Mariana Gil Cunha

Quantificação do fluxo e tempo de semi-vida dos microtúbulos do fuso mitótico em metafase

Departamento de Matemática
Faculdade de Ciências da Universidade do Porto
2012

Mariana Gil Cunha

Quantificação do fluxo e tempo de semi-vida dos microtúbulos do fuso mitótico em metafase

Tese submetida à Faculdade de Ciências da
Universidade do Porto para obtenção do grau de Mestre
em Engenharia Matemática

Departamento de Matemática
Faculdade de Ciências da Universidade do Porto
2012

Agradecimentos

Ao Professor Paulo Aguiar um agradecimento muito especial pela simpatia, boa disposição, amizade e total disponibilidade na orientação científica deste trabalho.
Ao Dr. Hélder Maiato por ter sido o primeiro a incentivar esta ideia. Por me ter proporcionado condições de trabalho que superaram as minhas expectativas.
Ao Dr. António Pereira pela sua incondicional disponibilidade para apoiar e orientar este trabalho ao longo de vários meses.
À minha família, em especial aos meus pais e irmãos, por me acompanharem sempre.
À Anne, ao Luís, à Margarida, à Sofia, por estarem sempre perto. À Ana e à Filipa pelo companheirismo e bons momentos ao longo do Mestrado.

Resumo

Este trabalho resulta de uma parceria entre o Mestrado em Engenharia Matemática e a equipa Chromosome Instability & Dynamics Lab, do Instituto de Biologia Molecular e Celular, liderada pelo Dr. Hélder Maiato, coorientador deste trabalho.
Foi proposto quantificar o fluxo dos microtúbulos de tubulina e o processo de turnover durante a metafase, em células de organismos diversos. A abordagem utilizada tirou partido da técnica Induced Speckle Imaging desenvolvida pelo investigador António Pereira.
Foi produzido um algoritmo automático para determinação de fluxo baseado no método de Block Matching. Como operadores de deteção espacial, foram avaliadas duas funções em situações de teste perfeitamente controladas: uma função de correlação generalizada a n sinais e uma função de diferenças absolutas em relação à média.
Através dos testes apurou-se a função de diferenças absolutas em relação à média como o operador mais eficaz e robusto a ruído na determinação dos fluxos. Os resultados apresentados para um filme de *Drosophila* S2 foram produzidos utilizando a estratégia de block matching com este operador.
Não foi desenvolvido na totalidade um método de medição de *turnover*, mas foi elaborado e implementado um método de determinação do decaimento da intensidade dos microtúbulos para os filmes disponibilizados.
Para estimar o decaimento de intensidade, foi criado um algoritmo semi-automático que permite obter estatísticas de parâmetros de regressões exponenciais ajustadas à intensidade dos microtúbulos. A robustez dos ajustes exponenciais à intensidade dos pixeis nas imagens foi testada em situações controladas, obtendo muito bons resultados.
O software utilizado para implementar os métodos elaborados foi o MATLAB (2010).
A determinação do fluxo e do decaimento nos microtúbulos permite realçar o protagonismo destes fenómenos na promoção de forças no fuso mitótico que movimentam os cromossomas em direção aos polos, permitindo descobrir um pouco mais sobre o mecanismo que garante a herança genética nas espécies.

Palavras-chave: CÁLCULO DE FLUXOS; PROCESSAMENTO DE IMAGEM; MAPAS CONFORMACIONAIS; REGRESSÃO NÃO LINEAR; BLOCK MATCHING; FUSO MITÓTICO

Abstract

This work results from a collaboration between the Master in Mathematical Engineering and the Chromosome Instability & Dynamics Lab team, from the Institute of Molecular and Cell Biology and led by Dr. Hélder Maiato, co-advisor of this work.
It has been proposed to quantify spindle microtubules flux and turnover during metaphase on cells of different organisms. The present approach took advantage of the Induced Speckle Imaging technique developed by researcher António Pereira.
An algorithm for automatic determination of flow based on a strategy derived from Block Matching was created. Two operators of spatial detection were evaluated in two perfectly controlled test situations: a correlation function generalized to n signals and a function measuring absolute differences from the average. Through testing it was possible to conclude that the absolute differences function is the most efficient and noise robust on the determination of displacement. The results obtained in the original images were produced using the Block Matching strategy with this operator.
A measuring method of turnover was not developed in its entirety, but a method for determining the decay of the intensity of microtubules for the films provided is designed and implemented.
For the estimation of color intensity decay a semi-automatic algorithm was created. The method performs exponential regression using the intensity of microtubules. The robustness of the exponential fits to the fluorescence decay was tested in controlled situations, obtaining very good results.
The software used to build the codes for these algorithms was MATLAB (2010).
The determination of microtubules flux and turnover allows us to highlight the role of these phenomena in the promotion of the mitotic spindle forces that move chromosomes toward the poles, making it possible to discover a little more about the mechanism that ensures genetic inheritance in the species.

Key-words: FLUX DETERMINATION; IMAGE PROCESSING; CONFORMAL MAPPING; NON-LINEAR REGRESSION; BLOCK MATCHING; MITOTIC SPINDLE

Conteúdo

Agradecimentos **iii**

Resumo **v**

Abstract **vii**

Índice de Tabelas **xi**

Índice de Figuras **xvii**

1 Introdução ao problema **1**
1.1 Enquadramento na Biologia Celular 1
1.2 Os Dados - Objectivo 4
1.3 Abordagens existentes - Revisão de Literatura 6

2 Abordagens aos problemas **7**
2.1 Análise do fluxo 7
2.2 Método implementado 8
2.2.1 O programa 18
2.3 Análise do Decaimento 22
2.3.1 O programa 24
2.4 Desvantagens das abordagens 27

3 Programas auxiliares **29**
3.1 Leitura de Frames 29
3.2 Marcação dos polos 29
3.2.1 O programa 30
3.3 Mapa conformacional super-elipse - linhas paralelas 34

4 Resultados **39**
4.1 Os dados de teste 39
4.1.1 Os dados teste para calculo de fluxo 2D 44
4.2 Resultados nos dados de teste 45
4.2.1 Cálculo do fluxo - Dados artificiais 1D 45
4.2.2 Cálculo do fluxo - Dados teste 2D 56
4.2.3 Resultados da análise de decaimento nos dados de teste 60
4.3 Resultados para os dados originais 65
4.3.1 Resultados do fluxo a 1D 65
4.3.2 Resultados da análise de fluxo a 2D: 74
4.3.3 Resultados da análise de decaimento 75

5 Conclusões **83**

6 Trabalho Futuro **85**

Referências **86**

A **i**

B **xi**

Lista de Tabelas

2.1 Exemplo de valores obtidos para F e para Ref. A verde, $F(\hat{v})$ 18
2.2 Valores exemplo obtidos para R e para Ref ordenados; 18

Lista de Figuras

1.1 Ilustração das várias fases da mitose celular. Imagem de domínio público adaptada de Wikipedia.org, autoria de Mariana Ruiz Villarreal. 2
1.2 Micografia evidenciando os elementos estruturais do fuso mitótico e a disposição dos cromossomas ao longo do equador do fuso mitótico. Imagem adaptada de Rieder (2006). 3
1.3 Ilustração do fluxo da tubulina nos microtúbulos conectados a cinetocoros e microtúbulos do fuso mitótico de célula Drosophila S2. Figura fornecida por cortesia de António Pereira, Chromosome Instability & Dynamics Lab. 3
1.4 Fusos mitóticos de células Drosophila S2 durante métafase (IBMC). 4
1.5 Imagem de *Drosophila* S2 obtida por microscopia de fluorescência. Induced Speckle Imaging: *speckles* no fuso mitótico (direita). Figura fornecida por cortesia de António Pereira, Chromosome Instability & Dynamics Lab. 5

2.1 Fusos com forma super-elíptica ajustada através do MATLAB. (A-B) Célula de *Drosophila S2*. (C) Neuroblasto. (D) Imagem C rodada e centrada no MATLAB. (E) Célula de *Drosophila S2*. (F) Imagem E rodada e centrada sobre o centro do fuso, limitada ao retângulo exterior ao fuso. 7
2.2 Ilustração de eliminação do ruído com a aplicação do mapa conformacional 8
2.3 Ilustração da manipulação de imagens com os métodos de cálculo de fluxo a 1D e 2D, dividida por 4 fases. (A) Fuso na posição inicial. (B) Rotação da imagem sobre o centro do fuso. (C) Corte da imagem em B pelo retângulo limitado aos 3 pontos. (D) Semi-fuso superior após aplicação do mapa conformacional à imagem em C. No cálculo de fluxo a 1D percorremos as 4 fases, enquanto que no cálculo de fluxo a 2D apenas as 3 primeiras fases. 8
2.4 Rotação e translação do fuso mitótico. 9
2.5 Ilustração de super elipses com diferentes valores de μ. $\mu = 1$ e $\mu = 0$ caraterizam os mapeamentos elíptico e cónico, respetivamente. $\mu = \infty$ não aplica mapeamento. 10
2.6 Frames originais, fusos e mapas conformacionais. (A,B) Imagem de *Drosophila S2*. (C) Imagem de célula de neuroblasto. (D,E,F) Imagens anteriores rodadas e cortadas pelos fusos mitóticos. (G,H,I) Mapas conformacionais das imagens da fila 2. 10
2.7 Início do tratamento da pilha de imagens selecionadas no algoritmo baseado na estratégia de Block Matching. Ilustração dos blocos (ou janelas) de trabalho. Pilha de imagens dividida por janelas de trabalho. 11
2.8 Sequência ilustrativa do algoritmo na fase de Block Matching. 11
2.9 Ilustração do block matching. Nos três casos, A,B e C, o bloco amarelo desloca-se para cima a uma velocidade de 1 *px/frame*. A vermelho estão representadas as janelas geradas em função da velocidade. (A) A vermelho, janelas obtidas para uma velocidade da janela de 2 *px/frame* (B) Janelas obtidas para uma velocidade da janela de 1 *px/frame*. (C) Janelas obtidas para velocidade da janela -1 *px/frame*. . 12

2.10 Ilustração das regiões das pilhas, das sub-pilhas selecionadas e da condensação das janelas em vetores. 12
2.11 Ilustração dos gráficos das intensidades de matrizes de trabalho hipotéticas de frames consecutivos. Simplificação da análise de velocidade na estratégia baseada no algortimo de Block Matching. 13
2.12 Esquema ilustrativo do cálculo do fluxo para um bloco. (A) Pilha de fusos após aplicação do mapa conformacional. (B) Sub-pilha ou pilha de janelas selecionadas. (C) Janelas da sub-pilha colapsadas em vetores-coluna. (D) Perfis de intensidade dos vetores-coluna. (C) Ilustração do conjunto de dados selecionados para avaliar em função da gama de velocidades. 14
2.13 Exemplo de função de referência e função F normalizadas, em função das velocidades testadas (eixo ox, oy), para caso de análise de fluxo a 2D. 15
2.14 Cálculo do fluxo a uma dimensão numa sub-pilha. Figura produzida através do código CalcFluxo_ NFrames_ 1D.m. Em cima vários blocos da sub-pilha analisada. Em cima à direita, perfis de intensidade dos blocos da sub-pilha. Em baixo, a azul, função $F(v)$ em função das velocidades analisadas (eixo ox). A preto, função de referência produzida. 15
2.15 Cálculo do fluxo a duas dimensões numa sub-pilha. Figura produzida através do código CalcFluxo_ NFrames_ 2D.m. Em cima vários blocos da sub-pilha analisada. Em baixo, função $F(v)$ em função das velocidades analisadas (eixos ox, oy). À direita, função de referência produzida para o caso. 16
2.16 Programa CalcularFluxo. Diagrama de fluxo (parte 1) 19
2.17 Programa CalcularFluxo. Diagrama de fluxo (parte 2) 20
2.18 Programa CalcularFluxo. Diagrama de fluxo (parte 3) 21
2.19 Programa CalcularFluxo. Diagrama de fluxo (parte 4) 22
2.20 Ilustração de decaimento médio 23
2.21 Programa DecaimentoSemiAutomatico, diagrama de fluxo (parte 1). 25
2.22 Programa DecaimentoSemiAutomatico, diagrama de fluxo (parte 2). 26
2.23 Fuso de *Drosophila S2* assimétrico. 27

3.1 Ilustração de super elipses com diferentes valores de μ. 30
3.2 Diagrama de fluxo do programa MarcarPolos, parte 1. 31
3.3 Diagrama de fluxo do programa MarcarPolos, parte 2. 32
3.4 Diagrama de fluxo do programa MarcarPolos, parte 3. 33
3.5 Rotação e translação das imagens para gerar as novas imagens. 34
3.6 Ilustração do mapa conformacional elipse - linhas paralelas 35
3.7 Fuso mitótico de célula de *Drosophila S2* após aplicação do mapa conformacional. Imagem anterior após interpolação. 36
3.8 Imagens de fusos mitóticos manipuladas com as funções de mapa conformacional . 37

4.1 Exemplo de frames consecutivos de controlo criados com o programa DadosTeste.m 39
4.2 Vetor *template* obtido no MATLAB sobre a as posições da imagem. Função gaussiana ou função K do código. A função de convolução aplica a cada ponto do vetor template a função gaussiana ilustrada. 42
4.3 Ilustração obtida no Matlab do vetor *template* após convolução com a gaussiana K 42
4.4 Ilustração obtida no Matlab da primeira matriz de uma pilha obtida na sequência das imagens anteriores. 42
4.5 Ilustração das velocidades dos *speckles* ao longo das linhas. 43
4.6 Ilustração das velocidades dos *speckles* ao longo das linhas. 44

4.7 Imagem de *Drosophila S2* rodada e limitada ao fuso mitótico com 3 *speckles* artificiais. 45
4.8 Mapa de velocidades reais (px/s) induzidas no filme do caso 1. Alguns valores de velocidades destacados para melhor interpretação do gráfico. 47
4.9 Perfil de intensidades da pilha produzida para o teste de controlo. 47
4.10 Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo da correlação generalizada, colapsando as janelas com o máximo. 48
4.11 Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo das diferenças absolutas à média, colapsando as janelas com o máximo. 48
4.12 Comparação dos resultados obtidos para o teste de controlo do caso 1. Gráfico das velocidades (px/s) obtidas em função das linhas para os dois métodos considerados e velocidades (px/s) atribuídas a cada pixel, no caso 1. 49
4.13 Perfil de intensidades da pilha produzida para caso 2. Relação sinal/ruído = 0.5, $\sigma = 5$. 50
4.14 Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo da correlação generalizada, colapsando as janelas com o máximo. Ruído = 0.5, $\sigma = 5$, Altura = 6. 51
4.15 Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo da correlação generalizada, colapsando as janelas com a mediana. Ruído = 0.5, $\sigma = 5$, Altura = 6. 51
4.16 Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo das diferenças absolutas à média, colapsando as janelas com o máximo. Ruído = 0.5, $\sigma = 5$, Altura = 6. 52
4.17 Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo das diferenças absolutas à média, colapsando as janelas com a mediana. Ruído = 0.5, $\sigma = 5$, Altura = 6. 52
4.18 Pormenor do equador de uma pilha de teste. Região central de uma pilha de teste em dois frames consecutivos. Janelas de trabalho evidenciadas. 53
4.19 Perfil de intensidades de uma coluna de imagem *Drosophila S2*. 54
4.20 Imagens de teste para o caso 3. Primeiro frame da pilha de teste utilizada (esquerda). Perfil da primeira coluna do primeiro frame da pilha utilizada. 55
4.21 Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo da correlação generalizada adptada, colapsando as janelas com o máximo. Ruído = 0.2, $\sigma = 5$, Altura = 18. 55
4.22 Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo DAM, colapsando as janelas com o máximo. Ruído = 0.2, $\sigma = 5$, Altura = 18. 56
4.23 Mapa de velocidades (px/s) e respetivos valores de τ-Kendall obtidas com aplicação do algoritmo de Correlação 2D para o teste do Caso 1. 57
4.24 Pormenor do mapa de velocidades (px/s) obtido para o caso 1. Zoom sobre o *speckle* artificial A. 58
4.25 Cálculo do fluxo a duas dimensões numa sub-pilha. Figura produzida através do código CalcFluxo_ NFrames_ 2D.m. Em cima vários blocos da sub-pilha analisada. Em baixo, função $F(v)$ em função das velocidades analisadas (eixos ox, oy). À esquerda, função de referência produzida. 58
4.26 Mapa de velocidades (px/s) e respectivos valores de τ-Kendall obtidas com aplicação do algoritmo de Correlação 2D para o teste do Caso 2. 59
4.27 Pormenor do mapa de velocidades (px/s) obtido para o caso 2. Zoom sobre o *speckle* artificial A. 59
4.28 Pormenor do mapa de velocidades (px/s) obtido para o caso 2. Zoom sobre o *speckle* artificial B. 60
4.29 Pormenor do mapa de velocidades (px/s) obtido para o caso 2. Zoom sobre o *speckle* artificial C. 60

4.30 *Fitting* exponencial obtido para o caso de teste 1. 62
4.31 Janelas de cálculo de decaimento. A janela da esquerda tem de altura $18px$ e a da esquerda de altura $5px$. 62
4.32 Histogramas das regiões selecionadas caso 1. Na primeira linha temos os frames analisados e os retângulos correspondentes às janelas selecionadas em cada frame. Na segunda linha os histogramas dos retângulos selecionados no respetivo frame. Na terceira linha histogramas dos frames (toda a área) respetivos à coluna. 63
4.33 *Fitting* exponencial obtido para caso de teste 2 com a função média. 64
4.34 *Fitting* exponencial obtido para caso de teste 2 com a função mediana. 64
4.35 *Speckles* avaliados no caso 1. Frame 3 do filme analisado com a localização de alguns *speckles*. 66
4.36 Mapa de velocidades (px/s) e mapa de τ-Kendal obtidos para o caso 1 nas regiões onde a intensidade dos pixeis no frame 3 é superior a 500. Alguns valores de velocidade registados assinalados. 67
4.37 *Speckles* avaliados no caso 2. Frame 7 do filme analisado com a localização de alguns *speckles*. 68
4.38 Mapa de velocidades (px/s) e mapa de τ-Kendal obtidos para o caso 2 nas regiões onde a intensidade dos pixeis no frame 7 é superior a 800. Alguns valores de velocidade registados assinalados. 69
4.39 *Speckles* avaliados no caso 2. Frame 16 do filme assinalado com a localização de alguns *speckles*. 70
4.40 Mapa de velocidades (px/s) e mapa de τ-Kendal obtidos para o caso 2 nas regiões onde a intensidade dos pixeis no frame 16 é superior a 500. Alguns valores de velocidade registados assinalados. 71
4.41 *Speckles* avaliados no caso 2. Frame 31 do filme analisado com a localização de alguns *speckles*. 72
4.42 Mapa de velocidades (px/s) e mapa de τ-Kendal obtidos para o caso 2 nas regiões onde a intensidade dos pixeis no frame 31 é superior a 500. Alguns valores de velocidade registados assinalados. 73
4.43 Frame 7 do filme e mapa de pesos obtido para o caso 1. 74
4.44 Velocidades (px/s) obtidas para região A do frame 7. 75
4.45 Velocidades (px/s) obtidas para região B do frame 7. 75
4.46 *Speckle* do hemisfério superior avaliado no teste de decaimento. 76
4.47 Fit exponencial para o *speckle* do hemisfério superior. 76
4.48 Histogramas das janelas do *speckle* inferior e dos frames utilizados. Na primeira linha temos os frames analisados e os rectângulos correspondentes às janelas selecionadas em cada frame. Na segunda linha os histogramas dos rectângulos selecionados no respectivo frame. Na terceira linha histogramas dos frames (toda a área) respetivos à coluna da figura. 77
4.49 Fit exponencial para o *speckle* do hemisfério superior em frames posteriores a 30. . 78
4.50 *Speckle* do hemisfério inferior avaliado no teste de decaimento. 78
4.51 Fit exponencial para o *speckle* do hemisfério inferior. 79
4.52 *Speckle* situado no equador do fuso avaliado no teste de decaimento. 79
4.53 Fit exponencial para o *speckle* do hemisfério inferior. 80
4.54 Região escura do filme das células *Drosophila* S2. 80
4.55 Fit exponencial para a região escura. 81

A.1 Ilustração dos blocos/janelas de trabalho ii
A.2 Exemplo de como se obtêm as variáveis f1 e f2 v

A.3 Exemplo de como se obtêm as variáveis f1 e f2 viii
A.4 Ilustração de regiões de uma imagem limitadas pela velocidade máxima onde é possível calcular fluxo. A verde região limitada pela dimensão das janelas e a amarelo região limitada pela velocidade máxima e número de frames a utilizar. viii

B.1 Imagem Matlab do programa DecaimentoSemiAutomatico.m, seleção da primeira região para todos os frames xiv
B.2 Ilustração da matriz linhas região do código DecaimentoSemiAutomatico.m xiv
B.3 Regiões selecionadas e histogramas respetivos xv
B.4 Exemplo de figura com diferentes *fittings* exponenciais para cada região. xvii

Capítulo 1

Introdução ao problema

1.1 Enquadramento na Biologia Celular

A reprodução é talvez a atividade mais fundamental na vida das células. Para organismos unicelulares, dividir é reproduzir, enquanto que nos organismos multicelulares a divisão celular assegura a produção de estruturas essenciais ao crescimento e desenvolvimento. Em 1885, estudos de biólogos como Walter Flemming mostraram que num organismo todos os núcleos são gerados por repetidas divisões de um único núcleo, este formado na fase embrionária através da fusão entre um óvulo e o núcleo de um espermatozoide. Esta conclusão permitiu interligar a *A Teoria Celular* (1838), de Schleiden and Schwann, com a *Teoria da Evolução* (1859) de Darwin (Rieder, 2006).

Mitose

Em células simples como algumas bactérias ou cianobactérias, cujo núcleo não está envolvido por uma membrana, a divisão celular ocorre por fissão binária. Em todos os organismos mais complexos, o processo que permite a transferência do legado genético das espécies ao produto de cada divisão celular é a mitose celular. O principal objetivo da mitose é formar duas novas células geneticamente idênticas à célula precursora (Pereira and Maiato, 2010).
A mitose pode ser dividida em cinco fases gerais: Profase, Prometafase, Metafase, Anafasee e Telofase. Após a mitose segue-se a divisão do citoplasma, durante a Citocinese. Estas etapas estão ilustradas na figura 1.1.
Em células de animais superiores, o fenómeno que premedita uma divisão celular é a replicação do material genético, no núcleo das células, e o início da sua condensação na forma cromossómica (Rieder, 2006). Na fase inicial da mitose, a Profase, o ADN replicado,em conjunto com algumas proteínas essenciais, organiza-se no interior do núcleo em estruturas condensadas, os cromossomas. Durante a divisão celular, os cromossomas apresentam duas porções idênticas de material replicado, os cromatídeos, ligadas por uma estrutura de conexão, o centromero. Por esta altura da divisão celular, a membrana nuclear desintegra-se, libertando os cromossomas no citoplasma.
Durante a Prometafase, os cromossomas conectam-se a uma estrutura de microtúbulos e centrossomas, denominada fuso mitótico. Esta estrutura é responsável por gerar as forças que movimentam os cromossomas e por direcionar os seus movimentos na célula (Rieder, 2006).
Uma vez ligados aos microtúbulos, os cromossomas dispõe-se alinhados ao longo do equador (eixo central) do fuso. Esta disposição alinhada ao longo do equador carateriza a Metafase da mitose celular.
No momento em que os cromossomas iniciam a sua divisão e separação em direções opostas entram em Anafase. Nesta altura são evidentes dois grupos independentes de cromossomas, cada um

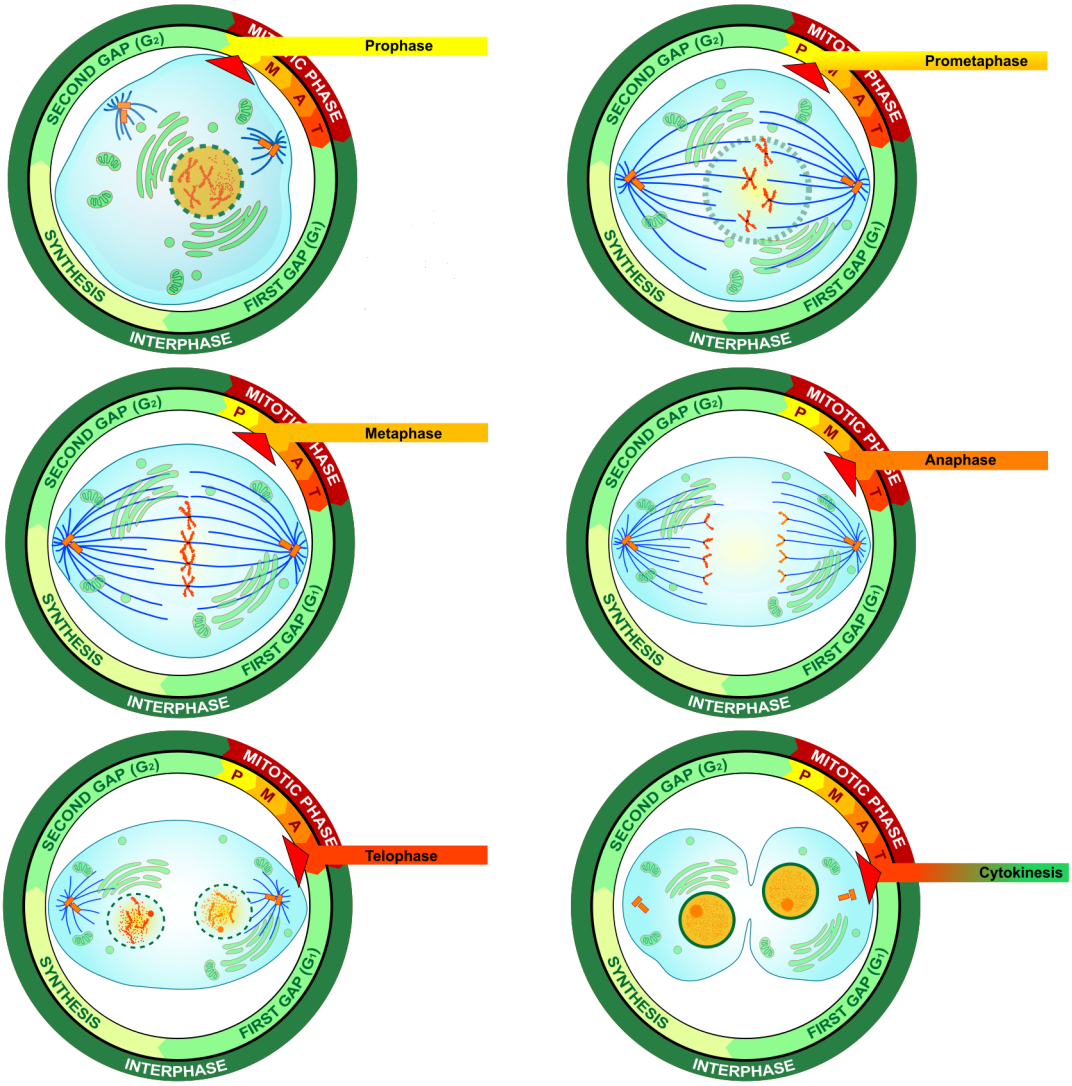

Figura 1.1: Ilustração das várias fases da mitose celular. Imagem de domínio público adaptada de Wikipedia.org, autoria de Mariana Ruiz Villarreal.

movendo-se em direção a extremos ou polos opostos do fuso mitótico.
Durante a Telofase, em redor dos dois grupos de cromossomas formam-se novas membranas nucleares, e o material genético retorna à consistência de cromatina. Por último, completa-se a divisão do citoplasma durante a Citocinese, terminando com o nascimento de duas novas células.

O fuso mitótico:

Durante a mitose, a estrutura biológica que assegura a correta segregação do genoma entre os dois novos núcleos é um complexo altamente dinâmico, o fuso mitótico (Matos et al., 2009). Em conjunto com um grande número de proteínas, denominadas de proteínas motoras, o fuso mitótico é capaz de dirigir o alinhamento dos cromossomas em metafase e a sua separação em direções opostas da célula (Pereira and Maiato, 2010). Este complexo contém três componentes estruturais primárias: os microtúbulos, os cromossomas e os centrossomas, como se pode observar na figura 1.2.

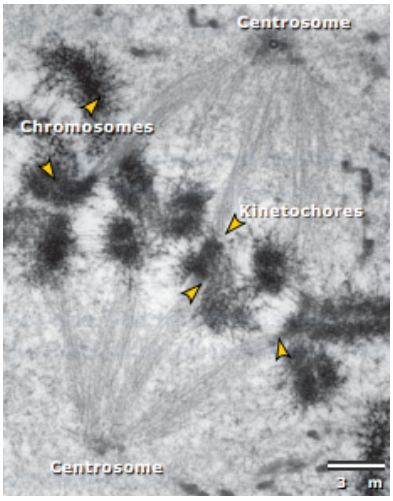

Figura 1.2: Micografia evidenciando os elementos estruturais do fuso mitótico e a disposição dos cromossomas ao longo do equador do fuso mitótico. Imagem adaptada de Rieder (2006).

Durante a progressão da mitose, a distribuição tridimensional dos cromossomas no núcleo evolui para uma distribuição bidimensional, ao longo do fuso (em metafase), atribuindo a esta disposição o nome de placa metafásica. Esta notória mudança conformacional é essencial, uma vez que liberta uma das três dimensões espaciais da distribuição dos cromossomas para definir um eixo de divisão celular. De facto, a grande maioria da dinâmica do fuso e dos cromossomas ocorre preferencialmente ao longo deste eixo (Pereira and Maiato, 2010).
Cada polo do fuso é definido por um centrossoma, um organelo duplicado na fase de replicação do genoma que antecede a mitose. Posicionados entre os centrossomas estão os cromossomas replicados.
Durante a Profase, surgem nos centrómeros dos cromossomas duas estruturas especializadas para interagir com os microtúbulos do fuso mitótico, os cinetocoros. Alguns microtúbulos do fuso conectam-se aos cinetocoros dos cromossomas num dos seus extremos, enquanto que no outro extremo estão conectados a um centrossoma ou perto deste. Os restantes microtúbulos conectam-se apenas ao polo do fuso, estando o outro extremo livre na região do fuso.

Os microtúbulos do fuso mitótico são fibras cuja unidade estrutural é a proteína tubulina.

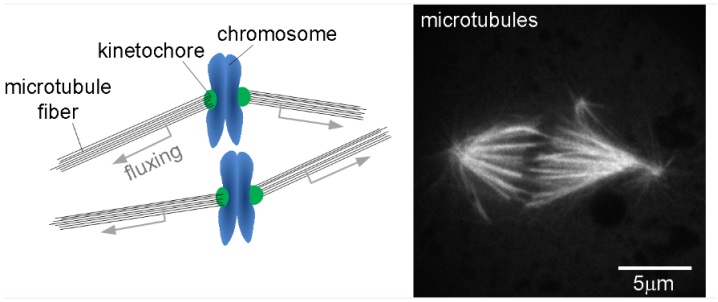

Figura 1.3: Ilustração do fluxo da tubulina nos microtúbulos conectados a cinetocoros e microtúbulos do fuso mitótico de célula Drosophila S2. Figura fornecida por cortesia de António Pereira, Chromosome Instability & Dynamics Lab.

À medida que o fuso se forma, surge uma nova dinâmica nos microtúbulos, o fluxo das suas

subunidades. As proteínas de tubulina dispersas no citoplasma são incorporadas nos extremos dos microtúbulos dispostos no equador (extremos positivos) e deslocam-se com os microtúbulos em direção ao outro extremo (negativo), onde são libertadas (figura 1.3).
O processo de renovação dos microtúbulos denomina-se de turnover.

O sentido e a finalidade da mitose são revelados no seu clímax durante a Anafase quando os cromatídeos homólogos migram em direção aos polos opostos de forma sincronizada (Matos et al., 2009). Os erros que ocorrem neste processo podem ter consequências devastadoras para a célula e para todo o organismo, uma vez que podem causar instabilidade genética, defeitos de nascimento ou cancro (Civelekoglu-Scholey and Scholey, 2010).
Atualmente admite-se que as forças que movem os cromossomas em direção aos polos são produzidas por dois mecanismos de forças, que frequentemente atuam em simultâneo (Rieder, 2006).
Um dos mecanismos baseia-se em forças geradas nos microtúbulos. Durante a metafase, o comprimento de um microtúbulo que esteja conectado a um cinetocoro permanece constante desde que a polimerização da tubulina no seu polo positivo seja equilibrada pela despolimerização de tubulina no seu polo negativo. Se a taxa de incorporação das subunidades nos cinetocoros decresce e a despolimerização da tubulina se mantém, o cinetocoro movimenta-se em direção ao polo do fuso (Rieder, 2006).
Para além do fluxo, várias proteínas motoras interagem com os microtúbulos do fuso durante a Anafase, tendo um protagonismo preponderante na movimentação dos cromatídeos, como por exemplo a proteína kinesin-5 (Miyamoto et al., 2004) e dineína citoplasmática (Yang et al., 2008). No mecanismo "Pac-man", a dineína pressiona ativamente o cinetocoro em direção ao polo negativo do microtúbulo. Esta compressão da cinetocoro provoca uma reação a essa força e a cinetocoro induz a despolimerização da tubulina nos extremos positivos dos microtúbulos a elas acopolados (Rieder, 2006).
Em Matos et al. (2009) mostrou-se que, em células de *Drosophila* S2, o fluxo nos microtúbulos representa uma forma de redistribuição de tensões excessivas nos cromossomas durante a metafase, contribuindo assim para a equalização das forças atuantes nos cinetocoros e para o bom desempenho da mitose.
O presente trabalho propõe um algoritmo automático de determinação do fluxo ao longo dos microtúbulos e um algoritmo semi-automático de determinação de turnover em filmes de células sujeitas à técnica Induced Speckle Imaging (ISI) (Pereira, 2011).

1.2 Os Dados - Objectivo

Os dados de estudo deste trabalho foram fornecidos pela equipa de investigação Chromosome Instability & Dynamics Lab, liderada pelo investigador Hélder Maiato, do Instituto de Biologia Molecular e Celular (IBMC).
Foram disponibilizados filmes de células de organismos diversos, nomeadamente *Drosophila* S2 e Neuroblastos, durante a metafase do ciclo celular (figura 1.4).

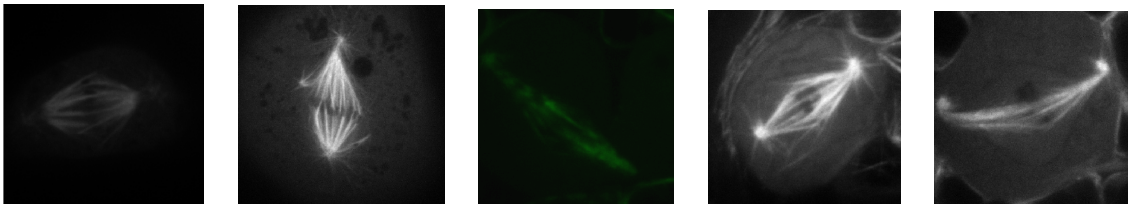

Figura 1.4: Fusos mitóticos de células Drosophila S2 durante métafase (IBMC).

Os objetivos deste trabalho são desenvolver algoritmos automáticos que permitam determinar as velocidades de fluxo e tempo característico de turnover nos dados fornecidos, e perceber se existe variação espacial destas taxas ou não. Por exemplo, perceber se há variações de comportamento entre a zona próxima dos cromossomas e a zona dos polos do fuso.

Induced Speckle Imaging

O fuso mitótico pode ser observado utilizando várias técnicas de microscopia. Contudo, os microtúbulos, as unidades estruturais primárias dos fusos, são demasiado pequenos para serem visíveis através de microscópios de luz visível. Hoje em dia há poderosas técnicas de microscopia de fluorescência que permitem acompanhar uma ou mais proteínas num fuso mitótico ao longo do processo mitótico (Rieder, 2006).
A técnica Induced Speckle Imaging (ISI) consiste em associar moléculas de fluorescência (GFP) às proteínas de tubulina dos microtúbulos e posteriormente remover a fluorescência em pequenas zonas, estabelecendo assim marcas de fluorescência de referência, denominadas de *speckles* (figura 1.5). Esta técnica foi desenvolvida pelo investigador António Pereira da equipa Chromosome Instability & Dynamics Lab.

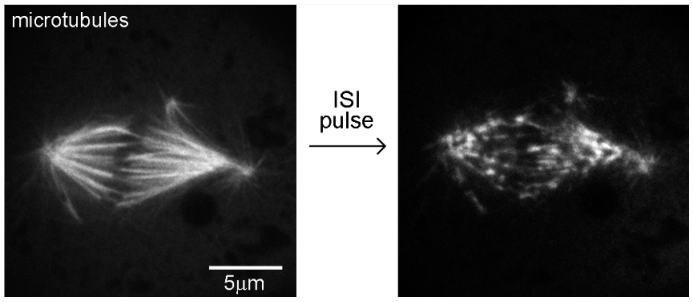

Figura 1.5: Imagem de *Drosophila* S2 obtida por microscopia de fluorescência. Induced Speckle Imaging: *speckles* no fuso mitótico (direita). Figura fornecida por cortesia de António Pereira, Chromosome Instability & Dynamics Lab.

A velocidade de deslocamento dos *speckles* na direção dos polos torna-se uma medida do fluxo (figura 1.5). Para além de possibilitar medições do fluxo, o tempo característico de desvanecimento das marcas fluorescentes permite obter medidas da taxa de turnover.
A fluorescência induzida nas células afeta todo o espectro das imagens, pelo que o decaimento da fluorescência total não traduz necessariamente o fenómeno de turnover dos microtúbulos. Neste trabalho assumiremos que a perda de contraste dos microtúbulos devido ao fenómeno de turnover é significativamente mais rápida do que as flutuações devido a alterações estruturais.

Um problema desta análise é que, em condições normais, os microtúbulos têm um nível de fluorescência bastante uniforme. As flutuações espaciais do sinal que emitem são residuais e devem-se apenas ao carácter estocástico do processo de ligação GFP-tubulina. Esta uniformidade espacial do sinal tem duas implicações no método:

i. Pode perturbar a observação do fluxo

ii. Obscurece os processos de substituição dos microtúbulos que acontecem a todo o momento.

1.3 Abordagens existentes - Revisão de Literatura

Em Pereira and Maiato (2010) foram apresentados, analisados e discutidos quimogramas obtidos para células *Drosophila S2*. Esta técnica consiste numa representação *bi* ou tridimensional de regiões de interesse, em que a variável tempo é uma das dimensões, permitindo uma inferência da velocidade direta por relação de distâncias nos gráficos.
Em Yang et al. (2008) foi possível efetuar seguimento individual de *speckles* de células de ovos de *Xenopus laevis* em meiose. Utilizou-se microscopia de fluorescência quantitativa para os mapeamentos de alta resolução. Ajustou-se uma função normal multivariada, de dois modos, à função de distribuição das velocidades dos *speckles*. Esta análise estatística permitiu determinar variação regional da velocidade dos *speckles* ao longo do fuso, resultando num modelo de fuso mitótico de dois grupos de microtúbulos estruturalmente em cada semi-fuso.
Em Matos et al. (2009) utilizou-se um modelo mecânico de translocação de microtúbulos em direção aos polos associados ao fluxo que reflete a relaxação do interface cinetocoro-microtúbulo. Foi verificado experimentalmente que acelerações na mitose impedem equalização de tensões e sincronização durante a anfase.

Capítulo 2

Abordagens aos problemas

2.1 Análise do fluxo

A alteração da distribuição tridimensional dos cromossomas durante a metafase para uma distribuição bidimensional no eixo do fuso mitótico, formando a placa metafásica (Pereira and Maiato, 2010), permite que análises de fusos a duas dimensões em metafase possam efetuar boas avaliações da realidade.
Numa tentativa de automatizar um método de cálculo do fluxo dos microtúbulos, procurámos um processo que requeresse, em cada filme, apenas um número reduzido de parâmetros fornecidos pelo utilizador.
Devido a alterações da posição relativa dos fusos nos frames dos filmes, para qualquer método seria necessário estabilizar o fuso. Isto foi conseguido com a marcação de 3 pontos em cada frame: dois de localização - os polos do fuso, e um terceiro - a largura do fuso. A marcação destes três pontos em cada frame de um filme permite estabelecer invariâncias espaciais para os fusos, quer a nível de rotação e translação do fuso, quer posteriormente a um ajuste de escala dos fusos. As imagens resultantes deste controlo de invariâncias efetuado mediante a marcação dos polos são as imagens originais rodadas, centradas sobre o fuso e limitadas ao retângulo que inscreve o fuso. O conjunto destas imagens foi denominado de novas imagens.

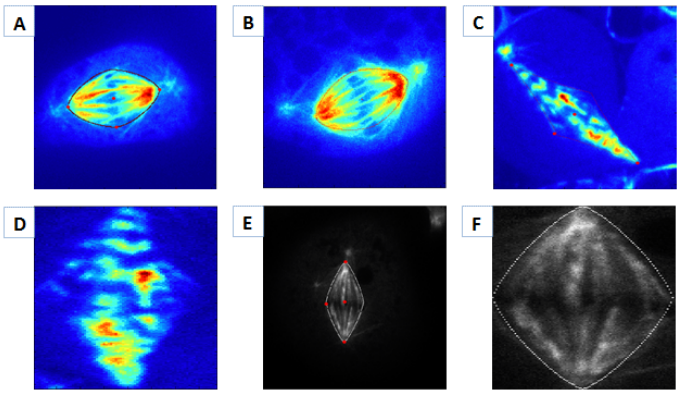

Figura 2.1: Fusos com forma super-elíptica ajustada através do MATLAB. (A-B) Célula de *Drosophila S2*. (C) Neuroblasto. (D) Imagem C rodada e centrada no MATLAB. (E) Célula de *Drosophila S2*. (F) Imagem E rodada e centrada sobre o centro do fuso, limitada ao retângulo exterior ao fuso.

