

MATTEO TAFURO

ASSESSMENT OF THE CONTRACTILE PROPERTIES OF ENGINEERED HEART TISSUES BY OPTICAL FLOW

**UNIVERSITY
OF TWENTE.**

BACHELOR OF SCIENCE THESIS

Advanced Technology

July 2021

Assignment committee

<i>Chairman</i>	P.J. Lucas
<i>Supervisor</i>	N. Strisciuglio
<i>Member</i>	J.M. Rivera Arbelaez
<i>External Member</i>	A. van Houselt

Research Chair

Data Management and Biometrics
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente

Assessment of the contractile properties of engineered heart tissues by optical flow,
Matteo Tafuro, Bachelor of Science thesis © 2021

E-mail:

tafuromatteo00@gmail.com

Abstract

Due to their physiological and behavioral resemblance to in-vivo cardiac muscles, Engineered Heart Tissues (EHTs) are a promising tool for cardiovascular drug discovery, cell therapy and disease modeling. The beating pattern of EHTs is a direct indicator of their health and function, hence various techniques have been devised to gauge their contractile behavior; nevertheless, the established methods rely on invasive, expensive and delicate procedures. This thesis presents a novel workflow for the extraction, characterization and assessment of the contractile properties of EHTs that relies on motion estimation by optical flow (OF) and is purely based on bright-field images. It returns information in the form of (1) motion maps, (2) contraction profiles and (3) Histogram of Oriented Optical Flow, then classifies the tissue as correctly- or incorrectly-behaving according to the extracted characteristics of motion. Additionally, this paper tests the proposed method with three OF algorithms (block-matching, FlowNet2 and Farnebäck) and compares their performance on the given dataset: block-matching was found unsuitable for this study; FlowNet2 was only capable of analyzing tissues undergoing significant displacement while Farnebäck was configured with an iterative refinement to reach notable versatility and robustness. The iterative Farnebäck algorithm resulted in a classification accuracy of 88.8%, proving the applicability and potential of OF for the assessment of EHTs' contractile properties.

Acknowledgements

I would like to thank my supervisor, Dr. Nicola Strisciuglio, for his daily support, valuable advice and infinite patience. Your plentiful experience and contagious passion for Computer Vision have encouraged me to continue my studies in the field of Artificial Intelligence. My gratitude extends to José Manuel Rivera Arbelaez, whose expertise was really influential in shaping my experiment methods and critiquing my results. I would also like to thank Dr. Peter Lucas and Dr. Arie van Houselt for their insightful comments and suggestions.

In addition, I would like to express my sincere gratitude to my family for their continuous support and belief in me. Thank you, *mamma* and *papà*, for all your sacrifices and *grazie*, Francesco and Giulia, for the overwhelming love that makes me feel at home even if we are 1500 km apart.

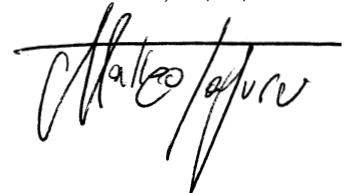
I am deeply thankful to Ludovica, the person I love, who stuck with me during these 10 weeks of researching, writing and coding even when I retreated to long days with my computer. The burden of this thesis was lessened substantially by your love, sweetness and patience. Thank you for always being by my side, thank you for believing in me. You made me who I am and I will be forever grateful. *Grazie*.

Special thanks go to Simba, our beloved kitten, for his unconditional companionship and comfort. His soothing morning purrs were indispensable to start the day.

My appreciation goes to all my friends for the cherished time spent together. A particular “thanks” to Rithvik, Luca and Michel and to my housemates Karolis and Emilija.

Finally, I dedicate this to me. I dedicate this to all my ups and downs, to my energy and motivation lacks, to my homesickness and to the endless hours of deprived sleep. It was all worth the effort!

Enschede, 12/07/2021

A handwritten signature in black ink, appearing to read 'Matteo', written over a horizontal line.

Contents

Contents	v
Abbreviations	viii
1 Introduction	1
1.1 Related work	2
1.1.1 CM quality assays	2
1.1.2 Optical flow	3
1.2 Aims and approach	4
1.3 Contributions	5
2 Dataset	7
3 Methodology	9
3.1 Video import	9
3.2 Pre-processing	9
3.2.1 Digital filtering	11
3.2.2 Region of Interest	11
3.2.3 Background removal	13
3.3 Optical flow estimation	15
3.4 Motion map video	15
3.5 Motion vectors analysis	16
3.5.1 Contraction profiles	17
3.5.2 Histogram of Oriented Optical Flow (HOOF)	17
3.6 Qualitative characterization	18
3.6.1 Smoothing of contractile profiles	19
3.6.2 Peak detection	20
3.7 Configuration of the workflow with specific OF estimation methods . . .	23
3.8 Assessment of OF estimation method performance	24
4 Experiments and results	25
4.1 Workflow with Block-matching	25
4.2 Workflow with FlowNet2	26
4.2.1 Assessment of Type 1 Tissue	26
4.2.2 Assessment of Type 2 Tissue	27
4.2.3 Effects of frame sampling on T2Ts	28
4.3 Workflow with Farnebäck	29
4.3.1 Influence of different window sizes	32

4.3.2	Influence of frame sampling	35
4.3.3	Assessment of T1Ts and T2Ts	36
4.3.4	Histogram of Oriented Optical Flow (HOOFs)	36
4.4	Comparison of performance	38
4.5	Results of classification	39
4.6	Algorithm limitations	40
5	Conclusions	43
5.1	Discussion and future outlook	43
5.2	Conclusion	44
APPENDIX		45
A	Appendix	47
A.1	Optical flow algorithms	47
A.1.1	Block-matching	47
A.1.2	FlowNet2	48
A.1.3	Farnebäck	51
A.2	Interpolation method in contraction profiles	55
A.3	Supplementary material	58
Bibliography		59

List of Figures

1.1	Types of iPSC-CM cultures	1
1.2	Bright-field images of four EHTs developed at the UT	2
1.3	Optical flow example	3
1.4	Middlebury color code	3
2.1	Differences among the videos of the dataset	8
3.1	Proposed method for EHTs motion analysis	10
3.2	Noise removal via bilateral filtering	11
3.3	Interference of debris in the motion map	12
3.4	Steps of the automatic definition of a Region of Interest	12
3.5	Automatic detection of a Region of Interest	13
3.6	Steps of background removal	14
3.7	Results of background removal	14
3.8	Contraction profile	16
3.9	Binning of an HOOF with $N = 18$	17
3.10	Qualitative assessment criteria	18
3.11	Effects of different SMA window sizes on a contraction profile	20
3.13	Peak detection notation	20
3.12	5-frames SMA applied to different contraction profiles	21
4.1	Contraction profile of a T1T obtained with block-matching	25
4.2	Results obtained on a T1T using FlowNet2	26
4.3	Final contraction profile and subsequent peak detection on a T1T	28
4.4	Contraction profile of a T2T obtained with FlowNet2	29
4.5	Interpolated contraction profiles at different sampling frequencies	29
4.6	Adjusted workflow of the EHT motion analysis using FlowNet2	30
4.7	T1T and T2T Farnebäck contraction profiles	32
4.8	Adjusted workflow of the EHT motion analysis using Farnebäck	33
4.9	Influence of the Farnebäck window size on the contraction profile	34
4.11	Missing valleys due to overlapping peaks of T2Ts	36
4.10	Study of contraction profiles obtained with different sampling frequencies	37
4.12	Comparison of profiles obtained with FlowNet2 and Farnebäck	38
4.13	Contraction profiles, peak detection and consequent classification	41
A.1	Displacement vector derived by a block-matching algorithm	47
A.2	Search window of a block-matching algorithm	47
A.3	Schematic representation of the FlowNet architecture	48
A.4	Contracting Part of FlowNet	49
A.5	Expanding part of FlowNet	50

A.6 Complete architecture of FlowNet 2.0	51
A.7 Effects of different interpolation methods on a contraction profile	57

List of Tables

3.1 Problems of the raw videos and proposed pre-processing solutions	9
3.2 Parameters of the peak detection algorithm	22
4.1 Farneback parameters	31
4.2 Confusion matrix of the classification problem	40
A.1 Supplementary material.	58

Abbreviations

BG	Background
BMA	Block Matching Algorithm
CMs	Cardiomyocytes
CNN	Convolutional Neural Network
CV	Computer Vision
CVD	Cardiovascular disease
EHT	Engineered Heart Tissue
FPS	Frames per second
hiPSC-CMs	hiPSC-derived cardiomyocytes
hiPSCs	Human induced pluripotent stem cells
HOOF	Histogram of Oriented Optical Flow
iPSC-CMs	iPSC-derived cardiomyocytes
iPSCs	Induced pluripotent stem cells
ML	Machine Learning
OF	Optical Flow
ROI	Region of Interest
SMA	Simple Moving Average
T1T	Type 1 Tissue
T2T	Type 2 Tissue
UT	University of Twente

Despite years of significant progress in the field of cardiology, cardiovascular diseases (CVD) remain the leading cause of death worldwide¹. Great effort has been put to understand the aetiology and pathogenesis of CVD, as well as to define prevention measures and possible cures. Yet, developments exhibit high failure rates during preclinical investigations and low likelihood of approval at all phases of clinical trials². The problem presumably resides in the unreliability of the model systems: for more than a century, laboratory studies have been limited to animal models due to the unavailability of *in vivo*-like human cardiomyocytes (CMs). Indeed, experiments were conducted on heart cells of small-sized animals like mice, rats and swine and larger mammals such as minipigs and goats. Although they resulted tremendously helpful to understand the basic mechanisms of the heart, animal models are relatively expensive, need experienced personnel, long-term housing, strict quality control, and are subject to ethical concerns³. Most importantly, human and animal CMs present major electrophysiological differences that compromise the validity of the results⁴.

Only recently, researchers at Kyoto University in Japan identified conditions that would allow specialized somatic adult cells to be genetically "reprogrammed" to a stem cell-like state⁶. These cells, called induced pluripotent stem cells (iPSCs), can differentiate into any cell type and open the possibility to investigate and understand, *in-vitro*, the behavior of different cells.

In fact, iPSCs can differentiate into CMs, offering a new source of *in vivo*-resembling human cardiac cells that could potentially represent a breakthrough in drug discovery, cell therapy and disease modeling. In practice, however, the realization and use of iPSC-derived cardiomyocytes (iPSC-CMs) are still hindered by poor levels of maturation⁴. The problem is especially prominent for two-dimensional cell cultures (Figure 1.1a), as cells *in vivo* behave very differently from their *in vitro* counterparts on stiff, flat, un-patterned 2D substrates³. Alternatively, bioengineered environments have been used to develop 3D cell culture systems, known as Engineered Heart Tissue (EHTs), that attempt to resemble *in vivo* conditions. A commonly used configuration features a matrix of biomaterials mixed with and populated by cells. These cells form an hourglass-shaped contracting tissue that is grown between two semi-flexible anchor points³ (Figure 1.1b), commonly referred to as micropillars (or simply pillars).

1.1	Related work	2
1.1.1	CM quality assays	2
1.1.2	Optical flow	3
1.2	Aims and approach	4
1.3	Contributions	5

¹ Reamy et al. (2018)

² Hay et al. (2014)

³ Zuppinger (2019)

⁴ Schaaf et al. (2011)

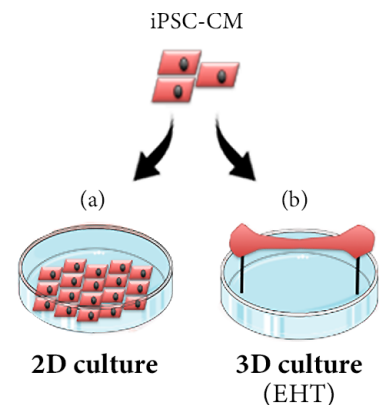


Figure 1.1: Two- and three-dimensional cell cultures of iPSC-CMs. Rearranged from "Learning from 3D Engineered Heart Tissue"⁵

⁵ Chase (2020)

⁶ Takahashi et al. (2006)

⁷ Kadota et al. (2012)

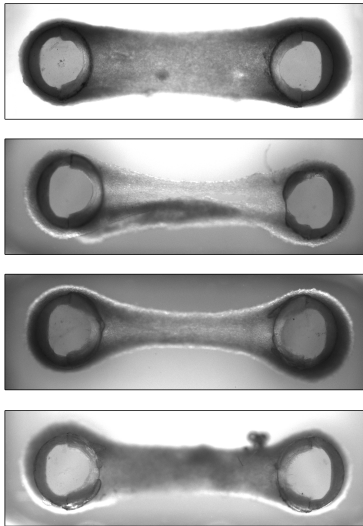


Figure 1.2: Bright-field images of four EHTs developed using hiPSC at day 10 of culture. Note how the images vary in illumination and how the tissues have different densities and conformations.

⁸ Dostanić et al. (2020)

⁹ Mannhardt et al. (2019)

¹⁰ Mathews et al. (2012)

¹¹ Sondergaard et al. (2012)

The Department of Applied Stem Cells Technologies AST and the BIOS lab-on-a-chip group of the University of Twente (UT) are collaborating on the development of a platform that allows for the fabrication of 3D cardiac tissues. EHTs are grown from *human* iPSCs (hiPSCs) and are kept in culture for 10 days under controlled temperature and humidity conditions (see Figure 1.2). Since the beating pattern of EHTs is a direct indicator of their health and function⁷, their contractile properties are assessed by electrically stimulating the tissues and recording their behavior at different time points.

Due to their ability to replicate *in vitro* the behavior of *in-vivo* cardiac tissues, a common application of EHTs is drug screening. When EHTs are used for such purpose, experiments are conducted with many compounds and each compound is tested at different concentrations. The result is a large number of recordings that require a careful and labor-intensive study. In these situations, an automatic and robust visual analysis becomes a necessity to examine the contractile properties of EHTs in a more accurate, efficient and effective way.

1.1 Related work

1.1.1 CM quality assays

Various techniques have been proposed to characterize and assess the contractile properties of EHTs. Dostanić et al. utilized a nanorobotic system with a micro-force sensing probe to measure the displacement of one of the pillars⁸. Mannhardt et al. replaced one micropillar with a piezo-bending actuator to capture the muscle contractions as a measurable electrical current⁹. Others, instead, attached a force transducer to one pillar to directly measure contraction forces^{10,11}.

While these methods yield accurate and reliable measurements, they have the following disadvantages: nanomechanical sensing is invasive and requires expensive machinery; current-based analyzes depend on specialized culture environments equipped with piezo-electric actuators and force transducers rely on delicate procedures to attach the micropillar to the experimental setup. In general, they all require extra expertise and costs, further complicating the validation processes of EHTs.

In contrast, the contractile properties of EHTs can be assessed by means of contact-less, bright field imaging-based techniques. These are generally cheaper and advantageous, since the simplest

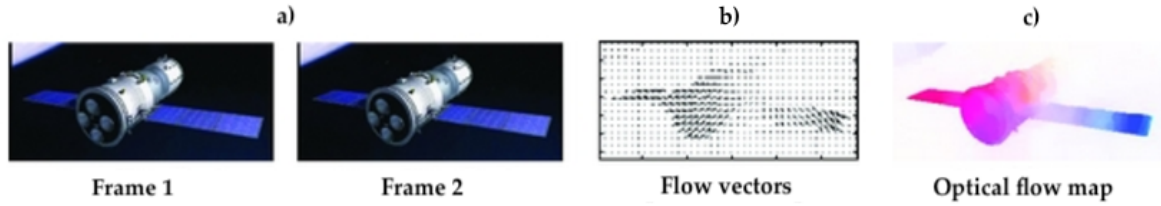


Figure 1.3: Application of optical flow on the two consecutive frames shown in **a)**. This results in the generation of flow vectors, as depicted in **b)**, which can be used to generate a motion map (see **c)** following the conventions of the Middlebury color code (refer to Figure 1.4). Retrieved and rearranged from Liang et al.²⁰.

setup is only dependent on standard laboratory and clinical imaging equipment. Hansen et al.¹², for instance, used a customized software package by Consulting Team Machine Vision (Pforzheim, Germany) to perform video optical analysis. The software relies on image processing to measure the distance between the top and bottom ends of the tissue and track the beating process.

Image processing and Computer Vision techniques are commonly used to analyze the contractile properties of 2D iPSC-CMs, with less effort towards EHTs. Many techniques have been proposed to assess the motion of iPSC-CMs monolayers, which incorporate various algorithms based on fast Fourier transforms¹³, transmitted light intensity changes¹⁴, velocity vectors and optical flow (OF)^{15–19}. This thesis aims to investigate the applicability of OF for the analysis of the contractile properties of EHTs.

1.1.2 Optical flow

Optical flow (or *optic* flow) is defined as the projection of the three-dimensional motion of objects, relative to the camera/observer, onto its image plane²². Figure 1.3 presents an example: in (a), two consecutive frames are analyzed via OF to obtain the flow vectors depicted in (b). This can be done pixel-wise to obtain a *dense* vector field or only for certain regions of the image to generate a *sparse* optical flow. Then, the Middlebury color convention (Figure 1.4) can be used to image the resulting vectors in the form of a motion map, where the hue and the brightness represent the direction and the magnitude of the vectors – respectively. Such map can be seen in Figure 1.3(c).

Classic approaches to compute the optical flow include the Horn and Schunk²³, the Lucas Kanade²⁴ and the Farnebäck²⁵ methods. They are based on brightness constancy and spatial coherence constraints²⁶. These are adequate for handling small subpixel motion but fail to perform in the presence of motion discontinuities, occlusions and large displacements²⁶, as often happens in real-world applications.

¹² Hansen et al. (2010)

¹³ Pasqualin et al. (2016)

¹⁴ Stummann et al. (2008)

¹⁵ Kunihiro et al. (2012)

¹⁶ Chen et al. (2014)

¹⁷ Hayakawa et al. (2014)

¹⁸ Kopljär et al. (2017)

¹⁹ Sala et al. (2018)

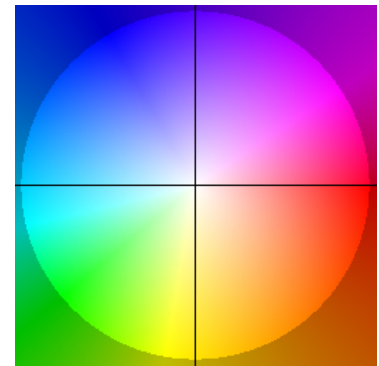


Figure 1.4: The Middlebury color scheme used to convert flow vectors into motion maps. Vectors with length zero sit at the center (and are imaged as white), while the longest vector(s) lie on the highlighted circumference (and are colored depending on their orientation); the remaining vectors are normalized between these two values. Image courtesy of Chantas et al.²¹.

²² Beauchemin et al. (1995)

²³ Horn et al. (1981)

²⁴ Lucas et al. (1981)

²⁵ Farnebäck (2003)

²⁶ Anthwal et al. (2019)

²⁶ Anthwal et al. (2019)

²⁷ Sun et al. (2008)²⁸ Rosenbaum et al. (2013)²⁹ Roth et al. (2005)³⁰ Dosovitskiy et al. (2015)³¹ Ilg et al. (2017)³² Ahmadi et al. (2016)³³ Yu et al. (2016)³⁴ Lai et al. (2017)³⁵ Yang et al. (2018)³⁶ Tavakoli et al. (2008)

Statistical learning techniques have also been used for over a decade. OF statistics have been studied using Gaussian scale mixtures²⁷, Gaussian mixture models²⁸ and depth maps²⁹.

Recently, Convolutional Neural Networks (CNNs) have been exploited to extract and match features in different images. A remarkable contribution was done by Dosovitskiy et al. with the development of FlowNet³⁰ and FlowNet2³¹. CNN-based methods proved to be extremely accurate in real-world applications, but the supervised learning techniques require massive ground truth training data²⁶. As a consequence, unsupervised^{32,33} and semi-supervised^{34,35} learning methods have been proposed as possible alternatives.

Finally, it is noteworthy that all techniques are application-dependant and therefore must be chosen carefully³⁶.

1.2 Aims and approach

The aim of this thesis is to develop an OF-based algorithm that automates the extraction, analysis and assessment of the contractile properties of 3D cardiac tissues from bright-field microscopy recordings. More specifically, given the footage of a beating EHT the algorithm should provide:

1. A *quantitative* characterization of the motion;
2. A temporal *visual* mapping of the movement over the tissue, to identify regions involved in the beating process;
3. A *qualitative* assessment of the contractile motion, to determine if the tissue is behaving as expected (for instance, contractions and relaxations should respect a predefined frequency).

For a pair of subsequent frames, OF produces a vector field that describes the displacement of each pixel, whereby the quantitative characterization required by Point 1 can be derived. Then, each vector field can be transformed into a motion map that gives a visual representation of the displacement; the concatenation of all these maps generates an image sequence (which will be referred to as "motion map video") that fulfills Point 2. Finally, the obtained characteristics of motion can be used to ascertain whether the tissue is behaving as expected, resulting in the qualitative categorization presented in Point 3; the exact expected behavior will be defined later in the [Methodology](#) section.

1.3 Contributions

Image-based assessments of the contractile properties of artificial cardiac tissues have been widely explored in the context of 2D iPSC-CMs, but we lack visual analysis tools for EHTs that are purely based on bright-field images. For this reason, this thesis proposes:

- ▶ A general optical flow-based workflow for the assessment of the contractile properties of EHTs, that is independent of the chosen OF estimation method;
- ▶ The investigation of the feasibility of three optical flow algorithms (block-matching, FlowNet2 and Farnebäck) in the proposed workflow, followed by a comparison of their performance on the given dataset;
- ▶ An iterative Farnebäck-based motion estimation algorithm that relies on the Signal To Noise Ratio of the estimated motion.

The rest of this thesis is structured as follows: [Chapter 2](#) describes the available dataset to gently introduce the reader to the topic; [Chapter 3](#) thoroughly discusses the general workflow and dives into the details of each step; [Chapter 4](#) applies the workflow to the three chosen OF algorithms and analyzes the results; finally, [Chapter 5](#) concludes the paper by summarizing and discussing the outcomes of the study.

The contractile properties of 3D cardiac tissues are assessed by recording and visually analyzing their response upon electrical stimulation. The dataset used in this work comprises 81 five-seconds-long videos of EHTs being electrically stimulated at 2 Hz. The bright-field microscopy recordings are captured using a Nikon Ti2-E Inverted Microscope and come in the *NIS-Elements* Image Sequence format (ND2). The 11-bit, 13.3 mm × 13.3 mm, 4.2 megapixel image sensor of the mounted Photometric Prime BSI camera offers the ability to transmit only a region of the sensor (Region of Interest, or ROI). This reduces the image size and the bandwidth required for each image, increasing the frame rate without causing resolution losses. Predefined ROIs were used to increase the default acquisition rate of 63 frames per second (fps) to 100 fps.

