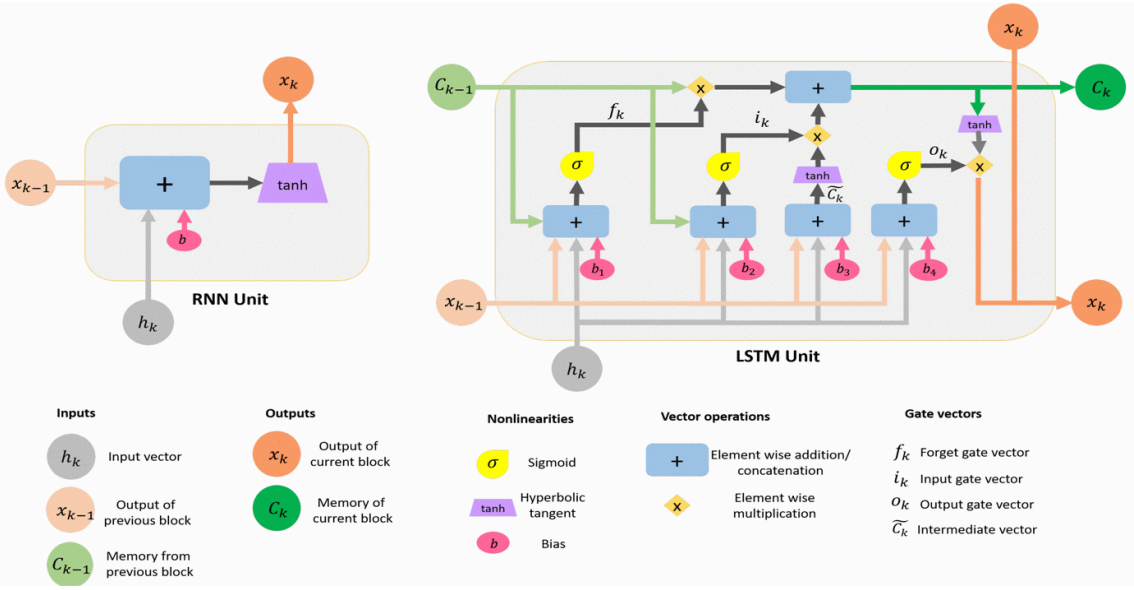



Figure 2.8: Architecture of a LSTM memory cell and its gate units, compared to a classic RNN structure, as seen in Khan and Zhang [22].

However, a problem arises when doing so: the vanishing or exploding gradient. The computation of the gradient of the network weights is essential in the training of the network. When frequently multiplying values slightly higher than one by any amount, the result quickly becomes very large. The same happens when considering values slightly smaller than one, with the results decreasing abruptly. In a RNN, the several timesteps relate to each other using multiplication, meaning that the gradients are very susceptible to vanishing and exploding. When dealing with exploding gradient, the weights associated become saturated, having too much influence in the final result, while vanishing gradients become too small for computers to handle, and makes the training process take a long amount of time [15].

As a solution to the vanishing problem, a RNN variation, the Long Short Term Memory unit, was proposed. LSTM preserve the error that can be propagated through time, making it more constant, and allowing the network to learn over many timesteps, without exploding or vanishing gradients [15]. This is done by using memory cells, seen in Figure 2.8. Information can be stored, written and read from a cell, by using analog gates, implemented with multiplication by sigmoids (keeping the process differentiable, and, thus, suitable for backpropagation). Based on the received signals, the gates will block and pass information, and filter it with a set of weights (adjusted during training, as RNN weights).

### 2.3.4 Supervised Learning in Motion Estimation

The main work done when it comes to supervised learning in motion estimation was the development of two "sister" networks to estimate the homography between two images, along with a new parameterisation of the traditional homography matrix and a technique for homography data generation [8].

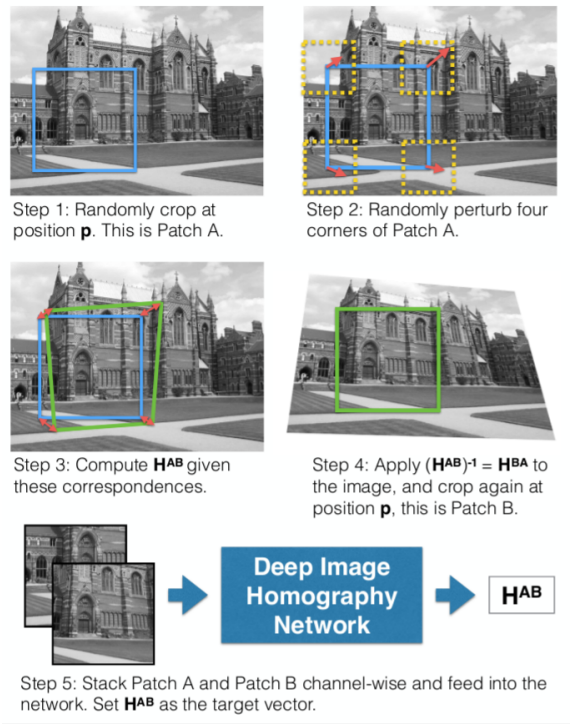



Figure 2.9: Training data generation: process for creating a single training example, as seen on DeTone et al. [8]

DeTone et al. [8] proposed a 4 point homography parameterisation, that eliminated some of the basic problems of the traditional homography matrix. Namely, the traditional homography mixes rotational and translational terms in a single matrix, which hinders the balance between the terms. The proposed parameterisation, however, is based solely on corner location, making it more suitable for deep learning approaches. Letting $\Delta u_n = u'_n - u_n$, and $\Delta v_n = v'_n - v_n$, for each $n$ point correspondence, with $u_n$ and $v_n$, and $u'_n$ and $v'_n$ being the point coordinates in each image. The 4-point parameterisation will be:

$$H_{4point} = \begin{pmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{pmatrix} \tag{2.12}$$

The 4-point homography can be converted to the traditional homography using DLT.

Concerning the data generation process, DeTone et al. [8] described a method that begins by cropping a random patch, at position $p$, from the image at hand (Patch A). Each corner of Patch A is then perturbed, within a predefined range $[-\rho, \rho]$, originating a 4-point homography: $H^{AB}$. The inverse of this homography ($H^{BA}$) can then be applied to the original image, generating our warped image, from which a patch will again be cropped at position $p$ (Patch B). Patches A and B can then be fed into a neural network, along with the computed homography, which will serve as ground truth. This process is depicted in Figure 2.9.

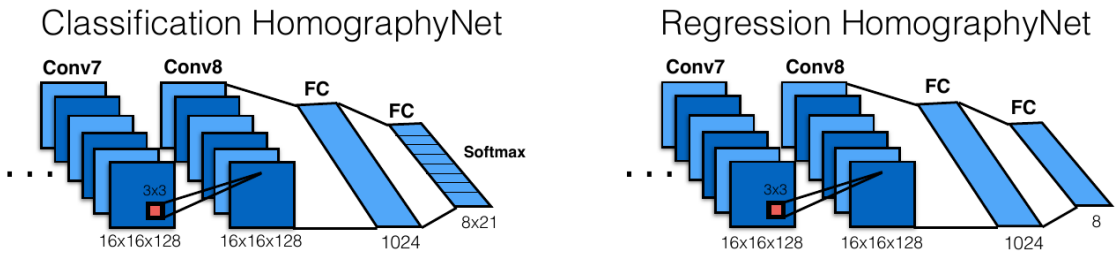



Figure 2.10: Classification HomographyNet vs Regression HomographyNet, as seen in DeTone et al. [8]. The network has 8 convolutional layers and 2 fully connected layers. The final layer is $8 \times 21$ for the classification network and $8 \times 1$ for the regression network.

The proposed networks themselves are CNN, with an architecture similar to a VGG Net [53], meaning that they have a series of convolutional blocks with batch normalisation [17] and rectified linear unit (ReLU) activation functions. Both networks take as input two $128 \times 128$ grey-scale images that are related by a homography (in this case, the patches produced in the previous step), and produce a 4-point homography.

The networks differ in that one of them works with a regression method, producing 8 real-valued numbers, while the other is a classification network, using a quantisation scheme with 21 bins for each of the 8 output dimensions (168 output neurons) (see Figure 2.10). Although the quantisation has an inherent error associated, the classification network is able to produce a degree of confidence for each element of the homography, which is impossible in the regression network [8]. Another major difference is the loss function: cross-entropy (Equation 2.13) for the classification network, and Euclidean, or L2, (Equation 2.14) for the regression network.

$$Cross-Entropy\ loss: -\sum p(x)\log q(x) \tag{2.13}$$

$$Euclidean\ loss: \frac{1}{2}\|p(x)-q(x)\|^2 \tag{2.14}$$

These techniques are able to obtain good results, when compared to the more traditional methods already described, with regression version of the network showing the best performance.

### 2.3.5 Unsupervised Learning in Motion Estimation

When it comes to motion estimation using unsupervised learning, there are two works that must be discussed: unsupervised deep homography and unsupervised depth and ego-motion estimation.

First, Nguyen et al. [38] proposes a semi-supervised network, based on the regression HomographyNet from DeTone et al. [8], in order to enable the estimation of the homography between to images, without the need to have a labelled dataset for training (Figure 2.11). Thus, the loss function is altered, becoming a pixel-wise photometric loss (Equation 2.15).

$$\mathbf{L}_{PW} = \frac{1}{|\mathbf{x}_i|}\sum |I^A(H(x_i)) - I^B(x_i)| \tag{2.15}$$

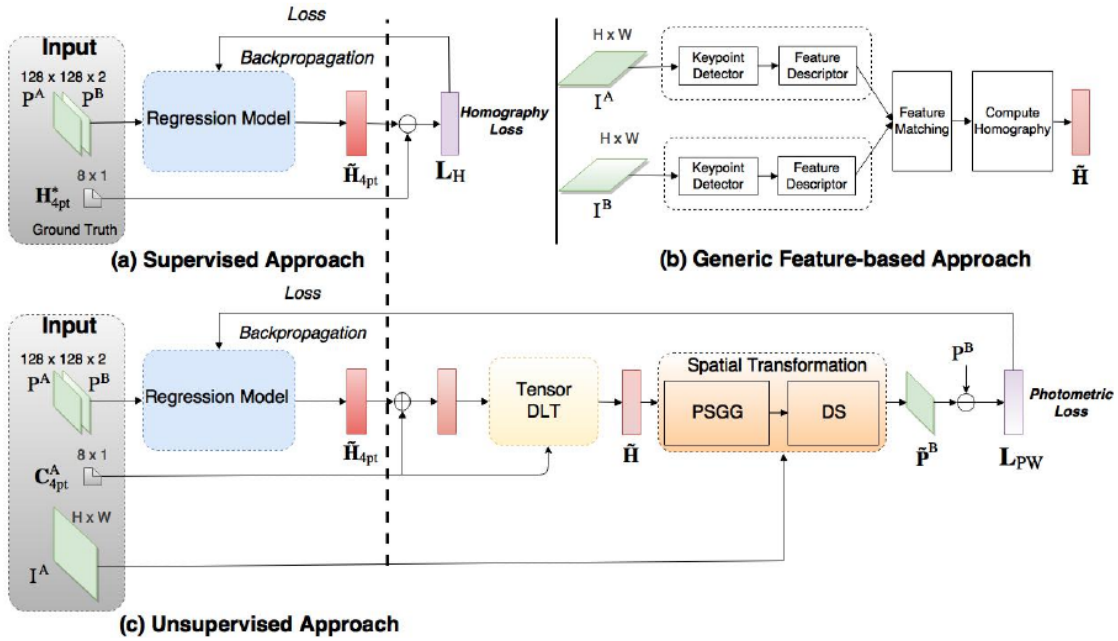



Figure 2.11: Overview of deep homography estimation methods, and contributions of the unsupervised homography, as seen in Nguyen et al. [38].

Given an image pair, $I^A$ and $I^B$, the photometric loss compares the original image $I^B$ with image $I^A$ warped according to the estimated homography. Since it does not use labels for the training process, the authors consider this an unsupervised network [38].

However, this loss function can only be applied if the operations remain differentiable, in order to allow back-propagation during the training of the network. With this in mind, two layers are added to the original HomographyNet architecture.

A Tensor Direct Linear Transform (TDLT, see Section 2.2.2.1) computes a differentiable mapping from the 4-point homography parameterisation to the traditional homography matrix. The inputs to this layer will be the corners of Patch A and Patch B (see Section 2.3.4), and the output the $3 \times 3$ homography matrix.

Following the TDLT, a spatial transformation is applied, to warp the original image $I^A$, according to the $3 \times 3$ homography estimated. Once again, this process needs to be differentiable, to allow the training of the network. Thus, first, the normalised inverse homography is computed. After this, a Parameterised Sampling Grid Generator (PSGG) will create a grid, $G = G_i$, with the same dimensions as $I_B$, with $G_i = (u'_i, v'_i)$. If we apply the inverse homography to the grid coordinates, the result will be a grid of pixels in $I_A$ (Equation 2.16).

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathscr{H}_{inv}(G_i) = \tilde{\mathbf{H}}_{inv} \begin{bmatrix} u'_i \\ v'_i \\ 1 \end{bmatrix} \tag{2.16}$$

Based on the sampling points computed, a Differentiable Sampling (DS) will originate an image $V$, where $V(\mathbf{x}_i) = I_A(H(\mathbf{x}_i))$.

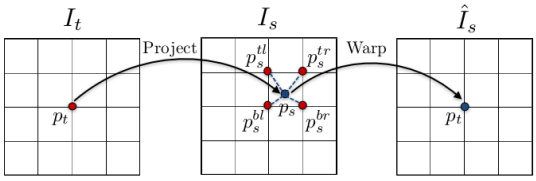



Figure 2.12: Differentiable image warping process, according to Zhou et al. [65]. First, the target point is projected onto the source view, and then bilinear interpolation is used to obtain its value in the warped image.

This set of differentiable operations, allow the model to be trained without the need for a ground truth label for the homography, as previously seen in DeTone et al. [8]. However, if the final intention of the algorithm is to obtain an absolute localisation of the camera, it would require further processing after the implementation of the network.

Zhou et al. [65], on the other hand, developed an unsupervised technique that not only regresses the pose of the camera at any moment in time, but also provides an estimation of the depth map of the scene, through a framework for jointly training two different CNN. The supervision of these networks will be based on view synthesis, according to which an target image is synthesised based on a depth map of that image and the pose from a nearby view, in a fully differentiable manner.