Após observação dos filmes, verificou-se que os fusos têm geralmente uma forma super-elíptica

(figura 2.1). Optou-se pela elaboração de um método que tirasse proveito da forma super-elipsoidal que praticamente todos os fusos mitóticos observados adotam durante a metafase (figura 1.4). Esta abordagem permite aplicar às novas imagens um mapa conformacional super-elíptico. Com a aplicação deste mapa, torna-se possível analisar o fluxo ao longo dos microtúbulos a uma dimensão apenas, e não necessariamente nas duas dimensões das imagens (figura (2.2)). Os mapas conformacionais trazem também a vantagem dos movimentos na direção ortogonal do fluxo, que poderiam confundir estimativas de fluxo, serem descartados.

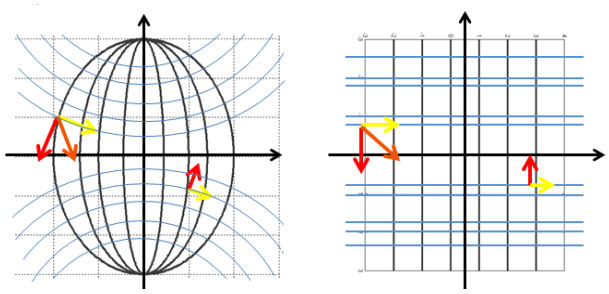

Figura 2.2: Ilustração de eliminação do ruído com a aplicação do mapa conformacional

Investiu-se em duas formas diferentes de avaliação do fluxo. Uma tira proveito da redução de dimensão dos microtúbulos após a aplicação do mapa conformacional adequado, estimando o fluxo a uma dimensão (Figura 2.3). A outra tira proveito também da forma elipsoidal dos fusos mas sem aplicação do mapa conformacional, o que permite trabalhar com imagens que sofrem menos alterações.

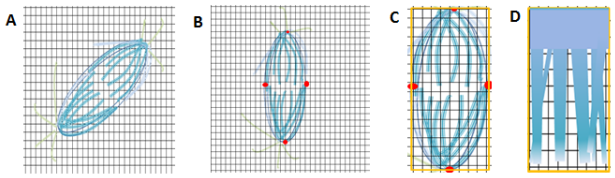

Figura 2.3: Ilustração da manipulação de imagens com os métodos de cálculo de fluxo a 1D e 2D, dividida por 4 fases. (A) Fuso na posição inicial. (B) Rotação da imagem sobre o centro do fuso. (C) Corte da imagem em B pelo retângulo limitado aos 3 pontos. (D) Semi-fuso superior após aplicação do mapa conformacional à imagem em C. No cálculo de fluxo a 1D percorremos as 4 fases, enquanto que no cálculo de fluxo a 2D apenas as 3 primeiras fases.

2.2 Método implementado

De seguida será explicada a abordagem adotada no algoritmo do cálculo do fluxo implementado no MATLAB (2010). O código elaborado para implementação deste método foi o código CalcularFluxo.m que será explicado na secção 2.2.1.. Este código tem como sub-rotinas as funções LeituraFramesPolosMiusTempos.m, ConfMapping.m, Calc_ Fluxo_ NFrames_ 1D.m, Calc_ Fluxo_ NFrames_ 2D.m e ConfMapping_ inverso.m.

Estabelecer invariâncias previamente:

Como referido anteriormente, para analisar fluxo ao longo de um filme é necessário estabelecer as invariâncias fundamentais para todos os frames do filme.

Invariâncias:

- Rotação (garantida após marcação dos três pontos de referência)
- Translação (garantida após marcação dos três pontos de referência)
- Escala

As condições de estabilidade dos frames são fundamentais para que os movimentos medidos nos microtúbulos não estejam corrompidos por movimentos do fuso mitótico, movimentos de câmara, etc..
Com os pontos de referência torna-se possível criar o conjunto das novas imagens todas elas centradas o fuso vertical, através da rotação e translação da imagem. Para marcar os pontos de referência nos frames foi criado o programa MarcarPolos.m, que funciona independentemente do programa de cálculo do fluxo, e será apresentado no capítulo 4.

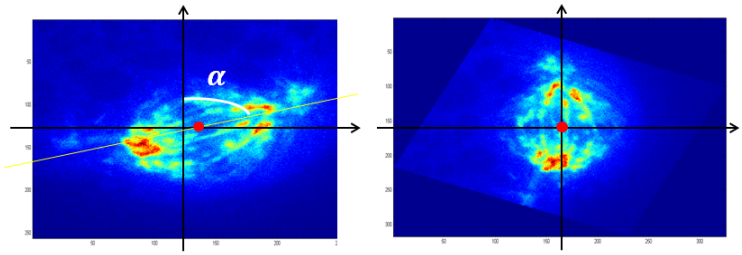

Figura 2.4: Rotação e translação do fuso mitótico.

O ajuste de escala das imagens é efetuado no código do cálculo do fluxo, CalcularFluxo.m, imediatamente antes de serem aplicados os métodos para o cálculo do fluxo.

Mapa conformacional:

Para o cálculo do fluxo a uma dimensão, é necessário produzir os mapas conformacionais super-elipsoidais de cada nova imagem. A adaptação da super elipse é efetuada no programa MarcarPolos.m aquando da marcação dos pontos em cada frame. Para além dos três pontos que o utilizador tem de marcar para garantir as invariâncias referidas, há um quarto parâmetro que o utilizador estabelece para todos os frames: o parâmetro μ da super-elipse, que determina a curvatura da super-elipse adaptada ao frame.

Equação normal reduzida da super elipse:

$$\left(\frac{x-x_0}{a}\right)^{\mu+1} + \left(\frac{y-y_0}{b}\right)^{\mu+1} = 1 \tag{2.1}$$

em que o parâmetro μ representa o grau de mapeamento cónico/elíptico.

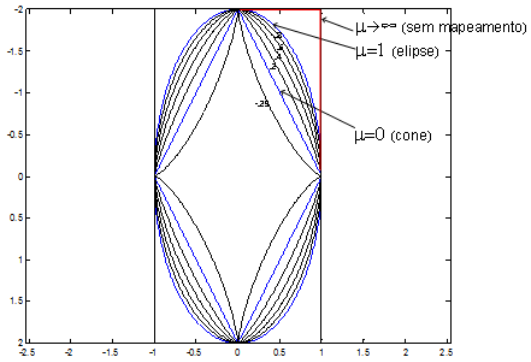

Figura 2.5: Ilustração de super elipses com diferentes valores de μ. $\mu = 1$ e $\mu = 0$ caraterizam os mapeamentos elíptico e cónico, respetivamente. $\mu = \infty$ não aplica mapeamento.

No programa MarcarPolos.m, após a marcação dos pontos de referência e do parâmetro μ em cada frame, o filme é automaticamente transformado numa pilha de imagens razoavelmente alinhadas (as novas imagens), podendo ser ou não aplicado o mapa conformacional a estas novas imagens. A aplicação do mapa conformacional à pilha de imagens depende unicamente da opção do utilizador de calcular o fluxo a uma ou a duas dimensões.
Na figura 2.6 são apresentados alguns fusos, novas imagens e resultados de aplicação dos mapas conformacionais. Ao longo das colunas de sub-imagens, os μ's das super-elipses são decrescentes, tendo sido atribuído ao fuso da última coluna um valor de μ negativo.

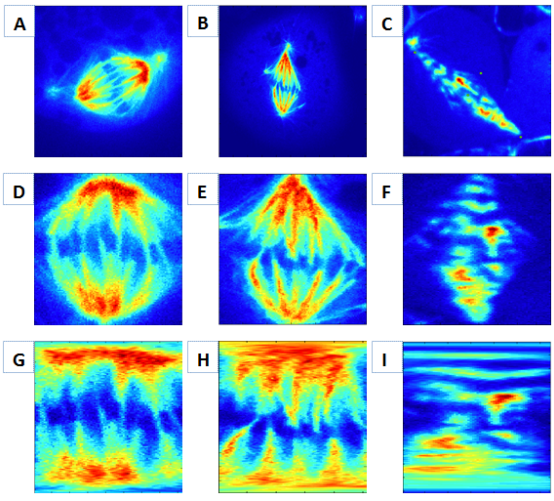

Figura 2.6: Frames originais, fusos e mapas conformacionais. (A,B) Imagem de *Drosophila S2*. (C) Imagem de célula de neuroblasto. (D,E,F) Imagens anteriores rodadas e cortadas pelos fusos mitóticos. (G,H,I) Mapas conformacionais das imagens da fila 2.

Assim, as imagens que serão utilizadas para análise de flxuo serão as novas imagens (Figura 2.6, 2ª linha) ou os mapas conformacionais destas (Figura 2.6 3ª linha), consoante se pretenda avaliar o fluxo a 2 ou 1 dimensões, respetivamente.

Algoritmo baseado na estratégia de Block Matching

O método segue agora para um ciclo que irá percorrer todos os pixeis das primeiras duas dimensões

da pilha, como ilustrado na figura **??**.

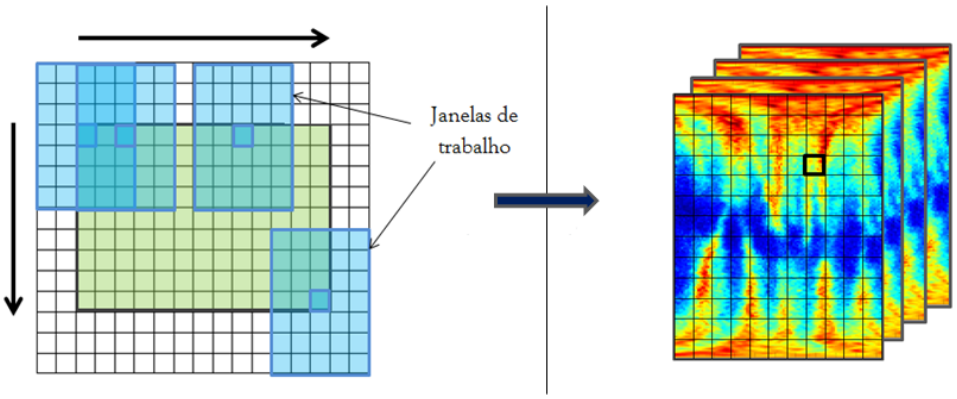

Figura 2.7: Início do tratamento da pilha de imagens selecionadas no algoritmo baseado na estratégia de Block Matching. Ilustração dos blocos (ou janelas) de trabalho. Pilha de imagens dividida por janelas de trabalho.

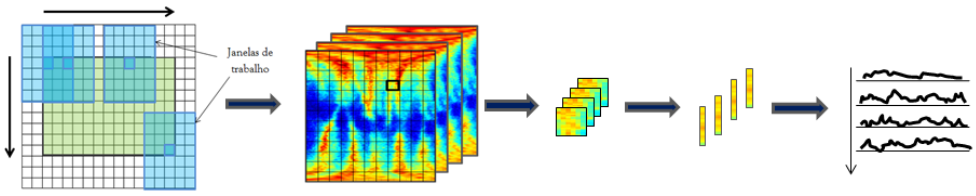

Figura 2.8: Sequência ilustrativa do algoritmo na fase de Block Matching.

O algoritmo percorre os pixeis do primeiro frame da sequência a avaliar. Será agora descrito o método utilizado para calcular o deslocamento de cada pixel da primeira imagem ao longo da sequência de frames.
O método de cálculo do deslocamento é baseado na estratégia de Block Matching . Este algoritmo percorre os dois eixos (linhas e colunas) da pilha de imagens selecionadas, gerando sub-pilhas de pequenas janelas centradas nos pixeis do ciclo, como ilustrado na figura **??**.
Considerando uma gama de velocidades previamente definida pelo utilizador, esta é usada para selecionar centros de janelas a serem escolhidas em cada frame, consoante as velocidades em questão. O esquema da figura 2.9 ilustra esta seleção das janelas em frames consecutivos de forma simplificada.
Na análise de cada sub-pilha, após a seleção da sequência de janelas pretendida, estas são colapsadas ao longo das suas linhas na mediana ou no máximo das linhas das janelas (Figura 2.10). Serão, a partir daqui, estes vectores colapsados que serão tratados e analisados.

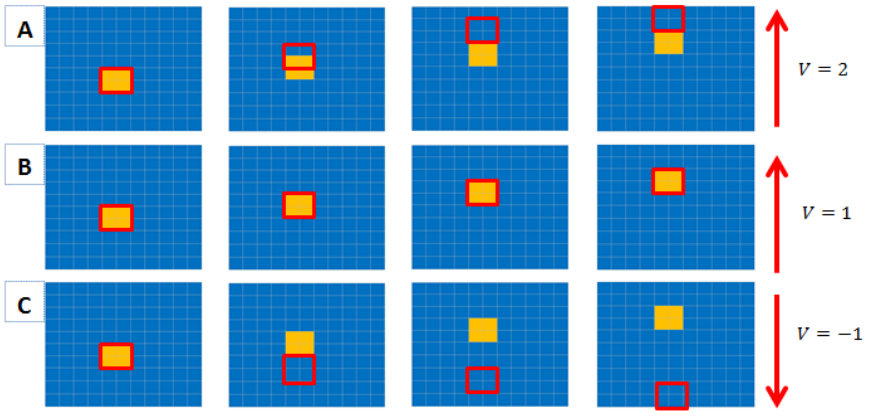

Figura 2.9: Ilustração do block matching. Nos três casos, A,B e C, o bloco amarelo desloca-se para cima a uma velocidade de 1 *px/frame*. A vermelho estão representadas as janelas geradas em função da velocidade. (A) A vermelho, janelas obtidas para uma velocidade da janela de 2 *px/frame* (B) Janelas obtidas para uma velocidade da janela de 1 *px/frame*. (C) Janelas obtidas para velocidade da janela -1 *px/frame*.

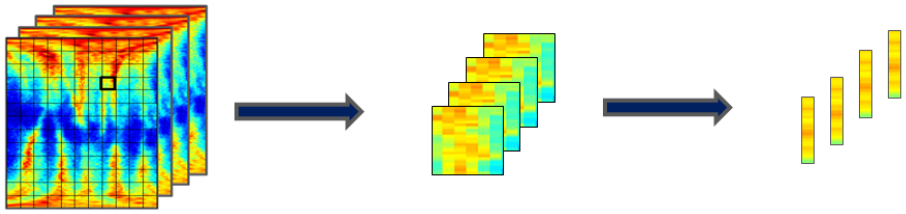

Figura 2.10: Ilustração das regiões das pilhas, das sub-pilhas selecionadas e da condensação das janelas em vetores.

Aqui, dois operadores diferentes podem ser aplicados a estas pilhas locais para avaliar se uma velocidade particular testada descreve adequadamente as transições de intensidade espacial entre os frames:

1. Função que generaliza a correlação (cruzada) a n sinais
2. Função de Diferenças Absolutas em relação à Média (DAM)

A ferramenta de Block Matching (Bradski and Kaehler, 2008) disponível no MATLAB permite detetar uma porção (ou bloco) de uma imagem f_i numa imagem f_j distinta de f_i. Da posição num e noutro frame deste bloco resulta imediatamente o deslocamento do bloco entre os dois frames. Para a abordagem baseada na estratégia de Block Matching aqui apresentada foi criada uma ferramenta de cálculo de deslocamento que permite utilizar a informação de n porções (ou blocos) na deteção do movimento ao longo de uma sequência de imagens.
Esta estratégia foi desenvolvida para se poder utilizar a informação de todos os frames contida na sub-pilha na determinação da velocidade, do primeiro para o último frame da pilha.
Assim consegue-se reduzir o impacto dos elevadíssimos níveis de ruído na imagem para a determinação dos valores de fluxo, uma vez que é tida em conta muito mais informação disponível nos filmes.

Para uma sequência de n frames há $n-1$ graus de liberdade nas operações de combinação de movimentos a considerar entre os vários vetores em análise, o que implica muito tempo de processamento do algoritmo.
Contudo, se assumirmos que em pequenos intervalos de tempo, e em regiões de pequena dimensão, a velocidade de deslocamento da tubulina nos microtúbulos (ou, analogamente, dos *speckles* nas imagens) é constante, o problema pode ser simplificado. Ao assumirmos esta hipótese, não é necessário testar todos os movimentos possíveis dos pixeis (ou dos *speckles*), mas apenas os movimentos que os pixeis efetuariam se se deslocassem a determinadas velocidades. Assim, passamos a considerar o deslocamento em função das possíveis velocidades dos *speckles*, $\Delta x = v \cdot \Delta t$. Em termos computacionais, esta simplificação permite tornar o algoritmo muito mais rápido, sem perder eficácia.
Na figura 2.11 está ilustrada a simplificação assumida.

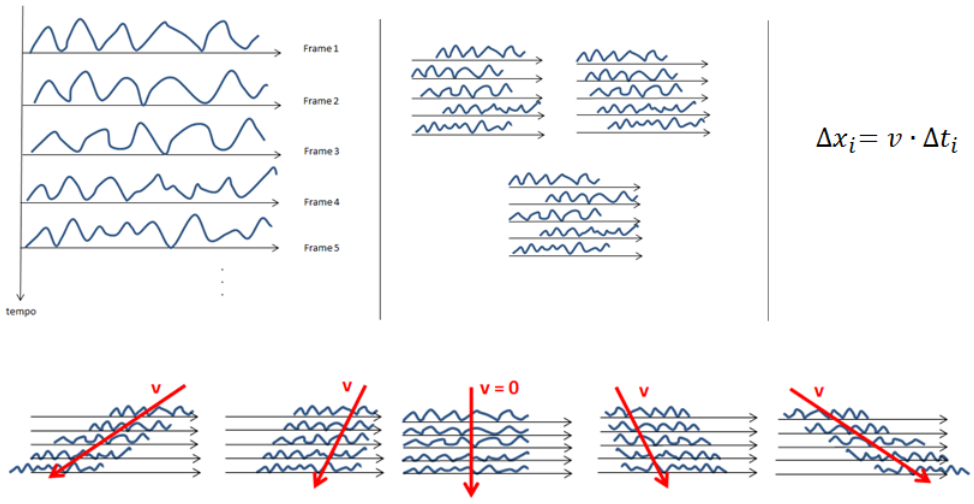

Figura 2.11: Ilustração dos gráficos das intensidades de matrizes de trabalho hipotéticas de frames consecutivos. Simplificação da análise de velocidade na estratégia baseada no algortimo de Block Matching.

As funções que são então por fim utilizadas para relacionar os diferentes deslocamentos possíveis e determinar a velocidade num pixel são:

Função de Correlação generalizada adaptada (caso discreto):

$$GCorr(f_1, ..., f_n, v) = h(v) = \sum_{x=1}^{L} \prod_{i=1}^{n} f_i(x - v \cdot \Delta t_i) \longrightarrow Max(h) \longrightarrow \hat{v} \tag{2.2}$$

Função de Diferenças absolutas em relação à media (caso discreto):

$$DAM(f_1, ..., f_n, v) = h(v) = \sum_{x=1}^{L} \sum_{i=1}^{n} |f_i(x - v \cdot \Delta t_i) - \langle f \rangle| \longrightarrow Min(h) \longrightarrow \hat{v} \tag{2.3}$$

O esquema da figura seguinte ilustra a análise geral do algoritmo baseado na estratégia de Block-Matching.

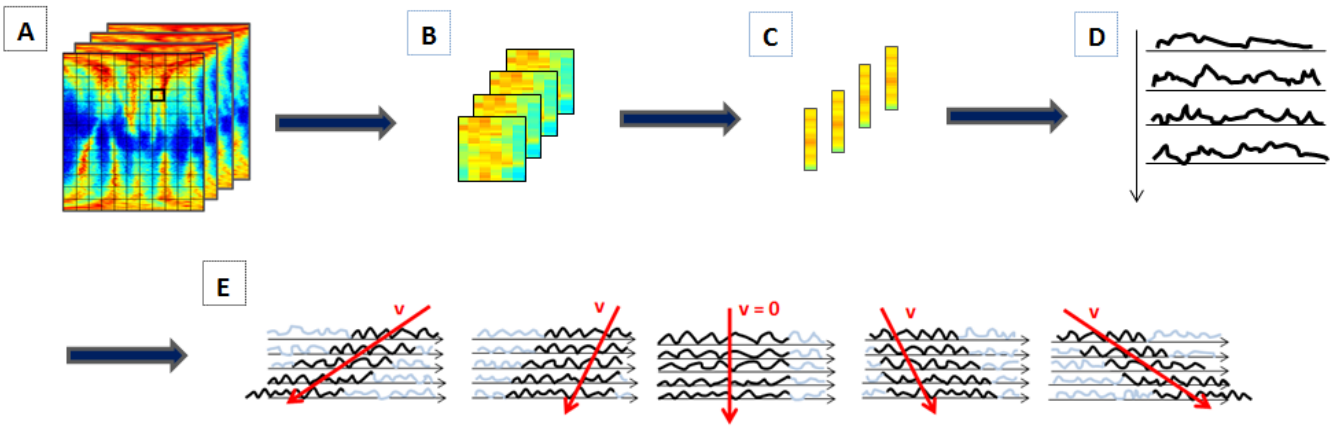

Figura 2.12: Esquema ilustrativo do cálculo do fluxo para um bloco. (A) Pilha de fusos após aplicação do mapa conformacional. (B) Sub-pilha ou pilha de janelas selecionadas. (C) Janelas da sub-pilha colapsadas em vetores-coluna. (D) Perfis de intensidade dos vetores-coluna. (C) Ilustração do conjunto de dados selecionados para avaliar em função da gama de velocidades.

A velocidade $\hat{v}$ que fica determinada para cada pixel é então aquela que maximiza a função de correlação generalizada adaptada (no caso de se utilizar esta função) ou a que minimiza a função de diferenças absolutas em relação à média (no caso de ser esta a utilizada).

Utilização de computação paralela

Dadas as características da análise de cada janela é possível distribuir os processos por vários processadores, permitindo que o algoritmo corra em paralelo, acelerando assim o processo. No Matlab, o comando `matlabpool` permite a funcionalidade completa de recursos em simultâneo, através de processamento em paralelo das tarefas.

A confiança nos resultados

Afim de avaliar se a velocidade que se determina pela estratégia baseada no Block Matching, $\hat{v}$, merece muita ou pouca confiança, criou-se um método de classificação da confiança das velocidades obtidas. O objetivo desta função é medir a relevância do máximo ou mínimo determinados pelo BlockMatching.
Um valor obtido pelo algoritmo, $\hat{v}$, é tão mais relevante quanto mais o valor de $GCorr(v)$ ou $DAM(v)$ se destacar dos restantes valores obtidos para os diferentes v's. Foi então utilizado um parâmetro de associação entre funções para medir a associação dos resultados obtidos com o comportamento ótimo previsto para estes casos.
Para cada sub-pilha analisada são registados os valores da função DAM, $DAM(v)$, ou da função de Correlação, $GCorr(v)$, para todas as velocidades testadas. Para generalizar os dois casos, seja $F(v) = DAM(v)$ ou $GCorr(v)$, consoante a função que tenha sido admitida.
Idealmente, o comportamento da função F deve indicar que o máximo ou mínimo atingidos se destacam dos restantes valores de forma coerente. Ou seja, quanto mais próximo da velocidade $\hat{v}$ for um valor de velocidade v, mais próximo de $F(\hat{v})$ deve estar $F(v)$. Assim, o valor de confiança nos resultados mede o grau de associação entre a função $F(v)$ e uma função monótona até $F(\hat{v})$, quando $v \to \hat{v}^+$ ou $v \to \hat{v}^-$. (Figura 2.13)
Para obter a confiança num determinado resultado, mede-se então o grau de associação então os valores de F com valores de referência específicos.

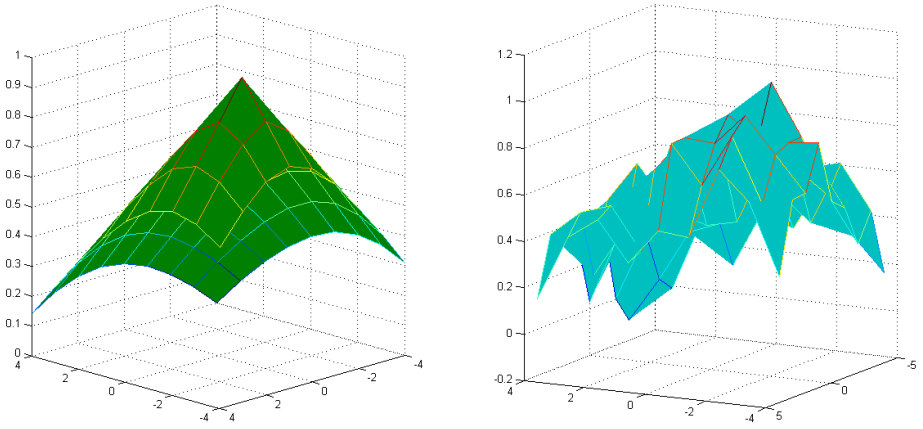

Figura 2.13: Exemplo de função de referência e função F normalizadas, em função das velocidades testadas (eixo ox, oy), para caso de análise de fluxo a 2D.

Os valores de referência:

Os valores de referência são criados individualmente para cada sub-pilha em função da velocidade $\hat{v}$ encontrada para essa sub-pilha e da gama de velocidades admitida.
No caso de análise de fluxo a uma dimensão, a função de referência é a função módulo deslocada do extremo de $F(v)$,

$$Referencia(v) = |v - \hat{v}| \tag{2.4}$$

onde $\hat{v}$ é a velocidade que otimiza a função $F(v)$ (ou seja, a encontrada na relação (3.2) ou (3.3)).

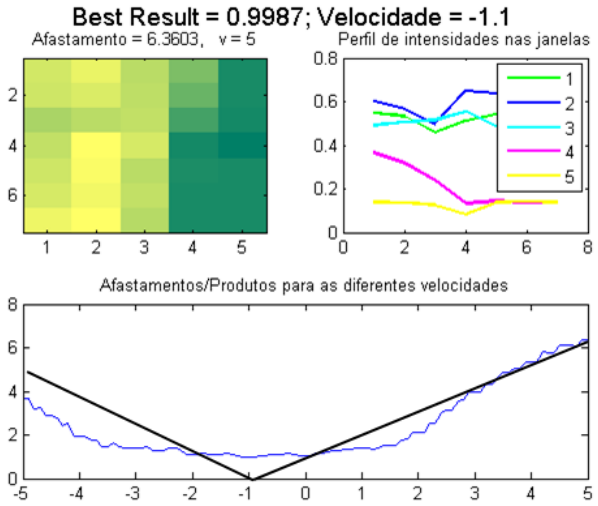

Figura 2.14: Cálculo do fluxo a uma dimensão numa sub-pilha. Figura produzida através do código CalcFluxo_ NFrames_ 1D.m. Em cima vários blocos da sub-pilha analisada. Em cima à direita, perfis de intensidade dos blocos da sub-pilha. Em baixo, a azul, função $F(v)$ em função das velocidades analisadas (eixo ox). A preto, função de referência produzida.

No caso do fluxo ser calculado a duas dimensões, a função de referência é uma função cónica:

$$Referencia(v) = \sqrt{(v_x - \hat{v_x})^2 + (v_y - \hat{v_y})^2}, \tag{2.5}$$

onde $(\hat{v_x}, \hat{v_y})$ é o extremo de $F(v_x, v_y)$.

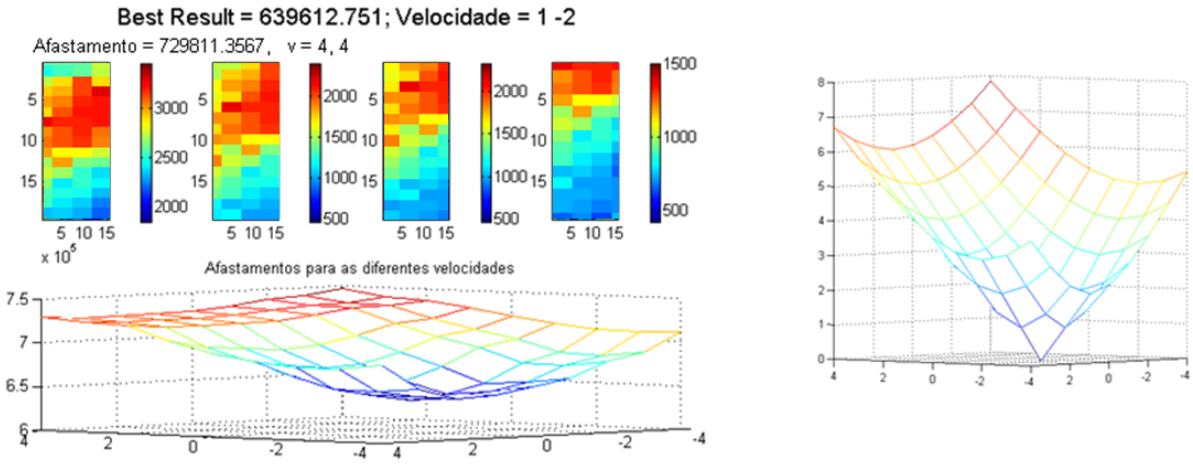

Figura 2.15: Cálculo do fluxo a duas dimensões numa sub-pilha. Figura produzida através do código CalcFluxo_ NFrames_ 2D.m. Em cima vários blocos da sub-pilha analisada. Em baixo, função $F(v)$ em função das velocidades analisadas (eixos ox, oy). À direita, função de referência produzida para o caso.

Para obter a confiança para o pixel em questão, calcula-se o grau de associação para o par de observações $F(v)$ e $Ref(v)$, $\forall v \in \{velocidades\ testadas\}$, que irá ser abordado como coeficiente de correlação.
O coeficiente de correlação escolhido foi o coeficiente τ de Kendall, em detrimento dos coeficientes de correlação r-Pearson e ρ-Spearman.
O objetivo com a função de correlação é verificar se o máximo ou mínimo atingidos são relevantes, mesmo que haja algumas oscilações no gráfico de $F(v)$.
O coeficiente de correlação de Pearson avalia a correspondência linear entre dados, o que não é o que se pretende nesta associação, porque nos casos em que as os valores comparados têm relação monótona, mas não linear, a confiança é alta e o coeficiente de Pearson não o traduz.
A ideia de correlação do coeficiente de correlação de Spearman é a substituição dos valores observados pelas suas ordens, ou *ranks*. Por exemplo, se uma amostra de três valores for $x_1 = 5.5, x_2 = 2.2, x_3 = 10.1$, então os respetivos *ranks* serão $r_1 = 2$, $r_2 = 1$, $r_3 = 3$. Após substituir cada uma das amostras pelos seus *ranks* o coeficiente de Spearman é a aplicação do coeficiente de Pearson aos *ranks* das variáveis originais.

$$\rho_S = 1 - \frac{6 \sum_{i=1}^{n} D_i^2}{n^3 - n} \tag{2.6}$$

No caso de haver observações empatadas, atribui-se um *rank* igual à média das posições desses valores na ordenação ascendente dos valores.
Um outro parâmetro de associação entre variáveis é o coeficiente de Kendall. A correlação de Kendall é muitas vezes descrita como uma medida de concordância entre dois conjuntos de classificações relativas a um conjunto de objetos ou experiências. Tal como no coeficiente de Spearman, este parâmetro utiliza os *ranks* da amostra, e para o determinar há que inicialmente

ordenar ambas as classificações de acordo com a ordem crescente de *uma* delas, para a partir daí contar os pares que vão no mesmo sentido.
O coeficiente define-se da seguinte forma:
Sejam X e Y variáveis aleatórias i.i.d. e sejam $X_1, X_2, ..., X_n$, $Y_1, Y_2, ..., Y_n$ amostras aleatórias de X e de Y respetivamente.

$$\tau = \frac{\sharp C - \sharp D}{\sharp C + \sharp D} \tag{2.7}$$

Um par de observações (X_i, X_j) diz-se concordante se $X_i < X_j \wedge Y_i < Y_j$, ou igualmente $sign(X_j - X_i) \cdot sign(Y_j - Y_i) > 0$, e discordante caso $sign(X_j - X_i) \cdot sign(Y_j - Y_i) < 0$. Se $(X_j - X_i) \cdot (Y_j - Y_i) = 0$ não há concordância nem discordância.
Basicamente, este coeficiente mede a diferença entre a probabilidade dos valores de X e de Y estarem na mesma ordem ($\frac{\sharp C}{\sharp C + \sharp D}$)e a probabilidade de estarem em ordens diferentes ($\frac{\sharp D}{\sharp C + \sharp D}$), em que estas probabilidades são dadas através das frequências relativas respetivas.
Nos casos de empates, quando há valores repetidos na amostra, utiliza-se o coeficiente τ_b de Kendall, que se traduz por:

$$\tau_b = \frac{\sharp C - \sharp D}{\sqrt{\frac{1}{2}n(n-1) - T_x}\sqrt{\frac{1}{2}n(n-1) - T_y}} \tag{2.8}$$

onde T_x, T_y são o número de empates na variáveis X e Y, respetivamente, e $\frac{1}{2}n(n-1) = C_2^n$ é o número de pares (i, j) que se avaliam. A equação 2.8 é uma generalização da equação 2.7. Caso não haja empates, $T_x = T_y = 0$ e $\sharp C + \sharp D = C_2^n$.
É de notar que o conceito de concordância de Kendall não avalia se as classificações são iguais para ambos os avaliadores, mas sim se estas apontam num sentido comum de classificação da amostra. No nosso caso, uma vez que um avaliador é a função módulo (ou cónica), para cada valor de velocidade v, o algoritmo de Kendall avalia em que medida é que os restantes valores $F(v')$, estão acima ou abaixo do valor de $F(v_i)$, $(v' \neq v)$, relativizando estas medidas uma em relação à outra.
O coeficiente de Spearman traduz significativamente valores pontuais não concordantes de atribuições dos *ranks* a duas variáveis. Contudo, por vezes a relevância que atribui a estes valores não é compatível com a significância atribuída ao sentido comum de classificação das velocidades, ou seja, ao comportamento de monotonia (embora com algum ruído) como o da função módulo, o que é muito importante para a confiança que se pretende.
Por outro lado, em muitas situações os valores de τ e ρ são muito próximos. Em caso de discrepância o mais seguro é, por defeito, interpretar o valor mais baixo, que é geralmente o do coeficiente de Kendall (Fredricks and Nelsen, 2007) e (Gibbons, 1993).
Uma vez que a lógica computacional das suas fórmulas é bastante diferente, as duas correlações têm magnitudes diferentes. Em Fredricks and Nelsen (2007) mostrou-se, sob condições de regularidade amenas que, à medida que a distribuição conjunta das variáveis aleatórias se aproxima de independência, a relação $\rho/\tau = 3/2$.
O coeficiente de correlação escolhido como medida de associação entre $F(v)$ e $Referencia(v)$ foi então o coeficiente de correlação de Kendall.
De seguida é apresentado um pequeno exemplo de determinação do τ_b de Kendall.

Exemplo: Suponhamos que num caso hipotético de avaliação do fluxo a 1D estamos a avaliar a gama de velocidades $\{-3, -2, -1, 0, 1, 2, 3\}$ $px/frame$. Sejam, por hipótese, valores de $R(v)$ os apresentados na tabela seguinte, e a correspondente função de referência, $Ref(v)$, a apresentada:

Tabela 2.1: Exemplo de valores obtidos para F e para Ref. A verde, $F(\hat{v})$

v	-3	-2	-1	0	1	2	3
$F(v)$	5.56	4.19	4.03	2.98	1.60	4.03	3.74
$Ref(v)$	4	3	2	1	0	1	2

Ordenando a tabela anterior consoante a ordem crescente de $F(v)$ resulta a tabela seguinte:

Tabela 2.2: Valores exemplo obtidos para R e para Ref ordenados;

v	1	0	3	-1	2	-2	-3
$F(v)$ (cresc)	1.60	2.98	3.74	4.03	4.03	4.19	5.56
$Ref(v)$	0	1	2	1	2	3	4

Para obter o τ_b de Kendall procedemos da seguinte forma:
Para a primeira coluna, em $v = 1$, $F(v_j) > F(v_1), \forall j > 1$ e $Ref(v_j) > Ref(v_1), \forall j > 1$. Ou seja, todos os valores de F à direita da primeira entrada são superiores a 1.60 e todos os valores à direita de 0 são superiores a 0. Tem-se portanto 6 casos concordantes e 0 casos discordantes. Fixando a segunda coluna, à direita de 1 ($= Ref(0)$) temos 4 valores superiores a 1, e à direita de 2.98 ($= F(0)$) temos 5 valores superiores a este valor. Portanto aqui temos apenas 4 casos concordantes ($sign(4.03 - 2.98) \cdot sign(1 - 1) = 0$, logo não é caso concordante) e de novo 0 casos discordantes. Para a terceira coluna, temos apenas duas entradas superiores a 2 ($= Ref(3)$), e uma menor do que 2. Todos os valores à direita de 3.74 são superiores a este valor, e portanto temos dois casos concordantes. Como $Ref(-1) < Ref(3)$, temos um caso discordante. Na quarta coluna, à direita de 1 temos 3 valores superiores a 1, mas para a linha de cima há um valor igual a 4.03, e portanto não é contabilizado como caso concordante. Aqui temos 2 casos concordantes e 0 discordantes. Na coluna 5 temos dois casos concordantes e na 6ª coluna há 1 caso concordante.
No total, há 17 casos concordantes e 1 caso discordante. O número de pares repetidos em $F(v)$ é 1, (4.03), e o número de pares repetidos de $Ref(v)$ é 2, (1) e (2). Portanto

$$\tau_b = \frac{17 - 1}{\sqrt{\frac{1}{2} \cdot 7 \cdot 6 - 1}\sqrt{\frac{1}{2} \cdot 7 \cdot 6 - 2}} = \frac{16}{\sqrt{21 - 1}\sqrt{21 - 2}} = 0.8208$$

De facto, os valores de Ref e F têm o mesmo sentido, o que é traduzido pelo elevado valor de τ.

2.2.1 O programa

O algoritmo para calcular o fluxo numa sequência de frames de um filme foi implementado no *script* CalcularFluxo.m, cuja descrição detalhada pode ser consultada no ANEXO A.
Para uma melhor compreensão do algoritmo e do programa elaborado apresenta-se de seguida um diagrama de fluxo que ilustra as principais etapas do algoritmo.