From a technical point of view, the videos contain variations with respect to:

Illumination: some recordings feature a completely white background. Others present much darker areas. This is exemplified in [Figure 2.1 \(a\)](#).

Compaction of the tissues: EHTs can contain regions where the tissue compaction is lower. In such situations the surface area appears nonuniform in terms of contrast and illumination, determining noisy motion estimations. See [Figure 2.1 \(b\)](#) for a visual reference.

Presence of debris: numerous videos display small particles that move within the image window, as seen in [Figure 2.1 \(c\)](#). These objects are dead cells displacing under the effect of the electric field that stimulates the EHTs. The movement of these debris can seriously interfere with the computation of the optical flow.

Presence of undesired artifacts: tissues can have unexpected objects anchored on their surface area. These artifacts may appear at any position relatively to the EHT and do not have a definite source: they may be cells that protrude due to a fault in the growing process, air bubbles or other undefined entities (see [Figure 2.1 d](#)).

Magnitude of the displacement: some videos show EHTs undergoing significant displacement. Others, instead, feature EHTs whose motion is very gentle and barely perceptible to the naked eye. This difference is caused by diverse biotechnological conditions of the tissues (e.g. stiffness of the pillars, biological variability of

the cells) that are outside the scope of this paper. For the sake of simplicity, EHTs that experience great displacement will be referred to as *Type 1 Tissues* (T1T) while the other typology will be named *Type 2 Tissues* (T2T). These behaviors are exemplified in [Supplementary Material 1](#) and [2](#).

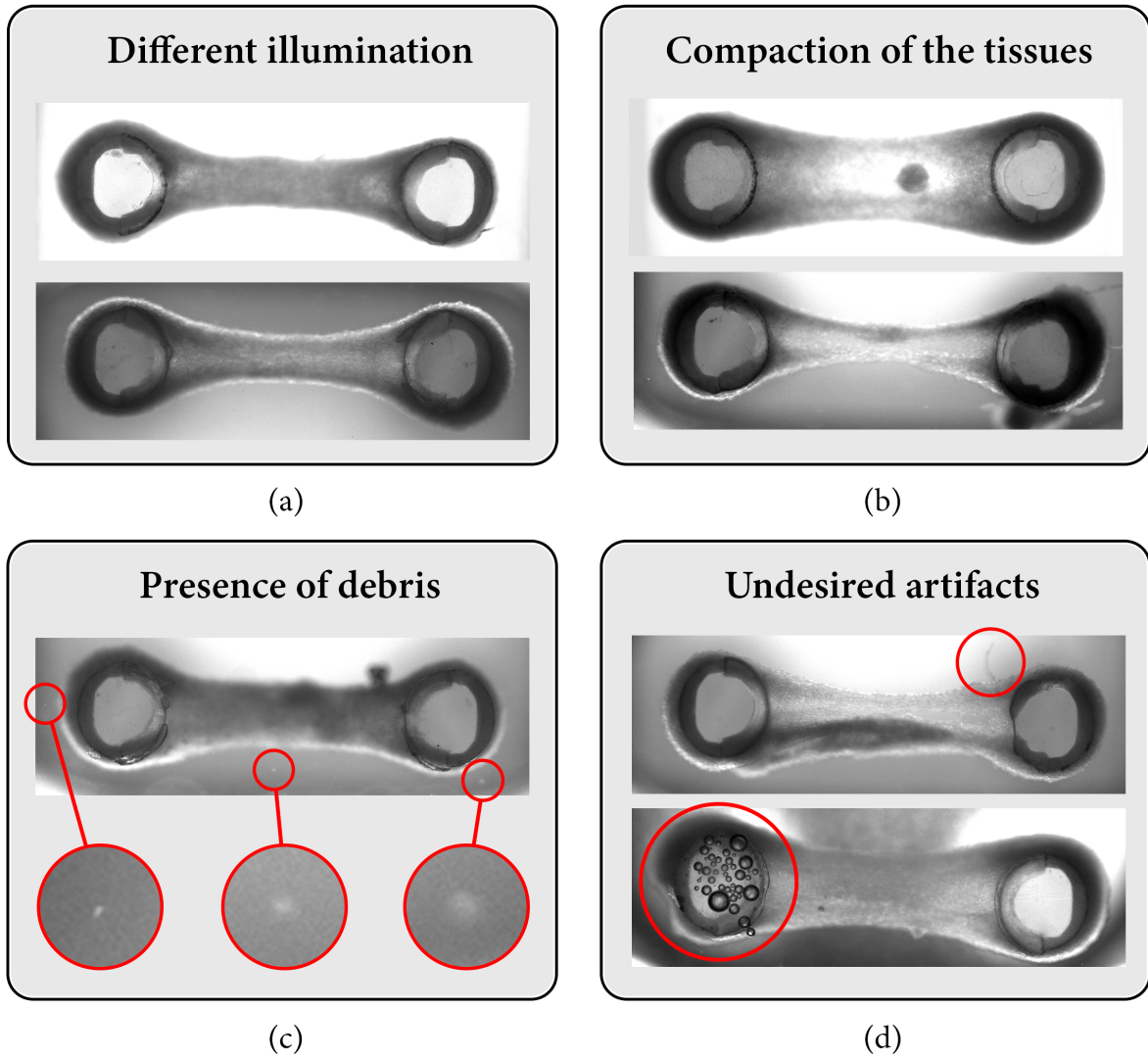


Figure 2.1: Bright-field images of different EHTs that show how the videos differ in terms of (a) illumination, (b) compaction of the tissue, presence of (c) debris and (d) undesired artifacts.

The proposed pipeline for the motion analysis of EHTs can be configured with any optical flow algorithm and covers all steps from the video import, through information extraction to the final qualitative assessment of the contractile behavior. It is schematically illustrated in [Figure 3.1](#) and thoroughly discussed in the next sections.

3.1 Video import

The EHTs recordings come as ND2 files. Such format is not supported by the majority of the media players and, for this reason, one could go through an intermediate step where the input file is converted into a standard video format (such as .avi). However, in view of the motion analysis this would be disadvantageous for several reasons: first, ND2 files contain the microscope metadata that would go lost in such conversion; secondly, the ND2 format preserves 11-bit image depth, which is not supported by avi files; lastly, the ND2-to-avi conversion implies iterating over the entire image sequence and therefore significantly increasing the computational cost.

For these reasons, the footage is imported in its raw ND2 format using a dedicated python library, `nd2reader`³⁷. The function `ND2Reader` opens the file as an image sequence and allows easy access to individual frames.

3.2 Pre-processing

The videos are pre-processed to enhance the quality of the image frames. There are two issues that we wish to solve prior to the OF estimation, and these are displayed in [Table 3.1](#) together with the corresponding proposed solutions.

	Problem	Pre-processing step
1	Image noise	Digital filtering
2	Presence of debris	Region of Interest
		Background removal

3.1	Video import	9
3.2	Pre-processing	9
3.2.1	Digital filtering	11
3.2.2	Region of Interest	11
3.2.3	Background removal	13
3.3	Optical flow estimation	15
3.4	Motion map video	15
3.5	Motion vectors analysis	16
3.5.1	Contraction profiles	17
3.5.2	Histogram of Oriented Optical Flow (HOOF)	17
3.6	Qualitative characterization	18
3.6.1	Smoothing of contractile profiles	19
3.6.2	Peak detection	20
3.7	Configuration of the workflow with specific OF estimation methods	23
3.8	Assessment of OF estimation method performance	24

³⁷ Rybarski et al. (2021)

Table 3.1: Technical problems encountered in the raw input videos and proposed pre-processing solutions.

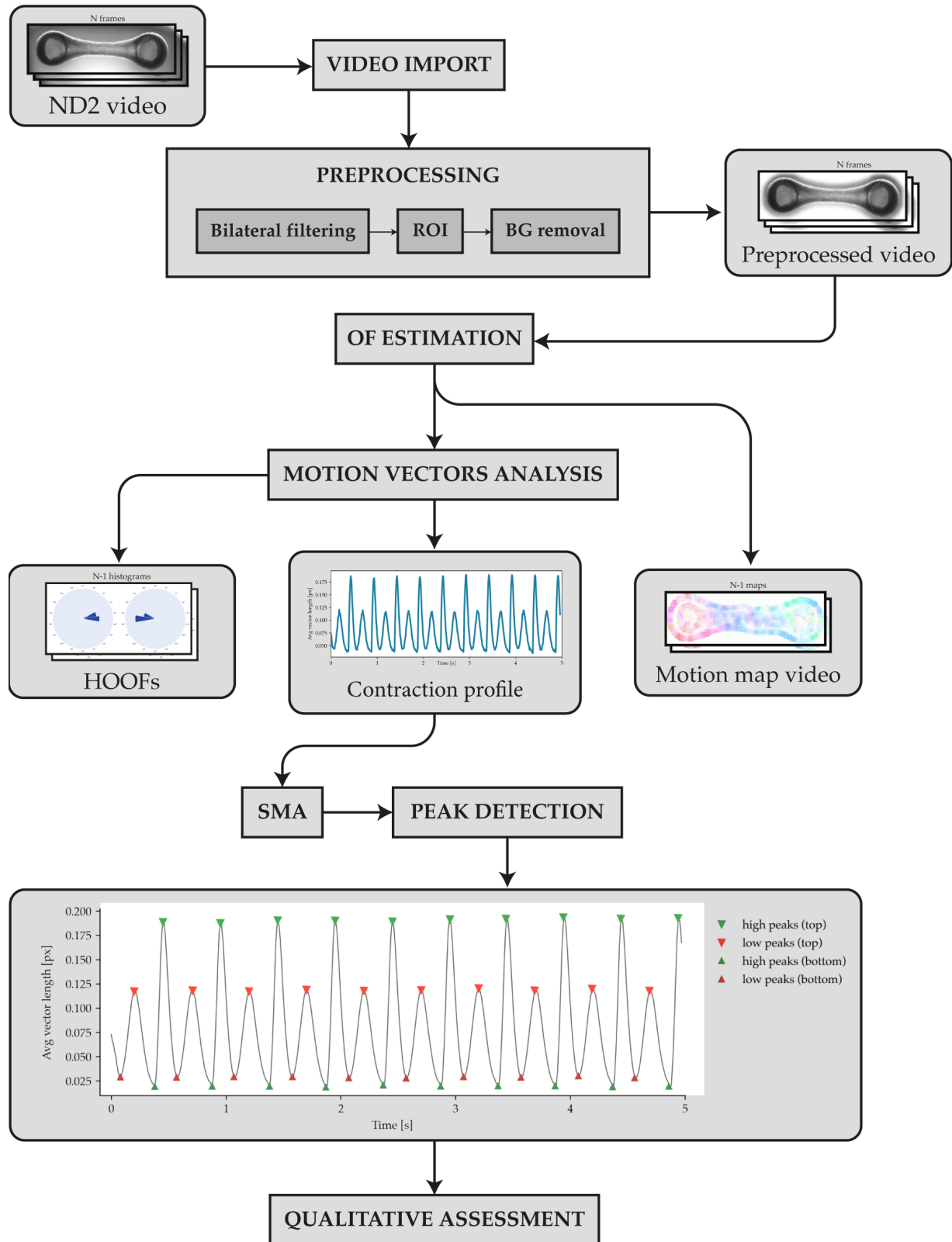


Figure 3.1: Proposed method for EHTs motion analysis. The pipeline can be configured with any optical flow algorithm and covers all steps from the video import, through information extraction to the final qualitative assessment of the contractile properties of EHTs.

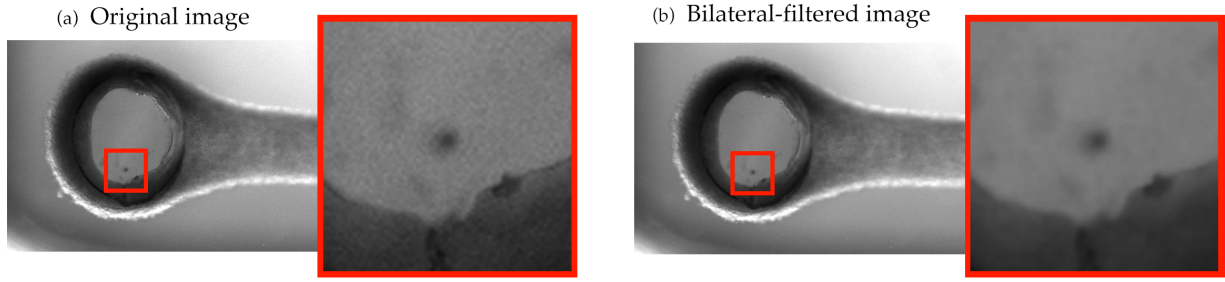


Figure 3.2: Noise removal via bilateral filtering. Image (a) shows the original image while (b) illustrates the bilateral-filtered counterpart. Notice how the filter reduces noise while preserving the edges.

3.2.1 Digital filtering

Noise is reduced by applying a digital filter to blur and smoothen the image. Both spatial- and frequency-domain filtering can be performed, but the edge-preserving *bilateral filter* was found to yield better OF estimates. Essentially, it replaces the intensity of each pixel with a weighted average of intensity values from its neighborhood, where the weights depend not only on Euclidean distance of the points, but also on their radiometric differences. As a result, sharp edges are preserved.

Accordingly, bilateral filtering depends on three parameters: (1) d , the diameter of each pixel neighborhood; (2) σ_s , the filter sigma in the coordinate space, where a larger value means that farther pixels will influence each other as long as their colors are close enough; (3) σ_c , the filter sigma in the color space, where greater values mean that farther colors within the pixel neighborhood will be mixed together.

The best results have been obtained setting these parameters to 5, 80 and 80 respectively. [Figure 3.2](#) illustrates an image frame before and after filtering.

3.2.2 Region of Interest

As illustrated in [Figure 2.1](#), dead cells may float in the chemical medium of the EHT and therefore they may be visualized in the image frame. We wish to remove these debris, as they can compromise the quality of the motion map video.

In a motion map, the hue and intensity of the colors represent the direction and the magnitude of motion. The colors (defined by the Middlebury convention, as seen in [Figure 1.4](#)) are normalized on the greatest displacement of the corresponding vector field so that the most intense color represents the strongest displacement and zero vectors are basically colorless. However, we wish to output a motion map *video* and, as a consequence, each motion map can

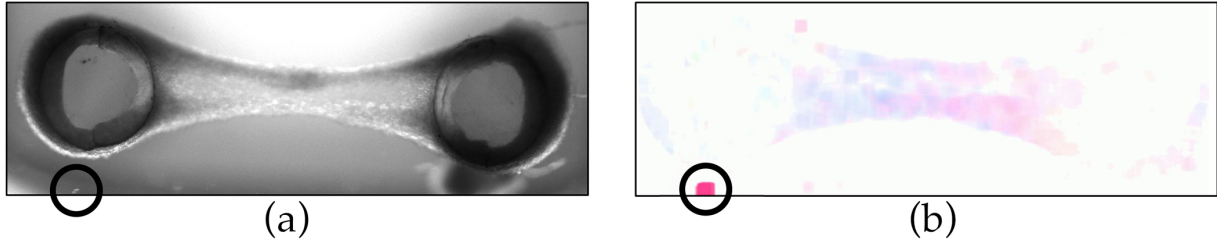


Figure 3.3: Presence of debris (Fig. a) and their interference in the motion map (Fig. b)

not be normalized on its own greatest displacement. Rather, the vectors are normalized on the strongest displacement of the *whole* image sequence (more details will come in [Section 3.4](#)).

The problem with the debris is that if the floating particles undergo movements that are more rapid than those of the tissue (if the displacement of the debris is greater than that of the EHT), the drifting molecules will be displayed with an intense color, while the moving tissue will barely be visible (as seen in [Figure 3.3](#)).

To account for this issue, one possible pre-processing step is to define a *Region of Interest* (ROI) and crop the image accordingly. This reduces the pixels belonging to the background and effectively decreases the chances of detecting a floating particle. Additionally, cropping the image enforces the alignment of the tissue in the center of the image frame. This will be particularly useful in the post-OF phase to perform region-specific analyses.

The ROI selection is automated and integrated in the workflow. Its implementation is inspired by an article of Munawwar Firoz³⁸ and is designed as follows:

³⁸ Firoz (2019)

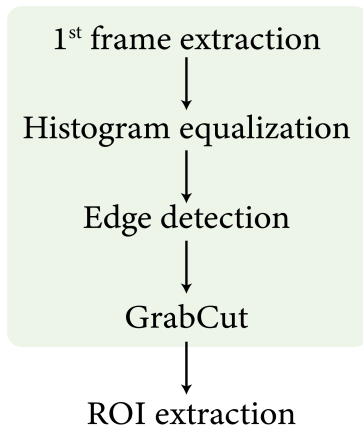


Figure 3.4: The five steps of the automatic definition of a Region of Interest. The green region contains the steps in common with background removal, an alternative pre-processing step that will be discussed later.

³⁹ Dollár et al. (2013)

⁴⁰ Rother et al. (2004)

1. The first frame of the input image sequence is extracted. The remaining steps are performed on this frame only.
2. Histogram equalization is applied to enhance the contrast and facilitate edge detection (see next point);
3. A structured forest approach³⁹ is used to detect the edges. Keeping the largest contour by area will return a rough selection of the tissue, as illustrated in [Figure 3.5](#) under "Edge selection";
4. The coarse selection undergoes a refinement process by running a GrabCut algorithm⁴⁰. GrabCut requires manually-provided information regarding the position of the foreground, background and uncertain areas. To do this automatically, a morphological filter is applied to the rough contour to shrink it and use it as SURE FOREGROUND (white); the initial, non-reduced contour is marked as PROBABLE FOREGROUND (gray) and the rest is indicated as SURE BACKGROUND (black), as depicted in [Figure 3.5\(a\)](#) under "GrabCut". Providing these regions, the algorithm has enough information to perform

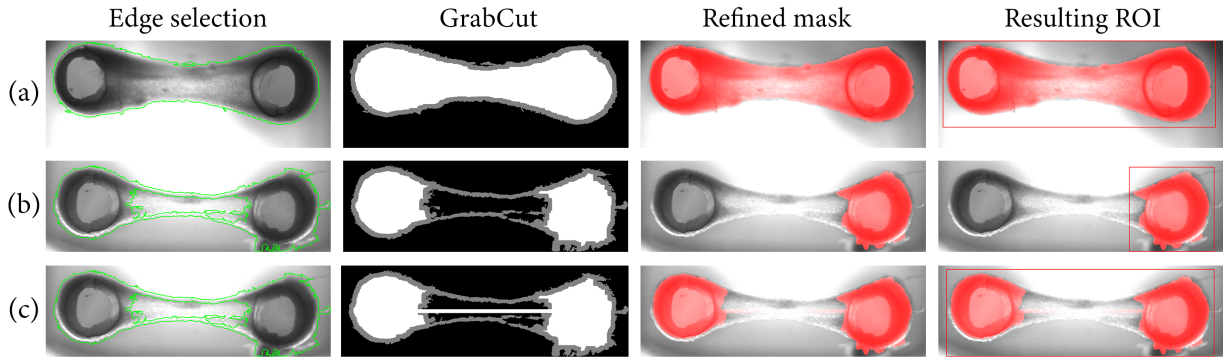


Figure 3.5: Automatic detection of a Region of Interest. First, edge detection is performed using a random forest approach (see the green contour under "Edge selection"). Then, GrabCut is run using the previously-derived information to refine the initial selection (illustrated under "GrabCut"). In this picture, white corresponds to SURE FOREGROUND, gray indicates PROBABLE FOREGROUND and black depicts BACK FOREGROUND. Lastly, the obtained contour (depicted under "Refined contour") is used to draw a bounding box (See the red rectangles under "Resulting ROI").

foreground-background segmentation and return a refined version of the initial selection.

5. Given the refined contour around the tissue, a bounding box can easily be derived by finding the top-left and bottom-right extreme points. Since the ROI is defined exclusively based on the first frame of the video, the resulting rectangle is expanded by 20 pixels (experimentally determined value) to account for the beating motion that will occur in later frames. See "Resulting ROI" of Figure 3.5(a).

This approach yields a success rate of 90.1%, returning a correctly-positioned bounding box for 73 out of 81 videos of the dataset. However, it fails when the initial edge detection is inaccurate (see the "Edge selection" of Figure 3.5b and c). In such scenarios, GrabCut does not have enough information to refine the selection and the resulting mask only contains the tissue around one of the two pillars. The automatically-generated ROI is misplaced: cropping the image on the bounding box would imply neglecting the majority of the image, where the motion information resides. This is exemplified in Figure 3.5(b).

To circumvent the issue, a rectangle is drawn and marked as SURE FOREGROUND before running GrabCut, as shown in Figure 3.5(c). Since in the 8 defecting videos the two pillars are always detected, this enforces the connection of the two components and therefore the generation of the correct bounding box. This workaround improves the robustness of the automatic ROI detection algorithm, returning the appropriate result for all 81 videos.

3.2.3 Background removal

The definition of a ROI reduces the chances of floating cells to appear by cropping the images and removing part of the background.

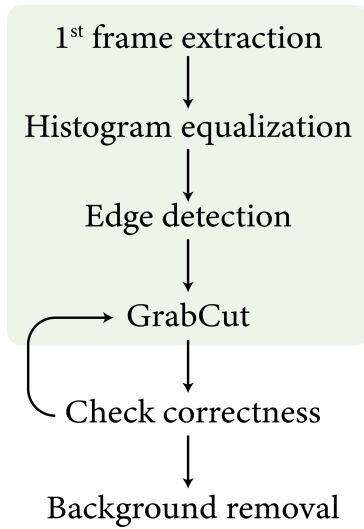


Figure 3.6: The steps of the *semi-automatic* background removal. The green region contains the steps in common with the ROI definition, as seen in Figure 3.4.

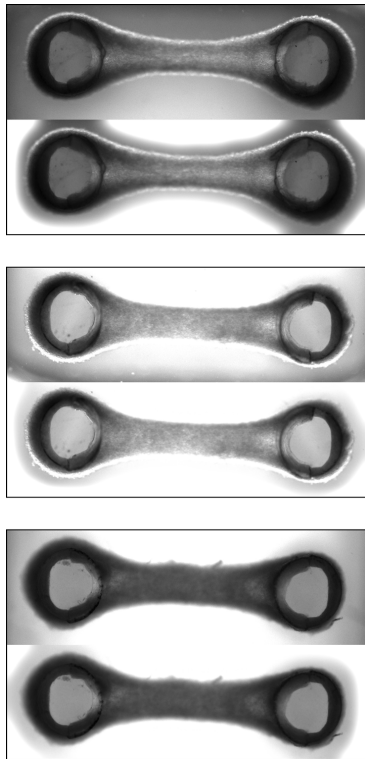


Figure 3.7: First frame of three videos processed with background removal.

To reduce the presence of debris even further, one could remove the background completely.