To do this, the first step is to define the input for the framework, which will be a sequence of frames $< I_1, ..., I_N >$, with one being the target frame $I_t$ and the rest source views $I_s$. The view synthesis objective function of the framework will be:

$$L_{vs} = \sum_s \sum_p [I_t(p) - \hat{I}_s(p)] \tag{2.17}$$

Here, $\hat{I}_s$ is $I_s$ warped to the target coordinate frame, and $p$ iterates over pixel coordinates. The warping will be performed based on the predicted depth map $\hat{D}_t$ and relative pose $\hat{T}_{t \to s}$, through a completely differentiable process. To project the coordinates $p_t$ of a pixel in the target view onto the source view, obtaining $p_s$, we can apply Equation 2.18, where $K$ is the camera intrinsics matrix.

$$p_s \sim K \hat{T}_{t \to s} \hat{D}_t(p_t) K^{-1} p_t \tag{2.18}$$

To obtain the value of $I_s(p_s)$, a linear interpolation of the values of 4-pixel neighbours is performed. The value of $I_s(p_s)$ is then used to populate $\hat{I}_s(p_t)$. This mechanism is depicted in Figure 2.12.

The use of view synthesis as supervision for the network means that some assumptions are made concerning the image sequence, including that the scene is static, with no occlusions. To improve the robustness of the network, Zhou et al. [65] trained an explainability network, jointly with the two previously mentioned networks, that outputs a per-pixel mask $\hat{E}_s$, that represents

the network's belief in where the view synthesis will be correctly modelled. This factor can be included in the loss function for the view synthesis, obtaining:

$$L_{vs} = \sum_{<I_1,\ldots,I_N>\in S} \sum_{p} \hat{E}_s(p)|I_t(p) - \hat{I}_s(p)| \tag{2.19}$$

However, this formulation means that the network would always predict $\hat{E}_s$ as zero. To prevent this, a regularisation term $L_{reg}(\hat{E}_s)$ is added, allowing the network to minimise the view synthesis objective, while considering the factors (such as occlusions and non-static scenes) not considered by the model [65].

Finally, since the gradients used for the training of the networks are mainly based on intensity differences between $I(p_t)$ and its four neighbours $I(p_s)$, if the correct $p_s$ is in a low-textured area or far from the current estimation, the training would be inhibited. To correct this issue, a multi-scale and smoothness loss is integrated to the existing loss function, obtaining the final objective function:

$$L_{final} = \sum_{l} L^l_{vs} + \lambda_s L^l_{smooth} + \lambda_e \sum_{s} L_{reg}(\hat{E}^t_s) \tag{2.20}$$

Here, $l$ indexes over different image scales, $s$ indexes over source images, and $\lambda_s$ and $\lambda_e$ are weighing the smoothness and explainability losses, respectively.

Concerning the network architecture, the single-view depth prediction uses a DispNet architecture [32], with ReLU activations, and a single view as input. For the pose estimation, the input will be the target view concatenated with all source views. This network consists of seven convolutional layers of stride 2, followed by a $1 \times 1$ convolution with $6 \times (N-1)$ (3 Euler angles and 3D translation for each source view) output channels, and an average pooling layer to aggregate predictions at the different spatial locations. The explainability network has the same first five layers as the pose network, followed by 5 deconvolution layers, all with ReLU activations.

As for the results obtained with these techniques, the depth estimation network shows worse results when compared to its supervised counterparts, although it is able to recover the general layout of the scene well. The results for pose estimation, although comparable, are still not to the same standard as the more classical approaches, with a higher absolute translation error.

## 2.4 Visual Odometry for Endoscopic Capsules

The localisation of endoscopic capsules has been an explored topic in recent years. Although some classic techniques presented in Section 2.2 have been applied to this end [55, 23], in this Section we will focus on works that apply deep learning techniques to perform visual odometry of endoscopic capsules.

### 2.4.1 Pre-processing Techniques for Endoscopic Images

The environment of acquisition of VCE images is very far from ideal, meaning that the images will often have very poor quality, either because of the movement of the capsule, that might blur the scene, because of the low camera resolution of endoscopic capsules, or due to the gastric content that often blocks the view of the walls of the GIT. Additionally, some of the features that may be essential in the MVS process are not very differentiable from the background, even if the image has a perfect quality. This means that the pre-processing step is essential in VCE problems, allowing for better results of the overall pipeline. Thus, several techniques have been tested in VCE frames [29, 50], and will be further explored in the following Section.

#### 2.4.1.1 Homomorphic Filtering

Homomorphic filters consist of the nonlinear mapping of the image into a different domain (frequency domain, for example), where a linear filter can be applied. In Ramaraj et al. [45], homomorphic filtering is used in order to improve the contrast in VCE images, using both the Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT) techniques.

Both techniques start by representing the image as a multiplication of illumination and reflectance (see Equation 2.21). The illumination represents the amount of source light incident on the scene, while reflectance translates the component of light that is reflected by the objects in the scene. In general, the illumination parcel has a lower frequency than the reflectance parcel, allowing the filter to reduce the significance of the illumination and making the images more homogeneous among themselves.

$$I(x,y) = i(x,y) \times r(x,y) \tag{2.21}$$

To achieve the filtering desired, the first step will be to apply the logarithm to the function above, to transform the multiplication into a sum. Each of the parcels can then be subject to either DWT or DFT, to convert the signal to the frequency domain. The entire image is then passed through a high-pass filter, eliminating the lower frequencies of the illumination parcel. Inverse DWT and DLT are applied to convert the image back to the spatial domain, in which it will pass through an exponential to reverse the logarithm. After these steps, the final filtered result is obtained.

In order to evaluate the results obtained, Ramaraj et al. [45] also applied two other contrast enhancing methods: anisotropic diffusion method (also known as Perona-Malik diffusion, or PM diffusion) (see Section 2.4.1.2), and Contrast-Limited Adaptive Histogram Equalisation (CLAHE). An example of the results obtained by Ramaraj et al. [45] can be seen in Figure 4.3. The best results, according to several image quality metrics, were the ones given by homomorphic filtering, using DFT.

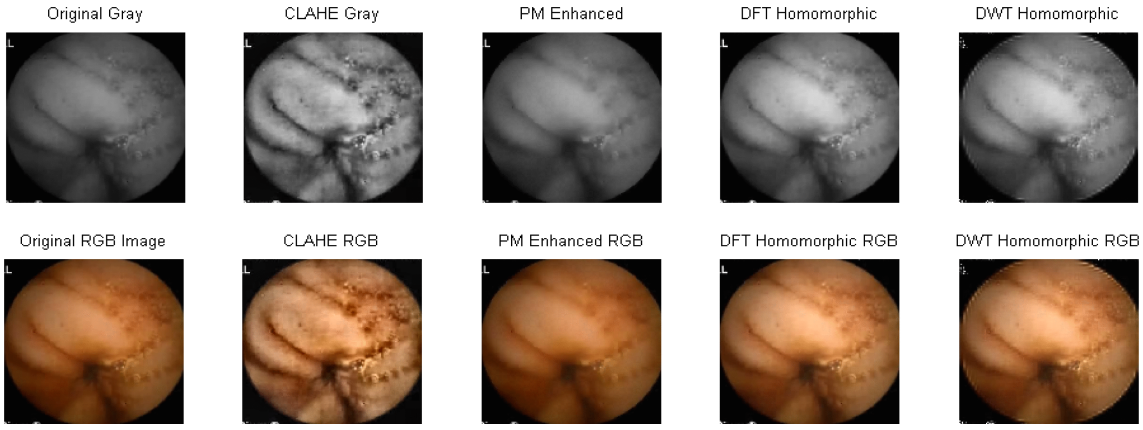



Figure 2.13: Application example of homomorphic filtering, using both DWT and DLT, and result of other types of filters (PM diffusion and CLAHE) for comparison. From Ramaraj et al. [45].

#### 2.4.1.2 Anisotropic Diffusion

Perona and Malik [41] first proposed the use of anisotropic diffusion (or PM diffusion) techniques, in order to reduce image noise, while keeping important features, such as edges, lines and corners, intact. This technique involves the convolution of a signal with Gaussians at different scales (equivalent to solving a diffusion equation [25]), in an iterative fashion: each new image is obtained by applying the diffusion equation (see Equation 2.22) to the previous one (starting with the original image).

$$\frac{\partial I(x,y,t)}{\partial t} = div(c(x,y,t)\nabla I) \tag{2.22}$$

Where $c(x,y,t)$ is the diffusion coefficient. To allow intra-region smoothing, while enhancing the edges of the image, Perona and Malik [41] proposed a diffusion coefficient that is chosen locally, based on the magnitude of the gradient (Equation 2.23).

$$c(x,y,t) = g(\|\nabla I(x,y,t)\|) \tag{2.23}$$

The function $g(\cdot)$ must be chosen so that the response is minimal for high values of the gradient (edges), and have a high response to low values of the gradient (intra-region smoothing). One example of an adequate $g(\cdot)$ can be seen in Figure 2.14. The function used in Perona and Malik [41] is defined as:

$$g(\|\nabla I\|) = \frac{1}{1+(\|\nabla I\|/K)^2} \tag{2.24}$$

The $K$ parameter will determine the amount of smoothing the filter will perform: higher $K$ will mean more smoothing.

Although this method works quite well in several applications, it relies on the capacity of the gradient to accurately identify edges. This, however, is not always possible, resulting in the smoothing of important areas of the image, and enhancing of irrelevant ones.

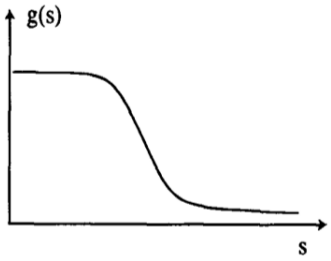



Figure 2.14: Example of diffusion coefficient function. From Perona and Malik [41].

#### 2.4.1.3 Adaptive Contrast Diffusion

It is due to the flaws of the PM diffusion that the adaptive contrast diffusion was developed, having in mind the difficulties that surround VCE images. This technique is very similar to the PM diffusion method, but, in this case, instead of adapting $g(\cdot)$ according to the gradient of the image, it is a function of the Hessian matrix (Equation 2.25), where each element is the second order derivative of the image along a different direction [25].

$$\mathbf{H}_\sigma(x,y) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \tag{2.25}$$

The function $g(\cdot)$ is defined as [25]:

$$g(c) = \frac{1}{1+(\|c\|/k)^2} \tag{2.26}$$

The parameter $k$ will determine the behaviour of the diffusion according to the Hessian matrix of the region: a large $k$ will mean more smoothness, and a lower $k$ will mean more sharpness. By using a fixed value for $k$, we may be smoothing important details of the image, or enhancing noise, depending on the characteristics of each region or even each image. Thus, Li and Meng [25] proposes an adaptive $k$, designed as a function of the eigenvalues of the Hessian matrix:

$$k(x,y) = \frac{1}{\sqrt{\lambda_1^2(x,y)+\lambda_2^2(x,y)}} \tag{2.27}$$

This way, $k$ should adapt not only to the image in question, but also to the specific region, obtaining high values for regions with low contrast (background), and low values when considering abnormalities in the image.

### 2.4.2 Endoscopic Capsule Location Estimation

Turan et al. [61] proposes a 6 degree of freedom (DoF) localisation of both endoscopic capsule robots and standard handheld endoscope using a CNN system. It intends to demonstrate that CNN learns the most relevant feature vector representation related to 3D position and orientation

estimation of the robot from 2D raw input images. The proposed system takes a single RGB image and regresses the camera's 6-DoF pose.

The CNN architecture proposed in Turan et al. [61] was inspired by GoogLeNet [56], having a stack of convolutional layers, max pooling layer, normalisation layer, 9 inception modules, pooling layer, fully connected layer and affine regressor layer, giving a total of 23 layers. The hyperparameters of the convolution layers are optimised during training, and ReLU activation functions are used, to avoid vanishing gradients. The use of inception modules allows the extraction of both large scale and small scale information, by using different sized kernels, acting as multiple convolution filter inputs with integrated pooling. This system will learn translational and rotational movements simultaneously.

During training, Adaptive Moment Estimation (Adam) optimisation method was used to optimise the goal function - Euclidean loss - , seen in Equation 2.28 [61].

$$loss(I) = \|\hat{x} - x\|_2 + \beta \|\hat{q} - q\|_2 \tag{2.28}$$

Where $x$ is the translation vector and $q$ is the rotation vector. A balance $\beta$ must be kept between the orientation and translation loss values, which are highly coupled, since they are learned from the same model weights [61].

For the first layers of the proposed architecture, Turan et al. [61] used transfer learning (using the weights acquired by ImageNet), in order to sidestep the problem of needing large amounts of training data. A dataset was created and labelled, to perform the fine-tuning of the network, using 3 different endoscopic cameras. The videos were recorded in a realistic surgical stomach model, and paraffin oil was used to increase the reflection inside the simulator. In total, 15 minutes of video were recorded. To extend the data, different image distortion techniques (Gaussian blur, median blur, brightness distortion) were applied to the raw images [61].

To evaluate the performance of the net, two experiments were conducted in Turan et al. [61]. The first used a set of 10000 frames (7000 for training and 3000 for validation) without any distortion. The second used 70000 frames, both raw and distorted data originated from the original dataset. In both cases, the CNN was then tested in a test dataset. The second experiment, including the distorted images, provided better results, with a rotation error (root-mean-square error, RMSE) of 1.60% in the $x$ axis, 3.01% in the $y$ axis, and 5.71% in the $z$ axis. The translation errors are 4.72%, 9.16% and 7.44% in the $x$, $y$ and $z$ axis respectively.

The techniques for estimation of capsule location are not limited to simple convolutional neural networks. Since VCE frames have an inherent sequence, a natural addition to these techniques is the use of recurrent networks. This is seen in Turan et al. [59], where an architecture consisting of recurrent networks embedded into a CNN is proposed.