Diagrama de fluxo do código Matlab CalcularFluxo.m

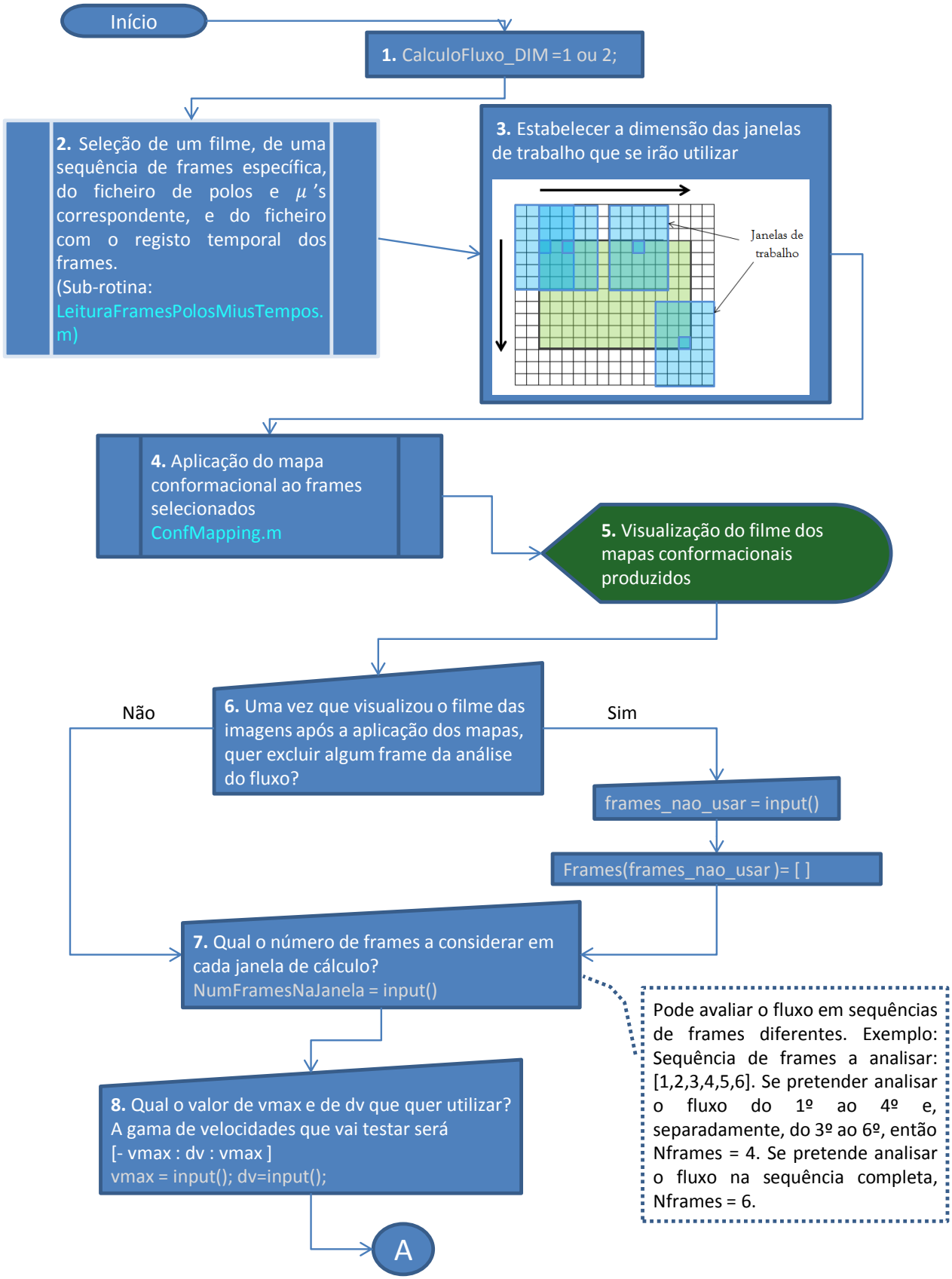

Figura 2.16: Programa CalcularFluxo. Diagrama de fluxo (parte 1)

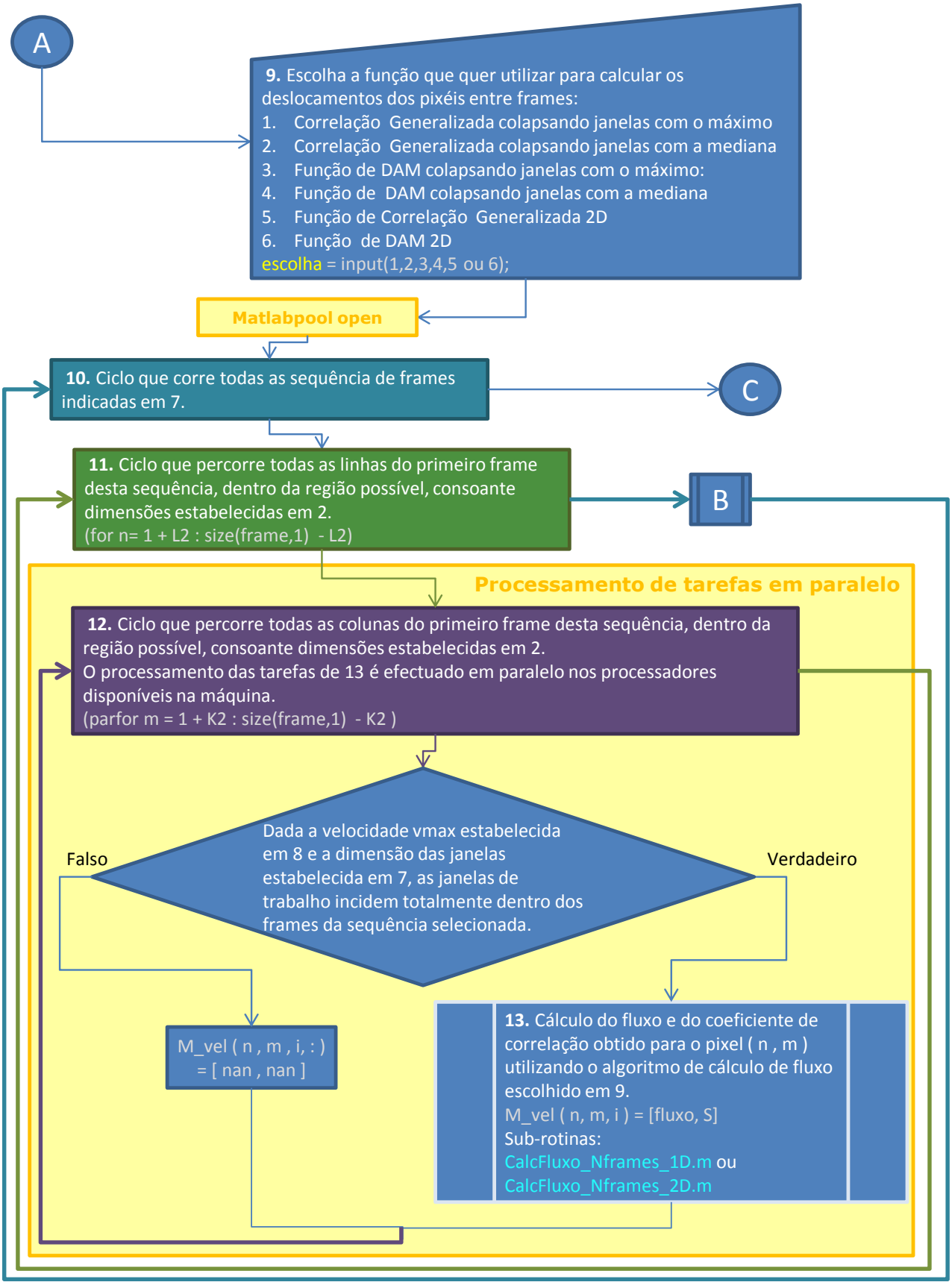

Figura 2.17: Programa CalcularFluxo. Diagrama de fluxo (parte 2)

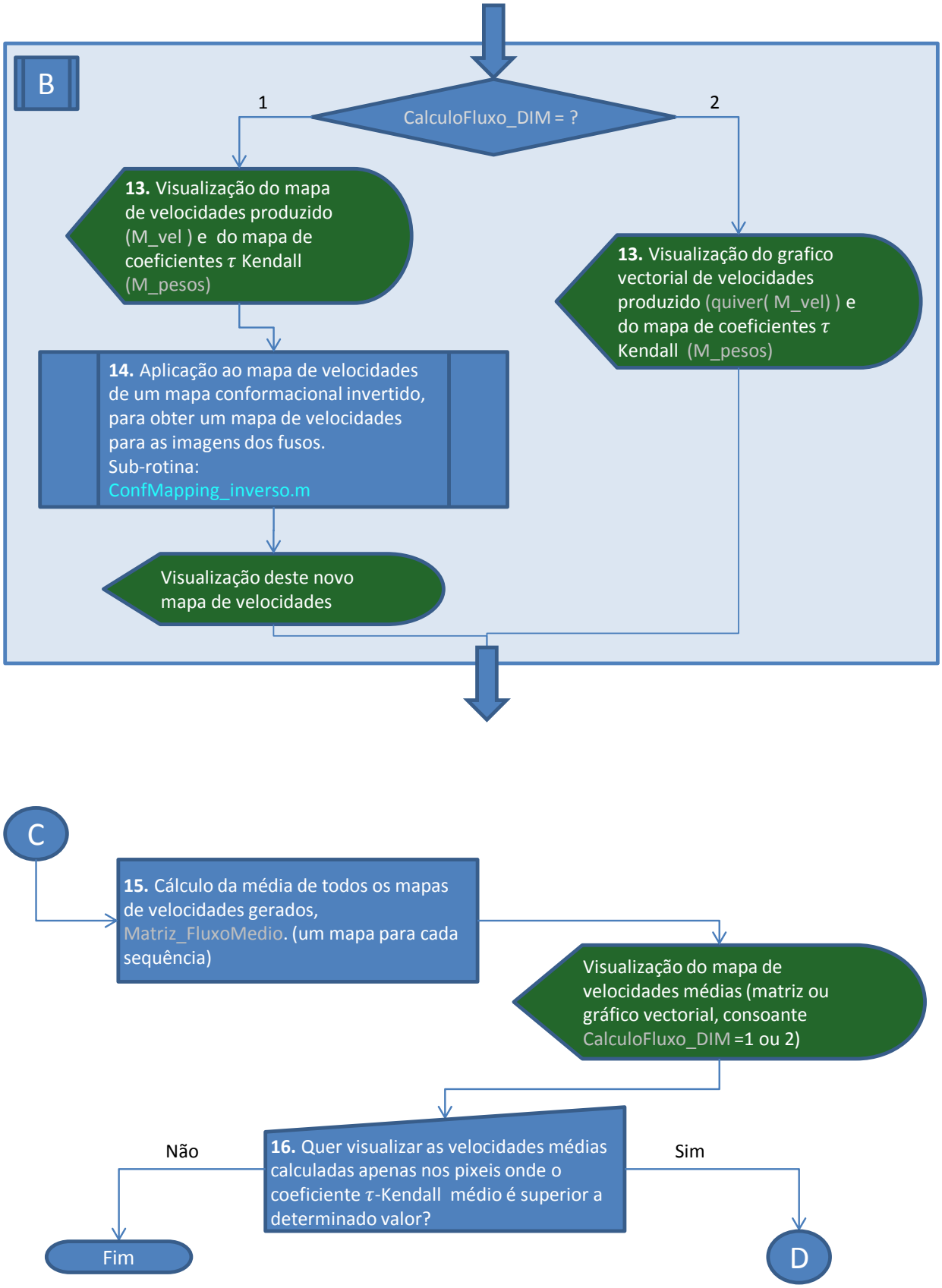

Figura 2.18: Programa CalcularFluxo. Diagrama de fluxo (parte 3)

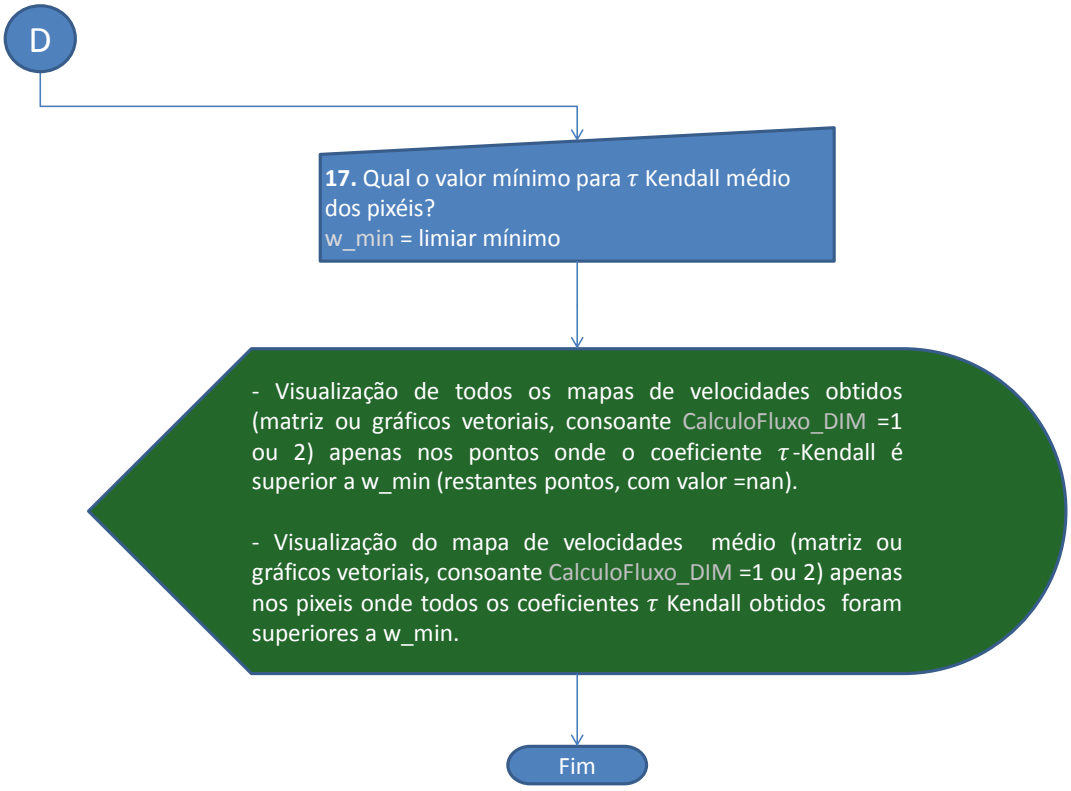

Figura 2.19: Programa CalcularFluxo. Diagrama de fluxo (parte 4)

2.3 Análise do Decaimento

Neste trabalho procurou-se ajustar a intensidade dos frames ao longo do tempo a uma função exponencial, segundo a lei descrita na equação seguinte.

$$I(t) = a \cdot \exp\left(-\frac{t}{decay}\right) \tag{2.9}$$

em que $I(t)$ é o valor da intensidade no instante t, t é o instante no tempo e $decay$ é o parâmetro de decaimento da intensidade.

Para análise de decaimento deve ter-se em conta que o decaimento das intensidades nos frames prende-se essencialmente com dois aspectos:

- Perda de fluorescência dos microtubulos
- Fenómeno de turnover

Graficamente, o comportamento da intensidade dos pontos marcados nos filmes, quer pontos de intensidade inicial mais alta, quer pontos de intensidade inicial baixa, pode ser visto como a combinação de dois mecanismos a atuar em escalas de tempo distintas (Figura 2.20).

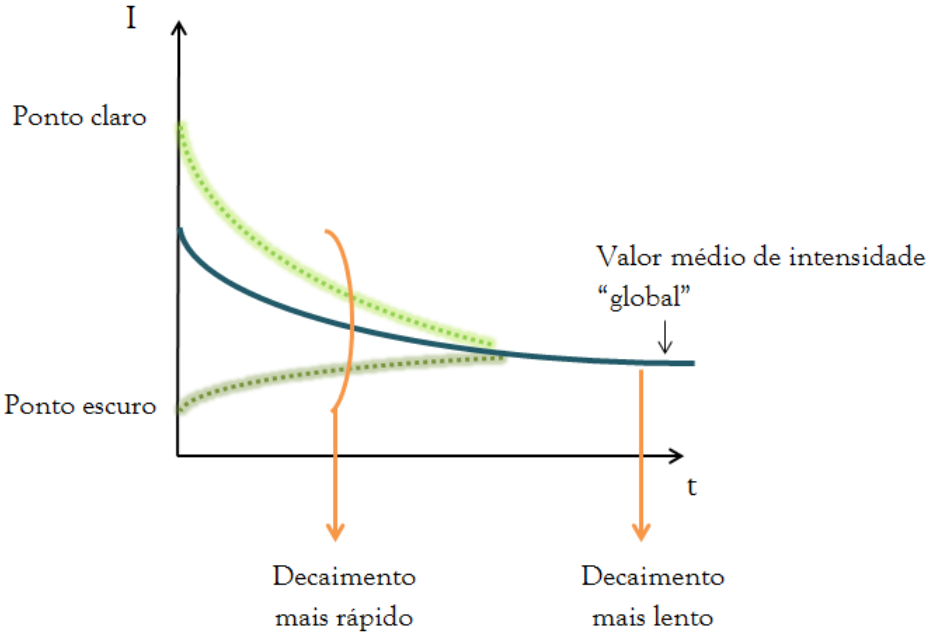

Figura 2.20: Ilustração de decaimento médio

Uma possível abordagem à determinação de turnover consiste em calcular a diferença entre as duas curvas exponenciais para obter uma região de confiança para a curva de turnover ao longo do tempo (2.20, curva azul).
Para esta região ser aceite, é necessário ter confiança nos ajustes exponenciais do decaimento de intensidade efetuados nas regiões de *speckles* e nas regiões escuras.

Os parâmetros utilizados para avaliar as regressões são o coeficiente de determinação e o coeficiente de correlação.
A regressão exponencial pode ser reduzida a um caso de regressão linear através da sua logaritmização.

$$y = ae^{bx} \tag{2.10}$$

$$\ln y = bx + \ln a \tag{2.11}$$

Fazendo $z = \ln y$, $w = ln(a)$, a equação 2.11 transforma-se na equação da reta:

$$z = bx + w.$$

No MATLAB, os parâmetros da regressão linear são determinados pelo método dos mínimos quadrados. O parâmetro inicial, $\hat{b}$, é o parâmetro determinado pelos mínimos quadrados e o parâmetro a vem da relação $a = e^{\hat{w}}$.

Confiança nos resultados:

Após calculada a regressão linear, é necessário verificar se os dados são descritos pelo modelo e determinar a variabilidade amostral que é de facto determinada pela equação encontrada.
Seja y_i um valor da amostra, $\hat{y}_i$ o valor da variável independente dado pela regressão e $\bar{y}$ a média

amostral. Considerando a relação simples

$$y_i = (y_i - \hat{y}_i) + (\hat{y}_i - \bar{y}) + \bar{y} \tag{2.12}$$

é possível demonstrar a equação

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2. \tag{2.13}$$

O primeiro membro desta equação pode ser determinado como proporcional à variância total de Y, enquanto que o segundo membro reflete a soma de termos proporcional à variância residual e explicada pelo modelo de regressão.

$$SQ_T = SQ_{Res} + SQ_F \tag{2.14}$$

em que SQ_T é a soma quadrática total, SQ_{Res} a soma quadrática dos desvios e SQ_F a soma dos desvios da regressão.
O coeficiente de determinação, R^2, é dado pela razão

$$R^2 = \frac{Var.\ Explicada\ pela\ Regressao}{Var.\ Total} = \frac{SQ_{Reg}}{SQ_T} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.15}$$

O coeficiente de determinação varia entre 0 e 1, e deve ser interpretado como a proporção da variância total da variável dependente Y que é explicada pelo modelo de regressão. Por outras palavras, ele traduz o quão melhor se pode prever y usando o modelo encontrado do que usando a média.
O coeficiente de correlação amostral, R, é a raiz quadrada do coeficiente de determinação, $R = \pm\sqrt{R^2}$.

2.3.1 O programa

O algoritmo para determinação de decaimento foi implementado no *script* DecaimentoSemiAutomatico.m.
O programa elaborado para determinar o fenómeno de decaimento é um programa semi-automático, onde o utilizador seleciona manualmente regiões dos frames originais (ie, sem invariâncias estabelecidas) onde pretende avaliar o decaimento.
Para uma melhor compreensão do algoritmo e do programa elaborado, apresenta-se de seguida um diagrama de fluxo com as etapas principais deste algoritmo.

Diagrama de fluxo do código Matlab DecaimentoSemiAutomatico.m

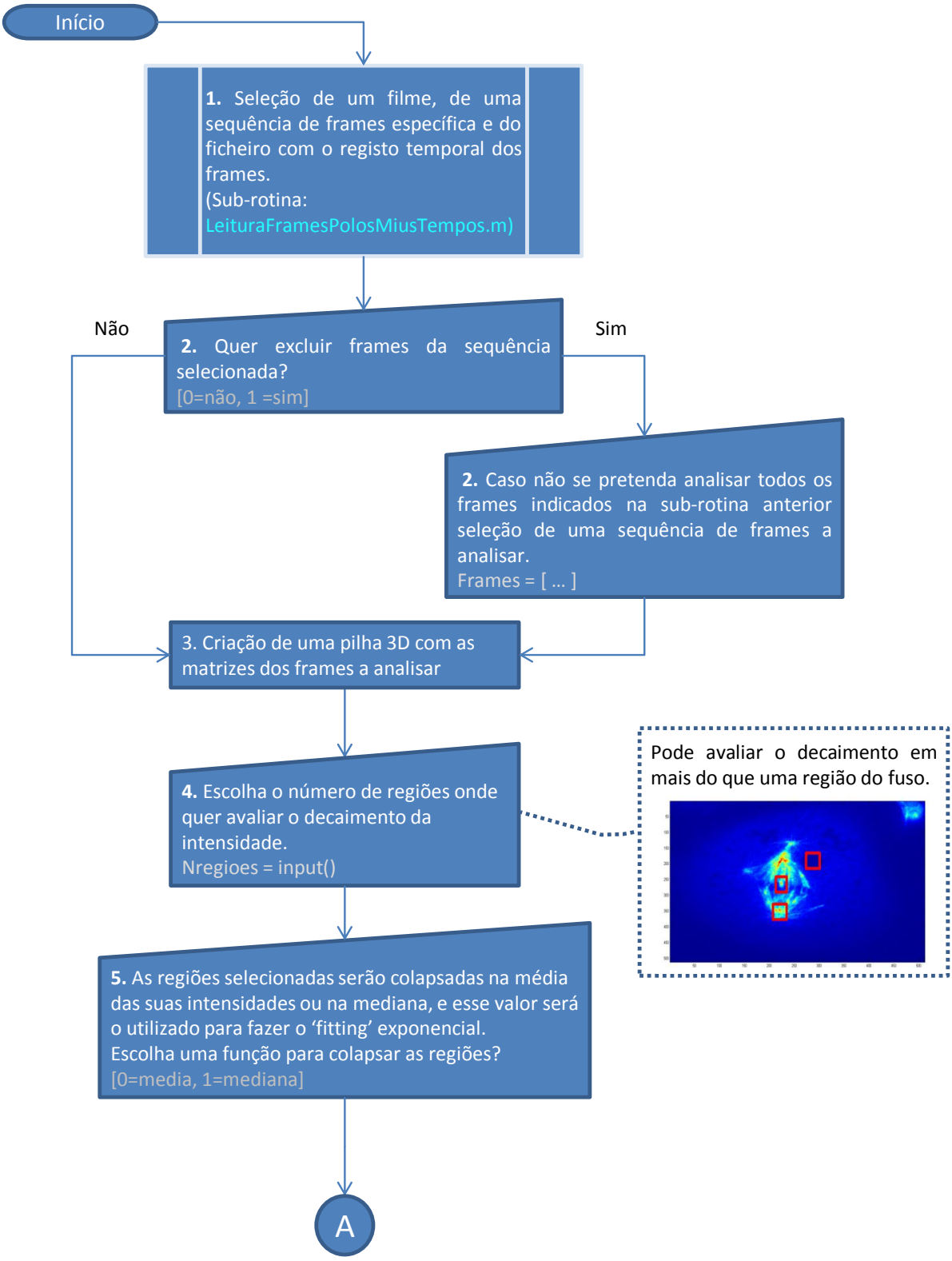

Figura 2.21: Programa DecaimentoSemiAutomatico, diagrama de fluxo (parte 1).

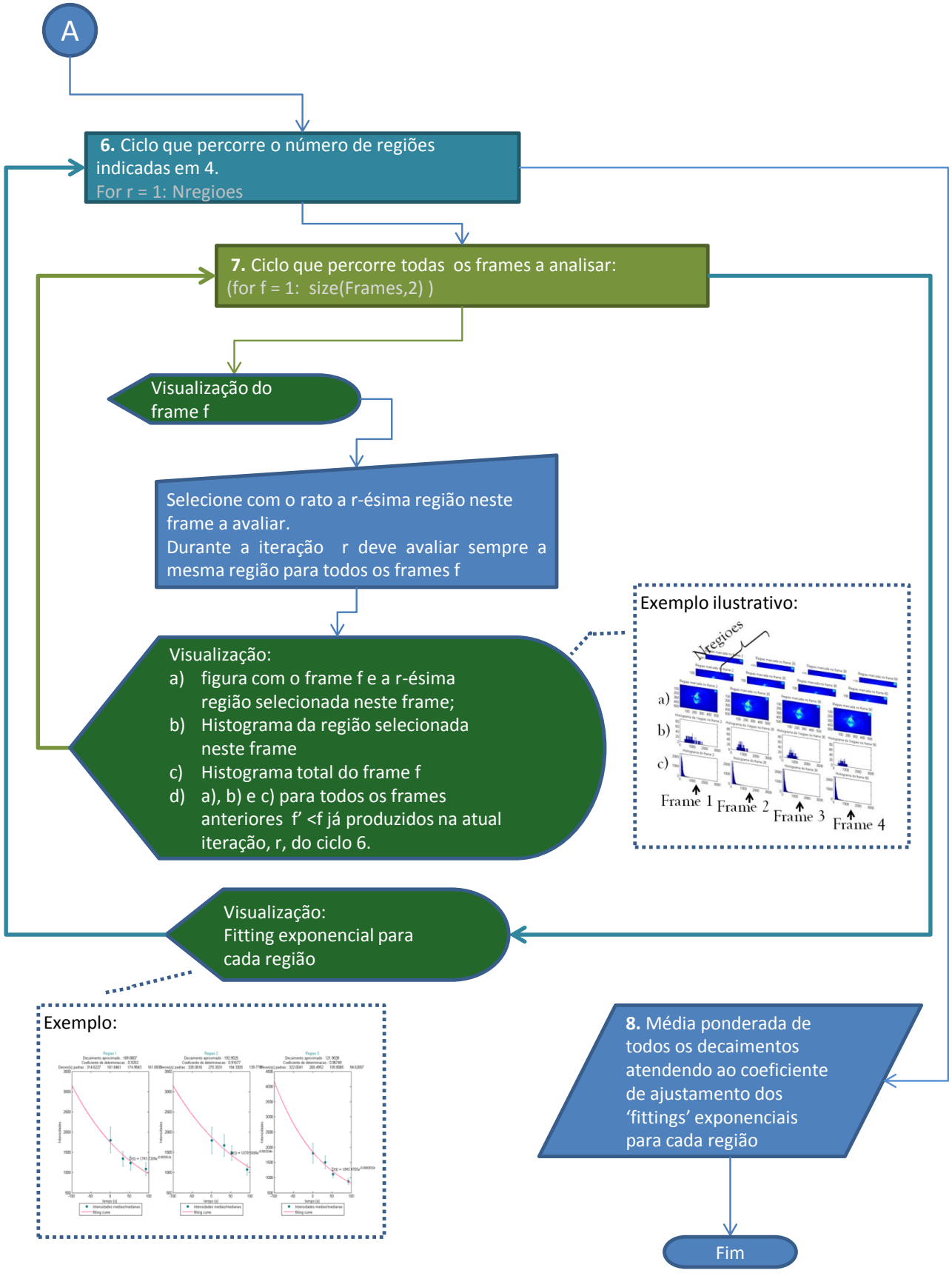

Figura 2.22: Programa DecaimentoSemiAutomatico, diagrama de fluxo (parte 2).

2.4 Desvantagens das abordagens

Cálculo do fluxo

O sucesso da abordagem adotada depende fundamentalmente da eliminação das variâncias posicionais dos fusos nas novas imagens, o que nem sempre é possível. Contudo, o programa MarcarPolos.m revelou-se eficaz e prático para estabelecer os pontos fixos dos fusos e os valores de μ.
Nos filmes disponibilizados pela equipa de investigação as células têm fusos de formas bastante diferentes. Apesar de uma grande parte dos fusos terem forma super-elipsoidal, muitos têm formas distintas no hemisfério superior e inferior, pelo que a forma super-elíptica não corresponde às formas destes fusos, como por exemplo no caso do fuso da figura 2.23.

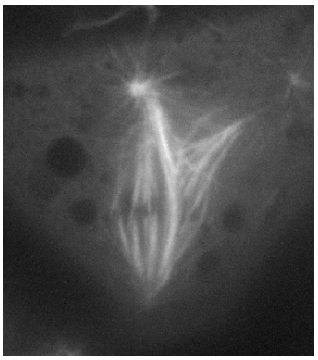

Figura 2.23: Fuso de *Drosophila S2* assimétrico.

As regiões onde é possível medir a velocidade são bastante limitadas devido à condição de que as janelas de trabalho têm de estar totalmente contidas em todos os frames a avaliar. A área que pode ser avaliada é limitada pela dimensão das janelas e também pela velocidade máxima que se pretenda avaliar.
Para obter melhores resultados no cálculo do fluxo, é importante maximizar o número de frames a usar, mas garantindo que o efeito de decaimento não seja notório.
O elevado número de frames a usar pode também contribuir para distorções nas velocidades calculadas. Estas distorções ocorrem, nomeadamente, nas situações em que é atribuída uma velocidade ao pixel do primeiro frame da pilha devido ao facto de, em frames posteriores, existirem *speckles* nas janelas da sua sub-pilha correspondente a deslocarem-se a essa velocidade, embora nos primeiros frames isso não aconteça.

Cálculo do decaimento

O algoritmo de cálculo de decaimento é semi-automático e necessita de precisão do utilizador na marcação das janelas a avaliar.
Deve ser realizado em múltiplos frames dispersos num largo intervalo de tempo. Isto porque a escala de tempo do decaimento é muito superior aos tempos característico de metafase.
Através do algoritmo referido para o cálculo do turnover, para avaliar este fenómeno é necessário comparar *fittings* diferentes produzidos pelo algoritmo, em casos bastante controlados, entre zonas de intensidade alta e zonas de intensidade baixa. Contudo, mesmo que estes procedimentos sejam efetuados com muita exatidão, não fica garantida uma boa determinação de *turnover*.

Capítulo 3

Programas auxiliares

3.1 Leitura de Frames

A sub-rotina que inicia os programas do cálculo do fluxo e do cálculo do turnover permite fazer a leitura dos filmes e dos parâmetros particulares de cada filme.
Esta sub-rotina foi implementada no script LeituraFramesPolosMiusTempos.m, e permite efetuar as seguintes opções:

1. Escolher o filme que se pretende analisar.
2. Visualizar o filme através do reprodutor de vídeos do Matlab.
3. Escolher uma sequência de frames para se analisar.
   Escolher o Prim_ Frame e Ultimo_ frame. Nesta altura surgem no ecrã os nomes dos ficheiros de polos disponíveis na pasta do filme selecionado, afim de se perceberem quais os frames com polos marcados.
4. Escolher um ficheiro de polos e mius a admitir para a sequência de frames escolhida.
5. Visualizar o filme dos mapas conformacionais das imagens inscritas nos retângulos dos polos
6. Ler um ficheiro Excel de registo temporal dos frames do filme selecionado.

O input desta função é um vetor-linha de três entradas binárias. A primeira refere-se à visualização do filme, opção 2. A segunda refere-se à intenção de ler um ficheiro de polos, opção 4, do filme selecionado em 1, e a terceira à leitura do eixo temporal do filme, opção 6.

3.2 Marcação dos polos

Para adaptar uma forma super-elíptica aos fusos foi necessário implementar um algoritmo que permitisse estabelecer invariâncias para cada filme. Com este propósito foi criada a rotina MarcarPolos.m que permite visualizar os frames individualmente e marcar os polos no ecrã com o rato. Com esta rotina é possível também comparar polos marcados em frames diferentes. Além da marcação dos polos, ajusta-se individualmente para cada frame o valor do parâmetro da curvatura da elipse.

Equação da super elipse:

$$\left(\frac{x-x_0}{a}\right)^{\mu+1} + \left(\frac{y-y_0}{b}\right)^{\mu+1} = 1 \tag{3.1}$$

em que o parâmetro μ representa o grau de mapeamento cónico/elíptico.

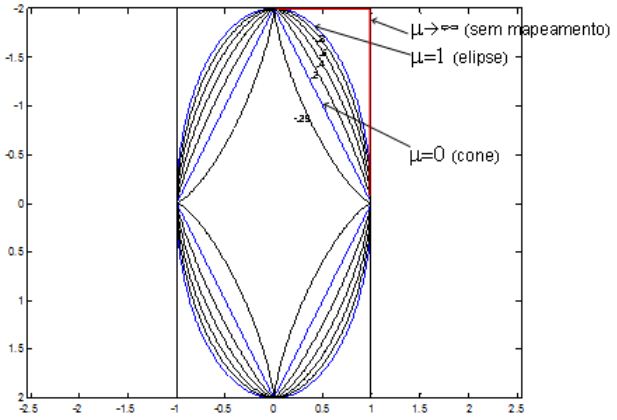

Figura 3.1: Ilustração de super elipses com diferentes valores de μ.

Para fazer a marcação das curvas da super-elipse (curva de Lamé) sobre o fuso, utilizamos as equações paramétricas da super-elipse uma vez que os fusos se encontram geralmente inclinados em relação aos eixos vertical e horizontal.

$$\begin{cases} x(\theta) = \mid cos(\theta) \mid^{2/(\mu+1)} \cdot a \cdot sign(cos(\theta)), \\ y(\theta) = \mid sen(\theta) \mid^{2/(\mu+1)} \cdot b \cdot sign(sen(\theta)). \end{cases} \tag{3.2}$$

Rotação de ângulo ϕ:

$$x = x(\theta) \cdot cos(\phi) - y \cdot sen(\phi)$$

$$y = y(\theta) \cdot cos(\phi) + x \cdot sen(\phi)$$

3.2.1 O programa

O algoritmo para marcação de polos foi implementado no script MarcarPolos.m
De seguida, apresenta-se um diagrama de fluxo do código elaborado para implementação do algoritmo.

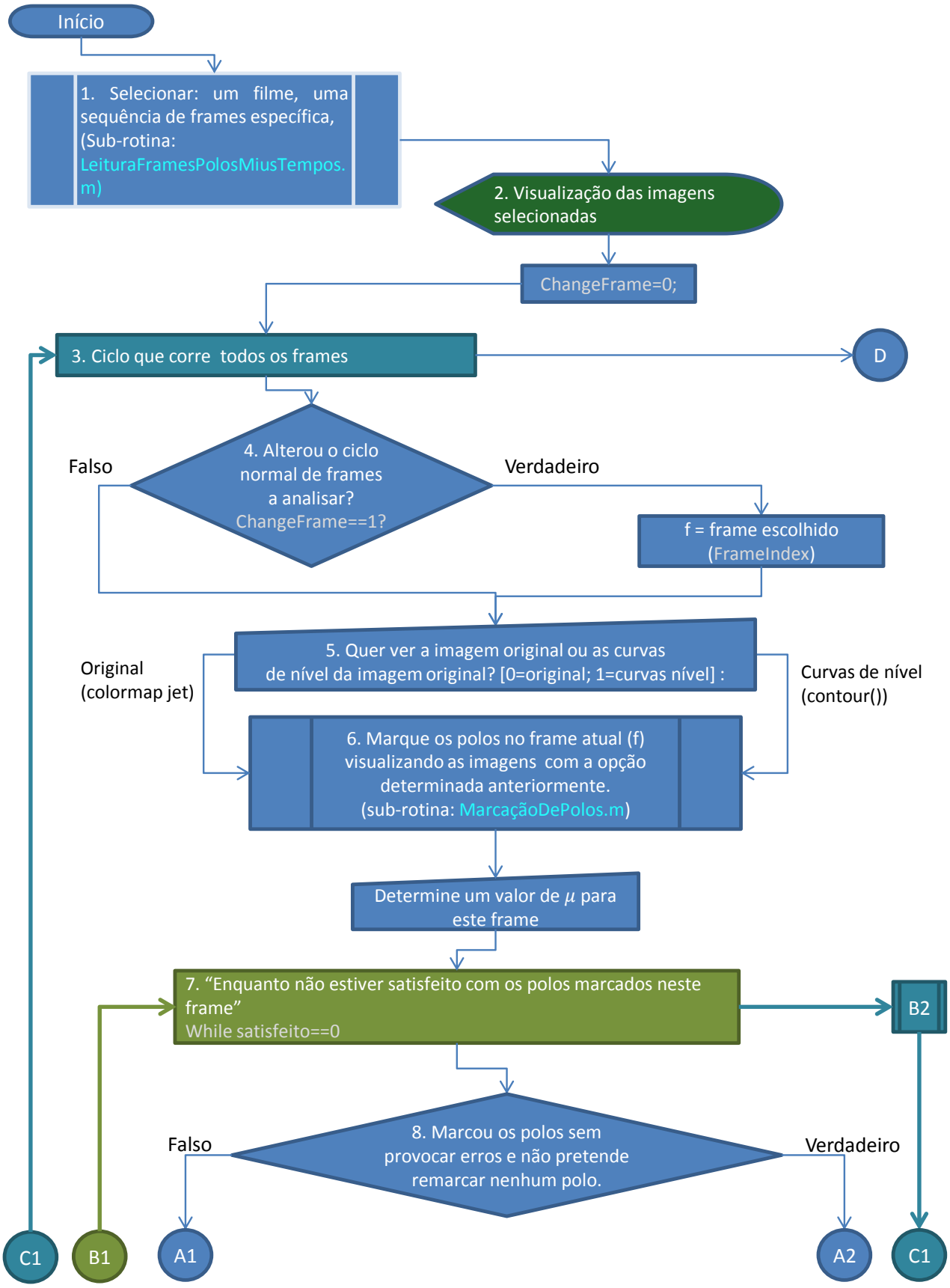

Figura 3.2: Diagrama de fluxo do programa MarcarPolos, parte 1.

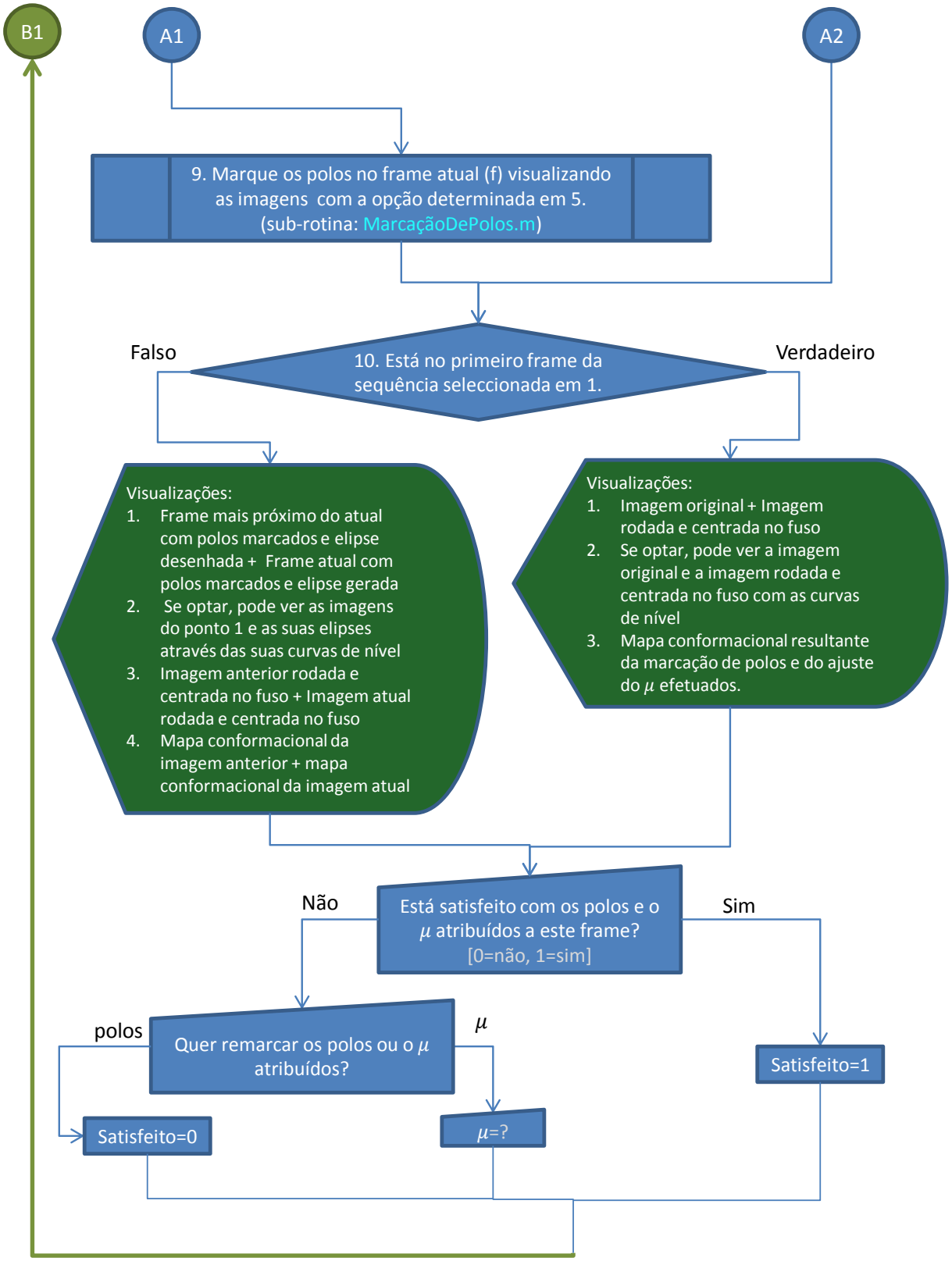

Figura 3.3: Diagrama de fluxo do programa MarcarPolos, parte 2.

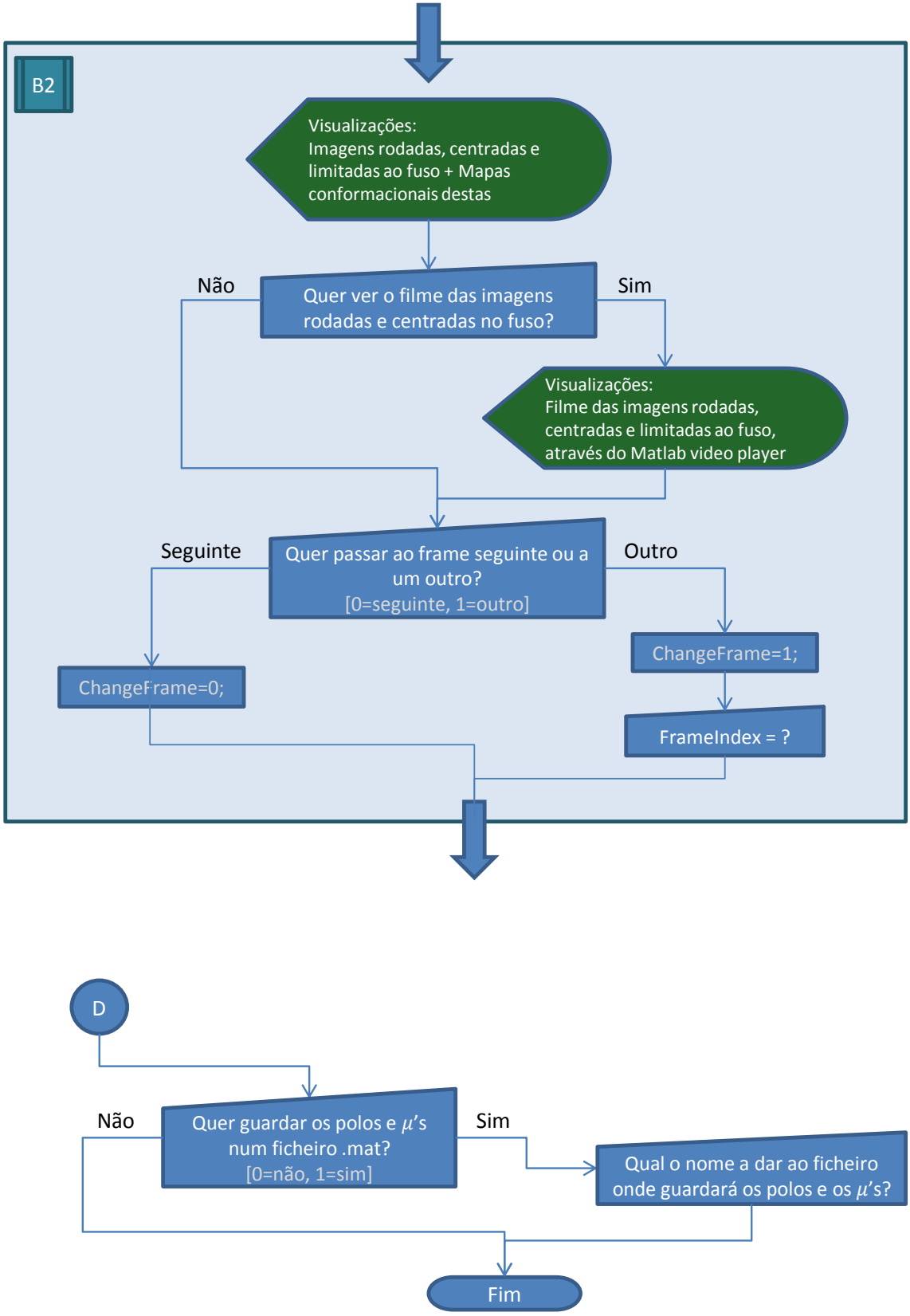

Figura 3.4: Diagrama de fluxo do programa MarcarPolos, parte 3.