Background (BG) removal is implemented along the same lines of ROI detection: the contour of the tissue is extracted from the first frame of the image sequence, following Steps 1-4; then, we wish to define a mask that specifies the areas to be kept and those to be removed.

The contour obtained by GrabCut (which was used to derive the final bounding box in the previous section) is transformed into the final background-removing mask in the following manner:

1. The contour is transformed into a binary image;
2. A morphological filter is used to dilate the resulting binary mask by 20 px (empirically defined value), to account for the beating motion that will occur in later frames;
3. A Gaussian filter (window size = 30 px, also chosen experimentally) is applied to the mask in order to blur the edges and smoothen the transition between background and foreground.
4. Finally, the mask is applied to the image sequence to remove the background. Figure 3.7 displays some results.

This approach relies on the assumption that GrabCut returns the correct contour. In reality, it gives proper results for 73 out of the 81 tissues but still fails in 8 cases, only selecting the area around one pillar. In those cases BG removal would result in the elimination of most of the image, which would cause the loss of *all* the motion information. To resolve the issue, background removal is made *semi-automatic*: the user is asked whether GrabCut returned the right contour. If it did, the algorithm proceeds; if not, the user is required to give GrabCut additional information by marking background and foreground areas on the image. This process is repeated until the selection is correct, after which background removal is run.

The final user could decide whether they want to pre-process the image sequences using ROI detection, BG removal or both. On the one hand, simply cropping the image using a ROI makes the process completely automatic, but increases the interference caused by the debris. On the other hand, BG removal ensures the elimination of any floating particle but requires the supervision of the user.

The current implementation includes both pre-processing steps: ROI aligns the tissue in the center of the image frame, while BG removal eliminates all obstructions to the OF computation.

3.3 Optical flow estimation

As introduced in [Section 1.3](#), one of the contributions of this thesis is to include and compare 3 OF algorithms in the pipeline. Obviously, these must be chosen carefully according to the final application.

For this reason, the following criteria were delineated to determine whether an algorithm is suitable for this study:

1. **Source code:**

The source code of the corresponding paper is freely available on the internet and is written in *Python*. If applicable, the authors provide pre-trained models with pre-trained weights;

2. **Small displacements:**

The algorithm is capable of detecting subpixel motion;

3. **Dense optical flow:**

The resulting optical flow is preferably dense;

4. **Computational cost:**

The OF computation time does not exceed one hour on the experimental setup (MacBook Pro with Intel Core i7 Processor (2x 3.5 GHz), 16 GB LPDDR3 RAM and Intel Iris Plus 650 GPU). This is to ensure the feasibility of batch analyses of large datasets.

Based on these criteria, three algorithms were selected:

1. **Block-matching**
2. **FlowNet2**
3. **Farnebäck**

(The reader can consult [Appendices A.1.1](#), [A.1.2](#) and [A.1.3](#) respectively to find more details about the three algorithms)

Many popular and powerful frameworks, instead, were disregarded because they were conflicting with these criteria. For instance, the classic *Lucas Kanade* technique is suitable to handle small displacements but returns a sparse optical flow; the first version of *FlowNet* relies on CNNs but performs poorly on subpixel motion; *RAFT*, another CNN-based algorithm, took almost 9 hours to process one video; finally, the code of the majority of the OF estimation methods could not be freely downloaded from the internet.

3.4 Motion map video

As alluded to multiple times, each vector field outputted by the OF estimation can be imaged in the form of a *motion map*. The resulting

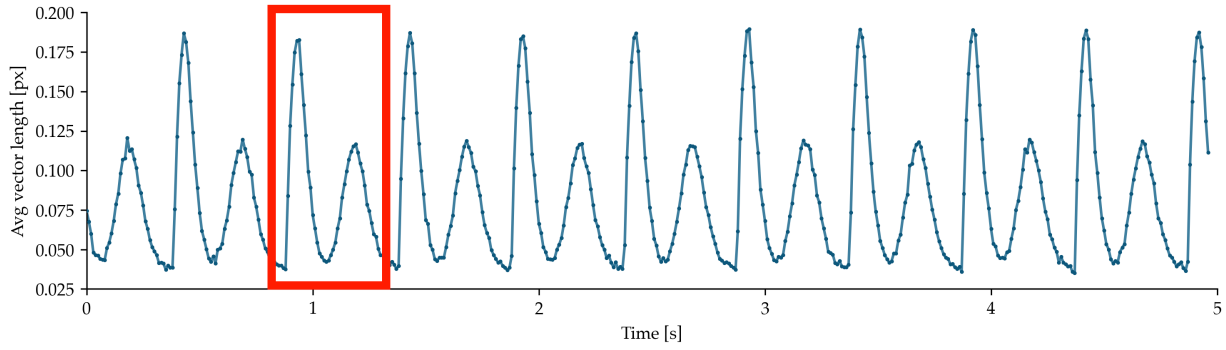


Figure 3.8: Contraction profile derived by averaging the vector length over time. Each beating cycle consists of a contraction and relaxation phase, and both are characterized by a peak.

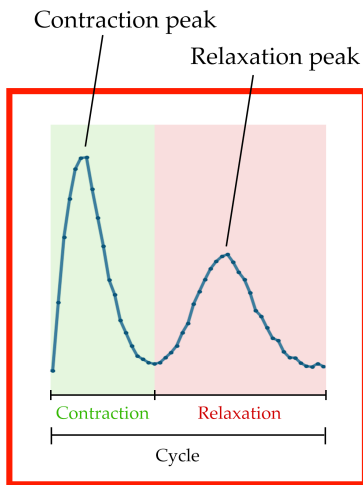


image varies in hue and intensity following the Middlebury color convention: vectors with length zero sit at the center of the color scheme of Figure 1.4 (and are imaged as white), while the longest vector(s) lies on the highlighted circumference (and is colored depending on its orientation); the remaining vectors are normalized between these two values.

Researchers require a *temporal* visual characterization that shows the distribution of the movement over time. This can not simply be achieved by concatenating all individual motion maps, as these are normalized *locally* on the greatest vector length of the respective OF field. Instead, to obtain a more relevant and insightful visualization of the contractile motion, each motion map is generated using the *global* greatest displacement as a normalization factor (note that this implies that each video has its own color scale, hence different tissues are not directly comparable). Only then, the maps are concatenated and exported in AVI video format using the original frame rate.

3.5 Motion vectors analysis

For a pair of subsequent image frames, the OF estimation returns a matrix of vectors. However, the raw optical flow data is a high-dimensional representation of the motion that does not allow for an immediate quantitative study of the contractile properties. For this reason, reducing the dimensionality of the data becomes a necessity.

A variety of techniques have been devised to address this problem. The majority of the related papers reduce the dimensionality of the OF output by only considering the magnitude of the vectors^{15,17,18}. The information contained in each matrix is summarized by the average vector length, which is eventually plotted against time to give a characteristic curve that will be referred to as *contraction profile* (Fig. 3.8). Other publications^{41,42}, although not related

⁴¹ Solichin et al. (2018)

⁴² Huang et al. (2017)

to hiPSC-CMs, *do* consider the direction of the vectors and use Histograms of Oriented Optical Flow (HOOFs) to summarize the angle distribution of the movement (Fig. 3.9). In this paper, both methods are used to extract motion information, although – due to the limited time available for this study – only the former is used for the final qualitative assessment.

3.5.1 Contraction profiles

By averaging and plotting the magnitude of the vectors, motion information can be determined with a time resolution equal to the frame rate. Although contraction and relaxation occur in opposite directions, the magnitude of a vector $\mathbf{f} = (u, v)$ is a positive value given by $\|\mathbf{f}\| = \sqrt{u^2 + v^2}$: this results in a two-positive peak profile that reflects the contraction–relaxation process (see Figure 3.8).

3.5.2 Histogram of Oriented Optical Flow (HOOF)

Chaudhry et al. firstly introduced HOOF for the recognition of human actions⁴³ and, subsequently, many other classification tasks were solved using HOOF. In this paper the histograms are used to quantify the contractile motion of EHTs in the following manner: first, one of the chosen optical flow algorithms is applied to generate OF fields from successive frames of the video; secondly, since each motion vector can be oriented between 0 and 360 degrees, this range is quantized in N bins and each vector is assigned to the respective group based on its orientation; lastly, the value of each bin is calculated by summing the magnitude of the contained vectors. In other words, each vector $\mathbf{f} = (u, v)$ of the optical flow field is binned according to its primary direction from the horizontal axis ($\theta_f = \arctan(v/u)$) and weighed according to its magnitude ($\|\mathbf{f}\| = \sqrt{u^2 + v^2}$).

In this work, the directions of flow vectors are quantified into 18 groups ($N = 18$), so that the normalized directions (denoted by b_θ) are codified into numbers $b_\theta = \{1, 2, 3, \dots, 18\}$ and represent angle ranges of 20 degrees, as illustrated in Figure 3.9. The assignment of a vector to a specific bin is governed by Equation 3.1, while the value of each bin is computed with Equation 3.2. Note that the first bin is purposely *centered* rather than *edge-aligned* at 0° to capture more information about the contractile motion. Same holds for 180° .

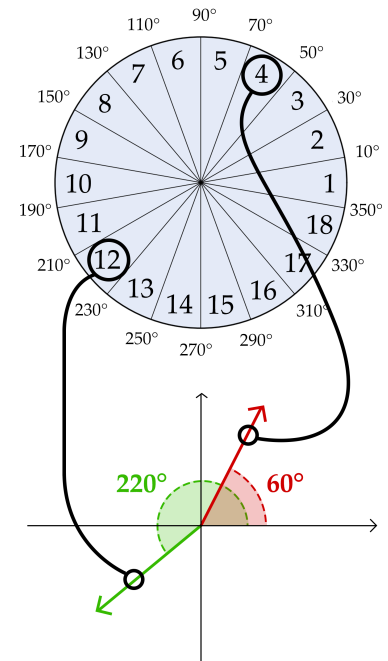
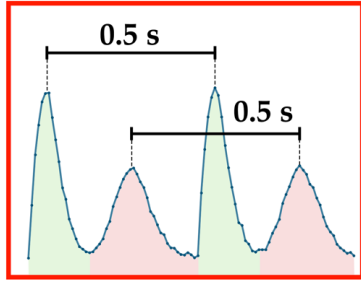
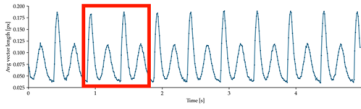
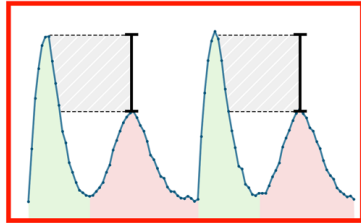


Figure 3.9: Binning of an Histogram of Oriented Optical Flow with $N = 18$.

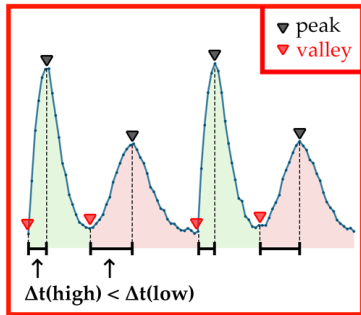
⁴³ Chaudhry et al. (2009)



(a) Frequency constraint



(b) Amplitude constraint



(c) Velocity constraint

Figure 3.10: Qualitative characterization of an EHT from its contraction profile. The assessment is performed based on the frequency constraint (Fig. a), the amplitude constraint (Fig. b) and the velocity constraint (Fig. c).

$$b_{\theta} = \begin{cases} 1, & \text{if } 350^{\circ} \geq \theta_f > 10^{\circ} \\ 2, & \text{if } 10^{\circ} \geq \theta_f > 30^{\circ} \\ 3, & \text{if } 30^{\circ} \geq \theta_f > 50^{\circ} \\ 4, & \text{if } 50^{\circ} \geq \theta_f > 70^{\circ} \\ 5, & \text{if } 70^{\circ} \geq \theta_f > 90^{\circ} \\ 6, & \text{if } 90^{\circ} \geq \theta_f > 110^{\circ} \\ 7, & \text{if } 110^{\circ} \geq \theta_f > 130^{\circ} \\ 8, & \text{if } 130^{\circ} \geq \theta_f > 150^{\circ} \\ 9, & \text{if } 150^{\circ} \geq \theta_f > 170^{\circ} \\ 10, & \text{if } 170^{\circ} \geq \theta_f > 190^{\circ} \\ 11, & \text{if } 190^{\circ} \geq \theta_f > 210^{\circ} \\ 12, & \text{if } 210^{\circ} \geq \theta_f > 230^{\circ} \\ 13, & \text{if } 230^{\circ} \geq \theta_f > 250^{\circ} \\ 14, & \text{if } 250^{\circ} \geq \theta_f > 270^{\circ} \\ 15, & \text{if } 270^{\circ} \geq \theta_f > 290^{\circ} \\ 16, & \text{if } 290^{\circ} \geq \theta_f > 310^{\circ} \\ 17, & \text{if } 310^{\circ} \geq \theta_f > 330^{\circ} \\ 18, & \text{if } 330^{\circ} \geq \theta_f > 350^{\circ} \end{cases} \quad (3.1)$$

$$h(b_{\theta}) = \sum \|f\| \quad \forall f \in b_{\theta} \quad (3.2)$$

3.6 Qualitative characterization

Motion information is summarized in (1) contraction profiles and (2) HOOFs but, as already done by others^{15,17,18}, this paper only utilizes the former to draw qualitative conclusions. The latter, instead, currently act as an additional source of information, but in future implementations the HOOFs could serve as input features for a binary classification problem (well-behaving / misbehaving tissues).

Based on the expertise of BIOS lab researchers, criteria have been established to judge the correct functionality of an EHT from its contraction profile:

1. The contraction peaks (also referred to as *high* peaks) occur at a constant frequency of 2 Hz, which is the frequency of the electric signal that stimulates the tissue. This implies a time interval of 0.5 seconds between subsequent peaks, as shown in Figure 3.10(a).
2. For the same reason, the relaxation peaks (also denoted *low* peaks) occur at a constant frequency of 2 Hz, as seen in

Figure 3.10(a). This and condition 1 will be referred to as the *frequency constraints* and constitute the main indicator of the correct functioning of the tissue.

3. The *amplitude constraint*: each contraction peak is always higher than the following relaxation peak, as illustrated in Figure 3.10(b);
4. *Velocity constraint*: contraction happens faster than relaxation. This aspect can be recognized in the contraction profile by looking at the time distance between the peaks and their preceding valley, see Figure 3.10(c).

The aim is to state whether the tissue is behaving correctly or not. In the proposed workflow, these criteria are checked by examining the results of a fine-tuned peak detection algorithm, which is run after applying a Simple Moving Average to the contraction curve to smooth noisy profiles while preserving their shape.

3.6.1 Smoothing of contractile profiles

Some contraction profiles (as the ones shown so far) present clearly distinguishable contraction and relaxation peaks. These curves are derived from videos where the entire tissue is undergoing significant motion, i.e. T1Ts. Instead, the recordings of T2Ts often result in noisy contraction profiles that hinder the peak detection step.

For this reason, the contraction curves are smoothed out prior to peak detection. An effective low-pass filter can be designed to remove high-frequency fluctuations, but only upon a careful analysis of the curves' frequency spectrum. A more direct approach is preferred for this work, namely the Simple Moving Average (SMA): SMA is a statistical technique that smoothens time series data, reducing noise and allowing for easier identification of patterns and trends. It relies on a sliding window that moves along the data and simply computes the arithmetic mean of the previous N samples, with N being the size of the sliding window. In our case, the SMA of a contraction profile P with window size N at frame t is:

$$\text{SMA}(N, t) = \frac{P_t + P_{t-1} + \dots + P_{t-(N-1)}}{N} \quad (3.3)$$

where the larger the window, the smoother the final curve. We wish to eliminate unwanted fluctuations but still preserve the overall shape of the profile, since it plays a major role in the identification of the peaks. Therefore, the choice of the optimal window size becomes crucial.

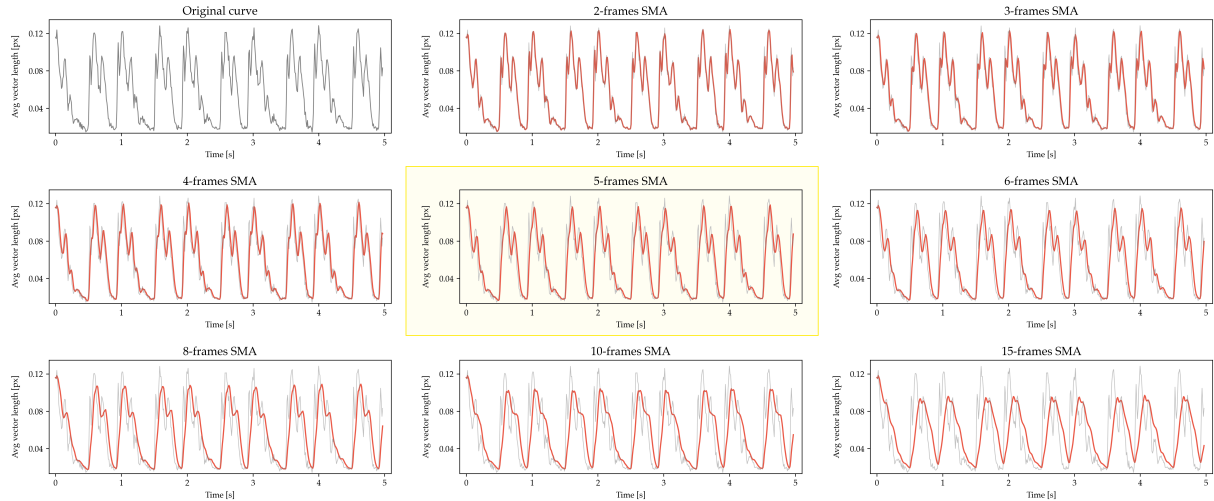


Figure 3.11: Effects of different SMA window sizes on a contraction profile. After numerous experiments, a time period of 5 frames was found to yield proper results.

Numerous experiments have been conducted to determine the most favorable value. As exemplified by [Figure 3.11](#), a small window does not help to reduce the high-frequency noise, while an exaggerated size introduces information losses that compromise the analyzability of the curve. The optimal balance was found with a time period of 5 frames, which allows to obtain smoother curves that yet maintain their pre-SMA properties. Indeed, such value greatly improves the insightfulness of noisy profiles (see [Figure 3.12b](#)) while preserving the quality of already-clean ones ([Figure 3.12a](#)).

Finally, one may notice that SMA dampens the amplitude of the peaks. For our work this is actually tolerable: on the one hand, the *frequency* and *velocity* constraints do not rely on the peaks amplitude; the *amplitude* constraint, on the other hand, is not affected since the relative height between contraction and relaxation peaks is maintained.

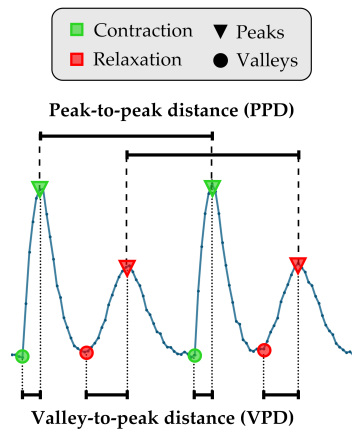


Figure 3.13: Notation used for the peak detection.

3.6.2 Peak detection

To check if a contraction profile fulfills the four previously mentioned conditions, it is sufficient to retrieve the x-position and the respective y-axis value of peaks and valleys. Then, following the notation indicated in [Figure 3.13](#):

- Conditions 1 and 2 (the *frequency constraints*) are checked by comparing the time that elapses between subsequent contractions and relaxations. Ideally, the Peak-to-Peak Distance (PPD) of both phases is constant and equals 0.5 seconds (50 frames).

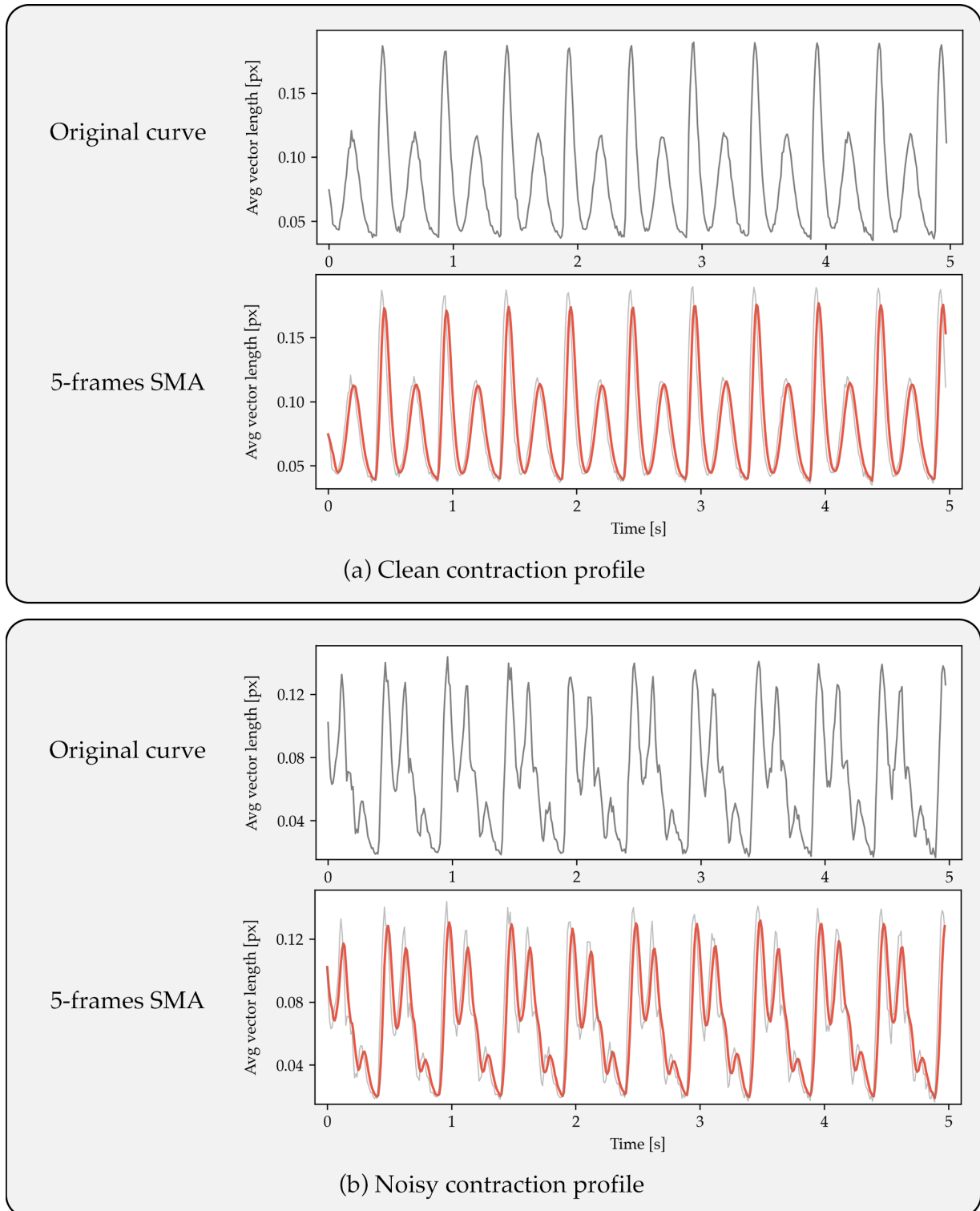


Figure 3.12: 5-frames SMA applied to different contraction profiles. SMA improves the quality of the noisy curves (Fig. b) while preserving the quality of already-clean ones (Fig. a).

