UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Non-Rigid Registration Methods for Skirting Line Detection in Wool

by Daniel Alejandro Perez Banuelos

Thesis submitted in fulfilment of the requirements for the degree of

Master of Engineering (Research)

under the supervision of Dr. Alen Alempijevic and Dr. Raphael Falque

University of Technology Sydney
Faculty of Engineering and Information Technology

August 2024

Certificate of Original Authorship

I Daniel Alejandro Perez Banuelos declare that this thesis, is submitted in fulfilment of the requirements for the award of Master of Engineering (Research), in the Faculty of Engineering and IT, School of Mechanical and Mechatronic Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signed: Production Note:
Signature removed prior to publication.

Date: 08 of August 2024

UNIVERSITY OF TECHNOLOGY SYDNEY

Abstract

Faculty of Engineering and Information Technology
Robotics Institute

Master of Engineering (Research)

by Daniel Alejandro Perez Banuelos

Automation has emerged as a transformative force in modern society, revolutionizing industries by enhancing efficiency, reducing labour costs, and increasing production output. In agriculture, automation has been particularly impactful, with examples such as precision farming and automated harvesting systems significantly improving crop production and management. Despite these advancements, the wool industry has remained largely untouched by automation. Automation in contaminant detection on fleece can severely improve wool production and reduce on-farm costs, by reducing the reliance on manual labour and enhancing the consistency of the product. To address this automation problem, the thesis introduces an approach to contaminant detection on fleece that centres on the development of an image registration technique. This technique compares images of wool before and after the skirting process, facilitating the detection of the skirting line by identifying the contaminated areas that had been removed in the after-skirt image.

To align the after-skirt image with the before skirt image, it is essential to employ an image registration method capable of accommodating the transformations that happen during the skirting process. These transformations are detected with the use of feature matching algorithms and subsequently filtered to remove all outliers. Thus, a novel filtering process is utilised, which accounts for the deformations that happen to the wool during handling. With the use of these correspondences, image alignment is then achieved through the implementation of a non-rigid deformation method. The experiments demonstrate a lower error when aligning the wool images compared to rigid methods.

Feature matching algorithms fail to find correspondences on areas with extreme deformation located near to the skirting area. Thus, a method that integrates dense correspondences from damaged areas to the previously established correspondences is devised using a learning-based filtering method. By analysing the continuity in deformation, it becomes feasible to identify dense correspondences in the more affected areas found along the edges of the fleece.

Finally, a physics-based simulation is employed to find correspondences at the edge of the fleece which undergoes significant stretching. Given the distinctive material properties of wool, feature matching algorithms proved inadequate in identifying correspondences on areas with significant stretching along the skirting line location. This simulation method is capable of replicating the elasticity observed at the fleece's edge during the skirting process. As a result, it becomes possible to identify correspondences at the fleece's edge, thereby increasing the accuracy of skirting line location and enabling accurate image registration for elastoplastic objects that have undergone extreme deformations.

Acknowledgements

I wish to extend my heartfelt gratitude to my academic mentors, Alen Alempijevic and Raphael Falque, for their continuous support and encouragement. Collaborating with both of you has been an absolute pleasure. I would also like to express my gratitude to my former supervisor, Timothy Patten, for his initial encouragement that paved the way for my research journey.

Furthermore, I would like to convey my profound thanks and appreciation to my parents and brother for their unwavering support and encouragement throughout my academic journey.

Contents

List of Figures

List of Tables

Acronyms & Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
CNN	Convolutional Neural Network
RANSAC	Random Sample Consensus
RGB	Red, Green and Blue colour
HSV	Hue Saturation Value
FFD	Free Form Deformation
EFFD	Extended Free-Form Deformation
RKHS	Reproducing Kernel Hilbert Space
DFFD	Dirichlet Free-Form Deformation
CPD	Coherent point drift
SDF	Sign Distance Field
PCA	Principal Component Analysis
ICP	Iterative Closest Point
SIFT	Scaled Invariant Feature Transform
SURF	Speeded up Robust Feature
FAST	Features from accelerated segment test
ORB	Oriented Fast and Rotated Brief
BRISK	Binary Robust Invariant Scalable Keypoint

FREAK	Fast Retina Keypoint
FEM	Finite Element Methods
XFEM	Extended Finite Element Methods
SPH	Smooth Particle Hydrodynamics
PIC	Particle In Cell
FLIP	Fluid Implicit Particle
MPM	Material Point Method
PFF	Phase Field Fracture
ARAP	As Rigid As Possible
ED	Embedded Deformation
MLP	Multi-layer Perceptron
SVD	Singular Value Decomposition
LoFTR	Local Optical TRansformer
GNN	Graph Neural Networks
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
SCV	Sparse Correlation Volume

Nomenclature

I_s	Source Image
I_t	Target Image
p	Individual pixel
$\mathcal{P}$	Pixels
F	Set of features
f	Individual feature
S	Set of sparse features
s	Individual sparse feature
D	Set of dense features
d	Individual dense feature
M	Mesh
τ	Threshold for ARAP
G	Set of graph nodes
g	Individual graph node
E	Set of graph edges
R	Rotation matrix
$\mathbb{R}^2$	2 Dimensional Space
$\mathbb{R}^3$	3 Dimensional Space
r	Column in Rotation Matrix
$\mathbf{rig}$	Rigidity

t	Translation vector
N	Neighborhood set
EX	Extrinsic Error
E_{con}	Constrain term
E_{reg}	Regularisation term
E_{rot}	Rotation Matrix Errors
E_{ppl}	Point to Plane Error
E_{pp}	Point to Point Error
E_{align}	Alignment Error
E_{rigid}	Rigidity Error
E_{ARAP}	ARAP Error
d_{max}	Maximum Distance
w	Weight
w_{ij}	Per-Edge Weights
w_i	Per-Cell Weights
x	Point
$\hat{x}$	Target Point
df	Displacement Field
n	Surface Normals
$\|\|_F$	Frobenius norm
I	Identity Matrix
$\mathcal{N}$	Connected Vertices
m	Number of Points
m_{pos}	Mass
ρ_{pos}	Density
v_{pos}	Velocity

F_{pos}	Deformation gradient
σ_{pos}	Cauchy stress tensor
T_{pos}	Temperature
$\boldsymbol{E}$	Youngs Modulus
γ	Fibre Scale
η	Viscosity
ρ	Inextensibilty
SP	Superpixels

Chapter 1

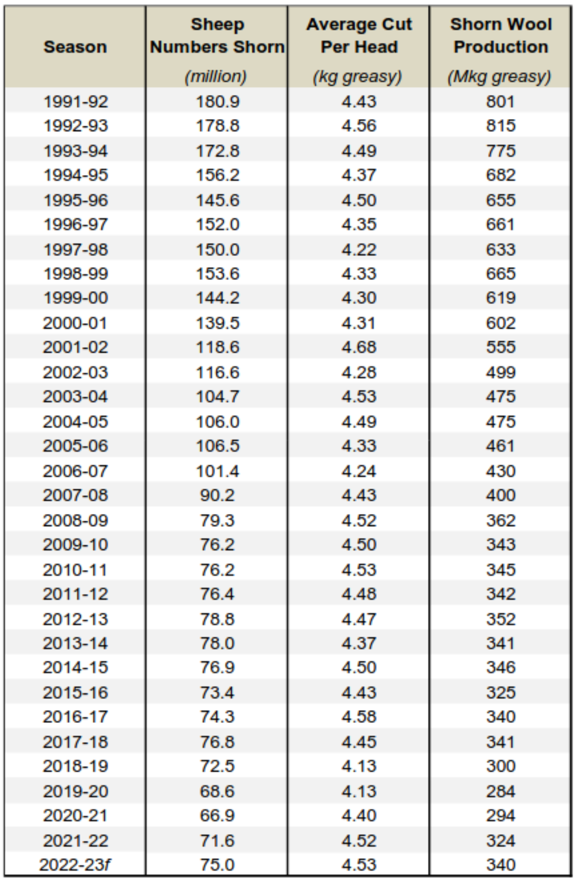
Introduction

1.1 Wool in the Australian Market

Wool production and exportation remain integral to the Australian market, with wool exports generating around $2.7 billion in 2021-2022, positioning Australia as the leading exporter and producer of wool globally [1]. However, the sector has faced challenges in recent years, including a significant reduction in wool production and exports over the last five years. As indicated by Figure 1.1, the number of sheep shorn in Australia has consistently declined over the past three decades, with a notable drop in the last four years, exacerbated by the COVID-19 pandemic. This decline can partly be attributed to major clothing companies shifting toward natural fibres like cotton, favoured for their easier transportation and production capabilities. Cotton production has witnessed significant automation advancements since the early 20th century, notably with the gin machine capable of processing 230,000 kilograms of cotton per day Cotton Australia [2].

In contrast, wool production has seen minimal advancements in automation. Shearing efficiency remains closely tied to manual skills, despite the introduction of motorized equipment. The skirting process, involving the removal of contaminants, is also predominantly a manual task, in contrast with the more automated processes in cotton production.

Despite these challenges, the Australian wool market is trying to adapt to the current needs and demands. The Australian Wool Production Forecasting Committee (AWPFC)



Season	Sheep Numbers Shorn	Average Cut Per Head	Shorn Wool Production
	(million)	*(kg greasy)*	*(Mkg greasy)*
1991-92	180.9	4.43	801
1992-93	178.8	4.56	815
1993-94	172.8	4.49	775
1994-95	156.2	4.37	682
1995-96	145.6	4.50	655
1996-97	152.0	4.35	661
1997-98	150.0	4.22	633
1998-99	153.6	4.33	665
1999-00	144.2	4.30	619
2000-01	139.5	4.31	602
2001-02	118.6	4.68	555
2002-03	116.6	4.28	499
2003-04	104.7	4.53	475
2004-05	106.0	4.49	475
2005-06	106.5	4.33	461
2006-07	101.4	4.24	430
2007-08	90.2	4.43	400
2008-09	79.3	4.52	362
2009-10	76.2	4.50	343
2010-11	76.2	4.53	345
2011-12	76.4	4.48	342
2012-13	78.8	4.47	352
2013-14	78.0	4.37	341
2014-15	76.9	4.50	346
2015-16	73.4	4.43	325
2016-17	74.3	4.58	340
2017-18	76.8	4.45	341
2018-19	72.5	4.13	300
2019-20	68.6	4.13	284
2020-21	66.9	4.40	294
2021-22	71.6	4.52	324
2022-23*f*	75.0	4.53	340

FIGURE 1.1: Number of fleeces shorn in Australia since 1991 (Australian Wool Innovation [3])

anticipates wool production volumes to continue increasing in 2023 for the third consecutive year, estimating production for the 2022/23 sale season at 340 million kg, a 4.9% increase from the previous year [4]. However, increased stock numbers and wet weather will continue to challenge shearing operations, which are already strained by a labour supply shortage.

Technological innovations in wool processing are creating new opportunities to meet current demands. Innovations in wool processing technologies have led to more efficient and environmentally friendly processes in yarn development, knit and weave manufacture, and dyeing and finishing. This progress opens up new market opportunities for wool as an eco-friendly and performance fibre. Notable recent advancements include:

- **Smart Shearing Handpiece:** Developed to improve the efficiency and ergonomics of wool shearing, this innovation promises to enhance shearer comfort and reduce physical strain, potentially addressing labour shortages in shearing operations [3].

- **Automated Wool Classing Systems:** Utilising advanced imaging and analysis technologies, these systems aim to automate the wool classing process, enhancing the accuracy and efficiency of wool sorting and grading [3].
- **Biological Wool Harvesting:** Advances in biological wool processing methods eliminating the shearing pieces increasing wool quality [3].

Despite the promising advancements in technology, the wool industry continues to confront significant workforce challenges. The physical and skill-intensive nature of wool shearing makes it a demanding profession. Shearing requires the person to be in good physical condition as they manoeuvre heavy animals and maintain precision in wool removal. This physically demanding task, coupled with the need for specialized skills, contributes to the current shortage of shearers.

The industry faces a substantial decrease in wool shearers, with around 2,000 active shearers today, a sharp decline from 30 years ago. This shortage has led to considerable delays in shearing operations, some extending up to eight weeks, particularly exacerbated by the COVID-19 pandemic. The pandemic, along with other factors like travel restrictions and a lack of confidence in interstate movement, has further strained the availability of shearers, traditionally supplemented by workers from New Zealand [5].

These delays pose significant risks to sheep welfare, including the danger of fly strike and illness due to long wool, especially during Australia's wet summers. Extended wool can also impact the sheep's breeding schedule and lamb production. Furthermore, the delayed shearing schedules lead to stock losses and increased costs for farmers, with shearing wages rising significantly [5].

Additionally, the ongoing labour shortages threaten the overall sustainability of the industry. The risk of losing a substantial portion of this $3 billion-plus industry could have far-reaching implications, affecting both downstream and upstream jobs.

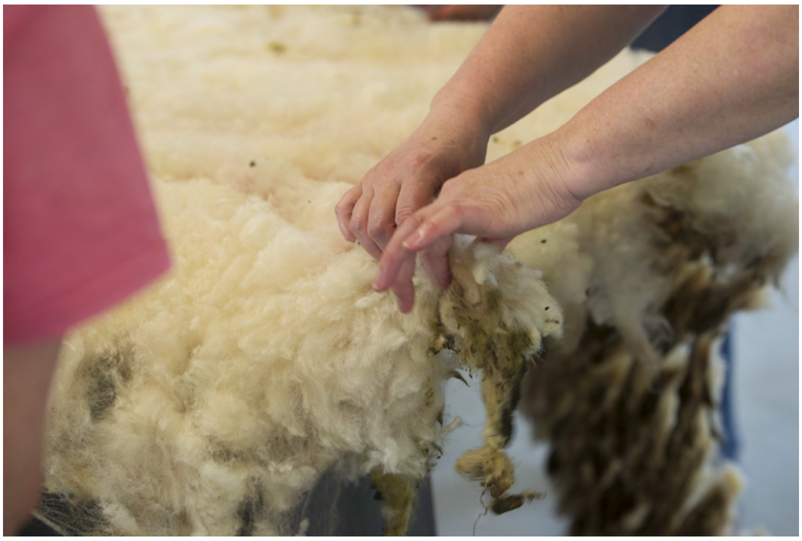


Figure 1.2: The Skirting Process involves placing freshly shorn fleece on a skirting table, where wool handlers manually remove contaminated sections, typically found around the fleece edges [6].

1.2 Automation in contaminant detection

Wool production stands to gain significantly from the automation of its processes, particularly in the area of skirting. Skirting is a critical stage where contaminants are removed from freshly shorn fleece, as depicted in Figure 1.2.

To automate this process, a high degree of confidence is required when detecting and removing the contaminants since the presence of contaminants after the skirting process has a severe effect on the whole wool pipeline. Not removing all contaminants can affect the quality of products such as cloth/fabric from the manufacturer. On the other hand, excessive skirting or over-skirting leads to significant financial losses in the wool industry.

Research into the feasibility of detecting contaminants on wool fleece using vision sensors dates back several decades [7]. Early efforts utilised Near-Infrared cameras to observe the wool, applying PCA and soft independent modelling of class analogies to differentiate between polymeric materials and uncontaminated wool. A more recent advancement by Zhang et al. [8] involved a system using RGB cameras. This system identifies polypropylene contaminants through adaptive thresholding in RGB and HSV colour spaces and is complemented by a mechanical system for real-time removal and sorting of these contaminants. However, these methods primarily target foreign contaminants such as plastics and packaging materials, leaving natural contaminants like blood, dirt, and urine, which are equally important in the skirting process, undetected.

Subsequent research has shifted towards automatically detecting the skirting line around the fleece edges [9]. This process involves using manual annotations by wool handlers, who mark the anticipated skirting line on RGB images. However, significant variability in annotations between different handlers has been noted, likely due to the subjective nature of human judgment in this process. This variability underscores the challenges in automating the skirting process and highlights the need for more objective, automated methods that can consistently identify the location of contaminants near the edge of the fleece.

For this reason, this thesis investigates the automation of the skirting process by focusing on the delineation of the skirting line. To overcome the biases inherent in human annotation, this work employs image registration techniques to align images of contaminated and clean wool.

Although the primary focus of this thesis is the detection of the skirting line in wool processing, the methodologies introduced in the following chapters can be applied to analyse the deformation between two images across various fields. These applications include medical imaging, such as tracking organ or skin deformation; geology, for monitoring glacier movement; and manufacturing, for assessing fabric deformation.

1.3 Contributions

This research aims to automate part of the pipeline during wool production. This is achieved by automatically delineating the skirting line around the edges of the fleece. The skirting line is the separation between clean and contaminated wool, indicating the parts of the fleece that should be removed during this process. In order to achieve this, this thesis makes the following contributions:

- **Non-Rigid Filter for Image Registration:** Non-rigid registration technique that incorporates a filtering process for feature correspondences to align the before and after skirt images of wool
- **Learning-Based Filtering for denser Feature Matching:** Learning-based filtering method that utilises the geometric information from less affected areas to find correspondences in the severely deformed areas in wool images
- **Material Properties Identification via Fracture Simulation for Skirting Replication:** Optimisation-based simulation that accurately replicates the stretching of the fibres along the fractured area in wool.

By integrating the methodologies presented in each chapter, this thesis introduces a unified approach that automatically detects the components of the fleece that were removed during the skirting process. This method leverages the contributions of non-rigid image registration, learning-based filtering, and material properties identification to detect the locations of contaminants and handle the significant deformation present in wool images.

1.4 Research Outputs

Academic papers published during the completion of this thesis are listed below.

D. Banuelos, R. Falque, T. Patten, and A. Alempijevic. Skirting line estimation using sparse to dense deformation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023

D. Banuelos, R. Falque, T. Patten, and A. Alempijevic. Skirting line annotation via deformation modelling. In *Australasian Conference on Robotics and Automation*, 2021

1.5 Structure of the Thesis

Following the introduction covering the wool market in Australia and its automation problem, the thesis is organised into several key chapters, each dedicated to a different approach to automatically detecting the skirting line in wool fleece based on the data and severity of the deformation.

Chapter 2 is split into two major parts, each of which gives a review of the literature related to non-rigid deformation. The first section details non-rigid deformation fields that are used to deform an image during image registration. In addition, it overviews some of the most popular feature-matching registration methods that are used as handles for the deformation fields. The second section provides an overview of physics-based non-rigid deformation, focusing on fracture simulation; covering the most common practice to simulate fracture in anisotropic materials.

Chapter 3 focuses on the development of an image registration technique to align images of fleece before and after the skirting process. This registration technique is developed using the non-rigid deformation fields outlined in the previous chapter to address the complexities in handling non-rigid deformations in fleece. It also highlights some of the difficulties of testing the algorithm's performance on field data, leading to a new control dataset being acquired.

Chapter 4 builds upon the methodology proposed in Chapter 3 and introduces a learning-based image registration algorithm. This learning-based methodology is employed to enhance the accuracy of the skirting line estimation by incorporating sparse to dense feature correspondence. The algorithm's performance is then tested on newly acquired data that allows for a denser evaluation of the non-rigid registration methods.

Chapter 5 incorporates a physics-based simulation into the methodology proposed in Chapter 4, to detect the deformation of the fibres that have been significantly stretched. By simulating the stretching of the fibre, the chapter aims to find more correspondences in areas where feature-matching algorithms failed in previous chapters. This final chapter incorporates the methodologies proposed in the previous chapters to develop an approach capable of handling significant deformation in wool images.

The thesis concludes with a summary of the findings and contributions, followed by a discussion of potential avenues for future research on how all the methodologies can be further applied.

Chapter 2

Related Work

Prior to introducing how the thesis aims to automate the detection of the skirting line, this chapter provides background information on various non-rigid deformation methods found in current literature. These methods alongside feature-matching algorithms, are commonly used for the alignment of two or more images. For this reason, the chapter further provides a review of some of the most popular and recognised feature-matching algorithms. Additionally, the background chapter provides some insight into some of the cost functions used for rigid deformation, these functions provide an alignment quality and decide the final location of the control points obtained from the feature matching algorithms.

Furthermore, the review of related work will extend into the realm of physics-based approaches to non-rigid deformation. These approaches can model the material behaviour of fleece more realistically, considering factors such as elasticity and fracture dynamics during the skirting process. Giving a better representation of the deformation of each of the components during the skirting process.

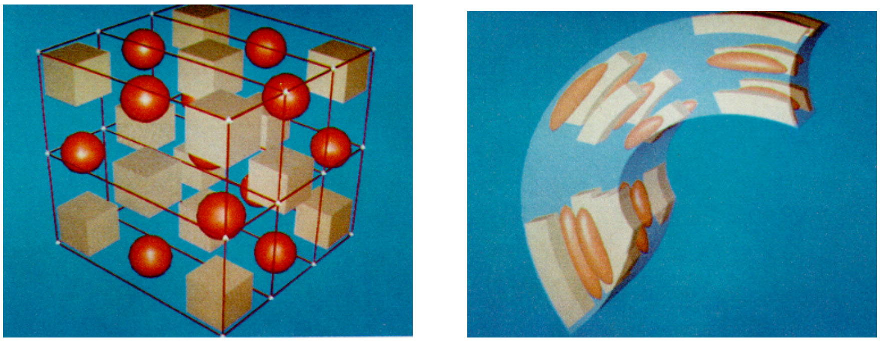
2.1 Non-Rigid Registration

Image registration plays a crucial role in effectively automating the skirting line detection. This field allows the user to align two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors. Within this domain, non-rigid registration is a critical sub-module, particularly when adjustments in the volume and shape of a solid or figure in these images are necessary. Non-rigid registration allows for local changes in wool shape without resorting to global deformations, making it ideal for aligning the after-skirt image with the before-skirt image.

To achieve the local changes in shape, non-rigid registration implements non-rigid deformation methods and feature-matching algorithms. The non-rigid deformation algorithms work by deforming the figure using the feature correspondences as the control points to determine how much the figure has to be deformed. Multiple non-rigid deformation methods have been proposed over the years trying to increment the accuracy and degrees of freedom for the user to align two objects or images.

For this reason, the related work commences by analysing the evolution of non-rigid deformation models and how they progressed over time. Further, it will look at different representations of the deformation fields and how they deform a solid or figure. Different deformation fields usually trade between accuracy, amount of detailing, and computational costs. Higher degrees of freedom increase the number of variables, reducing the computational efficiency. However, models with higher degrees of freedom can offer more precise alignment between source and target images, which is critical in detecting the skirting line in fleece accurately.

Furthermore, this section provides some insight into some cost functions used to determine the alignment quality for the deformation process. Within each different deformation field, multiple different cost functions have been proposed, these functions are used to provide an alignment quality and decide the final location of the control points obtained from the feature correspondences. These functions ensure that the resulting deformation maintains a reasonable rotation or translation, contributing to a realistic appearance of the deformed image.



(A) Original Lattice

(B) Deformed Lattice

FIGURE 2.1: Lattice structure used for early non-rigid deformation. The control nodes are coloured in white, with red bars indicating the neighbouring nodes [12].

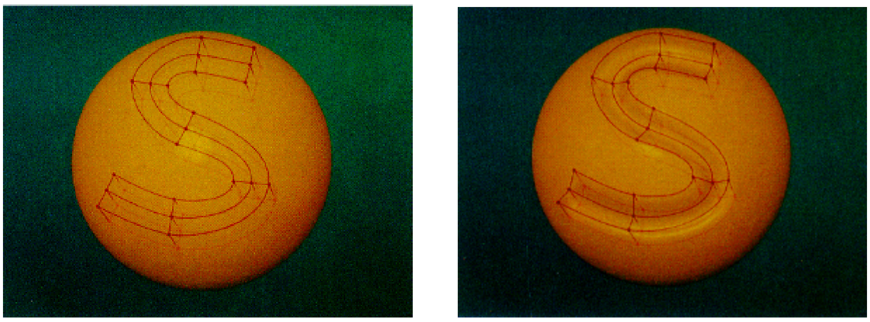
2.1.1 Non-Rigid Deformation Field

Before reviewing the different non-rigid deformation methods, it is important to provide some background into how the deformation fields are defined and their different components. Early works in non-rigid deformation use lattice structures to deform a solid. This lattice structure is portrayed as a polyhedron surrounding the object, with multiple control points or nodes and bars indicating the neighbour control nodes. When a control point is moved, the portion of the object in its vicinity is correspondingly deformed. Figure 2.1 shows an example of a lattice structure.

Recent advancements have seen a shift towards employing polymeshes for non-rigid deformation, diverging from the traditional lattice. In these approaches, the shape is deformed by directly modifying its faces. Each face in a polymesh is a flat surface, typically a polygon, that forms part of the object's surface. These faces are defined by their vertices, which are the corner points where the edges of a face meet. By altering the positions of these vertices, the shape and orientation of the faces change, leading to the deformation of the shape.

2.1.2 Early Non-Rigid Deformation Methods

Shape manipulation has been demonstrated to be an easy and intuitive way for the users to deform a Three-Dimensional (3D) object, making it an indispensable tool for object deformation. This section delves into the historical development of deformation fields over



(A) S figure from two lattices (B) S shape deformed into a ball

Figure 2.2: Extended Free-Form Deformation by Coquillart [13]

recent decades, offering insights into the evolution of non-rigid deformation techniques. By tracing the evolution from initial approaches to non-rigid deformation to contemporary methods, this section seeks to enhance the understanding of how modern deformation methods work.

A seminal work in non-rigid deformation by Sederberg and Parry [12] introduced a Free Form Deformation (FFD) that allowed objects to be deformed by warping the global space of the object. FFD operates through the utilisation of control points arranged in a lattice structure. Sederberg and Parry [12] proposed method also allowed for control over the volumetric change of the object. However, this method only allowed for global deformations, making it too limited for local single-component deformations. These deformations rarely resemble to the original object when applied to any complex solid.

Coquillart [13] further improved the work on FFD by presenting an Extended Free-Form Deformation (EFFD). Coquillart [13] method introduced the use of non-parallelepipedal lattices. This allowed the user to define a circular figure on the solid with the use of a cylindrical lattice. Further improvements include the use of multiple two-dimensional lattices to allow for multiple curves in the deformation. An example of this can be seen in Figure 2.2 where an 'S' shape is formed into a spherical object with the use of two two-dimensional lattices welded together and extruded to form a three-dimensional lattice. Although EFFD allowed for more complex lattices, merging two or more lattices presented some continuity problems. However, EFFD did not allow for lattice augmentation for non-prismatic volumes due to continuity problems, making it less stable and slower than FFD.

Chang and A. [14] introduced a new approach to the deformation of objects by using a single Bezier [15] curve. Chang and A. [14] method eliminated the multiple control points along a specific curve and used a single Bezier curve that was deformed by repeatedly applying affine transformations in the space. This approach helps to generalise the deformation and reduce the amount of control points required, thus significantly decreasing the computing time required to deform a solid. The approach still lacked the capabilities of deforming a complex object without applying a global deformation.