3.3 Mapa conformacional super-elipse - linhas paralelas

O algoritmo:

Foi criada uma função no MATLAB para aplicar o mapa conformacional, a função ConfMapping.m. Esta função é aplicada às imagens rodadas e centradas nos fusos.
Os inputs desta função são:

- frame: Frame com o fuso em questão ao qual se quer aplicar o mapa conformacional
- polos: Coordenadas dos polos marcados no frame
- μ: Parâmetro μ atribuído a este frame

No início do algoritmo é efetuada uma rotação à imagem de entrada consoante o ângulo entre o vetor $\vec{P1P2}$ e o eixo horizontal (em que P1 e P2 são os polos ilustrados na figura 3.5, α o ângulo de rotação).

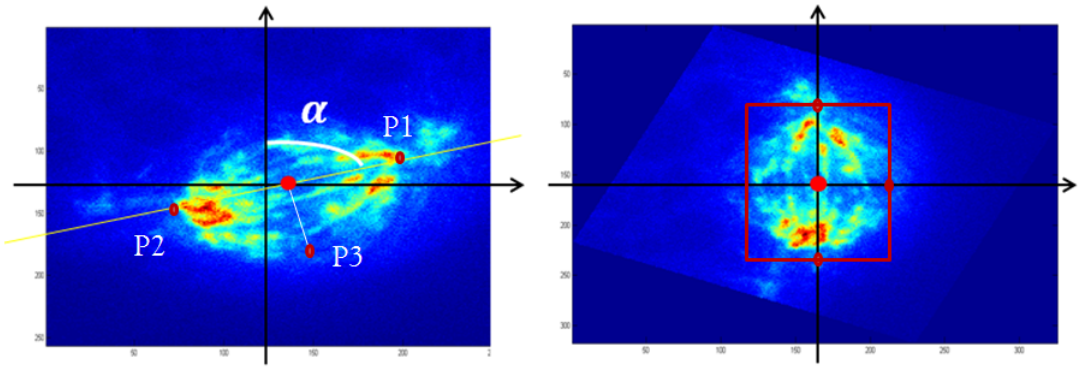

Figura 3.5: Rotação e translação das imagens para gerar as novas imagens.

Recorta-se de seguida a imagem pelo retângulo que inscreve a elipse subjacente aos pontos marcados. Ao fazer este recorte da imagem tem-se o cuidado de criar imagens com um número ímpar de linhas e colunas. Assim consegue-se que o centro da imagem incida precisamente no centro do fuso calculado através das coordenadas dos polos. Seja (a, b) o pixel central da imagem, $2a + 1$ o número de linhas e $2b + 1$ o número de colunas desta nova imagem.
A construção dos mapas foi simplificada com deslizamento horizontal dos pixeis sobre as elipses de forma a construir linhas paralelas verticais. Esta aproximação só não seria apropriada junto aos polos. No entanto essa região nunca é usado nas estimativas do fluxo.
De seguida entra-se num ciclo que percorre todas as linhas do hemisfério superior da imagem. Para cada par ordenado (x_1, y_1) da nova_ imagem "adapta-se" uma elipse centrada em $(a+1, b+1)$ que passe nesse ponto e tenha semi-eixo menor igual a b, como ilustrado na figura 3.6.

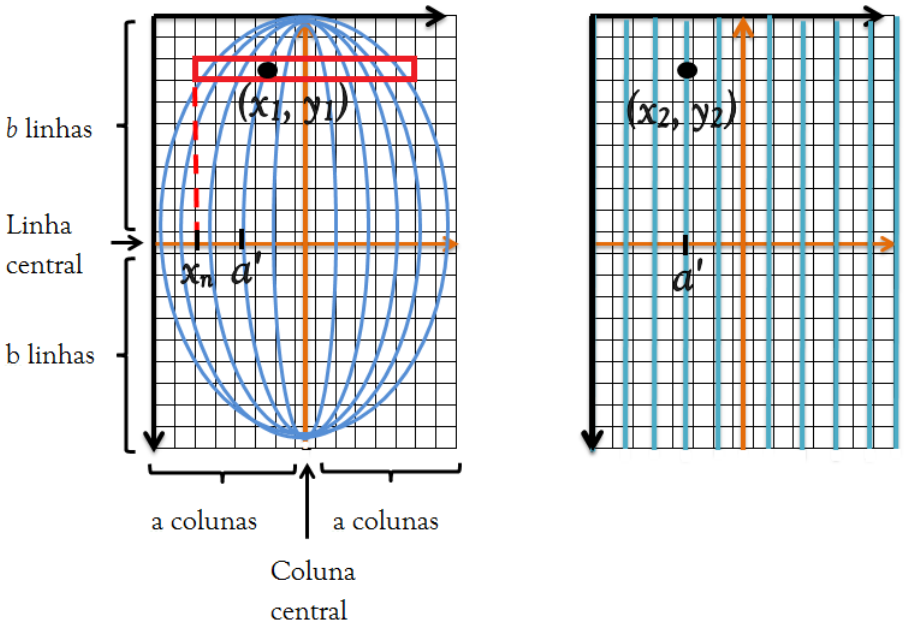

Figura 3.6: Ilustração do mapa conformacional elipse - linhas paralelas

.

Para cada linha n da imagem determina-se o ponto x_n.
Considerem-se os eixos laranja da figura 3.6. Nestes eixos, o ponto x_n é um ponto da elipse centrada em (0,0) com semi-eixo maior b e semi-eixo menor a, que tem ordenada na linha n. A ordenada do ponto x_n é obtida em função da linha n por $y = b + 1 - n$. A abcissa de x_n, nos eixos laranja, é obtida então através da equação da elipse e dos parâmetros já disponíveis:

$$\left(\frac{x}{a}\right)^{\mu+1} + \left(\frac{y}{b}\right)^{\mu+1} = 1 \Leftrightarrow$$

$$\Leftrightarrow x = \pm a \sqrt[\mu+1]{1 - \left(\frac{y}{b}\right)^{\mu+1}}$$

$$\Longrightarrow x = a \sqrt[\mu+1]{1 - \left(\frac{b+1-n}{b}\right)^{\mu+1}} + a$$

Consideram-se de seguida todos os pontos da linha n que estão no interior da elipse, ou seja, todos os pixeis do vetor $[a + 1 - x_n : a + 1 + x_n]$. Seja (x_1, y_1) um desses pontos. Existe uma elipse centrada em $(a + 1, b + 1)$ que passa no ponto (x_1, y_1) e tem semi-eixo maior igual a b, como ilustrado na figura 3.6.
É possível então determinar o semi-eixo menor dessa elipse, a', através da equação da super-elipse. Considerando de novo os eixos laranja:

$$\left(\frac{x}{a'}\right)^{\mu+1} + \left(\frac{y}{b}\right)^{\mu+1} = 1 \Leftrightarrow \left(\frac{x}{a'}\right) = \sqrt[\mu+1]{1 - \left(\frac{y}{b}\right)^{\mu+1}}$$

$$\Leftrightarrow a' = \pm \frac{x}{\sqrt[\mu+1]{1 - \left(\frac{y}{b}\right)^{\mu+1}}}$$

$$\Longrightarrow a' = \frac{x}{\sqrt[\mu+1]{1 - \left(\frac{b+1-n}{b}\right)^{\mu+1}}}$$

Nos eixos da imagem, obtém-se $a' = a + 1 - a'$.
A transformação de cada par ordenado (x_1, y_1) da linha n que esteja no interior da super-elipse é então dada por:

$$(x_1, y_1) \longrightarrow (a', y_1)$$

em que a' é o semi-eixo menor da elipse que passa por (x_1, y_1) e tem semi-eixo maior b.
Esta abordagem é aplicada a todas as linhas da imagem, produzindo então um mapa conformacional elíptico da imagem inicial.
Pela natureza discreta dos mapas de pixeis e pela área da elipse ser inferior à do retângulo, o resultado desta aplicação irá conter valores Nan (not a number). Para preencher os pixeis que ficam sem valores atribuídos, efetua-se interpolação nas linhas da imagem. A interpolação é efetuada através do comando interp1, com 'method'='spline' (interpolação linear cúbica).

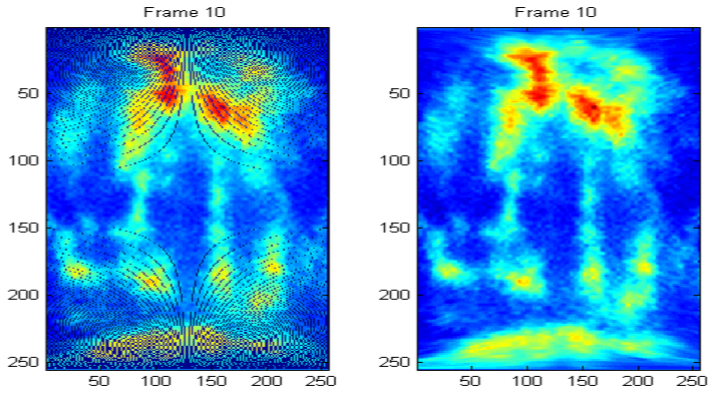

Figura 3.7: Fuso mitótico de célula de *Drosophila S2* após aplicação do mapa conformacional. Imagem anterior após interpolação.

Mapa conformacional linhas paralelas - super-elipse:

Para obter imagens dos mapas de velocidades com a conformação dos frames originais, criou-se a função ConfMapping_ inverso.m.

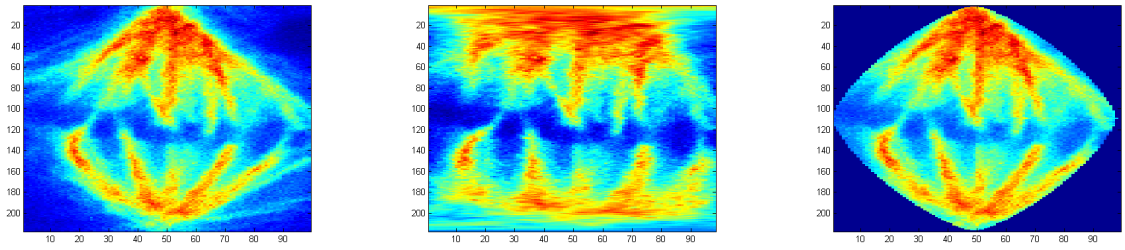

Figura 3.8: Imagens de fusos mitóticos manipuladas com as funções de mapa conformacional

Esta função é bastante simples, e os acertos de linhas e colunas funcionam da mesma forma da função anterior.
A cada pixel (x, y) faz corresponder a posição (x', y), em que x' é dado por:

$$\left(\frac{x'}{x}\right)^{\mu+1} + \left(\frac{y}{b}\right)^{\mu+1} = 1 \Leftrightarrow$$

$$\Leftrightarrow x' = \pm x \sqrt[\mu+1]{1 - \left(\frac{y}{b}\right)^{\mu+1}}$$

$$\Longrightarrow x = x \sqrt[\mu+1]{1 - \left(\frac{b+1-n}{b}\right)^{\mu+1}}.$$

Capítulo 4

Resultados

Afim de validar os resultados finais obtidos, efetuou-se um estudo paralelo em dados de controlo para testar os algoritmos implementados para o cálculo de fluxo e para o cálculo do decaimento. Neste estudo, criaram-se três *scripts* de controlo. Um que produz os dados teste para fluxo a uma dimensão, o *script* DadosTeste.m; outro que efetua o cálculo do fluxo sobre estes dados, CalcularFluxo_ Testes.m; e um terceiro que calcula o decaimento nos dados de teste, Decaimento-SemiAutomatico_ Testes.m.. Estes programas aplicam os mesmos algoritmos descritos no Capítulo 1, tendo apenas alterações nos preâmbulos, mas não nos ciclos que efetuam os cálculos principais. Para testar o fluxo a 2D foram utilizadas dez imagens do filme de células *Drosophila S2* no qual se marcaram individualmente três *speckles* artificiais com velocidades controladas.

4.1 Os dados de teste

Dados de teste para fluxo a 1D

O código para criar imagens artificiais encontra-se no *script* DadosTeste.m. Aqui foi implementado um algoritmo para construir pilhas artificiais de imagens totalmente controladas, em que nessas imagens existem *speckles* simulados.
As imagem da pilha têm dimensão $Dim \times Dim$, em que Dim é o número de linhas e colunas das imagens, determinado no início do *script*. Ao longo das colunas da imagem, as entradas são sempre iguais, ou seja, os *speckles* são representados como feixes horizontais ao longo do comprimento das imagens.
A construção destas imagens inicia-se com a indução de *speckles* no primeiro frame das pilha das imagens. Os *speckles* são construídos por convolução entre um vetor das amplitudes dos *speckles* e uma função gaussiana de parâmetro σ controlado. Uma função linear de velocidade em função das linhas das imagens determina a velocidade a que os *speckles* se deslocam ao longo da pilha de imagens. O movimento é induzido no sentido ascendente, no hemisfério superior das imagens, e no sentido descendente no hemisfério inferior, como representado na figura 4.1.

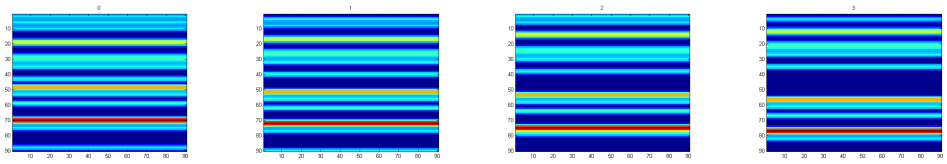

Figura 4.1: Exemplo de frames consecutivos de controlo criados com o programa DadosTeste.m

Os parâmetros que se podem determinar para esta pilha de imagens são:

- Dimensão das imagens (em pixeis); todas as imagens são, por defeito, quadradas.
- Dimensão das imagens (em μm)
- Número de frames na pilha
- Ruído das imagens
- Intervalo de tempo entre cada frame e respetivo eixo temporal da pilha
- Decaimento entre frames
- Número de *speckles* induzidos no primeiro frame da pilha
- Velocidades máxima e mínima a que os *speckles* se deslocam
- Amplitude de um *speckle* (número de linhas)

De seguida é apresentada uma explicação detalhada do *script* DadosTeste.m.

O código DadosTeste.m

1. Parâmetros do programa

Antes de correr o *script* verificamos se os parâmetros do programa são os que pretendemos simular.

```
% ------------------------------------------------------------------------
%% Parametros
Dim = 256;                % dimensao das imagens que vamos gerar
Nframes = 10;             % numero de frames que vamos gerar
Nspeckles = 30;           % numero de speckles no frame (inicial)
ruido = 0.1;
dt = 1.0;                 % intervalo de tempo entre cada frame [s]
L = 50;   % dimensao das imagens em microm.

vmax = 0.2;               % vel maxim de propagacao dos speckles (ondas)[um/s]
vmin = 0.1;               % vel min de propagacao das ondas [um/s]

vmax = vmax*Dim/(L*dt);   % [px/s]
vmin = vmin*Dim/(L*dt);   % velocidade de propagacao das ondas [px/s]

t = 0:dt:(Nframes-1)*dt;  % vetor de tempos [s]
decay = 1000.0;           % decaimento da diference entre extremos das ondas [s]

sigma = 3;                % escala espacial dos speckles (1 speckle deve ocupar
                          % 2*sigma pixeis em comprimento, aproximadamente)
% ------------------------------------------------------------------------
```

2. Variáveis e condições iniciais:

No frame inicial determinam-se aleatoriamente as linhas que vão conter os *speckles*. Têm-se $Nspeckles$ (= número de *speckles*) linhas com *speckles*. Estas linhas onde vão estar "centrados" os *speckles* são registadas na variável *Yspeckles*.

As intensidades dos *speckles* podem variar entre 0.8 e 1, sendo determinadas aleatoriamente neste intervalo. *Aspeckles* é o vetor onde são registadas as intensidades dos *speckles*.
Os *speckles* são ondas com amplitude aproximada de 2 × sigma pixeis de comprimento.

```
% ------------------------------------------------------------------------
%% Variaveis
Yspeckles = zeros(Nspeckles, Nframes);      % variavel das linhas em que
                                            % estao marcados os speckles
Aspeckles = zeros(Nspeckles, Nframes);      % intensidades dos speckles
                                            % situados nos Yspeckles

pilha = zeros( Dim, Dim, Nframes );         % conjuntos dos frames

%% Condições iniciais
% Posições para Nspeckles
Yspeckles(:,1) = randi(Dim,Nspeckles,1);
Aspeckles(:,1) = 0.8 + 0.2 * rand(Nspeckles,1);
% ------------------------------------------------------------------------
```

3. Produção de imagens:

É considerada uma função K (de Kernal) semelhante à função de distribuição gaussiana normalizada. Esta é produzida com amplitude aproximada de $2sigma + 1$ e normalizada. Cria-se uma variável nova, *template*, que consiste apenas na primeira coluna do primeiro frame (com os respetivos *speckles* e suas intensidades);
Aplica-se então convolução a esta variável template através da função K. Com o comando `repmat` replica-se o resultado da convolução para todas as colunas da imagem, resultando em colunas do frame todas iguais, e atingido o objetivo de reproduzir os *speckles* igualmente para todas as colunas.

```
% ------------------------------------------------------------------------
%% Algoritmo
Y = -50:50;
K = exp( -Y.*Y/(sigma^2));
K = K/sum(K);

template = zeros(Dim,1);
template( Yspeckles(:,1) ) = Aspeckles(:,1);
template = conv(template, K,'same');
pilha(:,:,1) = repmat(template,1,Dim);
% ------------------------------------------------------------------------
```

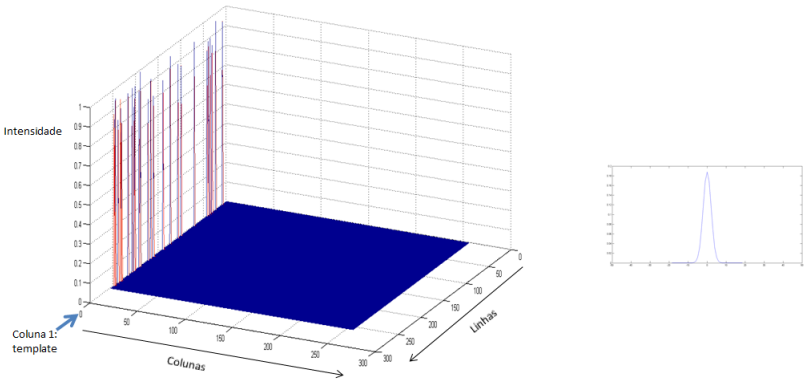

Figura 4.2: Vetor *template* obtido no MATLAB sobre a as posições da imagem. Função gaussiana ou função K do código. A função de convolução aplica a cada ponto do vetor template a função gaussiana ilustrada.

Função de convolução:

$$(K * v)(I) = \sum_i v(i)K(I - i) \tag{4.1}$$

onde v é o vetor *template*, K a função gaussiana e I uma entrada do vetor *template*, $I \in \{1, ..., Dim\}$.

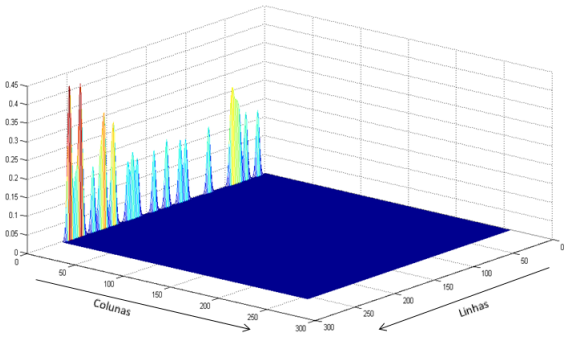

Figura 4.3: Ilustração obtida no Matlab do vetor *template* após convolução com a gaussiana K

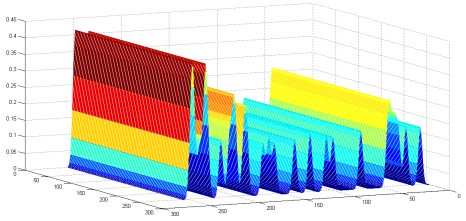

Figura 4.4: Ilustração obtida no Matlab da primeira matriz de uma pilha obtida na sequência das imagens anteriores.

5. Ciclo que reproduz as imagens da pilha, da 2ª à *Nframes*-ésima

Pretende-se que ao longo dos frames os *speckles* avancem com velocidades controladas mas não a velocidade constante ao longo das linhas. Para obter maior semelhança destes dados com os originais, manipulam-se os *speckles* do centro do fuso, ou seja, os que estão nas linhas centrais, para se deslocarem mais rápido, enquanto que os *speckles* que estão a atingir as primeiras ou as últimas linhas da imagem sejam os que se deslocam mais lentamente.
A equação escolhida para a velocidade, v, ao longo das linhas, y, foi uma equação quadrática ajustada à linha central $Dim/2$ das matrizes.

$$v(y) = m(y - Dim/2)^2 + b \tag{4.2}$$

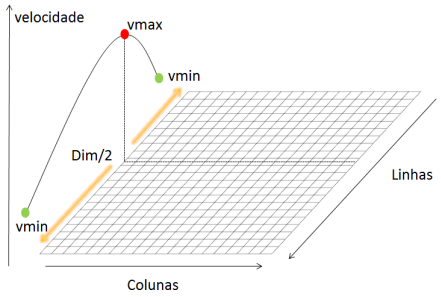

Figura 4.5: Ilustração das velocidades dos *speckles* ao longo das linhas.

Têm-se as seguintes condições para obter os parâmetros a e b da equação de cima:

- $v(\frac{Dim}{2}) = b = v_{max}$
- $v(0) = v_{min} = m\left(-\frac{Dmin}{2}\right)^2 + v_{max}$
  $\Longrightarrow m = \frac{v_{min} - v_{max}}{\left(\frac{Dim}{2}\right)^2}$

A equação final da velocidade, v, em função da linha y vem então:

$$v(y) = \frac{v_{min} - v_{max}}{\left(\frac{Dim}{2}\right)^2} \cdot \left(y - \frac{Dim}{2}\right)^2 + v_{max} \tag{4.3}$$

Como se pretende que os *speckles* da parte de cima da imagem (primeiras $Dim/2$ linhas) se desloquem para cima e os restantes se desloquem para baixo, ajustam-se as velocidades:

$$v_{sinal}(y) = v(y) \cdot sign(y - Dim/2) \tag{4.4}$$

```
% ------------------------------------------------------------------------
for f=2:Nframes
  v = (vmin - vmax)/(Dim^2/4) * ( Yspeckles(:,f-1) - Dim/2 ).^2 + vmax;
  v = v.* sign(Yspeckles(:,f-1) - 128);  % para termos velocidades positivas
    % e negativas
% ------------------------------------------------------------------------
```

Assim, de um frame para o seguinte sabe-se que um determinado *speckle* se desloca segundo a velocidade determinada, e pode-se determinar a sua posição em todos os frames.
A linha em que o *speckle* i se encontra no frame f, y_f^i, obtém-se da posição onde esse *speckle* se encontrava através de:

$$y_f^i = y_{f-1}^i + v(y_{f-1}^i) \cdot t$$

Para evitar que os *speckles* invertam o movimento quando a velocidade é negativa, faz-se um ajuste que determina a posição zero nesses casos, implicando depois que não são sejam considerados.

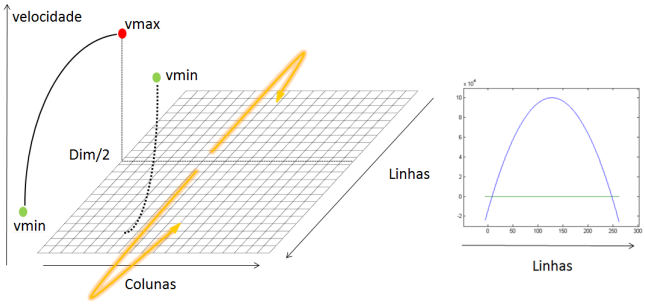

Figura 4.6: Ilustração das velocidades dos *speckles* ao longo das linhas.

```
% ------------------------------------------------------------------------
  Yspeckles(:,f) = Yspeckles(:,f-1) + v*dt ;
  temp = Yspeckles(:,f);
  Yspeckles( temp<1 | temp>Dim, f ) = 0;
% ------------------------------------------------------------------------
```

O vetor *Aspeckles(:,f)* é o vetor intensidades (amplitudes da onda) dos *speckles* no frame f. As intensidades dos *speckles* sofrem um decaimento exponencial ao longo do tempo à taxa decaimento fixada nos parâmetros iniciais pelo utilizador.
Cria-se a variável, *temp*, uma vez que vamos precisar das posições dos *speckles* em *Yspeckles* em inteiros positivos. Não se utiliza diretamente os *Yspeckles* arredondados para não acumular erros de arredondamento nos frames seguintes.
Assim, apenas consideramos os *speckles* com as devidas posições na convolução.

```
% ------------------------------------------------------------------------
  Aspeckles(:,f) = Aspeckles(:,f-1)*exp(-dt/decay);  % decaimento

  temp = round( Yspeckles(:,f) );
  index = find( temp>0);
  template = zeros(Dim,1);
  template( temp(index) ) = Aspeckles(index,f);
  template = conv(template, K,'same');
  pilha(:,:,f) = repmat(template,1,Dim);
  pilha(:,:,f) = pilha(:,:,f) + ruido*rand(Dim,Dim);
end
% ------------------------------------------------------------------------
```

4.1.1 Os dados teste para calculo de fluxo 2D

Os ensaios efetuados para avaliar o código elaborado quanto à performance a 2D foram testados em imagens de células *Drosophila S2* manipuladas com *speckles* artificiais.

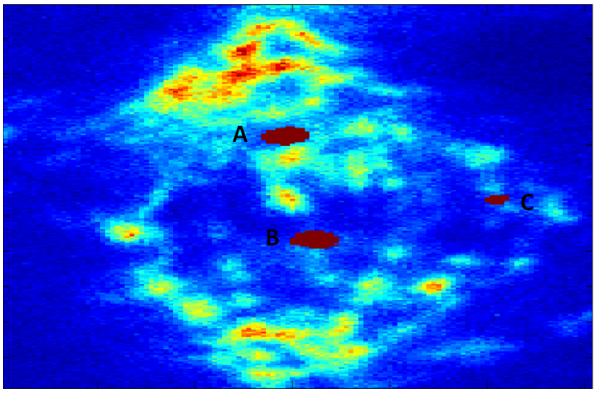

Figura 4.7: Imagem de *Drosophila S2* rodada e limitada ao fuso mitótico com 3 *speckles* artificiais.

Os 3 *speckles* foram marcados em 10 frames consecutivos do referido filme. O *speckle* A desloca-se ao longo das dez imagens a uma velocidade (-1,0) $px/frame$, ie, desloca-se para a esquerda 1px a cada dois frames consecutivos.
A mancha B desloca-se para baixo a uma velocidade média de (0,-1) $px/frame$, ou seja, desloca-se para baixo. Contudo, em alguns frames tem um desvio no eixo dos de um ou dois pixeis.
O terceiro *speckle* desloca-se a uma velocidade de, aproximadamente, $(-1,-1)$ $px/frame$, havendo também variações em alguns intervalos de frames.

4.2 Resultados nos dados de teste

4.2.1 Cálculo do fluxo - Dados artificiais 1D

O parâmetro σ:

Nas imagens originais, os spekcles têm diâmetros variados, e quase sempre superiores a 10px. Contudo, em média nos filmes de melhor qualidade do nosso conjunto de dados, os *speckles* têm cerca de 10 a 15 pixeis de diâmetro.
Nos testes de controlo realizados,admitimos diâmetro de *speckles* cerca de 5 px, uma vez que as imagens de teste são inferiores às imagens originais, para facilitar o processamento dos testes. As imagens de controlo geradas têm dimensão de 90x90 px.

A altura das janelas de trabalho:

Em função do diâmetro dos *speckles* artificiais, as janelas/blocos utilizados no cálculo das velocidades tem dimensão 13x10px. Isto porque a janela deve ter cerca de três vezes a altura dos *speckles*. Contudo, nos dados originais não "se pode" utilizar janelas que satisfaçam esta condição, pois janelas tão grandes restringem demasiado a área passível de ser analisada pelos blocos.
Assim, nos dados originais utilizaram-se janelas de dimensão 18x14px, pois foi a dimensão mais adequada que encontramos que não implicasse restringir demasiado a área de cálculo da velocidade, nem ter de baixar à velocidade máxima, *vmax*, no input do algoritmo.
Atendendo às restrições aos parâmetros a utilizar nos dados originais, tentámos simular situações semelhantes nos dados de teste. A altura admitida nos dados de controlo é 13px, enquanto que a largura é 10px.
Afim de comparar os quatro algoritmos codificados, estes foram aplicados a todos os filmes artificiais produzidos. Os algoritmos são:

1. Função de Correlação generalizada; Janelas colapsadas com o máximo.

2. Função de Correlação generalizada; Janelas colapsadas com a mediana.

3. Função das Diferenças Absolutas à Média; Janelas colapsadas com o máximo.

4. Função das Diferenças Absolutas à Média; Janelas colapsadas com a mediana.

Os testes que foram efetuados focaram-se principalmente no desempenho dos algoritmos em relação a três fatores dos filmes de teste:

- Ruído

- Relação entre a altura dos blocos de trabalho e a dimensão dos *speckles* artificiais

- Velocidade mínimas e máximas diferentes

De seguida são apresentados vários resultados que permitiram selecionar o algoritmo com melhor performance e eficácia no cálculo do fluxo. O número de frames utilizados nos testes é normalmente 5 para ser possível medir velocidade em regiões menos centrais das imagens.
O primeiro caso é o teste de controlo em que não existe ruído. Os *speckles* movem-se a velocidade controlada, sendo a velocidade máxima de $4.2px/s$ e a velocidade mínima $0.1px/s$.

Caso 1:

Neste caso a pilha de testes produzida consiste de 5 imagens. Os parâmetros utilizados foram os da lista seguinte.

Parâmetros do filme artificial:
• Dim = 120
• Nframes = 5
• Nspeckles = 30
• **ruído = 0**
• dt = 1.0
• t = (0:dt:(Nframes-1)*dt)'
• dv = 0.1
• **vmax = 4.2 px/s**
• **vmin = 0.1 px/s**
• decay = 1000
• sigma = 2

Parâmetros do cálculo do fluxo 1D:
• **Altura = 13**
• Largura = 10
• Não excluir frames, testando a sequência de frames [1,2,3,4,5]
• Vmax=5
• Dv=0.1

Uma vez que se pretende com este ensaio verificar a sensibilidade dos algoritmos para velocidades distintas no equador e na periferia das imagens, e que a velocidade máxima admitida é de 4.2px/s, foi necessário considerar uma dimensão relativamente grande das imagens, neste caso 120x120px. As velocidades induzidas na pilha estão mapeadas na figura 4.8.

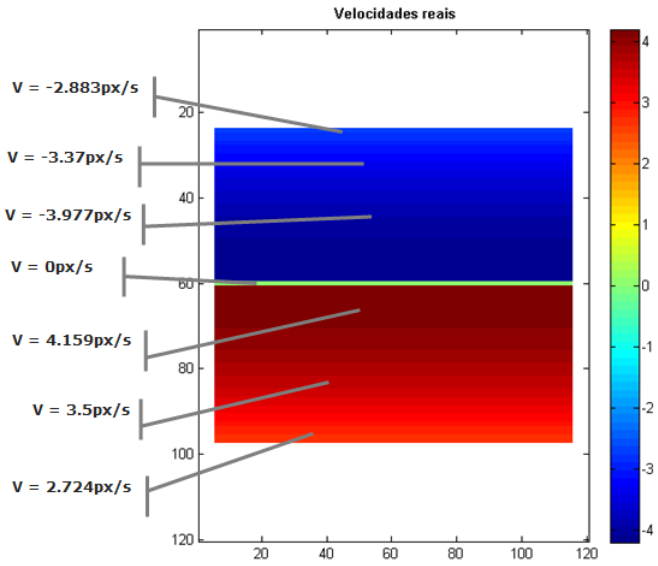

Figura 4.8: Mapa de velocidades reais (px/s) induzidas no filme do caso 1. Alguns valores de velocidades destacados para melhor interpretação do gráfico.

A função de velocidades neste caso é dada por $v(y) = -0.0014 \cdot (y - 60)^2 + 4.2$, onde $y \in \{1, 2, 3, ..., 120\}$. A figura 4.9 apresenta o perfil de intensidades da pilha considerada neste caso ao longo das colunas dos frames. Uma vez que não foi induzido nenhum ruído nas imagens, todas as colunas de cada imagem são iguais, e portanto o perfil ao longo de uma coluna traduz o perfil de intensidades para toda a pilha.
Na figura o eixo dos xx representa o eixo das linhas da pilha, enquanto que o eixo dos yy as intensidades. Este eixo está subdividido em pequenos eixos, um para cada frame.
Nesta imagem é possível detetar facilmente o movimento das ondas de intensidade em direção à periferia do fuso.

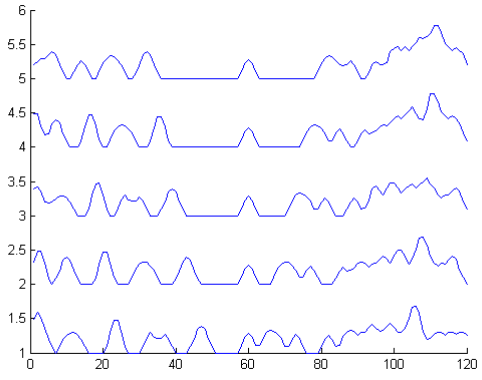

Figura 4.9: Perfil de intensidades da pilha produzida para o teste de controlo.

As pequenas variações do sinal de frame-para-frame devem-se ao facto deste sinal ser produzido através de uma operação que considera os índices das linhas dos pixeis.
Os resultados obtidos com os diferentes algoritmos são os apresentados nas figuras seguintes. Como não foi induzido ruído, os algoritmos 1 e 2 e os algoritmos 3 e 4 produzem o mesmo resultado.

Uma vez que as colunas são todas iguais, em cada linha o máximo ou a mediana das entradas é o valor de qualquer entrada, pois são todas idênticas.

1. Função de Correlação generalizada; Janelas colapsadas com o máximo.

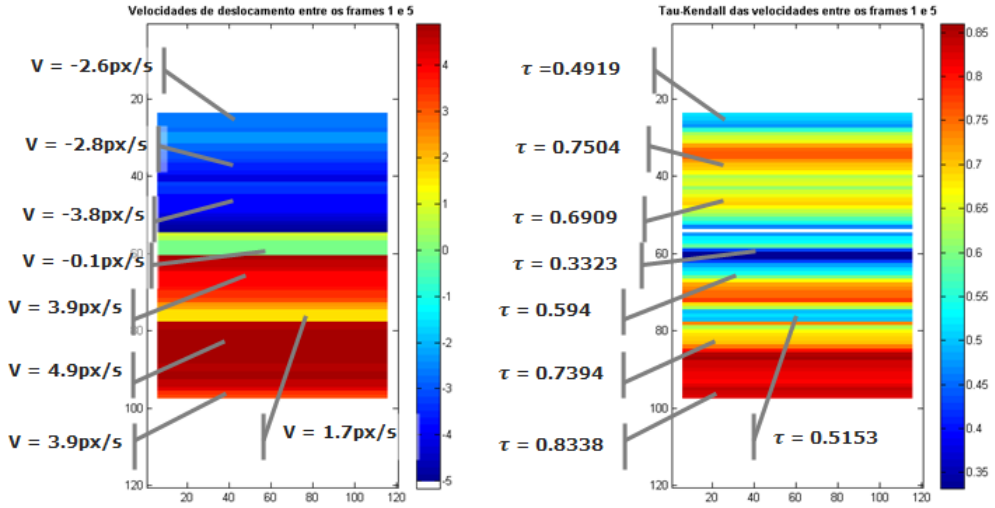

Figura 4.10: Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo da correlação generalizada, colapsando as janelas com o máximo.

3. Função das diferenças absolutas à média; Janelas colapsadas com o máximo.

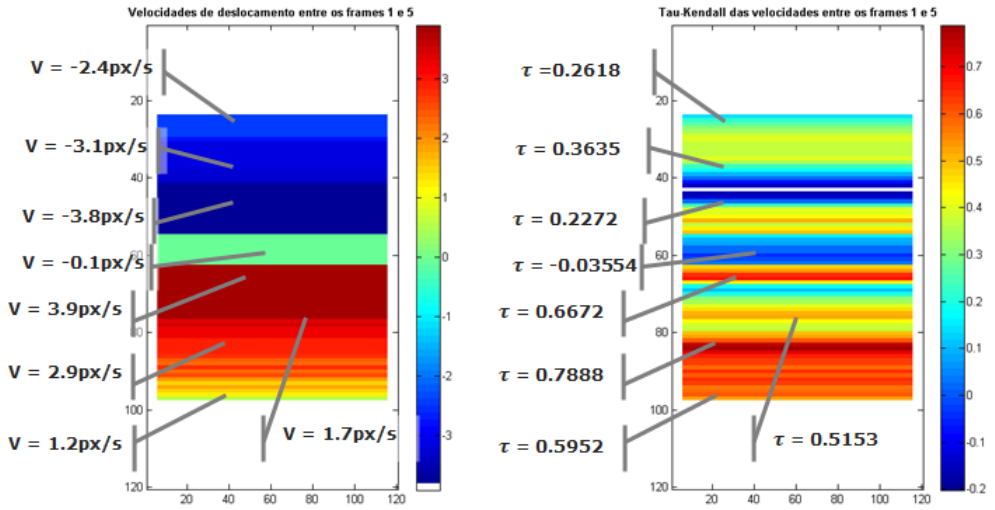

Figura 4.11: Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo das diferenças absolutas à média, colapsando as janelas com o máximo.

Os valores de velocidades realçados nos mapas de velocidades são referentes aos mesmos pixeis realçados no mapa de velocidades da figura 4.8.
Ambos os algoritmos tiveram um bom desempenho no hemisfério superior das pilhas, contudo o algoritmo da função de diferenças absolutas em relação à média aproximou-se mais das veloci-

dades reais. Também no hemisfério sul as velocidades determinadas por este algoritmo são mais aproximadas das velocidades reais, e para além disso, com o algoritmo da correlação adaptada verificou-se inversão da derivada da velocidade.
Para melhor interpretação dos mapas apresenta-se um gráfico 2D das velocidades dos mapas ao longo das linhas na figura 4.12.

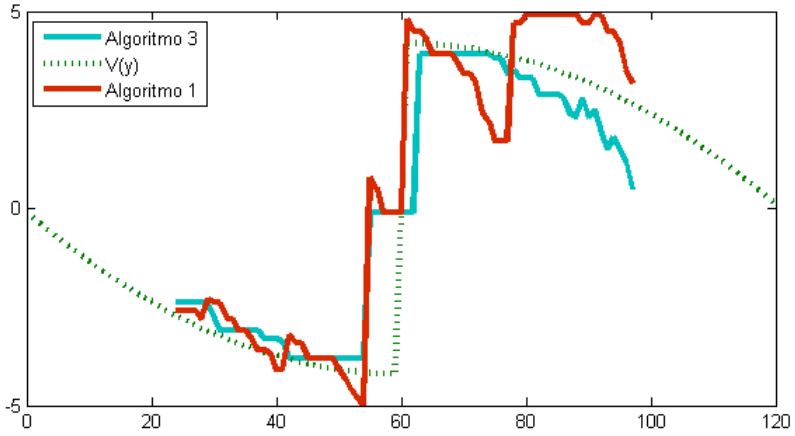

Figura 4.12: Comparação dos resultados obtidos para o teste de controlo do caso 1. Gráfico das velocidades (px/s) obtidas em função das linhas para os dois métodos considerados e velocidades (px/s) atribuídas a cada pixel, no caso 1.

Através dos gráficos anteriores compreende-se que a função de correlação generalizada adaptada oscila mais relativamente às velocidades reais do que a função de diferenças absolutas à média. Contudo, na proximidade da periferia, ou seja, fora do centro da imagem, a função de correlação começa a ter melhor desempenho, uma vez que há maior quantidade de picos (*speckles*) nestas regiões.
Foram efetuados vários testes para perceber o desempenho da função de correlação e da função de diferenças absolutas à média, consoante o tipo de perfil de intensidades dos frames.

Caso 2:

O teste seguinte exemplifica o desempenho dos algoritmos face a uma relação muito diferente entre a altura das janelas, Altura, e o diâmetro dos *speckles*, $2\sigma + 1$. Neste teste provocámos que as regiões a considerar não tivessem liberdade, e abrangessem praticamente apenas o diâmetro dos *speckles*.
Os parâmetros utilizados no teste foram os seguintes:

Parâmetros do filme artificial:
- Dim = 90
- Nframes = 5
- Nspeckles = 30
- **ruído = 0.5**
- dt = 1.0
- t = (0:dt:(Nframes-1)*dt)'
- dv = 0.1
- **vmax = 2.7**
- **vmin = 2.7**
- decay = 1000
- **sigma = 5**

Parâmetros do cálculo do fluxo 1D:
- **Altura = 6**
- Largura = 10
- Não excluir frames, testando a sequência de frames [1,2,3,4,5]
- Vmax=5
- Dv=0.1

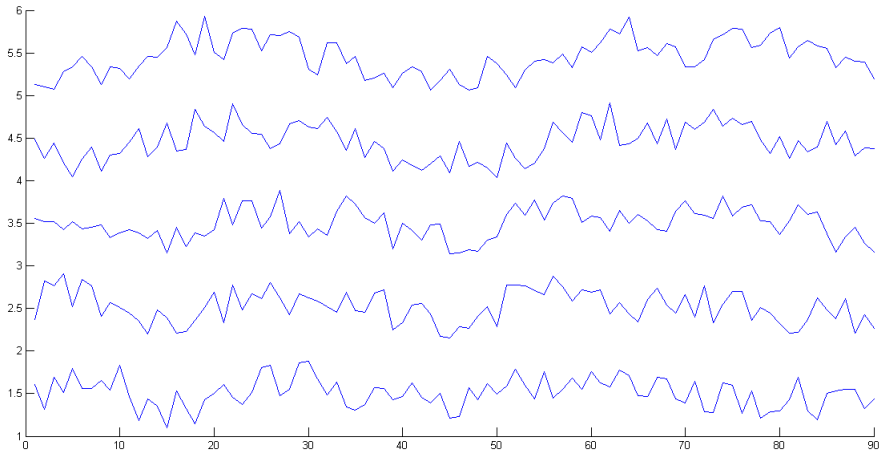

Figura 4.13: Perfil de intensidades da pilha produzida para caso 2. Relação sinal/ruído = 0.5, $\sigma = 5$.

