Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Dissertation

Presented to the University of Dublin, Trinity College

in fulfilment of the requirements for the Degree of

Doctor of Philosophy in Computer Science

December 2022

Spatio-Temporal Processes for Volumetric Video Content Creation

Matthew Moynihan

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Matthew Moynihan

December 31, 2021

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Matthew Moynihan

December 31, 2021

Abstract

Volumetric Video is an emerging media platform which has recently undergone many new captivating developments. It could arguably be stated that recent uptake of volumetric video in consumer media would suggest that the platform is approaching maturity. That said, there still exists a very large barrier to entry for content creators as the technological requirements far exceed that of budget-constrained creators. Furthermore, even the more affluent creators find it difficult to navigate the large data footprint of volumetric video. Hence, there is a huge demand from these communities for new systems that improve upon the quality and accessibility of this medium. Techniques which seek to ensure Spatio-temporal coherence have yielded great success with traditional 2D video from quality improvements to reduced data compression overheads. In this dissertation we aim to investigate how spatio-temporal processes may be applied to volumetric video content creation with the ultimate goal of improving quality and accessibility by means of editing and compression. Specifically we will investigate this under three applications, including upsampling and filtering of point cloud sequences, autonomous tracking and registration of mesh sequences and frameworks for learnable registration of mesh sequences. Improvements to point cloud sequences allow for volumetric video content pipelines to improve spatio-temporal coherence from early stage reconstructions, propagating these qualities towards the final volumetric mesh outputs. Tracking and registration of

meshes further improves the quality of volumetric video while also adding temporal redundancy that can be exploited for compression. Finally the advantages of deep learning provide faster processing times and present a framework for more spatio-temporally aware network architectures. Under these three applications this dissertation will seek to answer the question, *how can spatio-temporal processes be applied to improve volumetric video content creation?*.

To my Mother and Father for their guidance, support and encouragement

Acknowledgments

The work of a PhD is often thought of as a lonely journey, undertaken by the student with few shared experiences outside of the supervisor and immediate co-workers. In my personal experience this could not be further from the truth. Though this work may be attributed to a single person on the title, it was the result of years of support from many people to whom I am forever grateful. I can only hope that my words of gratitude can do justice to the following people, all of which truly deserve to be recognised.

I would like to thank my reviewers Prof. Carol O' Sullivan and Prof. Adrian Hilton for their time and dedication spent in reviewing this dissertation.

I would like to thank my wonderful supervisor Prof. Aljosa Smolic for his guidance and inspiration. His endless optimism and patience kept me on the steadfast path, especially during those times when doubt sets in.

I'm also grateful for Dr. Rafael Pagés' help throughout the years, both as an academic advisor and as a friend. Many times he went above and beyond to help refine my work and carry it over the edge when it needed that final push.

Also to Dr. Susana Ruano Sáinz, whose diligence and attention to detail elevates everything we've ever collaborated on. She has been a pleasure to

work with and has taught me to seek perfection wherever possible.

I would like to thank all of my companions from the V-SENSE lab, especially those of the *1.02*, many of whom I would be happy to call friends for life. They were truly an inspiring lot and helped make V-SENSE a lab I was proud to work with throughout my PhD.

For my family I am eternally grateful. My Mother, Father and sister provided boundless support, especially during the trying times of the pandemic. While life was difficult for all, they always went out of their way for me when I needed it.

Finally I must express my sincerest appreciation for my partner Méadhbh de Blacam who bore witness to a dynamic spectrum of emotions and experiences throughout my time as a PhD student. Her patience and sympathy knew no bounds through times of celebration and despair alike. Without her support I would have surely faltered many times.

Matthew Moynihan

University of Dublin, Trinity College
December 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

To begin the discussion of topics related to Volumetric Video, one must first have a clear understanding of what is considered as volumetric content. Thus, this chapter serves to prime the reader by defining Volumetric Video by modern standards as well as it's current and potential use cases. This chapter also explores the technical aspects of creating such content as well as some of the challenges that creators face when using this emergent technology. Following in this context, the key research questions, contributions and structure of the document are presented.

1.1 Motivation

In traditional media such as film and TV, the viewer has no control over their view direction and is subjected to a 2D impression of the world. However, with the rise of new immersive media like Augmented and Virtual Reality (**AR&VR**), there is a need for new types of media that allow the user to freely move around a 3D scene. Free-Viewpoint Video (**FVV**) is a form of 3D video in which the viewer experiences visual content with 6 degrees of freedom (**6DOF**). FVV offers viewers the freedom to explore content within a 3D environment and the choice to experience it from arbitrary viewpoints.

To capture FVV sequences a set of cameras in different locations surrounding the 3D scene are typically used. In this context we classify methods for representing this 3D environment in which the user can freely navigate as a continuum of two extremes: image-based methods and geometry-based methods [1, 2]. While image-based methods generate virtual intermediate views by interpolating available natural views, geometry-based methods seek to build a 3D model of the scene that can be rendered from any novel viewpoint. We define Volumetric Video (**VV**) as the group of geometry-based FVV methods that build a 3D model of the scene, especially human performances.

While various forms immersive media exist, VV offers more by means of immersion and realism where other immersive media might compromise. For instance, 360 video is a similarly emergent form of immersive media which has been adapted widely by major platforms such as YouTube due to its relative accessibility. However, while VV offers complete 6DOF, 360 video offers only 3 degrees of freedom in that the viewer is restricted to explore the content via rotation about a fixed point. In short, VV offers a true experience of immersive depth while 360 video is equivalent to viewing a video projected on the inside of a dome. Figure 1.1 better illustrates this distinction.