MacCracken and K. [16] further improves on the previous methods by defining the deformable space through a volume analogy of subdivision surfaces. This subdivision methodology is an extension of Catmull and Clark [17]. In Catmull and Clark [17] method, the lattice is repeatedly refined, generating a sequence of lattices that create a three-dimensional space. This is achieved by subdividing each polygon in the lattice into smaller polygons resulting in a smoother more detailed three-dimensional lattice. This refinement procedure allows for the parameterization of an embedded point. As the user deforms the points on the lattice, a new deformation is created, and the embedded points are relocated in the new deformed space. To further understand this method, there are four steps that this approach follows when deforming an object:

- **Construction of the lattice:** The user must select the type of lattices that will be used (cylindrical, hexagonal, ...) and merge them. The user also has the possibility to individually create cells and attach them face-by-face to create a lattice.
- **Placement of the lattice:** After creating the lattice, the user must place it around the solid they want to deform. This lattice can also be placed around a specific solid part without deforming the rest of the components.
- **Refinement of the lattice:** Once the lattices have been placed, each embedded point inside the lattice is assigned with a pointer to its respective cell and position.
- **Deformation of the lattice:** The lattices are then deformed by moving its vertices. This deformed lattice is refined n times, and each vertex is extracted. The vertices on the new deformed figure are used to calculate the position of the points embedded in each respective cell.

This proposed method is a significant improvement compared to previous works due to increased control over the lattice shape and location, allowing for more complex deformations. The approach still has the discontinuity problem during mapping, creating problems at extraordinary vertices such as irregular connectivity after the refinement process. This inconsistency can result in visual artifacts such as bump, pinches or uneven curvatures. Furthermore, manual lattice construction can be quite laborious and tedious for complex deformations.

Moccozet and Thalmann [18] approach aimed to overcome the geometric limitations of the FFD previously shown by using an approximation method. They proposed a method called Dirichlet Free-Form Deformation (DFFD) based on Delaunay and Dirichlet diagrams. Their work takes any set of points specified by the user and computes the neighbours based on Delaunay triangulations. These points are then used as Bezier control points to manipulate the connectivity line between one another. The other neighbour displacement can be interpolated using Delaunay neighbour displacement. Although Moccozet and Thalmann [18] approach was initially designed for hand simulations, its ability to remove previous constraints on the location of control points allowed for more realistic deformations when used with other solid objects. The major shortcoming of this approach was the singularities in these interpolating functions, creating undesirable deformations.

A different approach to overcome geometric limitations of FFD was presented by Singh and Eugene [19]. This approach was more surface-oriented, allowing for realistic deformations. The method used parametric curves called 'wires' to deform any surface of the object. This algorithm proved to be simple and efficient by having only two stages in the deformation process. The first one is bounding the object to a set of wires, and the second stage manipulates the object by alteration of the wire's length or radius.

Singh and Kokkevis [20] further improved the method by addressing the continuity problem and affine deformations present in Singh and Eugene [19]. Singh and Kokkevis [20] method deforms an object by controlling a polymesh. By creating a surface-oriented FFD, their approach enables the representation and manipulation of any underlying object using a singular control polymesh as seen on Figure 2.3. This process is divided into three different steps:

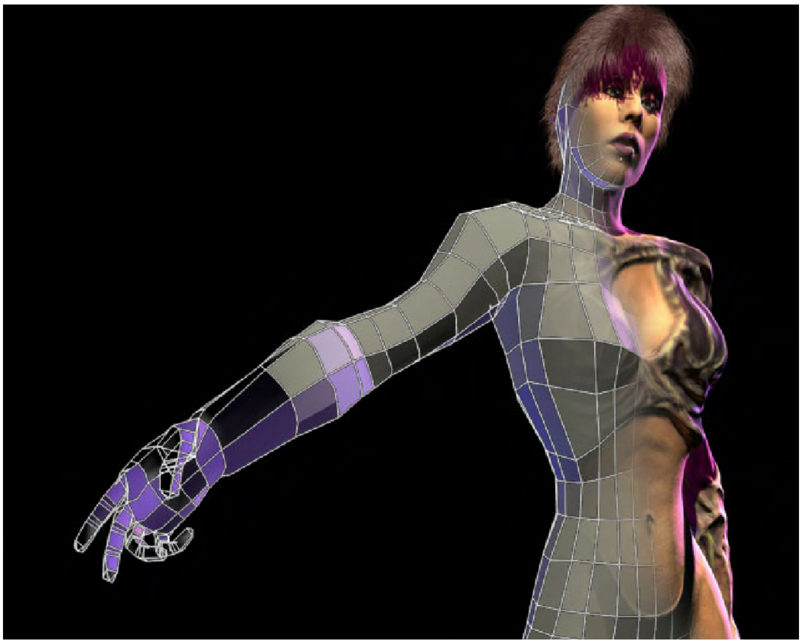


FIGURE 2.3: Surface-oriented FFD overlaid on a complex mesh [20]

- **Bind Face:** During the initial phase, the user specifies the face that will be deformed. This face becomes the 'driver' face, and an identical copy of it becomes the reference surface.
- **Registration phase:** The registration phase computes how much each surrounding face will be affected by the driver face. This influence weight is calculated based on the distance from the driver's face.
- **Deformation Phase:** The deformation phase happens once the user manipulates the driver face. This procedure maps each point on each of the phases undergoing manipulation. Each point in the deformed phase is then changed in position and orientation based on the weight of their relative face.

This approach exhibits a greater similarity to contemporary non-deformation methodologies, where a new polymesh is constructed to accurately replicate the original object's shape. However, with the widespread use of highly detailed meshes, Singh and Kokkevis [20] approach is no longer well suited due to the high amount of computation needed to achieve a fine detailed transformation of all the elements.

2.1.3 Current Non-Rigid Deformation Methods

While the previously outlined approaches are capable of non-rigid deformation on solids or figures, they fall short to current state of the art approaches it comes to amount of deformation the can compute or processing speed. These earlier methods are unsuitable for calculating the skirting line due to limitations such as lack of multiple degrees of freedom for single fibre movement and the presence of visual artifacts like uneven curvatures or bumps. For this reason, different methods have been proposed over the last two decades to improve and overcome this challenges presented in non-rigid deformation methods. One of the main focuses is how the deformation field is represented. The following subsections look at the most significant papers of each of the methods. This section aims to provide a clear understanding of the most widely used deformation fields and their respective strengths, assisting in the selection of a suitable non-rigid deformation method for aligning the before and after skirt images.

2.1.3.1 Pointwise Location Field

A common practice is to deform a deformation field by directly using the position of the target point and computing the deformation via optimisation. This results in a regularization of the variables through shape preservation (Liao et al. [21]) or local similarity. [21] used this approach to reconstruct a complete 3D model obtained over time with a depth camera. His approach assumes that the deformation is continuous during a short period of time. More approaches have been put forward trying to improve the shortcomings of this deformation field ([22]; [23]), however, as a result of the deformation's physical characteristics, a node is unable to move autonomously. For this reason, pointwise location methods can result in unnecessary degrees of freedom, causing undesired large local deformations.

2.1.3.2 Pointwise Affine Transformation

Rather than using the location of the points as variables, other deformation fields interpret them as affine transformations. Allen et al. [24] method employs displaced subdivision surfaces which uses locally different affine transformations. Compared to the previous methods discussed ([23], [22] and [21]) this model can deal with more complex deformations that include local rotations. This approach determines the affine transformation for each node using Equation (2.1).

$$\hat{\hat{x}} = R_i x_i + t_i \tag{2.1}$$

Where the target location is denoted as $\hat{x}$, while the current location of the node is represented by x. The affine transformation responsible for rotation is expressed as R, where $R_i \in \mathbb{R}^{3x3}$. The translation component is represented as t, with t_i residing in a three-dimensional space, denoted as $\mathbb{R}^3$. This framework allows us to precisely describe and manipulate spatial transformations within three-dimensional contexts.

Li et al. [25] improves upon previously proposed methods ([26]) by using a non-rigid registration model with constraints on the location and transformations. While the model exhibits the capacity to generate more intricate deformations, particularly through local rotations, it faces a challenge in terms of computational efficiency. Storing the deformation field within each of the source points normally leads to a high amount of computational memory consumption caused by the high amount of variables.

2.1.3.3 Graph Oriented Techniques

Sumner et al. [27] introduced a method that utilises an 'embedded deformation' graph. The graph maps the internal structure or topology of an object while strategically reducing the degrees of freedom, increasing the performance of the method. The graph is built by connecting multiple nodes through edges, which signify the proximity and relationships among these points. These nodes serve as control points for the deformation process, and their final positions are determined through the application of affine transformations. This affine transformation for node j is represented in Equation (2.2).

$$\hat{x} = \sum_{j=1}^{m} w_j(x_i)[R_j(x - g_j) + g_j + t_j] \tag{2.2}$$

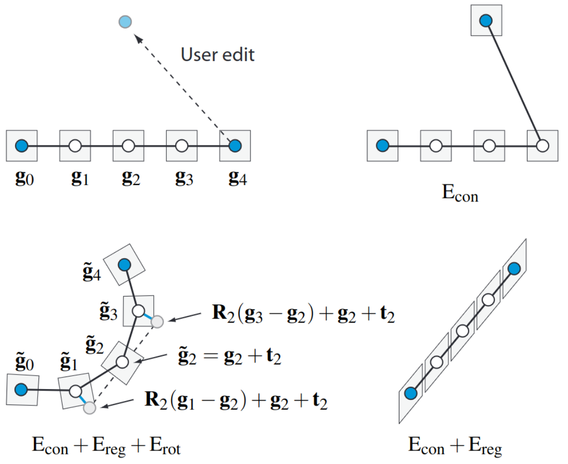


FIGURE 2.4: Deformation of nodes in embedded deformation [27]

Where the node positions are given by $g_j \in \mathbb{R}^3$, $j \in 1..m$. The weight w_j is dependant of each node's position, thus, Sumner et al. [27] precomputes it by limiting the influence of each node to the k nearest neighbours. As demonstrated in Equation (2.3):

$$w_j(x_i) = (1 - ||x_i - g_j||/d_{max}), \tag{2.3}$$

Where d_{max} represents the distance to the $k+1$ nearest node. An optimisation is then applied to the deformation graph, shown in Equation (2.4), to prevent unnatural behaviours in the deformation.

$$\underset{R_1,t_1,...,R_v,t_v}{\text{argmin}} \quad w_{con}E_{con} + w_{rot}E_{rot} + w_{reg}E_{reg} \tag{2.4}$$

Where E_{con} represents the location of the node moved by the user. E_{reg} is the regularization term that ensures consistency in the deformation caused by adjacent nodes. E_{rot} is the sum of the rotation errors of all the transformations in the graph. An example of this can be seen in Figure 2.4. Each optimisation term will be described in more detail in Section 2.1.4.

This method can also be used to match incomplete or noisy scans from 3D data as seen on Li et al. [28] and Li et al. [29]. Bonarrigo et al. [30] uses a very similar method but improves upon it by reducing the number of degrees of freedom each node has. Each node

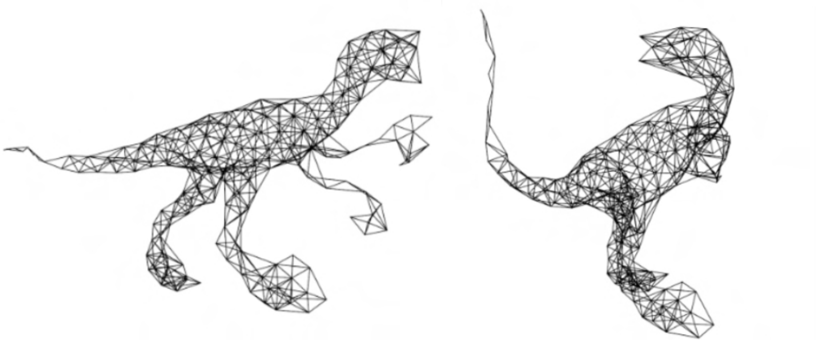


FIGURE 2.5: Deformation graph of dinosaur with pose changes [27]

on the embedded graph has 6 degrees of freedom compared to 12 from Li et al. [29] work, increasing its performance while maintaining a high accuracy.

Using fewer nodes for deformation in embedded graphs significantly reduces the number of variables present in pointwise transformations. This approach is notably user-friendly, resembling the 'wires' concept introduced by Singh and Eugene [19]. It allows users to intuitively define deformations by manually dragging control nodes within the graph.

2.1.3.4 Reproducing Kernel Hilbert Space (RKHS) Based Methods

Myronenko et al. [31] and Myronenko and Song [32] introduced Coherent point drift (CPD). This probabilistic method considers the alignment of two points with one set of points represented as a mixture model with Gaussian centroids initialized from the second model. The displacement field is regularly created with the Reproducing Kernel Hilbert Space (RKHS) to prevent an infinite amount of solutions. Myronenko et al. [31] and Myronenko and Song [32] model the deformation using a displacement field 'df' represented as Equation (2.5):

$$\hat{x}_i = x_i + df(x_i) \tag{2.5}$$

While recent studies, such as those by Ma et al. [33] and Ma et al. [34], have focused on reducing the complexity of RKHS based methods, these approaches inherently face higher computational demands due to the sophisticated mathematical framework of Hilbert

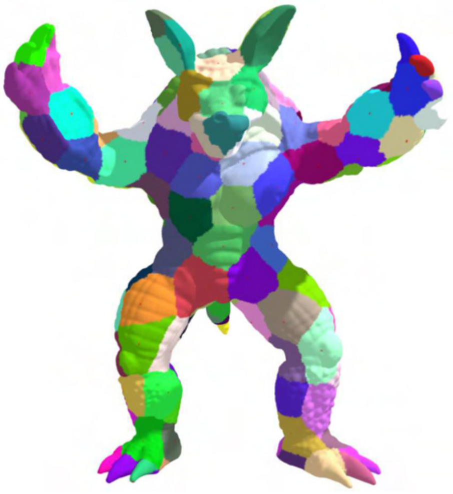


Figure 2.6: Patches on Armadillo Cagniart et al. [36]

spaces. Additionally, the implementation of RKHS based methods tends to be more intricate compared to graph-based methods. Graph-based methods offer a more straightforward and user-friendly approach, whereas RKHS based methods require a deeper understanding of complex mathematical concepts for effective implementation.

2.1.3.5 Patch Based Methods

Another approach to surface deformation is by combining sets of points or meshes into a single cluster to undergo a rigid transformation (Huang et al. [35] and Cagniart et al. [36] and Slavcheva et al. [37]). An example of these sets of points or meshes called patches by Cagniart et al. [36] can be seen in Figure 2.6. These algorithms work by first segmenting the surface into a set of patches or clusters, these clusters can be partially overlapping or non-connecting. The patches aim to follow the intrinsic structure of the deforming object. In order to create the patches, a random point is selected, working as the initial centroid for the first patch. From here, the subsequent patches are formed from the vertices that lie in the most patch boundaries. These patches are generated from their midpoint until they reach a threshold radius. This process will continue until the entire figure is covered with these patches. The deformation of each patch is computed by combining the Gaussian weights calculated from the Euclidean distance between the point and the patch.

This method operates on the assumption of local rigid deformation within each patch, an assumption that may not be true for some objects. Additionally, enhancing the method's flexibility by increasing the degrees of freedom requires reducing the size of each patch. However, this adjustment increases the computational costs of the method.

2.1.3.6 Grid Based Methods

Fujiwara et al. [38] introduced an innovative approach to achieve locally rigid and globally non-rigid registration by employing a derived FFD. During their approach, a secondary grid is superimposed over the primary grid, which contains all the FFD control points. This secondary grid is further segmented into smaller sampling regions, each encompassing a control point from the primary grid. The deformation is executed by translating these control points. This translation is guided by a rigid transformation aimed at minimising the differences between the Sign Distance Field (SDF) of the source and target objects. The SDF can be written at a point in $x \in \mathbb{R}^3$ as Equation (2.6):

$$SDF(x) = (x - x^*) \cdot n^* \tag{2.6}$$

Where n is the normals of the surface at x^*. Further approaches achieve local rigidity or higher efficiency by using a coarser grid ([39], [37]), however, grid based methods become computationally expensive if a high-resolution grid is required.

2.1.3.7 Spline Function Based

Spline function-based methods use a deformation field that is defined by a spline function to deform the object. These approaches are based on the thin-plate spline function by Bookstein [40]. Chui and Rangarajan [41] developed a robust point matching that uses affine mappings to deform 2D images. Rather than applying rigid deformations to each cell, such as grid-based methods, spline methods use non-rigid point matching to bend cells on a grid map to align with the target's deformation.

The non-rigid points can be controlled by the user within a single grid with one-to-one correspondences ([42], [43]). These methods, similarly to the graph-based methods, have proven to be particularly effective due to their user-friendly capabilities and the realistic deformations they can generate.

2.1.4 Alignment Quality

When performing non-rigid deformation, finding the optimal solution within a deformation field for effectively aligning two surfaces is crucial. This process requires the implementation of specific criteria to evaluate alignment quality. A common strategy employed in many methods is the assessment of alignment errors, which is based on the measurement of distances between the two surfaces in the space. This section explains some of the optimisation methods used in different deformation fields to find an optimal alignment.

One of the most common approaches to determine the optimal deformation is by focusing on minimising a specific target function. This target function is typically expressed as Equation (2.7). Where E_{align} assesses the degree of alignment error between the deformed source point and the target point. E_{reg}, on the other hand, is a term that imposes certain constraints on the deformation, such as smoothness. The parameter α serves as a balancing weight between the alignment error and the regularization term.

$$EX = E_{align} + \alpha E_{reg} \tag{2.7}$$

2.1.4.1 Alignment

A commonly used and straightforward alignment term is derived from calculating the distance between each of the surface points and its respective point on the target surface (Equation (2.8)). Commonly used in Pointwise Fields, this point alignment term is used among multiple approaches ([44], [45], [46], [47] and [48].

$$E_{pp} = \sum_{i}^{m} w_i ||x_i - \hat{x}_i||^2 \tag{2.8}$$

Where the weight (w_i) is used to determine the importance of the points using the quality of the correspondence. Although this error implementation is straightforward, it is sensitive to outliers and noise. For this reason, some other methods measure the alignment using a point-to-plane distance ([29], [49]). Normally formulated as Equation (2.9):

$$E_{ppl} = \sum_i w_i (n_i \cdot (x_i - \hat{x}_i))^2 \tag{2.9}$$

Where n_i is the surface normal of the target at $\hat{x}_i$. Function $(x_i - \hat{x}_i)^2$ refers to the point-to-plane distance also known as ICP. By incorporating information about surface orientation, Point to plane error (Equation (2.9)) has proven to be more robust, providing better alignments ([50]).

Similar to the function referenced in Equation (2.8), embedded graphs focus on optimising the location of control points. These positional constraints are applied to user-specified points to ensure alignment with required positions. Sumner et al. [27] defines their positional constraint as shown in Equation (2.10):

$$E_{con} = \sum_{X=1}^{m} ||\hat{x}_i - x_i||_2^2 \tag{2.10}$$

2.1.4.2 Regularization

Regularization terms are crucial in achieving adequate rotations, ensuring smoothness, or maintaining rigidity during the deformation processes. These constraints, often represented as a weighted sum, can either focus on multiple aspects or concentrate on a singular aspect, optimising for a more realistic and natural deformation.

To prevent unrealistic distortion, some deformation fields implement a smoothness term that enforces a more natural transition across the points or nodes. Point-wise fields achieve this by penalising the affine transformations on neighbouring points. Allen et al. [24], Pauly et al. [45] and Amberg et al. [47] use a smooth regularization defined in Equation (2.11), where the affine transformations are presented as R and $||_F$ used to quantify the difference between rotation matrices.

$$E_{smooth} = \sum_{i} \sum_{j \in \mathcal{N}(i)} ||R_i - R_j||_F{}^2 \tag{2.11}$$

Similarly to previous smoothness equations (Equation (2.11)), deformation graphs impose smoothness on the deformation by applying a regularization term Equation (2.12) on neighbouring nodes. However, instead of comparing the transformations directly, deformation graphs focus on comparing the positions of the transformed nodes.

$$E_{reg} = \sum_{i} \sum_{k \in \mathcal{N}_i} ||R_j(g_i - g_j) + g_j + t_j - (g_i + t_i)||_2^2 \tag{2.12}$$

Furthermore, embedded deformation graphs implement a second regularization term. This term controls the amount of rotation applied to the nodes to prevent unrealistic or physically impossible shapes. This local rigidity regularization term is described as Equation (2.13). Where R_i denotes the affine transformation matrix for each node.

$$E_{rigid} = \sum_{i} Rot(R_i) \tag{2.13}$$

$$\begin{aligned} Rot(R_i) = (r_1 \cdot r_2)^2 + (r_1 \cdot r_3)^2 + (r_2 \cdot r_3)^2 + \\ (r_1 \cdot r_1 - 1)^2 + (r_2 \cdot r_2 - 1)^2 + (r_3 \cdot r_3 - 1)^2 \end{aligned} \tag{2.14}$$

Equation (2.13) evaluates the rotation matrix used for each node. By analysing the columns on the rotation matrix (r_1, r_2, r_3) for each of the nodes, ED can quantify the deviation from the ideal rotation.

Sorkine and Alexa [51] introduced a different local rigidity equation, allowing for distances between the points to be maintained during the deformation. They define this local rigidity as Equation (2.15).

$$E_{ARAP} = \sum_{i=1}^{m} w_i \sum_{j \in \mathcal{N}} w_{ij} ||(\hat{x_i} - \hat{x_j}) - R_i(x_i - x_j)||^2 \tag{2.15}$$

Where w_i individually weighs vertices according to their respective areas, and w_{ij} takes into account the geometric structure of the mesh or surface, resulting in deformations that better adhere to the underlying physical properties.

2.1.5 Control Points or Nodes

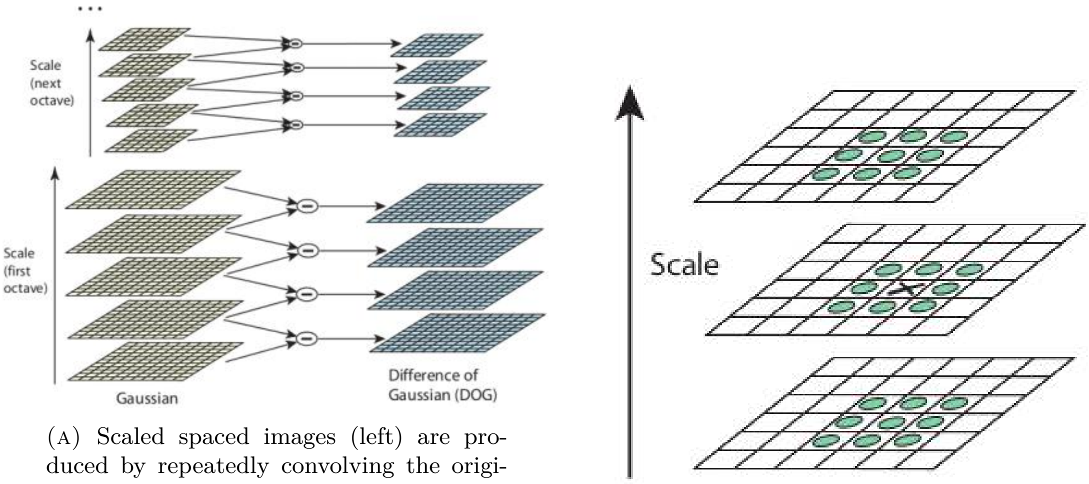
Features are commonly used to align two or more images. These features work as the 'control points' or 'nodes' used by previously mentioned non-rigid deformation methods, to manipulate the original image and transform it to meet a specific output. In images, features detect the relevant parts of the image by performing corner detection, binary patterns, and oriented gradients, among others, and represent this data in compact feature vectors for easier readability. Each method utilises a different approach to extract the maximum number of features in each image. The greater the number of features, the better the alignment between the skirted and non skirted images of the fleece.

This section will examine numerous traditional and machine learning feature detection algorithms, additionally, it will give an overview of different optical flow methods. Optical flow algorithms allow the user to track the motion of objects or features across consecutive frames of a video or between different perspectives in images.

2.1.5.1 SIFT

SIFT is one of the most widely used feature detection methods for pattern recognition. Lowe [52] method works by transforming the image into a set of local feature vectors. Each vector is invariant to translation, rotation, scaling and in most cases illumination variations.

The SIFT algorithm is mainly divided into four different parts.



(A) Scaled spaced images (left) are produced by repeatedly convolving the original image. Difference of Gaussian (right) is then obtained by subtracting adjoined images

(B) Local extrema obtained from pixel comparison to closest neighbours

Figure 2.7: Detection of Scale-space extrema in SIFT (Lowe [52])

1. **Detection of Scale-space extrema:** Firstly, the image is subdivided into different scales, where each of the octaves or scales contains different representations of the image with increasing blur. Once the images are blurred, each one is compared to the images immediately preceding and following it within its scale. This process, known as the 'Difference of Gaussian', is employed to identify potential keypoints as shown in Figure 2.7a. Subsequently, each pixel in the images obtained is compared to the eight surrounding neighbours and the nine pixels in the previous and next scale (Figure 2.7b). If the pixel is determined as a local extrema, meaning the brightest or darkest compared to its neighbours, the algorithm considers it a keypoint.

2. **Keypoint localization** After obtaining the keypoints from the previous step, [52] removes all the keypoints located along the edge.

3. **Orientation assignment** A specific neighbour is assigned to each of the keypoints to measure intensity changes around the keypoint. An orientation histogram is then created to cover all possible angles. From this histogram, the highest peak indicates the orientation of each of the keypoints.

4. **Keypoint descriptor** Around each of the keypoints, the algorithm creates a 16x16 window and divides them into 4x4 descriptors as shown in Figure 2.8. The algorithm then obtains the dimensional vector inside each 4x4 descriptor (each 4x4 descriptor

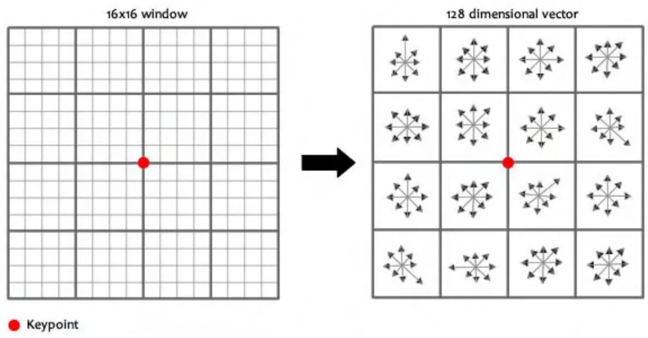


Figure 2.8: A 16x16 array created around each of the keypoints. This 16x16 array is further divided into a 4x4 descriptor with each descriptor having an 8-dimensional vector.(Lowe [52])

has 8 different vectors) to obtain the final descriptor. These 128 values (4x4 window x8 vectors) represent the descriptors for each keypoint.

2.1.5.2 SURF

Bay et al. [53] introduced a faster approach using similar principles to Lowe [52] method. Unlike the more computationally intensive techniques used in SIFT, SURF employs a method that quickly identifies points of interest in an image.

Similarly to SIFT, SURF detects potential keypoints by analysing changes in image intensity across various scales. However, instead of the iterative blurring and downsampling the image, SURF uses integral images for efficient keypoint identification. This allows SURF to utilize box filters of any size on the original image at constant computational cost, enhancing processing speed. Unlike SIFT's image pyramid, SURF does not reduce image size but scales up the filter size (9×9 to 15×15 to 21×21), allowing for the extraction of features at different scales without loss of speed. For orientation determination, SURF does not create a fixed 16x16 window around the keypoint but instead analyses the neighborhood within a dynamically scaled circular region using Haar-wavelets. These wavelets are effective in capturing abrupt intensity changes, facilitating the detection of edges or texture variations across the scale space.