Agora com a amplitude do ruído a aproximar-se da ordem de grandeza da amplitude do sinal, começa a ser difícil a perceção do fluxo ao longo dos gráficos do perfil de intensidades para as colunas (Figura 4.13).
Note-se que na região do equador a intensidade do sinal é ligeiramente mais baixa do que as intensidades em seu redor (contando com ruído). Também na região da linha 25 e da linha 65 se verifica uma ligeira redução da intensidade. Este aspeto irá ter influência nos resultados seguintes.

1. Função de correlação generalizada; Janelas colapsadas com o máximo.

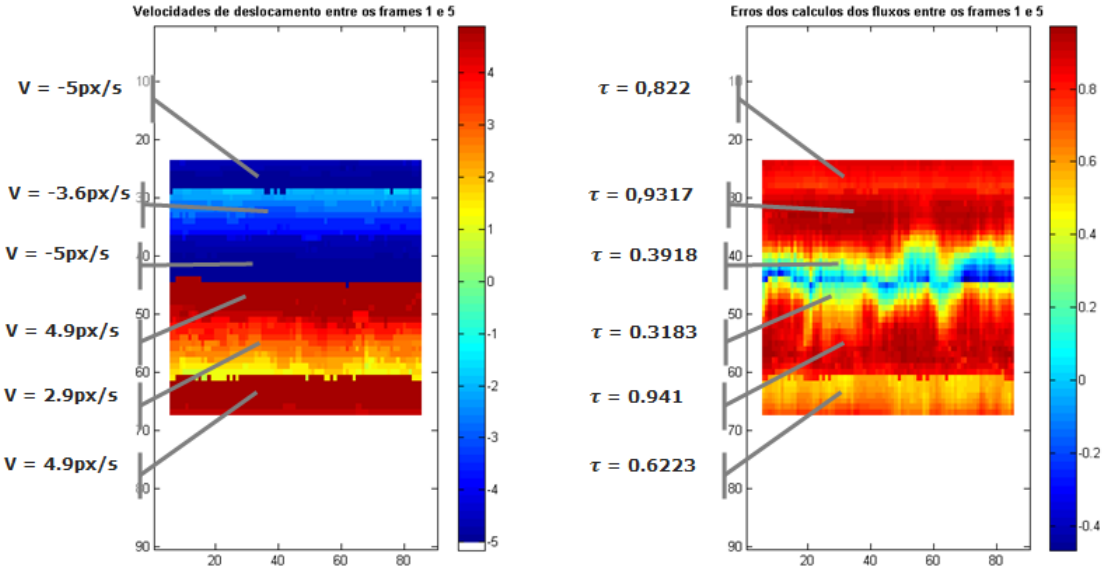

Figura 4.14: Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo da correlação generalizada, colapsando as janelas com o máximo. Ruído = 0.5, $\sigma = 5$, Altura = 6.

2. Função de correlação generalizada; Janelas colapsadas com a mediana.

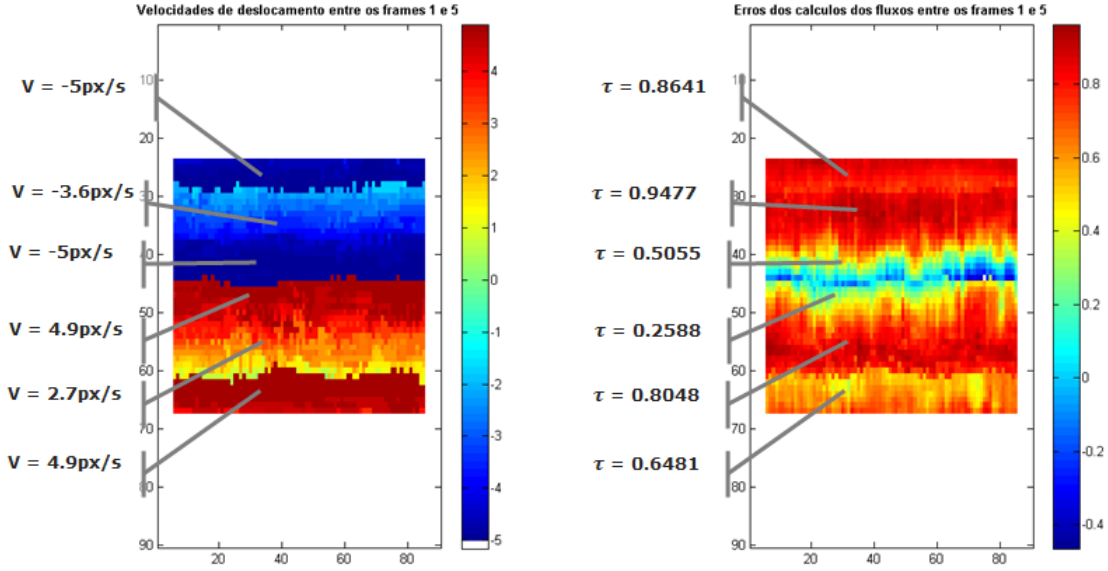

Figura 4.15: Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo da correlação generalizada, colapsando as janelas com a mediana. Ruído = 0.5, $\sigma = 5$, Altura = 6.

3. Função das diferenças absolutas à média; Janelas colapsadas com o máximo.

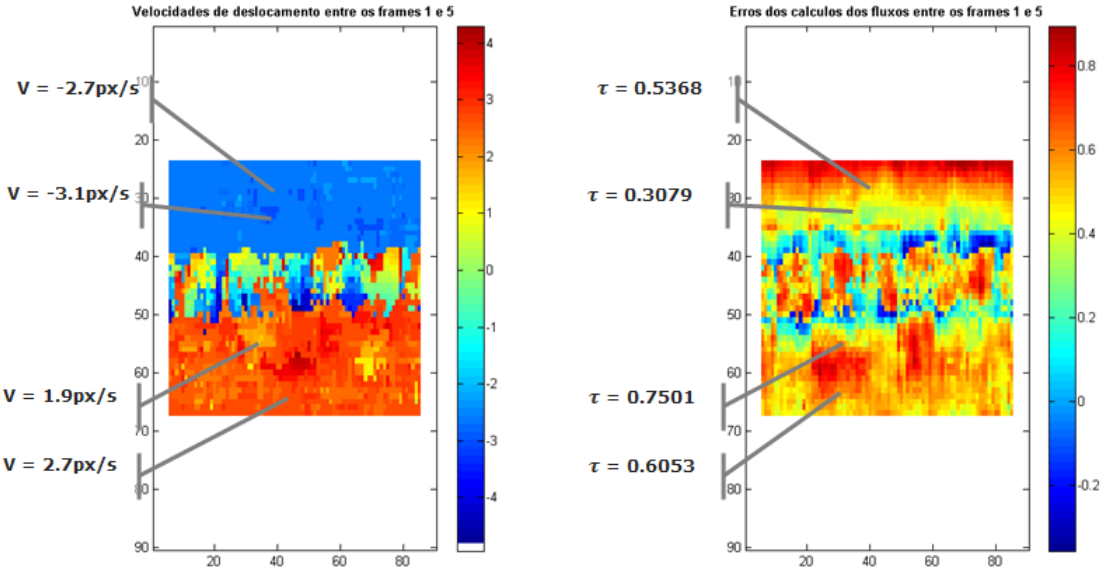

Figura 4.16: Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo das diferenças absolutas à média, colapsando as janelas com o máximo. Ruído = 0.5, $\sigma = 5$, Altura = 6.

4. Função das diferenças absolutas à média; Janelas colapsadas com a mediana.

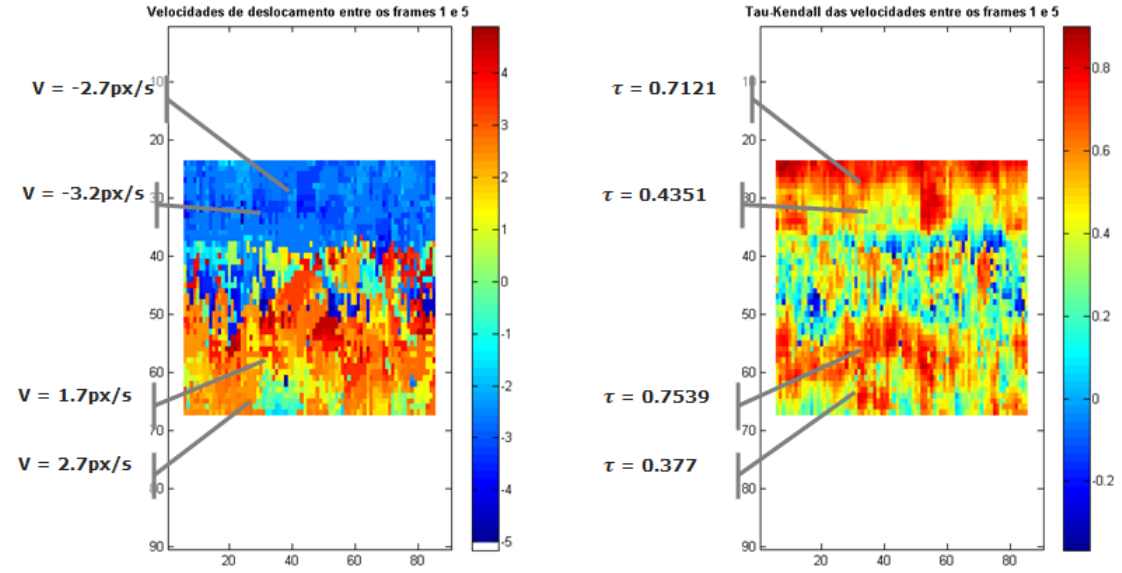

Figura 4.17: Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo das diferenças absolutas à média, colapsando as janelas com a mediana. Ruído = 0.5, $\sigma = 5$, Altura = 6.

Com o algoritmo de correlação generalizada adaptada, quer para os resultados apresentados nas figuras 4.14 e 4.15, quer para outros resultados realizados não registados aqui, as velocidades determinadas atingem, em vários pixeis, os valores limites de vmax.
Nas figuras referidas, várias zonas dos mapas de velocidades apresentam velocidades de -5px/s e 5px/s. Tal ocorre por estas zonas serem zonas de baixa intensidade e por o referido algoritmo

promover a velocidade que obtém um produto maior entre os blocos admitidos. Nestes casos, para maximizar o produto, a zona mais baixa tem de ser multiplicada por uma zona mais alta (as regiões de *speckles*). Por isso, nas regiões de intensidades mais baixas este algoritmo produziu velocidades mais discrepantes. (Por exemplo, linha 26 ou linhas centrais).

O algoritmo das diferenças absolutas à média produziu resultados mais satisfatórios. No mapa produzido com o Algoritmo 3, figura 4.16, a maioria das velocidades são condizentes com o presuposto, principalmente no hemisfério superior da pilha. Neste hemisfério a maior parte dos valores obtidos foi -2.7px/s, e na parte inferior 2.7px/s. Quer no mapa da figura 4.16, quer no mapa da figura 4.17, na zona do equador há bastantes oscilações. Estas devem-se ao facto de nesta região haver *speckles* a deslocarem-se em regiões opostas, como está ilustrado na figura seguinte.

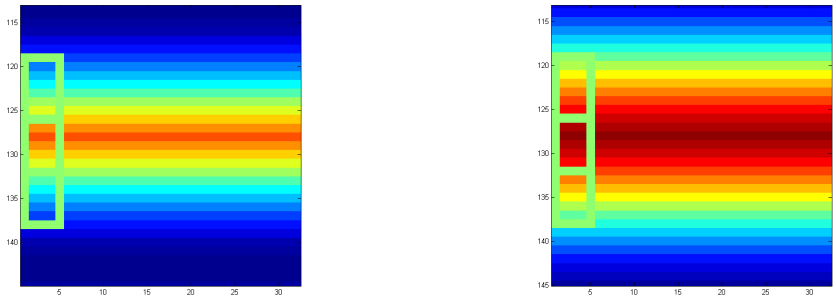

Figura 4.18: Pormenor do equador de uma pilha de teste. Região central de uma pilha de teste em dois frames consecutivos. Janelas de trabalho evidenciadas.

Na figura 4.18 a janela de trabalho incide numa região central de uma pilha. No primeiro frame há um *speckle* no centro da janela, e no frame seguinte há dois *speckles* a deslocarem-se em sentidos opostos. Enquanto que o algoritmo da correlação generalizada adaptada procura os pontos de intensidade mais alta para obter um produto mais elevado, o algoritmo DAM procura as regiões mais semelhantes. Assim, nas regiões centrais é usual os resultados com ambos os métodos estarem bastante enviesados, pois os blocos encaixam com os dois *speckles*, o que se desloca para cima e o que se desloca para baixo. Por isso os mapas de velocidades apresentam uma mistura de cores muito heterogénea na região do equador.

Vários testes foram efetuados para avaliar o desempenho dos algoritmos quando em presença de ruído. Tal como nos casos anteriores, o algoritmo DAM utilizando o máximo para colapsar as janelas de trabalho foi o mais bem sucedido em todos os casos.

Caso 3:

Afim de avaliar os algoritmos numa situação controlada que fosse o mais possivel próxima da realidade das imagens disponíveis, efetuou-se um teste sobre uma pilha de imagens que se produziu especialmente para se assemelharem aos dados reais.
O gráfico da figura 4.19 trata-se do plot da 60ª coluna do oitavo frame de um filme de células *Drosophila S2*. O gráfico da direita representa os mesmos dados, mas apenas nas primeiras 100 linhas.

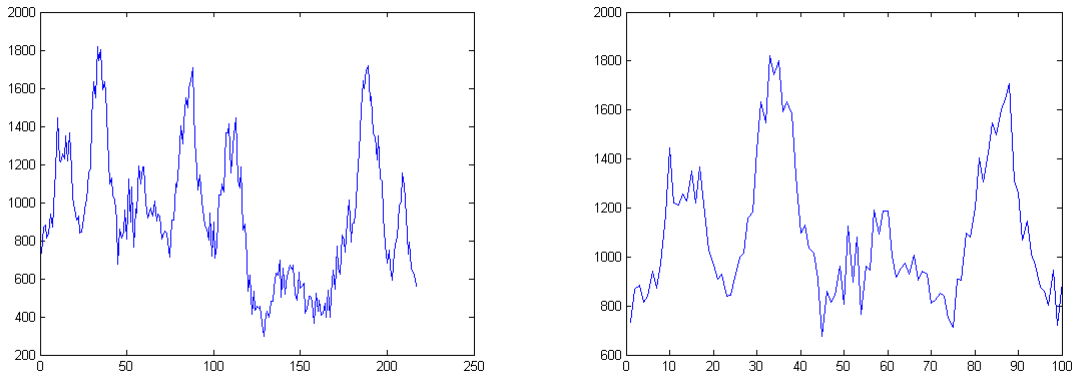

Figura 4.19: Perfil de intensidades de uma coluna de imagem *Drosophila S2*.

Parâmetros que produzem pilhas de teste com perfis de intensidades semelhantes ao da figura 4.19 são, por exemplo, os parâmetros do filme artificial listados de seguida. Os parâmetros para correr o programa de cálculo de fluxo que retornaram os resultados seguintes foram os da direita.

Parâmetros do filme artificial:
- Dim = 100
- Nframes = 4
- Nspeckles = 8
- **ruído = 0.3**
- dt = 1.0
- t = (0:dt:(Nframes-1)*dt)'
- dv = 0.1
- **vmax = 2.7**
- **vmin = 2.7**
- decay = 1000
- **sigma = 7**

Parâmetros do cálculo do fluxo 1D:
- **Altura = 18**
- Largura = 14
- Não excluir frames, testando a sequência de frames [1,2,3,4,5]
- Vmax=3.8
- Dv=0.1

É de notar que neste teste os parâmetros Altura, Largura e vmax são os mesmos que serão utilizados no cálculo do fluxo a 1D nas imagens originais, pois os testes com estes parâmetros serviram para aproximar o mais possível uma situação controlada da situação real de análise.
Neste caso a amplitude dos *speckles* não varia entre 0 e 1 mas entre 5000±300, antes de ser aplicada a convolução. No código, para implementar o teste altera-se da seguinte forma:

```
% ------------------------------------------------------------------------
Aspeckles(:,1) = 10000 + 300 * rand(Nspeckles,1);
.
.
pilha(:,:,1) = repmat(template,1,Dim) + ruido*rand(Dim,Dim)* max(template);
.
.
pilha(:,:,f) = pilha(:,:,f) + ruido*rand(Dim,Dim)*max(max(pilha(:,:,f)));
% ------------------------------------------------------------------------
```

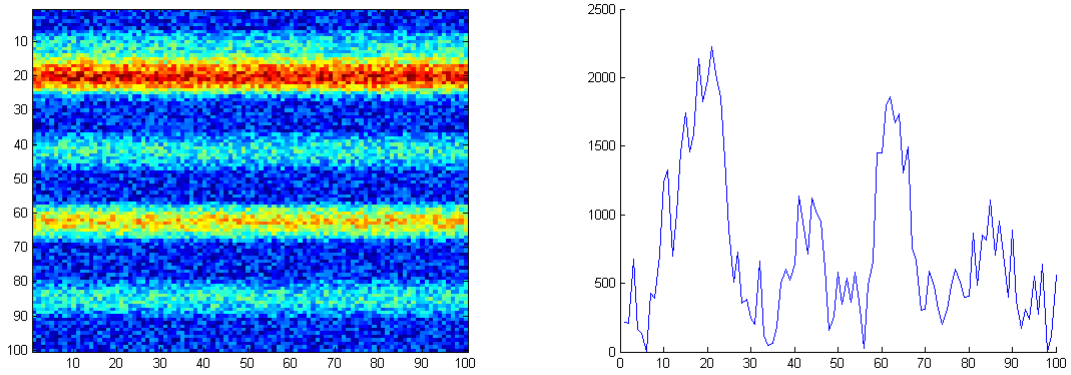

Figura 4.20: Imagens de teste para o caso 3. Primeiro frame da pilha de teste utilizada (esquerda). Perfil da primeira coluna do primeiro frame da pilha utilizada.

Comparando o perfil de intensidades das imagens artificiais com o perfil de intensidades da imagem de *Drosophila S2* as semelhanças são notórias. Neste teste apenas aplicámos o método 1, Correlação Generalizada utilizando o máximo, e o método 3, DAM utilizando o máximo.

1. Função de correlação generalizada; Janelas colapsadas com o máximo.

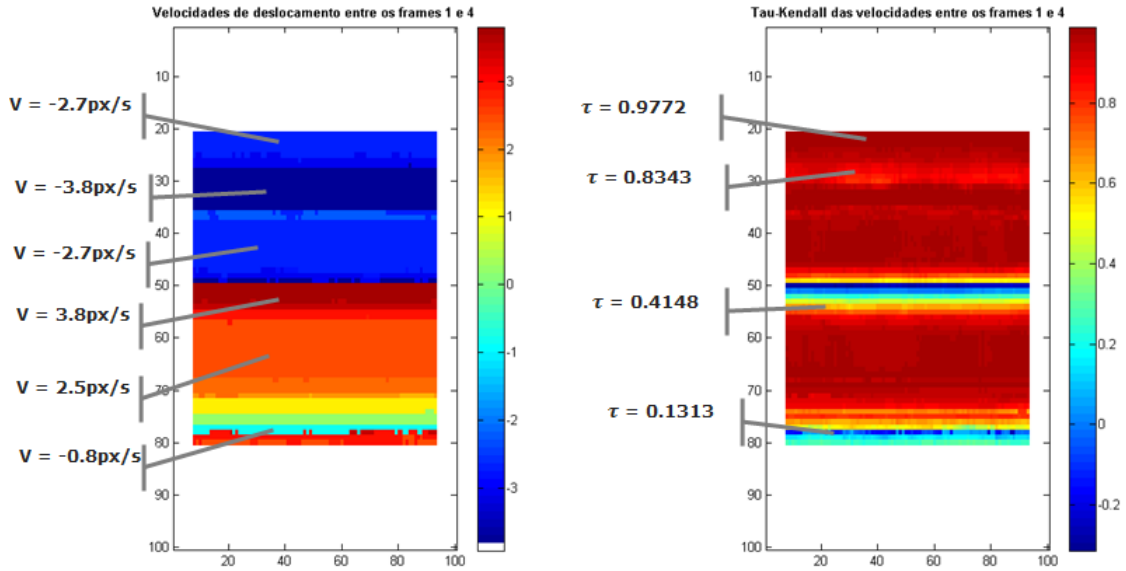

Figura 4.21: Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo da correlação generalizada adptada, colapsando as janelas com o máximo. Ruído = 0.2, $\sigma = 5$, Altura = 18.

3. Função das diferenças absolutas à média; Janelas colapsadas com o máximo.

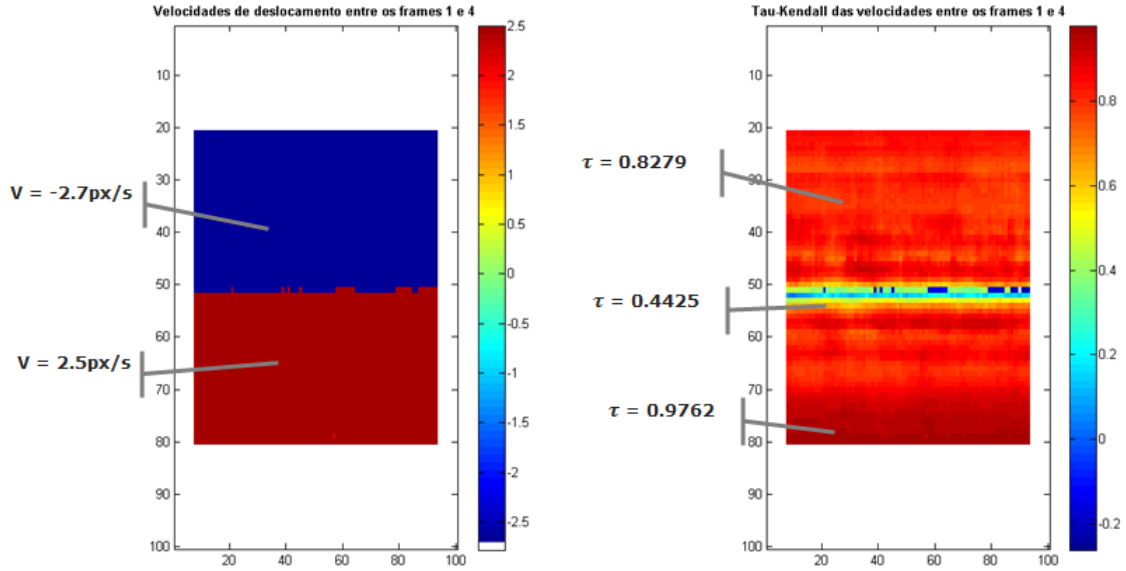

Figura 4.22: Velocidades (px/s) e valores de τ-Kendall obtidos com o algoritmo DAM, colapsando as janelas com o máximo. Ruído = 0.2, $\sigma = 5$, Altura = 18.

Os resultados obtidos neste teste foram bastante satisfatórios. Mais uma vez o algoritmo DAM teve um melhor desempenho que o algoritmo da correlação generalizada adaptada. Apesar das velocidades obtidas com o Algoritmo 1 não serem tão aproximadas da velocidade induzida, o mapa de velocidades deste algoritmo apresenta valores de τ-Kendal superiores ao do Algoritmo 2. De facto, esta tendência dos valores de τ-Kendall verifica-se na maioria dos testes. Isto acontece porque o algoritmo da correlação utiliza o produtório das intensidades enquanto que o DAM utiliza as diferenças.
Nos vários testes efetuados nas condições deste caso, o mapa de velocidades do algoritmo 1 apresenta bandas como as que se observam na figura 4.21. Estas bandas devem-se, uma vez mais, à procura que este algoritmo efetua do produto de vetores mais elevado, custando-lhe isso a credibilidade nas velocidades calculadas.
As velocidades representadas na figura 4.22 são bastante próximas das reais induzidas. Embora a velocidade no hemisfério inferior não seja 2.7px/s, a homogeneidade neste hemisfério e os bons valores de τ-Kendall tornam o resultado muito satisfatório.
Assim, o método elegido para calcular o fluxo nos dados orginais foi o método DAM, e a função escolhida para colapsar as janelas de trabalho foi o máximo.

4.2.2 Cálculo do fluxo - Dados teste 2D

Para os dados de teste 2D foram marcados 3 *speckles* artificiais em 10 frames consecutivos do filme *Drosophila S2*. O *speckle* A desloca-se ao longo das dez imagens a uma velocidade (-1,0) $px/frame$, ie, desloca-se para a esquerda $1px$ a cada dois frames consecutivos. O *speckle* artificial B desloca-se para baixo a uma velocidade média de (0,-1) $px/frame$, ou seja, desloca-se para baixo. Contudo, em alguns frames tem um desvio no eixo dos xx de um ou dois pixeis.
O terceiro *speckle* desloca-se a uma velocidade média de (-1,-1) $px/frame$.
Os testes efetuados são apresentados de seguida.

Caso 1

Frames = [1,...,10]
Altura = 20
Largura = 20
vmax = 3
dv= 1
Algoritmo = 5 (Correlação 2D)

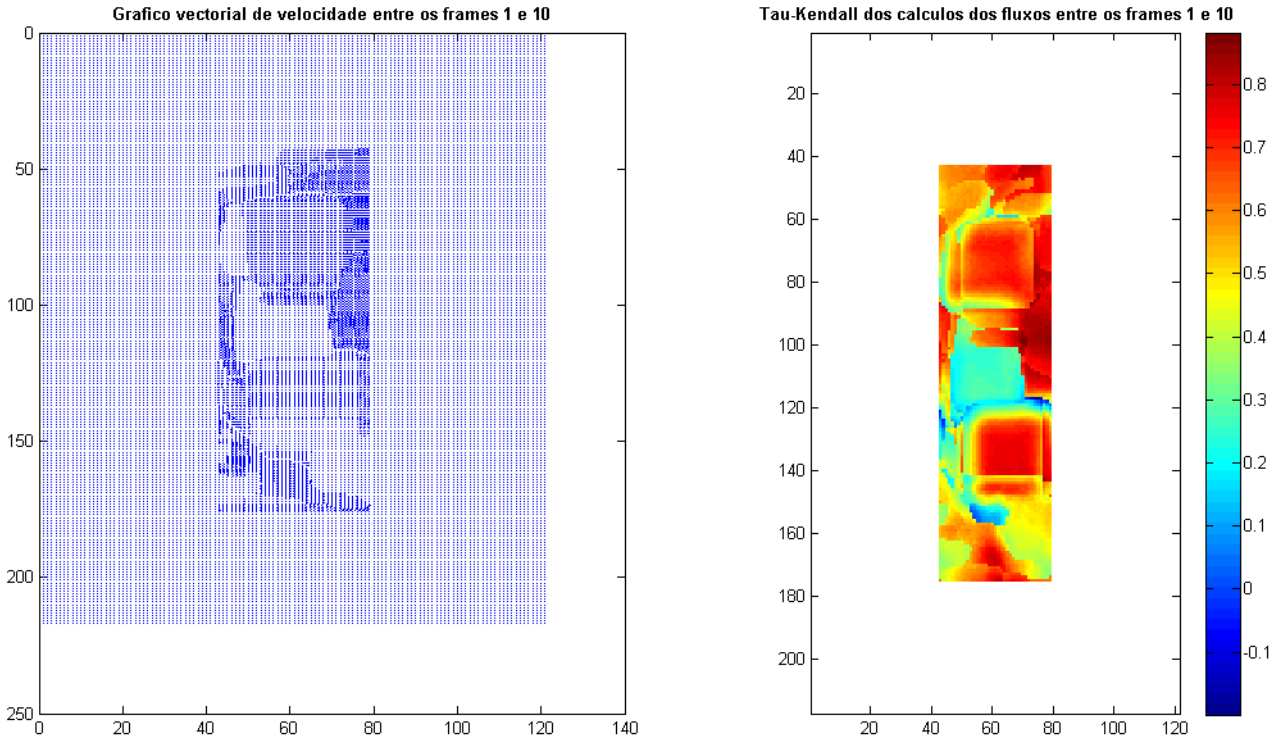

Figura 4.23: Mapa de velocidades (px/s) e respetivos valores de τ-Kendall obtidas com aplicação do algoritmo de Correlação 2D para o teste do Caso 1.

As velocidades obtidas e os respetivos valores de τ-Kendal aproximados nas regiões dos *speckles* artificiais foram:

A : (-1,0); $\tau \simeq 0.6908$
B : (0,1); $\tau \simeq 0.7562$
C : não determinada por estar fora da janela do mapa de velocidades

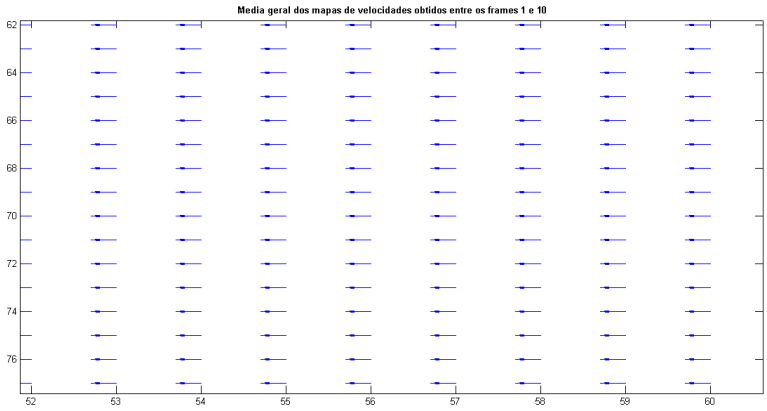

Figura 4.24: Pormenor do mapa de velocidades (px/s) obtido para o caso 1. Zoom sobre o *speckle* artificial A.

Como era de esperar, uma vez que se utilizaram muitos frames, as velocidades das regiões de interesse foram perfeitamente determinadas. Também de esperar, a região onde as velocidades foram determinadas é muito restrita.
Verificou-se que os *speckles* artificiais afetam fortemente o cálculo do fluxo nas regiões à sua volta. Isto acontece nos casos em que se verificam situações como a da figura 4.25, em que é atribuída uma determinada velocidade a um pixel do primeiro frame da sub-pilha por, em frames posteriores, existirem *speckles* nas janelas da sua sub-pilha a deslocarem-se a essa velocidade.

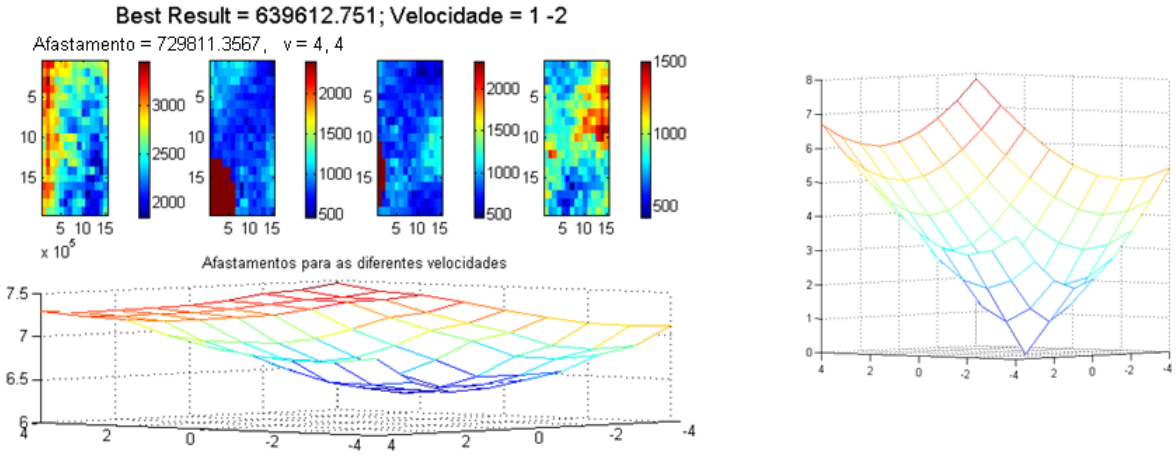

Figura 4.25: Cálculo do fluxo a duas dimensões numa sub-pilha. Figura produzida através do código CalcFluxo_ NFrames_ 2D.m. Em cima vários blocos da sub-pilha analisada. Em baixo, função $F(v)$ em função das velocidades analisadas (eixos ox, oy). À esquerda, função de referência produzida.

Caso 2:

Foi testado de seguida o mesmo algoritmo mas para uma janela de frames menor.

Frames = [2,3,4,5]
Altura = 20
Largura = 20
vmax = 5
dv= 1
Algoritmo = 5 (Correlação 2D)

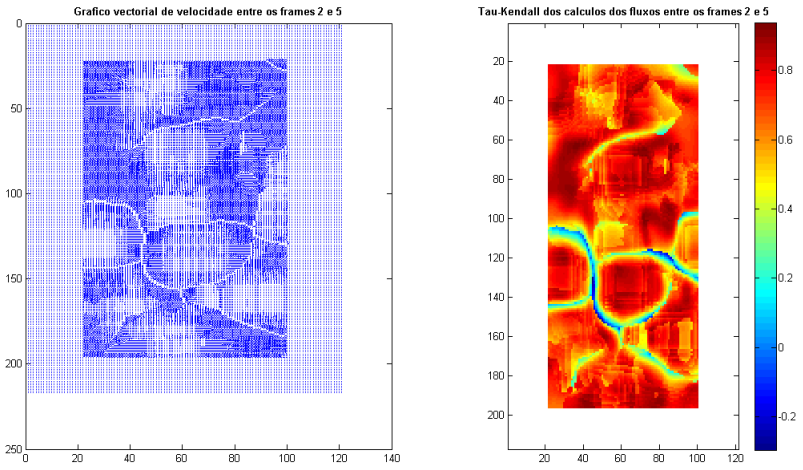

Figura 4.26: Mapa de velocidades (px/s) e respectivos valores de τ-Kendall obtidas com aplicação do algoritmo de Correlação 2D para o teste do Caso 2.

As velocidades obtidas para os três *speckles* foram:

A : (-2,0); $\tau \simeq 0.8142$
B : (1,2); $\tau \simeq 0.8464$
C : (-1,-2); $\tau \simeq 0.6255$

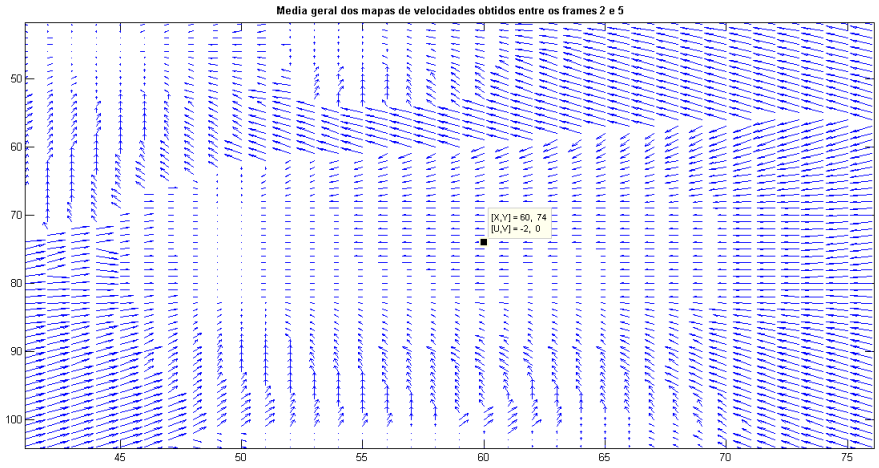

Figura 4.27: Pormenor do mapa de velocidades (px/s) obtido para o caso 2. Zoom sobre o *speckle* artificial A.

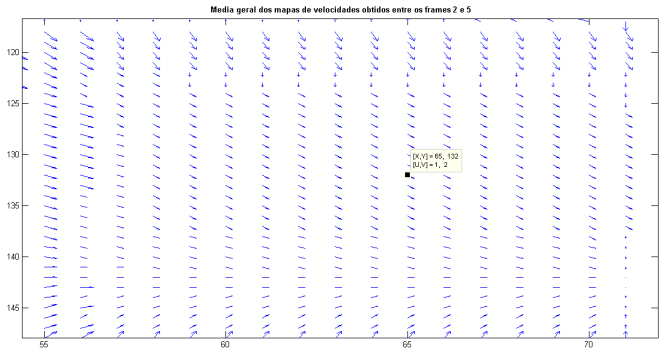

Figura 4.28: Pormenor do mapa de velocidades (px/s) obtido para o caso 2. Zoom sobre o *speckle* artificial B.

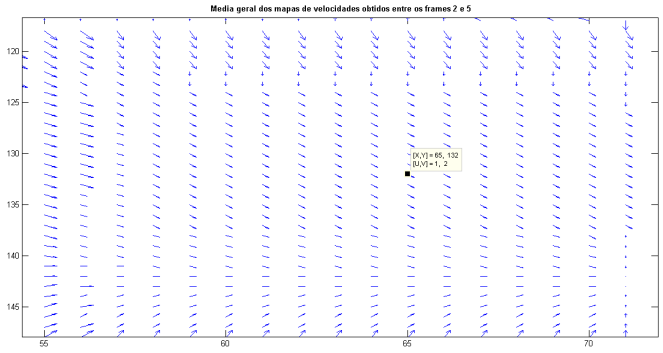

Figura 4.29: Pormenor do mapa de velocidades (px/s) obtido para o caso 2. Zoom sobre o *speckle* artificial C.

Neste caso as velocidades encontradas não foram tão próximas das velocidades reais. Este resultado pode ter-se verificado devido ao reduzido número de frames ou a ter sido utilizada função de correlação que, como visto nos casos a uma dimensão, não permite aproximações tão boas.

4.2.3 Resultados da análise de decaimento nos dados de teste

Os testes ao algoritmo de cálculo do decaimento de intensidade incidiram sobre imagens geradas com o *script* DadosTeste.m, tal como para os testes ao algoritmo do fluxo a 1 dimensão.
Os testes efetuados permitiram avaliar o desempenho deste algoritmo e comparar os resultados obtidos com a média e a mediana. Estas funções são as funções de que o algoritmo dispõe para colapsar as janelas selecionadas em cada frame.
Os parâmetros dos filmes de teste mais relevantes e sobre os quais os testes se baseiam são:

- Decay (decaimento)
- Ruído
- Dimensão das regiões selecionadas

Serão apresentados três casos que generalizam os testes efetuados para este algoritmo. Em cada caso são indicados os parâmetros utilizados para construir as pilhas de teste e a função escolhida

pelo utilizador, média ou mediana.

Caso 1:

O primeiro caso é referente ao teste de controlo, onde todos os parâmetros são ótimos. Ou seja, não há ruído nas imagens e o decaimento é baixo e portanto facilmente detetado.
Os parâmetros utilizados para produzir a pilha de teste e os parâmetros do programa de cálculo de decaimento foram os seguintes.

Parâmetros do filme artificial:
• Dim = 100
• Nframes = 100
• Nspeckles = 8
• **ruído = 0**
• dt = 1.0
• t = (0:dt:(Nframes-1)*dt)'
• dv = 0.1
• vmax=0.1 px/s
• vmin=0.1 px/s
• **decay = 100 s**
• sigma = 7
• Aspeckles(:,1) = 4+ 0.3*rand(Nspeckles,1);

Parâmetros do Decaimento Manual SemiAutomatico:
• **Frames a utilizar = [10,20,25,30,40,50]**
• Número de regiões = 1
• **Função : Média/Mediana**

Os parâmetros indicados utilizados para a pilha de testes permitem obter pilhas em que o pico dos *speckles* no primeiro frame está próximo de 1. Como para elaborar estas imagens é utilizada a função de convolução, a amplitude dos *speckles* varia após a aplicação desta função, e não se verificam exatamente os valores indicados pelos parâmetros. Por isso é necessário indicar que a amplitude dos *speckles* é de 4 ± 0.3*rand(Nspeckles,1) para provocar uma curva com valores de *speckles* próximos de 1.
Uma vez que o decaimento é de 100 s, ao fim de 100 frames espera-se que as intensidades tenham baixado $\frac{1}{e}$ px. Mais uma vez devido à função de convolução, tal não é o que se visualiza nos gráficos de perfis da pilha gerada, embora implicitamente seja este o valor de decaimento nos dados. Por isso o algoritmo de decaimento funciona para estes testes, embora visualmente não se percepcione o decaimento esperado.
Neste caso não há distinção entre as rotinas quando é utilizada a função da média ou a função da mediana, uma vez que não há ruído nos dados.
Como se pode ver na figura 4.30, verifica-se que nestas condições é possível o programa detetar perfeitamente o decaimento. O valor de decaimento obtido foi de 100 s. De facto, alterando o parâmetro decaimento para valores como 1000 ou 2000 s, o algoritmo continua a ter um bom desempenho, apesar de não ser tão preciso. Por exemplo, alterando o parâmetro *decay* para 1000 s obtêm-se resultados entre 1000 ± 200 s nos melhores casos.