Figure 1.1: **Volumetric Video vs 360 Video**. Top: Volumetric video refers to 3D media in which the user can freely navigate the scene and experience it from arbitrary viewpoints, [3]. Bottom: 360 video is a medium by which omni-directional videos are captured and projected to a viewing sphere, allowing only 3 DoF, [4]

1.2 How to Create Volumetric Video: A Typical Approach

Many sophisticated developments have occurred since the earliest proposed VV capture setups, from new capturing devices and compute power to efficient, high quality processing algorithms. What has remained constant in this time is the core concept of the VV studio configuration. This setup almost always consists of multiple cameras arranged about a central performance area in such a way as to maximise the number of viewing angles about the subject. This can be seen in the modern VV capture setup described by Collet *et al.* [5] whereby they utilize a combination of

RGB, Infra-red (**IR**) cameras and IR structured light projectors arranged in clusters alongside uniform white light on green-screen background. The use of RGB and IR allows for leveraging the advantages of multiple sensors while the uniform white light reduces shadows, allowing for more natural representation in digital environments where the VV content may need to realistically respond to digital environment lighting. Collet *et al*. propose a 3D reconstruction system which produces high-quality VV content but at the expense of requiring more than 100 sensors.

Others may take a more versatile approach, such as the system proposed by Pagés *et al*. [3], which opts for a more affordable approach using as few as 12 commodity RGB cameras. Removing the need for custom RGB and IR sensors results in a system which is affordable and lightweight but also highly dependant on the robustness of the 3D reconstruction algorithms. Such a system may reduce the barrier to entry for aspiring VV content creators but it can impart some constraints due to the lack of viewing angles as well as a comparable difference in quality against the aforementioned approach.

Further extreme developments towards accessibility of VV content creation have emerged only recently with the arrival of monocular 3D reconstruction approaches, often targeting consumer smart phone cameras as the only input to the system. The works of Saito *et al*. [6] and Habermann *et al*. [7] rely heavily on modern deep learning developments to propose 3D content creation captured by a single device, potentially unlocking volumetric video for the average smartphone user. While being the most accessible option, it continues the trend of being the most restrictive and lowest quality of the VV capture counterparts as one further reduces the number of imaging devices used. We show a visual comparison of the capturing methods discussed in figure 1.2.

Figure 1.2: **How To Capture Volumetric Video**. Left: The state of the art studio proposed by Collet *et al*. [5] featuring dense arrays of mixed sensors and lighting. Middle: A lightweight, mobile setup with sparse RGB cameras as proposed by Pagés *et al*. [3] and implemented by Vologrames [8]. Right: A single view input example as proposed by Saito *et al*. [6].

1.3 Problems with Volumetric Video Capture

Despite the many achievements of VV research and development in recent year, VV content still has some flaws and obstacles which will have to be overcome in order for it to experience widespread adaptability. Among these issues is the nature of how content is captured. As mentioned above, different studios will produce VV content with varying and inconsistent quality. This not only contributes to an overall negative quality of experience, but inconsistent content is highly challenging for standard compression and streaming algorithms, thus impacting distribution potential as well. As of recently, the need for such standards have been confirmed and recognised in the MPEG 136 call for proposals on dynamic mesh encoding [9]. In particular, many VV systems of the types described generate content on a per-frame basis, analogous to how old cartoons were hand-drawn one sheet at a time. Similarly, in frame-wise reconstructed VV we see temporal artifacts manifest in the form of *flickering* shape, incoherent topology and textures. Such spatio-temporal inconsistency has a drastic impact on the quality of VV content and offers none of the temporal redundancy which modern algorithms need for compression standards. Figure 1.3 shows an example of sequential meshes with differing topologies in service to no particular motion.

Additionally, as an emergent media, it's novelty combined with the aforementioned technical obstacles means that any form of post-production is either minimal or non-existent. Consequentially, creators have very little control over the output of a VV capture system after recording. This lack

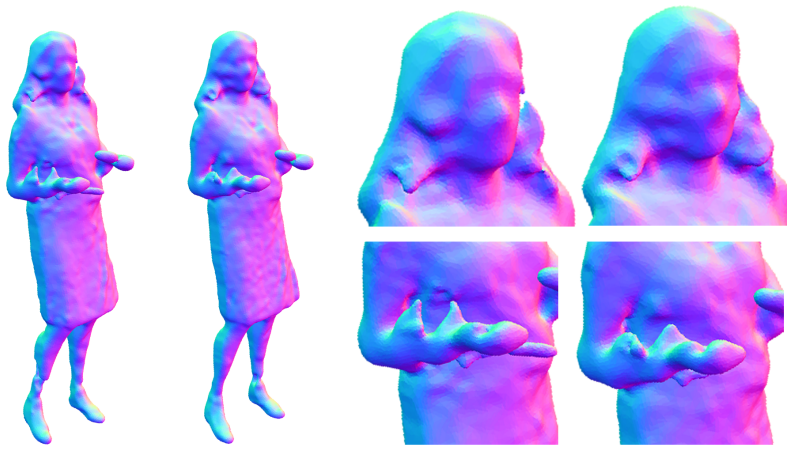

Figure 1.3: Typical unprocessed output from framewise VV reconstruction results in temporally incoherent geometry. The above image consists of temporally adjacent meshes in a sequence with face normals rendered as to highlight topology disparities.

of post-production along with the costs involved in assembling a high-quality professional studio make the idea of VV content creation somewhat unappealing to all but the more eager early-adopters. In summary, the following points can be seen as the most significant barriers to entry for VV content creation:

- Varying quality of reconstruction
- Lack of compression standards for distribution and playback
- Practically no existing post-production techniques
- Prohibitively demanding technical and equipment requirements

Across these points there is a common influencing factor of spatio-temporal incoherence. In short this refers to how the volumetric content behaves in a temporal manner i.e. how it moves and evolves about the duration of a VV sequence. While not immediately apparent, it can be argued that spatio-temporal analysis and applications can rectify or at least mitigate the major issues outlined. Indeed, it is the aim of this dissertation to formulate this argument and demonstrate it on applications for modern VV content. However, in order to present the research question in a com-

prehensive way, it will be necessary to explore what is meant by *spatio-temporal* coherence within the context of VV.