The pipeline proposed starts by performing some pre-processing techniques to the images, in order to eliminate reflections, and correct lens distortion. The images are then used for 3D surface reconstruction, using a Shape from Shading (SfS) tehcnique [58, 64], obtaining depth maps of the images. Finally, a scene flow estimation between consecutive images is performed, using the

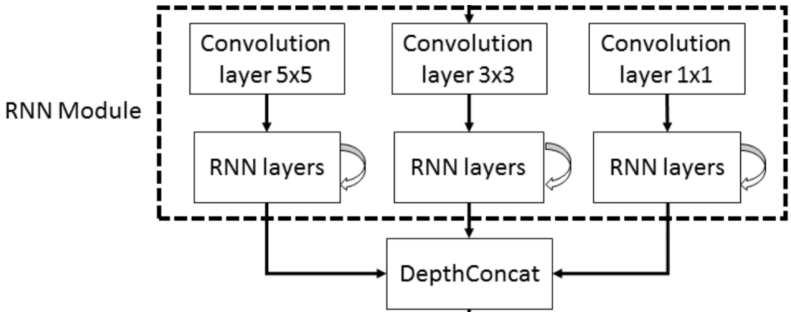



Figure 2.15: Schematic representation of the RNN module, as seen on Turan et al. [59]. After each RNN module, depth concatenation is performed to compile the results of the module.

depth maps, according to the technique described in Jaimez et al. [18]. It is the motion information between the frames, in the form of the scene flow, that will be fed into the network.

The architecture of the network proposed was inspired by PoseNet [21], with the addition of RNN modules within each convolutional module. The network is comprised of a convolutional layer, followed by two sets of RNN module (Figure 2.15), depth concatenation and max pooling layers, with the last layer being a regression layer (as opposed to the traditional softmax layer). The output will be the capsule orientation and position at each given frame. The training was done using the following loss function:

$$loss = ||\hat{x} - x||_2 + \beta||\hat{q} - \frac{q}{||q||}||_2 \tag{2.29}$$

As seen previously, the $\beta$ parameter ensures a balance between orientation and position values.

The dataset used to train was comprised of images recorded with 3 different camera models, to ensure that the model was not overfitted for a certain type of camera. The images were of a GIT surgical model, with paraffin oil applied, to imitate the mucosa of a real alimentary canal. Furthermore, several trajectories were tested, with different levels of complexity.

The final results were similar to those obtained previously with a simple CNN [61], with RMSE between 2.66% (average throughout all cameras), in the simplest trajectory, and 8.33%, in the most complex. This shows that the addition of recurrent units does not improve the results in a significant way.

Turan et al. [60] proposes the application of a deep RCNN for the visual odometry task, where convolutional neural networks and recurrent neural networks are used for feature extraction and inference of dynamics across the frames, respectively. The architecture makes use of inception modules for feature extraction and RNN for sequential modelling of motion dynamics to regress the robot's orientation and position in real time. It takes 2 consecutive endoscopic RGB Depth frames, each with timestamp, and regresses the 6-DoF pose of the capsule [60].

For the depth image creation from RGB input images, a SfS technique was used, assuming that the surface is Lambertian, there is a single point light source, and the surface has no self-shaded areas. After creating the depth images, the mean RGB Depth value of the training set is subtracted

from the RGB Depth images. Then, the preprocessed RGB Depth frame pair is stacked to form a tensor, which will serve as input to the network.

The proposed network consists of 3 inception layers and 2 RNN (more specifically, LSTM) layers concatenated sequentially. The final inception layer passes the feature representation into the RNN modules. Although the LSTM [15] is not prone to the vanishing gradient problem of RNN and is capable of detecting the long-term dependencies, its learning capacity can be increased further by stacking multiple LSTM layers vertically. Thus, the network proposed by Turan et al. [60] consists of 2 LSTM layers with the output sequence of one forming the input sequence of the second one. As seen in Turan et al. [61], the system learns translational and rotational motions simultaneously to regress the 6-DoF pose and is trained on Euclidean loss (Equation 2.28) using Adam optimisation method.

To train the network, two training sets were created. The first was recorded on five different pig stomachs, and the second was recorded in a surgical simulator. These images were recorded with four different camera models, and each stomach-camera combination has 2000 frames associated, making for a total of 40000 frames with pig stomach. For the second dataset, 10000 frames were acquired with each camera, for a total of 40000 frames. The test set consisted of frames from pig stomachs that were not used in the training set, with 40000 frames in total [60].

The developed network was used in 2 distinct ways: using only a synthetic dataset - simEndoVO -, and using both synthetic and real pig stomach dataset - realEndoVO. These nets, along with implementations of GoogLeNet and ResNet50 for comparison purposes, were evaluated based on the root mean squared error (RMSE) value for translational and rotational motions [60]. Across all the trajectories tested, realEndoVO showed the best results for the translation motion, followed by the simEndoVO, with the other nets having a far worse result. However, concerning rotational motion, although the realEndoVO performed considerably better than the others, simEndoVo was outperformed in shorter trajectories.

Some works do not compute the pose of the camera directly, instead estimating the homography between consecutive images, and using this information to compute the displacement between the two. This is the case with Pinheiro et al. [42], where HomographyNet [8] is applied to a endoscopy dataset, obtaining an homography estimation. Here, the same type of data generation as seen on DeTone et al. [8] is applied, and the network is trained using only synthetic data, using real data only for validation purposes. The network outputs an estimation of the 4-point homography between the two provided images, obtaining a Mean Average Corner Error (MACE) of 2 pixels. In Figure 2.16, the results of the homography estimation on real data tests can be seen.

To obtain the displacement between two real images, the pipeline first filters the images, eliminating those with gastric content or air bubbles, such that it partially or fully covers the image, leaving only images that are considered informative. If two consecutive frames are classified as informative, then they are submitted to the network, and a 4-point homography estimation is computed. The 4-point homography is then converted to the classical homography matrix, which is multiplied by the 2D coordinates of the previous camera position, obtaining the current camera position. The coordinates are then converted from pixels to meters, and only the displacement

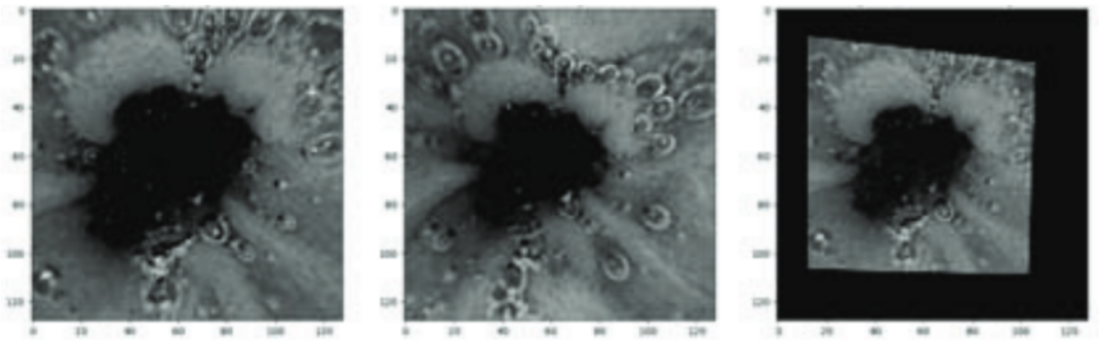


Figure 2.16: Homography predictions in real data, as seen on Pinheiro et al. [42]. From left to right: first frame, second frame, and first frame warped into the second frame perspective, according to the estimated homography.

in the $x$ axis is considered, since the shape of the small bowel renders any movement in other directions negligible.

In Turan et al. [62] a real-time localisation and depth estimation approach for endoscopic capsule robots by training an unsupervised deep neural network is proposed. The network consists of two simultaneously trained subnetworks, the first one assigned for depth estimation via encoder-decoder strategy, the second assigned to regress the camera pose in 6-DoF. Simultaneously, a reliability mask, which identifies pixels distorted by camera occlusions, non-rigid organ deformations and/or non-Lambertian surface, which commonly occur in endoscopic images. This work mimics the system seen in Zhou et al. [65], applying it to endoscopic images. The system was pre-trained with the KITTI dataset (transfer learning), and fine-tuned with 12000 porcine-stomach (ex-vivo) images. The results show that for both translational and rotational movement, the method proposed is better than classic methods (regarding Absolute Trajectory Error), but is still not up to the standards of the Deep EndoVO, proposed by Turan et al. [61]. Concerning the depth estimation, although the transfer learning improves the results significantly, there is still room for improvement.

## 2.5 Available Databases

One key aspect when performing work on endoscopic capsule localisation is the database used. These datasets are rare, since they are not always publicly available or do not fulfil all the requirements for each application, as is the case with Penza et al. [40], for example. There are, however, some available endoscopic video datasets that should be further explored in this Section.

**Private Dataset** First, we can consider the private dataset made available to be used in this work, and previously used in Martins Pinheiro et al. [31]. This private dataset consists of 449 endoscopic capsule videos, spanning the entire GIT. From this set, 338 videos were recorded with PillCam SB2 endoscopic capsule and 116 with PillCam SB3. Some of the videos recorded with PillCam

SB3 have a unitless displacement attached, based on landmarks marked by a physician and radio-frequency sensor array, and most include medical annotation about topographic landmarks, lesions and other abnormalities. Even though this information cannot be used as ground truth (that is not available for any of the videos), it can be helpful for the validation of the developed methods.

**Kvasir Dataset** Another publicly available database is the Kvasir Dataset [44]. This dataset consists of a set of endoscopic images, annotated by physicians with information relative to anatomical landmarks and pathological findings. The anatomical landmarks include Z-line (transition between oesophagus and stomach), pylorus (between the stomach and the small bowel), and cecum (most proximal part of the large bowel). The images were collected at *Vestre Viken Health Trust* (consisting of 4 hospitals), in Norway. The exact size of the dataset is not specified but it claims to be sufficient for the application in deep learning systems.

**GIANA Endoscopic Challenge** GIANA Endoscopic Vision Challenge also provides a dataset for its participants to use, to which we have access. This dataset does not include complete VCE, focusing on the detection of abnormalities in single frames. It is made up of 1812 VCE frames, divided into 3 classes: normal (600), inflammatory (607), vascular (605). Examples of the different classes can be seen in Figure 2.17. The images also have an attached binary mask that corresponds to the segmentation of the lesion (if a lesion is present). Although it can not be used for the application at hand, this dataset can be included in the data generation step of the work, allowing a larger amount of small bowel images to be learnt by the networks.


Figure 2.17: GIANA Endoscopic Vision Challenge dataset examples. On the left, a normal VCE frame; in the middle, a VCE frame with a inflammatory lesion; on the right, a VCE frame with a vascular lesion.

## 2.6 Summary

Throughout the state-of-the-art examination, it was possible to understand that, indeed, deep learning techniques are an improvement when compared to the more classical methods for multi-view

stereo, both when considering general MVS applications, as well as endoscopic capsule localisation problems. Computing the camera pose in each frame (using a CNN) will mean a simple pipeline, without the need to further process the results provided by the network. However, by following a more divided approach (computing first the homography, and then the displacement between frames), the process becomes easier to analyse, ensuring good performance in each of the steps applied.

The pre-processing of VEC frames seems to be a very important step in any pipeline which uses this kind of images, due to their generally poor quality (mainly because of low camera resolution).

Finally, although some datasets are available for endoscopy applications, publicly available and labelled databases are very difficult to find, making supervised deep learning approaches for capsule localisation unfeasible in most situations.

# Chapter 3

# Framework for Homography-based Capsule Displacement Estimation

## 3.1 Problem Characterisation

The increasing use of capsule endoscopy for the diagnosis of possible small bowel lesions and abnormalities, as also means a more pressing necessity to solve the issues that it still entails. Several works have been published in the attempt to obtain remote controlled capsules, or even capsules that can go beyond visualisation, being able to collect tissue samples for future biopsies [12]. However, one of the most important issues that still remains unsolved is the capability to immediately locate any lesion found through capsule endoscopy. VCE images have several characteristics which hinder the visual location of the capsule, making the creation of a novel method for the automatic localisation of VCE frames imperative.

The method developed should take into consideration some characteristics of the VCE, as well as the needs expressed by clinicians. It should be able, first and foremost, to provide accurate localisation of the capsule at any given frame. The localisation can either be in terms of displacement within the GIT (related, for example, to the beginning of the small bowel), or even 3D coordinates. Another important issue is the lack of ground truth information regarding the capsule position in any available database. This makes the implementation of a supervised method very difficult, unless synthetic data is used for training, as opposed to real data. Finally, the very nature of the organ hinders greatly any method applied. Gastrointestinal content, mainly in the form of green residue and air bubbles, may appear in the images, making it impossible to identify distinctive features in the organ itself. Additionally, the peristaltic movements characteristic of the small bowel make it possible for the capsule to move in any direction, and mean that the scene is not static: the changes in the scene are not only caused by the movement of the capsule but also by the movement of the organ itself.

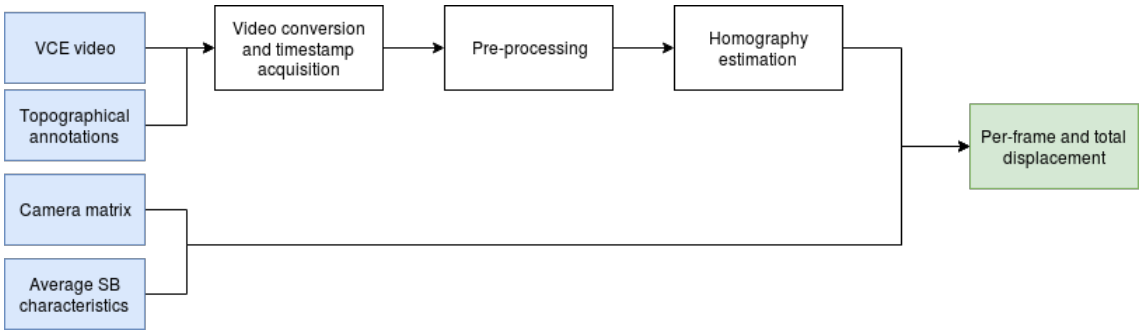



Figure 3.1: Schematic representation of the implemented pipeline. In blue, the needed inputs for the system; in green, the main output.