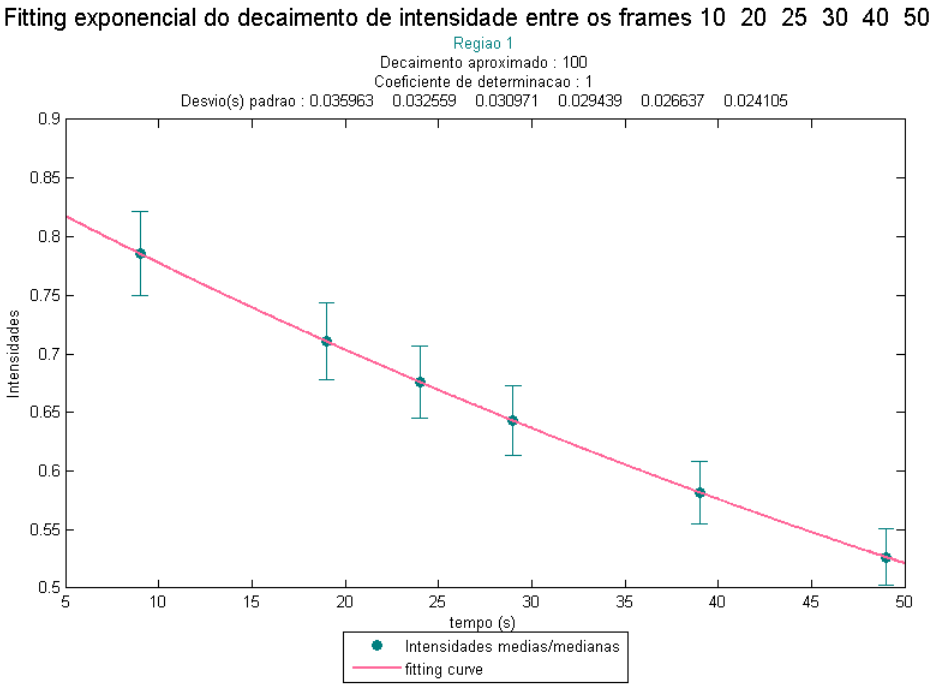

Figura 4.30: *Fitting* exponencial obtido para o caso de teste 1.

Contudo, os resultados dependem totalmente da janela utilizada em cada frame. Na figura 4.31 temos dois exemplos de janelas admitidas em testes com as mesmas condições deste. Os resultados obtidos com janelas como a da esquerda são muito menos exatos, podendo até dar resultados negativos, enquanto que os resultados com a janela da direita são sempre muito próximos dos decaimentos reais induzidos

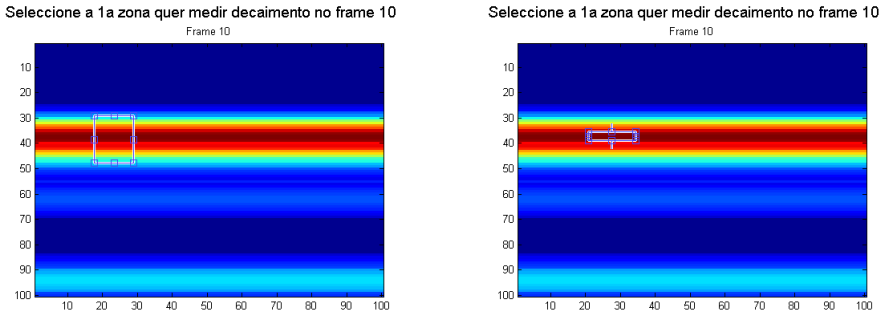

Figura 4.31: Janelas de cálculo de decaimento. A janela da esquerda tem de altura $18px$ e a da esquerda de altura $5px$.

Quanto mais restrita ao pico de um *speckle* forem as janelas a utilizar, melhor são os resultados, sem exceção. Tal é de esperar uma vez que ao precisar a zona da curva de um *speckle* que será observada, há menos erros de incidência na área pretendida do que se procurarmos abranger uma região mais ampla.
Como é fácil de compreender, quanto mais uniformes forem as janelas selecionadas e mais precisa a localização de cada janela na região em estudo, melhores os resultados.
Com os testes efetuados sem ruído pudémos também verificar, como era de esperar, que quanto

mais baixo o decaimento mais exactos são os resultados obtidos.
O gráfico seguinte (Figura 4.32) é relativo a um teste efetuado com os mesmos parâmetros do teste actual mas com ruído de 50%, ou seja, *amplitude de ruído* $\approx$ *amplitude do sinal*.

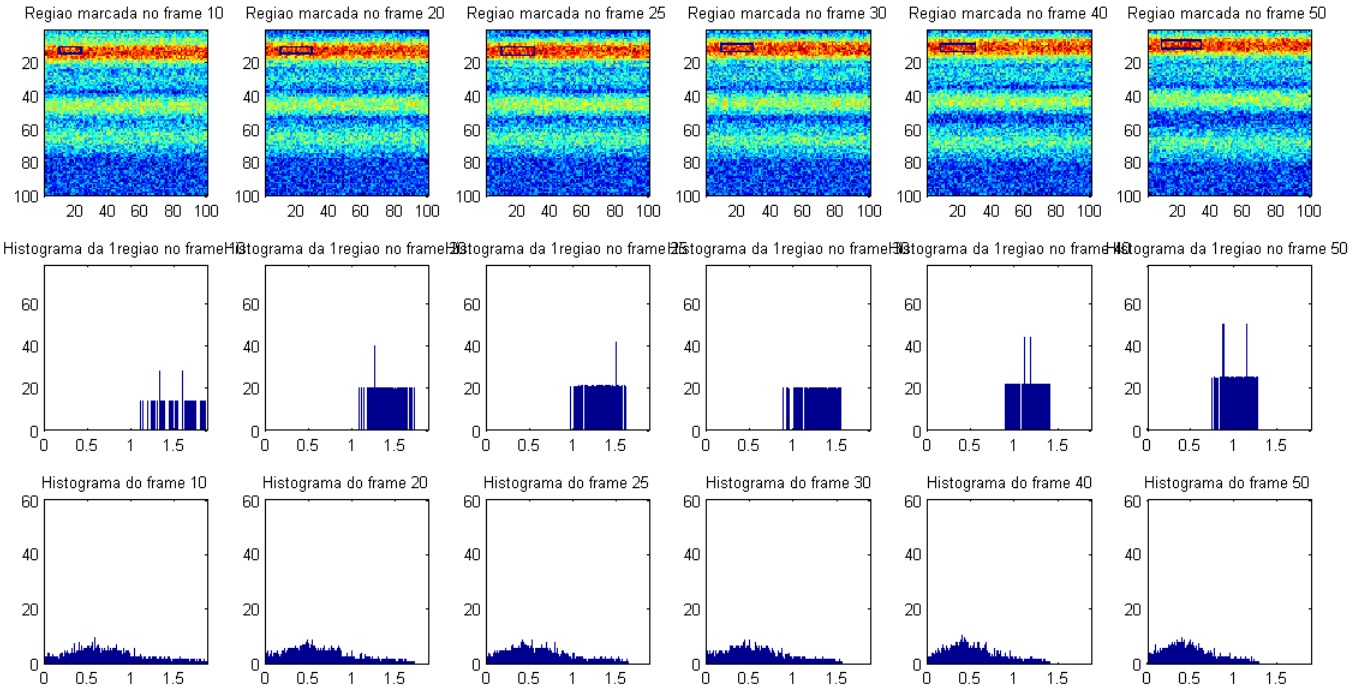

Figura 4.32: Histogramas das regiões selecionadas caso 1. Na primeira linha temos os frames analisados e os retângulos correspondentes às janelas selecionadas em cada frame. Na segunda linha os histogramas dos retângulos selecionados no respetivo frame. Na terceira linha histogramas dos frames (toda a área) respetivos à coluna.

Podemos verificar pelos gráficos dos histogramas o decaimento das intensidades. Na segunda linha temos a intensidade das regiões selecionadas enquanto que na terceira linha as intensidades dos frames na totalidade. Pelo histograma da janela selecionada no frame 10 verificamos que as intensidades nesta janela variaram aproximadamente entre 1.3 e 2. No frame 50 as intensidades na janela variaram aproximadamente entre 0.7 e 1.3.
Nos histogramas da terceira linha da figura podemos observar que no frame 10 ainda há bastantes valores de intensidades mais altas. Ao longo dos frames, estes valores deixam de estar presentes para se acumularem mais à esquerda nos histogramas, ou seja, há mais pontos com intensidades mais baixas.

Caso 2:

Os parâmetros utilizados para produzir a pilha de teste e os parâmetros do programa de cálculo de decaimento foram os seguintes.

Parâmetros do filme artificial:
- Dim = 100
- Nframes = 100
- Nspeckles = 8
- **ruído = 0.5**
- dt = 1.0
- t = (0:dt:(Nframes-1)*dt)'
- dv = 0.1
- vmax=0.1 px/s
- vmin=0.1 px/s
- **decay = 800**
- sigma = 7

Parâmetros do Decaimento Manual SemiAutomatico:
- **Frames a utilizar = [1,10,15,20,25,30,40,50]**
- Número de regiões = 1

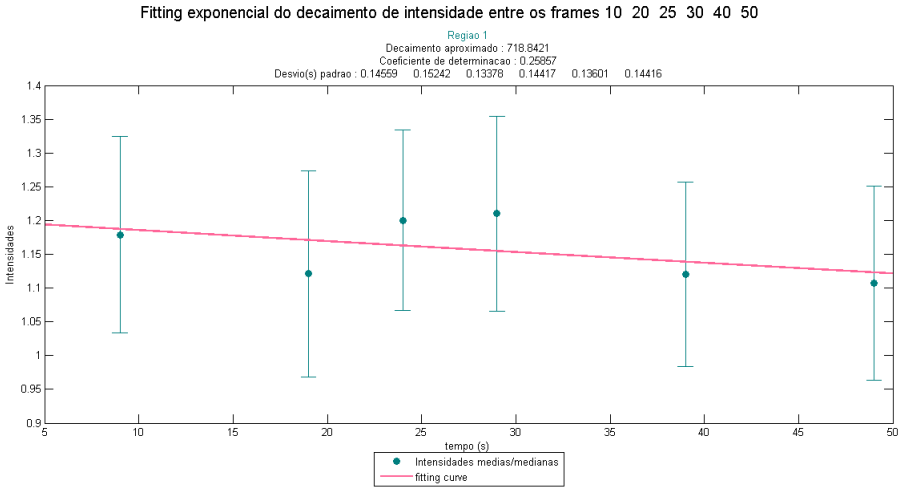

Figura 4.33: *Fitting* exponencial obtido para caso de teste 2 com a função média.

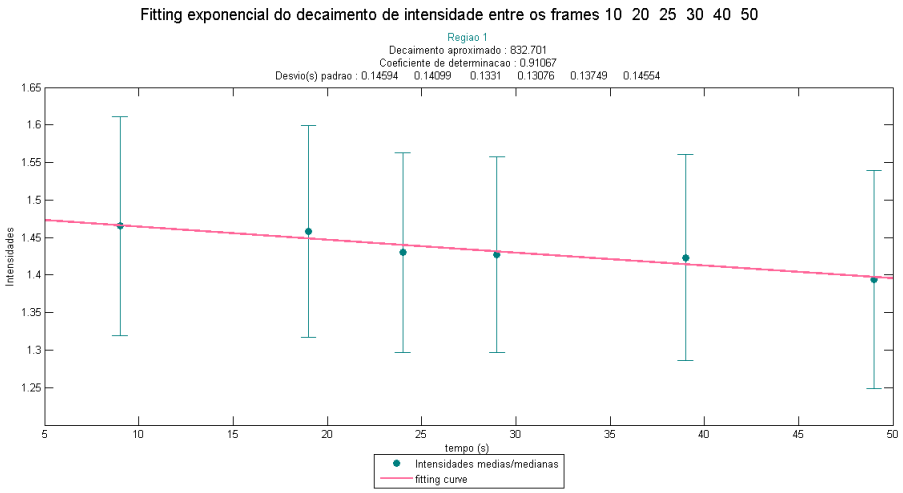

Figura 4.34: *Fitting* exponencial obtido para caso de teste 2 com a função mediana.

No caso da média o decaimento encontrado foi de 718 s, enquanto que no caso da mediana foi de 832 s. Os resultados obtidos com a função mediana são, em geral, melhores do que os resultados obtidos com a função média, como podemos verificar nas figuras 4.33 e 4.34.
O efeito do ruído é percetível no gráfico das figuras anteriores pelas barras de desvios-padrão para cada média ou mediana calculada.
Quando o ruído é igual ou superior a 0.5 convém selecionar retângulos com uma largura considerável, pois quanto mais dados menor o impacto do ruído nas médias ou medianas calculadas das janelas.
No caso das imagens originais, isto pode traduzir-se por utilizar mais do que um *speckle* para medir o decaimento na maioria dos *speckles*. Contudo, pode não ter bons resultados por os *speckles* terem intensidades diferentes. Para contornar este problema convém selecionar regiões com intensidades semelhantes, ou seja, *speckles* com cores semelhantes nas imagens.

4.3 Resultados para os dados originais

4.3.1 Resultados do fluxo a 1D

Os filmes mais adequados à abordagem elaborada foram os filmes das células *Drosophila S2*. É nestes filmes que se observam mais nitidamente os microtúbulos, que se constatou menor rotação e translação do fuso mitótico ao longo dos frames e nos quais há menos ruído. Os resultados seguintes foram obtidos para um filme de uma célula de *Drosophila S2* considerado o melhor filme disponível.

Serão apresentados vários casos analisados. Em cada caso são indicados os parâmetros utilizados na aplicação da rotina CalcularFluxo.m. Os casos estão ordenados por ordem crescente do primeiro frame das pilhas testadas.
Os frames avaliados são os frames das pilhas de novas imagens, ou seja, as imagens que se obtêm depois da marcação dos polos individualmente em cada filme.
Nas situações em que a sequência de frames avaliada não consiste somente de frames consecutivos, os frames em falta não foram considerados. Os frames que foram excluídos foram-no por estarem deslocados na sequência de frames de interesse. Alguns frames sofrem desfazamento em relação ao anterior e/ou seguinte após marcação dos polos, outros já apresentam desfazamento no filme original. Por exemplo, o frame 6 das novas imagens estava relativamente deslocado do centro da imagem em relação aos frames 5 e 7; o frame 12 sofre rotação relativamente aos frames 11 e 10 e a marcação dos polos não foi eficaz para anular esta rotação.

Em todos os casos a janela de trabalho (bloco de trabalho) utilizado teve dimensão [18x14]px. Esta foi a dimensão escolhida após várias medições de diâmetros de *speckles* e resultados em dados de teste.
Tal como já foi referido na secção de teste, no filme em questão, em média os *speckles* têm dimensão 10 a 20 px. Uma vez que na abordagem a uma dimensão o deslocamento é avaliado na vertical, optámos por criar janelas com cerca do dobro da altura do *speckle*. Contudo, uma vez que a altura do bloco é inversamente proporcional à área da imagem que pode ser avaliada, optou-se por não se considerar uma janela tão alta. A largura considerada foi a referida, uma vez que aumentando a largura aumenta também o número de dados a considerar em cada janela. Uma vez que ela será colapsada ao longo das suas linhas, não é aconselhável considerar demasiadas colunas na janela, para não confundir informação.

Os mapas de velocidades produzidos foram filtrados de forma a que sejam visualizadas apenas regiões de interesse. Estas são escolhidas através de filtragem das zonas do frame que tenham

intensidade superior a um valor escolhido.
Os mapas de velocidades e pesos produzidos nesta secção são mapas obtidos após a aplicação do mapa conformacional invertido aos resultados da velocidade e aos resultados de pesos. As velocidades são calculadas sobre as pilhas de imagens dos mapas conformacionais. De seguida aplica-se aos mapas resultantes a função de mapa conformacional linhas paralelas - elipse, o que permite visualizar as velocidades nos locais dos *speckles*. Após estas transformação aplica-se então uma filtração das regiões que no primeiro frame da sequência avaliada são regiões escuras, e para as quais não há tanto interesse em visualizar as velocidades obtidas.
Através dos mapas de velocidades e de pesos não é percetível se há alguma distinção concreta entre velocidades no hemisfério norte e sul. Contudo, com a visualização das velocidades apenas nas zonas de *speckles*, é possível percecionar e quantificar uma simetria em relação ao sinal das velocidades nos dois hemisférios.

Em cada caso dos seguintes, para cada hemisfério foram calculadas médias ponderadas das velocidades, em que os pesos de cada valor foram os respectivos τ-Kendall.

Caso 1:

Parâmetros a utilizar no programa:
• Sequência = [3,4,5,7]
• vmax = 3.8;
• dv = 0.1
• Algoritmo: Função das diferenças absolutas em relação à media, janela colapsada com o máximo.

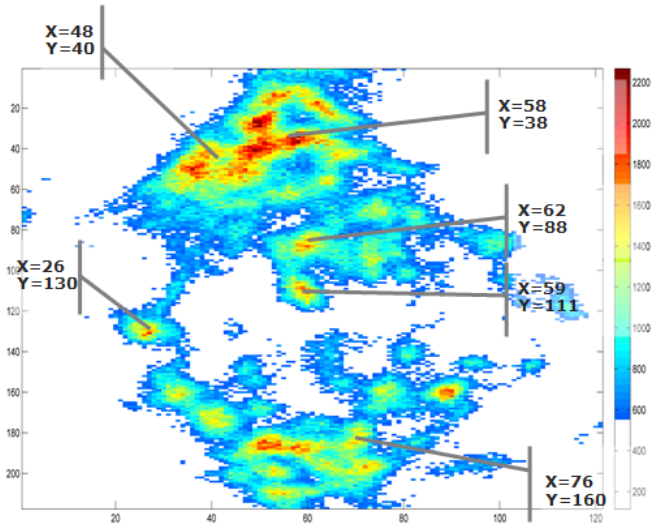

Figura 4.35: *Speckles* avaliados no caso 1. Frame 3 do filme analisado com a localização de alguns *speckles*.

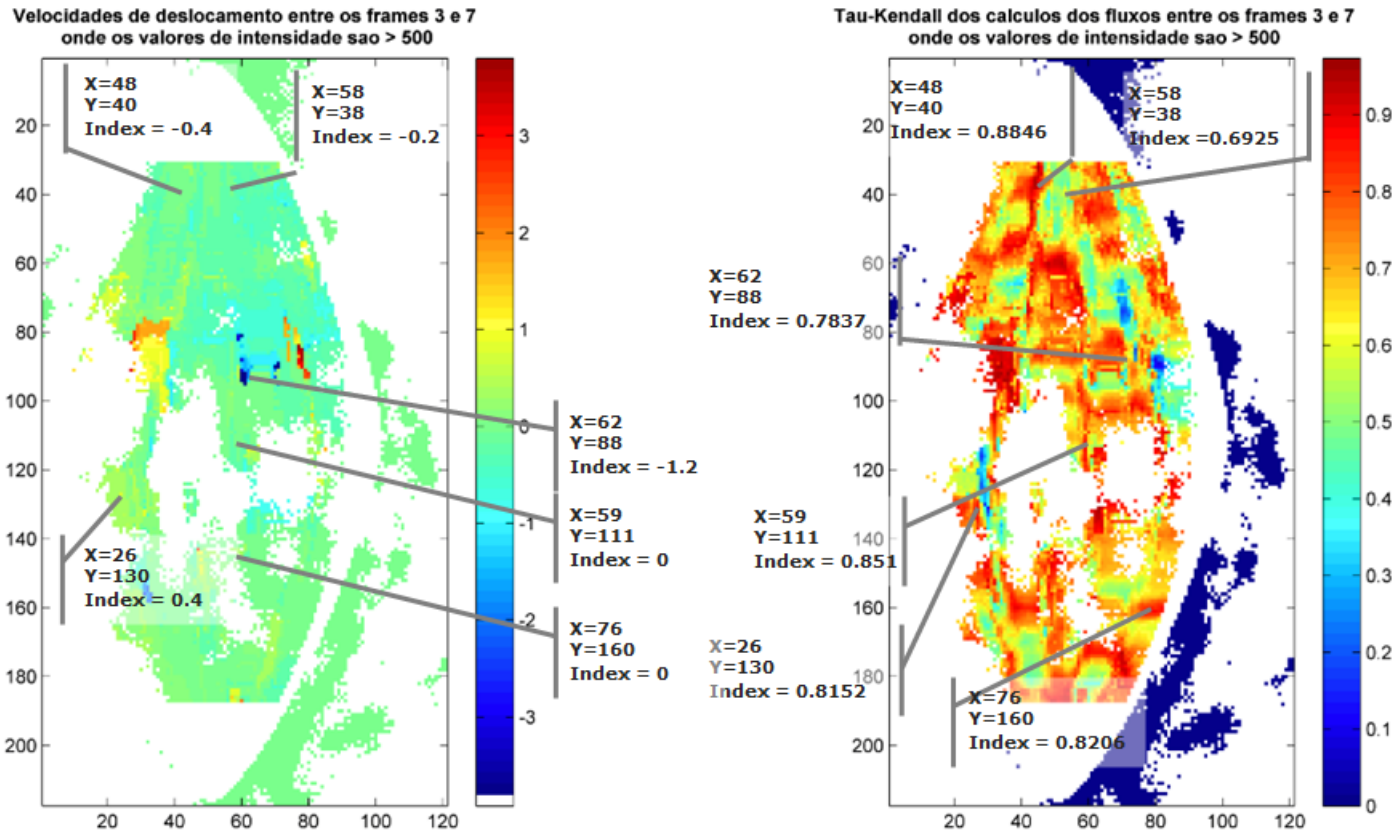

Figura 4.36: Mapa de velocidades (px/s) e mapa de τ-Kendal obtidos para o caso 1 nas regiões onde a intensidade dos pixeis no frame 3 é superior a 500. Alguns valores de velocidade registados assinalados.

Na figura 4.35 pode-se observar os *speckles* evidenciados através de manipulação do *colormap* utilizado. Foram identificados com a côr branca todos os pixeis com intensidades inferiores a 700. Para os pixeis evidenciados na imagem estão evidenciadas também as velocidades calculadas.
No mapa de velocidades da figura 4.36, todos os pixeis destacados no hemisfério superior têm valores negativos de velocidade. Em redor dos pixeis evidenciados também se verificam velocidades negativas. Os valores mais frequentes no hemisfério superior do mapa são o valor 0 e -0.2 px/s.
O *speckle* associado ao ponto (62,88) é o que regista velocidades negativas mais acentuadas, e os valores de confiança nesta região não são muito baixos, pelo que se pode entender que houve um movimento mais brusco nesta região.
Quer o *speckle* do centro do fuso quer o destacado na posição (76,170) apresentam velocidade nula e os valores de τ-Kendal nestas regiões são elevados. *De facto, as regiões onde o fluxo registado é nulo são as regiões de confiança mais elevada.*
O *speckle* da posição (26,130) deslocou-se para baixo a uma velocidade de 0.4px/s. Contudo, também no hemisfério superior a maioria das velocidades registadas foram 0 px/s.

A velocidade média registada no hemisfério superior foi $-0.181px/s = -0.8145\mu m/min$. No hemisfério inferior foi de $0.01px/s = 0.045\mu m/min$.
Em média, as regiões do hemisfério inferior do mapa da figura anterior mantiveram-se praticamente estáticas.

Caso 2:

Parâmetros a utilizar no programa:
• Sequência = [7,8,9,10,11]
• vmax = 3.8;
• dv = 0.1
• Algoritmo: Função das diferenças absolutas em relação à média, janela colapsada com o máximo.

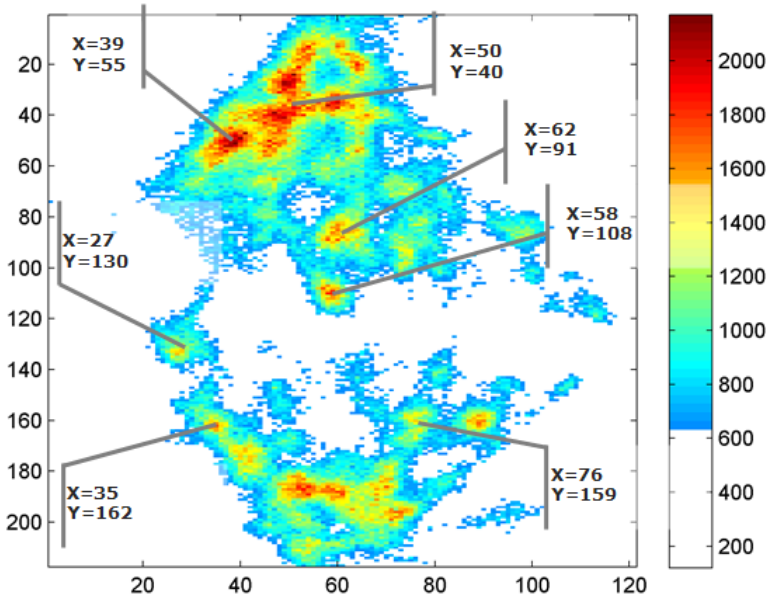

Figura 4.37: *Speckles* avaliados no caso 2. Frame 7 do filme analisado com a localização de alguns *speckles*.

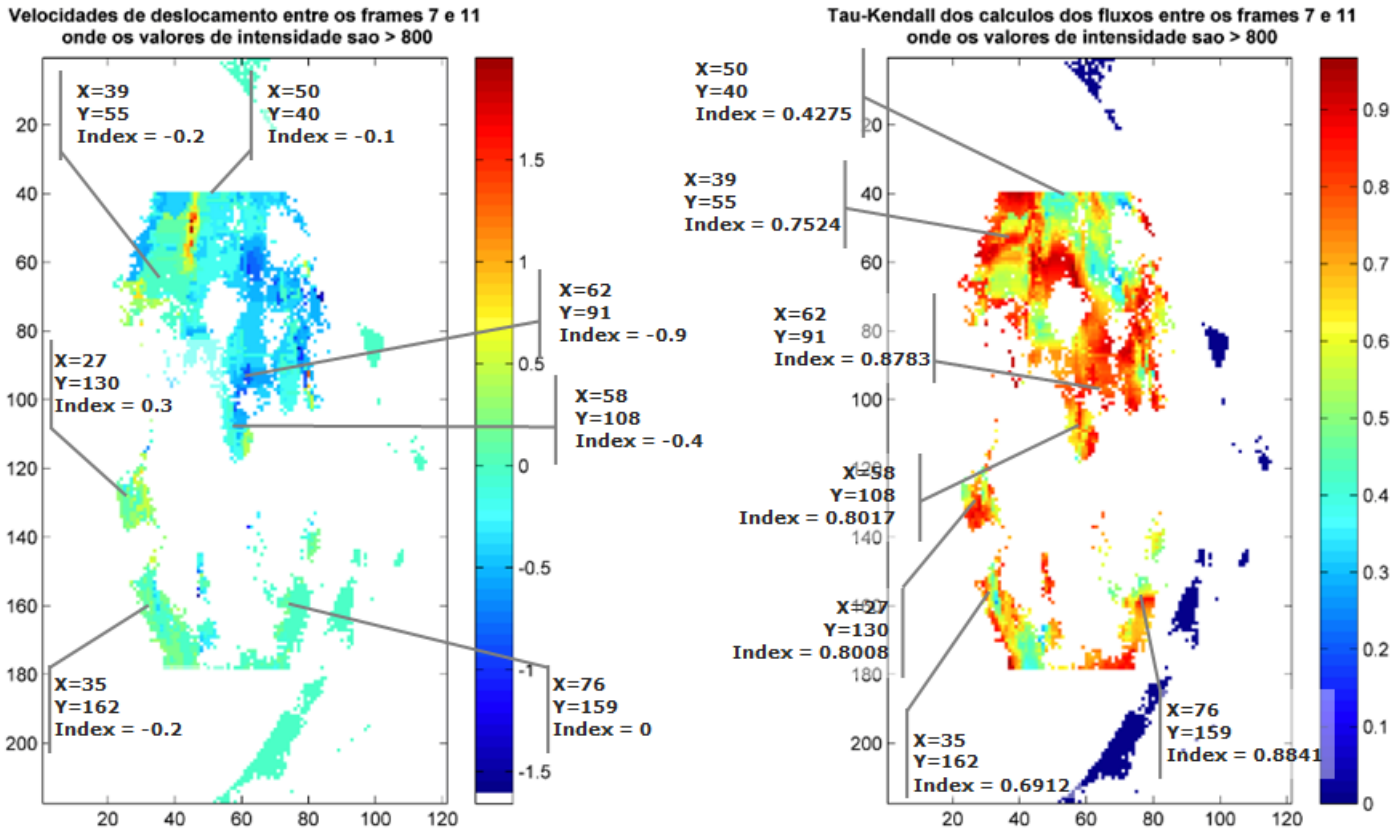

Figura 4.38: Mapa de velocidades (px/s) e mapa de τ-Kendal obtidos para o caso 2 nas regiões onde a intensidade dos pixeis no frame 7 é superior a 800. Alguns valores de velocidade registados assinalados.

Mais uma vez, todas as velocidades dos *speckles* registadas no hemisfério superior são negativas. Na figura 4.37 a região entre os pontos (39,55) e (50,40) assinalados é uma região de intensidades muito altas. As velocidades registadas na figura 4.38 nestas regiões são coerentes com as velocidades obtidas no caso anterior. Embora os valores de τ-Kendal nos picos dos *speckles* não sejam muito elevados, em redor dos picos dos *speckles* a confiança é elevada e os valores de velocidade são negativos de -0.1px/s ou -0.2px/s. Para além desta zona, em vários pontos de velocidade negativas (-0.1px/s, -0.2px/s e -0.3px/s) do hemisfério superior, a confiança é superior a 0.75.
O *speckle* associado ao pixel (27,130) registou velocidades positivas no seu centro, embora alguns pixeis ligeiramente ao lado do centro tenham registado velocidades negativas. Tal não é raro nos mapas obtidos, embora em alguns casos seja mais notório. Podemos perceber pelo mapa que um pouco acima deste píxel foram obtidas velocidades negativas.
O *speckle* associado à posição (35,162) registou algumas velocidades negativas, embora o seu valor de τ-Kendall não seja muito elevado.
A velocidade média registada no hemisfério superior foi $-0.2542px/s = -1.1439\mu m/min$. No hemisfério inferior foi de $0.0347px/s = 0.1562\mu m/min$.
As regiões superiores continuam, em módulo,a apresentar fluxo médio superior.

Caso 3:

Parâmetros a utilizar no programa:

- Sequência = [16,17,18,19,20,21]
- vmax = 3.8;
- dv = 0.1
- Algoritmo: Função das diferenças absolutas em relação à média, janela colapsada com o máximo.

Uma vez que nas imagens rodadas e centradas nos fusos há bastantes oscilações entre o frame 11 e o frame 15, a sequência testada de seguida começa no frame 16.

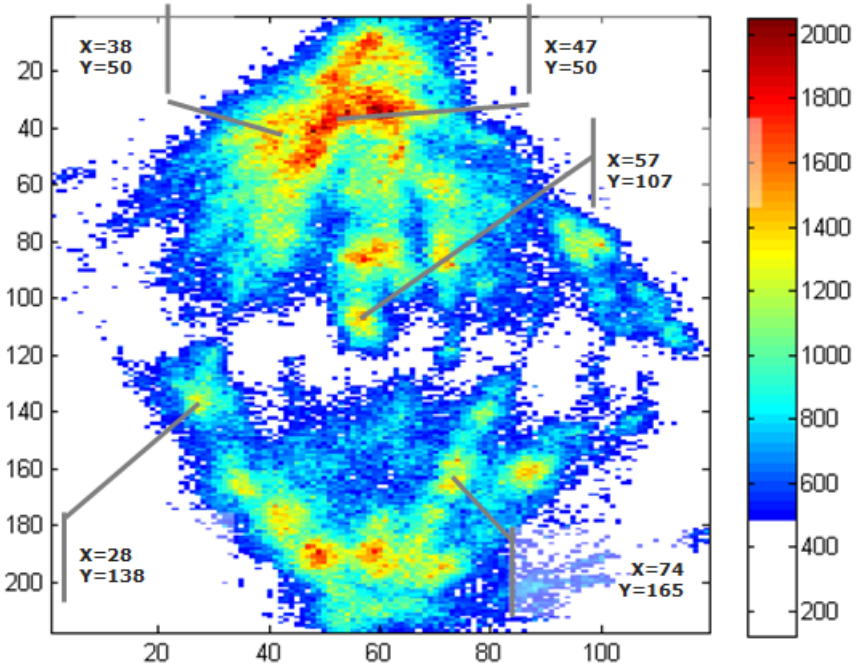

Figura 4.39: *Speckles* avaliados no caso 2. Frame 16 do filme assinalado com a localização de alguns *speckles*.

Para produzir a imagem da figura 4.39 coloriu-se de branco as regiões de intensidade sensivelmente inferior a 500. Note-se que as imagens são produzidas com o comando `imagesc`() do Matlab, e os colormaps atribuidos são diferentes nos casos aqui estudados, pelo que as cores no mapa da figura acima não podem ser comparadas com as cores do mapa da figura 4.37.

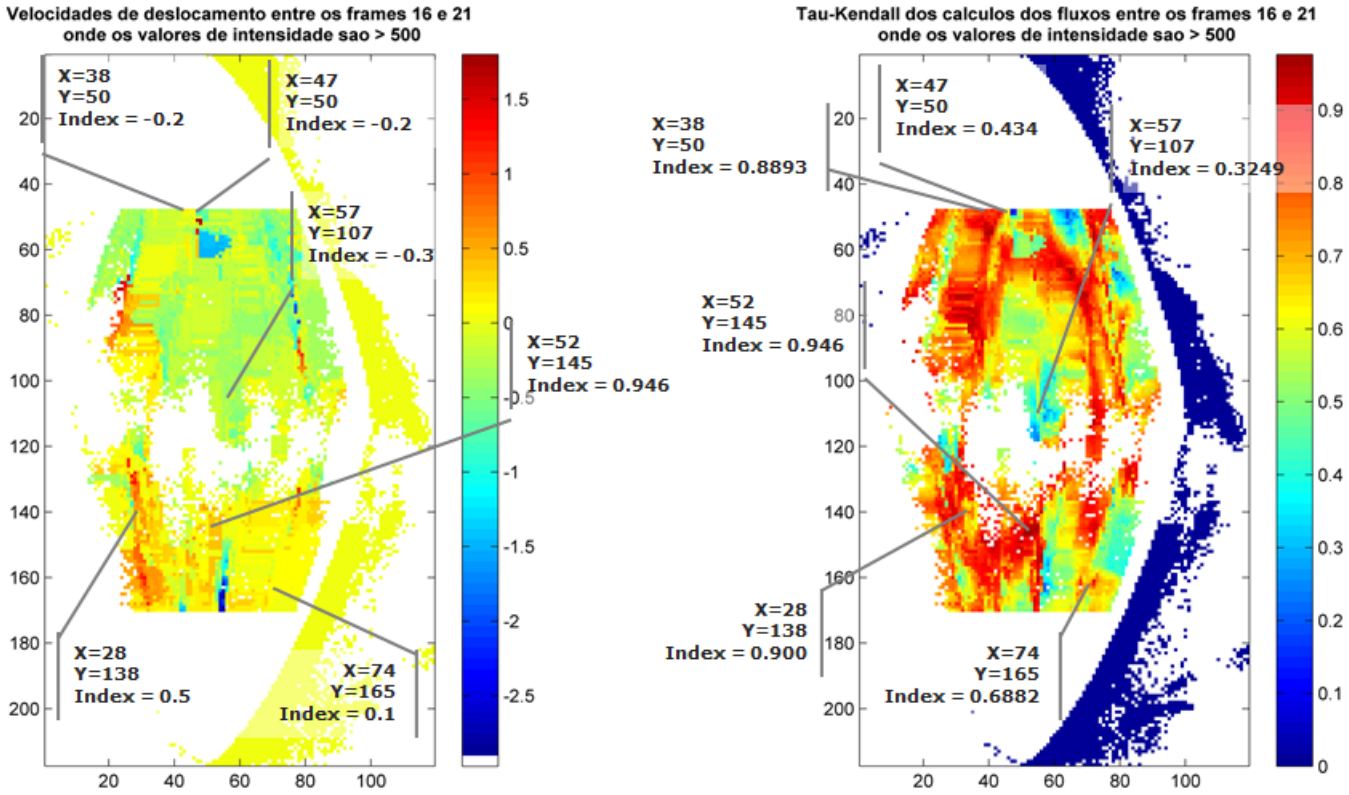

Figura 4.40: Mapa de velocidades (px/s) e mapa de τ-Kendal obtidos para o caso 2 nas regiões onde a intensidade dos pixeis no frame 16 é superior a 500. Alguns valores de velocidade registados assinalados.

Nos mapas de velocidades registados na figura 4.40 obtiveram-se valores absolutos superiores aos dos casos anteriores. Novamente no hemisfério superior verificaram-se maioritariamente valores negativos. Desta vez os valores mais frequentes estão entre -0.2 e -0.5 px/s. Neste mapa o *speckle* localizado mais acima na figura 4.39 já não é totalmente abrangido, pois movimentou-se para cima ao longo dos frames. Na região ainda abrangida deste *speckle* registou-se velocidade -0.2 px/s. Os valores de τ-Kendall para o pico deste *speckle* são novamente não muito elevados. Mas também se verifica que na região à esquerda do *speckle*, onde a intensidade original não é tão alta, as velocidades obtidas são idênticas e os valores de confiança muito razoáveis (cerca de 0.8).
No hemisfério superior à esquerda, na região próxima do pixel (25,85), as velocidades calculadas são positivas. Contudo, pela figura 4.39 podemos visualizar que esta região é de uma zona mais escura, e possivelmente não tem tanta significância.
O *speckle* central voltou a acusar velocidades negativas, -0.3px/s, embora com valores de confiança não muito significativos.
É de realçar que no hemisfério inferior as velocidades estão predominantemente entre 0.1 e 0.5 px/s, e nas zonas onde estas ocorrem há valores de confiança elevados. Agora a região do *speckle* de (28,138) apresenta maioritariamente velocidade 0.5 com bons valores de τ-Kendall.
Um dos valores de valor de $Kendall$ mais elevado é o do pixel (52,145), onde a velocidade obtida foi 0.5px/s e o valor de τ foi 0.946.

Em média, a velocidade registada no hemisfério superior foi $-0.2440px/s = -1.098\mu m/min$. No hemisfério inferior foi de $0.1347px/s = 0.16\mu m/min$.
As regiões superiores continuam, em média,a apresentar fluxo superior.

Caso 4

Parâmetros a utilizar no programa:

• Sequência = [31,32,33,34,35]
• vmax = 3.8;
• dv = 0.1
• Algoritmo: Função das diferenças absolutas em relação à media, janelas colapsadas com o máximo.

A última sequência testeada foi do frame 31 ao frame 35. O frame 31 está representado na figura 4.41 após manipulação do colormap para evidenciar os pixeis de intensidade superior a 500.

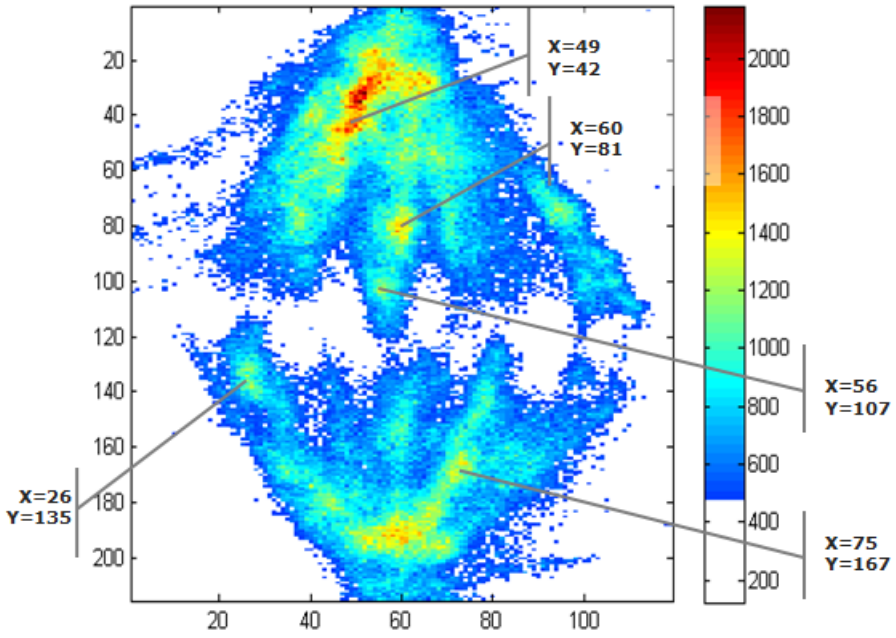

Figura 4.41: *Speckles* avaliados no caso 2. Frame 31 do filme analisado com a localização de alguns *speckles*.

Através da figura 4.41 pode-se percepcionar o fenómeno de decaimento no frame, pois as cores das regiões estão mais diluídas, e alguns *speckles* perderam significativamente intensidade.