1.4 Spatio-Temporal Analysis

The discussion of spatio-temporal (**ST**) coherence of VV content arises naturally as it increases in spatial complexity compared to 2D content and very rarely concerns temporally static content. From a practical point of view, VV content creation can involve multiple types of 3D data representations including point clouds, signed distance fields and polygon meshes. In the context of this dissertation, ST coherence pertains to the properties and consequences of these data representations. For instance, ST analysis of polygon meshes may refer to the evolution and persistence of different vertices and vertex connectivity across a sequence as well as the implications regarding texture mapping and redundancies for compression i.e. a sequence of meshes with identical connectivity may share a single texture map (or, at least, texture coordinates) and offer significant redundancy for compression. Furthermore, shared texture maps and topology opens up the possibility of some post-production editing.

Similarly, this context can be applied to point cloud sequences, a common intermediate step found in VV content pipelines [5, 3, 10, 11]. Typically, the 3D reconstruction methods used to generate point cloud sequences introduce some stochastic noise when observed sequentially (the nature of which we will explore later). This is especially true for sparse camera setups as in [3, 11]. As a result, this ST incoherence propagates through each subsequent step in the pipeline ultimately manifesting as noise in the final output. An example of this can be seen in figure 1.4 as the point cloud structure varies in patches about the subjects head as well as density in the far arm.

Figure 1.4: **Filtering Point Cloud Sequences**. An incoherent point cloud sequence typical of a modern sparse VV system such as that of Pagés *et al.* [3].Notice in particular the structure noise patches around the head.

With this context in mind we can now revisit the major issues outlined in the previous section and see how ST process may be applied to solve or mitigate them:

- **Varying quality of reconstruction** - As ST incoherence has a direct and obvious impact on quality, a good ST process should be robust to highly incoherent output from even sparse or monocular systems
- **Lack of compression standards for distribution and playback** - Ideally, ST processes could be applied in order to create data redundancies such as shared topology and textures.
- **Practically no existing post-production techniques** - Shared topologies and textures would allow for trivial editing of texture data and even some potential for propagating topology modifications
- **Unobtainable by those without access to the correct technical resources or intellectual property** - If ST processes can be applied to noisy data such as that which is generated from monocular and sparse camera setups, then high-quality VV content could be produced from low-cost studios

1.5 Research Question and Contributions

In this section, we formally present the research question studied in this dissertation and summarize the key contributions discussed above.

Research Question

Broadly, we investigate — "***How can spatio-temporal processes improve Volumetric Video content creation?***".

We study this problem in the context of three objectives:

- Filtering and Upsampling Point Cloud Sequences.
- Tracking Mesh Sequences to allow Compression and Editing.
- Learnable Frameworks for Mesh Tracking and Registration.

Contributions

- We study the effectiveness of spatio-temporal filtering on the task of **point cloud upsampling and de-noising** as an essential processing step for typical VV systems. The proposed method achieves state-of-the-art results on synthetic and real data captured from a professional VV studio.

- We present a pipeline for **autonomous tracking of mesh sequences** which ensures spatio-temporally consistent output and allows for propagation of user editing. We build this framework based on highly conditioned correspondence matching across a hierarchy of geometry abstractions. Our system outperforms the state-of-the-art on mesh tracking tasks, in particular those targeting low-fidelity data from camera-sparse VV studios.

- We propose a **deep learning framework for tracking mesh sequences** with significantly faster computation time than traditional alternatives. Our method achieves competitive results on common benchmarks with considerably less computational overheads than the state-of-the-art .

1.6 Publications

Publications Based on Dissertation Work

- Matthew Moynihan, Rafael Pagés, Aljosa Smolic **Spatio-Temporal Upsampling for Free Viewpoint Video Point Clouds** , 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, February 2019 (VISIGRAPP), Prague.[12] (Best Student Paper Nominee)

- Matthew Moynihan, Rafael Pagés, Aljosa Smolic **A Self-regulating Spatio-Temporal Filter for Volumetric Video Point Clouds** Communications in Computer and Information Science (CCIS)(pp. 391-408), Springer, 2020 [13]

- Matthew Moynihan, Susana Ruano, Rafael Pagés, Aljosa Smolic, **Autonomous Tracking For Volumetric Video Sequences**, IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2021 [14]

- Matthew Moynihan, others, Aljosa Smolic, **Neural Non-Rigid Registration for Volumetric Sequences**, (To be submitted for review)

1.7 Dissertation Structure

This document is divided into six chapters with two appendices. In this first chapter we presented a brief introduction to VV, as well as the motivation and problems addressed in the dissertation.

In Chapter 2, we discuss the evolution of VV including earlier systems as well as modern systems and the issues they face. We then discuss some applications of graphics within the context of VV and conclude with an exploration of modern deep learning on 3D data and how it could be used for VV content creation.

In Chapter 3 we present a system for upsampling and filtering noisy point cloud sequences as a method for improving modern VV pipelines.

Chapter 4 will delve into mesh sequence tracking and registration, using ST for improved visual quality and consistent topology to enable compression. The proposed system is entirely autonomous and allows for geometry recovery as well as propagated user-edits.

Having established the value of mesh tracking and registration in the previous chapter, chapter 5 will propose a deep learning framework for registration of mesh sequences.

Each of these chapters stand on their own and can be read independently. The ordering is based on a rough chronological order followed during this dissertation. In Chapter 6, we conclude this dissertation by summarizing the key contributions and future research directions.