## 3.2 Pipeline Overview

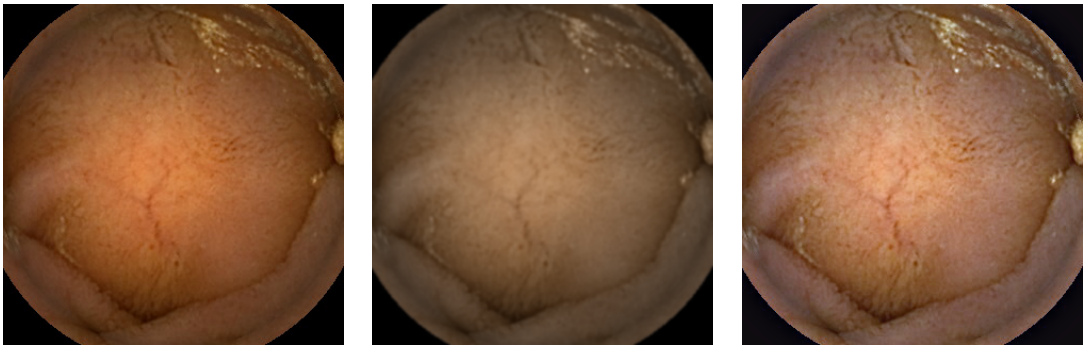
Considering the task at hand, as well as the issues it entails, a general pipeline was developed to reach the desired goal (see Figure 3.1). The system will need a set of inputs, which includes a set of capsule endoscopy videos, along with topographical annotations related to major small bowel landmarks, information about the camera itself (intrinsics matrix), and general information about the characteristics of the small bowel (such as average length and diameter). This data will be combined and passed through a homography estimation system, in order to estimate the homography between consecutive images. Using this information, the displacement between consecutive frames can be computed, as well as the total displacement (in relation to the beginning of the small bowel) in each frame of the video. The videos can be converted into a sequence of only small bowel images, using the topographical annotations. The intrinsics matrix of the camera will play a part in converting said homography into a translation value, which can then be validated by comparing it to the average characteristics of the small bowel.

Regarding the homography estimation system, both the image itself and the timestamp associated with each frame will be used to perform the computation. The path of deep learning was chosen based on the potential already shown in previous works [42, 61], and given the large amount of data made available. The network used will be unsupervised, to accommodate the limitations of the dataset, and maximise possible applications of the system in the endoscopic area.

## 3.3 Image Pre-processing

VCE frames in their raw state retain some unsuitable characteristics for future feature detection and matching between images. For example, since the endoscopic capsule has its own light source, the same features may be illuminated differently in consecutive frames, giving them different intensity values in the final images. Additionally, some features are very subtle, being easily mistaken for background on first glance. Considering these problems, a pre-processing technique was applied to the raw images, in the attempt to make them more homogeneous among themselves, as well

as to highlight potential features. Examples of the expected result for each of the processes are shown in Figure 3.2.



(a) Original image (b) Anisotropic diffusion (c) Homomorphic filtering

Figure 3.2: Examples of the pre-processing techniques applied to the dataset.

### 3.3.1 Homomorphic Filter

First, a homomorphic filter, using a DFT, was applied (see Section 2.4.1.1). Thus, the images were represented as a sum of two parcels: illumination, $i(x,y)$, and reflectance, $r(x,y)$. The DFT is then performed, converting the images to the frequency domain, where a high pass filter is applied, to eliminate the lower frequencies (illumination parcel). Additionally, two scaling factors are set, to adapt the relevance of each of the parcels, meaning that the final image will be defined by:

$$I(x,y) = \gamma_1 \times i(x,y) + \gamma_2 \times r(x,y) \tag{3.1}$$

By applying an inverse DFT, the image is converted back to the spatial domain and is ready to be used in the remaining steps. This technique was tested, as it seemed promising for the homogenisation of image lighting throughout the entire VCE. The effects of the homomorphic filter can be seen in Figure 3.2c.

### 3.3.2 Anisotropic Filter

The second technique used was anisotropic diffusion, or PM diffusion (see Section 2.4.1.2). In this case, the main goal was to highlight possible features, while smoothing texture-less regions. Thus, the image is filtered based on the gradient at each point of the image: for areas with a high gradient (edges, corners) the smoothing will be minimal, while areas with low gradients will produce a higher response of the filter. This effect can be seen in Figure 3.2b. The filter response $g$, determined by the gradient $||\Delta I||$, is given by:

$$g(||\Delta I||) = \frac{1}{1 + (||\Delta I||/K)^2} \tag{3.2}$$

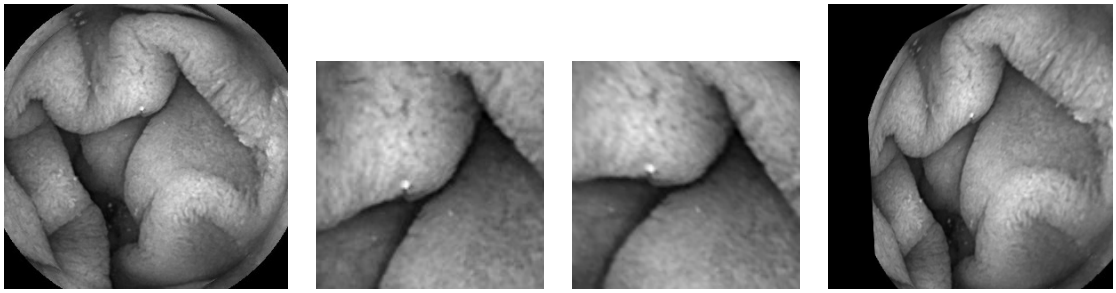
The process is iterative, and depends on the value of the constant $K$, set by the user. A higher number of iterations will mean more smoothness in the image, but it can lead to the loss of relevant features. In a similar way, a higher value of $K$ will provide a higher filter response for low gradient values, increasing the amount of smoothing performed.

## 3.4 Synthetic Data Generation

Given the intention to use deep learning techniques, the ability to have a large amount of training data is paramount. This data should be fit not only to train the network, but also provide enough information to allow the validation of the process. This means that, although the implemented networks are unsupervised, some form of ground truth labels should be available to adequately test its performance. With this in mind, the data generation process described in Section 2.3.4 was applied.

Through this process, the available images can suffer a distortion defined by an arbitrary 4-point homography parameterisation. Since the homography applied is known, this method provides 2 image patches, and the associated ground truth homography. Thus, the images can be used as input for the networks and the label can validate the results provided.

The technique itself starts by selecting a patch of the original image, at a given position $p$. The corners of the selected patch are then perturbed by an arbitrary amount within the interval $[\rho, -\rho]$. This originates a 4-point homography, which is then inverted and applied to the entire original image, resulting in the warped image. A second patch, at the same position $p$, is then cropped from the warped image, and both patches can be fed into the desired network. This process can be performed multiple times for each image, originating any number of training examples, given the high number of patches and distortions that can be applied. An example of the outcome of this process can be seen in Figure 3.3.



(a) Original image. (b) Original patch, cropped from original image. (c) Perturbed patch, cropped from perturbed image. (d) Perturbed image.

Figure 3.3: Example of the data generation process on VCE image.

## 3.5 Unsupervised Neural Network

The network selected as a basis for this work is the unsupervised version of HomographyNet [8], initially proposed by Nguyen et al. [38]. HomographyNet was chosen for its already proven efficacy on VCE frames [42].

The network takes 3 inputs: a pair of grayscale image patches ($128 \times 128 \times 2$), related by a homography (see Section 3.3); the original grayscale image ($320 \times 320$) to which the homography will be applied; and the coordinates of the position $p$ of the patches ($4 \times 2$). Additionally, the warped image is provided to the network, to compute the loss function. Technically, this loss function is not considered unsupervised, since it uses the warped image as ground truth label. The loss function compares the ground truth warped image, to the one generated by the network (based on the estimated homography). However, since it does not require the ground truth homography to be trained, using only image information, the network is referred to as unsupervised.

As for the architecture, seen in Table 3.1, the network is composed of 8 convolutional layers, with $3 \times 3$ kernels, with 64 filters in the first 4 convolutional layers and 128 in the last four. Every two convolutional layers, a batch-normalisation is performed, along with max pooling, with kernel $2 \times 2$ and stride 2. After the convolutional part of the network, a dropout layer ($p = 0.5$) is added, followed by a fully connected layer with 1024 units. To complete the 4-point homography estimation, final dropout and fully-connected layers, this time with only 8 units, are used. If the intention was to apply a supervised method, then the network would end here. However, as described in Nguyen et al. [38] (see Section 2.3.5), two new layers have to be added, to allow the differentiable transformation of the original image according to the estimated homography. With that in mind, a DLT (that converted the 4-point homography to the classical homography parameterisation) and a spatial transform (that warped the original image according to said homography) are applied, and the result is used to compute the loss function. Thus, the final output of the system is a $320 \times 320$ image, corresponding to the original image warped according to the estimated homography.

### 3.5.1 Neural Network with Timestamp

A second version of the network was also implemented, in order to take into consideration the time elapsed between consecutive VCE frames, which can be determinant for the amount of displacement, and therefore homography, that happens between the two moments.

The only alterations, when compared to the original network, was the addition of a fourth input: the time elapsed between the two frames, and the concatenation of that information to the vector resulting from the convolutional part of the network. The remaining steps stayed the same, and the output was identical.

## 3.6 Camera Calibration

One of the essential inputs of the proposed pipeline is the camera intrinsics matrix. However, we do not have direct access to this information, and so camera calibration of the PillCam SB3 needs

Table 3.1: Unsupervised HomographyNet architecture.

| Layer | Filters | Kernel/Pool Size | Stride |
|---|---|---|---|
| Convolutional | 64 | 3x3 | 1x1 |
| Convolutional | 64 | 3x3 | 1x1 |
| Max Pooling | | 2x2 | 2x2 |
| Batch Normalisation | | | |
| Convolutional | 64 | 3x3 | 1x1 |
| Convolutional | 64 | 3x3 | 1x1 |
| Max Pooling | | 2x2 | 2x2 |
| Batch Normalisation | | | |
| Convolutional | 128 | 3x3 | 1x1 |
| Convolutional | 128 | 3x3 | 1x1 |
| Max Pooling | | 2x2 | 2x2 |
| Batch Normalisation | | | |
| Convolutional | 128 | 3x3 | 1x1 |
| Convolutional | 128 | 3x3 | 1x1 |
| Max Pooling | | 2x2 | 2x2 |
| Batch Normalisation | | | |
| Flatten | | | |
| Dropout (p=0.5) | | | |
| Fully connected (1024 units) | | | |
| Dropout (p=0.5) | | | |
| Fully connected (8 units) | | | |
| DLT | | | |
| Spatial transform | | | |

to be performed.

This process is done based on multiple images of a chessboard, which were provided along with the VCE dataset (see Figure 3.4). A simple calibration API [3], developed as a wrapper to the already existing *OpenCV* tools, was used. This tool is able to perform calibration from 3 different pattern types: chessboard, symmetric circular grids, and asymmetric circular grids. The only inputs required are a set of images of the desired pattern, and its measurements (in this case, each chessboard square was 2 mm).

Thus, 209 images were used to obtain a first approximation of the intrinsics matrix, with a mean reprojection error of 0.46997 pixels. However, given the low image quality and the high error obtained in the first iteration, a second attempt was made, using only the images with a reprojection error below 0.1 in the first iteration. This originated the final intrinsics matrix $K$ result (see Equation 3.3), used throughout the rest of the work. The final iteration of the matrix originated a mean reprojection error of 0.06832 pixels.

$$K = \begin{bmatrix} 164.83214164 & 0 & 163.23021052 \\ 0 & 165.2686061 & 159.40966211 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Figure 3.4: Example of chessboard image, captured with PillCam SB3.

## 3.7 Capsule Displacement Estimation

The final step in the developed system is the computation of the displacement of the capsule between consecutive frames. This was done based on the classical homography parameterisation estimation, provided by the network after the application of the DLT layer. The process to obtain the per-frame displacement is represented in Figure 3.5.

The classical homography parameterisation can be written as a function of both the camera intrinsics matrix and the translation and rotation the camera suffered between the two frames at hand (see Section 2.2.2). Thus, a set of solutions for the 3D translation can be obtained by decomposing the 2D homography into its constituting parts: translation ($t$), rotation ($R$) and normal vector ($n$). This gives a total of 4 possible combinations of translation, rotation and normal vector. By using the correspondence points available, in the form of the 4-point homography estimated by the network, these solutions can be reduced, using the method described in Section 2.2.2.4.

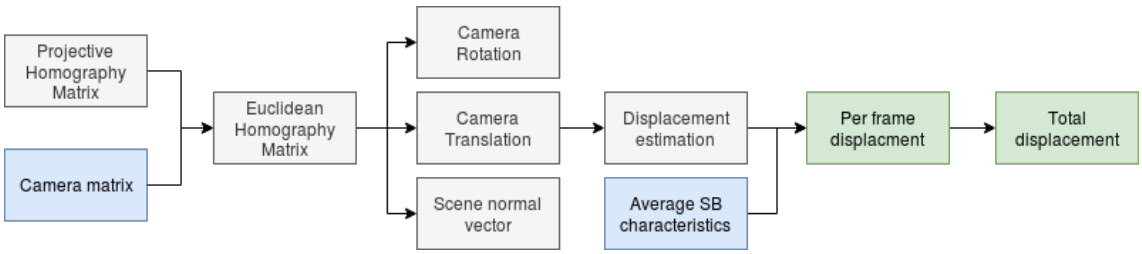



Figure 3.5: Schematic representation of the process of displacement computation, base don estimated homography. In green, the needed inputs; in blue, the main outputs.

Essentially, each possible solution will be tested against the correspondence points, in order to assess if said points are in front of the camera. The solution that allows all the points to be in front of the capsule will be the final translation. Although the translation is provided in 3 dimensions, only the displacement along the $z$ axis is considered, since it translates into movement of the capsule forward in the GIT.

A simple post-processing method is applied to eliminate unreasonable displacement estimations. Since the camera can move in both directions, we should only consider that a displacement estimation is nonsensical when it surpasses the distance the camera can travel at its maximum speed. Assuming a case where the camera remains transmitting at 6 fps throughout the entire video (maximum resolution possible), then the time $t$ elapsed between two frames is:

$$t = \frac{1}{6} = 0.17s \tag{3.4}$$

By dividing the number of frames $N$ in the VCE by the transmission rate $R$ (6fps), we obtain the least time possible for the capsule to travel the entire small bowel length ($T$). This information, combined with the average length $L$ of the small bowel (around 4 meters, although extremely variable) [13], gives us the velocity $v$ of the capsule in meters per second:

$$v = \frac{L}{T} = \frac{L}{\frac{N}{R}} \tag{3.5}$$

Finally, the maximum displacement $d$ between consecutive frames accepted will be:

$$d = vt \tag{3.6}$$

Additionally, there are images that do not overlap, and so the estimated homography as no physical meaning. In these cases, it is found that none of the translation-rotation-normal combinations allow the reference points to be in front of the camera. Thus, no translation value is found for these pairs.

The average displacement of the capsule between frames is estimated by dividing the average small bowel length by the number of frames of the present VCE. This will be the value used to replace any unreasonable estimations found, and in cases where no solution exists.