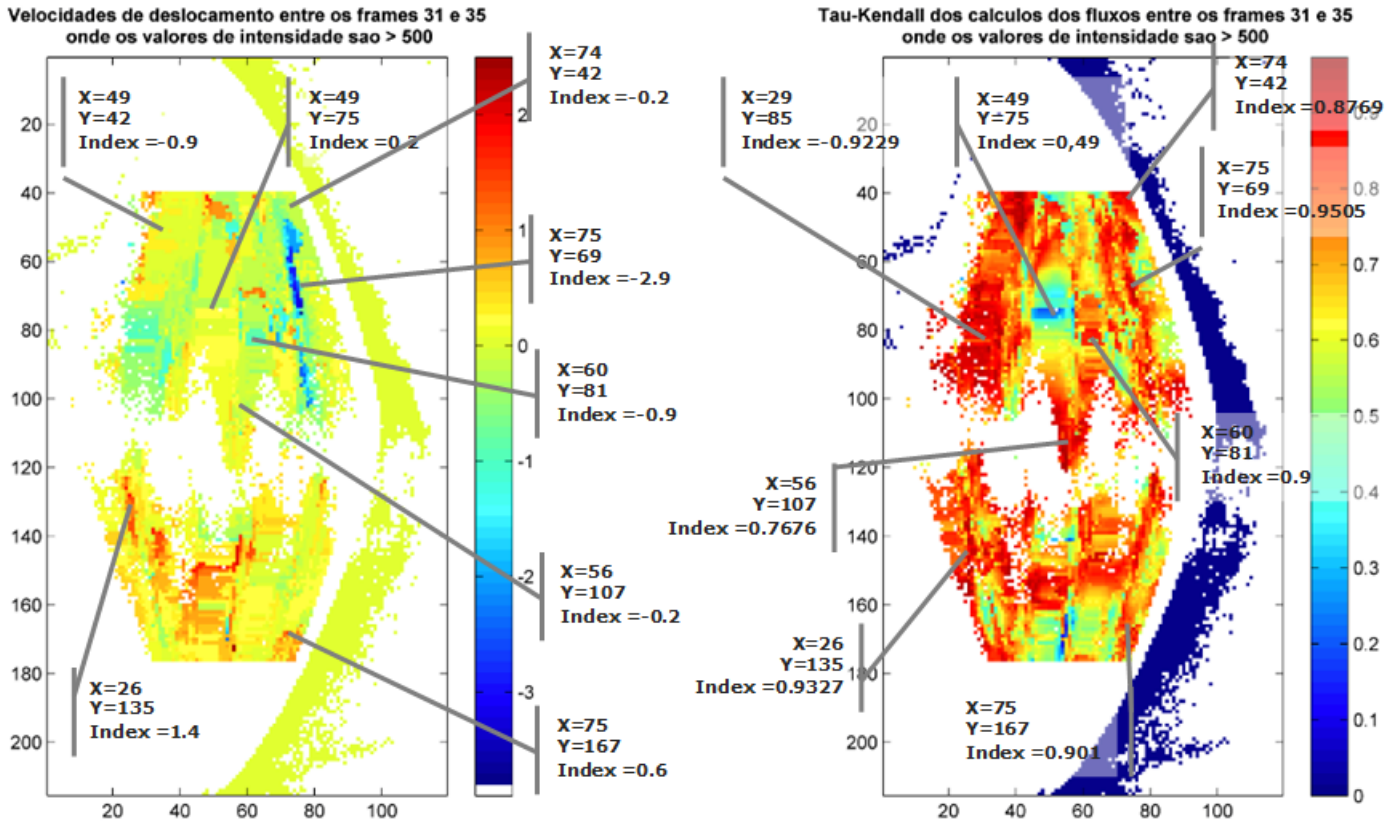

Figura 4.42: Mapa de velocidades (px/s) e mapa de τ-Kendal obtidos para o caso 2 nas regiões onde a intensidade dos pixeis no frame 31 é superior a 500. Alguns valores de velocidade registados assinalados.

O mapa de velocidades da figura 4.42 é o que apresenta maior predominância de valores de confiança satisfatórios. Quer no hemisfério superior quer no hemisfério inferior, as velocidades estão coerentes com o que era expectável, negativas na parte superior e positivas na parte inferior, e valores de τ superiores a 0.8 muito frequentes.
Na região próxima do pixel (49,42), os valores da velocidade são novamente aproximados a -0.3px/s. Embora um pouco à esquerda haja valores com velocidades positivais, estas parecem ser mais um vez associada a zonas mais escuras, como se pode verificar na figura 4.41. No pixel (49,42) foi medida uma velocidade positiva, e no respetivo ponto do mapa de τ's um valor francamente inferior ao predominante próximo de 0.8. A região azul neste ponto do mapa de τ's pode dever-se ao facto de se terem medido aquelas velocidades positivas.
É de realçar a zona assinalada com o pixel (75,69). Nesta zona as velocidades obtidas foram francamente negativas, entre -2.9 e 3.5 px/s, e os valores de confiança respetivos muito bons.
O *speckle* central verificou novamente uma velocidade de -0.2 px/s e valor de confiança 0.767.
No hemisfério inferior o *speckle* da esquerda deslocou-se a uma velocidade mais abrupta em relação às velocidades dos casos 1 e 2. Contudo, a confiança em toda a zona em redor do ponto assinalado neste *speckle* é muito positiva. Na região central do hemisfério inferior também se verificaram velocidades positivas entre 0.4 e 0.6 px/s, sustentadas por bons valores de confiança. O *speckle* da direita movimentou-se a 0.6px/s, e a sua zona apresenta muito bons valores de τ-Kendall.

Em média, a velocidade registada no hemisfério superior foi $-0.0628px/s = -0.2824\mu m/min$. No hemisfério inferior foi de $3.107px/s = 1.3982\mu m/min$. Agora, em média, os pontos de intensidade

alta moveram-se mais rapidamente em baixo.

4.3.2 Resultados da análise de fluxo a 2D:

Nesta secção será apresentado apenas um caso. Foram calculados todos os casos calculados a uma dimensão, mas devido à dificuldade de representação de resultados só um será apresentado.

Caso 1:

Parâmetros a utilizar no programa:
• Sequência = [7,8,9,10,11]
• vmax = 1.2;
• dv = 0.1
• Algoritmo: Função das diferenças absolutas em relação à media.

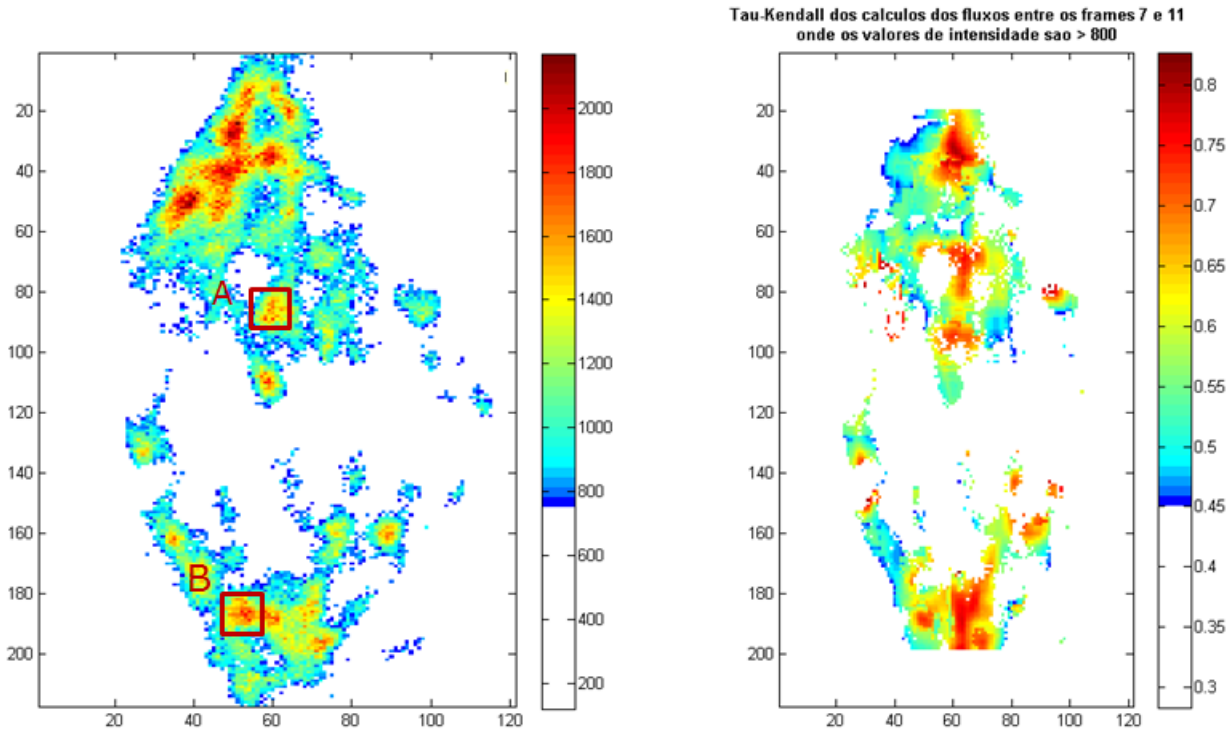

Figura 4.43: Frame 7 do filme e mapa de pesos obtido para o caso 1.

Nas figuras seguintes estão representadas as regiões indicadas nos rectângulos da imagem 4.43.

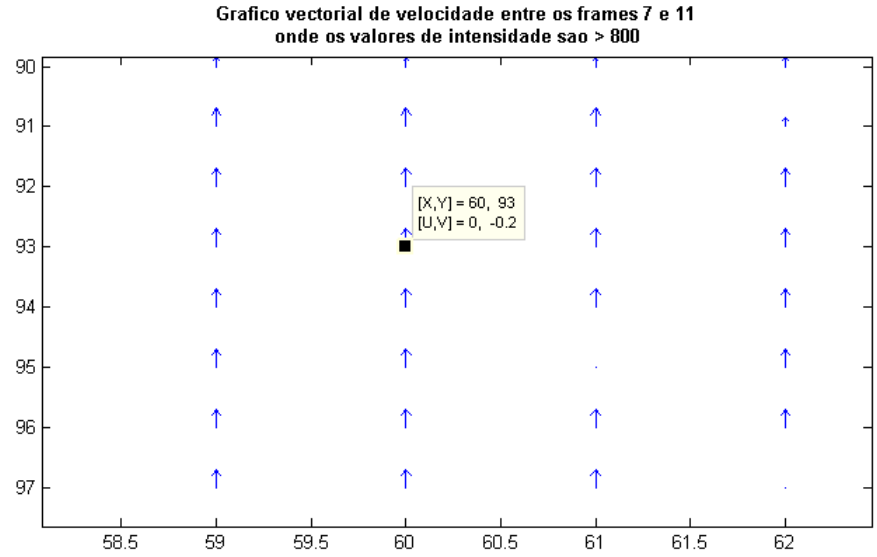

Figura 4.44: Velocidades (px/s) obtidas para região A do frame 7.

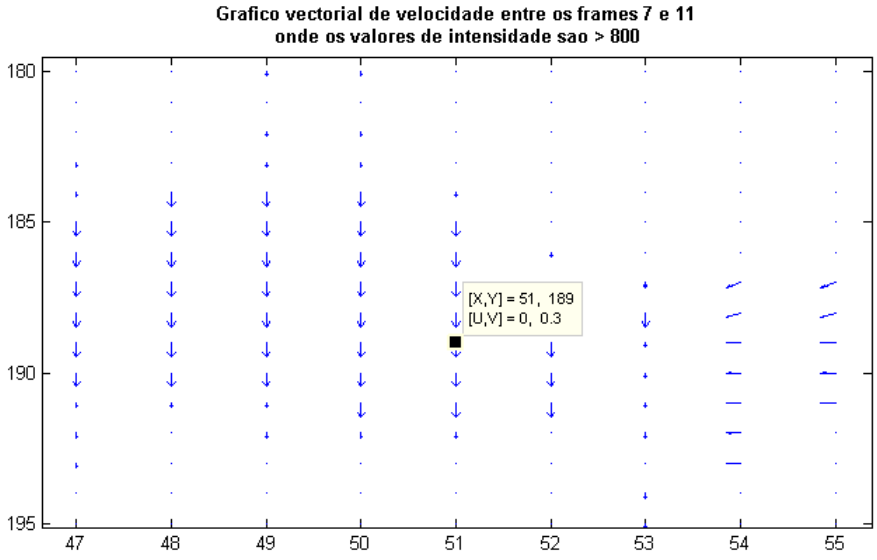

Figura 4.45: Velocidades (px/s) obtidas para região B do frame 7.

Em geral, as velocidades obtidas com os mapas de velocidades a 2D são bastante concordantes com as velocidades os mapas que se obtêm com o cálculo a 1D.

4.3.3 Resultados da análise de decaimento

Serão apresentados resultados relativos ao filme de células *Drosophila S2*, o mesmo analisado também na secção do fluxo a 1D. Os resultados foram obtidos com o código do *script* DecaimentoSemiAutomatico.m. A função utilizada para colapsar as janelas selecionadas no decorrer do programa foi a mediana, uma vez que foi apurado nos testes a dados artificiais ser a função mais adequada.

***Speckles* do hemisfério superior do fuso:**

O spekcle analisado nos hemisfério superior é que está identificado na figura 4.46.

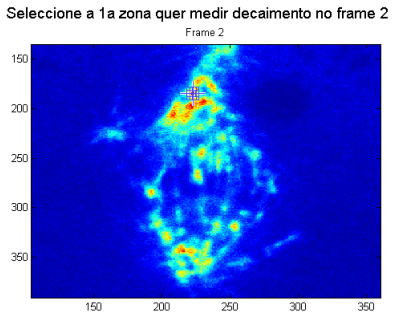

Figura 4.46: *Speckle* do hemisfério superior avaliado no teste de decaimento.

Numa primeira análise avaliámos o decaimento apenas nos primeiros 25 frames, uma vez que poderá haver interesse em subdividir os decaimentos ao longo do eixo temporal do filme.
O *speckle* analisado no hemisfério superior é dos de maior intensidade no fuso. Contudo, com o decorrer da anafase este vai-se fundindo com os *speckles* à sua volta, o que se pode ver em frames posteriores. Esta sobreposição de *speckles* tem influências nos resultados, embora para o *speckle* em questão não tenha impedido as medições de decaimento.

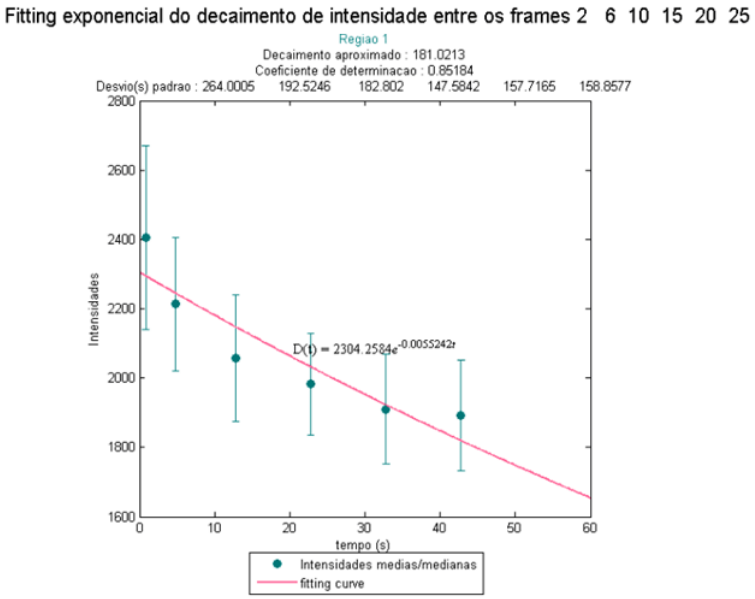

Figura 4.47: Fit exponencial para o *speckle* do hemisfério superior.

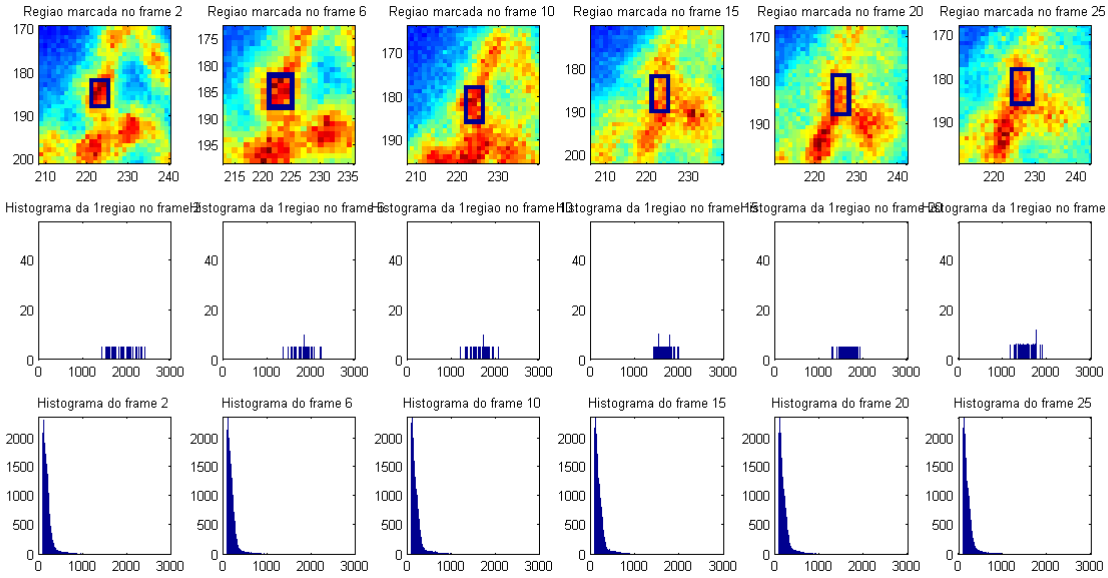

Figura 4.48: Histogramas das janelas do *speckle* inferior e dos frames utilizados. Na primeira linha temos os frames analisados e os rectângulos correspondentes às janelas selecionadas em cada frame. Na segunda linha os histogramas dos rectângulos selecionados no respectivo frame. Na terceira linha histogramas dos frames (toda a área) respetivos à coluna da figura.

A figura 4.47 representa a regressão encontrada para a intensidade do *speckle* em análise. O valor do coeficiente de determinação do *fitting* é satisfatório, 0.85184, e o valor de decaimento encontrado foi de 181.02 s. Foram efetuados mais dois testes ao *speckle* com os mesmos parâmetros e registou-se o mesmo valor de decaimento nos dois testes, com variações de $\pm$ 5 s.
A função de intensidade da região é $I(t) = 2304 \exp -\frac{1}{181.02}t$.
O gráfico da figura 4.48 apresenta as janelas selecionadas neste teste, os histogramas das janelas selecionadas nos respetivos frames e por fim os histogramas dos vários frames percorridos.
Na figura anterior os histogramas das janelas selecionadas traduzem o decaimento encontrado. No 2º frame a maioria das intensidades na janela estão no intervalo [1500,2500] enquanto que no 25º frame estão compreendidas no intervalo [1200,1900].

Avaliou-se também a região deste *speckle* em frames posteriores, do 30 ao 90.
A regressão encontrada é representada na 4.49:

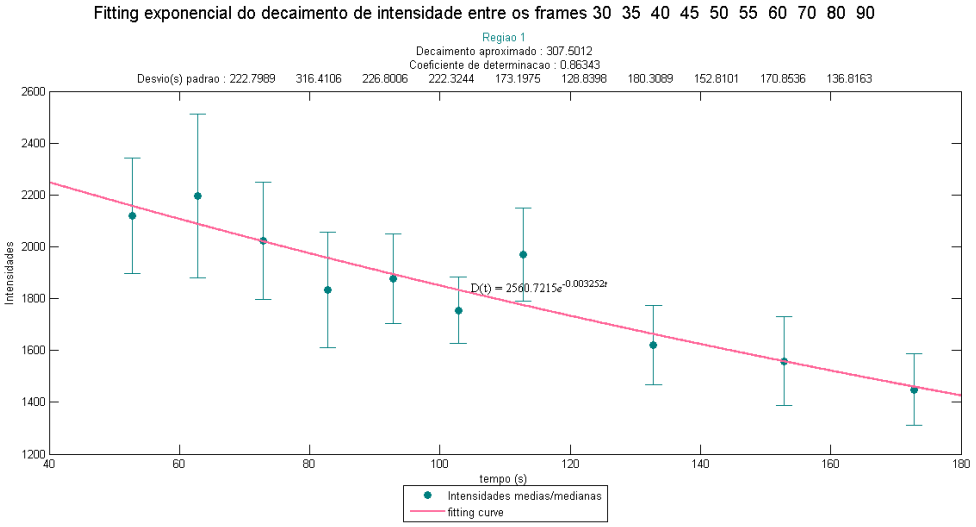

Figura 4.49: Fit exponencial para o *speckle* do hemisfério superior em frames posteriores a 30.

O valor do decaimento encontrado foi, aproximadamente, $307.5s$, e o coeficiente de determinação médio desta regressão foi 0.86343. Este resultado indica que a partir do 30° frame o decaimento é menos acentuado, o que é natural, uma vez que a tubulina vai perdendo a sua fluorescência nos primeiros segundos do filme.

Speckle do hemisfério inferior do fuso:

São agora apresentados os resultados da análise do *speckle* identificado na figura 4.50.

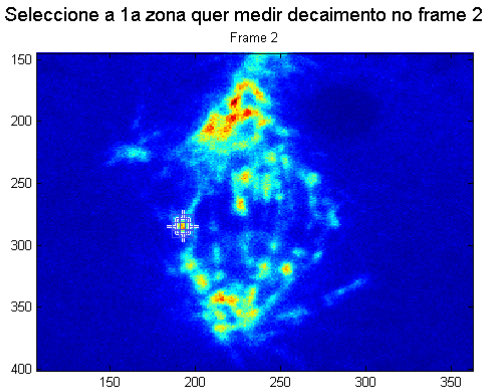

Figura 4.50: *Speckle* do hemisfério inferior avaliado no teste de decaimento.

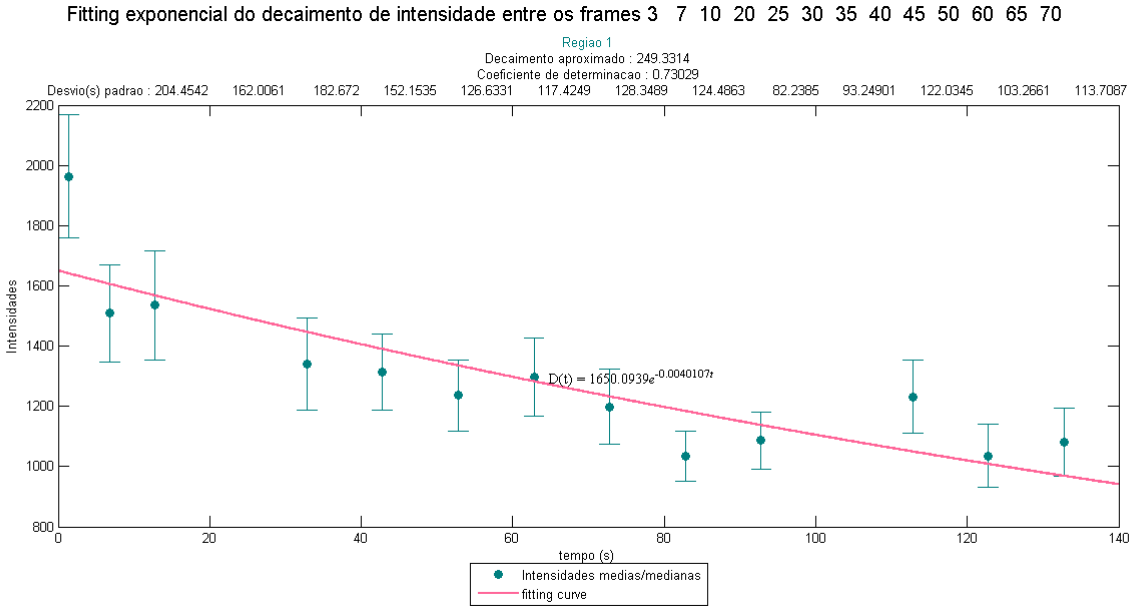

Figura 4.51: Fit exponencial para o *speckle* do hemisfério inferior.

Na figura 4.51 podemos visualizar o *fitt* exponencial obtido para este *speckle* entre os frames 3 e 70. O valor do decaimento obtido foi 249.3s e o coeficiente de determinação da regressão foi 0.73029. A função de intensidade da região é $I(t) = 1650.0939 \exp -\frac{1}{249.331}t$.

***Speckle* do equador do fuso:**

É apresentado de seguida o resultado da avaliação do decaimento num *speckle* do equador do fuso, representado na figura 4.52.

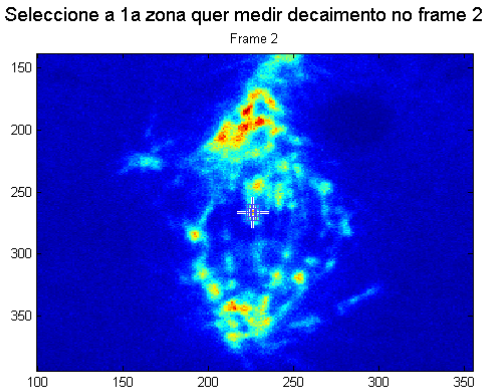

Figura 4.52: *Speckle* situado no equador do fuso avaliado no teste de decaimento.

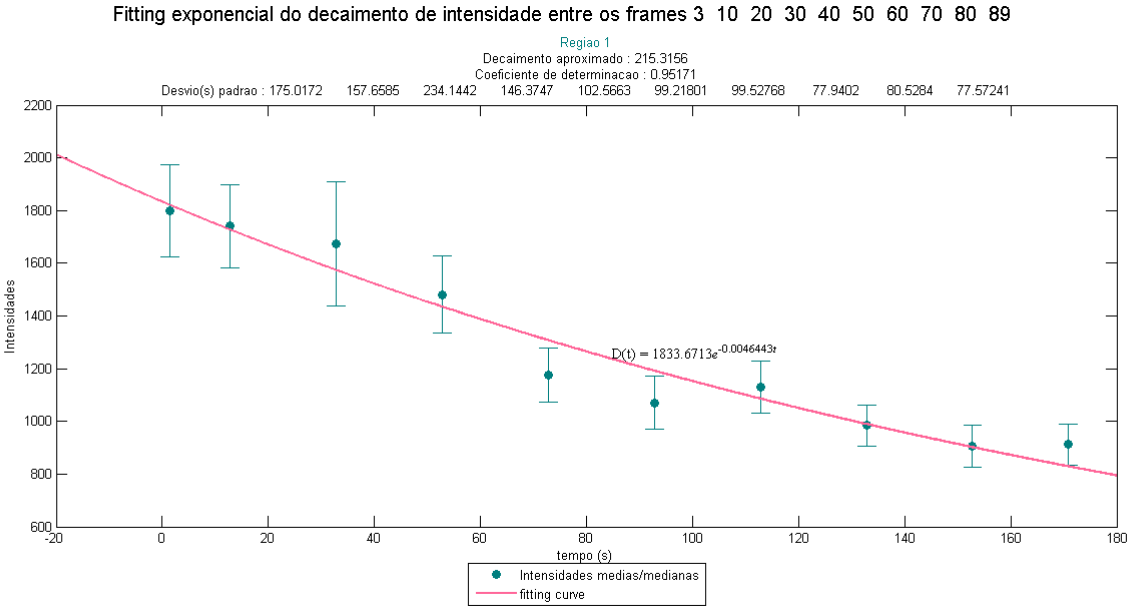

Figura 4.53: Fit exponencial para o *speckle* do hemisfério inferior.

Na figura 4.53 podemos encontrar a curva de regressão encontrada para o *speckle* actual. O valor do decaimento encontrado nesta análise foi de $215.3s$ e o coeficiente de determinação da regressão foi 0.95171. A função de intensidade do *speckle* em função do tempo encontrada para os primeiros 25 frames, é $I(t) = 1833.6713 \exp -\frac{1}{215.3156}t$.
Neste *speckle*, tal como no anterior, o decaimento encontrado foi menos acentuado do que o decaimento encontrado nos outros dois *speckles* para o mesmo intervalo de frames.

Zona escura:

É apresentado de seguida o resultado da avaliação do decaimento numa região escura, representado na figura 4.54.

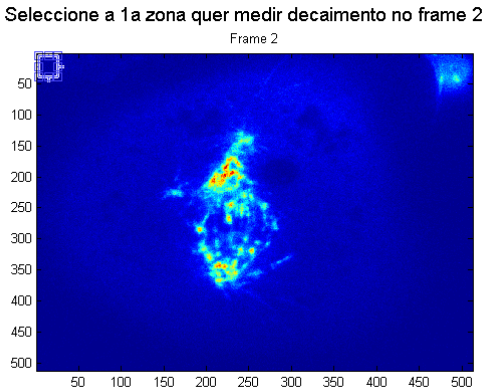

Figura 4.54: Região escura do filme das células *Drosophila* S2.

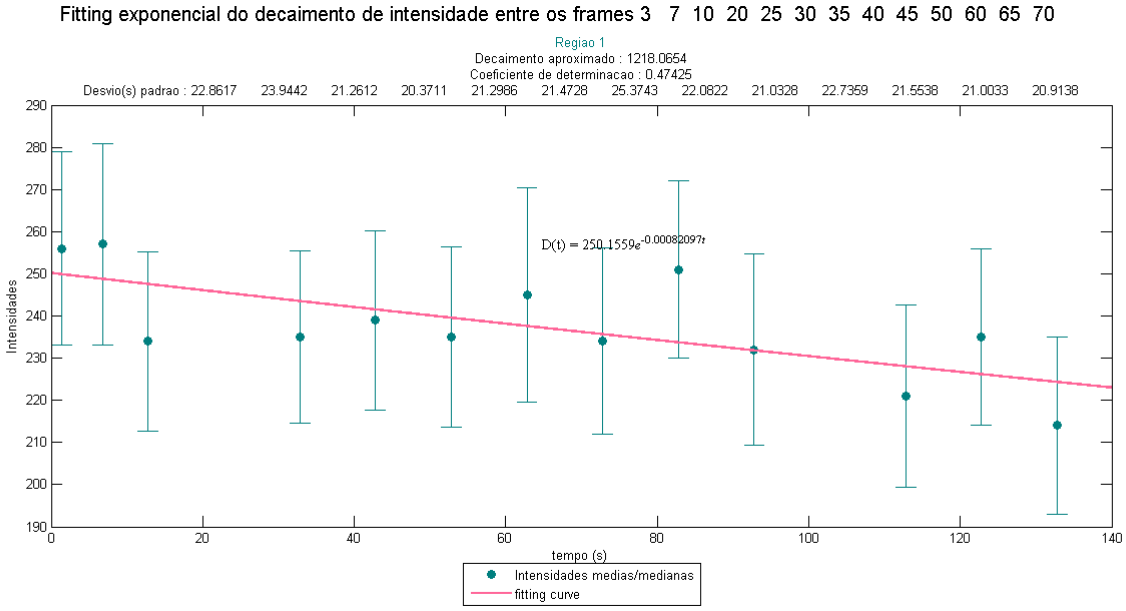

Figura 4.55: Fit exponencial para a região escura.

Como seria de esperar, as intensidades nesta zona dos frames decaem muito mais lentamente. O valor de decaimento encontrado para estes valores foi $1218.1s$, e o coeficiente de determinação foi 0.47425, o que não é muito bom.
A função de decaimento é $250.1559 \exp -\frac{1}{1218.0654}$.

Capítulo 5

Conclusões

Os objetivos propostos de construção de algoritmos fiáveis de estimação do fluxo nos microtúbulos e de decaimento da intensidade das imagens durante a metafase em células visualizadas com a técnica Induced Speckle Imaging foram atingidos.
Para utilizar o programa de cálculo do fluxo é necessário estabelecer as invariâncias de rotação e translação nos frames dos filmes. Estas são as principais condicionantes do desempenho do algoritmo. Sem uma boa marcação de pontos de estabilidade, o método não pode ser aplicado.
Os resultados obtidos para o cálculo do fluxo basearam-se na hipótese de forma super-elipsoidal do fuso mitótico em metafase, o que se revelou bastante ajustado. A ferramenta MarcarPolos.m criada permite eliminar significativamente as variações de rotação e translação das imagens dos fusos.
Foi adotada uma abordagem original de cálculo do fluxo baseada na estratégia de Block Matching. O algoritmo criado diferencia-se da estratégia de Block Matching na medida em que permite contabilizar toda a informação de sequências de frames na estimação do fluxo de intensidades ao longo do tempo.
A função DAM revelou ser um operador de deteção espacial mais robusto, exato e preciso do que a correlação generalizada na grande maioria das situações controladas.
Os algoritmos criados para quantificação de fluxo e o algoritmo para quantificação de decaimento têm um desempenho bastante bom em situações de teste controladas. O algoritmo do decaimento é eficaz na deteção de decaimento para filmes com amplitude de ruído de 50%, e o algoritmo de cálculo do fluxo é eficaz na deteção de fluxo, mesmo em dados com amplitude de ruído superiores à amplitude de sinal.
Os resultados obtidos para o fluxo no filme das células *Drosophila* S2 indicam fluxo, em média, $0.2px/s = 0.9\mu m/min$, no hemisfério superior do fuso, e $0.13px/s = 0.59\mu m/min$ no hemisfério inferior. Foram avaliadas várias sequências de frames distintas e as velocidades encontradas foram muito próximas destes valores para essas sequências. Estes valores estão concordantes com os valores estimados previamente, com métodos desenvolvidos anteriormente pela equipa de parceria. Uma grande parte dos valores obtidos de velocidade foi nulo, o que é um bom indicador do estabelecimento de invariâncias eficaz.
Contudo, pretendia-se diferenciar o fluxo nas regiões mais próximas do equador do fuso do fluxo das regiões mais próximas dos polos. Isto é possível utilizando estatísticas das velocidades de vários *speckles* em regiões homólogas do fuso mitótico, se possível em vários filmes.
Os valores de decaimento encontrados resultam da ação combinada do mecanismo de turnover e da perda de fluorescência.
No filme das células *Drosopohila* S2, avaliou-se o decaimento em três regiões distintas. O decaimento encontrado para um *speckle* da parte superior do fuso, nos primeiros 25 frames, foi

de aproximadamente, $160s$. Nos outros dois *speckles* onde se avaliou decaimento, nos primeiros 60 frames, o decaimento médio foi de $240s$. Os resultados foram obtidos com coeficientes de determinação médio das regressões encontradas (> 0.7).

Capítulo 6

Trabalho Futuro

Devido à grande influência que a eliminação de invariâncias tem sobre o sucesso do método de cálculo do fluxo, todas as medidas no sentido de eliminar invariâncias dos filmes são relevantes para a precisão e exatidão dos resultados.
De forma a obter uma dependência espacial dos fluxos, uma possível análise seria fazer estatísticas das velocidades de vários *speckles* em regiões homólogas do fuso mitótico.
O método do cálculo de decaimento pode ser aprofundado no sentido de produzir estimativas para o mecanismo de turnover. Por exemplo, seria interessante comparar valores de decaimento nos filmes em zonas escuras com decaimentos em zonas de *speckles*, para obter estimativas do turnover.

Referências

Bradski, G. and Kaehler, A. (2008). *Learning Open CV*. O'Reilly Media, Inc.

Civelekoglu-Scholey, G. and Scholey, J. (2010). Mitotic force generators and chromosome segregation. *Cellular and Molecular Life Sciences*, 67:2231–2250. 10.1007/s00018-010-0326-6.

Fredricks, G. A. and Nelsen, R. B. (2007). On the relationship between spearman's rho and kendall's tau for pairs of continuous random variables. *Journal of Statistical Planning and Inference*, 137(7):2143 – 2150.

Gibbons, J. D. (1993). *Non parametric measures of association*. 07-091. Sarge University Paper series on Quantitative Applications in the Social Sciences, Newbury Park, CA: Sage.

MATLAB (2010). version 7.10.0 (r2010a).

Matos, I., Pereira, A. J., Lince-Faria, M., Cameron, L. A., Salmon, E. D., and Maiato, H. (2009). Synchronizing chromosome segregation by flux-dependent force equalization at kinetochores. *The Journal of Cell Biology*, 186(1):11–26.

Miyamoto, D. T., Perlman, Z. E., Burbank, K. S., Groen, A. C., and Mitchison, T. J. (2004). The kinesin eg5 drives poleward microtubule flux in xenopus laevis egg extract spindles. *The Journal of Cell Biology*, 167(5):813–818.

Pereira, A. (2011). *Biophysical analysis of the metaphase mitotic spindle*. PhD thesis, Faculdade de Ciências da Universidade do Porto.

Pereira, A. J. and Maiato, H. (2010). Improved kymography tools and its applications to mitosis. *Methods*, 51(2):214 – 219.

Rieder, C. (2006). Mitosis. In *Cells*, textbook 10, pages 438–487. Jones and Bartlett.

Yang, G., Cameron, L. A., Maddox, P. S., Salmon, E. D., and Danuser, G. (2008). Regional variation of microtubule flux reveals microtubule organization in the metaphase meiotic spindle. *The Journal of Cell Biology*, 182(4):631–639.

Apêndice A

Descrição detalhada do código Matlab CalcularFluxo.m:

1. Determinar a dimensão do cálculo do fluxo:

Inicialmente determinamos se o *script* vai ser executado para o cálculo do fluxo a 1 dimensão ou duas.

```
% ------------------------------------------------------------------------
% Dimensão do fluxo:
% 1 = calculo do fluxo a 1D
% 2 = calculo do fluxo a 2D
CalculoFluxo_DIM =1;
% ------------------------------------------------------------------------
```

2. Seleção de um filme a analisar:

Através do programa LeituraFramesPolosMiusTempo.m, escolhe-se o filme para o qual se vai determinar o fluxo. De seguida pode-se visualizar o filme que foi escolhido, para selecionar apenas uma sequência de frames (consecutivos) que se queira analisar.
Aparece no visor os ficheiros de polos disponíveis para este filme. Note-se que mesmo que se pretenda determinar fluxo a 2D, é necessário ler o ficheiro de polos, pois o fluxo determina-se para as imagens rodadas e centradas no fusos que aparecem nos frames.
Para indicar a sequência basta indicar o primeiro frame da sequência, Prim_ Frame, e o último, Ultimo_ Frame, quando indicado no visor.
Selecionamos por fim o eixo temporal dos filme em questão.
Depois de correr esta função, gera-se o vector com o índice dos frames que vão ser analisados.
Por exemplo, caso o primeiro frame selecionado seja o 2º frame do filme, e o último frame seja o 10º frame, Frames = [2,3,4,5,...,10].

```
% ------------------------------------------------------------------------
% Leitura dos frames, polos, mius e EixoTemporal dos frames
[Prim_Frame Ultimo_frame file fpolos mius EixoTemporal EixoTempos_Frames] =...
 LeituraFramesPolosMiusTempos( 0, 1, 1);

Frames = Prim_Frame:Ultimo_frame;
% ------------------------------------------------------------------------
```

3. Estabelecer a dimensão das janelas de trabalho que se vão utilizar:

As janelas percorridas devem ser ajustadas consoante as medidas das regiões onde se queira avaliar o fluxo, por exemplo determinados *speckles* das sequências do filme em questão. Para a maioria dos

filmes disponibilizados, o ajuste Altura=13 pixeis, Largura=10, foi adequado. As janelas vão ter dimensão $(2 \cdot L2+1, 2 \cdot K2+1)$, para que ao correr todas as linhas e colunas se possam estabelecer janelas centradas na linha e coluna exata do momento dos ciclos.

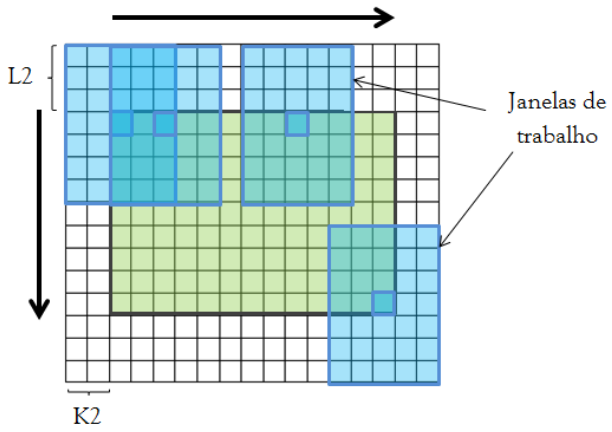

Figura A.1: Ilustração dos blocos/janelas de trabalho

```
% ------------------------------------------------------------------------
% Altura            : altura das matrizes de trabalho [px]
% Largura           : largura das matrizes de trabalho [px]

Altura=13;
Largura=10;
L2 = round(Altura/2);
K2 = round(Largura/2);
% ------------------------------------------------------------------------
```

4. Aplicação do mapa conformacional

A sequência de frames vai ser guardada numa estrutura de célula-linha, a célula pilha. A sequência de imagens limitadas aos fusos e a sequência dos mapas conformacionais destas vão ser guardados quer em estrutura de célula quer em estrutura de matriz 3D. Isto porque estas imagens não têm todas as mesmas dimensões (a marcação dos polos é feita frame-a-frame, e portanto o "corte" pelo retângulo dos polos varia ao longo dos frames). Depois de termos todas estas imagens então redimensionamo-las pela dimensão da primeira da sequência respetiva.

```
% ------------------------------------------------------------------------
pilha = cell(1,size(Frames,2));
%pilha=zeros(size(file(1).data,1),size(file(1).data,2),size(Frames,2));
ConfMap_pilha_cell = cell(1,size(Frames,2));
novas_imagens_cell = cell(1,size(Frames,2));
% ------------------------------------------------------------------------
```

No caso de não se ter marcado os polos para o primeiro frame do filme e este esteja a ser utilizado, pode-se utilizar os comandos comentados e comentar "contar=1;".
Para obter as imagens limitadas aos fusos, novas_ imagens, e as imagens dos mapas conformacionais, ConfMap, utiliza-se a sub-rotina ConfMapping.m. Esta rotina aplica-se a cada frame individualmente, tem como input a matriz do frame, os polos e o μ marcados para o frame previamente.
As imagens são agora redimensionadas. As novas_ imagens terão todas a dimensão da primeira nova_ imagem gerada, e o mesmo acontece para os ConfMap_ pilha.