Parameter	Description	Value (Peaks)	Value (Valleys)
Minimal height	Minimal required height of the peaks	$\min(P) + 0.425 \cdot (\max(P) - \min(P))$	$\min(-P) + 0.3 \cdot (\max(-P) - \min(-P))$
Prominence	Minimal vertical distance between the peak and its lowest contour line	$\max(P) \cdot 0.05$	$ \max(-P) \cdot 0.4 $
Distance	Required minimal horizontal distance in samples between neighbouring peaks	5	7

Table 3.2: Settings of the `find_peaks()` function used to peaks and valleys. The filtering parameters used are: (1) *minimal height*, the minimal required y-value of the peaks. For the peaks, this value is set to 40% of the normalized profile, to detect important peaks while disregarding meaningless ripples (see Figure 3.12); a similar line of thought is used for the valleys. (2) *prominence*, which measures how much a peak stands out from the surrounding baseline of the signal. It is defined as the vertical distance between the peak and its lowest contour line and is set to 5% and 40% of the maximum profile value for peaks and valleys respectively. This allows to only focus on meaningful peaks, making the algorithm more robust. (3) *distance*, the minimum horizontal distance between neighboring pixels. Its value is set to 5 for positive and 7 for negative peaks, in order to neglect adjacent (noisy) fluctuations but yet be able to capture peaks deriving from an anomalous behavior of the tissue.

Note that all the values are chosen relatively to considered contraction profile to account for different behaviors of EHTs.

- The *amplitude constraint* (condition 3) is fulfilled if the y-value of each contraction peak is greater than that of the subsequent relaxation peak.
- Finally, the *velocity constraint* (condition 4) is assessed by comparing the Valley-to-Peak Distance (VPD) of contraction and relaxation. We expect $VPD_{\text{contraction}} < VPD_{\text{relaxation}}$.

Many peak detection algorithms have been proposed in the literature. An article by Tournade⁴⁴ presents an extensive overview of those specifically available in Python, with a particular focus on their advantages and disadvantages. For this work, the choice fell on the `find_peaks()` function of the Signal Processing library (`scipy.signal`), because of its simplicity and filtering support. The function searches for local maxima based on simple value comparison of neighboring samples and returns those peaks whose properties match optionally specified conditions (such as minimum height, width and distance to each other). It is implemented as follows:

⁴⁴ Tournade (2019)

Detection of peaks. First, all peaks (with no distinction between contraction and relaxation) are detected by the chosen algorithm. The SMA-smoothed contraction profile is analyzed using the settings (defined by experiments) shown in Table 3.2 under “Value (Peaks)”.

Peaks differentiation. Not only do we need to detect all peaks, but also to determine whether they belong to the contraction or relaxation phase. To perform such task, the algorithm first ascertains if the profile starts with a contraction or with a relaxation; then, it assigns each peak to a different class in an alternating manner. This method relies on (1) the assumption that each contraction is followed by relaxation, but this always holds; (2) the correctness

of the peak identification, meaning that we assume that only *true* peaks (and not potential ripples caused by noise) are detected.

Detection of valleys. One drawback of the chosen peak detection algorithm is that it can not detect local minima directly. Therefore, the contraction profile is vertically mirrored (by multiplying it times -1) and analyzed again. See [Table 3.2](#) under “*Value (Valleys)*” for the fine-tuned parameters.

Qualitative classification. Finally, the conditions are checked. The algorithm generates a text file containing the name of the video, the qualitative conclusions and other relevant information such as the date and time. For example, for a correctly behaving tissue, the text file contains:

```
----- ASSESSMENT RESULTS FOR VIDEO: videoname -----
High peaks freq = 2Hz:    ✓
Low peaks freq = 2Hz:    ✓
High peaks > low peaks:  ✓
Velocity constraint:     ✓

dd-mm-yyyy hh:mm:ss
-----
```

3.7 Configuration of the workflow with specific OF estimation methods

The previous sections outlined the proposed workflow for extracting and assessing the contractile properties of EHTs. However, the final user should be aware that different OF estimation techniques may perform differently using the pipeline as is. The work is intended to be a general framework that can undergo adjustments and variations in order to accommodate specific OF methods. In fact, as will be evidenced in the [Experiments and Results](#) section, this is also true for the chosen algorithms: FlowNet2 requires post-processing of the OF fields to neglect irrelevant motion vectors, while Farnebäck necessitates an iterative refinement to achieve sufficient robustness.

3.8 Assessment of OF estimation method performance

This thesis not only implements three different optical flow algorithms, but it also compares their performance on the given dataset. However, being the first study of its kind, ground truth data or analogous results (which would enable a direct assessment of the algorithm's performance) are not available.

To gauge the applicability of a specific OF estimation method, *contraction kinetics* can provide valuable insights. Recall that in [Section 3.6](#) we delineated several criteria to assess the behavior of a tissue from the information contained in its contraction profile: actually, [conditions 3](#) and [4](#) *always* hold – even if the tissue misbehaves – due to the underlying kinetics. Thus, the appropriateness of a specific OF algorithm on the dataset can be determined by inspecting if the resulting contraction curves respect the two conditions.

4.1 Workflow with Block-matching

Block-matching is a motion estimation method that returns a more or less sparse output depending on the chosen parameters. As such, block-matching algorithms (BMAs) could potentially violate the 3rd choice criterion. An extensive survey of the existing literature, however, highlighted their abundant use as non-invasive 2D hPSC-CMs quality assays^{15,17,18} and for this reason, a BMA will be used for analysis and comparison purposes.

Because of the idea behind it, block-matching is very computationally expensive. Consequently, among the available Python-written implementations, the fast Three Step Search algorithm by Banuru⁴⁵ was selected (refer to [Appendix A.1.1](#) for the underlying theory).

⁴⁵ Banuru (2020)

The workflow proposed in the [Methodology](#) section is combined with block-matching and applied to the given dataset. To start with and facilitate the OF estimation, the algorithm is given the video of a T1T (attached in [Supplementary Material 1](#)), i.e. a tissue undergoing abundant motion. Surprisingly, this BMA was immediately found to be unsuitable for our study: as shown in [Figure 4.1](#), the resulting contraction profile depicts pure noise. Individual contraction and relaxation peaks are impossible to identify and, additionally, fluctuations occur around an average of ca. 745.80 pixels. Considering that the vertical size of an image frame does not exceed 500 px, such value clearly suggests a fault in the OF estimation.

Nevertheless, many combinations of block sizes (N) and search areas (p) were subsequently tested in order to evaluate different configurations of the BMA. Yet, all returned wrong estimates on both T1Ts and T2Ts.

One could argue that the problem is likely to reside in the high temporal resolution of the videos, since a high frame rate induces

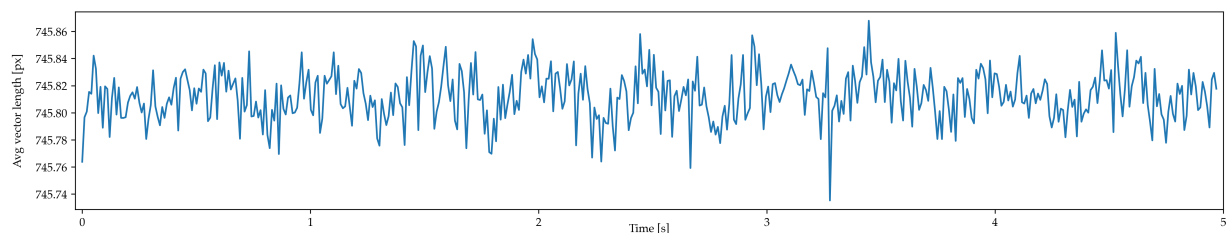
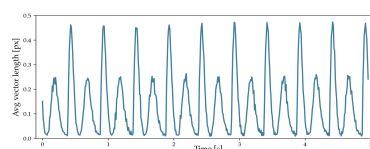


Figure 4.1: Contraction profile of a T1T obtained with block-matching (N = 16 px, p = 7 px). The curve is very noisy, making it impossible to distinguish individual contraction and relaxation peaks; additionally, the fluctuations occur at around 745.80 px which is clearly incorrect. The results are hence invalid.

⁴⁶ Huebsch et al. (2015)

⁴⁷ Reda et al. (2017)



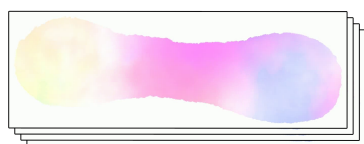
(a) Contraction profile



(b) Original motion maps



(c) Binary mask from BGR



(d) Masked motion maps

Figure 4.2: Results obtained on a T1T using FlowNet2. Although the contraction profile (Fig. a) seems correct, the motion maps (Fig. b) indicate a problem in the OF computation. To solve it, the OF fields are filtered using the binary mask from BG removal (Fig. c). The resulting masked map is shown in Fig. d.

sensor noise in the image. Combined with a window-matching approach that is surely sensitive to local noise patterns, it expectedly returns inaccurate OF estimates. However, the hypothesis is confuted by related works that prove the applicability of BMAs even in the presence of higher resolutions: Kunihiro et al. studied the contractile behavior of CM monolayers using 125 fps recordings¹⁵, while Hayakawa et al. conducted a similar study at 150 fps¹⁷. Huebsch et al., instead, investigated the effects of the camera frame rate on the resulting BMA-derived contraction profiles⁴⁶ and observed that, contrarily to the aforementioned supposition, a high temporal resolution is indispensable to obtain correct waveforms.

Hence, the problem is presumably intrinsic to the specific Python implementation. In fact, BMAs tend to be slow but are generally known to be applicable in a variety of situations, including – as supported by the available literature – in the analysis of CMs contractile motion. Unfortunately, the public internet lacks of Python-written BMAs so block-matching had to be disregarded and defined as not suitable for this study.

4.2 Workflow with FlowNet2

The implementation of FlowNet2 was made possible by the Python-compatible models provided by Reda et al. on GitHub⁴⁷. The repository offers the full pre-trained FlowNet2 model, as well as several sub-architectures of the neural network (e.g. FlowNetC, FlowNetCS, FlowNetSD etc. The reader is invited to consult [Appendix A.1.2](#) for an overview of the relevant theory). For this study, the complete FlowNet2 is chosen as the OF estimation method (since it is supposedly the most versatile version of the model) and it is integrated in the workflow as described in the [Methodology](#) section. Now the automatic assessment can be run and the results can be studied.

To better analyze the performance of FlowNet2 on our dataset, two main experiments are conducted: the assessment of a tissue undergoing significant displacement (T1T) and one whose motion is barely perceptible (T2T).

4.2.1 Assessment of Type 1 Tissue

The recording of a T1T ([Supplementary Material 1](#)) is inputted into the general pipeline of [Figure 3.1](#). As expected, the outputs are (1) a series of HOOFs, (2) a contraction profile, (3) a motion map video and (4) a final qualitative evaluation. For now, let us focus on the contraction curve and motion maps.

Although the contraction curve seems correct at first sight (Figure 4.2a), the visual characterization clearly indicates a problem in the OF computation (Figure 4.2b). FlowNet2 is detecting substantial movement in the background, but this is actually impossible since the image sequence was pre-processed with BG removal. The problem also persisted with other checkpoints of the models: FlowNetSD, FlowNetC and FlowNetCS were used for inference but yet returned similar results. Therefore, there must be a smoothing process responsible for the propagation of the detected motion beyond the tissue boundaries, causing wrong estimates and invalid results.

One can attempt to restore (to some extent) the quality of the OF estimates by recalling that we used a binary mask to perform BG removal, as demonstrated in Section 3.2.3. Then, the same mask (Figure 4.2c) can be applied to the OF fields to keep the vectors within the tissue and neglect those belonging to the background. The result of such filtering process is shown in Figure 4.2d.

Refining the OF vectors using the BG removal mask definitely improves the meaningfulness of the results: the amplitude of the processed profile is basically halved, as seen in Figure 4.3a. However, this does not solve the problem. The applied mask is static and, as such, it does not account for any movement occurring at the edges. Still, by introducing this adjustment in the overall pipeline, FlowNet2 is able to analyze T1Ts and to draw qualitative conclusions on the basis of the contraction profiles, as exemplified in Fig. 4.3b.

4.2.2 Assessment of Type 2 Tissue

Next, the adjusted pipeline (as schematically illustrated in Figure 4.6) is used to analyze a video of an EHTs whose displacement is very minimal, i.e. what we defined T2T or Type 2 Tissue (the concerned recording is available as Supplementary Material 2). OF fields are filtered with the BG removal binary mask, and the resulting contraction profile is plotted. This is shown in Figure 4.4, together with the SMA-smoothed version.

In this situation, FlowNet2 is not capable of detecting the contractile motion with the needed sub-pixel precision: the shape of the profile is irregular and individual contraction/relaxation peaks can not be distinguished. To investigate whether the problem is associated with the given pre-trained weights, different checkpoints of the model were tested. In particular, FlowNetSD (= FlowNet Small Displacement) seemed a promising alternative, since this part of the network is specifically fine-tuned to better handle small

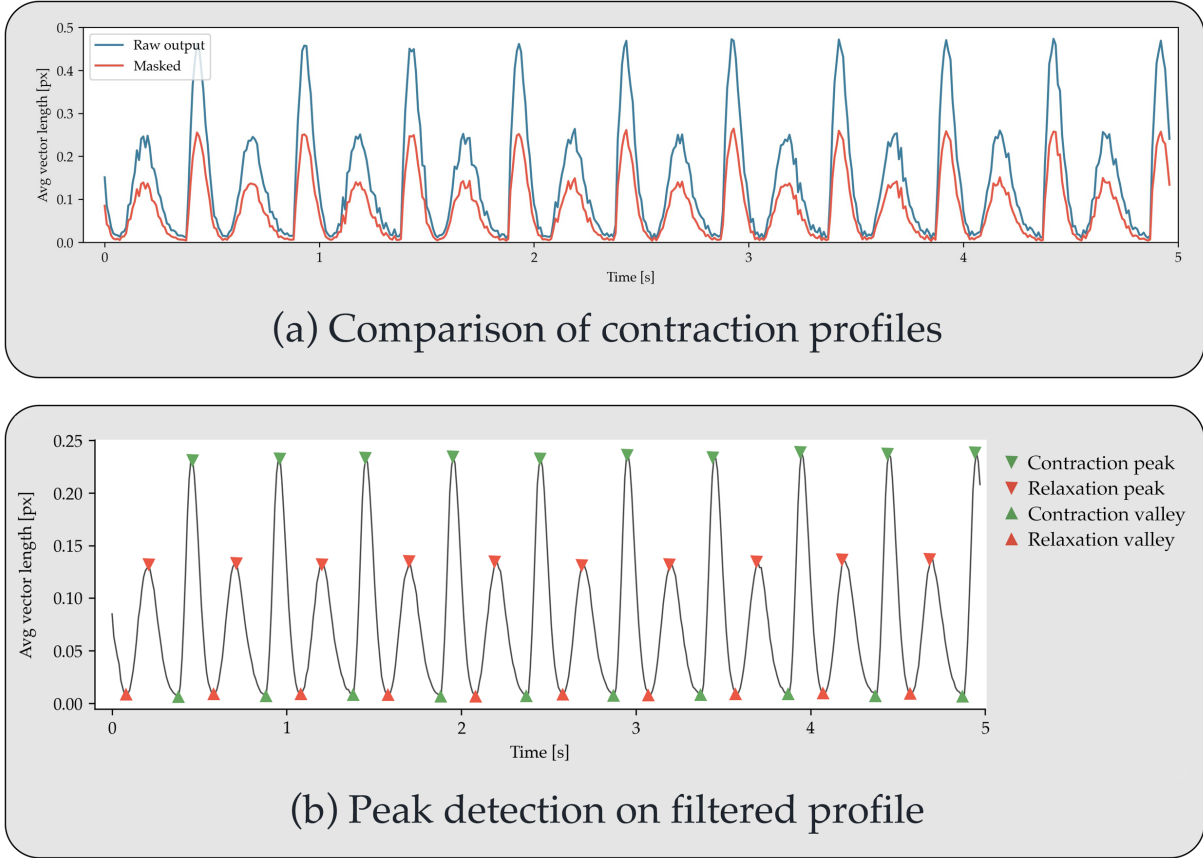


Figure 4.3: (a) Contraction profiles before and after filtering FlowNet2 OF fields with the BG removal binary mask. Note that the final contraction profile is almost halved compared to the initial one. (b) Results of the peak detection algorithm, which is run on the mask-filtered contraction profile after smoothing it with a 5-samples SMA (as discussed in the [Methodology](#) section).

displacements and sub-pixel motion. Unluckily, it performed no better than FlowNet2.

A theoretical solution to this problem is frame sampling. Sampling the frames of the original video means reducing the number of fps and effectively increasing the displacement detected between subsequent images. A separate experiment is conducted to examine the effects of frame sampling on the contraction profile of T2Ts.

4.2.3 Effects of frame sampling on T2Ts

Frame sampling could theoretically help to analyze the barely-visible displacement that T2Ts undergo. However, reducing the number of fps causes the loss of motion information that would only be captured at a high frame rate. To restore at least the original 100 fps temporal resolution, the resulting contraction profiles are interpolated.

The experiment of [Section 4.2.2](#) is repeated for different sampling frequencies: 1 (no sampling, resulting in 100 fps), 2 (50 fps), 3 (ca. 33 fps) and 4 (25 fps, close to the standard of 24 fps). Then, the

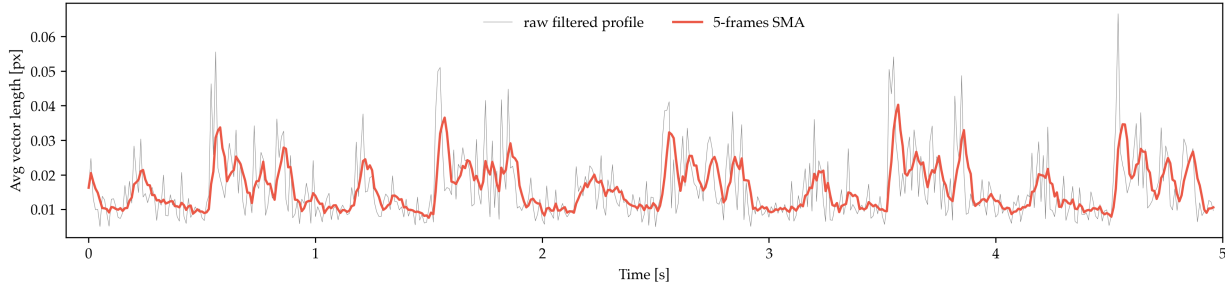


Figure 4.4: Contraction profile of a T2T obtained with the adjusted FlowNet2 pipeline (i.e. using the binary mask of BG removal to filter the OF fields) together with its SMA-smoothed counterpart.

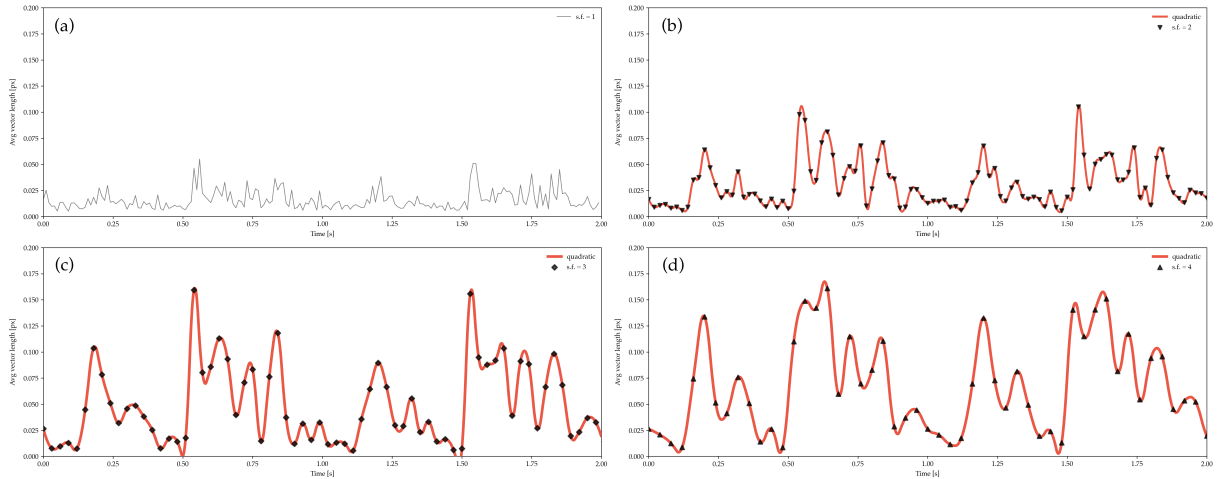


Figure 4.5: Interpolated contraction profiles of a T2T at different sampling frequencies. (a) shows the curve of the non-sampled video (sampling frequency = 1), while (b), (c) and (d) are derived from the same recording but with a sampling frequency of 2, 3 and 4 respectively.

contraction curves of the sampled videos are obtained by quadratic interpolation (the choice of the interpolation method is carefully reasoned in [Appendix A.2](#)). These are displayed in [Figure 4.5](#).

Although frame sampling serves its purpose and indeed strengthen the magnitude of the displacement (note how the amplitude of the peaks increases with the sampling frequency), it does not bring any benefit with respect to the overall quality of the profile. In fact, the peaks are still not clearly distinguishable and sometimes they do not even respect the contraction kinetics.

FlowNet2 performs decently on T1Ts but poorly and unreliably on T2Ts. This limit opposes one of the requirements of the motion analysis, i.e. robustness, and therefore makes this pre-trained version of FlowNet2 unsuitable for the given dataset.