## 3.8 Summary

The developed system is comprised of five main steps. First, a data generation technique is applied, to obtain enough images to use deep learning methods later on. Next, the image pre-processing methods proposed focus on the improvement of classical VCE characteristics, while considering how the images are going to be used in the following steps. Thus, methods for illumination variance, and feature highlight are proposed. For the homography estimation step, an unsupervised

neural network is proposed, in two versions: one using only image information, and one considering the time elapsed between frames. It is expected that the time information will help improve results, given the dependency between time and amount of movement. After performing the camera calibration, we are then able to convert the homography information into a single translation value, by using the point correspondences available. Finally, a simple post-processing method is applied to eliminate any nonsensical displacement values, thus obtaining the final per frame displacement.

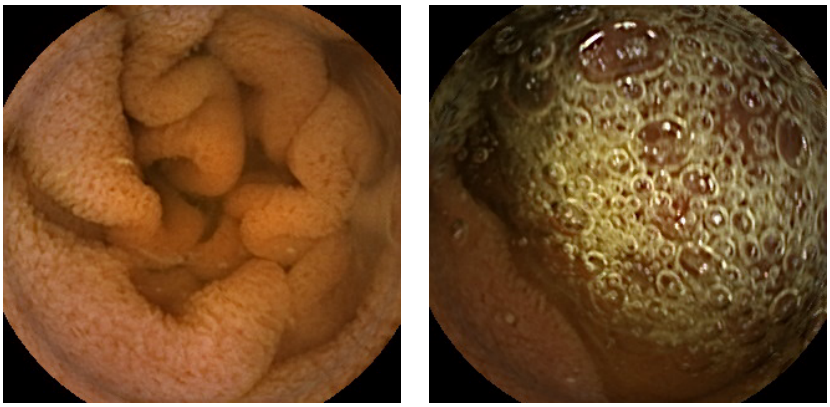
# Chapter 4

# Results

## 4.1 Dataset Characterisation

The dataset used in this work was constituted by 261 VCE, recorded using PillCam SB3 [35]. These videos were provided in a document format compatible with *Rapid Reader* software [34], developed specifically to analyse VCE videos from PillCam SB3. The PillCam SB3 has a frame rate of 2 to 6 fps, according to the speed of the camera. It has a length of 26.2 mm and a diameter of 11.4 mm, weighing 3g. It also has 4 white light emitting diodes on each side, to illuminate the GIT has it goes through it. It has an operating time of 8 hours or more [33].

Of the 106 videos, 30 were accompanied by topographical annotations, indicating the beginning of the small bowel (pyloric sphincter) and the beginning of the large intestine (ileocecal valve). Each video had a duration of about 9 hours, with varying frame rates throughout this time, and an average of $15,000$ frames. Several frames of videos were obscured by gastrointestinal content and air bubbles, but the vast majority of the frames retained an acceptable quality, as seen in Figure 4.1.



(a) (b)

Figure 4.1: Image examples from the private dataset. (a) Clean frame, with good quality and perfectly visible GIT wall. (b) Frame obscured by air bubbles, hindering the visualisation of the GIT.

Table 4.1: Summary of the advantages and disadvantages of the proposed pre-processing methods.

| | **Anisotropic Diffusion** | **Homomorphic Filtering** |
|---|---|---|
| **Feature highlight** | Highlights large features | Highlights small details |
| **Smoothing** | Smooths large textureless areas | Reduces smoothness in large areas |
| **Illumination Elimination** | Efficient | Not efficient |
| **Computation time** | Very fast | Moderate |

### 4.1.1 Video to Image Conversion and Timestamps

The VCE were provided in the form of full videos. However, for the application of the designed pipeline, the conversion to image sequence was necessary. To perform the conversion, the *SensArea* software [54] was used, which exported each video as an image sequence, with $320 \times 320$ size. Performing this process on the videos with attached topographical information, a total of $402,711$ small bowel frames, from 30 VCE from 30 different patients.

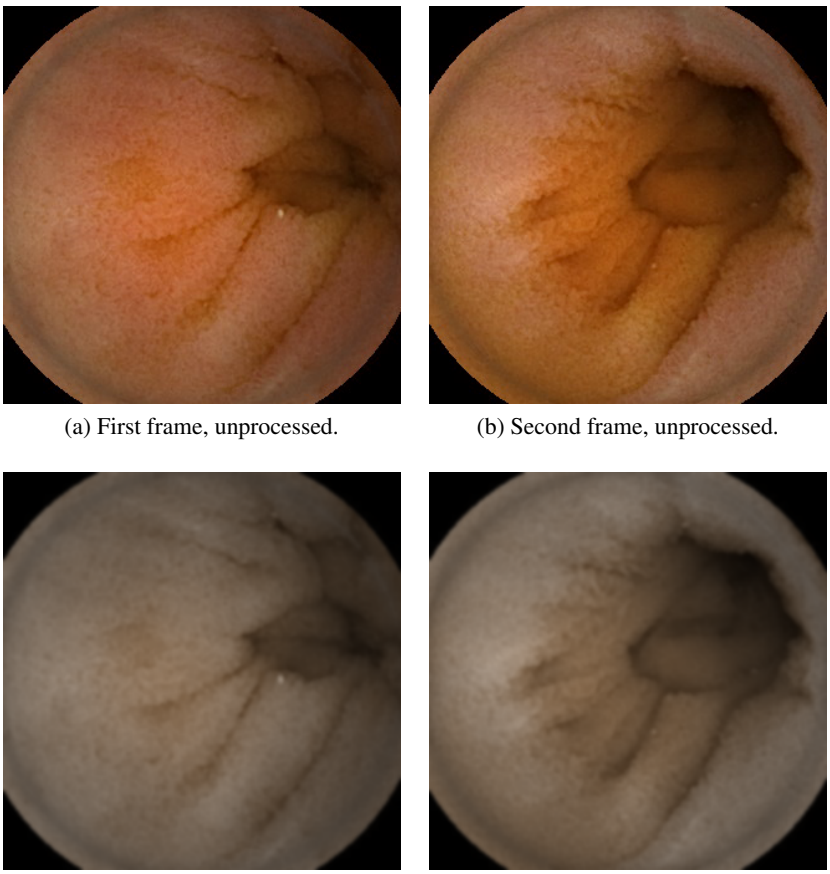
Additionally, 8 VCE without topographical information suffered further processing. These videos were used to test the possibility of including the time elapsed between consecutive frames in the estimation of the homography between them. To allow such a test, the timestamps associated with each frame had to be extracted. Thus, *MacroToolworks* [43] was used to record the simple motion of selecting the timestamp of the first VCE frame, copying it to a spreadsheet and advancing to the following frame, all in the *RapidReader* [34] program. The tool then repeated the motion, registering the timestamp of every frame of the video. After this process, the video was converted into an image sequence, as described above. This resulted in an additional $80,655$ VCE frames, each with associated timestamp.

## 4.2 Image Pre-processing

The evaluation of the pre-processing techniques (anisotropic diffusion and homomorphic filtering) was performed in two steps. First, both methods were applied to a set of example images, belonging to a single VCE, and then visually compared, to determine which technique better suits the dataset at hand. The advantages and disadvantages of each process are summarised in Table 4.1.

For the anisotropic diffusion, 5 iterations were computed, using $K = 50$. These parameters were chosen empirically, based on the images produced. The anisotropic diffusion worked better in the homogenisation of the images, eliminating illumination variances throughout image sequences. Figure 4.2 shows two consecutive images, both before and after being submitted to the anisotropic diffusion. In the first frame (Figure 4.2c) the walls of the small bowel adopt a pink hue, which in the following frame becomes more orange, due to the shift in the illumination source.

After the application of the filter, these differences disappear, and the walls become more smooth. Although this smoothing leads to the loss of some small scale features, the more relevant ones, such as the folds around the centre of the images, remain unperturbed.



(a) First frame, unprocessed.

(b) Second frame, unprocessed.

(c) First frame, after anisotropic diffusion.

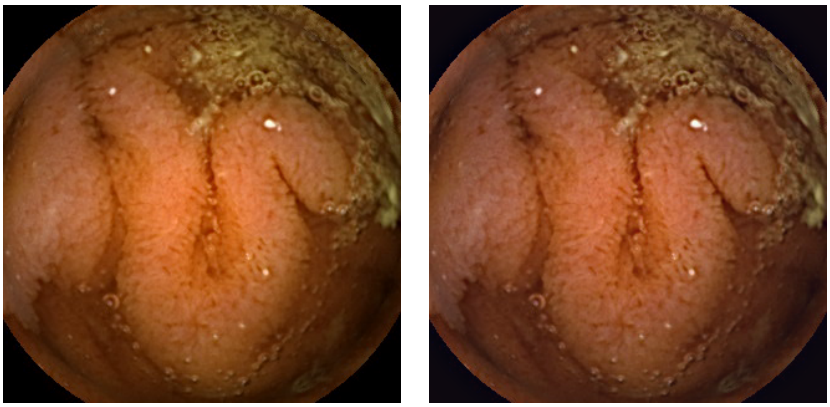
(d) Second frame, after anisotropic diffusion.

Figure 4.2: Results of the anisotropic filter on consecutive VCE frames.

However, the homomorphic filter performed better in the highlighting of small features, that were not as visible on the original image, as perceptible on Figure 4.3. The parameters $\gamma_1 = 1.5$, and $\gamma_2 = 2$ were used.

The decision was to use the anisotropic diffusion, mainly because of the use of an unsupervised network. Since the training of the network is based on photometric loss, it is ideal that the intensity values of corresponding pixels remain the same throughout different images. Additionally, it can be expected that the main features learned by the network will be on a large scale. This means that it should be able to learn effectively, even if the pre-processing technique used does not contribute to feature highlight.

The second step in the validation of the pre-processing techniques was the comparison of the network with and without their application. Thus, the anisotropic diffusion was applied to the entire dataset, and the images were then used in the remaining pipeline. The pipeline was also



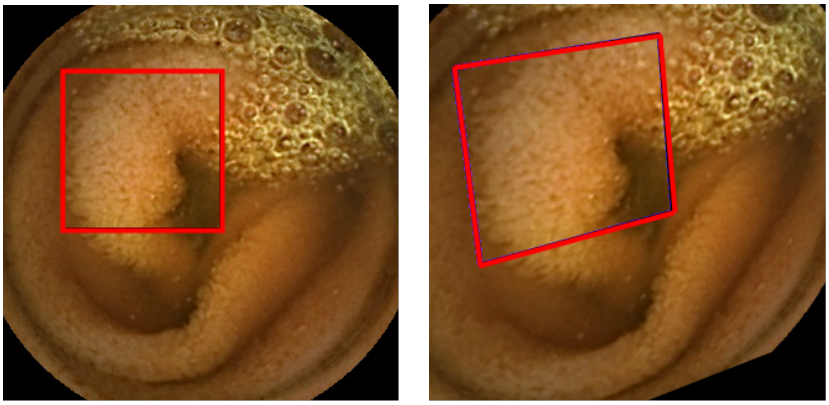
(a) Original image. (b) Image after homomorphic filtering.

Figure 4.3: Results of the homomorphic filter on sample VCE frame.

tested with the same set of images, but without pre-processing, to serve as a control group.

## 4.3 Data Generation

The data generation technique described in Section 3.4 was applied to the images in the dataset, to allow validation of the proposed network. Although the network itself does not need ground truth labels to be associated with each image, these labels are indispensable to provide error metrics for the system. Thus, data generation becomes a key part of the pipeline. In this case, it was applied to the training images, generating ten examples per image, giving a total of $2,243,190$ training image pairs, with a maximum perturbation $\rho = 32$ pixels. The results of this generation can be seen in Figure 4.4.



(a) Original image (b) Distorted image

Figure 4.4: Example of the data generation technique applied to the dataset. (a) Original image, with the selected patch in red. (b) Distorted image, with the distorted patch in red.

The use of such a technique proves to be extremely useful in situations where the amount of data (or, in this case, labelled data) is too small to use deep learning techniques. This method

allows for the creation of a practically infinite number of training examples, by combining different patches with different corner distortions. In this work in particular, although the network does not require ground truth information, the labelled images provide validation, allowing the comparison between the homography estimated by the system to the one applied.

## 4.4 Homography Estimation Network

Several experiments were performed with the unsupervised neural networks, in order to optimise the displacement estimation system. The first step was to explore the homography estimation, and optimise the process. To do so, a first test was used to compare the results with and without image pre-processing (see Section 4.4.1), then the network was trained further to minimise homography estimation error (see Section 4.4.2), and finally the timestamps of each frame were introduced to assess their usefulness (see Section 4.4.3).

The error associated with homography estimation will be, henceforth, translated by the Mean Average Corner Error (MACE) (see Equation 4.1). This metric is associated with the 4-point homography, and relates to the distance, in pixels, of each corner of the estimated patch to the ground truth corner.

$$MACE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\sum_{k=1}^{4}\left(p_{i,k}-q_{i,k}\right)^2} \tag{4.1}$$

In Equation 4.1, $n$ represents the total number of samples (test pairs), $k$ iterates through each corner of the sample patch, $p_{i,k}$ is the ground truth location of the patch corner, and $q_{i,k}$ the estimated location of the patch corner.

### 4.4.1 Pre-processing versus Raw Images

In this experiment, all 30 VCE with topographical annotations were used, divided into a training set, comprised of two-thirds of the set (224,319 frames, 21 VCE), and a test set, containing the remaining VCE (116,006 frames, 9 VCE). Data generation was applied to all images, producing a single training pair per frame. Each set was used in duplicate: a version with pre-processing applied to the images, and one without. Each group of images was treated in the same way, to allow comparison between the results found.

The network was trained with the images from the training set, using a stochastic gradient descent, with a batch size of 128, and an Adam optimiser, with learning rate 0.001. A total of 3,000 epochs were performed. After training, the test set was put through the network, and the 4-point homography estimation obtained.

The pre-processing proved to be extremely useful, lowering the MACE from 35 pixels to 21. This decrease is probably due homogenisation of the images, allowing the network to correctly predict correspondence points between the image pairs. With this in mind, all the tests that followed were preformed using pre-processed images.