Caso se pretenda fazer análise do fluxo a 2D, as imagens de trabalho não serão as dos mapas conformacionais mas as imagens limitadas aos fusos, as novas_ imagens.

```
% ------------------------------------------------------------------------
% if Frames(1)==1
%     ConfMap_pilha_cell{1}=file(Frames(1)).data;
%     pilha(:,:,1)=file(Frames(1)).data;
%     novas_imagens_cell{1} = file(Frames(1)).data;
%     contar = 2;
% else
%     contar=1;
% end
contar=1;
for i=contar:size(Frames,2)
    miu = mius(Frames(i));
    pilha{i}=file(Frames(i)).data;
    %pilha(:,:,i)= imresize(pilha(:,:,i),...
    [size(pilha(:,:,contar),1) size(pilha(:,:,contar),2)]);
    [novas_imagens_cell{i} ConfMap_pilha_cell{i}] = ...
    ConfMapping( file(Frames(i)).data, fpolos(Frames(i),:,:), miu,...
     Frames(i));
    novas_imagens_cell{i} = imresize(novas_imagens_cell{i},...
    [size(novas_imagens_cell{contar},1) size(novas_imagens_cell{contar},2)]);
    ConfMap_pilha_cell{i} = imresize(ConfMap_pilha_cell{i},...
    [size(ConfMap_pilha_cell{contar},1) size(ConfMap_pilha_cell{contar},2)]);
    if i==contar
        novas_imagens = zeros(size(novas_imagens_cell{i},1),...
        size(novas_imagens_cell{i},2),size(Frames,2));
        ConfMap_pilha = zeros(size(ConfMap_pilha_cell{i},1),...
        size(ConfMap_pilha_cell{i},2),size(Frames,2));
        novas_imagens(:,:,i)=novas_imagens_cell{i};
        ConfMap_pilha(:,:,i)=ConfMap_pilha_cell{i};
    else
        novas_imagens(:,:,i)=novas_imagens_cell{i};
        ConfMap_pilha(:,:,i)=ConfMap_pilha_cell{i};
    end
end
% Caso calculo do fluxo seja a duas dimensoes:
if (CalculoFluxo_DIM==2)==1
    ConfMap_pilha = novas_imagens;
end
% ------------------------------------------------------------------------
```

5. Visualização do filme do fuso após aplicação do mapa conformacional às imagens

O vídeo é produzido através do comando implay, e este só está preparado para dados uint8. As imagens são adquiridas a 12bits e gravadas em *double*. Os índices máximos são portanto 4096. É aplicado um fator de escala a estas imagens que as converte para uint8.

```
% ------------------------------------------------------------------------
img_stack = zeros(size(ConfMap_pilha_cell{1},1),size(ConfMap_pilha_cell{1},2),...
size(ConfMap_pilha_cell,2));
for filmeframe=1:size(ConfMap_pilha_cell,2)
    img_stack(:,:,filmeframe) = ConfMap_pilha_cell{filmeframe};
end
```

```
files_uint8 = uint8( img_stack / 4096 * 256 );
implay( files_uint8, numel(ConfMap_pilha)-1 );
pause;
% ------------------------------------------------------------------------
```

6. Excluir frames da primeira sequência selecionada

Surge no ecrã a questão "Quer excluir frames? (sim ou não)". Caso respondamos afirmativo surge outra questão "Quais os frames que pretende excluir?". Ao introduzirmos a resposta a esta questão estamos a criar a variável frames_ nao_ usar, que será um vector-linha com os índices dos frames que queremos excluir. Aqui podemos indicar frames que não constem da primeira sequência de frames selecionada, que é feito um ajuste para considerar da lista de frames a excluir agora introduzida apenas aqueles que constam da primeira sequência de frames.
De seguida é feito outro ajuste relativo ao índice dos frames. Por exemplo, se a lista inicial for a indicada no exemplo anterior, [2,3,4,5,6,7,8,9,10] e pretendermos excluir desta lista os frames [5,7,9], então frames_ nao_ usar = [5,7,9]. Faz-se um ajuste a este vector para ele conter não os frames a descartar mas as entradas do vector Frame que serão descartadas. Assim vem frames_ nao_ usar = [4,6,8].
Após este ajuste, são então retirados ao vector Frames os frames a excluir. No exemplo, no final teríamos um vector frames novo, Frames = [2,3,4,6,8,10].

```
% ------------------------------------------------------------------------
exclusao = -1;
display(' ')
while (exclusao ==1 || exclusao ==0)==0
exclusao = input('Quer excluir frames? [0=nao, 1=sim]:');
end
if exclusao ==1
    frames_nao_usar = input('Quais os frames que quer excluir? :');
    for jj = 1:size(frames_nao_usar,2)
        if (frames_nao_usar(jj)<Prim_Frame || frames_nao_usar(jj)>Ultimo_frame)==1
            frames_nao_usar(jj)=0;
        end
    end
    frames_nao_usar = frames_nao_usar - Prim_Frame +1;
    retirar = find(frames_nao_usar ==0);
    frames_nao_usar(retirar) = [];
else
    frames_nao_usar = [];
end
Frames(frames_nao_usar)=[];
% ------------------------------------------------------------------------
```

7. Qual o número de frames a avaliar em cada janela de cálculo?

É possível calcular o fluxo em várias sub-sequências de frames mais pequenas e uniformes contidas na sequência de frames inicial escolhida. No exemplo seguido, se escolhêssemos NumFramesNaJanela = 6, então iríamos calcular o fluxo ao longo de todos os frames, [2,3,4,6,8,10]. Se escolhêssemos NumFramesNaJanela = 4, iríamos calcular o fluxo para todas as sequências de apenas 4 frames consecutivos, ou seja, [2,3,4,6], [3,4,6,8], [4,6,8,10].

```
% ------------------------------------------------------------------------
display('Qual o numero de frames a considerar em cada janela de calculo?')
```

```
NumFramesNaJanela = input('Numero de frames : ');
% ------------------------------------------------------------------------
```

· Processamento das sequências a avaliar

A variável ParesDeFrames consiste dos pares (1º, Ultimo) frame de cada sequência a avaliar. Por exemplo, no caso de estarmos a avaliar a sequência anterior, Frames = [2,3,4,6,8,10], e caso tenhamos escolhido NFrames = 4, ou seja, as sequências a avaliar serem [2,3,4,6], [3,4,6,8], [4,6,8,10], então a matriz é ParesDeFrames = [2,6 ; 3,8 ; 4,10].

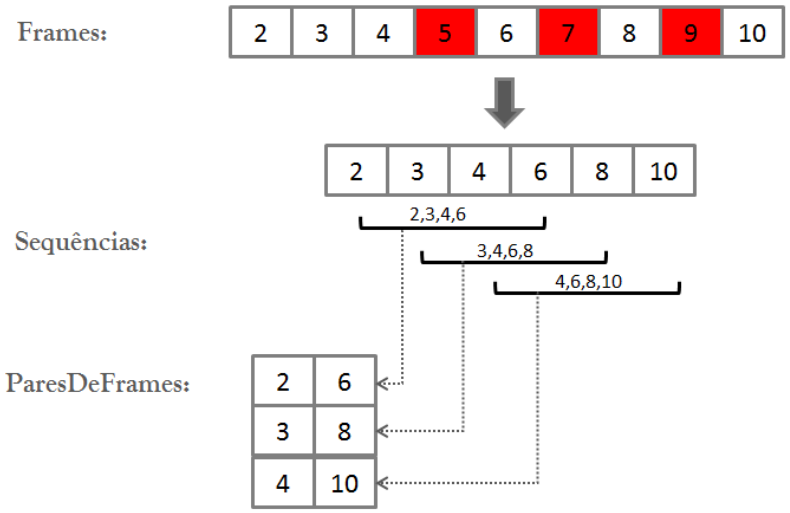

Figura A.2: Exemplo de como se obtêm as variáveis f1 e f2

A variável EixoTempos_ Frames contém os tempos de registo de cada frame da matriz Frames, enquanto que a matriz Eixo_ temporal_ pares tem a dimensão da matriz ParesDeFrames e contém os tempos de registos dos frames das respetivas entradas de ParesDeFrames.

```
% ------------------------------------------------------------------------
% Pares de Frames e Eixo temporal
ParesDeFrames = zeros(size(Frames,2)-(NumFramesNaJanela - 1),2);
for i=1:size(ParesDeFrames,1)
    ParesDeFrames(i,1) = Frames(i);
    ParesDeFrames(i,2) = Frames(i+NumFramesNaJanela - 1);
end

EixoTempos_Frames(frames_nao_usar)=[];
Eixo_temporal_pares = zeros(size(Frames,2)-(NumFramesNaJanela - 1),2);
for i=1:size(ParesDeFrames,1)
    Eixo_temporal_pares(i,1) = EixoTempos_Frames(i);
    Eixo_temporal_pares(i,2) = EixoTempos_Frames(i + (NumFramesNaJanela - 1));
end
% ------------------------------------------------------------------------
```

8. Qual o valor de vmax e de dv que quer utilizar?

A gama de velocidades que iremos avaliar será [-vmax : dv : vmax]. O valor de vmax e de dv afectam o tempo de execucao do algoritmo. Quanto mais amplo o vmax e menor o intervalos dv,

mais tempo o algoritmo leva a correr. Contudo, quanto mais amplo o valor de v mais resolução tem o mapa de velocidades e mais significância terá a função de confiança deste mapa. Ao especificar estes valores devemos ter em conta estas implicações.

```
% ------------------------------------------------------------------------
vmax=-1;dv=-1;
while vmax<=0
    display('Qual o valor de vmax e de dv que quer utilizar?')
    vmax = input('vmax :');
end
while dv<0;
    dv = input('dv : ');
end
display(' ')
% ------------------------------------------------------------------------
```

9. Selecionar a função a utilizar para calcular os deslocamentos

Escolhemos agora a função a utilizar para aplicar o algoritmo de estratégia semelhante à do Block-Matching. A cada pixel nas entradas (n,m) do primeiro frame de cada sequência, vamos atribuir um deslocamento tendo em conta as possíveis "posições do pixel" nos frames seguintes (figura 2.9). As funções de BlockMatching estão implementadas nas sub-rotinas CalcFluxo_ NFrames_ 1D.m e CalcFluxo_ NFrames_ 2D.m.
As opções de funções surgem no visor e são as seguintes:

1. Função de correlação generalizada adaptada; Janelas colapsadas com o máximo
2. Função de correlação generalizada adaptada; Janelas colapsadas com a mediana
3. Função de diferenças absolutas; Janelas colapsadas com o máximo
4. Função de diferenças absolutas; Janelas colapsadas com a mediana
5. Função de correlação generalizada adaptada 2D
6. Função de diferenças absolutas 2D

Ao selecionarmos uma das primeiras 4 opções, estamos a indicar a utilização da rotina CalcFluxo_ NFrames_ 1D.m, e se indicarmos as respostas 5 ou 6, aplicar-se-á a rotina CalcFluxo_ NFrames_ 2D.m.
A variável correspondente à função de BlockMatching é a variável Algoritmo, que pode significar aplicar função de Correlação ou função de DAM. Se dermos como resposta 1,2 ou 5, a variável Algoritmo assume o valor 1, o que significa aplicação da correlação generalizada. Se dermos a resposta 2,3 ou 4, Algoritmo = 2, correspondendo à função de DAM.
A variável que indica a função utilizada para colapsar as janelas selecionadas é a variável Funcao. Se for dada a resposta 1 ou 3 as janelas são colapsadas com o máximo, se for dada resposta 2 ou 4 utiliza-se a mediana para as colapsar.

```
% ------------------------------------------------------------------------
escolha=-1;
while (escolha==1 || escolha==2 || escolha==3 || escolha==4 || escolha==5...
 || escolha==6)==0
    display(' ')
```

```
    display('Escolha a funcao que quer utilizar para calcular')
    display('os deslocamentos dos pixeis entre frames:')
    display(' ')
    display('Correlacao colapsando janelas com o maximo:         1')
    display('Correlacao colapsando janelas com a mediana:        2')
    display('Funcao de DAM colapsando janelas com o maximo:         3')
    display('Funcao de DAM colapsando janelas com a mediana:        4')
    display('Funcao de correlacao 2D:                            5')
    display('Funcao de DAM 2D:                                      6')
    escolha = input('Funcao: ');
end
switch escolha
    case 1
        titulo = 'Correlacao / Janelas colapsadas com Maximo';
    case 2
        titulo = 'Correlacao / Janelas colapsadas com Mediana';
    case 3
        titulo = 'Funcao DAM / Janelas colapsadas com Maximo';
    case 4
        titulo = 'Funcao DAM / Janelas colapsadas com Mediana';
    case 5
        titulo = 'Funcao DAM / Janelas colapsadas com Mediana';
    case 6
        titulo = 'Funcao DAM / Janelas colapsadas com Mediana';
end
% Funcao = 0 (mediana)      Algortimo = 1 (correlacao)
% Funcao = 1 (max)          Algoritmo = 2 (DAM)
Funcao=0;Algoritmo=1;
if (escolha==1 || escolha==3)==1
    Funcao=1;
end
if (escolha==3 || escolha==4 || escolha==6)==1
    Algoritmo=2;
end
% ------------------------------------------------------------------------
```

10. Ciclo que corre todas as sequências de frames indicadas em 7

A matriz M_vel é a matriz onde será guardada a informação das velocidades para cada pixel em cada sequência analisada, enquanto que a matriz M_pesos é uma matriz dos valores de τ-Kedal para os respetivos pixeis e sequências avaliados.
Para cada sequência analisada é devolvido apenas um valor médio de decaimento e não valores discriminados para cada pixel, como acontece com os valores de fluxo calculados.

```
% ------------------------------------------------------------------------
frame = ConfMap_pilha(:,:,1);
M_vel = nan * ones(size(frame,1),size(frame,2),size(ParesDeFrames,1),2);
M_pesos = nan * ones(size(frame,1),size(frame,2),size(ParesDeFrames,1),1);
% ------------------------------------------------------------------------
```

O ciclo percorre cada sequência de frames a analizar determinada em 7.
As variáveis f1 e f2 indicam as duas entradas dos conjunto de frames inicial que fazem a sequência de frames a analisar.

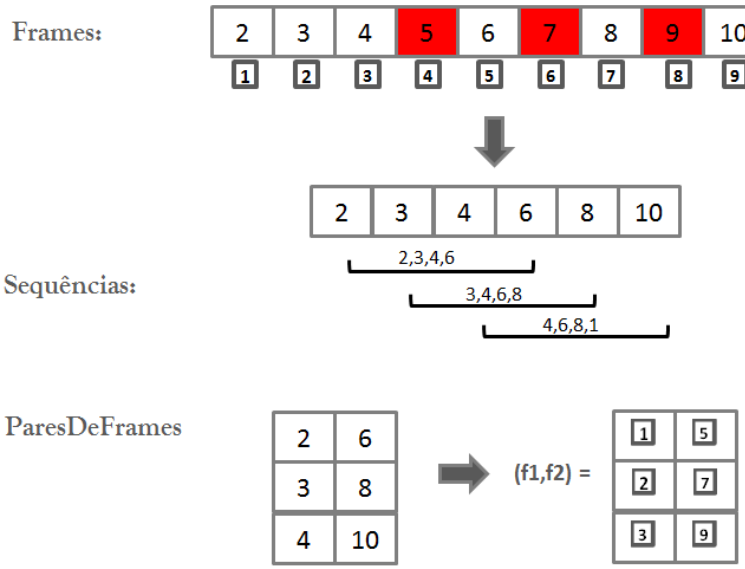

Figura A.3: Exemplo de como se obtêm as variáveis f1 e f2

A variável t0 é o tempo de registo do frame correspondente f1, enquanto que a variável t1 é o tempo de registo do frame correspondente a f2. A diferença de ambos é o intervalo de tempo entre os dois frames. O valor $d_{pos} = |[vmax(t1 - t0)]|$ indica-nos o deslocamento absoluto entre os frames f1 e f2 caso haja movimento à velocidade máxima admitida.
Este parâmetro será utilizado no cálculo do movimento de um pixel ao longo dos frames. Uma vez que para calcular o movimento do pixel consideramos uma janela em volta dele e supomos que esta se movimenta a determinada velocidade v, há um conjunto de pixeis ainda mais reduzido que pode ser avaliado para todas as velocidades para além daqueles em que a janela cabe no primeiro frame. As janelas têm também de "caber"totalmente dentro dos restantes frames quando estamos a avaliar as velocidades.
Ou seja, para além dos pixeis descartados automaticamente para as janelas no primeiro frame estarem totalmente dentro deste, há também a condição de que as janelas têm de caber na totalidade em todos os frames seguintes admitindo o deslocamento que elas efetuam para as velocidades a testar.

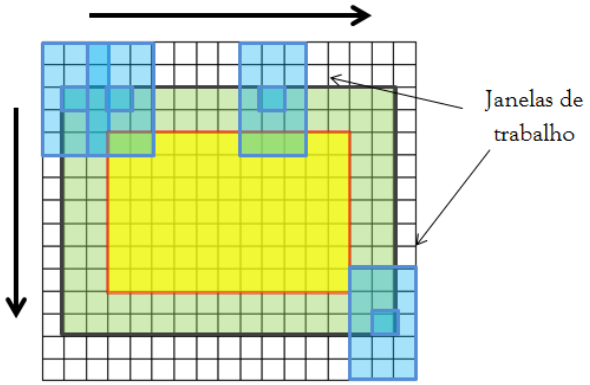

Figura A.4: Ilustração de regiões de uma imagem limitadas pela velocidade máxima onde é possível calcular fluxo. A verde região limitada pela dimensão das janelas e a amarelo região limitada pela velocidade máxima e número de frames a utilizar.

Dadas as características da análise de cada janela é possível distribuir os processos por vários processadores, permitindo que o algoritmo corra em paralelo, acelerando assim a análise. No Matlab, comando matlabpool permite a funcionalidade completa de recursos em simultâneo, utilizando computação paralela.
O comando é indicado fora do ciclo para iniciar a divisão das tarefas, e as tarefas que correram em paralelo serão as do cilo 12.

```
% ------------------------------------------------------------------------
matlabpool
tic
for i=1:size(ParesDeFrames,1);

    f1 = ParesDeFrames(i,1)-Prim_Frame+1;
    f2 = ParesDeFrames(i,2)-Prim_Frame+1;

    % Deslocamentos supondo a velocidade maxima:
    t0 = Eixo_temporal_pares(i,1);
    t1 = Eixo_temporal_pares(i,2);
    d_pos = abs( round( vmax * (t1 - t0) ) );

    % Para vermos em que sequencia vai enquanto o programa corre:
    display(' ')
    display(['Calculo das velocidades medias do frame ',...
     num2str(ParesDeFrames(i,1)), ' para o frame ',num2str(ParesDeFrames(i,2))])
% ------------------------------------------------------------------------
```

11. e 12. Ciclo que percorre os pixeis do primeiro frame da sequência para determinar a velocidade

A variável n indica a linha percorrida e variável m a coluna. As linhas e as colunas percorridas são limitadas pela dimensão da janela admitida (figura A.1), por isso têm-se as condições (*A) e (*B).
Há dois ciclos diferentes, um para o caso de estarmos a medir o fluxo a 2D, outra para a medição a 1D.
As condições (*C1) e (*C2) dizem respeito ao número de linhas e de colunas que é necessário de margem para ser possível admitir janelas completas até ao último frame da sequência (figura A.4), como explicado em 10.
A condição (*C1) é do ciclo que efetua a medição a 2D, e por isso esta condição sobre as linhas e colunas tem em conta as velocidades máximas admitidas quer para o deslocamento horizontal quer para o deslocamento vertical dos blocos de trabalho.

```
% ------------------------------------------------------------------------
    % Ciclo que vai correr todos os pixeis:
(*A)    for n = 1+L2:size(frame,1)-L2 % Linhas

        display(['Linha das imagens a ser processada : ',num2str(n)])
        display([' Total : ',num2str(size(frame,1))]);

        if (escolha==5 || escolha==6)==1

(*B)        parfor m = 1+K2:size(frame,2)-K2   % Colunas

(*C1)           if (n-d_pos-L2 < 1 || n+d_pos+L2 > size(frame,1) || ...
                m-d_pos-K2 <1 || m+d_pos+K2 > size(frame,2) )==0
```

```
                    [fluxo, S] = CalcFluxo_NFrames_2D( Algoritmo,[n,m], ...
                    ConfMap_pilha, [L2,K2], [f1,f2], ParesDeFrames(i,:),...
                    frames_nao_usar, EixoTemporal, vmax, dv );
                    M_vel(n,m,i,:) = fluxo;
                    M_pesos(n,m,i) = S;
                end
            end
        else
(*B)        parfor m = 1+K2:size(frame,2)-K2      % Colunas
(*C2)           if (n-d_pos-L2 < 1 || n+d_pos+L2 > size(frame,1))==0

                    [fluxo, S] = CalcFluxo_NFrames_1D( Funcao,Algoritmo,...
                    [n,m], ConfMap_pilha, [L2,K2], [f1,f2], ParesDeFrames(i,:),...
                    frames_nao_usar, EixoTemporal, vmax, dv );
                    M_vel(n,m,i,1) = fluxo;
                    M_pesos(n,m,i) = S;
                end
            end
        end
    end
% --------------------------------------------------------------------------
```

Apêndice B

Descrição detalhada do código Matlab DecaimentoSemiAutomatico.m

1. Leitura de frames e eixo temporal:

Através do programa LeituraFramesPolosMiusTempo.m, escolhe-se o filme para o qual se vai determinar decaimento e o ficheiro temporal respetivo. De seguida pode-se visualizar o filme que foi escolhido para selecionar a sequência de frames que se quer analizar. Para indicar a sequência basta indicar o primeiro frame da sequência, Prim_ Frame, e o último, Ultimo_ Frame. Depois de correr esta função, gera-se o vector com o índice dos frames que vão ser analisados.
Por exemplo, caso o primeiro frame seja o 2º frame do filme, e o último frame seja o 50º frame, Frames = [2,3,4,5,...,50].

```
% ------------------------------------------------------------------------
[Prim_Frame Ultimo_frame , file , ~, ~, EixoTemporal ,EixoTempos_Frames] =
 LeituraFramesPolosMiusTempos( 1, 0, 1);
Frames = Prim_Frame:Ultimo_frame;
% ------------------------------------------------------------------------
```

2. Seleção da sequência de frames a analisar:

De seguida tem-se a opção de excluir frames da sequência de frames indicada, Frames, que se considerem não estar nas melhores condições. O programa produz uma pergunta, "Quer excluir frames dentro da sequencia ?" à qual respondemos "não"(0) ou "sim"(1). Caso tenhamos respondido "sim" introduzimos um vector com os frames que pretendemos utilizar exclusivamente. Por exemplo, para utilizarmos apenas os frames 2,20,30 e 50 indicaríamos [2,20,30,50].
Note-se que não se devem indicar frames que não constem da lista de frames Frames gerada anteriormente.

```
% ------------------------------------------------------------------------
exclusao = -1;
display(' ')
while (exclusao ==1 || exclusao ==0)==0
    display('Quer excluir frames dentro da sequencia ');
    exclusao = input('que seleccionou? [0=nao, 1=sim]:');
end

if exclusao ==1
    Frames = input('Quais os frames que quer utilizar? :');
end
% ------------------------------------------------------------------------
```

• Criação da pilha de imagens a utilizar:

É criada de seguida uma pilha de imagens, a matriz pilha, que contém nela as matrizes das imagens, de Prim_ Frame a Ultimo_ frame. Esta pilha inclui também as imagens que excluímos no passo anterior. A variável frames consiste dos índices de pilha que realmente pretendemos utilizar.
No exemplo anterior, pilha é uma matriz de dimensão (Nlinhas x NColunas x 6) (frames do 2º ao 7º) e frames = [1,3,5,6], que é o vector de índices da pilha que efectivamente queremos utilizar.

```
% ------------------------------------------------------------------------
pilha=zeros(size(file(1).data,1),size(file(1).data,2),size(Frames,2));
for i=1:size(Frames,2)
    pilha(:,:,i)=file(Frames(i)).data;
end
frames=Frames-Prim_Frame+1;
f1 = frames(1);
pilha_pequena= pilha;
tf=EixoTemporal;
% ------------------------------------------------------------------------
```

• Indicar o número de regiões a utilizar:

Na secção seguinte determina-se o número de regiões nas quais queremos calcular fluxo, e este é registado na variável Nregioes.
Por exemplo, se tivermos intenção de avaliar o decaimento em 3 pontos específicos, por exemplo 3 *speckles*, Nregioes = 3.
A matriz Decaimentos irá conter os decays obtidos para cada região acompanhada, e a matriz pesos conterá o coeficiente de ajustamento obtido para o decaimento calculado em cada região.
Durante o ciclo que irá ser processado posteriormente no código, irão ser produzidos histogramas das intensidades em cada uma das regiões selecionadas e também de toda a imagem original.
A variável di é o tamanho dos intervalos de intensidades que iremos contabilizar nos histogramas, e estes são uniformizados para todos os histogramas produzidos. O parâmetro xx é a gama de intensidades avaliadas, que varia desde a intensidade mínima à intensidade máxima do primeiro frame, a intervalos de di.

```
% ------------------------------------------------------------------------
display(' ')
display('Para o calculo do decaimento utilizando os mapas anteriores, ')
Nregioes = input('quantas regioes quer marcar em cada frame? :' );
Decaimentos = zeros(1,Nregioes);
pesos = zeros(1,Nregioes);
di = ( max(max(pilha(:,:,1))) - min(min(pilha(:,:,1))) )/10000;
xx = min(min(pilha(:,:,1))):di: max(max(pilha(:,:,1)));
% ------------------------------------------------------------------------
```

• Escolha de uma função para colapsar as regiões selecionadas

As janelas a selecionar serão colapsadas com a média ou a mediana.

```
% ------------------------------------------------------------------------
escolha=-1;
while (escolha==0 || escolha==1)==0
    display('Escolha uma funcao para colapsar ao longo das linhas ')
    display('as janelas das imagens originais.')
```

```
    display('Media (Resposta=0) ou Mediana (Resposta=1) ')
    escolha=input('Resposta :');
end
switch escolha
    case 0
        FuncaoColapsar = @median;
    case 1
        FuncaoColapsar = @max;
end
% ------------------------------------------------------------------------
```

• Ciclo que irá percorrer as Nregioes

Vamos selecionar em todos os frames as Nregioes. O ciclo percorre cada região individualmente. Ou seja, supondo que Nregioes = 3, vamos começar por analisar uma dessas regiões, em todos os frames. Ou seja, primeiro vamos selecionar em todos os frames uma janela em volta da primeira região de interesse.
Na matriz Medias_ Std_ Regioes vamos registar os valores médios e os desvios padrões das janelas selecionadas em cada frame.
A matriz pilha_ 1 é apenas uma matriz de referência que irá ser usada para visualizarmos em cada frame a janela selecionada.

```
% ------------------------------------------------------------------------
% Ciclo que irá percorrer as regioes seleccionadas:
for r = 1:Nregioes

    Medias_Std_Regioes = zeros( size(frames,2),2 );
    pilha_1 = pilha;
    figure(112+r)
    .
    .
    .
% ------------------------------------------------------------------------
```

• Marcação da janela onde medir o turnover

De seguida entramos no ciclo que irá percorrer todos os frames selecionados (for f=1:size(frames,2)). Em cada um (para cada f) vamos visualizar o frame respetivo e marcar a janela pretendida com o rato. O comando para esta seleção são os comandos h = imrect; regiao = wait(h).

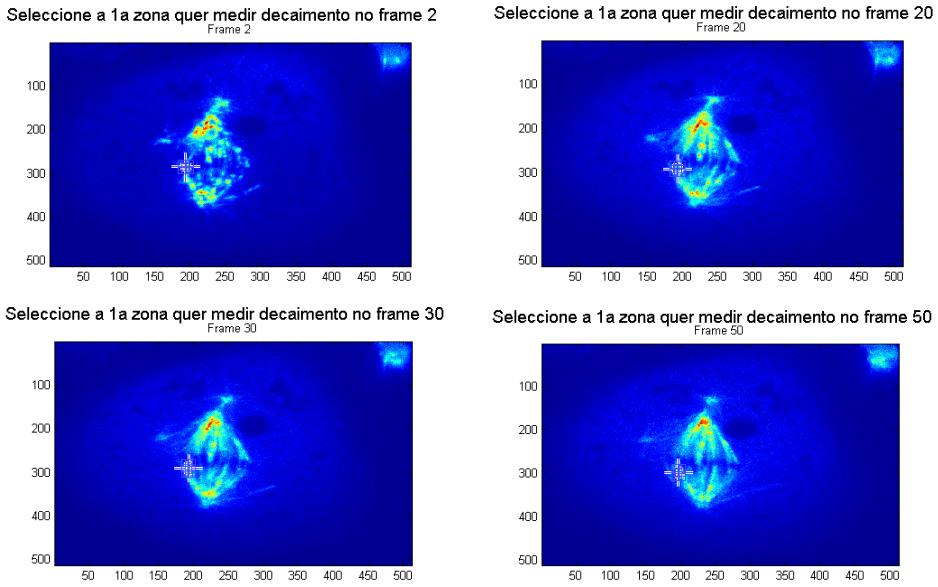

Figura B.1: Imagem Matlab do programa DecaimentoSemiAutomatico.m, seleção da primeira região para todos os frames

No exemplo seguido, visualizamos as 4 imagens da figura B.1. Após selecionar a primeira região no primeiro frame, neste caso o frame 2, não selecionamos imediatamente as regiões seguintes, apenas na iteração seguinte deste ciclo for.
A região selecionada é exportada depois de se selecionar a região. A matriz linhas_ região tem a dimensão da região selecionada, e em cada posição (i,j) tem o índice da i-ésima linha selecionada na imagem. Na ilustração seguinte a região selecionada é a região verde, e a matriz linhas região é a matriz a verde.

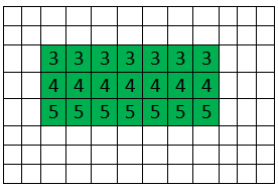

Figura B.2: Ilustração da matriz linhas região do código DecaimentoSemiAutomatico.m

Seleciona-se então na matriz do frame f o retângulo da matriz linhas_ região, e colapsam-se as suas entradas com a função escolhida, FuncaoColapsar. A média e o desvio padrão desta janela são registados na matriz Medias_ Std_ Regioes, na f-ésima linha.

```
% --------------------------------------------------------------------
    for f = 1:size(frames,2)
        figure('Name', 'Selecione as zonas onde quer medir decaimento')
        imagesc(pilha(:,:,f));
        title(['Frame ',num2str(frames(f)+Prim_Frame-1)]);
        suptitle(['Seleccione a ',num2str(r),'a zona quer medir decaimento no
         frame ',num2str(frames(f)+Prim_Frame-1)]);

        h = imrect;
        regiao = wait(h);
```

```
        regiao = round(regiao);
        linhas_regiao = (regiao(2):(regiao(2)+regiao(4)))';
        linhas_regiao = repmat(linhas_regiao,...
         size( regiao(1):(regiao(1)+regiao(3))));
        Intensidades_regiao = pilha( linhas_regiao,...
         regiao(1):(regiao(1)+regiao(3)),f);

        % vamos registar a media de intensidades na regiao, o desv padrao,
        % para todos os frames f
        media = FuncaoColapsar(FuncaoColapsar(Intensidades_regiao));
        dev_pad = std(Intensidades_regiao(1:end));
        Medias_Std_Regioes(f,:)=[media,dev_pad];
% ------------------------------------------------------------------------
```

• Representações das regiões selecionadas, dos histogramas respetivos e do histograma do fuso f

De seguida é apresentada uma imagem de 3 linhas de subplots e número de colunas igual ao número de frames a analisar. Na primeira linha são representados os frames com as regiões selecionadas nesses frames na iteração r (pilha_ 1), na segunda os histogramas das respetivas regiões selecionadas, e na terceira linha os histogramas do frame da respetiva coluna na totalidade.
Através dos comandos hold on e hold off, é possível representar todos os frames e gráficos numa única figura, a figura (112+r).

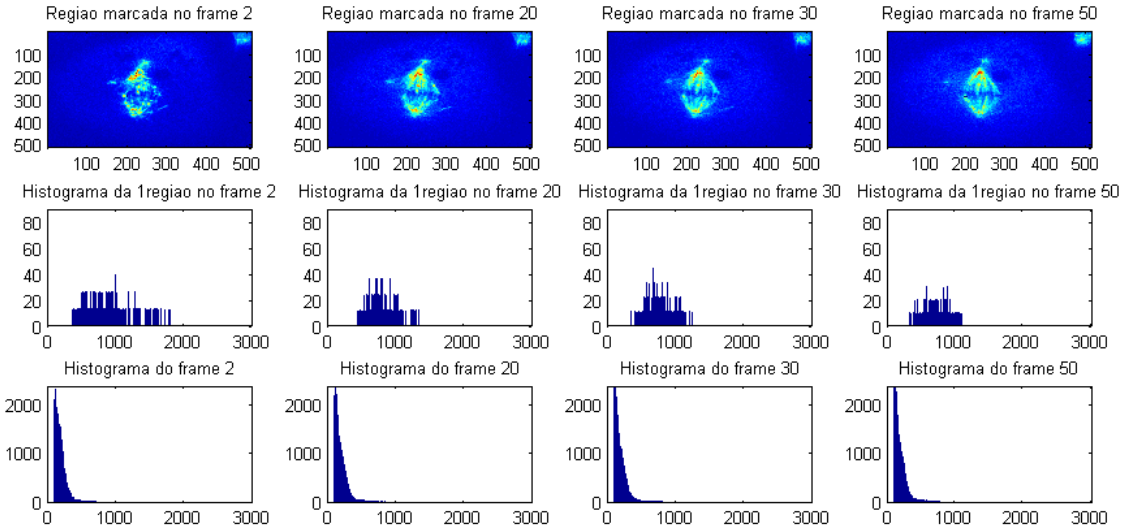

Figura B.3: Regiões selecionadas e histogramas respetivos

```
% ------------------------------------------------------------------------
        % representacao das regioes selecionadas nos frames
        figure(112+r),subplot(3,size(frames,2),f);
        pilha_1( linhas_regiao(1), regiao(1):(regiao(1)+regiao(3)), f ) = 0;
        pilha_1( linhas_regiao(end), regiao(1):(regiao(1)+regiao(3)), f ) = 0;
        pilha_1( linhas_regiao, regiao(1) , f ) = 0;
        pilha_1( linhas_regiao, regiao(1)+regiao(3) , f ) = 0;
        imagesc(pilha_1(:,:,f)); title(['Regiao marcada no frame ',...
```

```
         num2str(Frames(f))]);
        hold on

        % representacao do histograma da regiao selecionada
        figure(112+r),subplot(3,size(frames,2),f+size(frames,2));
        ar_f1=reshape(...
        Intensidades_regiao,size(Intensidades_regiao,1)*size(Intensidades_regiao,2),1);
        y_f1=hist(ar_f1,xx);
        if (frames(f)==frames(1) && r==1)==1
            yreference1 = max(y_f1)+50;
        end
        hist(ar_f1,xx);
        title(['Histograma da ',num2str(r)',...
         'regiao no frame ' num2str(frames(f)+Prim_Frame-1)]);
        ylim([0 yreference1]);
        hold on

        % representacao do histograma do frame todo
        figure(112+r),subplot(3,size(frames,2),f+size(frames,2)*2);
        % ar_f2 e uma lista das entradas do frame de forma a que possa ser
        % feito o histograma
        ar_f2=reshape( pilha(:,:,f),size(pilha(:,:,f),1)*size(pilha(:,:,f),2),1);
        y_f2=hist(ar_f2,xx);
        if (frames(f)==frames(1) && r==1)==1 % para so fazer uma vez referencia
            yreference2 = max(y_f2)+50;
        end
        hist(ar_f2,xx);title(['Histograma do frame ', num2str(frames(f))]);
        ylim([0 yreference2]);
        %plot(y_f1,x,'o', 'MarkerSize',10,'MarkerFaceColor','b');
        %plot(y_f2,x,'o', 'MarkerSize',10,'MarkerFaceColor','b');
        hold off

   end
% --------------------------------------------------------------------------
```

• *Fitting* exponencial

A variável t_ ajustado é a variável dos tempos de registo dos frames selecionados.
O *fitting* exponencial das intensidades médias é aplicado através do comando fit. O parâmetro gof2.rsquare retornado no ajuste e é o coeficiente de determinação da regressão efetuada. O coeficiente $1/c2.b = 1/rsquare$ é o decaimento verificado segundo a lei exponencial adaptada aos dados.
Estes serão guardados nos vectores pesos e Decaimentos, respetivamente.

```
% --------------------------------------------------------------------------
    % t_ajustado: tempos apenas dos frames que selecionamos
    t_ajustado = EixoTemporal(Frames);

    x_data = t_ajustado;
    y_data = Medias_Std_Regioes(:,1);

    fitoptions('Method','NonlinearLeastSquares',...
        'Lower',[0,0],...
        'Upper',[Inf,max(x_data)],...
        'Startpoint',[1 1]);
```

```
    ftip = fittype('Exp1');
    [c2,gof2] = fit(x_data,y_data,ftip);
    display(c2);display(gof2);
    Decaimentos(r) = -1/c2.b;
    display(Decaimentos);
    pesos(r) = gof2.rsquare;
    display(pesos);
% ------------------------------------------------------------------------
```

• Representação gráfica do *fitting* exponencial
Para produzirmos todos os gráficos dos *fittings* gerados para cada região numa mesma figura, geramos a figura Plots apenas da primeira vez que corremos o ciclo das regiões.
Ao longo das colunas da figura Plots temos os vários *subplots* dos *fittings* exponenciais obtidos para cada região.
O comando error bar permite desenhar no gráfico da curva os pontos observados, ou seja, as médias ou medianas das janelas selecionadas e a barra de desvios padrões para cada janela.
Plot(c2) processa o gráfico da curva exponencial obtida pelo comando fit anteriormente.

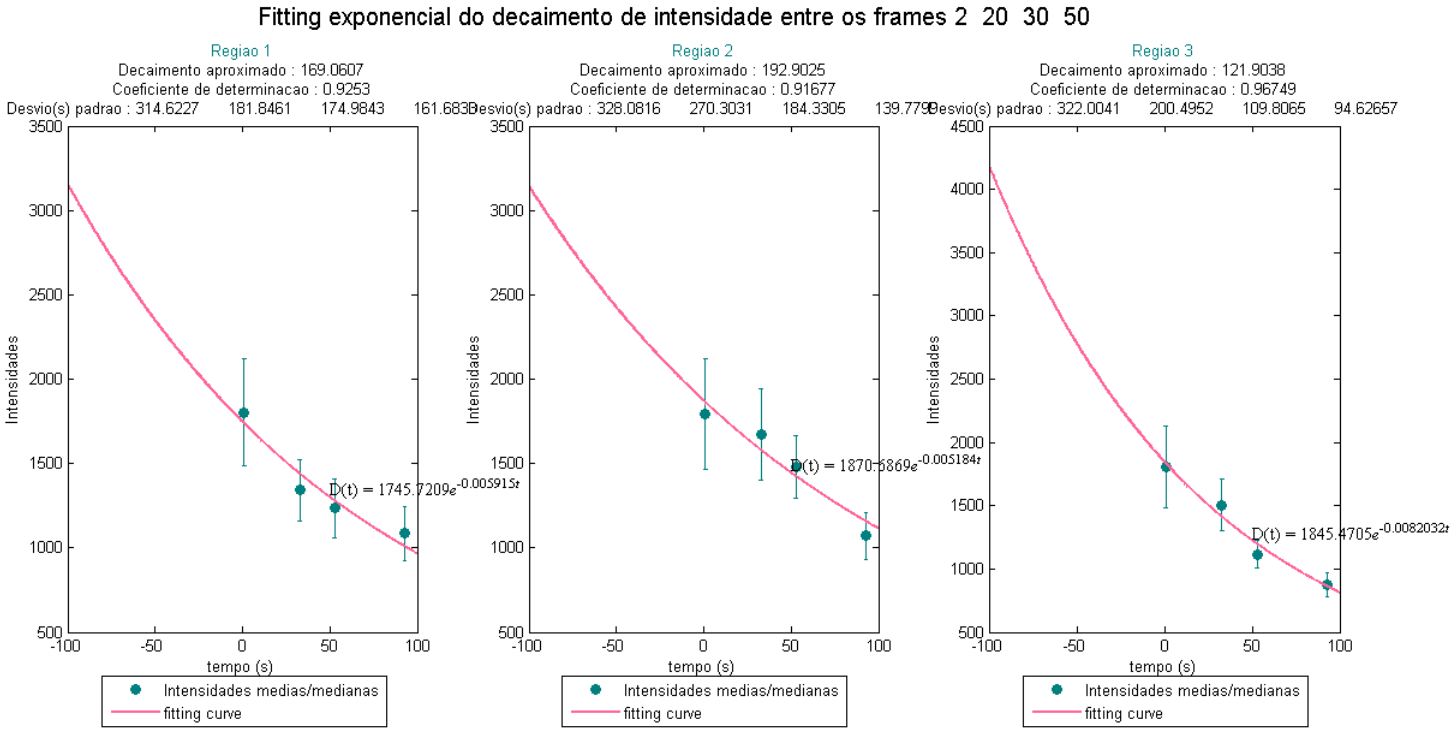

Figura B.4: Exemplo de figura com diferentes *fittings* exponenciais para cada região.

```
% ------------------------------------------------------------------------
    % Plot do fitting:
    if r==1
        Plots = figure;
        %suptitle('Fitting Exponencial de Turnover')
    end
    figure(Plots)
    Subplot = subplot(1,Nregioes,r);
```

```
    ax=get(Subplot,'Position');
    ax(2)=ax(2)-0.02; % ajuste local da posicao do subplot
    ax(4)=ax(4)-0.08; % ajuste local da posicao do subplot
    set(Subplot,'Position',ax)
    % set(p,'Color',[1,0.4,0.6],'LineWidth',2);
    errorbar(x_data,y_data,Medias_Std_Regioes(:,2),'ro','LineWidth',0.8,...
    'MarkerFaceColor',[0,0.5,0.5],'Color',[0,0.5,0.5]);
    hold on;
    Plot2 = plot(c2);
    set(Plot2,'Color',[1,0.4,0.6],'LineWidth',2);
    title({['\color[rgb]{0 .5 .5}Regiao ',num2str(r)] ;...
    ['\color[rgb]{0 0 0}Decaimento aproximado : ',num2str(Decaimentos(r))];...
    ['Coeficiente de determinacao : ',num2str(pesos(r))]});
    text(mean(x_data),c2.a * exp(c2.b*(mean(x_data)))+20,...
        ['  \fontname{times}D(t) = ',num2str(c2.a),...
        '{\ite}^{',num2str(c2.b),'{\itt}}']);
    text(x_data(1),max(Medias_Std_Regioes(:,2))+y_data(1)+20,...
        ['Desvio(s) padrao : ',num2str(Medias_Std_Regioes(:,2)')]);
    xlabel('tempo (s)')
    ylabel('Intensidades')
    legend('Intensidades medias/medianas','fitting curve','Location',...
    'SouthOutside');


    suptitle1(['Fitting exponencial do decaimento de intensidade entre os frames ',...
    num2str(Frames(1:end))])

end
% --------------------------------------------------------------------------
```

• Decaimento médio de todas as regiões

O decaimento médio considerando todas as regiões é obtido por uma média ponderada dos decaimentos encontrados, em que os pesos de cada decaimento são obtidos através do coeficiente de determinação de cada curva encontrada.

```
% --------------------------------------------------------------------------
Decaimento_medio = sum(pesos/sum(pesos).*Decaimentos);
display(' ')
display('O decaimento medio de todas as regioes calculado e:');
display(Decaimento_medio)
% --------------------------------------------------------------------------
```