The dominant orientation of the wavelet is estimated by calculating the sum of the wavelet responses. This approach ensures that the orientation assigned to each keypoint is the most

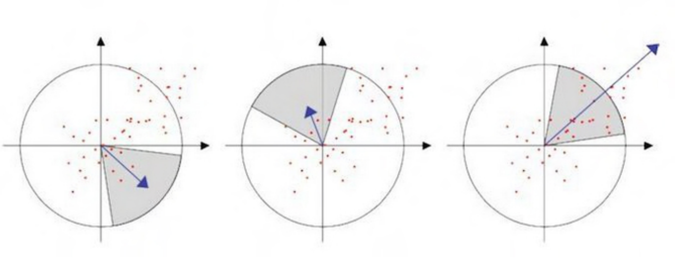


Figure 2.9: Haar-wavelet responses plotted around the specific keypoint in a circular neighbourhood with a radius of 6s, with s the scale at which the keypoint was detected when up-scaling or downs-scaling the filter size (Bay et al. [53])

significant one in its local context. An example of this orientation assignment can be seen in Figure 2.9.

2.1.5.3 FAST

Rosten and Drummond [54] continued to build on the processing speed of previous algorithms by introducing Features from accelerated segment test (FAST). Rosten and Drummond [54] method identifies possible corners in the image by considering a circle of sixteen pixels around each point or a Bresenham circle with a radius of 3. To select whether a point is a corner or not, the algorithm analyses if there is a set of continuous pixels around the point that are brighter than the selected pixel plus a threshold or if they are all darker.

To further speed up the algorithm, Rosten and Drummond [54] introduced another requirement to the neighbours of the pixel. If three of the four closest pixel neighbours (above, below, left, right) are not above or below a threshold then the interest points are not considered a corner. Otherwise, the point is considered a possible corner and all the 16 neighbours are analysed.

2.1.5.4 KAZE

Alcantarilla et al. [55] uses a similar approach to SIFT, by discretizing the scale space in a series of octaves. In contrast to SIFT, KAZE does not downsample or progressively blur each of the octaves. Potential keypoints are determined by points that are local maxima or minima (brightest or darkest) in both spatial and scale dimensions. For each keypoint, KAZE calculates the gradient of each vector orientation with the use of a point vector space instead of a histogram as used in SIFT. The vector with the highest value located on the dominant orientation is selected as the orientation of that specific keypoint as seen in Figure 2.9, similarly to Bay et al. [53] method.

2.1.5.5 ORB

Rublee et al. [56] uses Rosten and Drummond [54] approach of extracting interest points with the use of a Bresenham circle. However, it addresses the problem of not being scale invariant. To solve this, Oriented FAST and Rotated BRIEF (ORB) build a multiscale pyramid image. This pyramid image is the representation of the same image at different scales or resolutions. ORB then becomes partially scale invariant by extracting FAST keypoints at each of the different scales of the image.

2.1.5.6 BRISK

Leutenegger et al. [57] method also utilises the FAST algorithm with the multiscale pyramid proposed by Rublee et al. [56]. In Leutenegger et al. [57] method, FAST is firstly utilised to detect potential points of interest. Then, the points selected are subjected to a non-maxima suppression in scale-space. To compute the receptive descriptors for each of the keypoints, BRISK performs a brightness comparison test with its neighbours. This approach is used to allow for rotation and scale invariance.

2.1.5.7 FREAK

In contrast to ORB, fast retina keypoint (FREAK) utilises a sampling pattern inspired by the human retina, employing a set of equally spaced circular grids with expanding area. This design ensures that the sampling circles are denser near the centre of the grid and become sparser as the radius increases, mimicking the distribution of photoreceptors in the human eye. To obtain the FREAK description, the pixels within this pattern are compared. For each pair, the algorithm checks whether the intensity of one pixel is greater

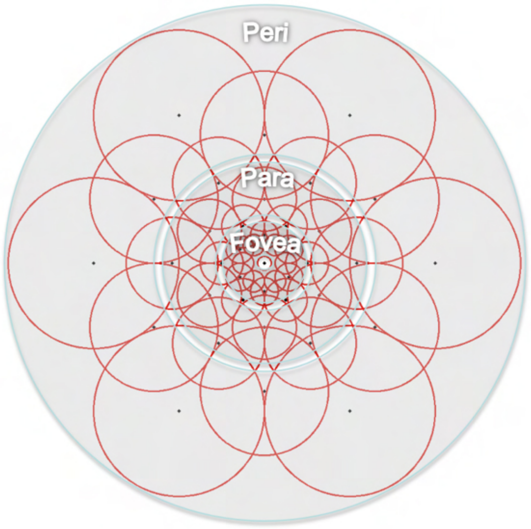


Figure 2.10: Sampling patter for FREAK descriptors. The grid is created from concentric circular grids with expanding areas, simulating the human retina.

than the other. These comparisons are binary, leading to a result that is either 0 or 1, depending on which pixel is more intense. The final FREAK descriptor is this binary string, with each bit representing the outcome of a pixel intensity comparison.

2.1.5.8 Learning Based Methods

In recent years, machine learning has significantly advanced feature detection in computer vision, surpassing traditional methods. These advanced algorithms focus on distinct steps in the processing chain: detecting features, determining their orientation, and extracting robust representations for cross-image matching. A key example is the Learned Invariant Feature Transform (LIFT) [58], which employs a three-part structure: a Detector, an Orientation Estimator, and a Descriptor. Each component is built on CNN, drawing inspiration from contemporary models. This approach marks a shift from conventional handcrafted features to data-driven, learned features, offering greater adaptability and robustness in various imaging conditions. Further learning-based methods will be explained in the following sections. These methods have gained the most popularity due to their superior performance compared to traditional feature-matching algorithms. When trained on appropriate data, learning-based methods have the potential to handle greater deformation, leading to better alignment of the before and after-skirt images, allowing for identification of the contaminated parts of the fleece.

Superpoint

Superpoint [59] addresses the image matching challenge by capitalizing on inherent regularities in the physical world. It recognises that the 3D world exhibits smoothness and planarity, and all correspondences for a static scene stem from a singular epipolar transform. Additionally, it considers the likelihood of certain poses and the projection of salient 3D points, such as corners or blobs, onto 2D keypoints. DeTone et al. [59] model is designed to identify all correspondences between reprojections of the same 3D points while also detecting keypoints that remain unmatched due to occlusion or detector failure. This approach demonstrates a better understanding of the physical constraints in feature matching, ensuring the identification of single correspondences per keypoint and the acknowledgement of inherent unmatched keypoints.

SuperGlue

SuperGlue [60], a recent innovation in local feature matching, stops the use of traditional nearest neighbour search methods used in previously mentioned methods and introduces a learning-based approach that employs a GNN for local feature matching. This method takes two sets of interest points with their descriptors and utilises a data-driven approach to learn their matches. However, it relies on the repeatability of detected interest points, which can be a limitation in non-distinctive regions, highlighting its dependency on the efficacy of the detector used.

LoFTR

Building upon the advancements made by Sarlin et al. [60], Local Optical TRansformer (LoFTR) introduces a Transformer-based method aimed at producing dense matches in low-texture areas, where traditional feature detectors often falter. This approach has been demonstrated to outperform state-of-the-art learning-based methods on both indoor and outdoor datasets.

2.1.5.9 Optical Flow

Another approach to feature detection is the use of optical flow methods, which provide insights into scene dynamics and structure. By utilising a sequence of images, optical flow is capable of determining the spatial movements within the image. These methods are highly efficient when applied to multiple consecutive frames. However, due to the positioning of the shearer's hands on top of the fleece during skirting, it is not possible to capture multiple frames during the skirting process. Nevertheless, optical flow methods are still capable of determining the deformation of multiple features given only two images, allowing for the alignment of the before and after-skirt image.

One of the foundational methods in optical flow is the Lucas and Kanade [61] algorithm. This algorithm assumes that the flow is constant in a local neighbourhood of the pixel under consideration and solves the optical flow equations for all pixels in that neighbourhood, by the least squares criterion.

Teed and Deng [62] proposed one of the most recent advancements in optical flow estimation using deep learning. Teed and Deng [62] method uses a recurrent neural network to refine the flow estimation iteratively. This method aggregates and updates a 2D correlation volume for all pairs of pixels between two images, leading to state-of-the-art performance on optical flow benchmarks. Teed and Deng [62] method is capable of handling large displacements and is adaptable to a wide range of datasets and environments.

Jiang et al. [63] improved Teed and Deng [62] work by addressing one of its major limitations: the need for a dense correlation volume to ascertain pixel displacement between images. While this dense correlation is valuable for precise estimation, its significant computational load and memory requirements pose challenges for efficient model training and deployment. Jiang et al. [63] method introduces the use of sparse correlations or matches, effectively reducing the computational demands of the RAFT algorithm without compromising accuracy. Their approach, names Sparse Correlation Volume (SCV), has demonstrated comparable results to RAFT without the need for a dense correlation, effectively reducing the computational demand.

2.2 Physics Based Deformation

Unlike traditional methods that rely on control points or 3D meshes for the deformation, physics-based methods offer a more dynamic and versatile approach. These techniques, predominantly used in computer graphics, enable a more realistic representation of the transformations that materials or fluids undergo. By using the material properties, these methods are capable of simulating how the object would behave and deform based on a specific force rather than using control points. One key transformation that benefits from this approach is the process of fracture. Fracture is distinguished by the splitting of a solid substance into several pieces. This occurrence is not only common, but also significant in understanding the structural integrity and durability of materials under stress. Since Terzopoulos and Fleischer [64] work on fracture and viscoelasticity, multiple methods have been developed for the animation of objects and fluids.

In the context of wool fleece skirting, physically-based methods are particularly relevant. These models are advantageous because they allow for the deformation or fracture of material realistically, without the need for an explicit definition of the object's characteristics or manual control over fracture propagation. This aspect is crucial in our research, where understanding the deformation behaviour of wool fleece can significantly enhance the accuracy of skirting line detection. The core techniques used in physics-based simulation for material deformation are explained in the following subsections. Each of these approaches offers unique insights into how to propagate fractures in an object or material.

2.2.1 Mass-Spring Models

One of the earliest approaches to deformation animation was Norton et al. [65] method. Norton et al. [65] model and Hirota et al. [66] propose a physical model framework that starts by dividing the original object into a set of nodes. Every node possesses its individual mass and position, and they are interconnected in pairs by springs, each characterized by its own rigidity, damping coefficient, and equilibrium distance.

Despite their innovative approach, these models faced substantial limitations. The sudden elimination of springs could result in conspicuous visual discrepancies; issues such as shearing and bending were not directly addressed by the spring systems. Furthermore, the exact surfaces of fractures were not explicitly known but were approximated based on the mass-spring system's configuration. This meant that the depiction of cracks was not only limited by the system's configuration and resolution but also often required the use of resource-intensive algorithms.

2.2.2 Finite Element Methods

The Finite Element Methods (FEM) breaks down the solid into discrete elements that are interconnected at specific points known as nodes. FEM then uses this finite collection of nodes to formulate the problem, resulting in a set of algebraic equations that are addressed through numerical techniques.

In the work from Terzopoulos and Fleischer [64], the authors model fracture behaviours using continuous formulations. They propose both Finite Element Methods (FEM) and finite differences for discretization, allowing for a broad spectrum of deformable properties, ranging from highly flexible to almost inelastic. This approach effectively demonstrates fracture on torn paper and cloth sheets.

Molino et al. [67] developed a virtual node algorithm that utilises a grid to introduce additional geometry, which helps avoid creating poorly conditioned elements. However, this method adds hidden nodes, potentially impacting performance.

Despite these advances in remeshing and cutting, XFEM poses additional complexities, such as floating point arithmetic in degenerate configurations and self-collision on their embedded meshes. Multiple other approaches trying to simulate fracture using FEM have shown to be among the most successful for different types of materials (Bao et al. [68]; Müller and Gross [69]; Busaryev et al. [70]; Agarwal et al. [71]; Koschier et al. [72],). However, they still present challenges such as boundary continuity, remeshing near cracks, and computational efficiency.

2.2.3 Meshless Methods

Meshless methods have emerged as a promising solution to augment FEM. In these methods, an object is discretized into a collection of points or particles with no explicit connectivity. The properties of the object are localized at these nodes, bypassing the need for remeshing and mitigating the complexities associated with cutting procedures inherent in FEM.

Lucy [73] and Gingold and Monaghan [74] pioneered one of the first meshless methods aimed at addressing astrophysical simulations, known as Smooth Particle Hydrodynamics SPH. This approach, however, wasn't widely adopted due to its instabilities and inconsistencies in resolving many problems.

Pauly et al. [75] builds upon previous methods by creating a model that could simulate cracks in both brittle and ductile materials. This innovative approach dynamically adapts the shape functions surrounding the cracks, while continuously adding surface samples during crack propagation, leading to more accurate and realistic simulations.

However, for solid mechanics, Sulsky et al. [76] approach has gained the most popularity when computing fracture in material due to its efficiency at computing realistic deformation of objects. Sulsky et al. [76] approach, called Material Point Method (MPM), is renowned for its capability of handling large deformations and discontinuities, making it a powerful tool for simulating complex physical phenomena in fracture simulations.

2.2.3.1 Material Point Method

The Material Point Method is one of the most current developments in Particle In Cell (PIC) methods. PIC, first introduced by Harlow and Welch [77], uses a large number of particles moved in an Eulerian grid to simulate the dynamics of a fluid. This Eulerian grid is a fixed, structured grid that covers the entire simulation. In contrast to Lagrangian methods, the material's physical properties are stored at the grid nodes rather than at each particle.

Although Harlow and Welch [77] method was very effective, it suffered from excessive dissipation. Brackbill et al. [78] addresses the excessive dissipation by introducing the Fluid Implicit Particle (FLIP) method. Brackbill et al. [78] method calculates the change of velocity in the grid and adds them to the change in particles instead of completely updating the velocities of the particles based on the grid, reducing the amount of information transferred every time.

Sulsky et al. [76] method extends the capabilities of FLIP, particularly for use in solid mechanics by introducing a Lagrangian-Eulerian method. In Sulsky et al. [76] method, the body or fluid is discretized into a set of material points, each of which carries properties such as mass, velocity and stress. The properties of each point are then transferred to a background grid, where the momentum equation is solved. This solution for the grid is used to update the material points' location and properties. Unlike FLIP, Sulsky et al. [76] method allows the full stress tensor to be carried by the material points, instead of being stored at the centre of each cell. This method is referred to as the MPM and has become one of the most utilised models for material deformations. The proposed method by Sulsky et al. [76] is comprised of the following steps:

Initialization phase

During the initialization phase, the object is transformed into material points with predefined properties. The primary function of the initialisation phase is to transfer the information from the material points to the nodes of an Eulerian grid. This process is accomplished using interpolation functions, which determine how the properties of the material points are spread out onto the grid nodes. This approach eliminates the need for direct particle-to-particle interactions, with particles interacting through the Eulerian grid. More recent methods utilise a Total Lagrangian MPM, proposed by de Vaucorbeil et al. [79], which incorporates a background grid that covers only the space occupied by the object.

Lagrangian phase

In the Lagrangian phase, the equations of motion are solved on the grid in an updated

Lagrangian frame. The updated Lagrangian frame means that the reference configuration for calculations evolves with the material, thus accounting for any deformation or movement that has occurred. This phase primarily uses the information provided by the initialization phase to solve the equations of motion at each grid node, producing updated values for velocity and other relevant physical properties.

Connective phase

Finally, in the convective phase, the particles or material points are updated using the newly computed properties from the grid nodes. This is again achieved through the use of interpolation or shape functions. Following the update of the material points, the grid is reset or redefined in preparation for the next time step. The updating of the material points involves the advancement of their positions based on the updated velocity and potentially other state variables depending on the specifics of the problem.

This iterative process continues for a pre-specified number of time steps or until a certain criterion is met, enabling the simulation of complex behaviours of various materials under different conditions.

While the original MPM by Sulsky et al. [76] only tracks the mass and volume of each particle, newer models track the shape of each sub-region and associate the position, Mass (m_{pos}), Density (ρ_{pos}), Velocity (v_{pos}), Deformation gradient (F_{pos}), Cauchy stress tensor (σ_{pos}), Temperature (T_{pos}), to each of the material points.

2.2.4 Fracture Propagation

Even though the material point method demonstrates remarkable accuracy in deforming materials using its Lagrangian-Eulerian model, the complexities of fracture propagation require the incorporation of specialized approaches to achieve accurate predictions of material failure. Fracture propagation modelling encompasses two main approaches: discontinuous and continuous.

Given that discontinuous approaches are predominantly tailored towards the simulation of brittle materials, such as ceramics and glass, this literature review will specifically

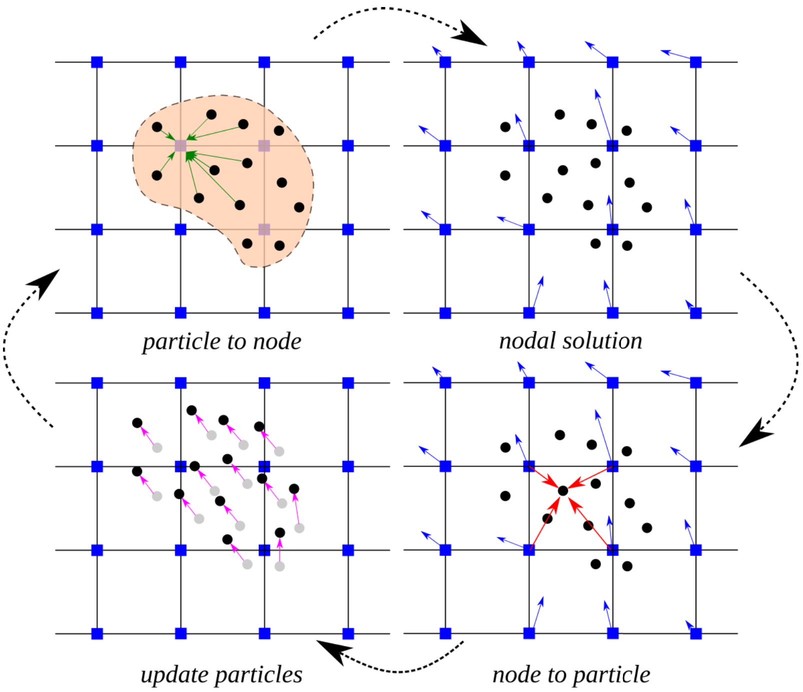


FIGURE 2.11: Material Point Method. The object is transformed into material points. Each of the points is transferred to an Eulerian Grid and the material point information is transferred to the nodes. Afterwards, the motion equations are solved for the nodes in the graph and the graph is updated. The updated nodes are then mapped back to the particles to update the position and velocity. Finally, the grid is reset and the iterative process is repeated until the object has undergone the desired deformation Kumar et al. [80]

concentrate on the research related to continuous approaches. This focus aligns with the relevance of these methods in accurately representing the deformation and fracture behaviours wool undergoes during skirting.

2.2.4.1 Continuous Approach

These models frequently employ a strain-based failure criterion to resolve issues related to perforation. This methodology asserts that a particle is deemed ”failed” when it fulfils a specific strain-based fracture condition. Moreover, adopting a strain-based failure criterion in continuous models offers the benefit of computational simplicity. Rather than tracing and defining the crack path, the fractures naturally progresses in line with the material's local mechanical response. This allows for the simulation of intricate, realistic fracture patterns without the need of detailed inputs that discontinuous models often require.

Kakouris and Triantafyllou [81] made the first move towards integrating Phase Field Fracture (PFF) into MPM, showcasing the potential of pairing both methods to simulate brittle fracture. In PFF, fractures and cracks are not treated as sharp boundaries but are instead represented by a continuous field. This phase field evolves over time to capture the extent and behaviour of cracks within the material. Their method, however, was limited to problems involving minor deformations without contact.

Wolper et al. [82] advanced the field by introducing a synergistic methodology, known as Continuum Damage Material Point Methods (CD-MPM), which amalgamates the principles of PFF and the MPM for the simulation of elastoplastic fractures. This innovative technique has demonstrated substantial proficiency in navigating sophisticated crack patterns, delivering superior visual accuracy when applied to elastic materials.

2.2.4.2 Anisotropic Fracture

While the majority of the previously discussed methodologies have centred on isotropic materials, there is an important subfield of study dedicated to the computational modelling of fracture in anisotropic materials. These materials distinguish themselves by showing unique properties that change based on the direction of observation. The defining characteristic of isotropic materials is the uniformity of their physical and mechanical properties in all directions. As a result, phenomena like crack propagation exhibit similar behaviour irrespective of the direction.

In contrast, anisotropic materials deviate from this uniform behaviour. They show significant variability in their properties depending on the direction due to differences in their structural composition, alignment, or orientation at micro or macroscopic scales. In the case of wool fracture, the direction of the fibres can greatly influence the path, speed, and pattern of crack growth and advancement.

For instance, Hakim and Karma [83] modelled anisotropy in fracture energy by adapting the phase-field approach to penalize different fracture interfaces in varying directions. Calvo et al. [84] put forward a three-dimensional finite-strain damage model for fibrous soft tissue.

In the context of MPM, anisotropic damage has proven to be effective in modelling brittle fractures as depicted in [85] and [86].

Jiang et al. [87] employs an innovative approach to simulating elastoplasticity with failure, drawing upon advanced techniques developed in the field of hair simulation. Their model is anchored by an affine Lagrangian framework, which is adeptly adapted to address the complexities of fibre fracture. Central to this approach is the introduction of a three-dimensional matrix, meticulously designed to penalize alterations in fibre length, shearing along fibres, and, crucially, frictional contact between fibres. As a result, Jiang et al. [87] method demonstrates remarkable proficiency in depicting the fracture of fibrous materials such as yarn balls breaking apart.

The approach introduced by Wolper et al. [88] is inspired by the methodology of Jiang et al. [87], particularly in its use of a similar matrix for fibre simulation. However, it diverges from [87] model by incorporating two additional terms specifically designed to penalize stretching in the fibre directions. This innovation is key to simulating fracture in orthotropic materials (fibres in three different directions, *i.e.*, wood), thereby extending the applicability of their model to a broader spectrum of shapes and materials. Furthermore, the model by Wolper et al. [88] enhances the definition of the stress differential compared to previous methods, resulting in a model that is not only more stable but also less computationally demanding. Significantly, this approach has established itself as one of the foremost methods for simulating fracture in fibrous objects and various other materials. It has demonstrated exceptional capability in realistically portraying their behaviour under diverse conditions.

2.3 Discussion

Throughout the years, the field has seen the development of various sophisticated methodologies for the alignment of figures, solids, or images, each designed to tackle the unique challenges posed by non-rigid registration. The literature review illuminates that a prevalent strategy in image alignment, particularly for non-rigid objects, involves a combined application of non-rigid deformation and feature extraction techniques. This approach has consistently demonstrated its efficacy, especially in scenarios requiring the precise alignment of images before and after significant transformations, such as those seen in the skirting of wool images.

Non-rigid deformation techniques allow for the flexible adjustment of images to accommodate a wide range of deformations, thereby ensuring a high degree of alignment accuracy. When coupled with feature extraction, which identifies and leverages key points or patterns within the images for alignment, the result is a robust method capable of handling the complexities inherent in before and after image comparisons.

Although physics-based deformation methods offer a theoretically more precise replication of complex deformations by closely mimicking the physical properties of the objects being aligned, they come with a significant computational cost. This complexity not only demands substantial processing power but also increases the time required for analysis, making these methods less feasible for applications where traditional methods offer great results.

Given these considerations, automatic detection of the skirting line could be achieved by implementing a non-physics-based approach that combines a deformation field with feature extraction algorithms. This approach would improve upon previous proposed methods by eliminating the human error present in manually annotated wool images and relying on feature extraction algorithms.

Chapter 3

Skirting Line estimation using Image Registration

3.1 Introduction

Automating the skirting line detection has the potential to reduce on-farm costs and improve product quality. A key difficulty in this process is the high degree of accuracy needed for skirting line detection. The presence of contaminants on wool after skirting affects the entire wool pipeline, impacting the textile manufacturing process complexity and eventual range of fabric that can be produced by the manufacturers. Conversely, over-skirting the fleece reduces the volume of product and can reduce the salable yield of the wool. Attempts have been made to automate the skirting line detection in wool [8, 9], however, they require large amounts of manually annotated data which is not only logistically challenging and time-consuming but also suffers from reliability issues.

In response to these challenges, this chapter looks at automating the detection of the skirting line through image registration between the before and after skirt images of a fleece. By doing so, we can identify the contaminants removed in the before skirt image, thereby eliminating the need for manual annotation, which is prone to human error.

A significant challenge when applying image registration with wool images is the substantial deformation that occurs along the skirting line during handling. The handler's technique of holding down the fleece while pulling away the section of the fleece containing contaminants results in a complex pattern of stretching and compacting fibres along the skirting line. To tackle this issue, we developed a method that includes a filtering process for feature correspondences, specifically designed to handle non-rigid deformations. This is followed by a deformation process aimed at aligning the images to extract the skirting line.

3.2 Methodology Overview

An overview of the proposed methodology can be seen in Figure 3.1, where the main contribution of this chapter is the filtering method for removing outliers present in the initial correspondences that inform the complete image deformation.

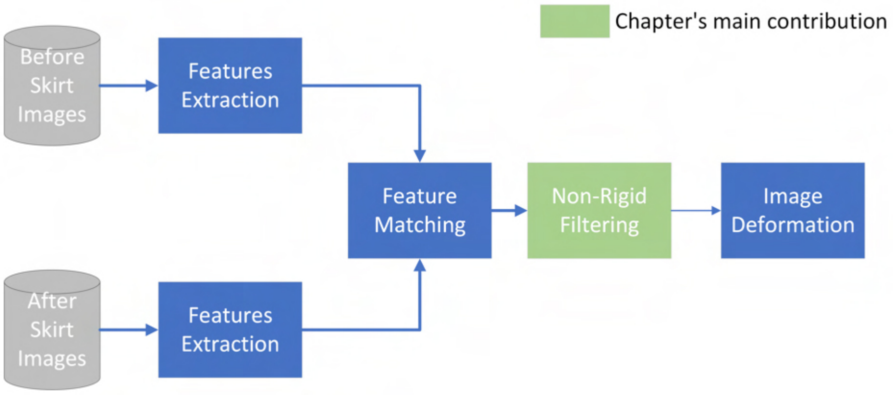


Figure 3.1: Methodology of proposed approach to perform non-rigid registration between the before and after skirt image. A set of feature correspondences are extracted from both images and filtered through the proposed non rigid filtering method. The filtered correspondences are then used to deform the after skirt image to extract the areas that were removed from the before skirt image.

The methodology of this chapter is organised as follows: Section 3.3 starts by detailing the feature correspondences that form control points for aligning the before and after skirt image. Section 3.4 outlines the proposed filtering method for removing outliers present in the initial correspondences. The filtered correspondences are then used in Section 3.5 as the control points to deform the after skirt image. Section 3.6 expands upon each of the subsections of the proposed methodology and evaluates which method outlined in the literature review is better suited for aligning the before and after skirt image.

3.3 Feature Extraction

As stated on the literature review, non-rigid registration is required to align two images that undergo local deformations. In order to find the correct alignment between the before and after skirt image, feature extraction methods are required. The feature correspondences between images work as the 'control nodes' for non-rigid deformation methods. The key requirement for selecting a feature descriptor is that it can be used to determine the quality of a potential match. The remaining parts of the proposed approach are independent of the chosen feature descriptor used for aligning the deformed image with original image.

Features are stored as two sets F^1 and F^2 where $F^i = \left\{f^i_1, \dots, f^i_n\right\}$ with $f^i_j \in \mathbb{R}^2$. F^1 corresponds to the features from the before skirt image and F^2 corresponds to the features from the after skirt image.