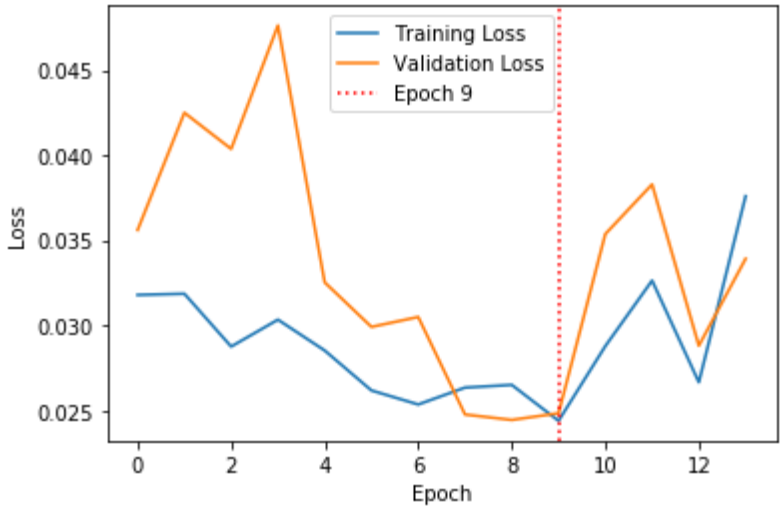



Figure 4.5: Evolution of the loss function of the Unsupervised HomographyNet. In blue, the loss value of the training set. In orange, the loss value of the validation set. In dotted red, the final epoch considered.

### 4.4.2 Baseline Results

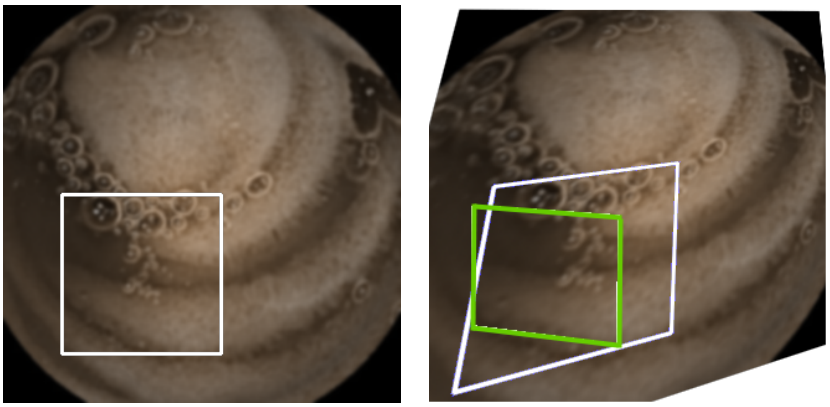
The baseline network (see Section 3.5) was trained using artificially generated images and then tested with both synthetic and real data, to validate it.

#### 4.4.2.1 Synthetic Data Tests

After determining that the use of pre-processed images would be beneficial, the baseline network was retrained, to better optimise the results. The network used the same hyper-parameters seen in Section 4.4.1. However, some changes were made. First, the entire dataset was divided equally into three sets: train, validation and test. The training set suffered data generation, producing 10 training pairs per frame (for a total of $1,114,830$ pairs), and was used to train the network. The validation set went thorugh the same data generation, giving $1,077,350$ pairs to monitor the evolution of the loss value throughout the training of the network. The network was trained using an early-stop mechanism, which stopped the training of the network when no improvement on the loss value of the validation set was detected. Figure 4.5 represents the value of the loss function on both the train and validation sets, throughout 14 epochs, showing that the ideal number of iterations is 9, which represents the lowest loss for the validation set, to prevent over-fitting of the network to the train set. It is important to note that due to the loss function not being directly related to the homography estimation, it becomes highly fluctuating, and difficult to converge. Thus, the homography predictions were monitored, along with the loss function, to ensure good quality estimations, and an adequate number of training epochs.

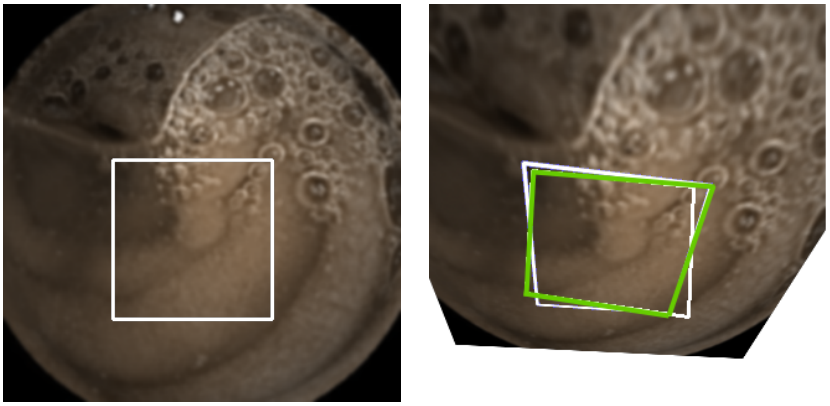
To evaluate the network performance, the remaining videos, belonging to the test set, were put through the data generation technique, to provide one test pair per frame (for a total of $121,107$ frames). This set was used to compute the final MACE of the network.

The test set provided very positive results, with a MACE of 19 pixels. Examples of both successful and failed estimations can be seen on Figure 4.6.



(a) Worst result, original image

(b) Worst result, distorted image



(c) Best result, original image

(d) Best result, distorted image

Figure 4.6: Examples of the homography estimation network. Worst result: (a) Original image, with the selected patch in white; (b) Distorted image, with the distorted patch in white, the predicted path in green. Best result:(c) Original image, with the selected patch in white; (d) Distorted image, with the distorted patch in white, and the predicted path in green.

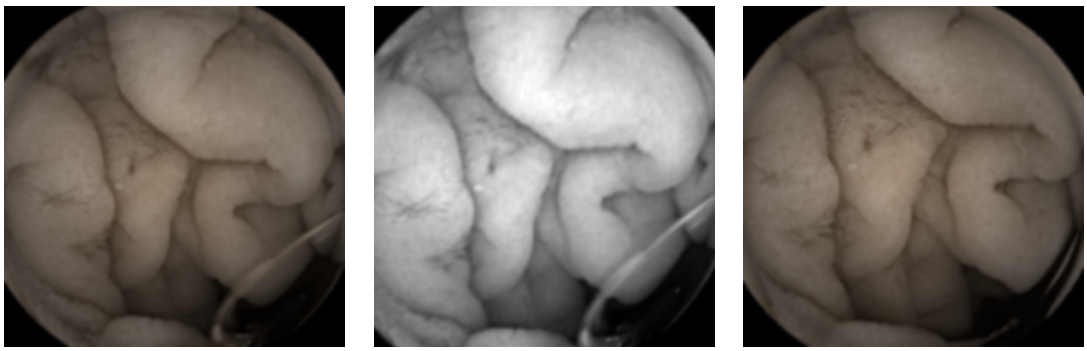
The network is able to fairly estimate the homography between frames, showing greatest difficulty in cases when the distortion is more pronounced. A good example of this phenomenon is depicted in Figure 4.6a and 4.7e. However, for most images, the process performs rather well, as is the case with Figure 4.6c and 4.7e. The difficulty presented in some of the images may be related to the erratic loss function. Such a behaviour makes the process of network convergence very hard to identify, and thus the results may not be optimised.

The produced results are very promising. Although Pinheiro et al. [42] was able to obtain a better MACE (2 pixels), the current results establish unsupervised methods as a feasible solution for homography estimation. This proof is a breakthrough in itself, even if the metrics are not yet able to reach the desired standard.

#### 4.4.2.2 Real Data Tests

Given the positive results in the synthetic data tests, the testing was extended to real data. In this case, only a qualitative assessment of the network is possible due to the lack of ground truth labels.

The tests on real data were performed on the same VCE as the synthetic tests. This time, two consecutive images (A and B) are used in their entirety, resized to $128 \times 128$, and fed into the network, as done previously with the image patches. The network can then compute an approximation of the homography matrix, and warp image A according to it, obtaining an approximation of image B. this process is depicted in Figure 4.7.



(a) Good result, original image A (b) Good result, warped image (c) Good result, original image B

(d) Bad result, original image A (e) Bad result, warped image (f) Bad result, original image B

Figure 4.7: Result of homography estimation and warping for real VCE images. Example of a good estimation: (a) Original image A, (b) Image A warped to approximate image B, (c) Original image B. Example of a bad estimation: (a) Original image A, (b) Image A warped to approximate image B, (c) Original image B.

From the first example of Figure 4.7, there are two main conclusions to be drawn. First, generally, the network seems to be able to recognise the forward movement of the capsule, translated in the images by a zoom in. Secondly, although some rotation is applied to the original image, the network seems to underestimate it. This is particularly clear in the left section of the frames, which is clearly higher on the original image B than in the estimated image. This frame set, along with the conclusions drawn, represent the majority of the results obtained. However, some images, as is the case with the second example of Figure 4.7, do not produce good results, in spite of being seemingly simple transformations. A possible justification is the amount of air bubbles present in

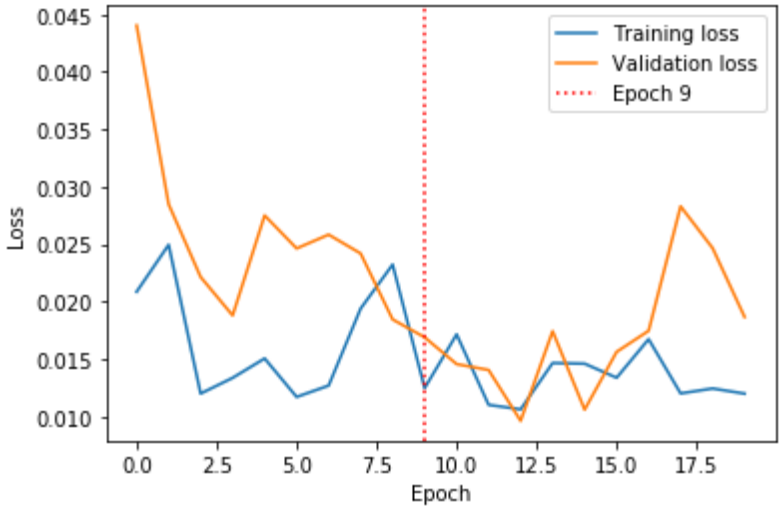



Figure 4.8: Evolution of the loss function of the Unsupervised HomographyNet with timestamp. In blue, the loss value of the training set. In orange, the loss value of the validation set. In dotted red, the final epoch considered.

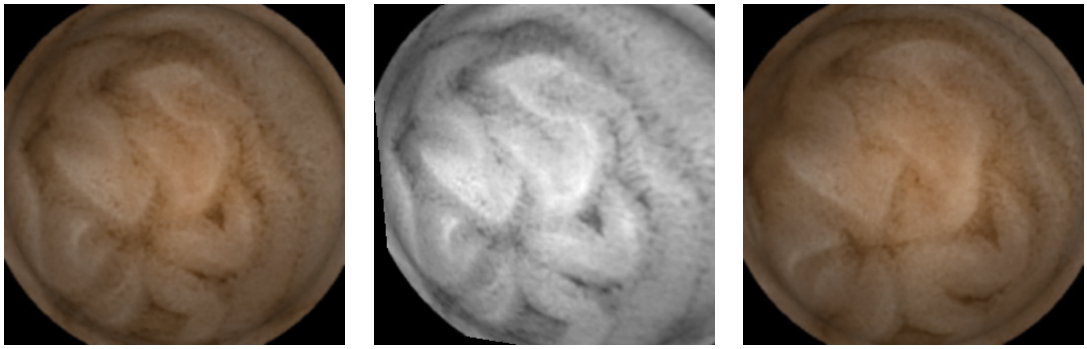
the images, which hinders the detection of image features belonging to the small bowel itself. Finally, when the consecutive frames do not overlap, the network fails to produce acceptable results, as would be expected.

An important factor to take into account when performing this kind of test is that the homography model is simply an approximation of the transformation that occurs between the consecutive frames, and it is not enough to describe it. This becomes evident when analysing real data results. Although the network is able to approximate the type of movement performed between frames, it will never be completely successful in morphing one frame into the next. The application of a different model could be very beneficial for the final results.

### 4.4.3 Timestamp Influence

The second version of the network, which included a timestamp for each VCE frame, can not be trained using artificial data, since the data-generation process does not include a synthetic timestamp for each artificial example generated. This means that the network had to be trained and tested using only real data, and so the evaluation was completely visual and qualitative. Once again, the data was divided into 3 sets: train (39,081 frames), validation (27,042 frames), and test (14,532 frames). The training was performed in a similar way as with the baseline network, with the validation set being used to perform early stopping. The loss function shows a very erratic behaviour throughout the training process (see figure 4.8), as expected (see Section 4.4.1), due to the fact that the loss function is not directly related to the homography estimation provided by the network. For this reason, the final epoch considered was chosen based not only on the loss for the validation set, but also on the images produced. The final epoch considered was epoch 9.

The analysis of the results shows that this version of the network is not as effective in approximating the transformation between consecutive images. Most images are warped in an exaggerated way, which is clearly visible in the second example of Figure 4.9. This exaggeration is verified both with excessive zoom in and zoom out. There are, however, some very accurate approximations, as seen on the first example of Figure 4.9. In this case, it is visible that the network is able to recognise not only the approximation but also the slight rotation that occurs between the two frames.



(a) Good result, original image A (b) Good result, warped image (c) Good result, original image B

(d) Bad result, original image A (e) Bad result, warped image (f) Bad result, original image B

Figure 4.9: Image sequence submitted to the unsupervised network, and respective results. In the left column, first original images; in the centre column, first imaged warped into second image; in the right column, second original image.

The underwhelming results may have several explanations. First, the amount of data used for training is considerably less, since data generation is not possible in this case. These results show the importance that the amount of data available is crucial when it comes to deep learning approaches, to ensure that the network is not overfitted to that specific set of images. A larger amount of data will generate a more generalisable network. This leads us, once again, to the conclusion that an unsupervised solution is the optimal choice for VCE localisation, since it does not further restrict the type of images that can be used.

Another important factor is that more time elapsed between frames does not always represent more movement between them. On the contrary, there are cases when the capsule moves at a very low speed, remaining seemingly still. These moments will lead to very low frame rates, with

several seconds passing without movement being registered. Additionally, the elapsed time used is given in seconds: a smaller unit (milliseconds) would better represent this information.