Chapter 2

Background

In this chapter, we discuss how VV came to be as well as previous research related to VV and other related areas. We explore some of the fundamental processes behind modern VV setups and the root of common issues across modern VV systems. Within the scope of VV we observe some overlaps between the fields of vision and graphics and how shared techniques lead to VV content creation. Finally, we briefly summarize some modern advances in 3D deep learning which allow us to develop neural network architectures for learning on VV outputs.

2.1 The Evolution of Volumetric Video

As is true for many innovative technologies, VV as we know it today has emerged from a history of preliminary works. Since the late 90's VV has roots in research streams that have gone under different titles in a relatively short space of time as the research evolved. Kanade *et al*. [15] present a very early work under the title of "Virtualized Reality" which proposed some of the key concepts such as achieving arbitrary viewpoints, using a dome-shaped capture studio and depth-estimation from multiple imaging devices. This approach inspired many follow-up works and had been cited among research streams presenting their new works under the title of "3D Video" [16, 17]. In these emergent works it can be seen that more attention has been given to the quality of the volumetric structure with

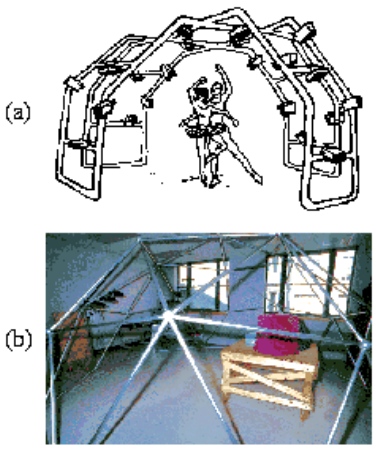

Figure 2.1: **Early Virtualized Reality Studio Concept** (a) The conceptual diagram for the virtualized reality studio (b) A photo of the actual studio from Kanade *et al*. [15].

the introduction of silhouette-based volume estimation to the pipeline. However, the term '3D Video' become interchangeable with early stereoscopic techniques involving creating two unique views which only gave the impression of depth [18]. As such the concept underwent another title change as the term "Free Viewpoint Video" appears in the work proposed by Carranza *et al*. [19]. While this work builds upon the 3D reconstruction techniques of it's predecessors, it deviates from direct shape estimation and instead introduces a system for performing marker-less motion-capture in order to drive the animation of pre-constructed 3D model.

The work by Smolic [1, 2] presents the taxonomy of the modern definition of FVV, where its complete pipeline, from capture to display, is analysed in detail. As introduced in Chapter 1.1, FVV can be understood as the functionality to freely navigate within real world visual scenes, and its methods for representing this 3D environment are classified as a continuum in between two extremes: image-based methods and geometry-based methods.

As an example of an image-based method, the work by Zitnick *et al*. [20] smoothly interpolates between the views of a camera array to generate novel views that compose the virtual viewpoints. This achieved a similar effect to the *'bullet-time'* effects made popular by in *The Matrix* movies.

These works are worth acknowledging due to the similar goals that they strove for however, they were concerned purely with view interpolation.

In contrast, geometry-based methods, which we include under the definition of VV as it is known today, include the works which attempt to acquire an actual 3D reconstruction of the scene. This is largely achieved using image-based 3D reconstruction, either through more traditional techniques, such as shape-from-silhouette and photogrammetry, or more modern algorithms that make use of deep learning. Most state-of-the-art VV pipelines include a combination of some of these methods to guarantee a complete and accurate result, either with dense camera setups and different types of sensors [5, 10] or with sparser setups that use commodity cameras [3, 11].

2.1.1 Feature-based 3D Reconstruction

Kanade *et al*. [15] built their seminal virtualized reality studio by largely building on a technique known as Multi-Baseline Stereo by Okutomi *et al*. [21]. Briefly, this technique allows one to use multiple overlapping images and triangulation to estimate and pixel depth and perform 3D reconstruction. This technique belongs to a family of processes known as photogrammetry which, as the name might suggest, involves using imaging sensors to infer 3D information from a scene. These techniques would form the primary backbone of nearly all traditional VV capture setups to follow.

The geometric premise of photogrammetry in it's simplest form is basic triangulation; for two or more views of a given point, estimate the distance between each view and said point. A stereo example of this is given in Figure 2.2 where we have two images, of the same point viewed from two different angles. In an ideal scenario, one would know the exact position and orientation of the cameras and could draw a line directly through the origin of each cameras and their respective points in the imaging plane to find that the lines intersect at the 3D point in space, marking the depth from each camera to this point.

This principle can be extended toward the multi-view scenario where the use of many cameras can provide multiple viewpoints and baselines for

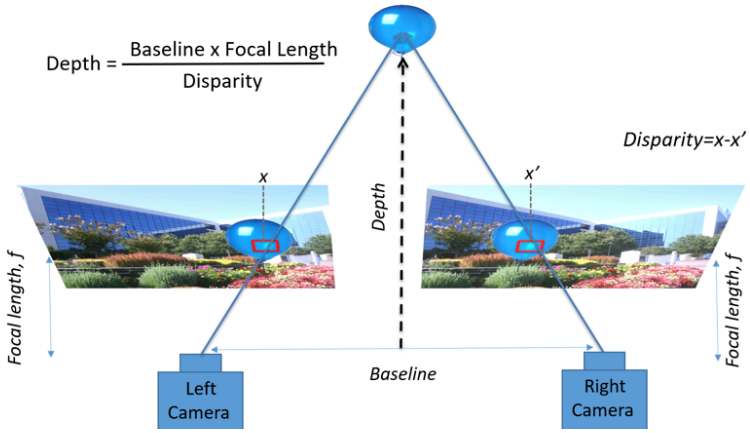

Figure 2.2: **Depth from Stereo Images** A simplistic model of an image sensing setup for depth estimation [22]

extracting 3D structure. This feature-based 3D reconstruction from multiple baselines is known as Structure-from-Motion which originated from [23] but more modern approaches follow the work of Snavely *et al*. [24]. It relies heavily on robust, invariant feature detection to be able to correctly match points from multiple views. Some widely used feature descriptors such as SIFT [25] or A-KAZE [26] have seen widespread use due to their ability to detect very robust features that are invariant to rotations and lighting changes. However, structure-from-motion generally assumes that the viewpoints are presented in an ordered manner and that each new baseline is an increment on the previous. As a consequence, the resulting point cloud structure can often be somewhat sparse. A technique called Multi-View Stereo [27] can be used to combine the sparse point cloud, multiple baselines and approximately known camera positions to further densify the output by estimating dense disparity maps between narrow baseline pairs. Modern applications exist for public use which implement some of the latest developments in structure-from-motion (COLMAP [28]) and multi-view stereo (OpenMVG [29]).