4.3 Workflow with Farnebäck

The Gunnar Farnebäck's algorithm requires the user to specify a number of input parameters, which were experimentally tuned

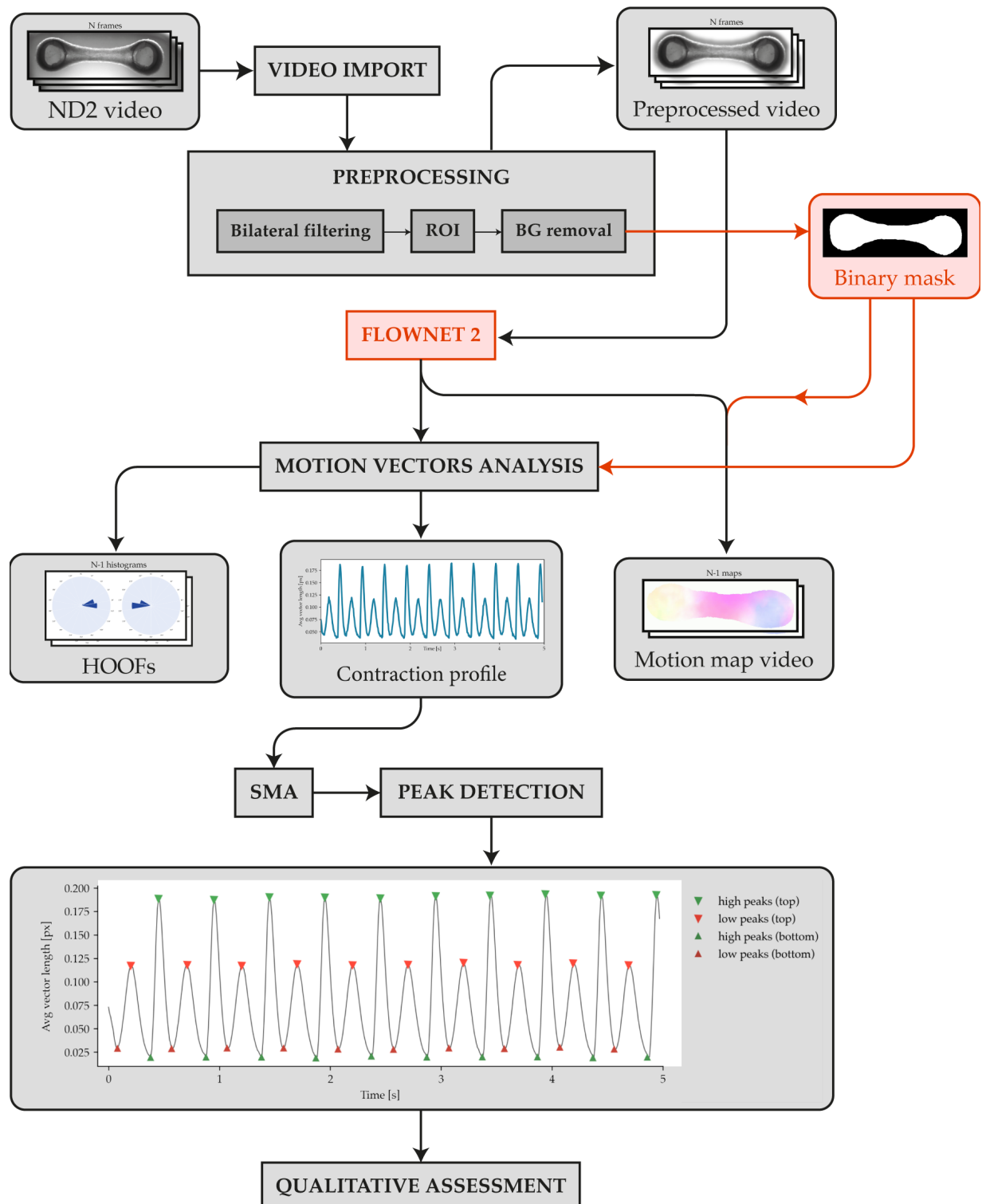


Figure 4.6: Adjusted workflow of the EHT motion analysis using FlowNet2. The red-colored parts correspond to additional steps added on top of the original workflow of Figure 3.1. Here, the binary mask used for BG removal is also applied to the OF fields to clean the OF estimates and only keep the vectors sitting within the tissue boundaries.

Parameter	Description	Value
pyr_scale	Specifies the image scale (<1) to build pyramids for each image; pyr_scale=0.5 means a classical pyramid, where each next layer is twice smaller than the previous one.	0.8
levels	Number of pyramid layers including the initial image; levels=1 means that no extra layers are created and only the original images are used.	0.5
winsize	Averaging window size; larger values increase the algorithm robustness to image noise and give more chances for fast motion detection, but yield more blurred motion field.	Varying
iterations	Number of iterations the algorithm does at each pyramid level.	5
poly_n	Size of the pixel neighborhood used to find polynomial expansion in each pixel; larger values mean that the image will be approximated with smoother surfaces, yielding more robust algorithm and more blurred motion field. Typically poly_n =5 or 7.	7
poly_sigma	Standard deviation of the Gaussian that is used to smooth derivatives used as a basis for the polynomial expansion; for poly_n=5, you can set poly_sigma=1.1; for poly_n=7, a good value would be poly_sigma=1.5.	1.5

Table 4.1: Parameters of the Farneback algorithm (nomenclature based on the OpenCV function `calcOpticalFlowFarneback`), with a short description of their role and their experimentally-defined values.

and reported in Table 4.1 (further details about the relevant theory in Appendix A.1.3). The most delicate and influential parameter was found to be `winsize`, the averaging window size: larger values increase the algorithm robustness to image noise and give more chances for fast and small motion detection, but yield more blurred motion maps. The problem is that there exists no optimal trade-off value for the specific application of Farneback to our dataset, since T1Ts are – by definition – characterized by greater amplitude of motion whereas the displacement of T2Ts is barely perceptible. For instance, as Figure 4.7 shows, a window size of 40 pixels is adequate to extract motion information of a T1T but does not suffice for a T2T.

To render the motion analysis algorithm more versatile and robust to the technical variations of the dataset, an iterative approach is preferred. Optical flow is estimated iteratively with different `winsizes` until the resulting contraction profile is clean enough for further analysis. To quantify how clean the profile is, the Signal-to-Noise Ratio is used.

The Signal-to-Noise Ratio (SNR) is a statistical measure used to compare the level of the signal to the level of background noise. It is defined as the reciprocal of the coefficient of variation, i.e. the ratio of mean μ to standard deviation σ (see Equation 4.1) and it is expressed in decibels (dB).

$$\text{SNR} = \frac{\mu}{\sigma} \quad (4.1)$$

OF is iteratively estimated until the average SNR value reaches at least 1 dB (experimentally defined value); should the threshold not be reached, the optical flow is re-computed with a higher

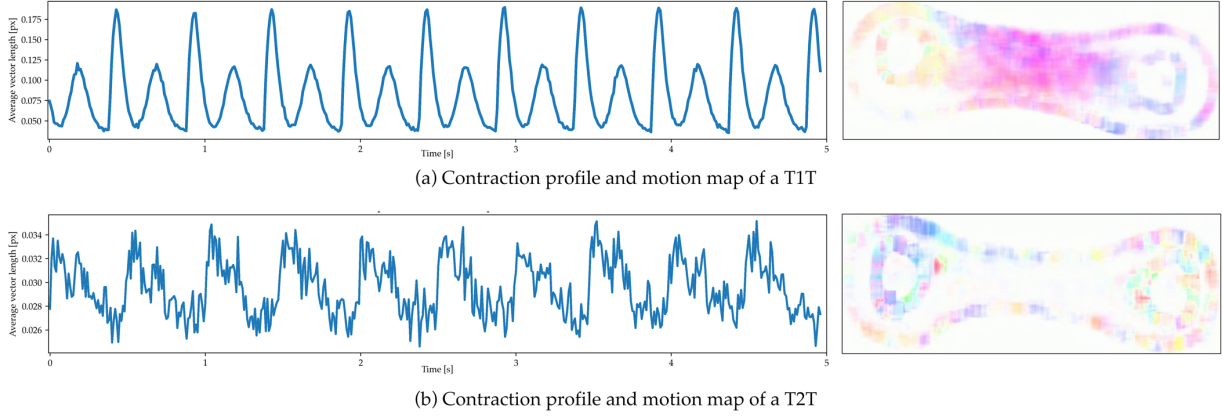


Figure 4.7: Contraction profiles and motion maps (extracted at time $x = 1s$) of the (a) T1T of [Supplementary Material 1](#) and (b) T2T of [Supplementary Material 2](#) obtained using Farnebäck and a window size of 40 px. For first, notice the huge difference of scales on the y-axes of the contraction profiles and secondly, realize that the entirety of the T1T is undergoing significant displacement while T2T’s contractile motion is concentrated at the edges. Both facts support the definitions of T1T and T2T.

winsize (four possible window sizes have been empirically fixed: 40, 150, 300, 450), which enables to capture more information about the contractile motion. In the eventuality that all pre-established winsizes are utilized and $SNR > 1$ dB is never reached, frame sampling is applied to the video (as described in [Sec. 4.2.3](#)) and the iterative refinement is repeated, testing all winsizes again. The resulting pipeline is illustrated in [Figure 4.8](#) and the corresponding pseudocode is presented under [Algorithm 1](#). Before proceeding to the tissues analysis, let us investigate the influence of different window sizes and sampling frequencies.

4.3.1 Influence of different window sizes

To study the effects of different window sizes on the final contraction profile, the same tissue of [Figure 4.7\(b\)](#) (the reader can find the corresponding video in [Supplementary Material 2](#)) was analyzed using the iterative algorithm. The stopping criteria (average SNR value greater than 1) was reached after 3 iterations with winsize = 300, as shown in [Figure 4.9\(d\)](#).

[Figure 4.9\(a\)](#) shows the average vector length together with its standard deviation. Note how the variations around the mean are reduced with a greater winsize: this is expected, since larger windows result in smoother and more blurred OF fields that suppress variations due to noise. These drastic reductions of the standard deviation are also crucial to reach convergence, as they are directly related to higher SNR values (recall [Eq. 4.1](#)).

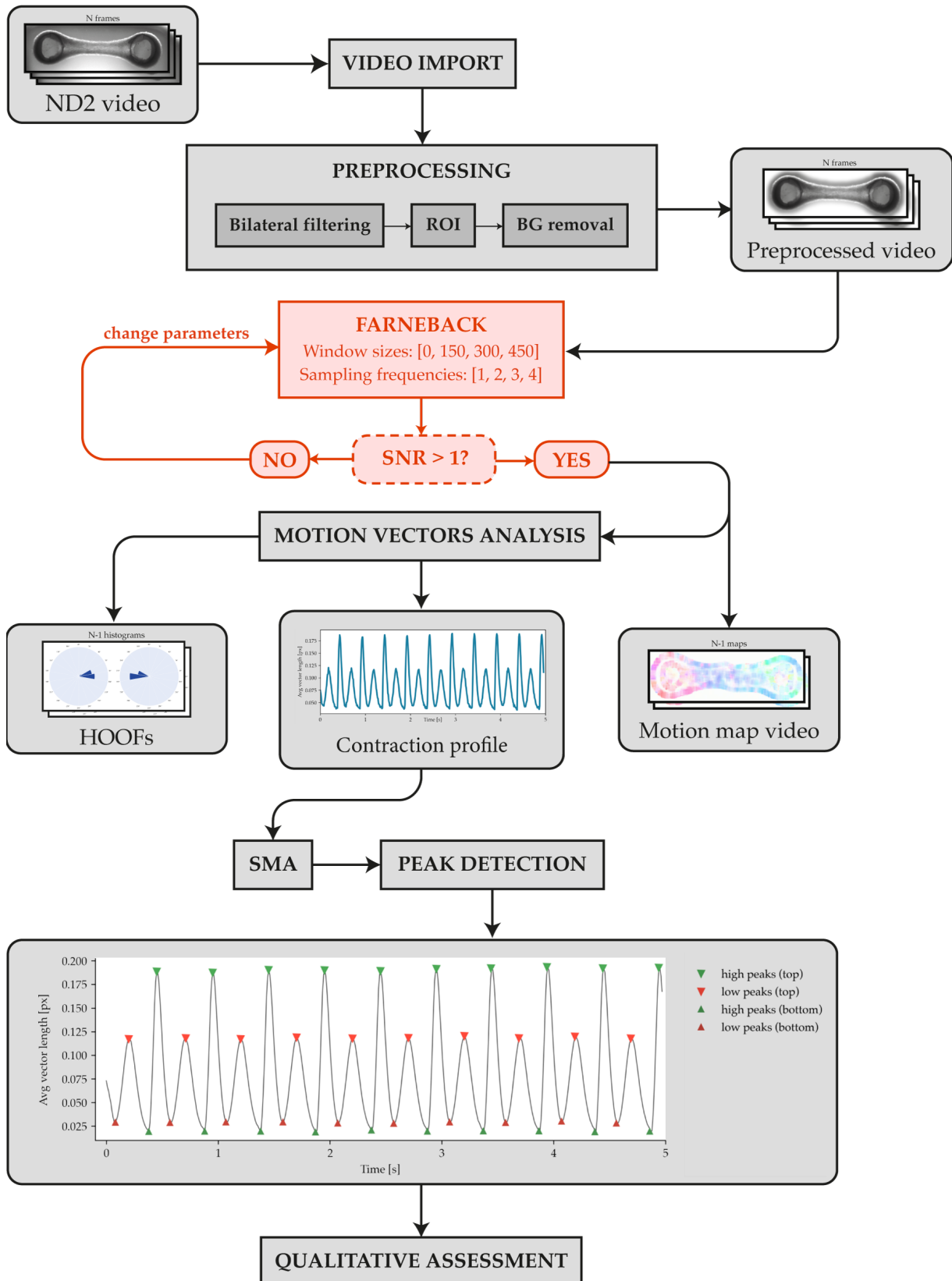


Figure 4.8: Adjusted workflow of the EHT motion analysis using Farneback. The red-colored parts correspond to additional steps added on top of the original workflow of Figure 3.1. The most important addition is the iterative approach, which estimates OF until SNR suggests that the resulting contraction profile is clean enough for further analysis.

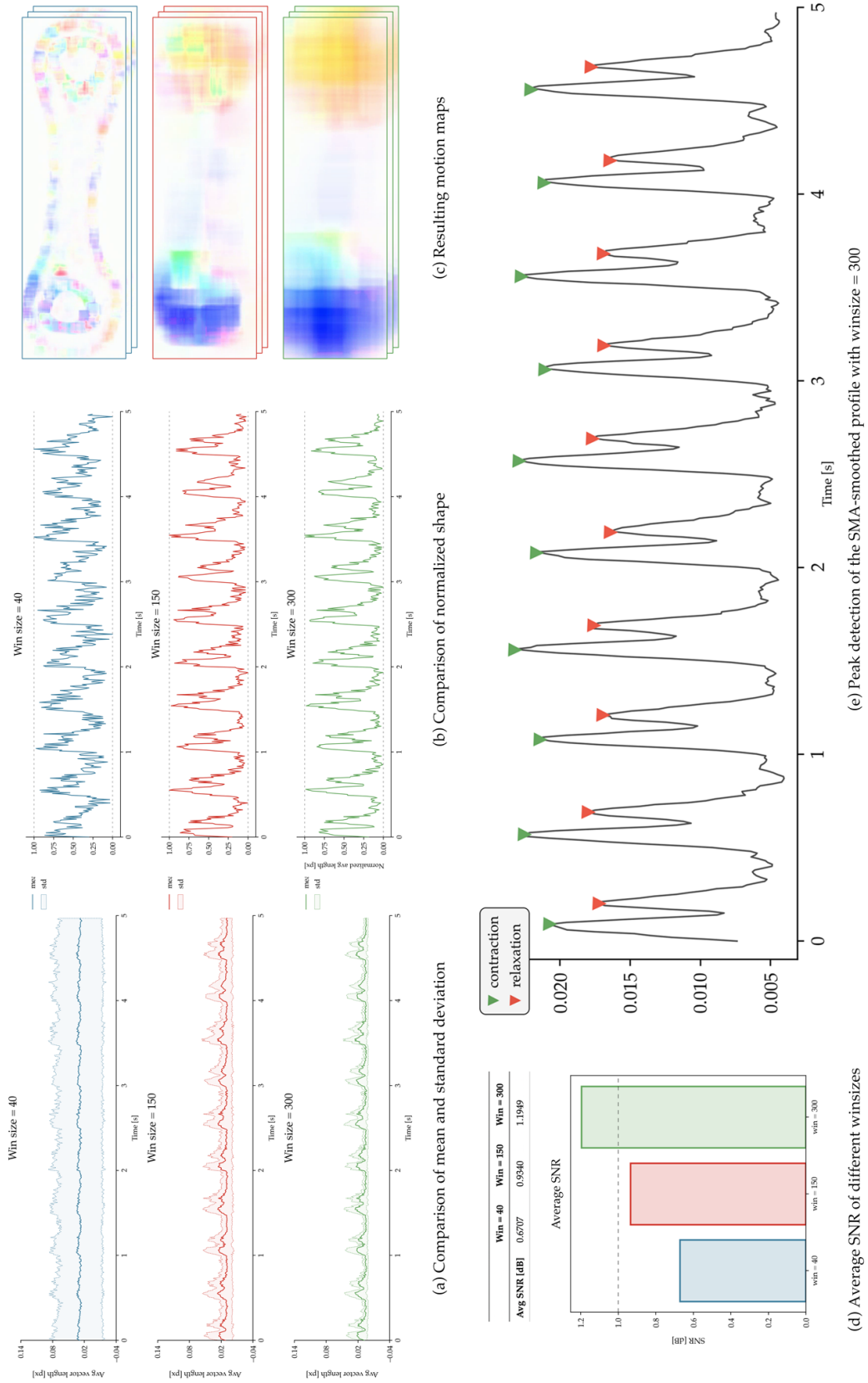


Figure 4.9: Influence of the Farneback window size (blue: 40; red: 150; green: 300) on the resulting contraction profile. Figure (a) shows the average vector length together with its standard deviation; (b) compares the profile obtained with different windows and (c) shows the effect of each window on the motion map (imaged at 1s). Finally, (d) illustrates the iterative process that was performed to obtain the curves: OF was estimated until the average SNR reached least 1 dB. After 3 iterations the stopping criteria was met and peak detection was performed (see Fig. e)

Figure 4.9(b), instead, compares the profiles shape-wise. The curves are normalized to facilitate observations and to focus attention on their shapes, clearly showing that the meaningfulness and analyzability of the profiles are greatly enhanced by larger windows. Indeed, the curve obtained in the last iteration (that is, the one that moves further down the pipeline) can be easily and correctly processed by the peak detection algorithm, as shown in Figure 4.9(f).

Finally, Figure 4.9(c) highlights the trade-off between the meaningfulness of the contraction profiles and the blurriness of the motion maps.

Algorithm 1: Farnebäck iterative motion analysis

```

1 import nd2 file;
2 preprocess image sequence;
3 while SNR < 1 and you can increase winsize or samp_freq do
4     Compute OF for all image pairs;
5     Calculate SNR for each OF field;
6     Compute average SNR;
7     if avg SNR > 1 then
8         break;
9     else
10        if all winsize-samp_freq combination were tried then
11            break;
12        else
13            if other winsize values still available then
14                choose next winsize value;
15            else
16                increase samp_freq and restore initial winsize

/* considering the OFs of the last iteration: */
17 generate motion map video;
18 generate HOOFs;
19 generate contraction profile;
20 smooth profile using SMA;
21 detect peaks;
22 final assessment of the contractile properties;

```

4.3.2 Influence of frame sampling

As already done multiple times, the 100 fps time resolution of the contraction profiles is restored by quadratic interpolation. Figure 4.10 shows the effects of such interpolation method on the contraction curves obtained with sampling frequencies of 1, 2, 3 and 4 and a winsize of 40 px. First of all, Figure 4.10(a) demonstrates that – contrarily to what observed with FlowNet2 – frame sampling

does help Farnebäck capture more information about the contractile motion of T2Ts. Figure 4.10(b), instead, shows that increasing the sampling frequency induces an increase of the standard deviation of the estimated motion. This is to be expected, since the window size remains constant throughout the experiments and nothing changes apart from the magnitude of the displacement. However, as a consequence, the average SNR increases much more slowly between iterations and hence the convergence time is increased. This is why different winsizes are given priority over different sampling frequencies, as seen in Algorithm 1.

4.3.3 Assessment of T1Ts and T2Ts

The proposed SNR-based iterative approach makes the motion analysis robust and versatile. In fact, for all 81 videos of the dataset the algorithm successfully extracts contraction curves that (1) present well distinguished peaks, meaning that the *frequency condition* (the main indicator of correct functioning) can always be checked and (2) respect the *amplitude condition* enforced by the underlying kinetics (each contraction peak is higher than the subsequent relaxation peak). The kinetics-governed *velocity constraint* can unfortunately only be assessed for the profiles of T1Ts. These are richer in motion information than T2Ts, since – as could be seen in Figure 4.9(f) – the latter often feature relaxation peaks that overlap with those of contraction. As a consequence, it impedes the detection of *correct* relaxation valleys (see Figure 4.11) and thus to check the velocity condition.

Supplementary Material 3 and 4 show the output of the algorithm on a T1T and T2T respectively.