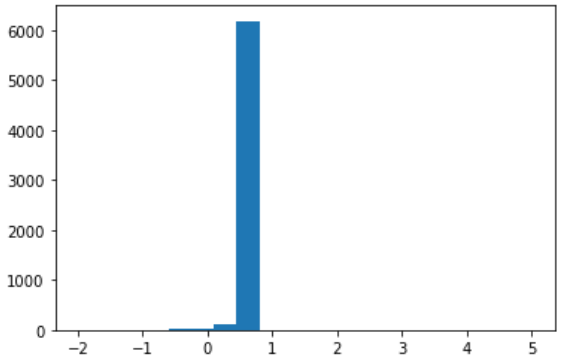
Finally, we must consider the network itself. Although HomographyNet has been used successfully for homography estimation, both in VCE applications [42] and in other types of problems [8], it may not be the most suitable network to assess movement between consecutive frames. For example, the use of a DispNet may be useful in this case, since it was designed specifically to detect disparity between images. The use of a network that includes inception modules may also be beneficial, given the various scales at which it operates, allowing the detection of more varied features.

## 4.5 Capsule Displacement Estimation

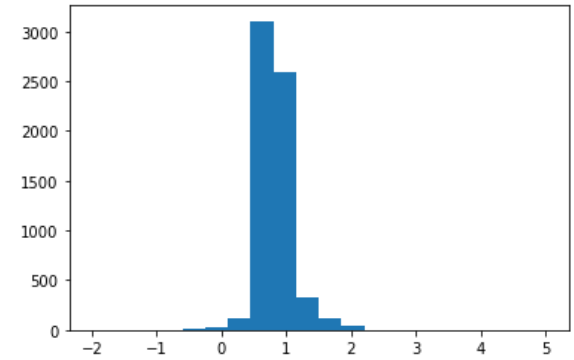
With the homographies of each VCE computed, the following step was the estimation of the displacement associated with said homography matrix. This process was applied to the VCE with associated timestamps, in order to compare the outcome for each version of the network. However, before this comparison was made, preliminary the displacement results were computed, to analyse the effectiveness of the post-processing step, and the overall validation of the displacement estimation.

When analysing the values of displacement estimated, after the post-processing method was applied, it is clear that most instances of the per-frame displacement have been replaced by the average value, in all the tested videos. Upon further inspection, we conclude that the maximum value computed for each VCE is actually a fairly low number, very close to the average of the video, and several frames surpass it. By increasing the maximum value of displacement allowed to, for example, 2 millimetres, the number of replaced instances lowers immediately, as can be seen in Figure 4.10. Figure 4.10 shows three histograms, representing the frequency of per-frame displacement results obtained of a real VCE video segment, with varying maximum displacements. Analysing Figure 4.10a, we can see that the value $d$ computed is not adequate to be used as a threshold, since it reduces the histogram to practically a single value, making all the previous estimations irrelevant. However, it is clear that the post-processing step is key. Without it (see Figure 4.10b), the result would be an exaggerated estimation of the small bowel length (or total capsule displacement), due to the high percentage of frames that do not overlap, and thus lead to nonsensical homography and displacement estimations. The best option seems to be to use a maximum displacement of 2 millimetres, since it provides the best balance between maintaining the original estimated values, while filtering most of the nonsensical values. It is also important to note that all histograms present a peak at the mean displacement value, given the high amount of consecutive frames that do not produce valid translation solutions, and therefore need to adopt the average displacement. These results led to the use of a maximum displacement of 2 millimetres throughout the remaining tests.

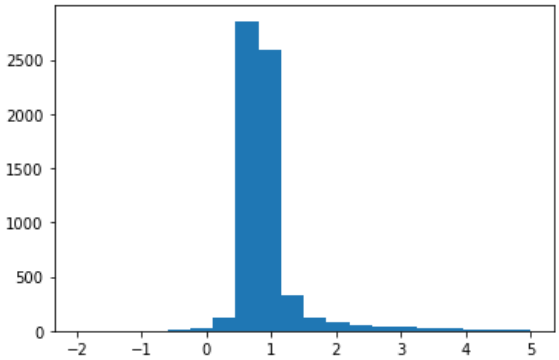
In Figure 4.11, the advances of the capsule throughout the frames of a test VCE are seemingly uniform. In this example, the final estimated value for the length of the small bowel is of $5,223$




(a) Maximum displacement = $d$; Total displacement = $3903mm$




(b) Maximum displacement = $2mm$; Total displacement = $5223mm$




(c) No maximum displacement applied; Total displacement = $8484mm$

Figure 4.10: Histograms of per frame displacement in test VCE, with varying maximum displacement, for an average displacement of $0,63mm$. (a) Maximum displacement set to $d$, according to Section 3.7. (b) Maximum displacement set to $2mm$. (c) No post-processing applied.

meters. This number is slightly higher than expected, but still falls within normal lengths for the small intestine. A smaller portion of the video can be studied to understand the estimations, as seen in Figure 4.12.

It is clear from Figure 4.12 that the system is able to learn the amount of movement, at least in some cases, of the capsule. While the estimated displacement from frame 0 to frame 1, and from frame 1 to 2 is roughly the same, as proved by the images themselves, the displacement estimated for the last pair is considerably lower. When analysing the frames, this estimation seems to make sense, since there is barely any visible movement between the last pair of frames, indicating very low translation, as estimated.

One important trait of the system should be to recognise in which direction the capsule is moving. When further analysing the tested VCE, it is clear that the system accepts negative values of displacement, meaning that it does recognise the backward movement of the capsule. This is the case seen on Figure 4.13, which shows the capsule performing backward movement between frames 0 and 1, translated in a negative per-frame displacement estimation.

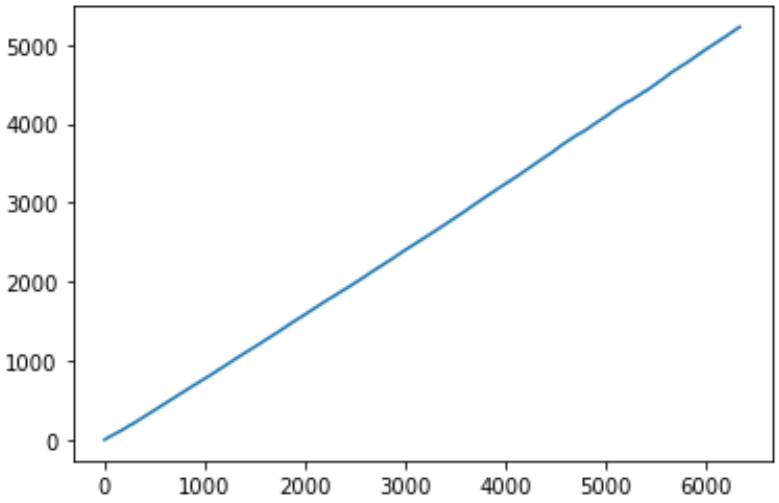
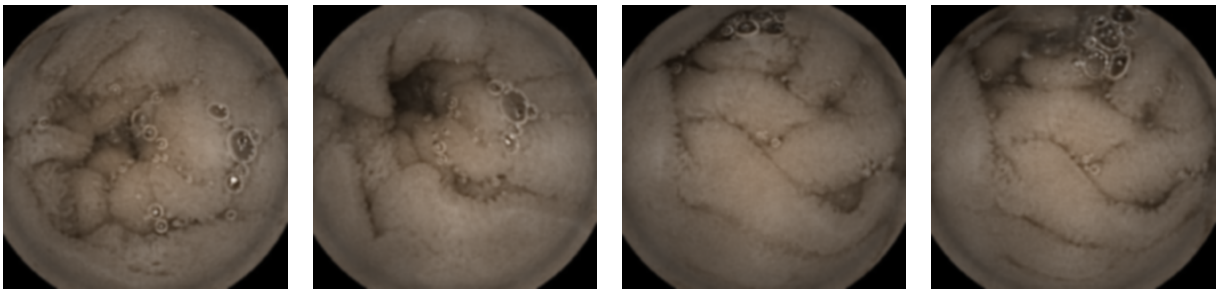
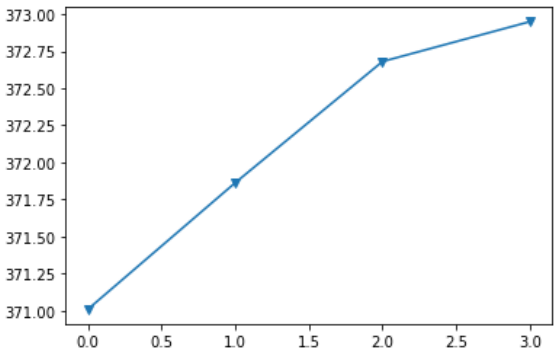



Figure 4.11: Total capsule displacement throughout test VCE frames.



(a) Frame 0 (b) Frame 1 (c) Frame 2 (d) Frame 3

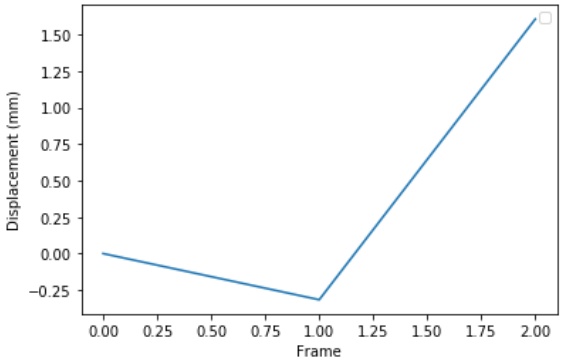



(e)

Figure 4.12: Image sequence and respective displacement estimation.

The test VCE produced total displacement values between $4,121$ and $5,223$ millimetres, all within the normal range of small intestine length, but somewhat above average, indicating the system may slightly overestimate the amount of movement.

(a) Frame 0 (b) Frame 1 (c) Frame 2




(d) Per-frame displacement, obtained from baseline network.

Figure 4.13: Image sequence with backward capsule movement and respective displacement estimation.

#### 4.5.0.1 Timestamp influence

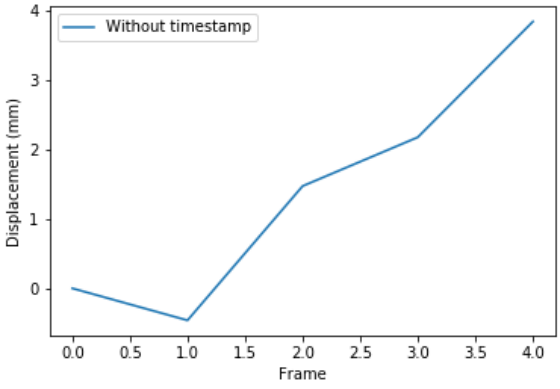
Given all the seemingly positive experiments presented thus far, we may move on to the comparison of the displacement estimation with and without the timestamp information. This test was performed using the same VCE, and the only difference in the two experiments was the network used to estimate the homography matrix. Given the results of the homography estimation, it is expected that the network with timestamp will perform worse than the baseline network. It is important to note that the VCE with associated timestamps do not have topographical annotations, and so the entire videos were analysed. According to literature (see Section 2.1), the entire GIT should have a length of about 7.5 meters. However, we must consider that the capsule may not be able to keep recording throughout the entire GIT, which means that the displacement values obtained should tend to be smaller than the average length. The post-processing technique was slightly altered, using $L = 7000$, to accommodate this change.

The two VCE with associated timestamps, that were not used for the training of the network, were used for testing. To provide a comparison term for the results obtained, the displacement was estimated based on both the homography obtained with the timestamp network, and the baseline network.

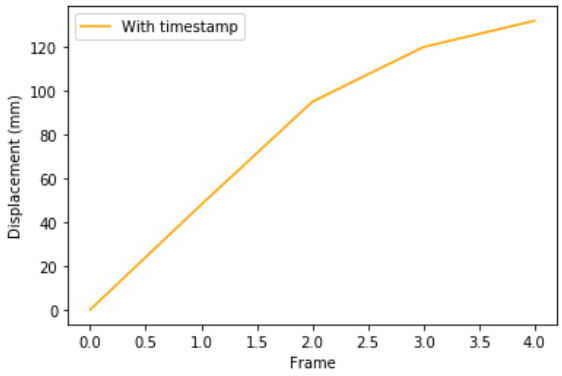
The first test, at first view, seemed to have worked very well, with an estimated GIT length of 6999.09 millimetres. However, upon further analysis, it is clear that all values of displace-

ment were replaced by the average displacement, during the post-processing stage. Without post-processing, the value obtained would be 415879.31 millimetres, which is a utterly impossible value. On the other hand, the test that used the baseline network as basis, provided a GIT length of 7723.04 millimetres, proving that the issue does not lie in the VCE used, but in the timestamp network. This is further proved when looking at the results closer, as on Figure 4.14, where the per-frame displacement of a sequence is detailed. From frame 0 to the frame 1, it is clear that the capsule moves backwards. The displacement computed by the network without the timestamp is able to recognise this, as well as the small amount of movement between frame 2 and frame 3, when compared to the rest. When considering the estimations obtained through the use of timestamps (and unprocessed), these details are not visible, and the network largely overestimates the displacements.

(a) Frame 0 (b) Frame 1 (c) Frame 2 (d) Frame 3 (e) Frame 4




(f) Per-frame displacement, obtained using the network without timestamps.




(g) Per-frame displacement, obtained using the network with timestamps.

Figure 4.14: Image sequence and respective displacement estimation, using both versions of the network.

The results of the second tested videos consolidate the previous conclusion, with total displacement of 341995.18 millimetres using the timestamp and without post-processing, and 7065.06 millimetres with the baseline network homography estimation.

In both tests, the estimated displacements of essentially all frames were very large in absolute value, but they presented both positive and negative signal. This means that, although the network tends to exaggerate the movement between consecutive frames, it is able to recognise the direction of the movement. This leads us to believe that, with the use of a larger amount of data, and

upon further training, the homography estimation network could perform considerably better. The displacement results only mirror what was seen when analysing the homography estimation, and do not imply that the displacement computation process is flawed.

## 4.6 Summary

When compared to the state of the art (see, for example, Pinheiro et al. [42]), the performance of the proposed method falls somewhat behind. However, it is of paramount importance to note that the main objective of the developed system is to allow the use of unlabelled data for a task that has been, so far, dependent on transfer learning or synthetic data generation to produce adequate results. This work shows that unsupervised capsule displacement estimation, as well as homography estimation, between VCE frames is, indeed, possible. However, the system has some downsides, not being able to reach the standard set by supervised techniques. To tackle this issue, some possibilities can be explored, such as alterations to the network, to accommodate different types of features (through the use of inception modules, for example), or incorporate the timestamps in a different way. The collection of more data would also be useful, especially to allow the training using solely real data. Finally, it would be useful to have the input of a specialist, to better assess the performance of the displacement estimation technique, and further validate it.