A myriad of other refinement techniques can be applied to reduce outliers (RANSAC [30]) and refine camera positions (Bundle-Adjustment [31]), but ultimately where feature-based reconstruction excels is in providing detailed structure for high-fidelity, feature-rich components such as tex-

Figure 2.3: The effect of image feature richness vs feature sparsity on feature-based 3D Reconstruction

tured garments or faces. In contrast, reconstructions can be somewhat lacking when presented more planar, feature-sparse objects as seen in figure 2.3. Even more so, when presented with translucent or non-lambertian surfaces prominent artifacts may sabotage the reconstruction entirely. To mitigate this shortcoming, it is common to use silhouette-based volume estimation techniques for a more low-fidelity but robust support solution.

2.1.2 Multi-View Volume Estimation

Shape-from-silhouette (**SfS**) is a group techniques that derive the volume of the scene from the intersection of silhouette cones arranged about the subject. These cones have the camera position as apex and the segmented silhouette of the subject as the base. Figure 2.4 illustrates the process first introduced by Baumgart [32] that has been refined and expanded on for applications in VV by others [16, 33]. The resulting ”visual hull” of the scene can be considered as an upper bound of the 3D scene that is being reconstructed.

In early works this was used in isolation as a volumetric solution on its own [34], albeit a low-fidelity one in comparison to later developments.

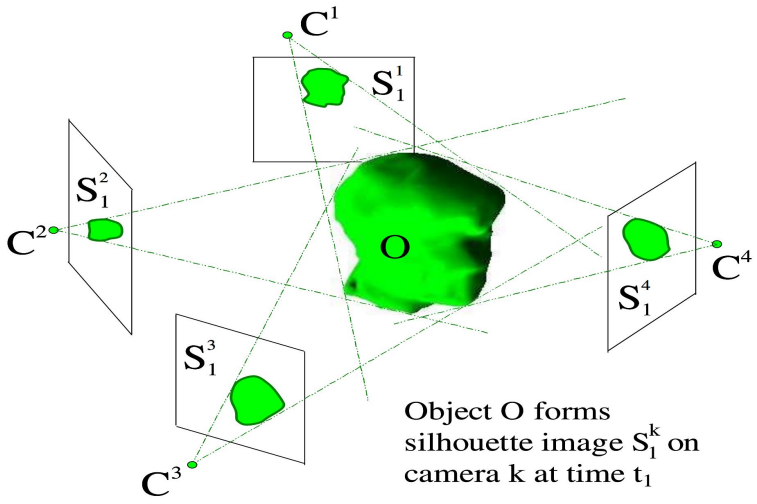

Figure 2.4: **Shape From Silhouette Concept** Given a set of cameras arranged about a subject with known position and orientations, a reconstruction of the subject can be formed by the intersection of silhouette images in space[33].

It allows for a very robust method of extracting a volume from multiple viewpoints and is not dependant on image feature detection, allowing it to be used to support some of the weaknesses of feature-based photogrammetry.

It does have it's drawbacks however. In addition to relying on the quality of the silhouettes and camera extrinsics, it also features a heavy reliance on angular resolution i.e. significant camera coverage. A consequence of this is how it responds when presented with significant occlusions. For example, in a lot of physical setups the vertical angular coverage is not as dense as the horizontal coverage, practically speaking, as it is more difficult to suspend or mount cameras and even more so to place cameras beneath the subject without obstruction. This in turn creates "shelf-like" artifacts when a horizontal occlusion occurs as seen in Figure 2.5. Furthermore, SfS-based methods cannot create concavities in the resulting volume, as these concavities (such as the eye sockets) cannot be represented in the subject silhouettes. This is partly mitigated by more advanced SfS techniques that use photo-consistency across cameras [35] to further decimate the visual hull, converting it into a "photo hull" [36].

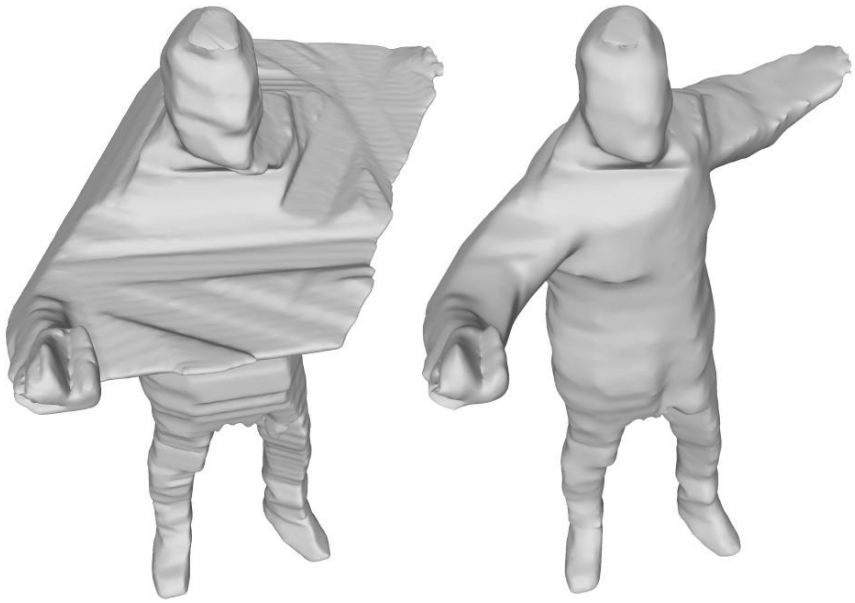

Figure 2.5: **Shape from Silhouette: Occlusion Artefacts** Left: A case of severe occlusions where few if any cameras were used to resolve the silhouette from a higher vantage. Right: A more ideal visual hull as a result of photo-consistent refinement and point cloud conditioned carving [3]