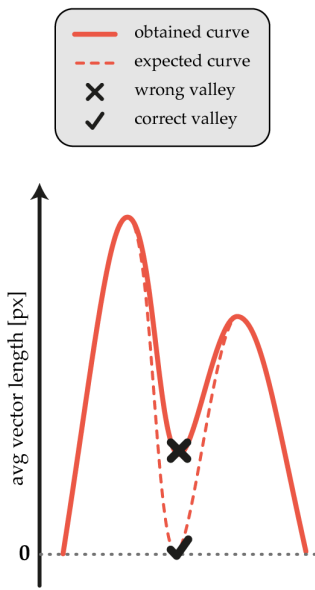
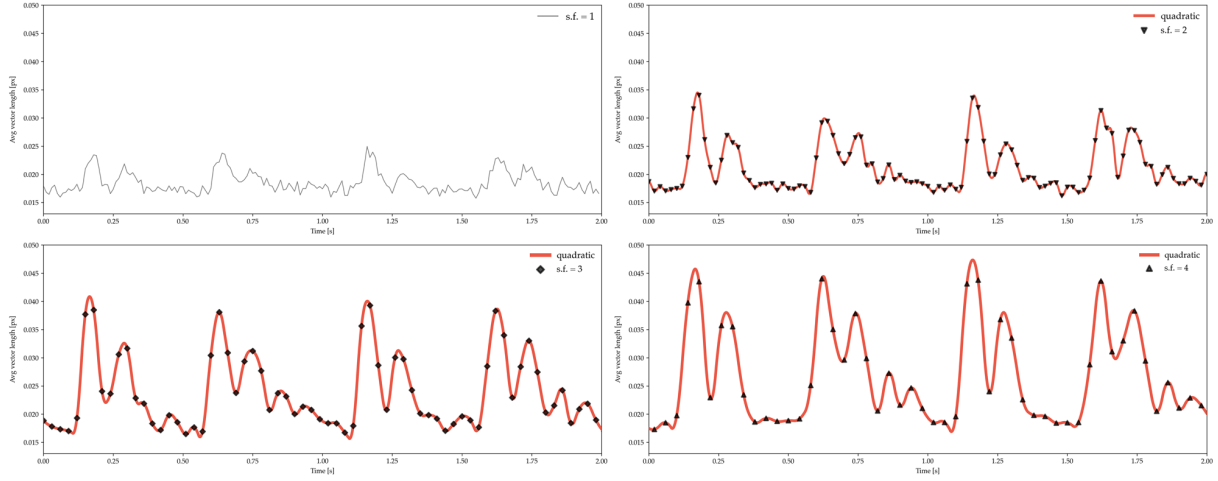


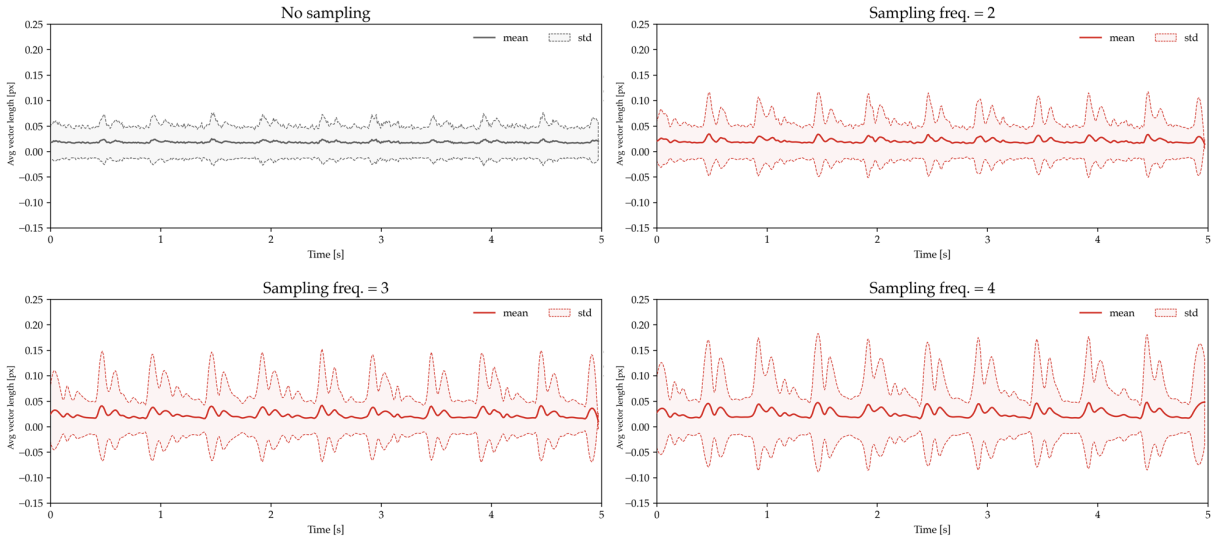
Figure 4.11: Impossibility to detect the correct valley in the contraction profiles of T2Ts. The relaxation peak overlap with that of contraction.

4.3.4 Histogram of Oriented Optical Flow (HOOFs)

Although not involved in the final assessment of the contractile properties, HOOFs still reveal valuable information of EHTs hidden in the videos. For instance, they are especially helpful to determine the dominant direction of motion of T2Ts, where the displacement is scarcely detectable. Supplementary Material 5 demonstrates such situation.

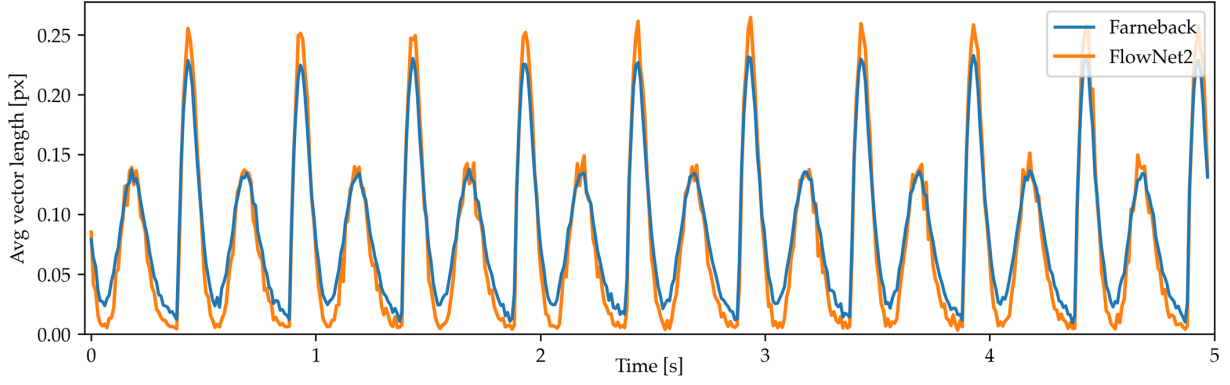


(a) Interpolated contraction profiles

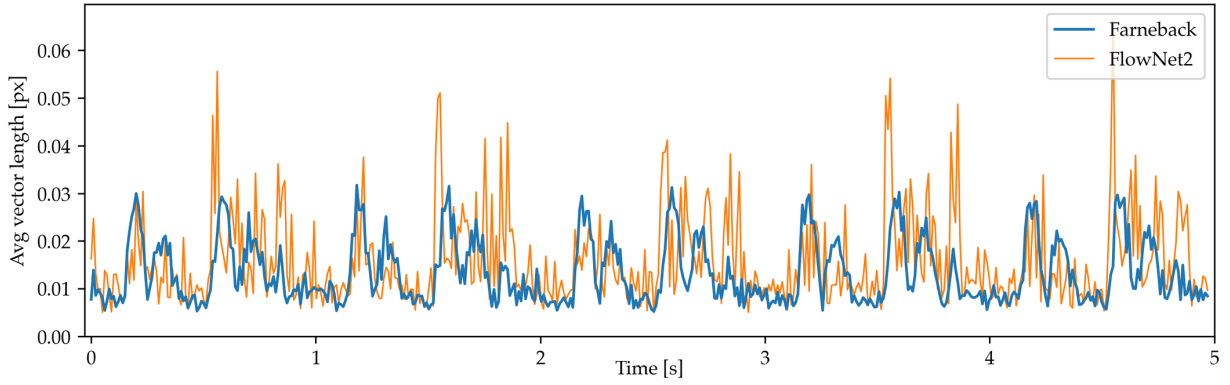


(b) Mean and standard deviation

Figure 4.10: Study of a T2T's contraction profile obtained using the same window size (= 40) and various sampling frequencies (1, 2, 3, 4). Fig. (a) shows the interpolated curves and emphasizes the different peak amplitudes and shapes. Fig. (b), instead, shows the average vector length and its standard deviation. It is interesting to notice that the latter does not undergo major changes as observed when studying the effects of different window sizes (see [Figure 4.9a](#))



(a) Contraction profiles of a T1T



(b) Contraction profiles of a T2T

Figure 4.12: Comparison of the contraction curves obtained using FlowNet2 and Farneback on (a) T1T (whose video is available as [Supplementary Material 1](#)) and (b) T2T (also available under [Supplementary Material 2](#)). The results of block-matching were not included on purpose.

4.4 Comparison of performance

We have observed that block-matching is not suitable for this study, FlowNet2 is limited to T1Ts and Farneback is the most versatile algorithm. Let us now focus on these last two and compare their performance.

The recording of a T1T ([Supplementary Material 1](#)) has been analyzed with both FlowNet2 and Farneback and the resulting contraction profiles have been plotted in [Figure 4.12\(a\)](#). Several things can be observed:

- The shape of the two waveforms is very similar. In view of a peak detection-based qualitative characterization, the performance of FlowNet2 and Farneback on T1Ts is alike;
- The curve obtained with FlowNet2 reaches zero in between cycles. Contrarily, the average vector length of Farneback is never completely zeroed, possibly due to some leftover noise;

- The oscillations captured by FlowNet2 have greater amplitude compared to those extracted by Farnebäck. Such behavior is expectable: FlowNet2 fields have been filtered with the binary mask generated earlier during BG removal, which – as explained in [Section 3.2.3](#) – is derived from an edge selection that is later expanded to account for possible motion. Resultantly, the profile does not only contain the motion vectors belonging to the EHT, but also those lying in the immediate vicinity of the tissue boundaries; these extra vectors are likely to contribute to greater amplitude.

[Figure 4.12\(b\)](#) compares the profiles obtained with the two different OF estimation methods on the recording of a T2T ([Supplementary Material 2](#)). FlowNet2 is unsuitable for this analysis, while a careful choice of Farnebäck’s parameter allows the algorithm to be robust and versatile.

4.5 Results of classification

As already proved, the iterative Farnebäck approach extracts meaningful contraction profiles for all 81 videos. However, such curves still have to go through the peak detection step, which eventually determines the correctness of classification.

A faulty identification of the peaks results in the classification of the tissue as *misbehaving*, since the time position of contractions and relaxations are incorrect. For this reason, the implemented peak detection algorithm was finely tuned to analyze correctly-operating EHTs, whose contraction profiles exhibit distinct peaks (see [Figure 4.13a](#) and [b](#)). Yet, it is prone to failure in the presence of anomalous spikes and extra cycles ([Figure 4.13c](#)), but this is negligible since these abnormal situations are an indicator of faulty behavior even upon accurate peak detection.

In fact, let us compare the robustness of peak detection with that of the final qualitative categorization. To determine the accuracy of the peak detection algorithm, one could ideally investigate whether the kinetics-governed constraints are fulfilled. However, as already discussed in [Section 4.3.3](#), overlapping peaks of T2Ts make it impossible to verify the *velocity condition*. As a consequence, to establish a homogeneous judgment criterion that is valid for the entire dataset, the detection of the peaks is defined correct if the *amplitude condition* is satisfied. Peak detection was found successful in 59 out of 81 cases, yielding an accuracy of 72.8%.

To also evaluate the accuracy of the classification algorithm, one would need to compare the results of the motion analysis with the ground truth labels of the tissues. However, such labels do not exist

yet and researchers from the BIOS lab were unable to generate them within the timeframe given. To supply an approximate measure of performance, we can assume that the curves obtained with Farnebäck are correct, and manually categorize each tissue on the basis of the motion information contained in its contraction profile. Only then the results of the classification algorithm can be compared with those of the manual categorization to generate the confusion matrix shown in [Table 4.2](#).

Table 4.2: Results of classification summarized in a confusion matrix.

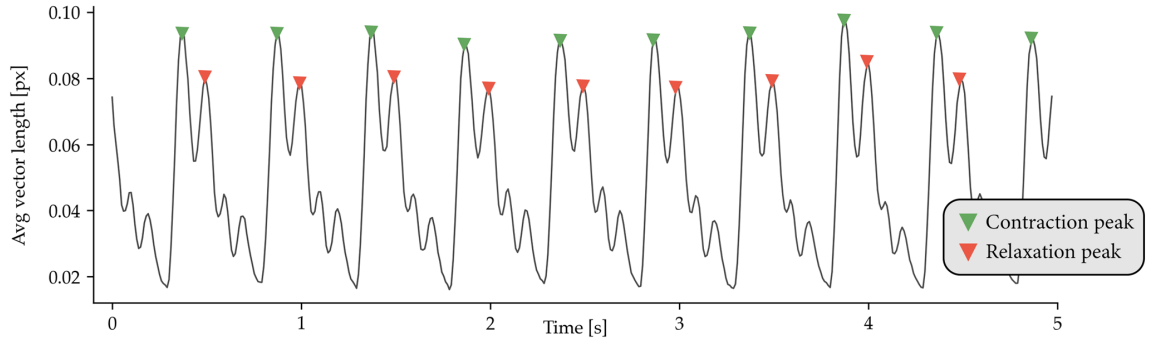
		Prediction	
		Positive	Negative
Ground truth	Positive	37	9
	Negative	0	35

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (4.2)$$

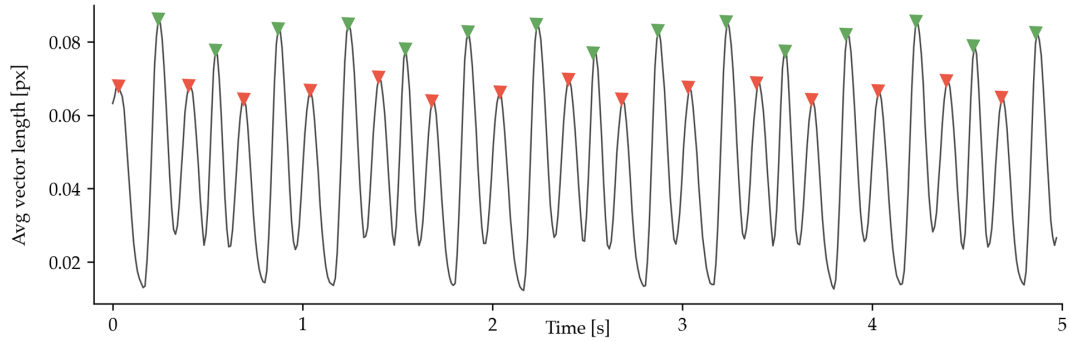
Several performance measures such as *accuracy*, *precision*, *recall* and *F1 score* could be extracted and interpreted. Since the ground truth data are based on an assumption, however, a detailed evaluation of the model is left for future work. Here we restrict the analysis of the confusion matrix to the *accuracy*, in order to enable a comparison with the peak detection algorithm. Thus, it follows from [Equation 4.2](#) (where T=True, F=False, P=Positive, N=Negative) that the accuracy of classification is 88.8%. Such value supports the hypothesis that peak detection performs poorly on misbehaving tissues but still (generally) succeeds to return a correct classification.

4.6 Algorithm limitations

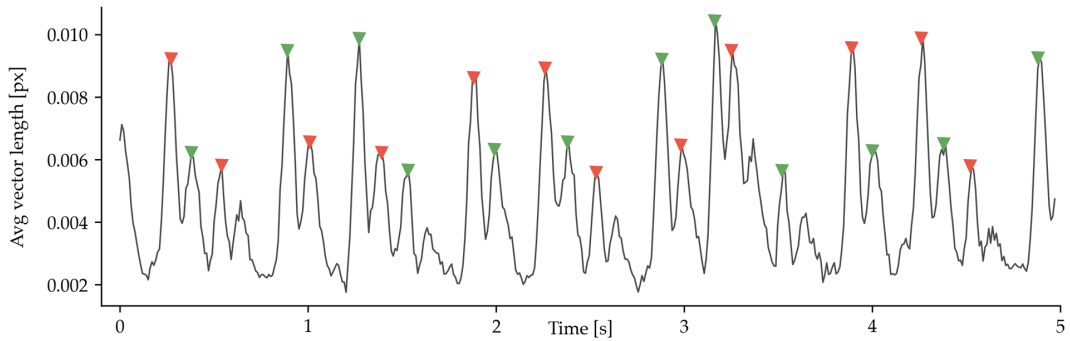
The proposed algorithm performs flawlessly under the condition that the EHTs undergo sufficient displacement. Although the SNR-based iterative refinement introduced significant improvements with respect to robustness and versatility of the model, the curves of T1Ts are still richer in information than those of T2Ts. In fact, the only limitation of the algorithm is that contraction and relaxation peaks overlap in the profiles of T2Ts, impeding the verification of the *velocity constraint*.



(a) Correct peak detection | Correct classification (*well-behaving*)



(b) Correct peak detection | Correct classification (*misbehaving*)



(c) Incorrect peak detection | Correct classification (*misbehaving*)

Figure 4.13: Contraction profiles, peak detection and consequent classification in different scenarios. The peaks are properly detected in (a) and (b), leading to the correct classification of the tissues as *well-behaving* and *misbehaving*, respectively. In (c) instead, peak detection fails: several contractions are marked as relaxation, and vice versa. Resultantly, peaks are not equally time-distant and the *frequency condition* is not fulfilled. The tissue is classified as *misbehaving* and actually, the tissue does indeed behave unexpectedly and causes the failure of peak detection. The qualitative categorization is thus not altered.

5.1 Discussion and future outlook

The results of the algorithm provide interesting points of discussion. First of all, the iterative procedure was never found to require a sampling frequency higher than 1. Larger window sizes were sufficient to reach the desired SNR threshold, meaning that frame sampling was never employed. Secondly, several contraction profiles contain ripples that compromise the correctness of the peak detection outcomes. Those fluctuations are certainly caused by noisy OF estimations, hence one could increase the SNR threshold to further enhance the performance of the algorithm. Of course, this is at the expenditure of computational time.

Future implementations could benefit from several other improvements that were precluded by the limited time frame. Firstly, we remarked the limitation of the algorithm to provide well-distinguished peaks in the profiles of T2Ts. Huebsch et al. attributed this problem to an insufficient temporal resolution⁴⁶, thus further works could experiment the effects of higher fps on the waveform shape. Additionally, Farnebäck was the only algorithm defined suitable for this study and it would be worth investigating variations of the other methods. A reliable block-matching algorithm could be written from scratch or ported from other available implementations (e.g. many MatLab-written algorithms can be found on MathWorks, see for example⁴⁸). FlowNet2, instead, could ideally be trained on the given dataset if the latter grows significantly in size. This would however require the labor-intensive task of generating ground truth data, which implies drawing a displacement vector for all pixels contained in each frame of each video. A demanding effort (with doubtful feasibility) that would yet bring study-worthy changes of FlowNet2 performance.

Moreover, there is particular room for improvement in the qualitative characterization phase. As already mentioned, the classification model requires a thorough evaluation based on reliable ground truth data. Besides, the simplest revision to the current implementation would be rendering the peak detection robust, possibly by testing different algorithms; instead, more drastic changes would be introduced by tackling the qualitative assessment as a binary classification problem (*correctly* and *incorrectly* behaving tissues), where the input features are either derived from the contraction profiles or HOOFs themselves. In any case, these features could eventually be projected on a low-dimensional Euclidean space

⁴⁸ Barjatya (2011)

(so that each recording is described by spacial coordinates) and clustered using unsupervised machine learning techniques. Presumably, a correlation between the behavior of the tissues and the drug that they were subject to would emerge.

Finally yet importantly, this work can be potentially extended to go beyond the "simple" characterization of the contractile properties and also assess the clinical utility of EHTs. In fact, clinically useful cardiac tissues respect the conditions imposed by the Frank-Starling law⁴⁹, the mathematical relation that plays a major role in treatment of CVDs. It describes the ability of the heart to change its force of contraction and stroke volume in response to changes in venous return⁵⁰; in other terms, the law relates the initial length of CMs to the force generated by contraction. Therefore, the studied contraction profiles can be further examined to derive velocity, acceleration and possibly forces of contraction. This would undoubtedly require additional developments but, combined with a separate study on the initial length of myocardial fibers, the workflow can mature into a valuable tool to determine the clinical usefulness of artificial cardiac tissues.

⁴⁹ Asnes et al. (2006)

⁵⁰ Lange et al. (2021)

5.2 Conclusion

The contractile properties of EHTs are directly related to their health and function. The current state of research offers a variety of methods to assess their beating behavior, but still lack of established contact-less techniques. This thesis outlined a novel optical-flow based workflow for the extraction, analysis and assessment of the contractile qualities of artificial cardiac tissues. Block-matching, FlowNet2 and Farnebäck were employed as OF estimation methods and consequently compared in terms of performance: block-matching was found unsuitable for this study; FlowNet2 was only applicable under specific circumstances while Farnebäck, combined with an iterative refinement, succeeded in reaching sufficient robustness. The estimated motion vectors were summarized in (1) Histogram of Oriented Optical Flow, (2) motion map videos and (3) contraction profiles. The latter underwent a SMA smoothing process before being analyzed with a peak detection algorithm, which correctly identified contraction and relaxation peaks with an accuracy of 72.8%. Despite the discrete robustness of peak detection, the accuracy of the classification based on the extracted characteristics of motion amounted to 88.8%. Finally, this thesis was temporally confined in 10 weeks but provided numerous insights and points of reflections that will hopefully incite future students to develop this work further or, at least, to undertake a similar journey.

APPENDIX

A

Appendix

A.1 Optical flow algorithms

The following sections focus on each of the three selected OF algorithms, diving into the technical and theoretical details of their implementation.

A.1.1 Block-matching

A block-matching algorithm is a way of locating matching macro-blocks in a sequence of video frames. It segments an image into blocks of size $N \times N$ pixels and assumes that each block undergoes independent uniform translation, described by the vector $\mathbf{v} = (dx, dy)$ seen in Figure A.1. Each macro-block is searched in the adjacent frame within a search window, centered around the initial block. The size of the search window is dictated by the "search parameter" p , which corresponds to number of pixels on all four sides of the reference macro-block (as illustrated in Figure A.2).

The best match is found by minimizing a cost function, whose output gives a numerical indication for the amount of mismatches. The most common criteria are⁵¹ *Mean Square Error* (MSE), *Mean Absolute Difference* (MAD), *Sum of Absolute Difference* (SAD) and *Peak to Signal Noise Ratio* (PSNR), described by Equations A.1, A.2, A.3 and A.4 respectively.

$$MSE(i, j) = \frac{1}{N^2} \sum_{i,j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (A.1)$$

$$MAD(i, j) = \frac{1}{N^2} \sum_{i,j=0}^{N-1} |C_{ij} - R_{ij}| \quad (A.2)$$

$$SAD(i, j) = \sum_{i,j=0}^{N-1} |C_{ij} - R_{ij}| \quad (A.3)$$

$$PSNR = 10 \log_{10} \frac{(\text{peak to peak value of data})^2}{MSE} \quad (A.4)$$

where C_{ij} and R_{ij} are pixels value of the current and reference macro-blocks.

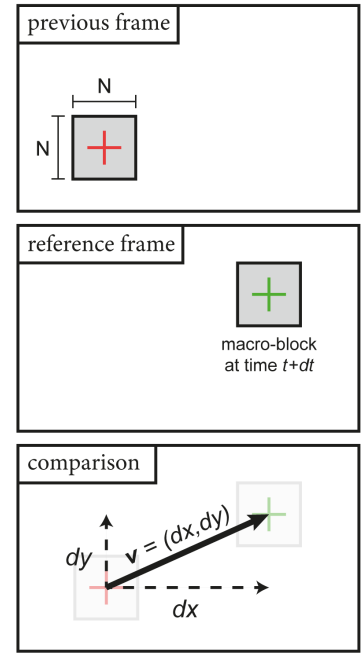


Figure A.1: Displacement vector $\mathbf{v} = (dx, dy)$ of an $N \times N$ px macro-block calculated from two consecutive video frames.

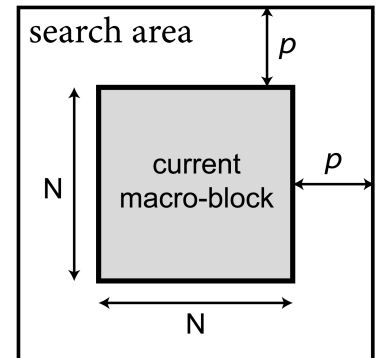


Figure A.2: Search window of a block-matching algorithm. A macro-block of size $N \times N$ px is surrounded by a search area that spans for p pixels around the four sides of the reference macro-block.