# Chapter 5

# Conclusions and Future Work

Video capsule endoscopy has become the main GIT screening method. Besides being more comfortable and practical than traditional endoscopy, it has the ability to reach the small bowel, unlike handheld endoscopy and colonoscopy. This technique provides an efficient method of diagnosis for small bowel lesions, and other abnormalities. However, to ensure adequate treatment of any lesion found, it is essential to know where the abnormality was found. Although there are some hardware solutions, like radio transmitters, these techniques do not satisfy all the requirements needed: either they do not provide results with enough accuracy, or they imply the use of additional hardware, making them uncomfortable for the patient. Image analysis solutions, that provide the capsule location based on the images acquired exist, but still have room for improvement.

It was with this in mind that the present dissertation was proposed, aiming to develop a system to automatically estimate capsule per frame capsule displacement in VCE, using only image information. The proposed pipeline is based on deep learning techniques, using a convolutional neural network to estimate homography between consecutive VCE frames. Using this homography, the second part of the pipeline is able to compute displacement between the frames. In addition to this basic system, both pre-processing and post-processing methods were suggested, further improving the system. The pre-processing technique, anisotropic diffusion, is able to homogenise the dataset, without losing relevant features. The post-processing was based on thresholding the possible displacement solutions, and replacing the outliers with the average displacement of the VCE.

The final product demonstrates that unsupervised techniques are a viable option for visual odometry in scenarios where labelled datasets are very difficult, or even impossible, to obtain, particularly in the case of VCE. Such a system eliminates the need to create artificially labelled data, being able to simply use the raw image information to achieve the desired results. This is considered to be the most important contribution of the present dissertation.

## 5.1 Objectives Accomplishment

The main objective of the work was the development of a deep-learning based capsule progression estimation technique for VCE frames of the small bowel. Although there is still room for improvement concerning the developed method, particularly when it comes to validation issues, we can say that this goal has been fulfilled. The system is able to successfully estimate homography between both artificial and real images. In the artificial data tests, the network was able to estimate homography with a mean average corner error of 19 pixels. The visual analysis of the test performed on real data served to further validate the previous results.

Another important objective was to make the system completely independent of ground truth displacement or homography labels, relying only on image information. This requirement was also met, with the entire system relying on photometric loss. Although a second version of the network was developed that included a timestamp associated with each VCE frame, even so, no ground truth is used. The simplicity of the method means that no additional hardware must be used to provide ground truths when acquiring the video, thus maximising patient comfort.

The dissertation also sought to gather a comprehensive literature review of the discipline at hand. With the analysis of solutions applied to both MVS and capsule localisation, along with the study of the mathematical and anatomical background behind the problem, we can say that the goal was, once again, met.

Given the accomplishment of the proposed objectives, we can consider that the present dissertation contributed positively to the field, in both the gathering of previously proposed techniques, and the development of a new pipeline suitable to tackle the problem of capsule localisation. However, the project can still move forward, with different implementations and small adjustments, to further improve the results obtained. Some possibilities for future work will be presented in the following section.

## 5.2 Future Work

The work developed thus far, although very satisfactory, would benefit from additional steps, to solve the main problems encountered.

First, to improve the network with timestamps, it would be advisable to collect timestamps for a higher number of videos, to allow more efficient training of the network, as well as to provide more test subjects.

Another interesting experiment would be to compare results between this network architecture to others, as described in Chapter 2. For example, the DispNet architecture should be very efficient, as well as the inclusion of inception modules. Although the proposed architecture has already proved to be useful in MVS and endoscopy applications, there are others that may reach better results.

The development of a more complex post-processing system may also be beneficial. The threshold used in the present work was chosen entirely empirically, based on the VCE tested. Although this method works for the dataset at hand, it would be key to develop a universal algorithm for determining the ideal threshold.

Finally, the use of RCNN, as seen in some previous works, may be useful. However, this would also imply the creation of a very large, real, and labelled VCE dataset, which is a very difficult task within itself. It would only be possible through the use of high-accuracy 3D localisation systems, which could be very difficult to use on a real test subject. Thus, the use of surgical models should be considered.

# References


[1] Acharya, D., Yan, W., and Khoshelham, K. (2018). Real-time image-based parking occupancy detection using deep learning. In *Proceedings of the 5th Annual Conference of Research*, page 33–40, Adelaide, Australia.

[2] Agarwal, A., Jawahar, C. V., and Narayanan, P. J. (2005). A Survey of Planar Homography Estimation Techniques. Technical report.

[3] Balaji, A. (2018). A simple Python API for single camera calibration using opencv.

[4] Bradski, G. and Kaehler, A. (2008). *Learning OpenCV*.

[5] Brown, M. and Lowe, D. (2002). Invariant Features from Interest Point Groups. *Procedings of the British Machine Vision Conference 2002*, pages 1–23.

[6] Brown, M. and Szeliski, R. (2008). Multi-image feature matching using multi-scale oriented patches. *Conference on Computer Vision and Pattern Recogniti*.

[7] Collins, J. T. and Badireddy, M. (2019). *Anatomy, Abdomen and Pelvis, Small Intestine*. StatPearls Publishing.

[8] DeTone, D., Malisiewicz, T., and Rabinovich, A. (2016). Deep Image Homography Estimation. *arXiv preprint*.

[9] Dubrofsky, E. (2009). Homography Estimation. Technical report, University of British Columbia, Vancouver.

[10] Furukawa, Y. and Hernández, C. (2015). Multi-View Stereo : A Tutorial. Technical report, University of Washington, St. Louis.

[11] Garcia, V., Debreuve, E., Nielsen, F., and Barlaud, M. (2010). K-nearest neighbor search: Fast GPU-based implementations and application to high-dimensional feature matching. *Proceedings - International Conference on Image Processing, ICIP*, pages 3757–3760.

[12] Goenka, M. K., Majumder, S., and Goenka, U. (2014). Capsule endoscopy: Present status and future expectation. *World Journal of Gastroenterology*, (29):10024–10037.

[13] Helander, H. F. and Fändriks, L. (2014). Surface area of the digestive tract-revisited. *Scandinavian Journal of Gastroenterology*.

[14] Hladnik, A. and Tomc, H. G. (2016). SURF : Detection , description and matching of local features in 3D computer graphics. In *8th International Symposium on Graphic Engineering and Design*, Novi Sad, Serbia.

[15] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*.

[16] Iakovidis, D. K. and Koulaouzidis, A. (2015). Software for enhanced video capsule endoscopy: Challenges for essential progress. *Nature Reviews Gastroenterology and Hepatology*.

[17] Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprin*.

[18] Jaimez, M., Souiai, M., Gonzalez-Jimenez, J., and Cremers, D. (2015). A Primal-Dual Framework for Real-Time Dense RGB-D Scene Flow. In *IEEE international conference on robotics and automation (ICRA)*.

[19] Jain, A. K. and Mao, J. (1996). Artificial Neural Networks: A Tutorial. *Computer*, pages 31–44.

[20] Jia, Q., Wan, X., Hei, B., and Li, S. (2018). DispNet based stereo matching for planetary scene depth estimation using remote sensing images. In *10th IAPR Workshop on Pattern Recognition in Remote Sensing*.

[21] Kendall, A., Grimes, M., and Cipolla, R. (2015). PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946.

[22] Khan, A. and Zhang, F. (2017). Using recurrent neural networks (RNNs) as planners for bio-inspired robotic motion. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1025–1030. IEEE.

[23] Koulaouzidis, A., Iakovidis, D., Yung, D., Mazomenos, E., Bianchi, F., Karagyris, A., Dimas, G., Stoyanov, D., Thorlacius, H., Toth, E., and Ciuti, G. (2018). Novel experimental and software methods for image reconstruction and localization in capsule endoscopy. *Endoscopy international open*.

[24] Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, pages 436–444.

[25] Li, B. and Meng, M. Q. (2012). Wireless capsule endoscopy images enhancement via adaptive contrast diffusion. *Journal of Visual Communication and Image Representation*.

[26] Li, J., Li, E., Chen, Y., Xu, L., and Zhang, Y. (2010). Bundled depth-map merging for multi-view stereo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2769–2776.

[27] Liao, Z., Gao, R., Xu, C., and Li, Z. S. (2010). Indications and detection, completion, and retention rates of small-bowel capsule endoscopy: a systematic review. *Gastrointestinal Endoscopy*.

[28] Lin, Y. M., Yeh, C. H., Yen, S. H., Ma, C. H., Chen, P. Y., and Jay Kuo, C. C. (2010). Efficient VLSI design for SIFT feature description. In *International Symposium on Next-Generation Electronics, ISNE 2010 - Conference Program*, pages 48–51.

[29] Liu, H., Lu, W. S., and Meng, M. Q. (2011). De-blurring wireless capsule endoscopy images by total variation minimization. In *IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing - Proceedings*.

[30] Malis, E. and Vargas, M. (2007). Deeper understanding of the homography decomposition for vision-based control. Technical report.

[31] Martins Pinheiro, G., Manuel Trigueiros de Silva Cunha Second Supervisor, A., and Filipe Pinto de Oliveira, H. (2018). Visual Odometer on Videos of Endoscopic Capsules (VOVEC). Technical report.

[32] Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2015). A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[33] Medtronic. PillCam™ SB 3 System | Medtronic.

[34] Medtronic. RAPID™ Reader Software v8.3 Update | Medtronic.

[35] Medtronic (2018). PillCam ™ SB 3 System Advanced imaging and visualization of the small bowel.

[36] Mount, D. M., Netanyahu, N. S., and Moigne, J. L. (1999). Efficient algorithms for robust feature matching. *Pattern Recognition*, pages 17–38.

[37] Netter, F. H. and Colacino, S. (1989). *Atlas of human anatomy*.

[38] Nguyen, T., Chen, S. W., Shivakumar, S. S., Taylor, C. J., and Kumar, V. (2017). Unsupervised Deep Homography: A Fast and Robust Homography Estimation Model. *IEEE Robotics and Automation Letters*.

[39] OpenStax (2013). *Anatomy and Physiology*. OpenStax.

[40] Penza, V., Ciullo, A. S., Moccia, S., Mattos, L. S., and De Momi, E. (2018). EndoAbS dataset: Endoscopic abdominal stereo image dataset for benchmarking 3D stereo reconstruction algorithms. *The International Journal of Medical Robotics and Computer Assisted Surgery*.

[41] Perona, P. and Malik, J. (1990). Scale-Space and Edge Detection Using Anisotropic Diffusion. Technical report.

[42] Pinheiro, G., Coelho, P., Salgado, M., Oliveira, H. P., and Cunha, A. (2019). Deep Homography Based Localization on Videos of Endoscopic Capsules. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 724–727.

[43] Pitrinec (2019). Automation Software for Windows - Macro Toolworks, Perfect Keyboard.

[44] Pogorelov, K., Schmidt, P. T., Riegler, M., Halvorsen, P., Randel, K. R., Griwodz, C., Eskeland, S. L., de Lange, T., Johansen, D., Spampinato, C., Dang-Nguyen, D.-T., and Lux, M. (2017). KVASIR. In *8th ACM on Multimedia Systems Conference*, pages 164–169, New York, New York, USA.

[45] Ramaraj, M., Raghavan, S., and Khan, W. A. (2013). Homomorphic filtering techniques for WCE image enhancement. In *IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–5.

[46] Ramos Diaz, E. and Ponomaryov, V. (2005). A Review of 3D Reconstruction from Video Sequences. *Revista Facultad de Ingenier Universidad de Antioquia*, page 111–121.

[47] Salahat, E. and Qasaimeh, M. (2017). Recent advances in features extraction and description algorithms: A comprehensive survey. *IEEE International Conference on Industrial Technology*, pages 1059–1063.

[48] Sarle, W. S. (1994). Neural Networks and Statistical Models. In *Nineteenth Annual SAS Users Group Intern ational Conference*.

[49] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 519–526.

[50] Shahril, R., Baharun, S., and Muzahidul Islam, A. K. (2016). Pre-processing technique for wireless capsule endoscopy image enhancement. *International Journal of Electrical and Computer Engineering*.

[51] Shashua, A. (1997). Trilinear Tensor : The Fundamental Construct of Multiple-view Geometry and Its Applications. In *International Workshop on Algebraic Frames for the Perception-Action Cycle*, Berlin, Heidelberg.

[52] Silpa-Anan, C. and Hartley, R. (2008). Optimised KD-trees for fast image descriptor matching. *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

[53] Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint*.

[54] Softpedia (2001). SensArea 1.12.1.

[55] Spyrou, E. and Iakovidis, D. K. (2012). Homography-based orientation estimation for capsule endoscope tracking. In *IEEE International Conference on Imaging Systems and Techniques*.

[56] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–9.

[57] Than, T. D., Alici, G., Zhou, H., and Li, W. (2012). A review of localization systems for robotic endoscopic capsules. *IEEE Transactions on Biomedical Engineering*.

[58] Tsai, P.-S. and Shah, M. (1994). Shape from Shading Using Linear Approximation. *Image and Vision Computing*.

[59] Turan, M., Abdullah, A., Jamiruddin, R., Araujo, H., Konukoglu, E., and Sitti, M. (2017a). Six Degree-of-Freedom Localization of Endoscopic Capsule Robots using Recurrent Neural Networks embedded into a Convolutional Neural Network. *arXiv preprint*.

[60] Turan, M., Almalioglu, Y., Araujo, H., Konukoglu, E., and Sitti, M. (2018a). Deep EndoVO: A recurrent convolutional neural network (RCNN) based visual odometry approach for endoscopic capsule robots. *Neurocomputing*.

[61] Turan, M., Almalioglu, Y., Konukoglu, E., and Sitti, M. (2017b). A Deep Learning Based 6 Degree-of-Freedom Localization Method for Endoscopic Capsule Robots. *arXiv preprint*.

[62] Turan, M., Ornek, E. P., Ibrahimli, N., Giracoglu, C., Almalioglu, Y., Yanik, M. F., and Sitti, M. (2018b). Unsupervised Odometry and Depth Learning for Endoscopic Capsule Robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[63] Valipour, S., Siam, M., Jagersand, M., and Ray, N. (2017). Recurrent fully convolutional networks for video segmentation. In *IEEE Winter Conference on Applications of Computer Vision*, pages 29–36.

[64] Zhang, R., Tsai, P. S., Cryer, J. E., and Shah, M. (1999). Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[65] Zhou, T., Brown, M., Noah, G., Google, S., and Lowe Google, D. G. (2017). Unsupervised Learning of Depth and Ego-Motion from Video. In *IEEE Conference on Computer Vision and Pattern Recognition*.