2.1.3 Active Depth Sensing

Up to this point we've discussed purely passive sensing methods for 3D reconstruction. It would be remiss to discount the benefits of using a direct approach such as direct depth sensing. These systems are quite robust in that true depth can be acquired due to the nature of active depth sensing. In the literature this was achieved using two types of technology:

- Structured Light: An infrared pattern is projected from a source at a known baseline from the sensor. The sensor then receives a distorted version of the pattern which can be used to infer a disparity map
- Time of Flight: A sensor floods an area with infrared pulses and calculates depth based on the time it takes for reflected light to return to the sensor

While both achieve a similar goal of using IR light to directly observe depth, ToF has emerged as the superior approach in recent years due to it's relatively improved quality and clarity. In general these are both widely used

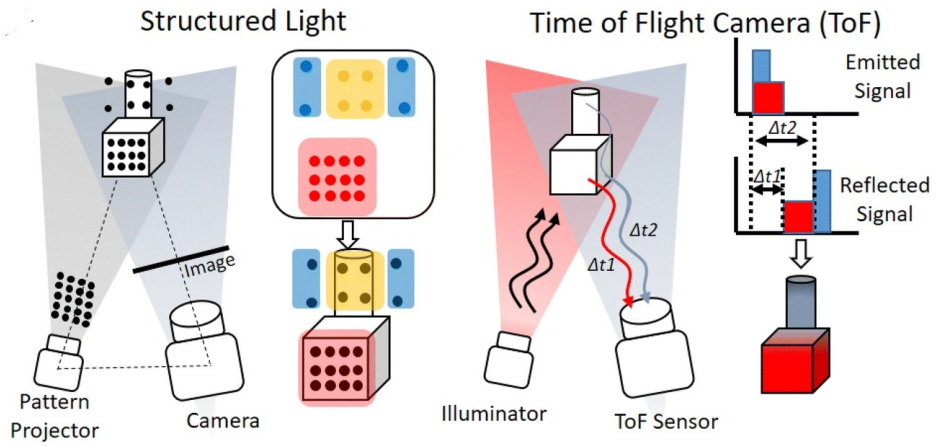

Figure 2.6: **Time of Flight Vs Structured Light Depth Sensors** A simple illustration of the principles behind these two sensors [37].

technologies for various commercial applications and although they present different issues when used for VV capture: artifacts along depth discontinuities, temporal flickering of depth values which increases significantly with distance, problems with some materials and dark colours that absorb IR light, and in some cases, interference among sensors when using multi-sensor setups that need to be especially synced.

Nevertheless, there are plenty of interesting works that rely on RGBD information to reconstruct dynamic humans [38, 39, 40, 41, 42].

How Does It Fit Together?

It can be seen that the discussed techniques each have their own strengths which support the other's weaknesses when it comes to different aspects of 3D reconstruction. However, up to this point the discussion has neglected the consequences of the temporal domain, opting only to look at artefacts on a static momentary basis.

Considering that these reconstructions are performed on a per-frame basis, the impact of incoherent random noise encountered across a temporal sequence of reconstructions is amplified. To loosely draw comparison to a familiar example, this noise could be considered akin to the temporal flickering artefacts seen in cartoons of the mid-20^{th} century, as these too were created in a frame-wise manner.

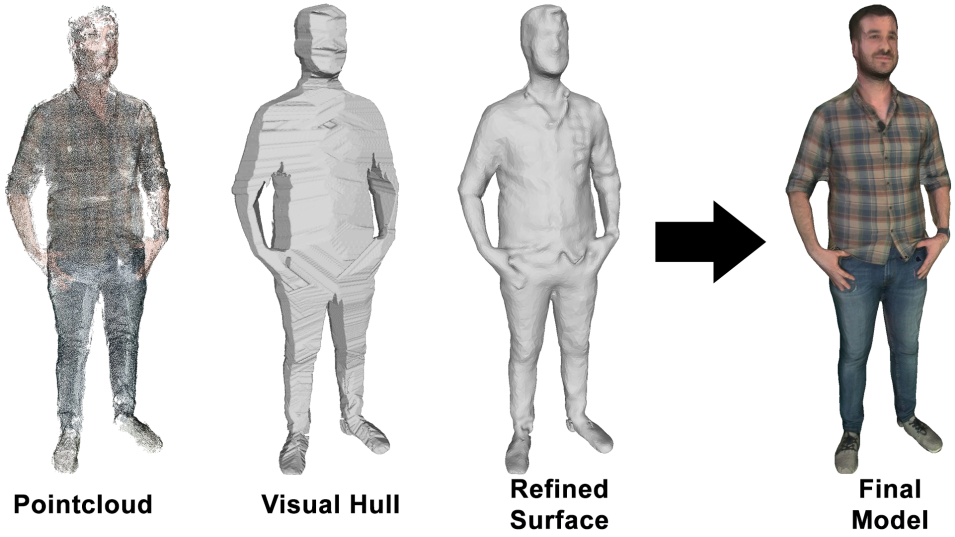

Figure 2.7: Modern Volumetric video often uses a combination of 3D reconstruction techniques to achieve the final topology. In this image we show a typical point cloud from a modern photogrammetry pipeline, a visual hull extracted from image silhouettes and a refined surface model which combines these inputs along with some other mesh refinements as in Pagés *et al*. [3]