⁵¹ Khawase et al. (2017)

Many algorithms have been developed throughout the years, but this thesis implemented the Three-Steps Search. It runs as follows:

1. Start with search location at center;
2. Choose a step size S and search parameter p ;
3. Search 8 locations $\pm S$ pixels around location $(0,0)$ and the location $(0,0)$;
4. Among the 9 locations searched, pick the one with minimum cost function;
5. Set the new search origin to the above picked location;
6. Set the new step size as $S = S/2$;
7. Repeat the search procedure until $S = 1$

The resulting location for $S=1$ is the one with minimum cost function and the macro block at this location is the best match.

A.1.2 FlowNet2

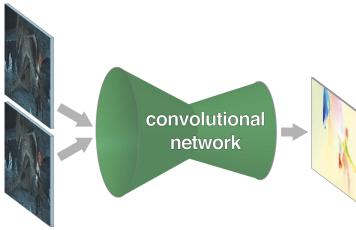


Figure A.3: Schematic representation of the FlowNet architecture: end-to-end neural networks trained to estimate optical flow from image pairs. Retrieved from³⁰.

FlowNet is the pioneering application of Convolutional Neural Networks for OF estimation. The first version (FlowNet 1.0 or simply FlowNet) was proposed by Dosovitskiy et al. in 2015³⁰ and proved that optical flow estimation can be cast as a learning problem. However, it could still not compete with traditional methods on small displacements and real-world data. FlowNet 2.0³¹ consolidated and enhanced the original FlowNet idea, drastically improving performance and bringing the algorithm on par with other state-of-the-art methods.

Since the version 2.0 highly depends on FlowNet 1.0, the latter will be examined first.

FlowNet 1.0. The idea of FlowNet is to take an end-to-end learning approach to predicting optical flow, meaning that a network is trained to predict the $x - y$ flow fields directly from the images (see Figure A.3). The proposed networks contain a **contracting part** and an **expanding part**. These are depicted – respectively – in Figures A.4 and A.5 and will be discussed individually.

Contracting part

This portion of the network uses pooling to aggregate information over large areas of the input images. Dosovitskiy et al. proposed two different approaches:

1. Stack the two input images together. Feed them through a generic network and let the network decide itself how to process the inputs and how to extract the motion information. This network, depicted in Figure A.4 (top) is named FlowNetSimple.

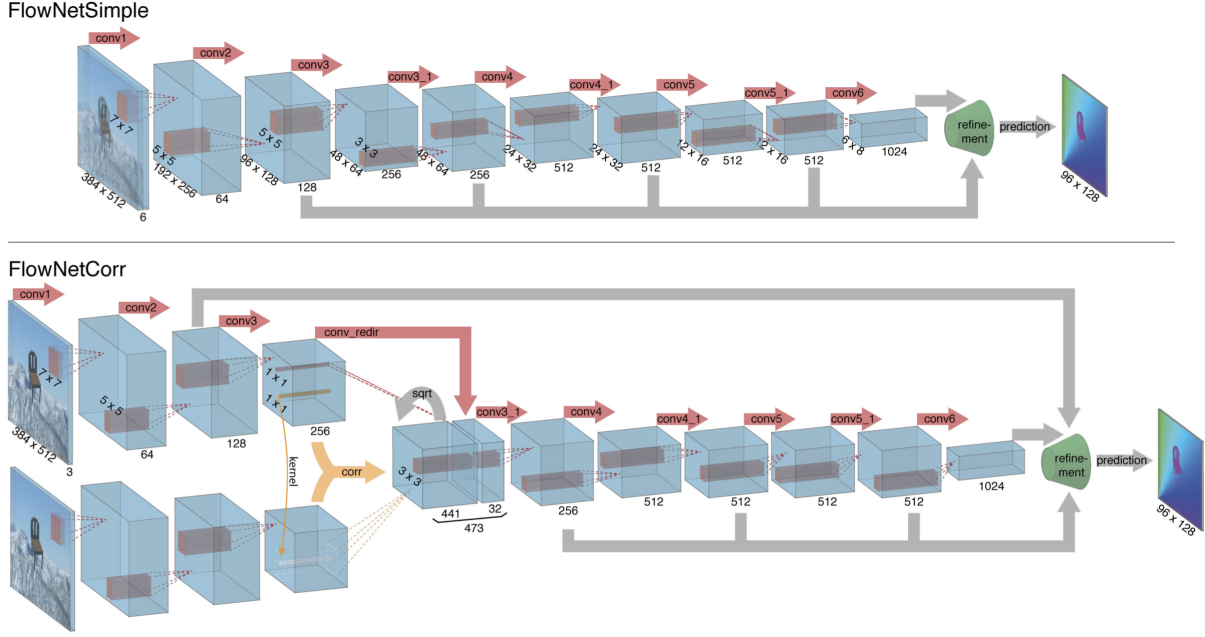


Figure A.4: The contracting part of the two network architectures: FlowNetSimple (**top**) and FlowNetCorr (**bottom**). The green funnel is a placeholder for the expanding refinement part shown in Fig A.5. Taken from³⁰.

2. Use the image pair as input for the two branches of a Siamese network. By using two parallel yet identical processing streams, the network is constrained to first produce meaningful feature representations of the two images separately and then combine them on a higher level, resembling the standard matching approach in which features are first extracted and then compared. In the case of this network, illustrated in Figure A.4 (bottom) and named FlowNetCorr, the correspondences of the two feature maps are extracted by a convolutional layer that performs multiplicative patch comparisons. More precisely, it compares patches from f_1 with patches from f_2 , where f_2 and f_1 are the two multi-channel feature maps $f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R}^c$ with w, h , and c being their width, height and number of channels.

For computational reasons, correlations $c(\mathbf{x}_1, \mathbf{x}_2)$ are only computed in a neighborhood of size $D := 2d + 1$, where d denotes the maximum possible displacement. Then the correlation of two patches centered at $\mathbf{x}_1$ in the first feature map and $\mathbf{x}_2$ in the second map is defined as:

$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o} \in [-k, k] \times [-k, k]} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle \quad (\text{A.5})$$

for a square patch of size $K := 2k + 1$.

Expanding part

As mentioned above, the contracting part is based on pooling. While it is essential for the aggregation of information coming from the input images, pooling also results in a reduction of the resolution

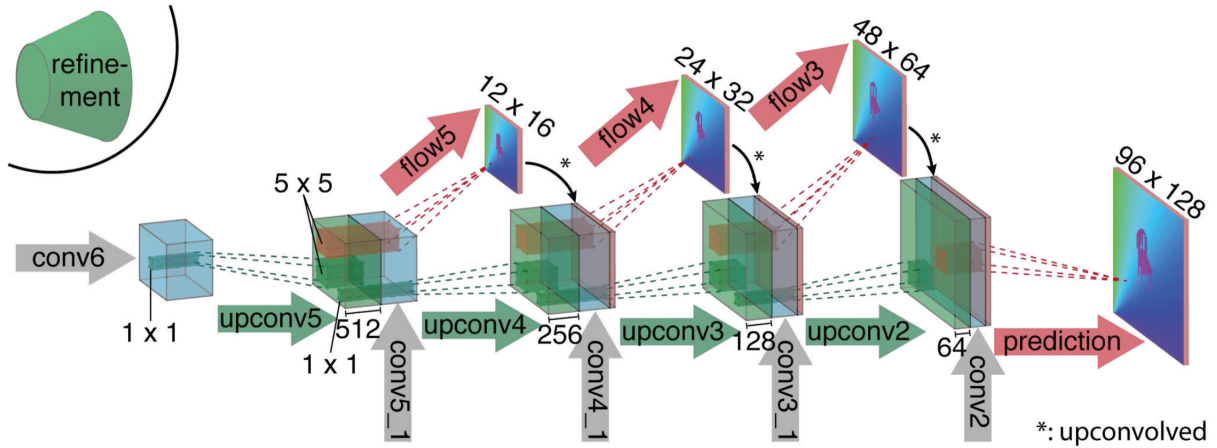


Figure A.5: Expanding part of the FlowNet architecture, refining the coarse flow to high resolution. Retrieved from³⁰.

that disallows for the generation of dense per-pixel estimations. For this reason, a second part of the network is required to refine the coarse flow (obtained by pooling) to high resolution.

The expanding part of the network is driven by "upconvolutional" layers and the refinement is indeed performed by applying the upconvolution to feature maps. Then, to preserve both the high- and low-level information, the authors concatenated the upconvoluted feature maps with corresponding feature maps from the contractive part and with an upsampled coarser flow prediction. This is repeated 4 times, as illustrated in Figure A.5.

Evolution of FlowNet 1.0. FlowNet 2.0 inherits the advantages of the original FlowNet while also resolving problems with small displacements and noisy artifacts. The architecture of FlowNet 2.0 is depicted in Figure A.6. It features:

- **A new training data schedule**

The authors experimented with the training phase to conclude that the final performance is not only determined by the quality of training data themselves but also by the order they are presented during training. FlowNetSimple and FlowNetCorr networks trained with the new schedule are now denoted with FlowNet2-S and FlowNet2-C respectively;

- **Stacked architecture**

The paper remarks that all state-of-the-art OF techniques rely on iterative methods and successfully introduce iterative refinement in FlowNet by stacking multiple network architectures. In the final configuration, the stack comprises (in order) 1 FlowNetC and 2 FlowNetS (i.e. FlowNet2-CSS);

- **A new network specialized on small displacements**

A FlowNet2-CSS architecture was subject to further training and fine-tuning to better handle small displacements and

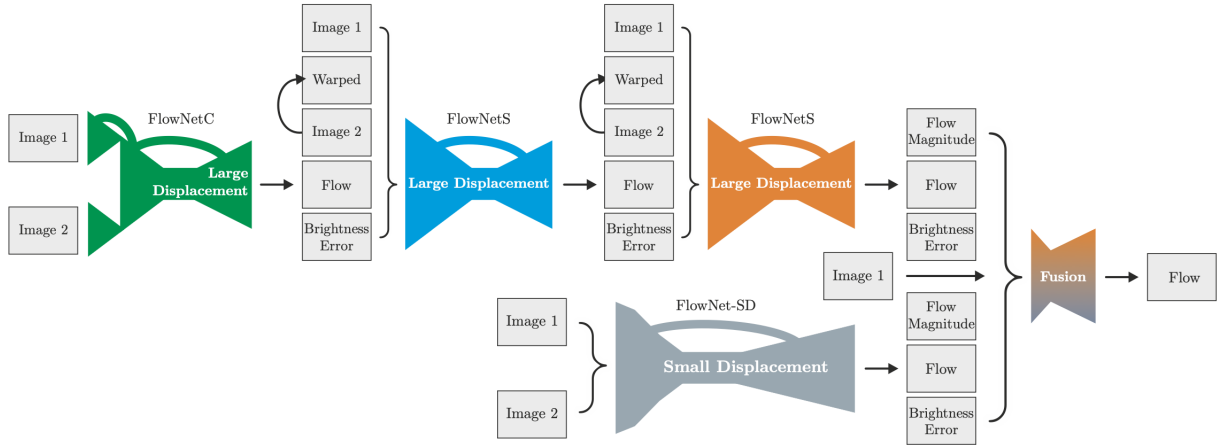


Figure A.6: Schematic view of the complete FlowNet 2.0 architecture, as seen in³¹. It comprises (i) a stack of networks (1 FlowNet2-C and 2 FlowNet2-S, resulting in FlowNet2-CSS); (ii) a sub-network specialized in small displacements, named FlowNet2-SD; (iii) a small deep network that fuses FlowNet2-CSS and FlowNet2-SD.

subpixel motion. The resulting sub-network was denoted as FlowNet2-SD.

► **Fusion network**

Finally, the authors introduced a small and simple deep network to fuse FlowNet2-SCC and FlowNet2-SD. The resulting network is FlowNet2.

The final architecture reduced the estimation error of the original FlowNet by more than 50% and is capable of handling large as well as small displacements.

A.1.3 Farneback

The Gunnar Farneback (or simply Farneback) technique was introduced in 2003 by the homonym researcher in his paper *"Two-Frame Motion Estimation Based on Polynomial Expansion"*⁵². It is a two-frame motion estimation algorithm that relies on the approximation of both frames' neighborhoods by quadratic polynomials, which can be done using polynomial expansion. Then, using the polynomial expansion coefficients, displacement fields can be estimated by observing how an exact polynomial transforms under translation.

Polynomial expansion. Instead of working directly on the image intensity values, most image analysis methods start with some transformation of the data (e.g. Fourier transforms). Gunnar Farneback introduced the concept of polynomial expansion⁵²: the basic idea is to approximate a neighborhood of each pixel with a polynomial and to assume that its coefficients capture sufficient information about the signal. The Farneback method is based on *quadratic* polynomials, which give the local signal model expressed

⁵² Farneback (2002)

in a local coordinate system:

$$f(\mathbf{x}) \sim \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c \quad (\text{A.6})$$

where $\mathbf{A}$ is a symmetric matrix, $\mathbf{b}$ a vector and c a scalar. More specifically:

$$c = r_1, \quad \mathbf{b} = \begin{pmatrix} r_2 \\ r_3 \end{pmatrix}, \quad \text{and } \mathbf{A} = \begin{pmatrix} r_4 & \frac{r_6}{2} \\ \frac{r_6}{2} & r_5 \end{pmatrix} \quad (\text{A.7})$$

so that

$$\begin{aligned} f(x, y) &\sim p(x, y) \\ &= r_1 + r_2 x + r_3 y + r_4 x^2 + r_5 x^2 + r_6 xy \end{aligned} \quad (\text{A.8})$$

The coefficients r_i are estimated from a weighted least squares fit to the signal values in the neighborhood⁵³:

⁵³ Antonielli et al. (2011)

$$\arg \min_{r_1, \dots, r_6} \sum_{x, y} \left(w(x, y) (f(x, y) - p(x, y)) \right)^2, \quad (\text{A.9})$$

where:

$$w(x, y) = \begin{cases} e^{-\frac{x^2+y^2}{2\sigma^2}}, & \text{if } |x| \leq \frac{N-1}{2}, |y| \leq \frac{N-1}{2} \\ 0, & \text{otherwise} \end{cases} \quad (\text{A.10})$$

Ideal displacement estimation. We wish to estimate the displacement between the first and the second frame. In his paper, Farnebäck introduces the reader to displacement estimation by analyzing the ideal case in which an image is described by an exact polynomial:

$$f_1(\mathbf{x}) = \mathbf{x}^T \mathbf{A}_1 \mathbf{x} + \mathbf{b}_1^T \mathbf{x} + c_1 \quad (\text{A.11})$$

and the second translated image is constructed by introducing a global displacement $\mathbf{d}$:

$$\begin{aligned} f_2(\mathbf{x}) &= f_1(\mathbf{x} - \mathbf{d}) = (\mathbf{x} - \mathbf{d})^T \mathbf{A}_1 (\mathbf{x} - \mathbf{d}) + \mathbf{b}_1^T (\mathbf{x} - \mathbf{d}) + c_1 \\ &= \mathbf{x}^T \mathbf{A}_1 \mathbf{x} + (\mathbf{b}_1 - 2\mathbf{A}_1 \mathbf{d})^T \mathbf{x} + \mathbf{d}^T \mathbf{A}_1 \mathbf{d} - \mathbf{b}_1^T \mathbf{d} + c_1 \\ &= \mathbf{x}^T \mathbf{A}_2 \mathbf{x} + \mathbf{b}_2^T \mathbf{x} + c_2 \end{aligned} \quad (\text{A.12})$$

Then, equating the coefficients in the quadratic polynomial yields:

$$\mathbf{A}_2 = \mathbf{A}_1 \quad (\text{A.13})$$

$$\mathbf{b}_2 = \mathbf{b}_1 - 2\mathbf{A}_1 \mathbf{d} \quad (\text{A.14})$$

$$c_2 = \mathbf{d}^T \mathbf{A}_1 \mathbf{d} - \mathbf{b}_1^T \mathbf{d} + c_1 \quad (\text{A.15})$$

Finally, if $\mathbf{A}_1$ is non-singular the global displacement $\mathbf{d}$ can be solved from Equation A.14:

$$\begin{aligned} 2\mathbf{A}_1\mathbf{d} &= -(\mathbf{b}_2 - \mathbf{b}_1) \\ \mathbf{d} &= -\frac{1}{2}\mathbf{A}_1^{-1}(\mathbf{b}_2 - \mathbf{b}_1) \end{aligned} \quad (\text{A.16})$$

Practical considerations. This derivation was based on the (unrealistic) assumptions that:

- An entire image is exactly described by a single polynomial;
- Two consecutive frames are related by a global translation.

To account for these issues, Farnebäck began with replacing the *global* polynomial with *local* polynomial approximations by doing a polynomial expansion of both images. This gives the expansion coefficients $\mathbf{A}_1(x)$, $\mathbf{b}_1(x)$ and $c_1(x)$ for the first frame and $\mathbf{A}_2(x)$, $\mathbf{b}_2(x)$ and $c_2(x)$ for the second frame. Ideally we wish to have $\mathbf{A}_1 = \mathbf{A}_2$ according to Equation A.14, but a practical solution is to settle for the arithmetic mean:

$$\mathbf{A}(x) = \frac{\mathbf{A}_1(x) + \mathbf{A}_2(x)}{2} \quad (\text{A.17})$$

Next, we wish to move from *global* to *local* translations. To simplify notation, the author introduced the variable $\Delta\mathbf{b}(\mathbf{x})$:

$$\Delta\mathbf{b}(\mathbf{x}) = -\frac{1}{2}(\mathbf{b}_2(\mathbf{x}) - \mathbf{b}_1(\mathbf{x})) \quad (\text{A.18})$$

so that plugging A.17 and A.18 in Equation A.14 yields:

$$\mathbf{A}(\mathbf{x})\mathbf{d}(\mathbf{x}) = \Delta\mathbf{b}(\mathbf{x}) \quad (\text{A.19})$$

which is the final system that we wish to solve. Here, $\mathbf{d}(\mathbf{x})$ indicates that we have also replaced the global displacement with a spatially varying displacement field.

Estimation over a neighborhood. Equation A.19 can be solved pointwise, but leads to noisy results. Instead, Farnebäck assumes that the displacement field is only slowly varying, allowing to integrate information over a neighborhood of each pixel. Thus we try to find $\mathbf{d}(\mathbf{x})$ from Eq. A.19 which also minimizes:

$$\sum_{\Delta\mathbf{x} \in I} w(\Delta\mathbf{x}) \|\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})\mathbf{d}(\mathbf{x}) - \Delta\mathbf{b}(\mathbf{x} + \Delta\mathbf{x})\|^2 \quad (\text{A.20})$$

Where $w(\Delta \mathbf{x})$ is a weight function for the points in the neighborhood. The minimum is obtained for:

$$\mathbf{d}(\mathbf{x}) = \left(\sum w \mathbf{A}^T \mathbf{A} \right)^{-1} \sum w \mathbf{A}^T \Delta \mathbf{b} \quad (\text{A.21})$$

and the minimum value is given by:

$$e(\mathbf{x}) = \left(\sum w \Delta \mathbf{b}^T \Delta \mathbf{b} \right) - \mathbf{d}(\mathbf{x})^T \sum w \mathbf{A}^T \Delta \mathbf{b} \quad (\text{A.22})$$

Parametrized Displacement Fields. The robustness can be improved if the displacement field can be parametrized according to some motion model. In this 2D case, the author uses:

$$\begin{aligned} d_x(x, y) &= a_1 + a_2x + a_3y + a_7x^2 + a_8xy \\ d_y(x, y) &= a_4 + a_5x + a_6y + a_7xy + a_8y^2 \end{aligned} \quad (\text{A.23})$$

This can be rewritten as:

$$\mathbf{d} = \mathbf{S} \mathbf{p} \quad (\text{A.24})$$

$$\mathbf{S} = \begin{pmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{pmatrix} \quad (\text{A.25})$$

$$\mathbf{p} = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \end{pmatrix}^T \quad (\text{A.26})$$

Inserting this into leads to weighted least squares problem:

$$\sum_i w_i \|\mathbf{A}_i \mathbf{S}_i \mathbf{p} - \Delta \mathbf{b}_i\|^2 \quad (\text{A.27})$$

where i is used to index the index the coordinates in a neighborhood. The solution is

$$\mathbf{p} = \left(\sum_i w_i \mathbf{S}_i^T \mathbf{A}_i^T \mathbf{A}_i \mathbf{S}_i \right)^{-1} \sum_i w_i \mathbf{S}_i^T \mathbf{A}_i^T \Delta \mathbf{b}_i \quad (\text{A.28})$$

Iterative and multi-scale displacement estimation. In his paper, Farnebäck highlighted the possibility to incorporate a priori knowledge in the calculation of the displacement, allowing for better estimates. Essentially, we compare the polynomial at $\mathbf{x}$ of the first image to the polynomial at $\mathbf{x} + \tilde{\mathbf{d}}(x)$ in the second image, where $\tilde{\mathbf{d}}(x)$ is the a priori displacement rounded to integer values. Then, we simply estimate the relative displacement between the real value and the rounded a priori estimate. More formally, we replace

Equations A.17 and A.18 – respectively – with:

$$\mathbf{A}(\mathbf{x}) = \frac{\mathbf{A}_1(\mathbf{x}) + \mathbf{A}_2(\tilde{\mathbf{x}})}{2} \quad (\text{A.29})$$

$$\Delta \mathbf{b}(\mathbf{x}) = -\frac{1}{2} (\mathbf{b}_2(\tilde{\mathbf{x}}) - \mathbf{b}_1(\mathbf{x})) + \mathbf{A}(\mathbf{x}) \tilde{\mathbf{d}}(\mathbf{x}) \quad (\text{A.30})$$

where

$$\tilde{\mathbf{x}} = \mathbf{x} + \tilde{\mathbf{d}}(\mathbf{x}) \quad (\text{A.31})$$

The a priori knowledge can be used to estimate the displacement iteratively or using a multi-scale approach. In both cases, we repeatedly estimate the displacement in one step and use it as a priori displacement in the next step, usually setting it to zero for the first iteration. In the iterative approach the same polynomial expansion coefficients are used in all iterations and need to be computed only once. The second approach uses a multi-scale estimation to better handle large displacements, propagating the image trough a coarse-to-fine scale that gradually increases the accuracy of the estimates. This approach, however, implies the re-computation of the coefficients for each scale.