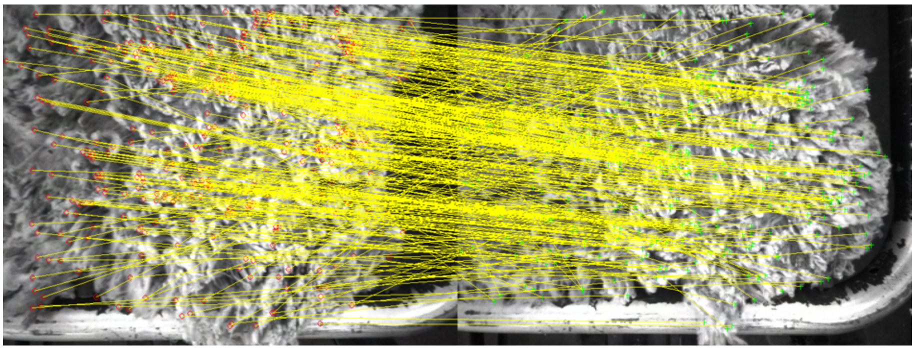


Figure 3.2: Example of putative matches between before and after skirt image. While most matches are correct, some of the matches are outliers and need to be removed.

3.4 Non-Rigid Outliers Rejection

The presence of outliers is a common problem when performing feature matching, as illustrated in Figure 3.2. Typically, this is solved using RANSAC [89] to obtain a subset of feature matches that provide a consensus with respect to a rigid transformation. While RANSAC can be reformulated for non-rigid cases [90], it embeds the computation of the non-rigid transformation in an iterative process, which is computationally ineffective.

For this reason, we propose a filter that prunes the feature matching iteratively with a small computational footprint. This filter is initiated by building graphs on the feature sets and performs the pruning based on the local rigidity difference between the two graphs.

The filter is initialised by building a graph that connects the feature points from F^1 to its k neighbours. These connections between neighbours are then concatenated into an edge set E^1 and the first graph is then defined as $\{F^1, E^1\}$. The graph on the second set of features F^2 is generated by copying the edges or connection between neighbours E^1, generating the graph $\{F^2, E^1\}$.

For each feature node present in the graph, the local rigidity changes are computed based on the formulation of the rigidity as defined in Equation (2.15). Given two nodes, f_i^1 and f_j^2, which correspond to two SIFT matches, we can redefine Equation (2.15) to:

$$\mathbf{rig}(i) = \sum_{i=1}^{m} w_i \sum_{j \in \mathcal{N}} w_{ij} ||(f_i^1 - f_j^1) - R_i(f_i^2 - f_j^2)||^2 \tag{3.1}$$

As demonstrated by [51] we can set $w_i = 1$ so that each element on the mesh is equally resistant to deformation regardless of the area. Additionally, we can establish $w_{ij} = 1$, effectively treating all edges in the graph with equal significance during the filtering process. Considering these simplifications, the local rigidity for the i^{th} node is defined as:

$$\mathbf{rig}(i) = \sum_{j \in N(i)} ||(f_i^1 - f_j^1) - R_i(f_i^2 - f_j^2)||^2 \frac{1}{||f_i^1 - f_j^1||^2} \tag{3.2}$$

Where the left-hand part of the equation is the simplified formula which quantifies the deformation operated on the edge from $(f_i^1 - f_j^1)$ to $(f_i^2 - f_j^2)$ while accounting for a rigid rotation $\hat{R}_i$ (computed using SVD similarly to Sorkine and Alexa [51]). The added right-hand part of the equation ensures the rigidity term is scale invariant.

To remove the outliers in the feature matching, the feature node with the maximum ARAP residual is pruned from the graph. The graph and the rigidity terms are then updated and the pruning process is repeated iteratively until the ARAP residual falls below a defined threshold. An example of the result after feature matching pruning is shown in Figure 3.3 and the pseudo-code is given in Algorithm 1.

Algorithm 1 Prune feature matches until local rigidity falls under threshold τ.

```
Input: $F^1$, $F^2$, $\tau$
Output: $\hat{F}^1$, $\hat{F}^2$
function ARAP filter
    build graph $\{F^1, E^1\}$
    copy edges on $\{F^2, E^1\}$
    compute rigidity terms $\mathbf{rig}(i)$ with (3.2)
    while $\max(\mathbf{rig}) < \tau$ do
        prune nodes $f_{\arg\max(\mathbf{rig})}$ in $F^1$ and $F^2$
        update graph $\{F^1, E^1\}$
        copy edges on $\{F^2, E^1\}$
        compute rigidity terms $\mathbf{rig}(i)$ with (3.2)
    end
end
```

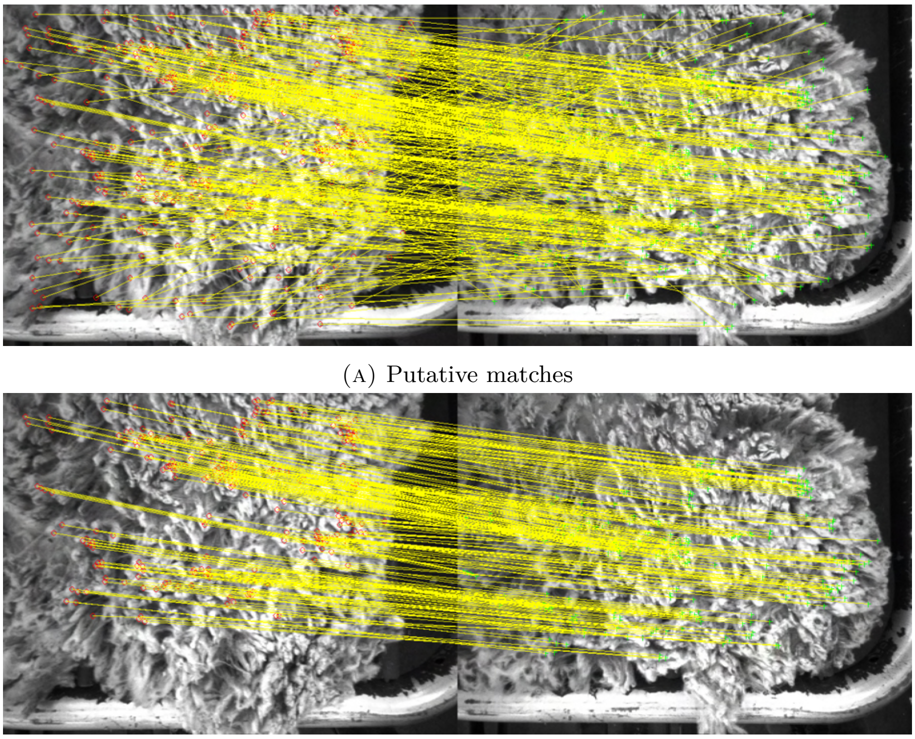
3.5 Non-Rigid Deformation

Given the filtered correspondences, the proposed approach deforms the after skirt image to find the location of the skirting line. This is achieved by utilising the correspondences as the 'control handles' for any non-rigid deformation approach. For alignment of the two images, the user can use any non-rigid deformation method. However, for the alignment of wool images an ablation study is conducted in Section 3.6.3.

The deformed after skirt image is then superimposed on the before image. The location of the skirting line is obtained by extracting the boundary of the superimposed image on the before skirt image.

3.6 Ablation Study: Feature Matching Algorithms and Non-Rigid Deformation Methods

In the following section, we conduct a comparative analysis of feature matching algorithms and non-rigid deformation methods outlined in the Chapter 2. This analysis aims to



(A) Putative matches

(B) Filtered matches

Figure 3.3: Filtered matches are shown in Figure 3.3b after implementation of proposed filtering method on the initial putative matches shown in Figure 3.3a

determine the most suitable feature matching algorithm in current literature and non-rigid deformation method for aligning the wool images with the proposed methodology.

3.6.1 Traditional Feature Matching Algorithms

Feature correspondences work as the control handles or nodes used for non-rigid deformation algorithms to alter the shape of the after-skirt wool to match the before-skirt wool. Thus, not having enough correspondences or incorrect ones will create undesired deformation in the final output resulting in wrong results. A comparative evaluation is performed to analyse each descriptors performance with wool images collected from a shearing shed. Each of the descriptors are filtered with the proposed filtering method to evaluate their performance on non-rigid data.

The data used to test the algorithms has been skirted by professional wool handlers. This allows for a better evaluation of how each algorithm performs when the wool undergoes significantly higher deformation.

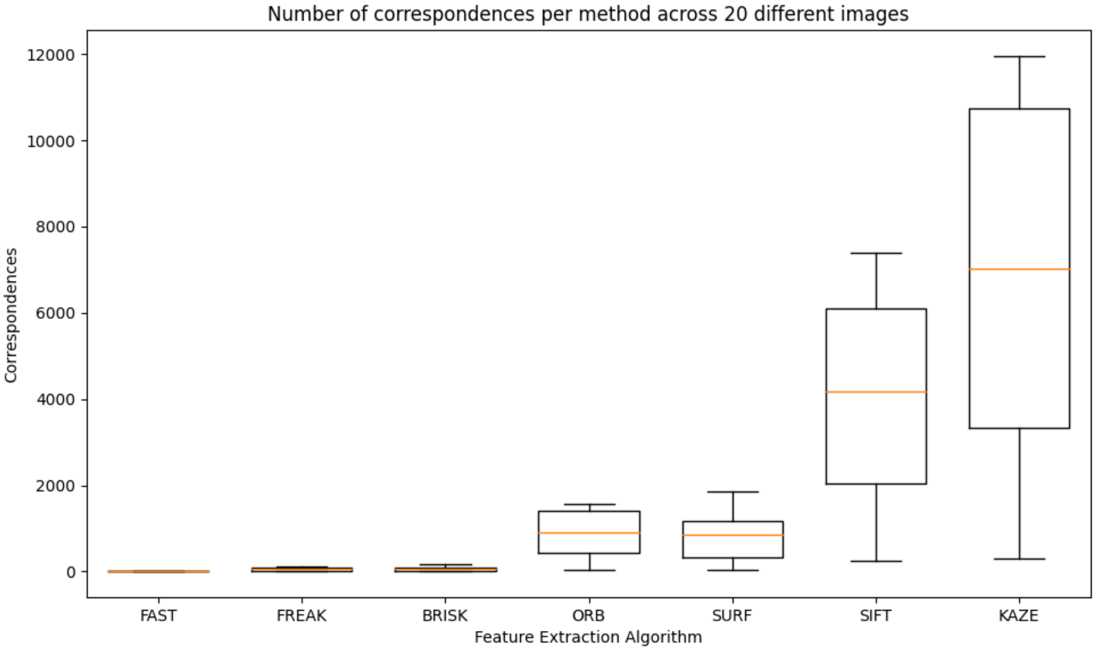
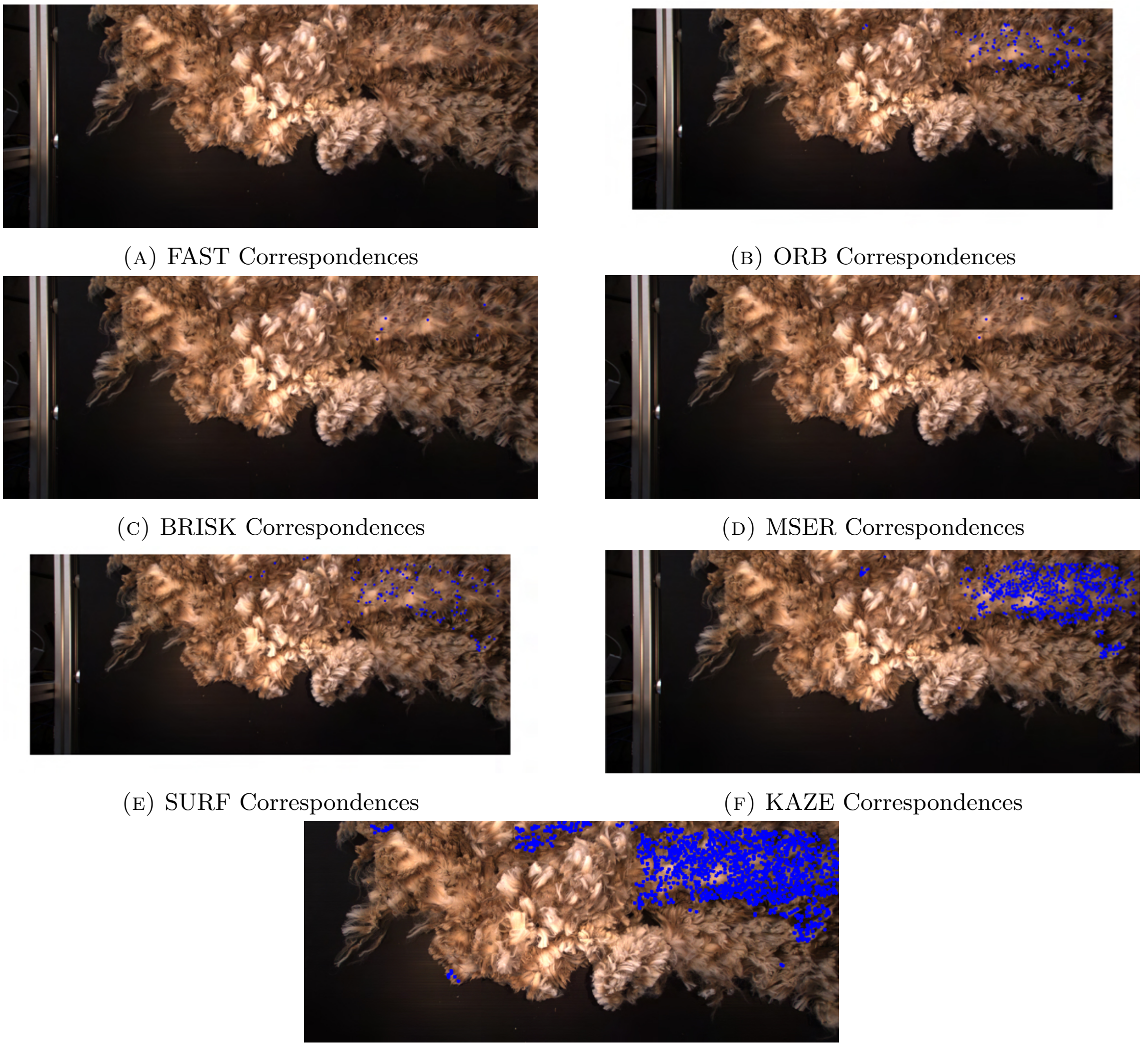


Figure 3.4: Number of correspondences per method across 20 different images

The data presented in Figure 3.4 demonstrates that the algorithms with the least effective performance are those lacking scalability and rotational invariance. Adversely, SIFT and KAZE stand out for their significantly higher number of correspondences relative to other methods. However, as highlighted in Figure 3.5g, both algorithms struggle to identify correspondences near the fleece's edge. Further comparison between these two methods was performed, where a different fleece is selected and only the deformed edge of the fleece can be seen. As observed in Figure 3.6, SIFT outperforms KAZE when higher amounts of deformation are present on the fleece.

3.6.2 Learning Based Feature Matching Algorithms

Building upon traditional feature-matching algorithms, learning-based algorithms pursue more sophisticated approaches which can result in a more accurate skirting line detection. One of the state-of-the-art methods in this domain is the Local Optical TRansformer



(A) FAST Correspondences

(B) ORB Correspondences

(C) BRISK Correspondences

(D) MSER Correspondences

(E) SURF Correspondences

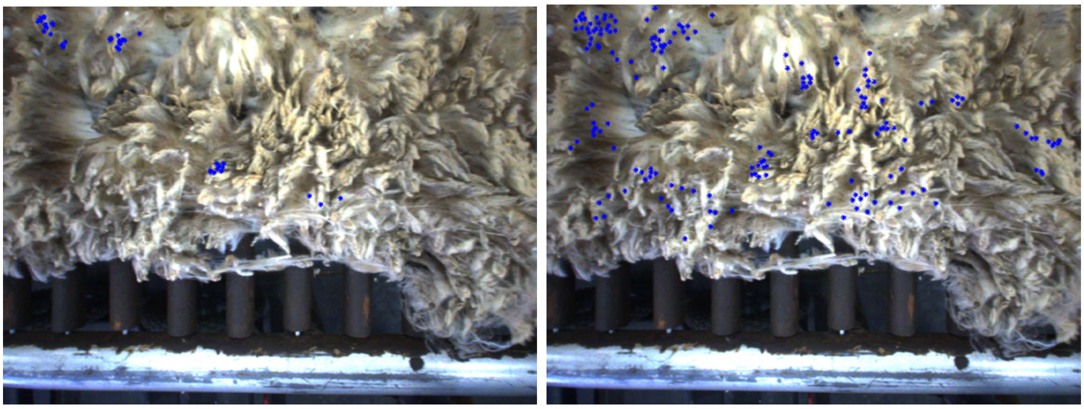
(F) KAZE Correspondences

(G) SIFT Correspondences

FIGURE 3.5: Location of matched features for each algorithm using data obtained from field testing. Blue markers represent the location of the correspondences in the after skirt image.

(LoFTR), which utilises a transformer architecture to find feature correspondences between two images.

An evaluation of these features can be seen in Figure 3.7, where the filtered correspondences between two sets of wool images are highlighted in blue. Similarly to traditional feature extraction methods, LoFTR struggles to identify correspondences near the edge where most of the deformation is present. Although LoFTR is capable of finding more correspondences in the non-deformed area, it exhibits similar performance to SIFT in terms of the number of correspondences near the edges, as illustrated in Figure 3.7a compared

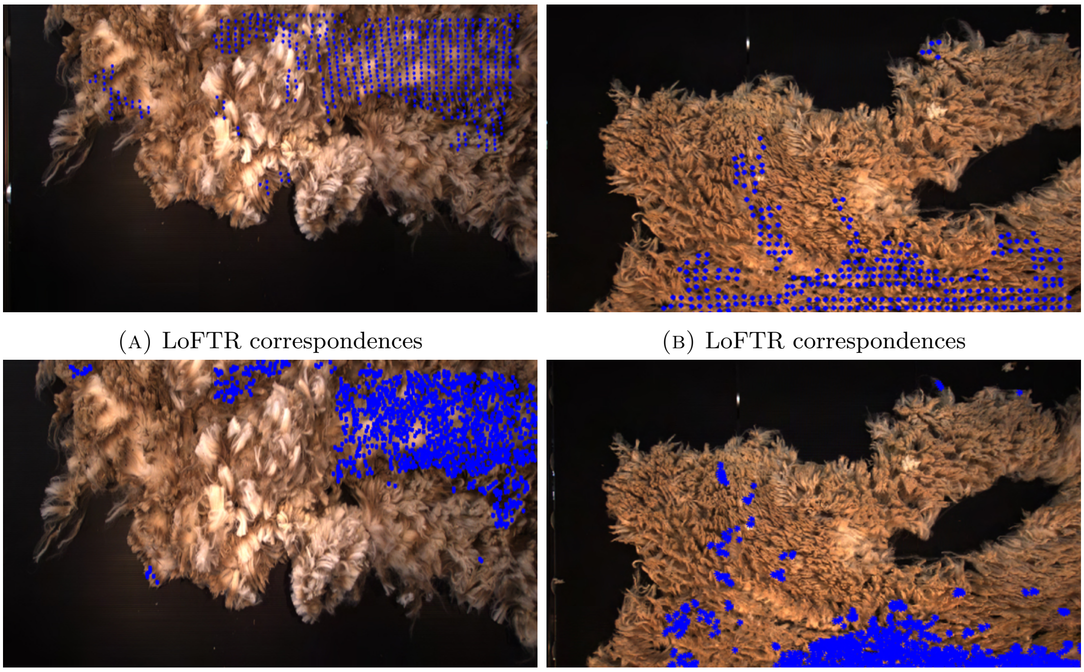


(A) KAZE Correspondences

(B) SIFT Correspondences

FIGURE 3.6: Comparison between feature correspondences obtained using SIFT and KAZE. The location of the correspondences is highlighted in blue. SIFT outperforms KAZE when extracting correspondences on very damaged areas near the edge of the fleece.

to Figure 3.7c, and in Figure 3.7b compared to Figure 3.7d. For this reason, SIFT is chosen as the preferred method for feature correspondence due to its scale and rotation invariance, in contrast to learning-based methods, which may be less reliable.



(A) LoFTR correspondences

(B) LoFTR correspondences

(C) SIFT correspondences

(D) SIFT correspondences

FIGURE 3.7: Location of correspondences after non-rigid filtering using transformer-based network compared to SIFT. Correspondences are coloured in blue

Learning-based feature correspondences require annotated data. This is not only time-consuming but also introduces potential errors, as exemplified by the issues highlighted in the training data used in [9] (Section 3.7.2). Furthermore, it is important to note that the performance of the learning-based correspondence model is directly tied to the quality of the correspondences used in its training (*i.e.*, training a method using the obtained SIFT correspondences would limit its performance to that of the SIFT method).

3.6.3 Non-Rigid Deformation Methods

Following the comparison between feature extraction algorithms, we perform a comparative analysis of the non-rigid deformation fields reviewed in Chapter 2 (Section 2.1.3.3). Two of the most popular deformation fields used in current approaches are ED, with one of the most notable implementations by [27] and spline fields discussed in Section 2.1.3.7. Both of these methods have shown their capability to handle large deformations while remaining user-friendly.

The initial approach evaluated is deformation graphs, more specifically ED by [27]. This is done by transforming the input of the non-rigid deformation as a set of scattered points with coordinates defined in three dimensions (the third dimension is stacked with a row of zeros). ED performs the non-rigid optimisation by solving the deformation on a graph. The control points or nodes of the graph, $G = \{g_1, \dots, g_v\}$, are obtained by heavily downsampling the transformed feature correspondences and the edges are generated by connecting each node with its closest neighbour.

Once the deformation graph is created, we optimise the local rotations $R_i \in \mathbb{R}^{3\times3}$ and local translations $t_i \in \mathbb{R}^3$ for each node in the deformation graph G. This optimisation is performed by minimising the energy function Equation (2.4) previously shown in Section 2.1.3.3. By replacing f^1 and f^2 into Equation (2.10) we account for the pairwise distance between feature nodes:

$$E_{con} = \sum_{l=1}^{m} \left\| F_l^1 - F_l^2 \right\|_2^2 . \tag{3.3}$$

The regularisation term E_{reg} remains constant as per Equation (2.12):

$$E_{reg} = \sum_{i}^{\nu} \sum_{j=1}^{\mu} ||R_j(g_i - g_j) + g_j + t_j - (g_i + t_i)||_2^2 \tag{3.4}$$

Similarly to Jiawen et al. [91], we define the errors in rotation E_{rot} as Equation (3.5). This approach reduces stretching by favoring solutions that promote isometric properties.

$$E_{rot} = \sum_{j=1}^{\nu} ||{R_j}^T R_j - I||_F{}^2. \tag{3.5}$$

The energy function described in Equation (2.4) is then minimised. Once the new position of the deformation graph nodes are known, the points of $\mathcal{P}_1$ are updated using:

$$p_i^* = \sum_{j=1}^{\mu} w_j(p_i)[R_j(p_i - g_j) + g_j + t_j], \tag{3.6}$$

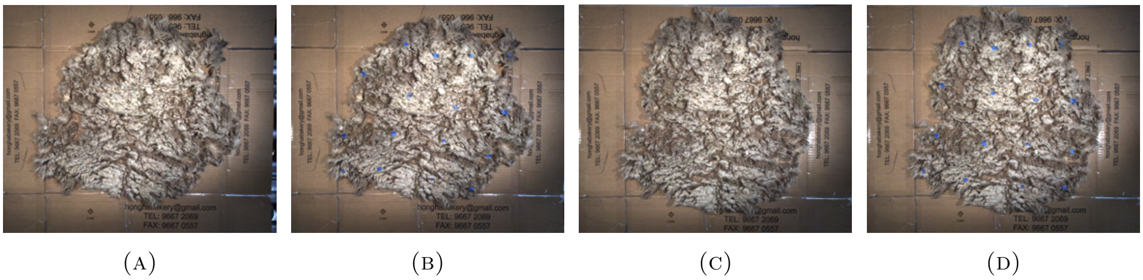
with the neighbour's nodes g_j from $\mathcal{P}_i$ found using a search with a KD-tree. The weight for each vertex is defined as

$$w_j(p_i) = (1 - ||p_i - g_j||/d_{max}), \tag{3.7}$$

where d_{max} is the maximum distance of the vertex to the $\mu + 1$ nearest node from G.

The second evaluated deformation field is spline deformation, which has gained prominence as one of the most favoured approaches for non-rigid deformations in medical images. The approach used for evaluation is derived from [92], including the smoothness penalty as specified in [92]. Additionally, it utilises the Jacobian diffeomorphic function as a further regularization term for smoothness in the deformation.

To quantitatively evaluate these deformation fields, annotated data was collected where markers were placed on the wool. By using markers as ground truth, we can assess the location of the specific area after deformation has been applied to the fleece. The data utilised for testing is illustrated in Figure 3.8.



(a) (b) (c) (d)

Figure 3.8: Data used for testing of algorithms. Blue markers are used as ground truth to analyse the deformation. The sample includes a) Source Image, b) Source Image with markers attached, c)Target image and d) Target image with markers attached

As demonstrated by Figure 3.9, the error difference between Embedded Deformation and spline deformation is significant. Spline fields are not able to replicate the intricate deformations that occur during skirting with single fibers being independently stretched.

Further experiments were also conducted to test for computational efficiency of each of the methods. As demonstrated in Table 3.1, spline deformation proved to be the most efficient compared to Embedded Deformation however the error in deformation does not make it a viable deformation for use in the proposed pipeline.