While point cloud sequences are not often then end-product for volumetric video sequences, they are an essential part of the pipeline and as such, noise introduced at this stage can propagate into later stages and eventually the final output. We explore this further in Chapter 3 where the motivation behind this problem is explored in greater detail and a solution is proposed in the form of a self-regulating temporal filter. Similarly, as the outputs from these systems are fused into unified meshes, we see temporally incoherent structures and disruptions to the surface geometry manifesting across the volumetric sequence. To cease the discussion here would be to yield a volumetric mesh sequence which is geometrically incoherent across time and provides not only sub-par visual quality but also no redundancy for compression and distribution. As we take the discussion further along the typical VV processing pipeline we begin to venture into the space between computer vision and graphics. In the next section we discuss how developments in computer graphics can be used to improve the temporal coherence and redundancy in volumetric sequences.

2.2 Graphics Applications in Volumetric Video

Previously we discussed how to acquire the most fundamental 3D reconstructions on a per-frame basis and how multiple types of acquisition can be performed to optimize the result, producing a volumetric representation of the scene. It was also mentioned that if we consider only the per-frame results we ignore the temporal inconsistency of the final result when processed or experienced as a volumetric sequence. In the following section we will discuss how developments in computer graphics can be used to greatly improve the ST coherence of a volumetric sequence as well as add computational redundancy to improve encoding and compression.

As an emergent field there exists no such industry-standard processes or data formats for VV. Thus, in order to present a focused discussion we will assume that the desired output for a VV system is a sequence of polygon meshes with corresponding texture images.

2.2.1 Coherence and Compression

When considering how to improve coherence and compression of a sequence of individual textured polygon meshes it helps to consider them as two data streams, a geometry sequence and texture sequence. Both need to be compressed to enable efficient distribution and both need to be spatio-temporally coherent for quality of experience. Luckily the answer for texture is quite simple given the widely established literature on image/video coding. For image-based texture compression there is ample choice of compression methods with trade-offs in PSNR vs compression to suit a range of applications. On the other hand, geometry compression does not enjoy such a variety of options. We have seen some success with point cloud compression from the DRACO format [43], however polygon mesh compression is still a very open question for research. The current naive method would be to add redundancy by means of shared connectivity across meshes. That is, to attempt to apply consistency to the number of vertices, the ordering of vertices and the vertex neighbour indices in a temporal manner. In doing so we remove the overheads caused by monitoring these parameters on a per-frame basis and instead must

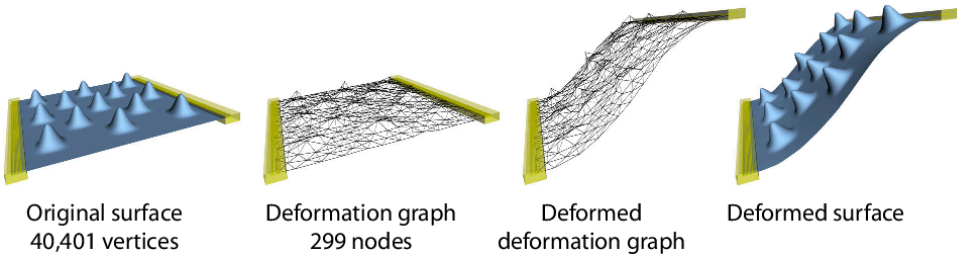

Figure 2.8: **Deformation Graphs**: The work of Sumner *et al*. proposes an embedded graph which supports shape deformation by applying an affine transformation at each graph node. The vertices of the underlying shape are deformed by a blended skinning towards nearby graph nodes [46]. In this example the deformations are driven by the displacement of the vertices highlighted in yellow.

only consider the vertex positions for a given sequence. Furthermore, the use of shared vertex topology allows for a singular texture mapping, reducing overheads even more.

For volumetric sequences the most natural way to achieve this redundancy is to perform non-rigid shape registration, a process by which one estimates a non-rigid deformation of one shape towards a target shape. This can be performed in sequence by deforming a ”keyframe” mesh iteratively onto a sequence of target meshes, thus maintaining the keyframe topology and storing only the deformations estimated at each step [5, 44, 14, 45]. While the concept seems straight forward to some degree, there is significant complexity involved in non-rigid shape registration which warrants further discussion.

2.2.2 Non-Rigid Shape Registration

Shape registration is a task which has seen numerous applications in computer graphics problems across medical, entertainment, automotive and other industries [47, 48, 49, 50]. Non-rigid shape registration (**NRR**) is a subset of registration tasks which extends the solution space beyond affine transformations, greatly increasing the complexity of the task. This complexity largely results in having to make trade offs between the accuracy and rigidity. Thankfully, due to the demand for solutions in this field there exists a rich state of the art for general shape registration emanating from

seminal works such as the Iterative Closest Point (**ICP**) by Besl and McKay [51] (and simultaneously to some degree but more application-specific, Chen and Medioni [52]). The basic procedure of ICP is as follows:

- For each point in the source shape, find the closest point in the target shape
- Estimate the rigid transformation which minimizes the point to point distance for all matched points
- Apply the estimated transformation
- Iterate until convergence e.g. no significant difference between successive transformations

The ICP algorithm can be considered one of the most fundamental algorithms used in shape processing today, being widely cited and providing the basis for further developments such as Generalized ICP [53]. It has also been expanded towards NRR by works such as the Optimal Step Nonrigid ICP Algorithms by Amberg *et al*. [54] and the probabilistic approach of Coherent Point Drift by Myronenko and Song [55]. With the exception of Amberg *et al*., these methods are generally most applicable to tasks concerning point cloud data. That is, when applied to surface meshes they neglect to retain some of the vital structural information contained in mesh topology data and can lead to destructive effects. The seminal work by Li *et al*. addresses this issue by supporting the non-rigid transformation of the surface with a deformation graph and applying a more conditional ICP variant that filters outliers [56]. This deformation graph is the core concept behind the *As Rigid As Possible* deformation scheme. The graph itself is created by uniformly sampling points about the surface and establishing edges via geodesic proximity. Each node in the graph represents an estimated rotation and translation which is applied to the underlying surface by interpolation of the skinning weights between the graph nodes and the surface vertices.