A.2 Interpolation method in contraction profiles

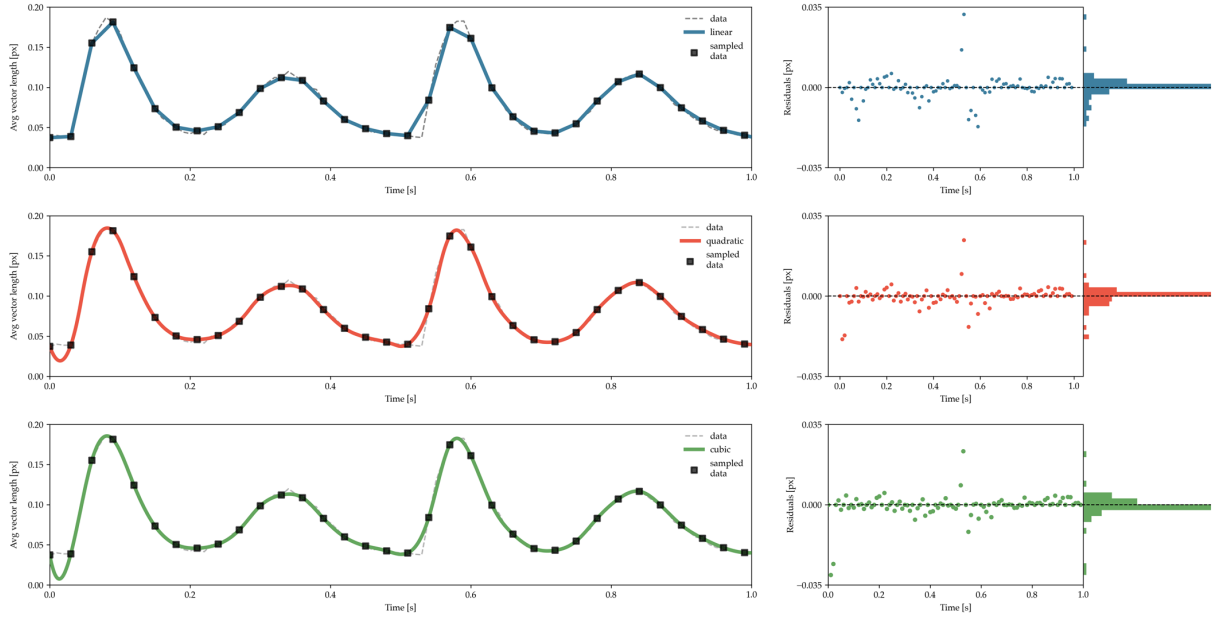
Throughout the report, we attempted to obtain cleaner T2T profiles by frame sampling the original video. Then, to restore the original 100 fps time resolution, the contraction curves were obtained by interpolation. This section shows the motivation behind choosing a specific interpolation method, i.e. quadratic.

One possible way of determining the best interpolation method would be to compare the contraction profile obtained from a sampled video with that of the non-sampled version. More specifically, one could try to interpolate the profile of the sampled recording and see which interpolation method (linear, quadratic, cubic) yields a curve that resembles the non-sampled video's profile the closest. The problem with this strategy is that the two curves are not directly comparable, since the sampled video results in a profile that has much higher peaks (the higher the sampling frequency, the greater the displacement, hence the higher the peaks). As a workaround, before the analysis one could do a normalization of these curves (to make them comparable shape-wise), but due to the information loss introduced by frame sampling the maximum (and/or minimum) value of the original curve will likely not correspond to those extracted from the sampled video.

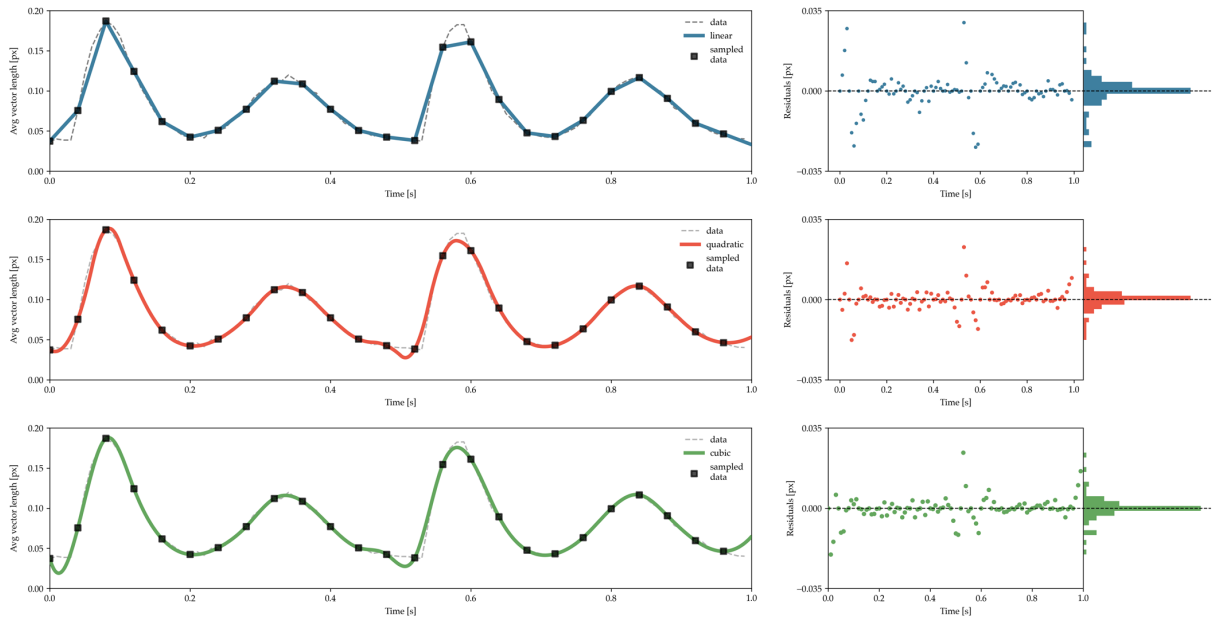
Still, we don't want to blindly assume that the displacement is linear/quadratic/cubic. What can be done to reason a choice is to perform the experiment on a synthetic sampled profile:

1. Take the contraction profile derived from a video without any frame sampling;
2. From that curve, sample points at a specific frequency. The result will be a synthetic sampled profile that can now be directly compared to the non-sampled curve;
3. Interpolate the sampled points using different interpolation methods (see [Figure A.7](#));
4. Conclude which method reconstructed the original curve the closest.

Linear interpolation is the one that (obviously) performs the worst. Both quadratic and cubic interpolation, instead, yield more accurate curves as well as similar distributions of the residuals. Their performance difference will likely become more prominent for higher sampling frequencies, but here the difference is minimal. Therefore, to keep the model simple, quadratic interpolation is chosen as the preferred interpolation method.



(a) Interpolation study with sampling frequency = 3



(b) Interpolation study with sampling frequency = 4

Figure A.7: Effects of different interpolation methods on a 1 seconds-range of a contraction profile. Linear (blue), quadratic (red) and cubic (green) interpolation are tested with a sampling frequency of 3 (Fig. a) and 4 (Fig. b).

A.3 Supplementary material

Several videos have been attached to the report to facilitate the understanding of the text. This supplementary material is attached below.

Supplementary Material 1	Raw video of a T1T	link
Supplementary Material 2	Raw video of a T2T	link
Supplementary Material 3	Results of the motion analysis of a T1T	link
Supplementary Material 4	Results of the motion analysis of a T2T	link
Supplementary Material 5	HOOFs of a T2T	link

Table A.1: Supplementary material.

Bibliography

Here are the references in citation order.

- [1] Brian V. Reamy, Pamela M. Williams, and Daniel P. Kuckel. 'Prevention of Cardiovascular Disease'. In: *Primary Care: Clinics in Office Practice* 45.1 (2018). Cardiovascular Disease, pp. 25–44. doi: <https://doi.org/10.1016/j.pop.2017.11.003> (cited on page 1).
- [2] Michael Hay et al. 'Clinical development success rates for investigational drugs'. In: *Nature biotechnology* 32.1 (Jan. 2014), pp. 40–51. doi: [10.1038/nbt.2786](https://doi.org/10.1038/nbt.2786) (cited on page 1).
- [3] Christian Zuppinge. '3D Cardiac Cell Culture: A Critical Review of Current Technologies and Applications'. In: *Frontiers in Cardiovascular Medicine* 6 (2019), p. 87. doi: [10.3389/fcvm.2019.00087](https://doi.org/10.3389/fcvm.2019.00087) (cited on page 1).
- [4] Sebastian Schaaf et al. 'Human Engineered Heart Tissue as a Versatile Tool in Basic Research and Preclinical Toxicology'. In: *PLOS ONE* 6.10 (Oct. 2011), pp. 1–11. doi: [10.1371/journal.pone.0026397](https://doi.org/10.1371/journal.pone.0026397) (cited on page 1).
- [5] Amanda Chase. 'Learning from 3D Engineered Heart Tissue'. 2020. URL: https://med.stanford.edu/cvi/mission/news_center/articles_announcements/learning-from-3d-engineered-heart-tissue.html (visited on 05/24/2021) (cited on page 1).
- [6] Kazutoshi Takahashi and Shinya Yamanaka. 'Induction of Pluripotent Stem Cells from Mouse Embryonic and Adult Fibroblast Cultures by Defined Factors'. In: *Cell* 126.4 (2006), pp. 663–676. doi: <https://doi.org/10.1016/j.cell.2006.07.024> (cited on page 1).
- [7] Shin Kadota et al. 'Development of a reentrant arrhythmia model in human pluripotent stem cell-derived cardiac cell sheets'. In: *European Heart Journal* 34.15 (Nov. 2012), pp. 1147–1156. doi: [10.1093/eurheartj/ehs418](https://doi.org/10.1093/eurheartj/ehs418) (cited on page 2).
- [8] Milica Dostanić et al. 'A Miniaturized EHT Platform for Accurate Measurements of Tissue Contractile Properties'. In: *Journal of Microelectromechanical Systems* 29.5 (2020), pp. 881–887. doi: [10.1109/JMEMS.2020.3011196](https://doi.org/10.1109/JMEMS.2020.3011196) (cited on page 2).
- [9] Ingra Mannhardt et al. 'Piezo-bending actuators for isometric or auxotonic contraction analysis of engineered heart tissue'. In: *Journal of Tissue Engineering and Regenerative Medicine* 13.1 (2019), pp. 3–11. doi: <https://doi.org/10.1002/term.2755> (cited on page 2).
- [10] Grant Mathews et al. 'Computational Analysis of Contractility in Engineered Heart Tissue'. In: *IEEE Transactions on Biomedical Engineering* 59.5 (2012), pp. 1429–1435. doi: [10.1109/TBME.2012.2187899](https://doi.org/10.1109/TBME.2012.2187899) (cited on page 2).
- [11] Claus Svane Sondergaard et al. 'Contractile and Electrophysiologic Characterization of Optimized Self-Organizing Engineered Heart Tissue'. In: *The Annals of Thoracic Surgery* 94.4 (2012), pp. 1241–1249. doi: <https://doi.org/10.1016/j.athoracsur.2012.04.098> (cited on page 2).
- [12] Arne Hansen et al. 'Development of a Drug Screening Platform Based on Engineered Heart Tissue'. In: *Circulation Research* 107.1 (2010), pp. 35–44. doi: [10.1161/CIRCRESAHA.109.211458](https://doi.org/10.1161/CIRCRESAHA.109.211458) (cited on page 3).

- [13] Côme Pasqualin et al. 'SarcOptiM for ImageJ: high-frequency online sarcomere length computing on stimulated cardiomyocytes'. In: *American Journal of Physiology-Cell Physiology* 311.2 (2016). PMID: 27335170, pp. C277–C283. doi: [10.1152/ajpcell.00094.2016](https://doi.org/10.1152/ajpcell.00094.2016) (cited on page 3).
- [14] Tina C. Stummann et al. 'Digital Movie Analysis for Quantification of Beating Frequencies, Chronotropic Effects, and Beating Areas in Cardiomyocyte Cultures'. In: *ASSAY and Drug Development Technologies* 6.3 (2008). PMID: 18532900, pp. 375–385. doi: [10.1089/adt.2008.129](https://doi.org/10.1089/adt.2008.129) (cited on page 3).
- [15] Takeshi Kunihiro et al. 'Noninvasive Evaluation of Contractile Behavior of Cardiomyocyte Monolayers Based on Motion Vector Analysis'. In: *Tissue Engineering Part C: Methods* 18.1 (2012). PMID: 21851323, pp. 21–32. doi: [10.1089/ten.tec.2011.0273](https://doi.org/10.1089/ten.tec.2011.0273) (cited on pages 3, 16, 18, 25, 26).
- [16] Aaron Chen et al. 'Integrated platform for functional monitoring of biomimetic heart sheets derived from human pluripotent stem cells'. In: *Biomaterials* 35.2 (2014), pp. 675–683. doi: <https://doi.org/10.1016/j.biomaterials.2013.10.007> (cited on page 3).
- [17] Tomohiro Hayakawa et al. 'Image-based evaluation of contraction–relaxation kinetics of human-induced pluripotent stem cell-derived cardiomyocytes: Correlation and complementarity with extracellular electrophysiology'. In: *Journal of Molecular and Cellular Cardiology* 77 (2014), pp. 178–191. doi: <https://doi.org/10.1016/j.yjmcc.2014.09.010> (cited on pages 3, 16, 18, 25, 26).
- [18] Ivan Kopljär et al. 'Chronic drug-induced effects on contractile motion properties and cardiac biomarkers in human induced pluripotent stem cell-derived cardiomyocytes'. In: *British Journal of Pharmacology* 174.21 (2017), pp. 3766–3779. doi: <https://doi.org/10.1111/bph.13713> (cited on pages 3, 16, 18, 25).
- [19] Luca Sala et al. 'MUSCLEMOTION'. In: *Circulation Research* 122.3 (2018), e5–e16. doi: [10.1161/CIRCRESAHA.117.312067](https://doi.org/10.1161/CIRCRESAHA.117.312067) (cited on page 3).
- [20] Meiyu Liang et al. 'Spatiotemporal Super-Resolution Reconstruction Based on Robust Optical Flow and Zernike Moment for Video Sequences'. In: vol. 2013. May 2013, pp. 1–6. doi: [10.1109/ISIE.2013.6563661](https://doi.org/10.1109/ISIE.2013.6563661) (cited on page 3).
- [21] Giannis Chantas, Theodosios Gkamas, and Christophoros Nikou. 'Variational-Bayes Optical Flow'. In: *Journal of Mathematical Imaging and Vision* 50 (Nov. 2014). doi: [10.1007/s10851-014-0494-3](https://doi.org/10.1007/s10851-014-0494-3) (cited on page 3).
- [22] S. S. Beauchemin and J. L. Barron. 'The Computation of Optical Flow'. In: *ACM Comput. Surv.* 27.3 (Sept. 1995), pp. 433–466. doi: [10.1145/212094.212141](https://doi.org/10.1145/212094.212141) (cited on page 3).
- [23] Berthold K.P. Horn and Brian G. Schunck. 'Determining optical flow'. In: *Artificial Intelligence* 17.1 (1981), pp. 185–203. doi: [https://doi.org/10.1016/0004-3702\(81\)90024-2](https://doi.org/10.1016/0004-3702(81)90024-2) (cited on page 3).
- [24] Bruce D. Lucas and Takeo Kanade. 'An Iterative Image Registration Technique with an Application to Stereo Vision'. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'81. Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679 (cited on page 3).
- [25] Gunnar Farneback. 'Two-Frame Motion Estimation Based on Polynomial Expansion'. In: *Proceedings of the 13th Scandinavian Conference on Image Analysis*. SCIA'03. Halmstad, Sweden: Springer-Verlag, 2003, pp. 363–370 (cited on pages 3, 51).

- [26] Shivangi Anthwal and Dinesh Ganotra. 'An overview of optical flow-based approaches for motion segmentation'. In: *The Imaging Science Journal* 67.5 (2019), pp. 284–294. doi: [10.1080/13682199.2019.1641316](https://doi.org/10.1080/13682199.2019.1641316) (cited on pages 3, 4).
- [27] Deqing Sun et al. 'Learning Optical Flow'. In: *Computer Vision – ECCV 2008*. Ed. by David Forsyth, Philip Torr, and Andrew Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 83–97 (cited on page 4).
- [28] Dan Rosenbaum, Daniel Zoran, and Yair Weiss. 'Learning the Local Statistics of Optical Flow'. In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges et al. Vol. 26. Curran Associates, Inc., 2013 (cited on page 4).
- [29] S. Roth and M. J. Black. 'On the spatial statistics of optical flow'. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 1. Oct. 2005, 42–49 Vol. 1. doi: [10.1109/ICCV.2005.180](https://doi.org/10.1109/ICCV.2005.180) (cited on page 4).
- [30] A. Dosovitskiy et al. 'FlowNet: Learning Optical Flow with Convolutional Networks'. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 2758–2766. doi: [10.1109/ICCV.2015.316](https://doi.org/10.1109/ICCV.2015.316) (cited on pages 4, 48–50).
- [31] Eddy Ilg et al. 'FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks'. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1647–1655. doi: [10.1109/CVPR.2017.179](https://doi.org/10.1109/CVPR.2017.179) (cited on pages 4, 48, 51).
- [32] A. Ahmadi and I. Patras. 'Unsupervised convolutional neural networks for motion estimation'. In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 1629–1633. doi: [10.1109/ICIP.2016.7532634](https://doi.org/10.1109/ICIP.2016.7532634) (cited on page 4).
- [33] Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis. 'Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness'. In: *Computer Vision – ECCV 2016 Workshops*. Ed. by Gang Hua and Hervé Jégou. Cham: Springer International Publishing, 2016, pp. 3–10 (cited on page 4).
- [34] Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. 'Semi-Supervised Learning for Optical Flow with Generative Adversarial Networks'. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017 (cited on page 4).
- [35] Yanchao Yang and Stefano Soatto. 'Conditional Prior Networks for Optical Flow'. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 282–298 (cited on page 4).
- [36] Vahid Tavakoli et al. 'An Evaluation of Different Optical Flow Techniques for Myocardial Motion Analysis in B-Mode Echocardiography Images'. In: vol. 21. Jan. 2008. doi: [10.1007/978-3-540-69139-6_127](https://doi.org/10.1007/978-3-540-69139-6_127) (cited on page 4).
- [37] Jim Rybarski et al. 'nd2reader: GitHub Repository'. 2021. URL: <https://github.com/rbnvrw/nd2reader> (visited on 05/24/2021) (cited on page 9).
- [38] Munawwar Firoz. 'Background Removal (Segmentation) with OpenCV'. 2019. URL: <https://www.codepast.com/computer-vision/2019/04/26/background-segmentation-removal-with-opencv-take-2.html> (visited on 05/17/2021) (cited on page 12).
- [39] Piotr Dollár and C. Lawrence Zitnick. 'Structured Forests for Fast Edge Detection'. In: *ICCV '13*. USA: IEEE Computer Society, 2013, pp. 1841–1848. doi: [10.1109/ICCV.2013.231](https://doi.org/10.1109/ICCV.2013.231) (cited on page 12).

- [40] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 'GrabCut -Interactive Foreground Extraction using Iterated Graph Cuts'. In: *ACM Transactions on Graphics (SIGGRAPH)* (Aug. 2004) (cited on page 12).
- [41] Achmad Solichin, Agus Harjoko, and Agfianto Eko Putra. 'Movement Direction Estimation on Video using Optical Flow Analysis on Multiple Frames'. In: *International Journal of Advanced Computer Science and Applications* 9.6 (2018). DOI: [10.14569/IJACSA.2018.090625](https://doi.org/10.14569/IJACSA.2018.090625) (cited on page 16).
- [42] Yali Huang et al. 'Quantitative Analysis of Intracellular Motility Based on Optical Flow Model'. In: *Journal of healthcare engineering* 2017 (2017). DOI: <http://doi.org/10.1155/2017/1848314> (cited on page 16).
- [43] Rizwan Chaudhry et al. 'Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions'. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 1932–1939. DOI: [10.1109/CVPR.2009.5206821](https://doi.org/10.1109/CVPR.2009.5206821) (cited on page 17).
- [44] Yoan Tournade. 'Overview of the peaks detection algorithms available in Python'. Sept. 2019. URL: <https://pythonawesome.com/overview-of-the-peaks-detection-algorithms-available-in-python/> (visited on 06/18/2021) (cited on page 22).
- [45] Gautam Banuru. 3 Step Search H.264 Compression Block Matching Algorithm. 2020. URL: <https://github.com/gbanuru18/BlockMatching> (visited on 06/22/2021) (cited on page 25).
- [46] Nathaniel Huebsch et al. 'Automated Video-Based Analysis of Contractility and Calcium Flux in Human-Induced Pluripotent Stem Cell-Derived Cardiomyocytes Cultured over Different Spatial Scales'. In: *Tissue Engineering Part C: Methods* 21.5 (2015). PMID: 25333967, pp. 467–479. DOI: [10.1089/ten.tec.2014.0283](https://doi.org/10.1089/ten.tec.2014.0283) (cited on pages 26, 43).
- [47] Fitsum Reda et al. *GitHub repository*. 'Pytorch implementation of FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks'. 2017. URL: <https://github.com/NVIDIA/flownet2-pytorch> (cited on page 26).
- [48] Aroh Barjatya. 'Block Matching Algorithms for Motion Estimation'. *MATLAB Central File Exchange*. 2011. URL: <https://www.mathworks.com/matlabcentral/fileexchange/8761-block-matching-algorithms-for-motion-estimation> (visited on 06/27/2021) (cited on page 43).
- [49] Clara F. Asnes et al. 'Reconstitution of the Frank-Starling Mechanism in Engineered Heart Tissues'. In: *Biophysical Journal* 91.5 (2006), pp. 1800–1810. DOI: <https://doi.org/10.1529/biophysj.105.065961> (cited on page 44).
- [50] Willem J. de Lange et al. 'Human iPSC-engineered cardiac tissue platform faithfully models important cardiac physiology'. In: *American Journal of Physiology-Heart and Circulatory Physiology* 320.4 (2021). PMID: 33606581, H1670–H1686. DOI: [10.1152/ajpheart.00941.2020](https://doi.org/10.1152/ajpheart.00941.2020) (cited on page 44).
- [51] Sonam Khawase et al. 'An Overview of Block Matching Algorithms for Motion Vector Estimation'. In: June 2017, pp. 217–222. DOI: [10.15439/2017R85](https://doi.org/10.15439/2017R85) (cited on page 47).
- [52] Gunnar Farneback. 'Polynomial expansion for orientation and motion estimation'. eng. PhD thesis. 2002 (cited on page 51).

- [53] Alessio Antionielli, Giannantonio D'Avico, and Marco Paciscopi. 'Flusso ottico. Concetti generali e metodi di calcolo'. University Lecture. Università degli Studi di Firenze, Laurea Magistrale in Ingegneria Informatica. Corso di Analisi di Immagini e Video.
URL: http://www.allafinedelpalo.it/uploads/opticalFlow_presentazione.pdf. 2011 (cited on page 52).