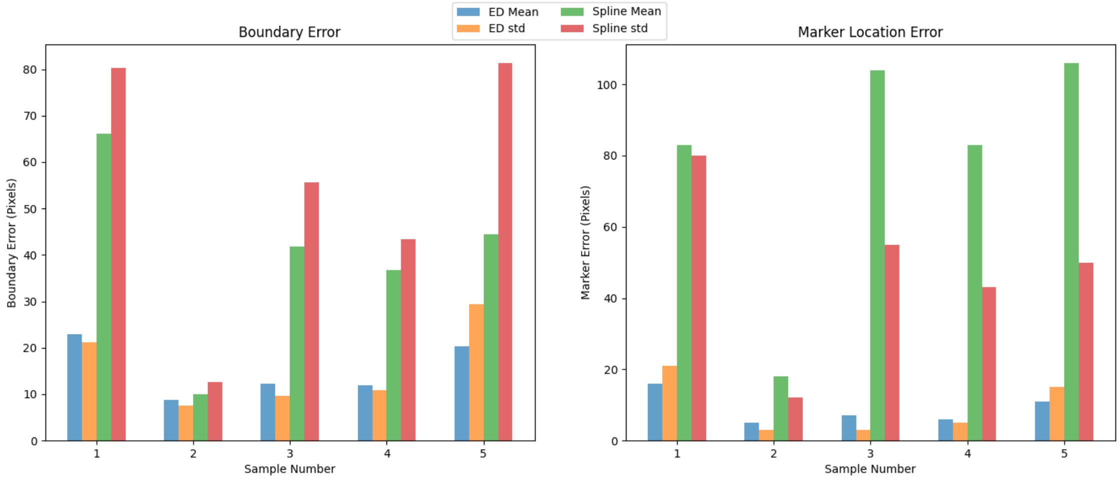


Figure 3.9: Marker and Boundary error for each of the non-rigid deformation methods

3.7 Results

The proposed pipeline with the use of SIFT and ED is tested in two different scenarios. As demonstrated in the ablation study, these two methods outperform other methods when

Sample Number	**ED**	**Spline**
Sample 1	67.55 s	0.81 s
Sample 2	4.54 s	0.86 s
Sample 3	45.89 s	0.80 s
Sample 4	47.12 s	0.88 s
Sample 5	56.61 s	0.81 s

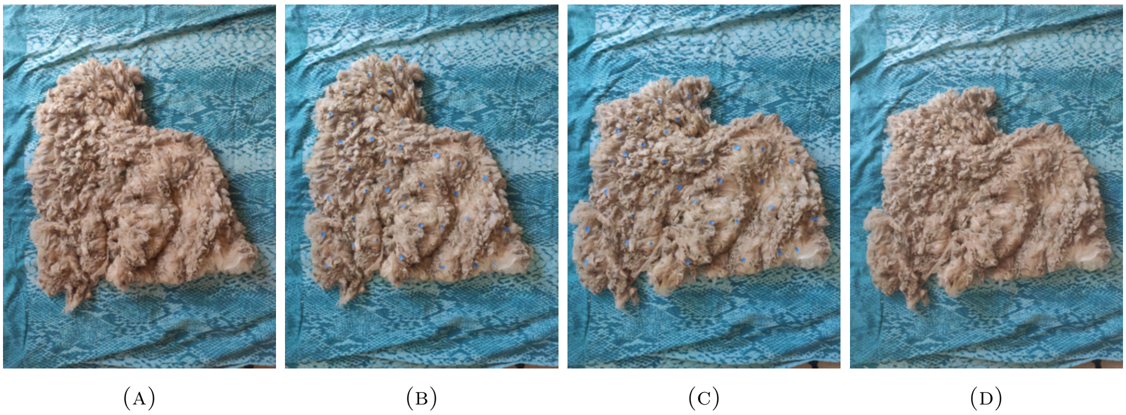
Table 3.1: Processing speed of each of the methods (seconds). Experiments were conducted on an 11th Gen Intel Core i7-1185G78 with 16GB RAM

used in wool images. The initial experiment is performed in a controlled environment, designed to quantitatively measure the performance of each approach by utilising ground truth data. The second experiment is performed with data collected from the field and compared to annotations done by experts.

3.7.1 Controlled Experiment

Obtaining ground truth data for analysing deformation on wool during the skirting process poses a significant challenge due to the repetitive nature of the wool's pattern, making it difficult to accurately determine the boundary between clean and contaminated parts of the fleece from images alone. To address these challenges, markers are attached to the wool deformation process. These markers work as annotations to know with certainty where the parts of the wool move during deformation. Differently from the data obtained for evaluation of the deformation fields, the wool is placed on top of a blue background to accurately determine the boundary of the fleece.

The data collected for the controlled experiment is presented in Figure 3.10. This dataset comprises of two initial images before deformation: one with annotations Figure 3.10b and another without the annotations Figure 3.10a. The annotated image is subsequently subjected to stretching and compression, resulting in the acquisition of two new images: one with annotations Figure 3.10c and another with annotations removed Figure 3.10d. The annotations have been removed to prevent any corruption of SIFT during the feature extraction. These annotations, along with the wool's boundary, are utilised to quantify the outcome of the non-rigid deformation



(A)

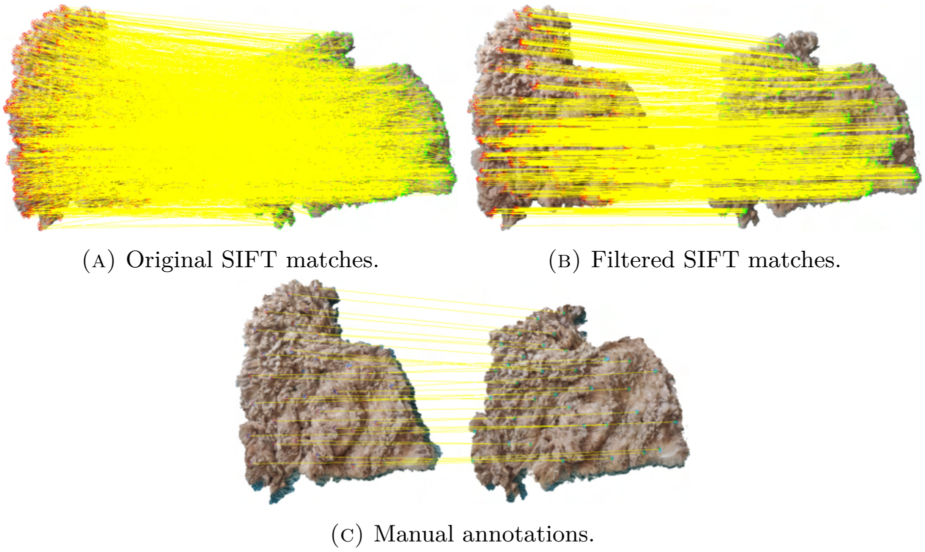
(B)

(C)

(D)

Figure 3.10: Data used for the controlled experiment. The sample (a) is annotated (b), stretched and compressed (c), and the annotations are then removed to enable evaluating SIFT feature correspondence (d).

The SIFT features matches are shown in Figure 3.11a. As shown, many outliers are present in the feature matching and need to be removed. The correspondences are then filtered using the proposed filtering method described in Section 3.4, the results can be seen in Figure 3.11b. The manual annotations from Figure 3.10b and Figure 3.10c are then matched as shown in Figure 3.11c for reference.



(A) Original SIFT matches.

(B) Filtered SIFT matches.

(C) Manual annotations.

Figure 3.11: SIFT features matching before (a) and after (b) the filtering process described in Section 3.4. In (c), we show the correspondences from the annotations.

Given the filtered feature matching, we then perform the non-rigid deformation using ED as discussed in Section 3.5. The updated contours of the deformed image are displayed in Figure 3.12 in green alongside the original contour (in red) and the targeted image (in

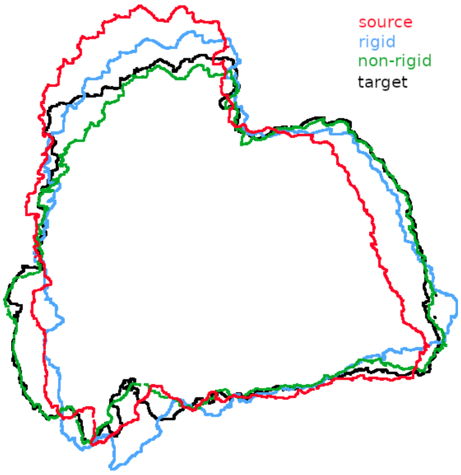


Figure 3.12: Contours using rigid and non-rigid deformation used for evaluation. The rigid deformation resulting from SIFT+RANSAC filter is shown in blue and the non-rigid deformation produced using SIFT+ARAP filter is shown in green. The original contour (red) and the target (black) are shown for reference.

black). We compare this approach to a standard method that includes SIFT matching, RANSAC filtering and rigid deformation. The contour of this deformation is displayed in blue in Figure 3.12.

A quantitative evaluation is further performed by analysing the distance between the deformed contours and the deformed manual annotations. The annotations (which are extracted manually using tags on the wool) provide information regarding the accuracy of the deformation across the whole surface. It provides us with an accurate quantification of the deformation error. The mean error and standard deviation of these annotations are reported in Table 3.2.

As a complementary quantitative evaluation, we measure the distance between the boundaries of the deformation, which closely relates to detecting the skirting line. The distribution of the distance between the boundaries is defined by finding the nearest point of the targeted boundary, in Euclidean space, for each point of the deformed boundary. These distributions are displayed in the violin plots of Figure 3.13 and the mean and standard deviation are given in Table 3.2.

The ablation study in Table 3.2, which considers rigid / non-rigid filtering and rigid / non-rigid deformation, shows that the problem needs to be tackled non-rigidly for both the features matching filtering and the deformation.

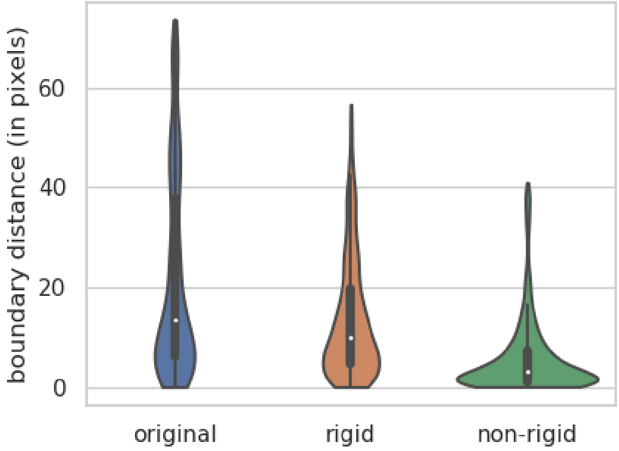


Figure 3.13: Distribution of the distance between the contours (shown in Figure 3.12). Our proposed method generates contours that are significantly closer to the target.

		Boundary		Annotations	
Filtering Method	Deformation	Mean	Std	Mean	Std
none	none	21.83	20.18	40.17	18.22
rigid*	rigid	21.16	17.98	36.63	19.39
rigid*	non-rigid	18.86	16.57	34.29	17.85
non-rigid	rigid	19.57	19.73	23.90	18.88
non-rigid	non-rigid	5.75	7.44	14.04	6.66

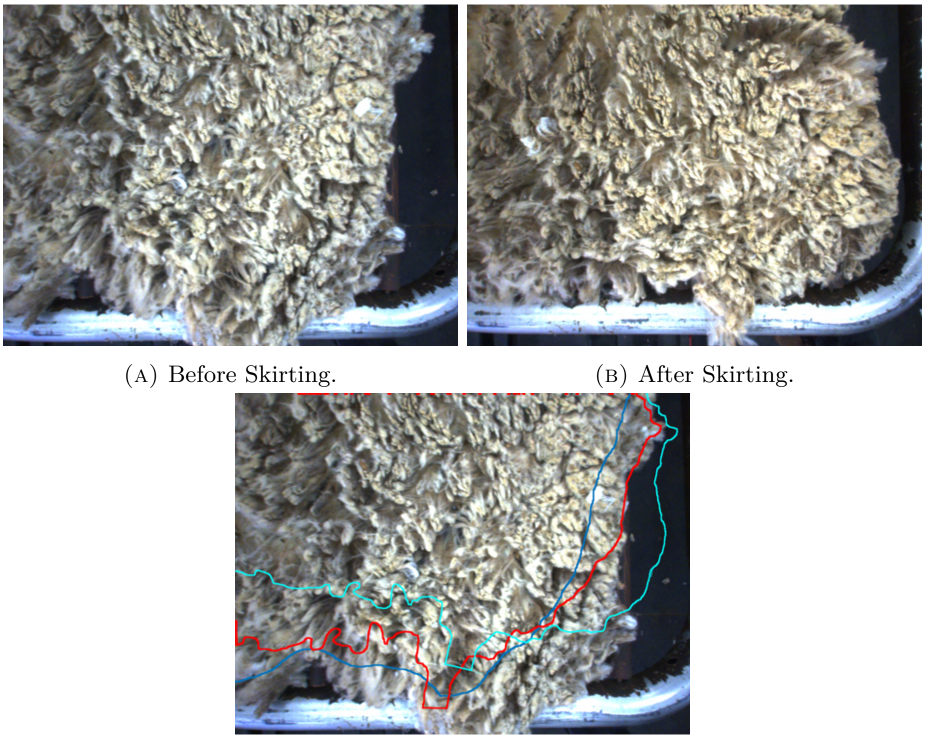
Table 3.2: Quantitative evaluation and ablation study: starting from the SIFT features matching, the outliers are filtered using RANSAC (rigid) or the proposed ARAP filter (non-rigid). The image is then deformed rigidly or non-rigidly. The measurement without deformation is provided for reference. Distances are in pixels. * these experiments were run 50 times given the non deterministic property of RANSAC.

3.7.2 Field Experiment

Data collected from the field experiment are shown in Figures 3.14a and 3.14b. With the use of this data we are able to further evaluate the proposed approach against rigid methods and annotations done by expert wool handlers ([9]), as per Figure 3.14.

The deformation of the wool after skirting has been manually assessed as shown in Figure 3.15, demonstrating convincing results regarding the quality of the deformation (specific patterns in the wool are visible in the same location for both the before and after skirt images).

However, as shown in Figure 3.14, there is a significant difference between the manual annotation (which corresponds to a regular trimming of a few centimetres) and the actual skirting by the wool handler. The discrepancy further accentuates the difficulty for a human to assess the quality of the wool in an image without manipulating it or having a close inspection.



(A) Before Skirting.

(B) After Skirting.

(C) Contour comparison between methods.

FIGURE 3.14: In (c) we demonstrate the accuracy of each method with the use of contours: Wool handlers annotation (blue), proposed method (red) and rigid deformation (cyan).

While the results indicate superior performance compared to other proposed methods, there is still a minor issue near the edge in Figure 3.14, where the boundary extends beyond the edge of the before skirt wool. These minor errors can have a significant impact on the automation of the skirting line, potentially leading to financial losses for wool producers. As a result, further ablation studies have been conducted to gain a better understanding of the approach's limitations.

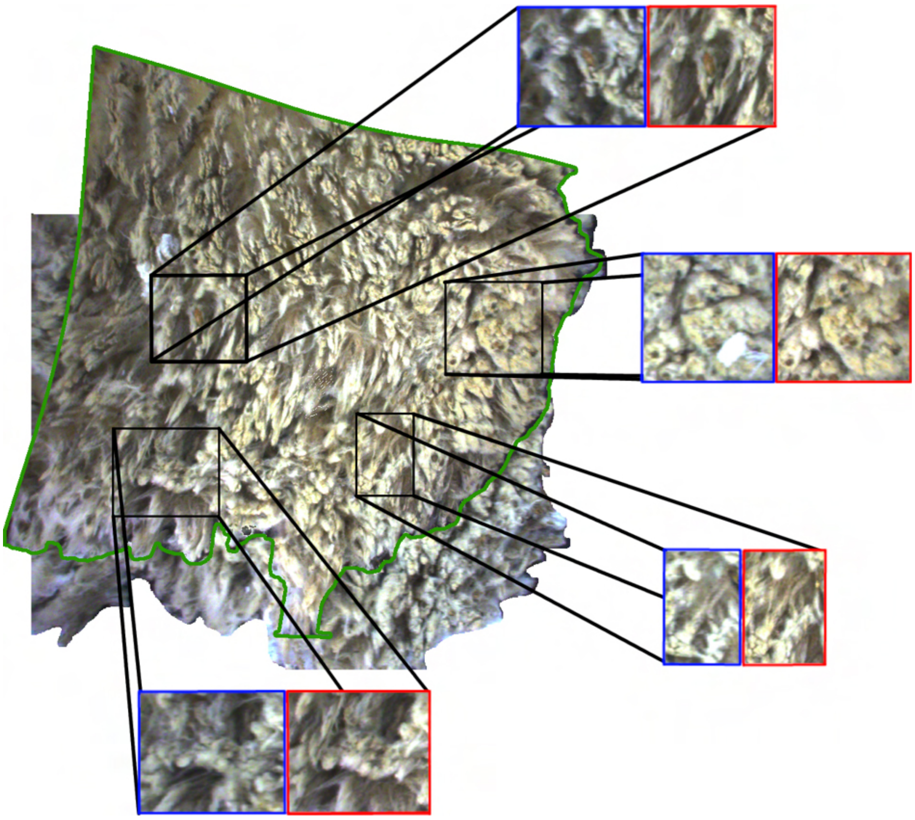


FIGURE 3.15: Qualitative evaluation of the morphing on data from the field. The deformed skirted image (green border) is superposed to the original wool image. In blue, we show the wool image prior to skirting and in red the sample after skirting. The areas selected are strictly superposed.

3.8 Conclusion

In this chapter, we introduced a method for automatically detecting the skirting line in freshly shorn fleece through the use of image registration. The method incorporates a filtering strategy for feature correspondences, designed to take into account the non-rigid deformations that happen to the fleece during the skirting process. It then proceeds to align the before and after skirt image using the filtered features and embedded deformation to extract the skirting line.

We tested this approach on both the controlled experiments and practical applications. These tests prove the efficiency of the method in estimating the deformations that happen to the wool during the skirting process while highlighting some shortcomings of feature-matching algorithms when high amounts of deformation are present in the data.

While the controlled experiments demonstrate that the proposed pipeline is capable of handling the deformation in the fleece when repositioned, it fails to properly align the edges when the fleece has been skirted.

This is further highlighted during the ablation study of feature matching algorithms. The results indicate that the major limitation in the proposed approach is present when extracting feature correspondences. While SIFT and pretrained methods are capable of identifying correspondences in non-damaged areas, they fell short in identifying correspondences near the edge of the fleece. This phenomenon can be attributed to the high presence of outliers in that region. When an inlier correspondence is surrounded by outliers, it can be mistakenly categorized as an outlier itself. Therefore, it becomes necessary to adopt a different strategy to include these previously identified outliers into the set of inlier correspondences.

In the next chapter, we address this feature correspondence problem by integrating a learning-based methodology. This approach focuses on identifying deformations in less-affected areas and subsequently extends these deformations to the fleece edges by integrating dense correspondences based on their spatial consistency to the inlier set.

Chapter 4

Skirting Line estimation using Sparse to Dense approach

4.1 Introduction

The work presented in the previous chapter demonstrates the feasibility of automating the skirting line detection by using image registration between two RGB images. By aligning the before and after skirt images, it is possible to identify the parts that were removed along the edge of the fleece, resulting in the detection of the skirting line.

However, traditional feature matching algorithms such SIFT, SURF and KAZE are not capable of finding correspondences near the edge of the fleece when the deformation is severe, leading to inaccuracies when detecting the skirting line. Furthermore, state-of-the-art learning-based feature matching methods (LoFTR) also suffer from these limitations. The unique characteristics of wool, including its tendency to stretch and tear during the skirting process, require a more sophisticated approach to detect deformations occurring near the edge.

Most existing approaches allow for non-rigid registration between images, though they fail in areas with repeated features. As a consequence, existing methods only perform well when wool is gently handled in lab settings, but are unsatisfactory when using field data.

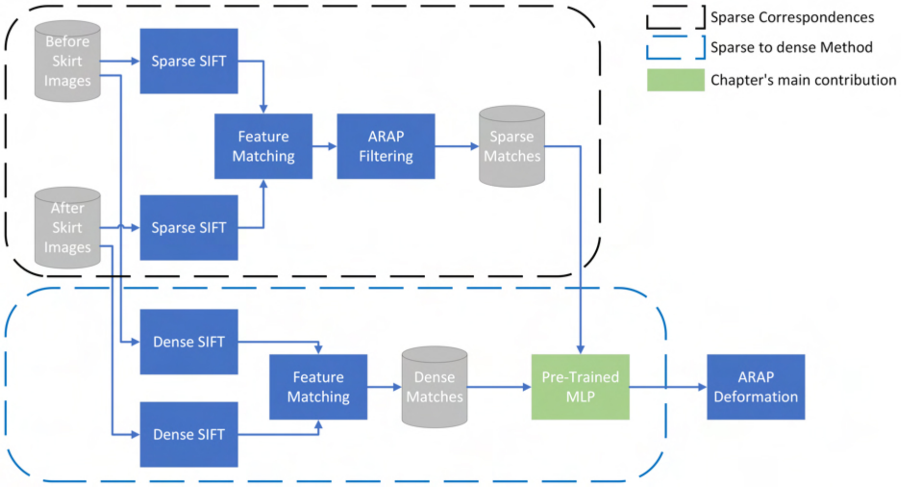


FIGURE 4.1: Flowchart of the proposed method. A sparse set of inlier correspondence is first obtained using the approach mentioned in Chapter 3. Then, a sparse-to-dense approach integrates dense feature correspondences along the edge of the wool. This is achieved by incorporating a dense set of correspondences into an initial sparse set of inliers by using a learning-based classifier

For this reason, this chapter presents an approach for non-rigid alignment of wool fleece, centred around a feature descriptor that can account for local pose changes with respect to its neighbours. Utilising this spatial data along with a learning-based filtering technique, the approach aims to expand the correspondences from low-deformation regions to severely affected areas, normally around the wool's edge.

4.1.1 Methodology Overview

An overview of the proposed approach can be seen in Figure 4.1, where we extract both a sparse and dense set of correspondences to perform image registration. This chapter incorporates an MLP to filter the dense correspondences based on the sparse set spatial information. With this approach, we aim to address the limitations of feature detection algorithms, enhancing the sparse set of correspondences and expanding them to the edge of the fleece while enforcing continuation in deformation of the wool.

4.2 Sparse Feature Correspondences

4.2.1 Feature Extraction

Similarly to the approach in the previous chapter, the proposed method uses SIFT descriptors Lowe [93] to extract the features from the two input images. As highlighted in Section 3.6, SIFT descriptors have been shown to outperform other non-learning based feature extracting methods by finding more correspondences around the severely deformed areas in the fleece, typically located around the edges. Additionally, due to the absence of training data or the potential for human error in annotated data Patten et al. [9], SIFT has demonstrated performance comparable to learning-based methods such as LoFTR around the edges of the fleece. For these reasons, SIFT descriptors are the selected for the proposed methodology over other traditional or learning based features. The robustness of SIFT to rotations, translations, and changes of viewpoint that can occur during deformation renders it highly suitable for the proposed approach. Furthermore, to ensure continuity between the sparse and dense sets, the descriptor information will subsequently serve as a crucial input for the filtering method, facilitating the removal of outliers in the dense set.

The extracted feature descriptors of the before and after skirt images are then matched using the Lowe's ratio criterion [93] resulting in a set of unfiltered correspondence $\left\{F^1, F^2\right\}$, where F^1 and F^2 correspond to the features of the before skirt image and after skirt image respectively.

4.2.2 Non-rigid Filtering

Outlier rejection is handled by using previously proposed method in Section 3.4. As previously demonstrated, this method is capable of accurately filtering the correspondences for non-rigid deformations without being computationally expensive. Once all the outliers are pruned, a set of sparse feature correspondences is obtained. This is referred to as $S = \left\{s^1, s^2\right\}$ and is used to initialise the inlier set of correspondences.

4.3 Dense Feature Correspondences

Denser features are extracted using *dense* SIFT Bosch et al. [94]. This variant of the popular algorithm SIFT operates on a dense grid of points throughout the image rather than on a sparse set of keypoints. To improve computational efficiency, our method extracts potential matches for each of the descriptors located outside the boundary of the sparse data. This is achieved by applying the Sobel algorithm Sobel et al. [95], which enables the construction of a mask defined by the boundary of the sparse correspondences. Features located within the mask boundary are then deleted, consequently significantly reducing the number of feature matches to be computed.

All the putative matches obtained from the dense descriptors are stored in two different sets $D = \left\{D^1, D^2\right\}$, containing the *dense* SIFT features for the source and target images. The i^{th} dense feature correspondence stores both the information of the feature coordinates, referred to as $d_i^{1|2}$, and the SIFT feature descriptor, referred to as $f_i^{1|2}$.

4.4 Dense Inlier/Outlier Classification

Once all the correspondences are obtained $\{S, D\}$, an MLP is used to classify the dense correspondences into inliers and outliers. MLP are predominantly used to learn complex patterns and relationships in data. Our approach learns the geometric consistency from the inliers and filters the dense correspondences using this geometric consistency.

Given the i^{th} correspondence $\{d_i^1, d_i^2\}$ such that $d_i^{1,2} \in D^{1,2}$, the MLP takes as input the concatenated information of the feature descriptors difference and the geometric consistency between d_i^1 and d_i^2.

The feature descriptor difference consists of the L_2-norm between the SIFT feature descriptors of the before and after skirt image. More formally, the SIFT difference is defined as:

$$e^{SIFT} = ||f_i^1 - f_i^2||_2. \tag{4.1}$$

The geometric consistency information is obtained by analysing the consistency of the dense feature neighbours within the inlier set S. Similarly to the sparse feature filtering, a graph is built by connecting d_i^1 to its k neighbours in D^1. In contrast to the sparse filtering, neighbours are only selected from the inlier set, making them more reliable. The graph is then *copied* onto the target dense correspondence d_i^2, by using the previously calculated correspondences from the sparse set $S^{1,2}$. A sample of such a graph is illustrated in Figure 4.6. The optimal rotation that would align d_i^2's neighbours to d_i^1's neighbours is then obtained using SVD and applied to align the neighbours of the source and the target. Finally, the geometric difference between each neighbour is computed as

$$\mathbf{e^{geometric}} = \{e_1^{geometric}, \ldots, e_k^{geometric}\}, \tag{4.2}$$

$$e_j^{geometric} = ||(\mathcal{N}(d_i^1)_j - d_i^1) - (\mathcal{N}(d_i^2)_j - d_i^2)||_2 \tag{4.3}$$

where $\mathcal{N}(d_i^{1|2})_j$ represents the coordinates of the j^{th} neighbour of $d_i^{1|2}$. In other words, Equation (4.3) consists of the residual of the distance between the source and target graph's neighbours while accounting for translation and rotation. The geometric information is ordered from closest to furthest with respect to the source dense feature d_i^1.

The feature information e^{SIFT} and the geometric information $\mathbf{e^{geometric}}$ are then concatenated into a single vector and used as inputs for the MLP. These inputs are summarised in Figure 4.2.

4.5 Iterative Update of the Sparse Feature Correspondences

The sparse set of feature correspondences between the before and after skirt images is updated using an iterative process that adds correspondences from the dense set. To keep the geometric information described in Equation (4.3) meaningful (i.e., allowing global non-rigid deformation while minimising local deformation), the order in which the dense correspondences are processed is obtained by iteratively finding the closest correspondence from the inlier set and growing towards areas with higher deformation. After computing

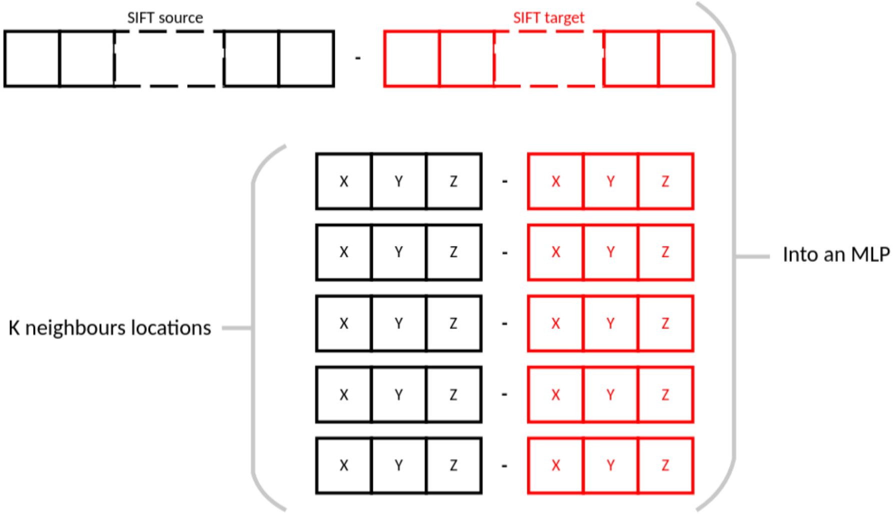


Figure 4.2: Inputs to the MLP classifier. The MLP takes as input the difference between the feature correspondences. These feature correspondences are composed of the Scaled Invariant Feature Transform (SIFT) descriptor vector and the location of the k closest neighbours in both the source and target image.

the MLP prediction for the closest dense descriptor to the sparse set, the descriptor is added to the sparse set if the Multi-layer Perceptron (MLP) classifies it as an inlier. The descriptor is then removed from the dense set. The process is repeated iteratively until all dense correspondences have been classified. This iterative process starts from the areas less affected by the deformation and gradually adds more features in the areas with more deformation.

4.6 Non-rigid Deformation

In this chapter, we propose using As Rigid As Possible (ARAP) as an alternative to Embedded Deformation (ED) to enhance computational efficiency in computing non-rigid deformations. In contrast to the approach presented by Sumner et al. [27], Sorkine and Alexa [51] optimises for rigidity and rotation using Equation (2.15), with the primary objective of preserving local rigidity, encouraging small parts of the figure to change as rigidly as possible, thereby improving computational efficiency in the process.

Table 4.1 highlights the improvement of processing speed upon ED while having similar error as demonstrated in Figure 4.3. These experimenters we performed using the data collected for the ablation study in Figure 3.8, Section 3.6.3.

Sample Number	**ED**	**ARAP**
Sample 1	67.55 s	16.4 s
Sample 2	4.54 s	1.35 s
Sample 3	45.89 s	13.58 s
Sample 4	47.12 s	15.03 s
Sample 5	56.61 s	13 s

Table 4.1: Processing speed comparison between ARAP and ED in seconds

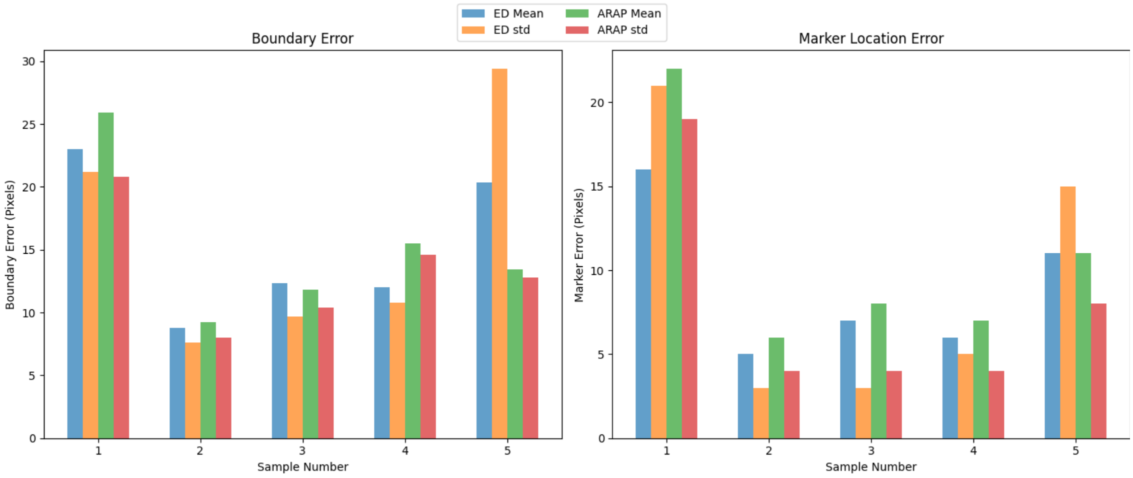


Figure 4.3: Error in deformation between ARAP and ED using features obtained from approach in Chapter 3 and data collected in Figure 3.8

Given that ARAP requires a mesh as input for the deformation, a mesh is generated from the image by storing this image as a set of points, which are then transformed into a mesh structure through Poisson surface reconstruction as proposed by Kazhdan et al. [96]. The requirement for this mesh arises from the connectivity between nodes required from ARAP deformation.

This method constructs the mesh by formulating it as a Poisson equation, which balances the smoothness of the surface and the conformity to the input points. The result is a surface mesh that accurately captures the geometric features of the original object.

The sparse correspondences obtained from the MLP, output as (S^1 and S^2), are then used as the control points for ARAP to deform the image. ARAP deforms the image by minimising the energy function in Equation (2.15) shown in Section 2.1.4. The energy function can be rewritten for the sparse correspondences as:

$$E_{ARAP} = \sum_{i=1}^{N} \sum_{j \in N_i} w_{ij} ||(S_j^1 - S_i^2) - R_i(\hat{S}_j^1 - \hat{S}_i^2)||^2 \tag{4.4}$$

where N_i is the set of indices of neighbouring vertices of vertex i, S_i^1 and S_j^2 are the given set of deformed vertex positions, S_i^1 and S_j^2 are the positions of the control points corresponding to vertices i and j, and R_i is a rotation matrix that minimises the difference between the positions of neighbouring vertices in the deformed mesh and the positions of the corresponding control points. The term $(S_j^1 - S_i^2)$ represents the relative displacement between vertices i and j in the deformed mesh. Similarly to Section 3.4, w_i is set to 1 to prevent any bias regarding the area of the surfaces in the mesh.

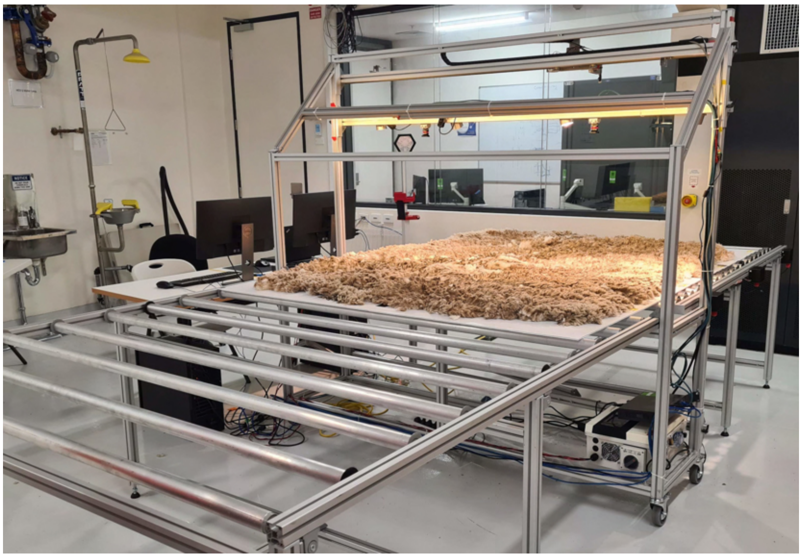


FIGURE 4.4: Wool contaminant detection rig. RGB camera on the top rail observe the wool fleece that is thrown on the actuated rollers by a handler. This rig is used to collect the dataset presented in this work.

4.7 Experiments

Markers are placed on the wool during the controlled experiment to precisely analyse the location of specific parts after the deformation process, similar to the data collection in previous chapter. However, in contrast to previous experiments, the wool is skirted rather than just repositioned or stretched, providing a closer resemblance to field conditions.

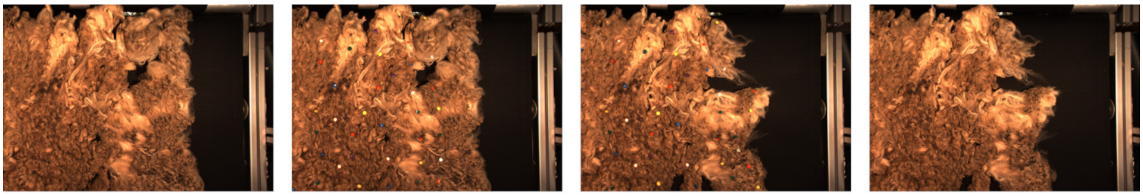


Figure 4.5: Sample from the dataset showing from left to right: source image, source image with markers, target image with markers and target image.

Additionally, a denser number of markers are attached to the wool to trace most moving sections of the fleece.

The data capture process is explained as follows, with an image captured after each step:

- fleece is placed under camera
- markers are placed on fleece
- fleece is skirted with the markers on
- markers are removed

The first and last images form the input pairs for the evaluation as they contain no markers. The second and third images are used as ground truth, containing the markers ti be used as reference points for the evaluation of the algorithm performance. The procedure is illustrated in Figure 4.5.

The ”ground truth” correspondences between the markers in each pair of images is determined using the unique marker colour. A colour filter, median clustering enables point wise nearest neighbour correspondence to match all markers individually. This dataset consists of 24 pairs of images (one before and one after a sample of wool is skirted). The data is captured with the contaminant detection rig shown in Figure 4.4, where wool is placed on the roller table and observed by an overhead camera (Balluff BVS0037 RGB camera with a resolution of 2464×1026 pixels).

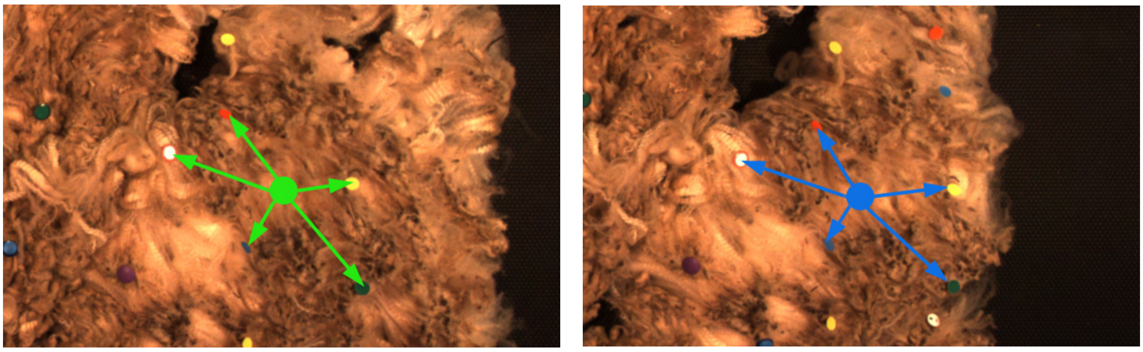


FIGURE 4.6: Features created extracting a SIFT feature and its closest k neighbours. The green descriptor represents an inlier on the before-skirt image, which shows the graph toward the closest markers. The blue descriptor is the matching feature of the SIFT descriptor in the after-skirt image, connected to the same colour markers as the feature in the before-skirt image.

4.7.1 MLP Training

4.7.1.1 Inlier Generation

Similarly to the construction of the dense features, the inlier generation constructs two different graphs for the sparse descriptors in the before skirt image (I_s) and the after skirt image (I_t). A new descriptor is created for each sparse descriptor in both graphs by extracting the k closest colour markers in the before skirt image and their locations in the after skirt image. An example of this descriptor can be seen in Figure 4.6. SVD is used to deform the descriptor in the source image, which is composed of the sparse descriptor and its neighbours, to match the sparse descriptor in the target image and its respective neighbours.

4.7.1.2 Outlier Generation

For each sparse descriptor, we displace the location of the source point (S_n^1) to a new arbitrary location. We extract the k closest colour markers from the original source point and assign them to the new source point. This results in a descriptor with different lengths to the closest markers and a different distribution of marker locations. We then apply the same deformation process used for the inlier data, deforming the descriptor and its neighbours to match the target descriptor. In this case, the target descriptor remains unchanged, with the same k closest neighbours being the markers extracted for the original source point.

4.7.1.3 Noisy Inlier Generation

To increase the robustness of the MLP, we introduce noise to the inlier data by randomly modifying point location of some of the source points. Specifically, we randomly select a subset of the inlier points and change one of their neighbour markers in the source image to a different arbitrary marker. We then compute the optimal rotation between the modified source descriptor and the original target descriptor using SVD. Despite the added noise, the resulting deformed descriptor still has $k - 1$ neighbours with a small norm distance, while one has a significantly larger distance. This suggests that the point remains an inlier since the majority of its neighbours are correctly assigned.

4.7.1.4 Implementation Details

The MLP is implemented in PyTorch. It is trained with the cross-entropy loss function and a learning rate of 0.0001. The training data consists of $302,400$ features and the testing data consists of $137,500$ features. These features are equally distributed between inliers and outliers, with a randomised number of noisy inliers interspersed among the inlier data. The MLP is trained for 200 epochs with a batch size of 250.

4.7.2 Baseline Methods

For comparison, we assess our approach against optical flow methods rather than feature matching algorithms. This choice is informed by the findings presented in the previous chapter, which demonstrated that feature matching methods struggle to accurately determine changes wherever their is severe stretching of the fleece.

The first approach we evaluate against is Lucas-Kanade optical flow Lucas and Kanade [61]. Additionally, the approach is compared against Jiang et al. [63] approach (SCV) due to its efficiency in estimating the optical flow field with a limited number of correspondences.

This property is relevant to our problem, as we can only obtain correspondences in areas with low deformation. To apply SCV, the data from a large number of images of fleece collected at a wool shed are used. This data comprises a diverse collection of 42 wool fleeces, each of which contains around 20 image pairs. For each of the image pairs, sparse descriptors are extracted and dense correspondences for all pixels between the image pairs are computed from the output of ED following the previous chapter approach. The weights are initialised from a model trained on the Sintel dataset Butler et al. [97], then trained for a further 50000 iterations with a learning rate of 0.00025 and batch size of 2. All other parameters are the same parameters as in Jiang et al. [63]. Finally, the method is also compared to previous skirting line estimation methods proposed in Chapter 3.

4.8 Results

The first section showcasing the results of the training of the MLP. Thereafter, the efficiency of the approach is tested in two different scenarios. The first one is a controlled experiment where newly collected data is utilised to test the algorithm's performance. The second scenario is on the data collected from the field, to evaluate the performance of the algorithms in real world scenarios.

4.8.1 MLP

The result of the proposed method demonstrates that the MLP effectively distinguishes between an inlier or outlier feature based on the new features constructed. The performance of the MLP is evaluated using an AUC-ROC curve, which plots the true positive rate against the false positive rate at various threshold values. From the curve in Figure 4.7, it is observed that the MLP is capable of correctly identifying true positives at a threshold of 0.5. However, in the subsequent experiments, a higher threshold of 0.75 is used to extract only the features with the highest probabilities of being an inlier. This is done to ensure that there is a continuous deformation on the fleece, as these features reflect the highest probability of continuous propagation on the deformation from the sparse data.

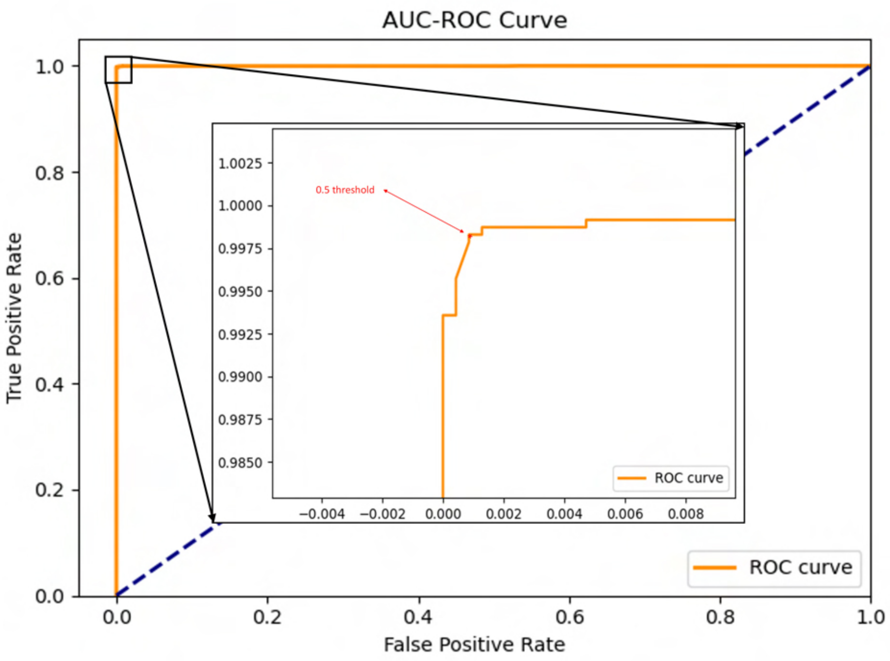


FIGURE 4.7: AUC-ROC curve for MLP classification performance. The curve demonstrates the model's ability to distinguish between true inliers and outliers based on the probability of each class. At a probability threshold of 0.5, the MLP is capable of correctly identifying the distribution between inliers and outliers.

4.8.2 Controlled Experiment

For the controlled experiment, a quantitative evaluation on the marker dataset is performed by deforming I_s to match I_t based on the correspondences obtained from each of the methods. The pixel location of each of the markers is extracted and matched to the image without the markers. After deforming I_s, the translation is computed and the error between the pixel location and markers on I_t is extracted. To ensure a fair evaluation of the algorithm's performance and avoid bias towards fleeces with low deformation, the error is only computed for the markers that have been displaced beyond a certain threshold. This ensures that the translation error is only calculated using the areas with large deformation.

As seen in Figure 4.8, our approach yields an improvement on learning-based optic flow methods such as Jiang et al. [63] for calculating deformation on fleece.

To further evaluate the location of the markers error, a qualitative comparison between the methods is carried out to show the areas where each method fails. This comparison

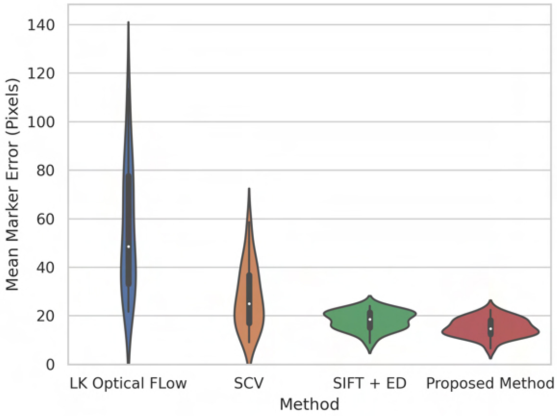
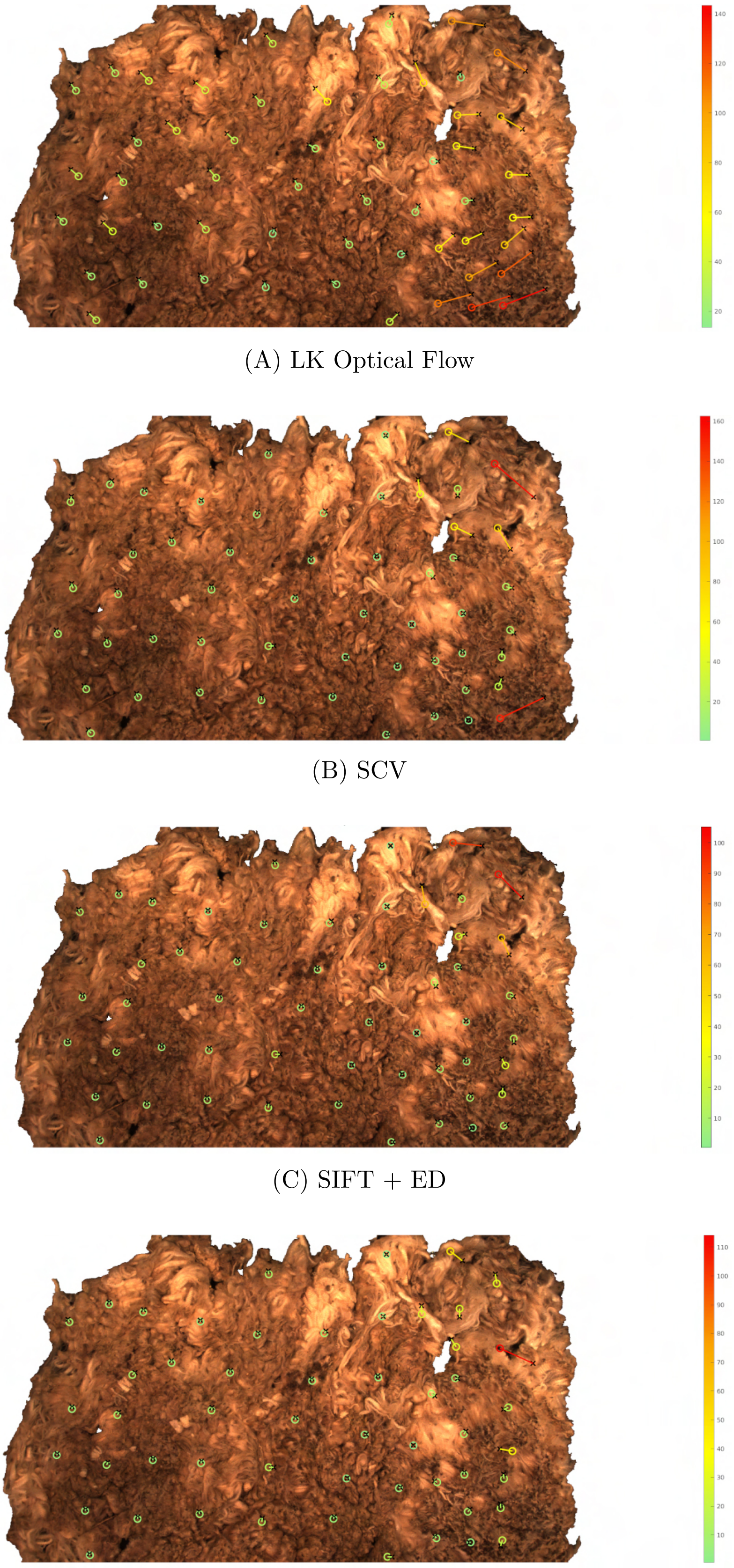


Figure 4.8: Quantitative evaluation of proposed method: Comparison between Lucas and Kanade [61], Jiang et al. [63], Banuelos et al. [11] and the proposed method to calculate the deformation error in the fleece. Using the colour markers as ground truth we calculate the distance between the location after deformation and the expected location.

in Figure 4.9, shows the location of the deformed and target markers. The observed error locations indicate that the other methods exhibit higher error rates, especially near the edge due to the incapability of extracting correspondences in areas with large deformation.

This evaluation demonstrates that an optic flow learning-based approach might not be suitable for calculating the deformation in fleece when only two images are available. Optic flow often requires several successive images to accurately calculate the deformation that occurs along the scenes. Additionally, the presence of previously unseen fleece coming from underneath during the skirting process severely affects the calculated flow fields, resulting in an inaccurate representation of the deformation.



(A) LK Optical Flow

(B) SCV

(C) SIFT + ED

(D) Proposed Method

Figure 4.9: Location error comparison between the deformed markers and target markers. The target markers are denoted with a coloured circle, while the location of each of the deformed markers is indicated with an 'x'. The colour of each of the circles represents the pixel error between the deformed and target marker, indicated by the colour bar.

This is supported by Figure 4.10, which shows the correspondences obtained using the previous chapter approach in green and the added correspondences by our approach in blue. It is clear that there is a noticeable increase in the number of correspondences near the edge of the fleece.

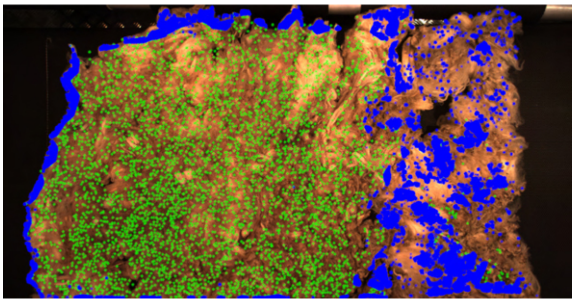


FIGURE 4.10: Added correspondences using the proposed method. Initial correspondences (green) were obtained using the previous chapter's approach and added correspondences (blue) after the new proposed filtering method.

4.8.3 Field Data

Further qualitative evaluation was performed on the data obtained from the field. As seen on Figure 4.11, a greater number of correspondences are observed closer to the edges of the fleece, in contrast to the approach proposed in Chapter 3. Furthermore, when compared to learning-based correspondences identified in LoFTR (Chapter 3, Figure 3.7), the proposed approach yields a higher concentration of correspondences near the fleece edges.

However, there are notable shortcomings when applying this approach to field data. The qualitative analysis highlights instances where correspondences are significantly sparse or completely absent. This issue is particularly evident in Figure 4.11b, which shows areas very close to the edge remaining without any correspondences. This lack of correspondences is attributed to substantial fibre stretching during the skirting process, which results in white fibres lacking recognisable patterns for feature correspondences.

4.9 Conclusion

In this chapter, we presented a novel image registration method to find the skirting line in two images of fleece. This method utilises a descriptor that incorporates the location

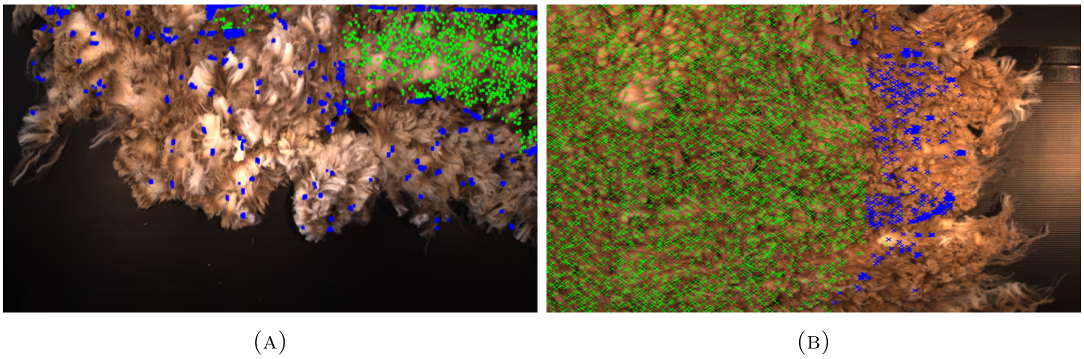


Figure 4.11: Added correspondences in field data using proposed method. Initial correspondences (green) were obtained using the previous chapter's approach and added correspondences (blue) after the new proposed method.

of the nearest neighbours, to ensure there is a continuous deformation during the filtering process. By utilising a learning-based filtering approach, we are able to gradually integrate dense descriptors into the inlier set, allowing us to start from the sparse correspondences extracted from the areas with low deformation and propagate to the edges where most of the deformation occurs during the skirting process. As demonstrated in controlled experiments, the proposed method outperforms the previously proposed approach and optical flow methods in areas with high amounts of deformation.

However, as noted on the results from the field data, this method fails to find correspondences at the very edge of the fleece that has been heavily stretched. The fibrous nature of wool can lead to transformations around the edge into an indistinct appearance in certain areas. This characteristic poses a challenge for feature extraction algorithms, making them incapable of finding feature correspondences in such regions. Furthermore, while quantitative results indicate an improvement over previous methods, the errors could result in either overskirting or underskirting, which may negatively impact the final product's value.

These shortcomings highlight the need for an approach capable of identifying how the fibres were stretched, decreasing the error around the edges when performing image registration. This could be achieved incorporating a physics based approach capable of replicating the fibre stretching phenomena.

Chapter 5

Skirting Line estimation using physics-based deformation

5.1 Introduction

The previous chapter introduced a learning-based non-rigid image registration method for aligning two RGB images of fleece. The method extended correspondences from less affected areas to the fleece's edges, which typically exhibit the most distortion. This was done by introducing a learning-based filtering method that incorporates a set of dense correspondences to the initial sparse correspondences and ensuring the continuity in deformation.

However, due to the unique characteristics of wool, this method fails to find correspondences at the very edge of the fleece where the fibre has been stretched. The stretching of the fibres alters the visual appearance of wool fibres located around the edges, such that no correspondences can be detected. Feature matching algorithms are incapable of matching features in these areas, leading to inaccuracies on the skirting line.

To address these challenges, this chapter introduces a physics-based simulation, designed to replicate the complex dynamics occurring at the wool's edge during the skirting process. As highlighted in Chapter 2, physics-based deformation methods have proven capable of accurately replicating the deformation or tearing of complex objects. Implementation of these approaches could improve the registration in areas where feature matching algorithms tend to fail due to the severe stretching of fibres.

Physics-based and non physics-based deformation methods have been combined to find denser correspondences in deformable objects ([98], [99], [100]). However, the simulation must be tailored to handle the deformation of fibrous materials such as those in Section 2.2.4.2 to accurately emulate the stretching of fibers and enable the identification of correspondences on heavily deformed areas.

5.1.1 Methodology Overview

A pipeline of the proposed approach can be seen in Figure 5.1, where a 3D mesh model of the before skirt wool is used in combination with a physics-based fracture simulation to extract correspondences on areas where the fleece undergoes significant stretching. These newly obtained correspondences at the very edge are then added to the previous set of correspondences obtained in Chapter 4. This allows for a denser tracking of the deformations occurring all over the fleece.

To accurately depict how the fleece deforms in real life, the physics-based simulation uses a learning-based process to replicate how the fracture propagates during the skirting process. This is done by optimising the materials physical attributes using the approach depicted in Figure 5.2. This optimisation initiates by preprocessing the mesh fleece to extract the different fibres in the fleece as outlined in Section 5.2. The fleece is then deformed using an initial set of material properties. Subsequently, the material properties are optimised through a learning-based approach, to accurately emulate the transformations that occur during the skirting process with the use of the simulation and correspondences.

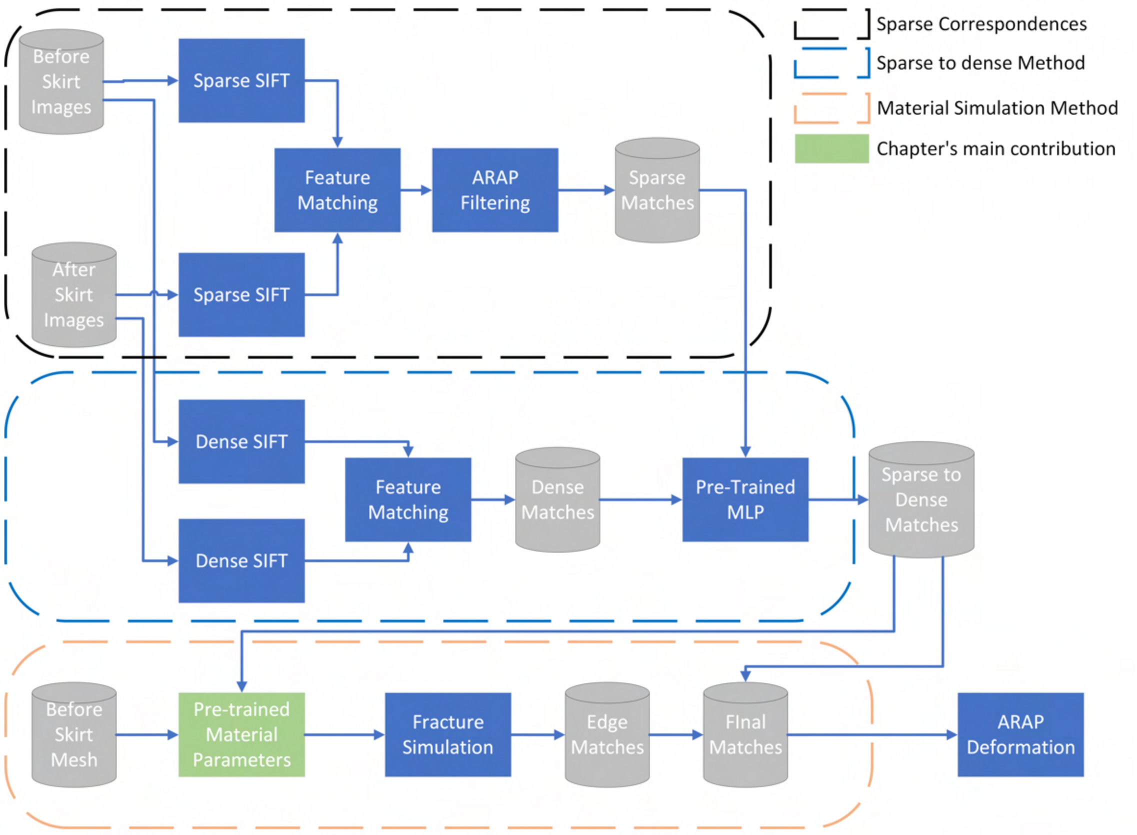


Figure 5.1: Pipeline of proposed methodology. A 3D mesh of the before skirt wool is obtained. This mesh is then deformed using a material fracture simulation with pre-trained parameters obtained using Figure 5.2. From the simulation, correspondences where the wool has been stretched are obtained. The after skirt image is then deformed using a combination of these newly obtained correspondences, previous correspondences from Chapter 4 (Sparse to dense) and initial correspondences from Chapter 3 (Sparse Matches) to deform the after skirt image to obtain the skirting line.

5.2 Mesh Preprocessing

5.2.1 Fibre Clusters

The initial stage of the pipeline focuses on the extraction of multiple fibre clusters found within the fleece in the before skirt image. The approach utilises superpixel segmentation [101] to extract the different fibre clusters present in the before skirt image. Superpixel segmentation groups pixels based on shared attributes such as colour similarity and spatial proximity. The proposed method employs Achanta et al. [102] approach, which adopts a k-means clustering approach to find the boundaries of each superpixel. This adaptation

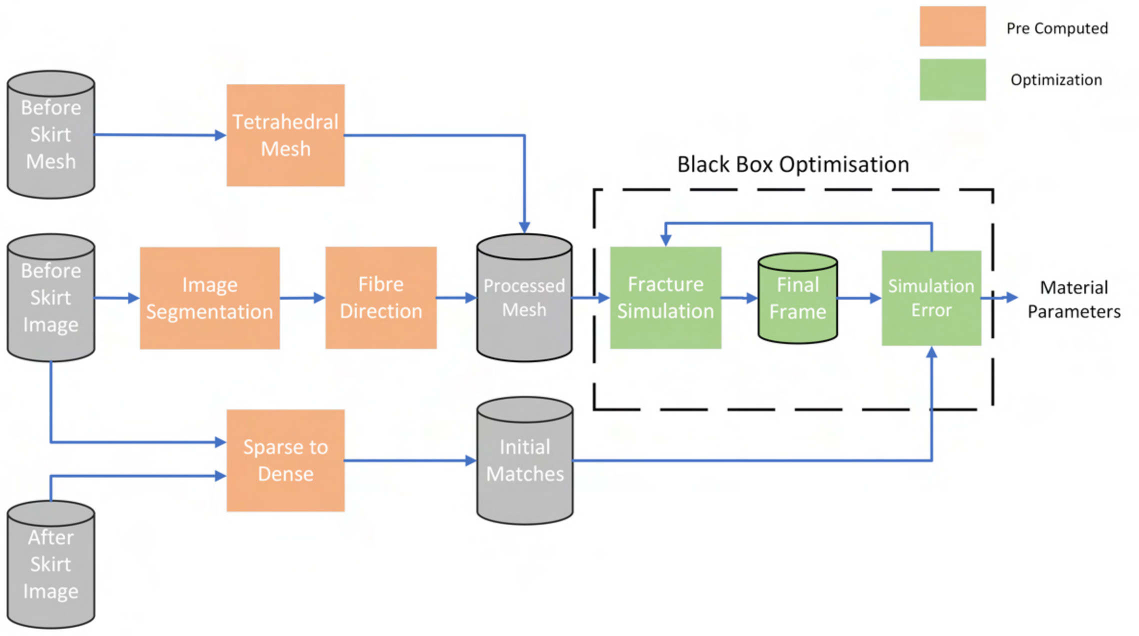


FIGURE 5.2: Pipeline of the proposed approach to learn material's parameters. An image of the before skirt wool is segmented into different superpixels. The fibre direction within each superpixel is then calculated using the gradients inside each superpixel. The mesh is then segmented into multiple sections obtained from the superpixels, each with their respective fibre direction. The material properties are then refined until the deformation in the simulation matches the after skirt image.

enables the efficient generation of superpixels, facilitating the accurate identification of the distinct fibre clusters in the before skirt image.

Each superpixel is stored as SP where $SP^i = \{p_1^i, \ldots, p_m^i\}$ with $p^i \in \mathbb{R}^2$. The term $p_{1..m}^i$ corresponds to the pixels locations within SP^i boundary.

5.2.2 Fibre Direction

Once the superpixels are defined, the Sobel operator [95] is employed to compute the image gradient within each superpixel area. The Sobel operator functions by detecting changes in intensity across a specific pixel array, generating a gradient vector. This method is particularly effective when applied to the grayscale image of the before skirt fleece, as it accentuates the distinction between the fibres and the background. This can be seen in Figure 5.3, where the fibres appear significantly darker.

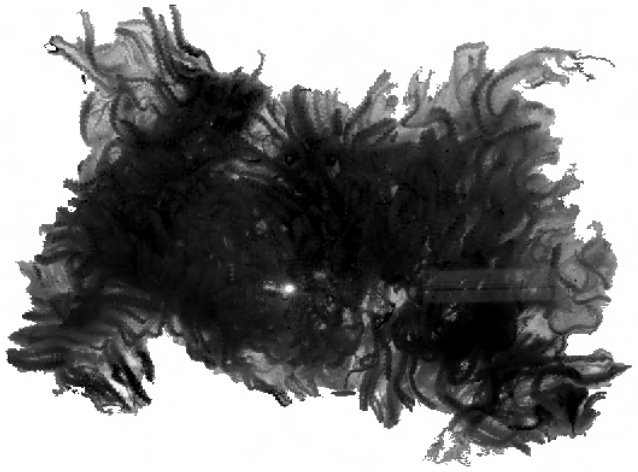


Figure 5.3: Grey scale image of fleece. Image highlights the location of the different fibres around the fleece

To obtain the direction of the fibre, the gradient vectors inside each superpixel are added together. During the summation process, opposing gradients or sides of the fibres effectively cancel out, leaving a resultant vector that represents the direction of the fibre orientation within the superpixel. These fibre directions are then stored as shown in Equation (5.1).

$$\vec{D}^i = \sum_{j=1}^{m} \vec{G}_j \in SP^i \tag{5.1}$$

where $\vec{D}$ represents the resultant direction vector of the fibres of the i^{th} superpixel, denoted as SP^i. The summation is taken over the gradient vectors $\vec{G}$ computed within a 3x3 area around each pixel. This cumulative gradient represents the predominant fibre direction within each superpixel.

5.2.3 Mesh Refinement

Once all the superpixels are obtained, the before skirt mesh M is then spatially transformed to match the scale and coordinates of the before skirt image. The transformed mesh is segmented into multiple meshes by cutting it along the superpixel boundaries.

To prevent discontinuities in vertices during the segmentation process, the initial mesh is transformed into a tetrahedral mesh using Delaunay tetrahedralization [103]. This Delaunay triangulation transforms the hollow mesh into a filled interconnected structure by using an algorithm that subdivides the mesh into tetrahedra. Subsequently, each segment is assigned its respective directional vector $\vec{\boldsymbol{D}}$, representing the fibre orientation within that mesh segment. The location of each of the vertices in the transformed and segmented mesh is then stored as $\boldsymbol{G}^1$.

5.3 Parameters Optimisation

5.3.1 Fracture Animation

We propose the use of *AnisoMPM* [88] for the simulation. As stated in Chapter 2, this simulation is capable of simulating fracture in fibrous anisotropic materials similar to wool during the skirting process. This capability stems from its utilisation of a three-dimensional matrix, which penalizes alterations in fibre length, shearing along fibres, and, frictional contact between fibres. Furthermore, its implementation is straightforward and demonstrates resilience under conditions of extreme deformation which is typical of wool.

The previously segmented meshes with their respective fibre direction are then integrated into *AnisoMPM*. This integration ensures that the simulated fractures do not propagate in the direction of the fibre orientations within each segmented area, allowing for a more accurate and real-world behaviour of wool during the skirting process. The locations of the deformed vertices from the simulation are then stored as $\boldsymbol{G}^2$.

5.4 Blackbox Optimisation

AnisoMPM can simulate accurate fractures in various materials by fine-tuning different parameters associated with the MPM properties. The parameters that need to be fine-tuned for the simulation are Youngs Modulus (E), Fibre Scale (γ), Viscosity (η), Inextensibilty (ρ) and Cauchy stress tensor (σ_{pos}). To tailor these parameters to accurately depict the fracture propagation in fleece, we employ a black box optimisation approach using CMA-ES [104].

Optimisation involving multiple parameters often poses significant challenges for conventional learning algorithms like gradient descent. These challenges arise from the non-smooth nature of the problem space and the excessive computational resources needed to calculate gradients. To effectively address these issues, CMA-ES is selected for its proficiency in navigating non-convex problem spaces, as encountered in behaviours such as stretching and fracturing under a range of forces.

Although CMA-ES is employed for this methodology due to its ease of application and popularity, other derivative-free algorithms, such as xNES (Exponential Natural Evolution Strategies) [105], MADS (Mesh Adaptive Direct Search) [106] and MCS (Multilevel Coordinate Search)[107], are also suitable for the proposed methodology.

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) works by employing an evolutionary strategy to optimise complex, non-linear, and multi-dimensional problems. This method systematically enhances a set of candidate solutions through an iterative process, favouring the propagation of lower error solutions for future generations. The method adapts by adjusting the covariance matrix of the distribution, effectively learning the shape of the problem space and guiding the search process towards the optimal solution.

5.5 Experiments

5.5.1 Data Collection

The dataset used to analyse the algorithm's performance consists of 6 different pieces of fleece. Each of them contains their respective 3D mesh and the before and after skirt image. The data was captured using the skirting rig shown in Figure 5.4, where the fleece is placed on the top of the table and the camera is placed underneath.

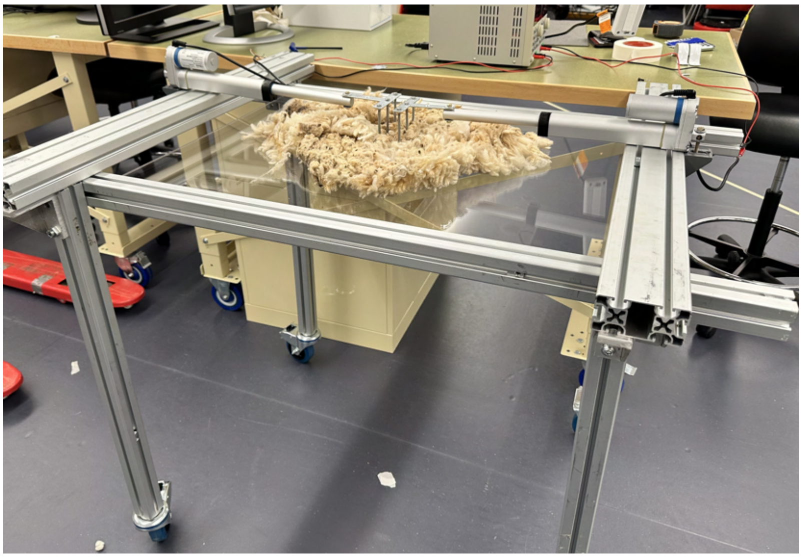


Figure 5.4: RIG used for data collection. The RIG consists of two linear actuators with an attachment used to simulate the hands of the wool handlers during the skirting process.

Eliminating outside variables when collecting the data represented a significant challenge. The non-linearity of the pulling during the skirting process increases the amount of errors for the simulation. Furthermore, the inability to capture the areas near the edge of the fleece during the skirting process due to obstruction from the hands also presents a challenge if wanting to obtain more information on how the fibres are stretched and compressed around this location.

To address these challenges, a new rig is built for the data collection of this experiment. In this rig, the fleece is placed above a perspex sheet and recorded from below. This setup eliminated occlusions and enabled full recording of the fleece during skirting. Variability in the pulling path and force was reduced using two linear actuators for pulling the fleece from each side. These linear actuators allow for a constant speed and pulling direction when skirting the fleece. Additionally, an attachment is placed at the end of each linear actuator to emulate the fingers from the wool handlers.

5.5.2 Mesh Processing

For the simulation, a total of 800 superpixels are generated to segment the image into distinct fiber clusters. The tetrahedralization process utilises a minimum allowable dihedral angle of 10 and a maximum allowable radius-edge ratio of 2.5. Furthermore, the number of final vertices employed for training in the simulation and experiments is set at 50,000. Additionally, the training for the blackbox optimisation utilises a subset of 9 different parameters for each of the iterations.

5.5.3 CMA-ES Training

The optimisation process involves minimising the error function shown in Equation (5.2), to ensure continuity between the deformations in the simulation and the images captured.

$$E_{sim} = E_{\text{correspondence}} + E_{\text{pixel similarity}} \tag{5.2}$$

Where $E_{correspondence}$ represents the Euclidean error between a set of correspondences and $E_{\text{pixel similarity}}$ is the pixel similarity between the simulation and the images. Both errors have been normalised to prevent any bias towards one error.

5.5.3.1 Correspondences Error

The training begins by extracting the correspondences between the before skirt and after skirt images denoted as I_t and I_s respectively. The correspondences are extracted using the previous approach in Chapter 4. These correspondences are then stored as two sets F^1 and F^2 where $F^i = \{f^i_1, \ldots, f^i_m\}$. With F^1 and F^2 corresponding to the features before and after skirt image respectively.

Additionally, to prevent any bias to the areas with low deformation, the features furthest from the fractured area are discarded. An example of the correspondences used can be seen in Figure 5.5.

The closest vertices from the segmented mesh G^1 to the features in F^1 are extracted and tracked over the simulation until they reach the same frame as the after skirt image.

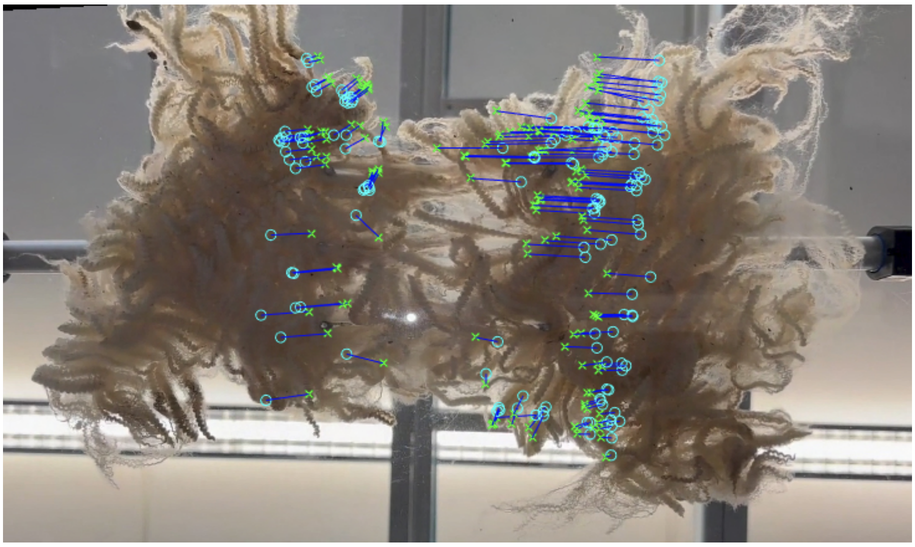


FIGURE 5.5: Tracked Points during skirting process. The blue lines represent the correspondences that are tracked during the skirting process for optimisation of the physics-based approach. These points move from the cyan circles to the green cross. They are located near the skirting location to track the stretching of the fibres.

The error is calculated by extracting the Euclidean distance between the after skirt features F^2 and the corresponding vertex in the deformed mesh (G^2). This error is obtained as shown in Equation (5.3):

$$E_{\text{correspondence}} = \frac{1}{m} \sum_{i=1}^{m} ||F_i^2 - G_i^2||^2 \tag{5.3}$$

Where, F_i^2 represents the location of the i^{th} correspondence in the after skirt image, G_i^2 denotes the corresponding vertex in the simulation at the same timestep, and m represents the total number of correspondences.

5.5.3.2 Pixel Similarity Error

The pixel ratio similarity between the after-skirt image and the deformed mesh is compared to ensure continuity in the deformation between both methods. To prevent any error from boundary discontinuities between the after skirt image and mesh, only the pixels in the middle section of the fleece are compared where the fibre stretching occurs. The similarity is computed by aligning the mesh after skirt simulation and the after skirt image and transforming them to a binary representation. The binary images represent the difference between the fleece and the background. The pixel similarity is then extracted as Equation (5.4)

$$E_{\text{pixel similarity}} = 1 - \frac{\sum_{i=1}^{m} |\mathcal{P}_i^1 - \mathcal{P}_i^2|}{m} \tag{5.4}$$

Where $\mathcal{P}_i^1$ represent the pixels of the in the after skirt image and $\mathcal{P}_i^2$ represent pixels of the transformed mesh M'. With i representing the different pixels in both images.

5.6 Results

5.6.1 Parameters Optimisation

The outcomes from the proposed methodology indicate that the optimisation effectively adjusts the material parameters to more accurately mimic real-life phenomena. As evidenced in Figure 5.6, there is a notable reduction in the mean squared error (MSE) between the correspondences and the simulation. Additionally, there is an enhancement in the pixel similarity ratio between the image and the mesh. This suggests that the fibres stretched in the middle section increasingly resemble their real counterparts in Figure 5.5 as the training advances.

While the optimisation method's training demonstrates continuous error reduction, it fails to plateau due to the high computational demands of certain iterations and the limited time of the research. This computational cost for the iterations can be seen in Table 5.1.

Due to the Lagrangian-Eulerian properties inherent in MPM, a sparser mesh cannot be utilised for optimising the simulation's parameters. This limitation arises from the optimisation method, which adjusts material properties based on particle density within each grid cell. For this reason, a sufficiently dense particle configuration is necessary for effective learning of the material's parameter, for accurate replicability on other pieces of fleece.

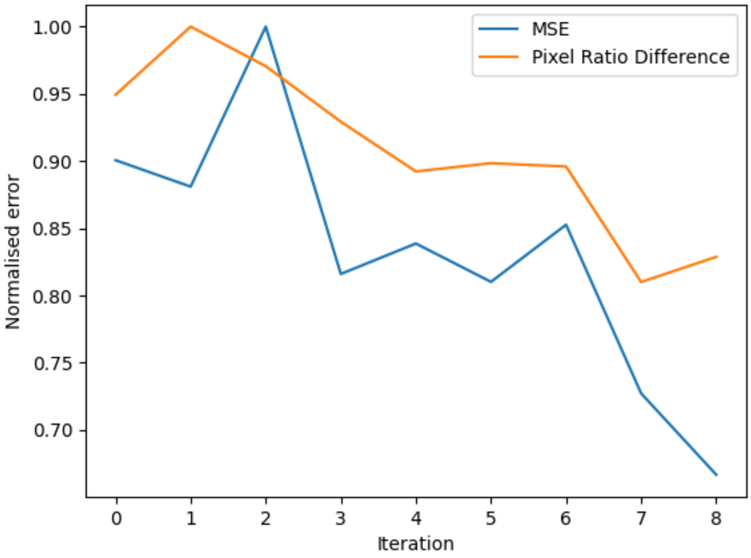


Figure 5.6: Variations in error across eight iterations of CMA-ES are presented, with the Mean Squared Error (MSE) depicted in blue and the pixel ratio in orange. To enhance readability, the Pixel Similarity Ratio has been adjusted to represent the Difference ratio between the mesh and the image.

Computational Cost	Under 1hr	1-3 hrs	3-8 hrs	Over 8 hrs
Candidates	19	17	23	13

Table 5.1: Computing time for each of the candidates in all the iterations. A total number of 8 iterations with a population size of 9 candidate solutions for each iteration was evaluated.

To further support the results of the optimisation, a qualitative evaluation of the proposed method is performed as seen in Figure 5.7, highlighting the mesh's development throughout the training phase in comparison to Figure 5.5. Figure 5.7a demonstrates the initial behaviour of the simulation during the first iteration, where the fleece exhibits a liquid-like consistency due to its material properties. Figure 5.7b demonstrates an improvement upon previous alterations, however, the fractured area is significantly larger than the one observed in Figure 5.5. Conversely, Figure 5.7c displays an outcome closer to the desired results, with the fracture concentrated in the central section of the fleece and surrounding segments undergoing stretching. This outcome more accurately reflects how the fleece behaves in Figure 5.5.

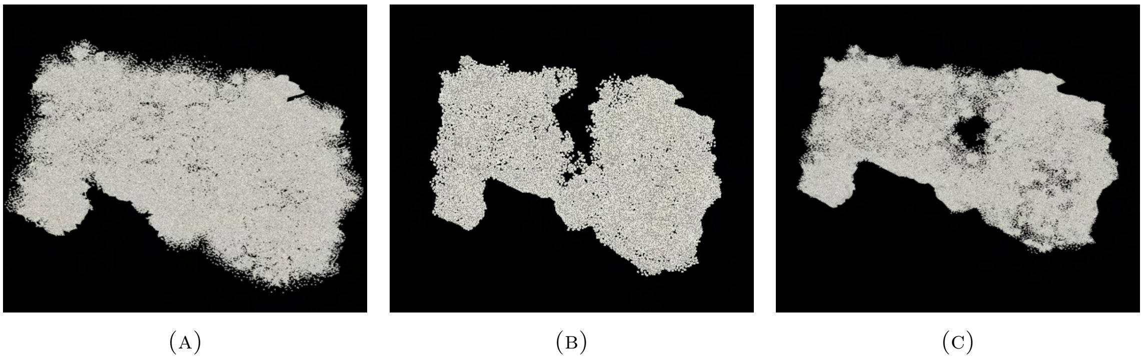


Figure 5.7: Change in fracture across training of parameters. The fracture propagation starts with a cut that extends through the entire fleece and gradually transitions to smaller areas where the fibres are more stretched as the simulation parameters are optimised. This progression closely mirrors what is observed in real-life videos.

5.6.2 Controlled Experiment

For the controlled experiment, a qualitative evaluation is performed using a different fleece obtained during the data collection outlined in 5.5.1 and a RGB video of the skirting process rather than just the before and after skirt image. This video allows for tracking of specific interest points across the entire deformation process. Utilising Doersch et al. [108] *TAPIR* implementation, we can accurately track any point in the video to serve as ground truth. For our specific application, we focus on tracking the points nearest to the fracture locations in the fleece, as these points undergo significant stretching and are the main focus of the approach. The tracked points can be seen in Figure 5.8.

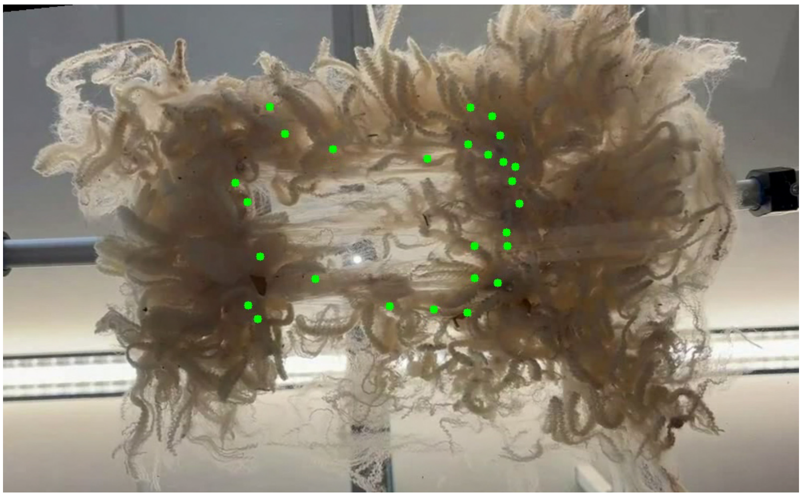


Figure 5.8: Points tracked during the skirting process. Points are extracted using TAPIR [108] and a video of the skirting process.

	Test 1		**Test 2**		**Test 3**	
Method	Pixel Error	Metric Error	Pixel Error	Metric Error	Pixel Error	Metric Error
SIFT + ED	205.91	111.30	150.08	81.12	210.2	113.61
Sparse to Dense	108.48	58.63	96.07	51.92	162.27	88.25
Proposed Method	92.23	49.85	86.57	47.79	144.97	80.03

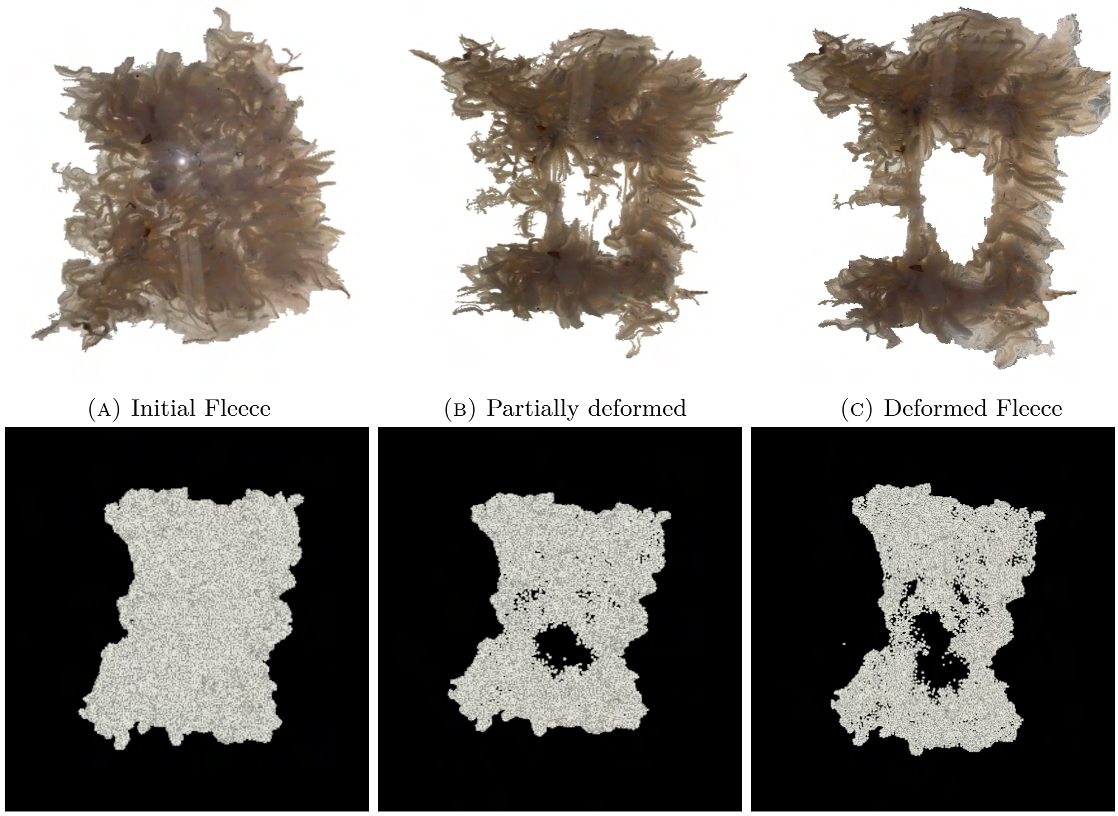
Table 5.2: Qualitative analysis of error. The pixel distance is obtained between the points obtained from TAPIR and the output of each of the previous approaches. Metric error is in mm.

The quantitative evaluation is performed by deforming I_s to match I_t based on the correspondences obtained from each of the methods. After deforming I_s, the error between the pixel location of the deformed I_s and ground truth correspondences on I_t is extracted.

The correspondences used for the proposed approach are obtained from the combination between Chapter 4 and the deformed mesh as previously stated on Section 5.1.1. Furthermore, to provide a comprehensive evaluation, a comparative analysis is conducted between our proposed approach and other registration methods, as detailed in Table 5.2. The registration methods selected for comparison are drawn from previous chapters, as described in Chapter 3 (SIFT + ED) and the approach presented in Chapter 4 (Sparse to Dense).

As seen on Table 3.2, there is a decrease in the amount of error compared to previous approaches. The robustness of the proposed approach is further evaluated by comparing two subsequent fleece samples against methodologies proposed in previous chapters. This is further supported by the qualitative analysis of the results shown in Figure 5.9, where a visual comparative analysis is demonstrated between the output of the simulation and the skirted fleece at the same time step. As observed in the results, both the fleece and the simulation exhibit comparable material behaviours, where the fleece being pulled experiences breakage primarily in the middle portion while the sides remain interconnected.

To further support the increment in correspondences, Figure 5.10 illustrates the additional correspondences introduced through the new approach. As indicated in the image, a denser distribution of correspondences is observed across the fleece, enhancing the accuracy of skirting line delineation following image registration.



(A) Initial Fleece (B) Partially deformed (C) Deformed Fleece

(D) Initial Mesh (E) Partially deformed (F) Deformed Mesh

FIGURE 5.9: Qualitative evaluation of the simulated vs real life deformed piece of fleece. A new piece of fleece, unseen during the optimisation process, was employed with the obtained parameters to assess the optimisation performance. As depicted in the image pairs, the simulated fleece maintains connectivity on the sides while exhibiting separation in the middle section during skirting, mirroring observations from the real-life video.

5.7 Conclusion

In this chapter, we introduced a simulation method capable of replicating the stretching behaviour of fibres at the edge of the fleece. This approach involved a fracture simulation that relies on accurate material properties to faithfully replicate the phenomena that occur during skirting. These material properties were acquired using a learning-based approach along with the method presented in Chapter 4. The material optimisation method's results were tested on controlled data acquired obtained where the user knows the locations of where forces were applied for the simulation

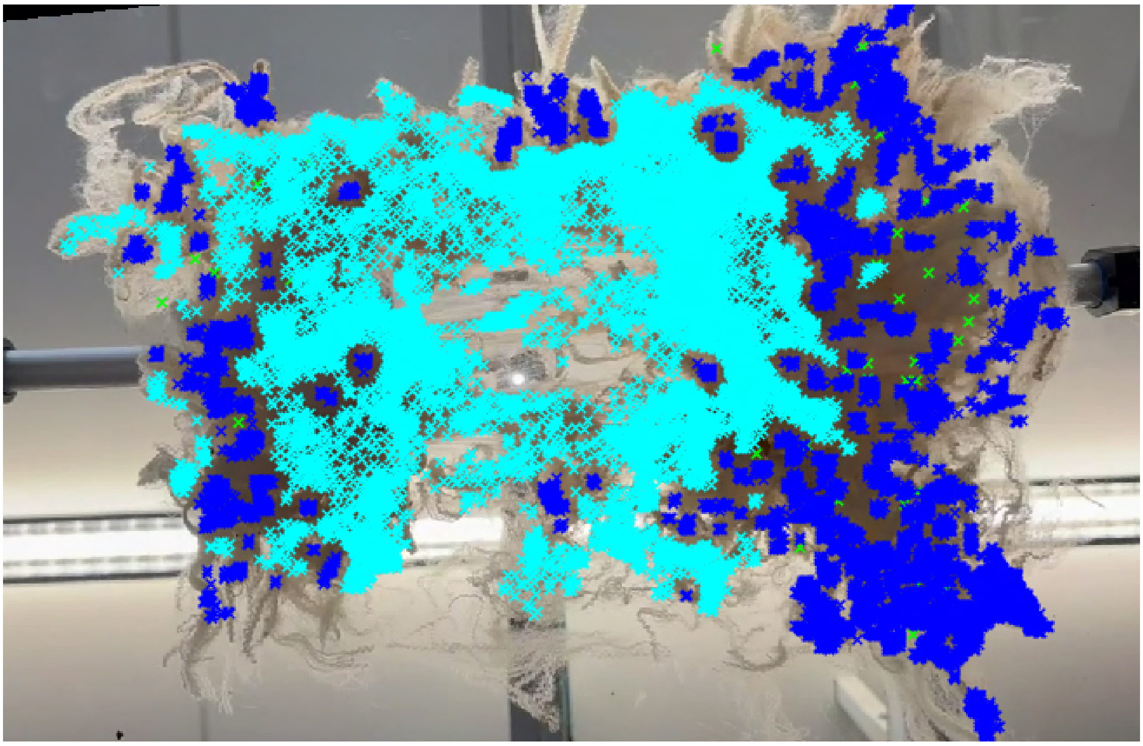


Figure 5.10: Correspondences from physics-based deformation. Newly obtained correspondences from the proposed approach are depicted in cyan. The green correspondences were obtained using the initial approach, as detailed in Chapter 3. The blue correspondences represent those identified by the learning-based method introduced in the previous chapter (see Chapter 4).

These material properties were subsequently applied to a different piece of fleece for validation of the parameters. The results demonstrate a reduction in error when compared to previously proposed methods. When tested against ground truth correspondences, the new methodology achieved a lower error in the location of the final correspondences. Despite the optimisation errors not reaching a stable state due to time constraints during training, qualitative results indicated an increase in similarity as training progressed, particularly in how fractures extended in the fleece compared to real-life skirting. This suggests that further improvements in skirting line detection can be achieved as the optimisation errors converge to a steady state during the training process. To reduce the computational time required to achieve this steady state, the employment of distributed computing techniques could be implemented in the proposed methodology.

Moreover, the quantitative results highlighted the methodology's ability to identify stretched fibres, enabling correspondence detection in areas where feature-matching algorithms falter. This error reduction leads to a more accurate skirting line, ultimately minimising waste and cost values in farm production applications.

Chapter 6

Conclusions and Future Work

In this thesis, we presented a method to automatically detect the location of the skirting line by aligning two images. The primary objective of this work is to address the existing limitations within the wool industry. By automating the detection process, this thesis aims to mitigate the industry's challenges related to the high demand for its products and the shortage of wool handlers.

One of the major challenges when performing image registration between wool images is the extreme non-rigid deformation the wool undergoes. To address this challenge, the thesis introduced multiple non-rigid deformation methods that overcome this issue.

6.1 Summary of Contributions

By aligning the before skirt and after skirt images, we presented a methodology capable of finding the location of the skirting line by extracting the areas that had been removed from the after skirt image. This model utilised a combination of feature matching and non-rigid deformation algorithms to perform non-rigid registration between both images. The outlier features are then removed using a filtering method based on optimisation formulas for non-rigid deformation (ARAP). While this method proved to be more effective than rigid registration methods, it encountered challenges in handling repeated patterns within the wool, leading to the removal of certain correspondences in highly deformed areas due to the presence of outliers.

For this reason, the initial methodology was further developed by introducing a learning-based filtering method that incorporates dense correspondences into the initial set. This approach overcame the limitations of the previous filtering method, by creating a feature descriptor that carried spatial information based on the nearest neighbours. With the use of this descriptor, the initial correspondences obtained were expanded to areas closer to the edge to obtain a better alignment between the two images.

Further correspondences in areas with low texture were then incorporated into the correspondences obtained in the previous approach. This was done by implementing a physics-based deformation approach capable of replicating fracture propagation in anisotropic materials. By optimising the material parameters that serve as input for the deformation, the simulation was capable of replicating what happens to the wool during the skirting process. Using this simulation, we successfully extracted correspondences in the stretched fibres, which are low-texture areas and have undergone extensive deformation, making them unrecognisable.

The combination of all these approaches resulted in a denser amount of correspondences between the skirted and unskirted images of the fleece. This leads to improved alignment between the images, reducing waste and minimising errors in the skirting line, which can lead to significant financial losses

6.2 Future Work

Although the proposed methodology enables us to find correspondences throughout the fleece, there are still some challenges that require attention.

For simulating the material fracture, the user must input the location where forces were applied to deform the object. While this information was readily available during controlled experiments, additional methods are required to extract it from field data. This can be achieved by utilising the correspondences obtained in Chapter 4. By identifying areas where the fleece moves in the opposite direction to the stretched fibres, the user can pinpoint the locations where the force was applied.

Further complications arise due to the variability inherent in the datasets used to train the proposed model. Although the method shows promising results in translating the trained model to unseen fleeces, the physical properties of the wool may vary depending on the type and the elapsed time since shearing. Delays in processing can lead to increased stiffness in the wool, thus altering its material characteristics.

The computational complexity of this approach also poses significant challenges, particularly for on-farm applications. The need for multiple components to detect deformation between the before and after skirt images makes it unsuitable for real-time use. Therefore, an alternative approach is to implement an end-to-end learning method, that could leverage the current methods capability in determining dense correspondence. Using multiple images acquired from the proposed methodology as training data to teach the system to automatically identify the skirting line's location, thus eliminating human errors in wool image annotation.

Bibliography

[1] Australian Department of Agriculture, Fisheries and Forestry. Wool. https://www.agriculture.gov.au/agriculture-land/farm-food-drought/meat-wool-dairy/wool#statistics-and-information, 2022.

[2] Cotton Australia. Cotton production. https://cottonaustralia.com.au/assets/general/Education-resources/CA-resources/CEK_Chap_7_Processing_From_Gin_To_Fabric.pdf, 2022.

[3] Australian Wool Innovation. Wool. https://www.wool.com/globalassets/wool/market-intelligence/wool-production-forecasts/australian-wool-production-forecast-report---september-2022.pdf, 2022.

[4] R. Bank. What to expect from wool in 2023, 2023. URL https://afdj.com.au/what-to-expect-from-wool-in-2023-according-to-rural-bank-research/.

[5] H. Wootton. Shearer shortages hurt wool industry, risks sheep welfare. https://www.afr.com/companies/agriculture/shearer-shortages-hurt-wool-industry-risks-sheep-welfare-20220121-p59q5p, 2022.

[6] J. Detz. Skirting the issue, 2017. URL https://ecori.org/photo-blog-skirting-the-issue/.

[7] J. Church and J. O'Neill. The detection of polymeric contaminants in loose scoured wool. *Vibrational spectroscopy*, 19(2):285–293, 1999.

[8] L. Zhang, A. Dehghani, Z. Su, T. King, B. Greenwood, and M. Levesley. Real-time automated visual inspection system for contaminant removal from wool. *Real-Time Imaging*, 11(4):257–269, 2005.

[9] T. Patten, A. Alempijevic, and R. Fitch. Learning image-based contaminant detection in wool fleece from noisy annotations. In *International Conference on Computer Vision Systems*, pages 234–244. Springer, 2021.

[10] D. Banuelos, R. Falque, T. Patten, and A. Alempijevic. Skirting line estimation using sparse to dense deformation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023.

[11] D. Banuelos, R. Falque, T. Patten, and A. Alempijevic. Skirting line annotation via deformation modelling. In *Australasian Conference on Robotics and Automation*, 2021.

[12] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *SIGGRAPH*, 20:151–160, 1986.

[13] S. Coquillart. Extended free-form deformation: A sculpturing tool for 3d geometric modeling. *Computer Graphics*, 24:187–196, 1990.

[14] Y. Chang and R. A. A generalized de casteljau approach to 3d free-form deformation. *SIGGRAPH*, 1994.

[15] P. Bezier. Mathematical and practical possibilities of unisurf. *Computer Aided Geometric Design*, pages 127–152, 1974.

[16] R. MacCracken and J. K. Free-form deformations with lattices of arbitrary topology. *SIGGRAPH*, pages 181–188, 1996.

[17] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *IPC Business Press*, pages 350–255, 1976.

[18] L. Moccozet and N. Thalmann. Dirichlet free-form deformations and their application to hand simulation. *IEEE*, pages 93–102, 1997.

[19] K. Singh and F. Eugene. Wires: a geometric deformation technique. *SIGGRAPH*, pages 405–414, 1998.

[20] K. Singh and E. Kokkevis. Skinning characters using surface-oriented free-form deformations. *Graphics Interface*, 2000.

[21] M. Liao, H. Wang, Q. Zhang, R. Yang, and M. Gong. Modeling deformable objects from a single depth camera. *IEEE*, pages 161–174, 2009.

[22] P. Huang, C. Budd, and A. Hilton. Global temporal registration of multiple non-rigid surface sequences. *IEEE*, pages 3473–3480, 2011.

[23] S. Yamazaki, S. Kagami, and M. Mochimaru. Non-rigid shape registration using similarity-invariant differential coordinates. *IEEE*, pages 191–198, 2011.

[24] B. Allen, B. Curless, and Z. Popovic. Articulated body deformation from range scan data. *ACM Trans. on Graphics*, pages 587–594, 2003.

[25] K. Li, J. Lai, and D. Guo. Robust non-rigid registration with reweighted position and transformation sparsity. *IEEE*, 2017.

[26] J. Yang, K. Lai, and K. Li. Sparse non-rigid registration of 3d shapes. *Eurographics Symposium on Geometry Processing*, 2015.

[27] R. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics*, 26:80–87, 2007.

[28] H. Li, R. Sumner, and M. Pauly. Global correspondence optimization fornon-rigid registration of depth scans. *Eurographics Symposium on Geometry Processing*, 5, 2008.

[29] H. Li, B. Adams, L. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics*, 28:1–10, 2009.

[30] F. Bonarrigo, A. Signoroni, and M. Botsch. Deformable registration using patch-wise shape matching. *Graphical Models*, 76(5):554–565, 2014.

[31] A. Myronenko, X. Song, and M. Carreria-Perpinan. *Non-rigid point set registration: Coherent Point Drif.* MIT Press, 2007.

[32] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE*, 31: 2262–2275, 2009.

[33] J. Ma, J. Zhao, J. Tian, Z. Tu, and A. L. Yuille. Robust estimation of nonrigid transformation for point set registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2013.

[34] J. Ma, J. Zhao, J. Jiang, and H. Zhou. Non-rigid point set registration with robust transformation estimation under manifold regularization. In *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[35] Q. Huang, B. Adams, M. Wicke, and L. Guibas. Non-rigid registration under isometric deformations. *Comput. Graph. Forum*, 27:1449–1457, 2008.

[36] C. Cagniart, E. Boyer, and S. Ilic. Free-form mesh tracking: A patch-based approach. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1339–1346, 2010. doi: 10.1109/CVPR.2010.5539814.

[37] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1395, 2017.

[38] K. Fujiwara, K. Nishino, J. Takamatsu, B. Zheng, and K. Ikeuchi. Locally rigid globally non-rigid surface registration. In *2011 International Conference on Computer Vision*, pages 1527–1534. IEEE, 2011.

[39] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015.

[40] F. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6): 567–585, 1989.

[41] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89:114–141, 2003.

[42] C. Domokos, J. Németh, and Z. Kato. Nonlinear shape registration without correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34: 943–958, 2012.

[43] R. Huang, J. Zhao, F. Duan, X. Li, C. Liu, X. Deng, Z. Pan, Z. Wu, and M. Zhou. Automatic craniofacial registration based on radial curves. *Computers Graphics*, 82: 264–274, 2019.

[44] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)*, 22(3):587–594, 2003.

[45] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Symposium on geometry processing*, pages 23–32, 2005.

[46] W. Chang and M. Zwicker. Global registration of dynamic range scans for articulated model reconstruction. *ACM Transactions on Graphics (TOG)*, 30(3):1–15, 2011.

[47] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. *IEEE*, 2007.

[48] H. Hontani, T. Matsuno, and Y. Sawada. Robust nonrigid icp using outlier-sparsity regularization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 174–181. IEEE, 2012.

[49] W. Li, S. Zhao, X. Xiao, and J. Hahn. Robust template-based non-rigid motion tracking using local coordinate regularization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 401–410, 2020.

[50] H. Pottmann, Q.-X. Huang, Y.-L. Yang, and S.-M. Hu. Geometry and convergence analysis of algorithms for registration of 3d shapes. *International Journal of Computer Vision*, 67:277–296, 2006.

[51] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. *Symposium on Geometry Processing*, pages 109–116, 01 2007.

[52] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[53] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[54] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[55] P. F. Alcantarilla, A. Bartoli, and A. J. Davison. Kaze features. In *European conference on computer vision*, pages 214–227. Springer, 2012.

[56] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[57] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011.

[58] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform, 2016.

[59] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018.

[60] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: Learning feature matching with graph neural networks, 2020.

[61] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[62] Z. Teed and J. Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419, 2020.

[63] S. Jiang, Y. Lu, H. Li, and R. Hartley. Learning optical flow from a few matches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16592–16600, 2021.

[64] D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: viscolelasticity, plasticity, fracture. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 269–278, 1988.

[65] A. Norton, G. Turk, B. Bacon, J. Gerth, and P. Sweeney. Animation of fracture by physical modeling. *The visual computer*, 7:210–219, 1991.

[66] S. Hirota, K. Isozaki, Y. Moriyama, K. Hashimoto, T. Nishida, S. Ishiguro, K. Kawano, M. Hanada, A. Kurata, M. Takeda, et al. Gain-of-function mutations of c-kit in human gastrointestinal stromal tumors. *Science*, 279(5350):577–580, 1998.

[67] N. Molino, Z. Bao, and R. Fedkiw. A virtual node algorithm for changing mesh topology during simulation. *ACM Transactions on Graphics (TOG)*, 23(3):385–392, 2004.

[68] Z. Bao, J.-M. Hong, J. Teran, and R. Fedkiw. Fracturing rigid materials. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):370–378, 2007.

[69] M. Müller and M. H. Gross. Interactive virtual materials. In *Graphics interface*, volume 2004, pages 239–246, 2004.

[70] O. Busaryev, T. K. Dey, and H. Wang. Adaptive fracture simulation of multi-layered thin plates. *ACM Transactions on Graphics (TOG)*, 32(4):1–6, 2013.

[71] S. Agarwal, V. Sriram, S. Yan, and K. Murali. Improvements in mlpg formulation for 3d wave interaction with fixed structures. *Computers & Fluids*, 218:104826, 2021.

[72] D. Koschier, J. Bender, and N. Thuerey. Robust extended finite elements for complex cutting of deformables. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[73] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal, vol. 82, Dec. 1977, p. 1013-1024.*, 82:1013–1024, 1977.

[74] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.

[75] M. Pauly, R. Keiser, B. Adams, P. Dutré, M. Gross, and L. J. Guibas. Meshless animation of fracturing solids. *ACM Transactions on Graphics (TOG)*, 24(3):957–964, 2005.

[76] D. Sulsky, S.-J. Zhou, and H. L. Schreyer. Application of a particle-in-cell method to solid mechanics. *Computer physics communications*, 87(1-2):236–252, 1995.

[77] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids*, 8(12):2182–2189, 1965.

[78] J. U. Brackbill, D. B. Kothe, and H. M. Ruppel. Flip: a low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48(1):25–38, 1988.

[79] A. de Vaucorbeil, V. P. Nguyen, and C. R. Hutchinson. A total-lagrangian material point method for solid mechanics problems involving large deformations. *Computer Methods in Applied Mechanics and Engineering*, 360:112783, 2020.

[80] K. Kumar, B. Sordo, C. Sung, and A. Vajapeyajula. Material point method, 2023. URL `https://www.geoelements.org/research/mpm/`. May, 2023.

[81] E. G. Kakouris and S. P. Triantafyllou. Phase-field material point method for brittle fracture. *International Journal for Numerical Methods in Engineering*, 112(12): 1750–1776, 2017.

[82] J. Wolper, Y. Fang, M. Li, J. Lu, M. Gao, and C. Jiang. Cd-mpm: continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.

[83] V. Hakim and A. Karma. Crack path prediction in anisotropic brittle materials. *Physical review letters*, 95(23):235501, 2005.

[84] B. Calvo, E. Peña, M. Martinez, and M. Doblaré. An uncoupled directional damage model for fibred biological soft tissues. formulation and computational aspects. *International journal for numerical methods in engineering*, 69(10):2036–2057, 2007.

[85] M. J. Borden, T. J. Hughes, C. M. Landis, and C. V. Verhoosel. A higher-order phase-field model for brittle fracture: Formulation and analysis within the isogeometric analysis framework. *Computer Methods in Applied Mechanics and Engineering*, 273:100–118, 2014.

[86] E. Kakouris and S. Triantafyllou. Material point method for crack propagation in anisotropic media: a phase field approach. *Archive of Applied Mechanics*, 88(1-2): 287–316, 2018.

[87] C. Jiang, T. Gast, and J. Teran. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.

[88] J. Wolper, Y. Chen, M. Li, Y. Fang, Z. Qu, J. Lu, M. Cheng, and C. Jiang. Anisompm: Animating anisotropic damage mechanics. *ACM Trans. Graph.*, 39(4), aug 2020. ISSN 0730-0301.

[89] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[90] Q.-H. Tran, T.-J. Chin, G. Carneiro, M. S. Brown, and D. Suter. In defence of ransac for outlier rejection in deformable registration. In *European Conference on Computer Vision*, pages 274–287, 2012.

[91] C. Jiawen, I. Shahram, and F. Andrew. KinEtre: Animating the world with the human body. *ACM symposium on User interface software and technology*, pages 435–444, 2012.

[92] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: application to breast mr images. *IEEE transactions on medical imaging*, 18(8):712–721, 1999.

[93] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1150–1157, 1999.

[94] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[95] I. Sobel, G. Feldman, et al. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272, 1968.

[96] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Eurographics Symposium on Geometry Processing*, 2006.

[97] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625, 2012.

[98] A. Mostayed, R. R. Garlapati, G. R. Joldes, A. Wittek, A. Roy, R. Kikinis, S. K. Warfield, and K. Miller. Biomechanical model as a registration tool for image-guided neurosurgery: evaluation against bspline registration. *Annals of biomedical engineering*, 41:2409–2425, 2013.

[99] G. Joldes, G. Bourantas, B. Zwick, H. Chowdhury, A. Wittek, S. Agrawal, K. Mountris, D. Hyde, S. K. Warfield, and K. Miller. Suite of meshless algorithms for accurate computation of soft tissue deformation for surgical simulation. *Medical image analysis*, 56:152–171, 2019.

[100] Y. Yu, G. Bourantas, B. Zwick, G. Joldes, T. Kapur, S. Frisken, R. Kikinis, A. Nabavi, A. Golby, A. Wittek, et al. Computer simulation of tumour resection-induced brain deformation by a meshless approach. *International journal for numerical methods in biomedical engineering*, 38(1):e3539, 2022.

[101] Ren and Malik. Learning a classification model for segmentation. In *Proceedings ninth IEEE international conference on computer vision*, pages 10–17. IEEE, 2003.

[102] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. doi: 10.1109/TPAMI.2012.120.

[103] S. Hang. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw*, 41(2):11, 2015.

[104] R. Hamano, S. Saito, M. Nomura, and S. Shirakawa. Cma-es with margin: lower-bounding marginal probability for mixed-integer black-box optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2022.

[105] T. Glasmachers, T. Schaul, S. Yi, D. Wierstra, and J. Schmidhuber. Exponential natural evolution strategies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 393–400, 2010.

[106] C. Audet and J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on optimization*, 17(1):188–217, 2006.

[107] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14:331–355, 1999.

[108] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. *arXiv preprint arXiv:2306.08637*, 2023.