In chapter 4 we will present an autonomous shape registration system for unstructured mesh sequences which expands on the deformation graph framework with robust correspondences and geometry recovery.

2.3 Deep Learning for Volumetric Video

In prior sections we considered the fundamental components of traditional VV production. We have established what may be the closest resemblance to a conventional pipeline for such an emergent and dynamic field. There are however, some significant and potentially disruptive developments emerging from deep learning research that could have future implications on how we create VV. In recent years we have seen some extensions from established 2D deep learning techniques such as pose estimation being applied to specific VV tasks with some success. However, it is with recent revelations in the 3D learning space that we can see vastly different approaches to learning-based VV creation. In this section we will present some of the earlier uses of Deep Learning (**DL**) for VV creation and its limitations. We will also look at the challenges of applying deep learning to 3D data and how recent developments have unlocked new potential for high quality, learning-based shaped estimation as well as direct learning on surfaces. In many cases, the desired task for learning-based VV is to allow for monocular capture as opposed to cumbersome studios. Thus, most of the works presented in this section will be targeting data from personal devices such as smartphone cameras.

2.3.1 Template-driven Monocular VV

As deep learning was rapidly becoming ubiquitous among the computer vision community, we saw some early attempts to translate this success into the 3D domain for graphics applications [57, 58, 59]. The immediate challenge with adapting existing architectures was that the fundamental building block behind the most popular networks, the convolution kernel, is not particularly scalable when applied to 3D data. Some success was seen with PointNet[60], a network designed particularly for structured point clouds but significant obstacles appear when we consider such approaches for the primary format of VV, the polygon mesh. Indeed, essential concepts for convolution such as stride and padding become non-trivial to implement in the context of graph-like structures. As the literature evolved to tackle this problem we began to see novel architectures such as graph nets[61, 62] as well as spectral convolution oper-

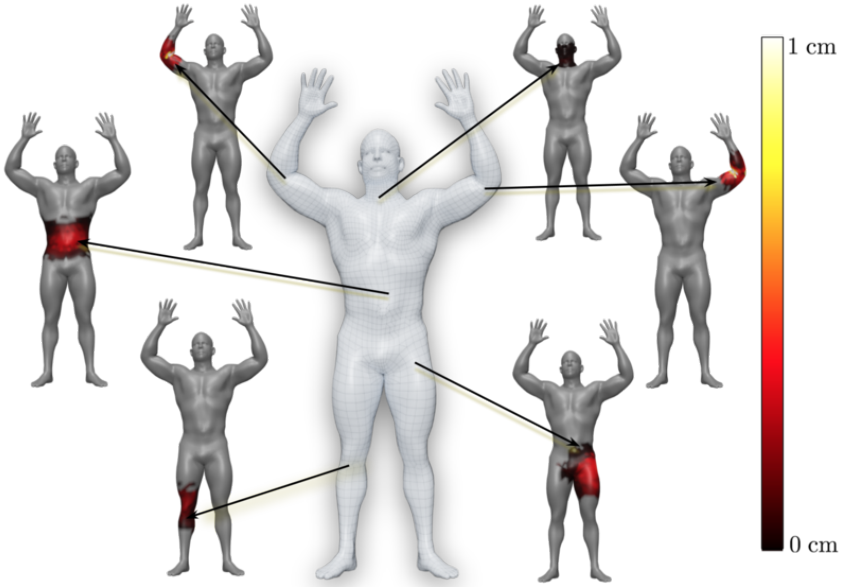

Figure 2.9: **STAR: A Sparse Trained Articulated Human Body Regressor**. STAR is a parametric human body with learned local blend shapes, that is proposed as a modern update for the SMPL model[67]

ations designed to replace or generalize traditional convolution kernels [63, 64, 65, 66]. We will explore these in more detail in section 2.3.3 after we consider some of the novel approaches to monocular VV used before it was possible to directly infer high quality meshes.

One way monocular VV was achieved efficiently was through the use of deformable templates created *a priori* of the given capture[7, 68]. Some of these methods saw transferable success from leveraging robust pose estimation networks in order to drive the deformation of the template[69]. The estimated pose could be lifted to 3D space where it would drive the articulated parameters of a skinned model. Improvements on this saw works which fit the scanned template to the parameters of the SMPL model [70], a parametric human body model. This modification allowed networks to perform estimations of these parameters instead of predicting a kinematic structure. In this way the estimated template deformations could model more fine-grained parameters than the limited kinematic joints thus leading to more visually pleasing deformation. This template-based approach is a continuous trend of research today and sees con-

sistent novelties and improvements, however some limitations are still present. As fine-grained and robust deformations continue to improve, the fidelity of the output is still limited to the representative capacity of the initial template. For example, one would be limited to the geometry and connected components represented by the template i.e. a template of an unclothed person could not accommodate the introduction large flowing garments or represent the sudden introduction of new objects like props or bags. Some very recent work has acknowledged this and is beginning to show some capacity for modelling clothed humans [71, 72, 73].

2.3.2 Occupancy Estimation for Shape Generation

A more generalized approach to monocular VV over template manipulation is to directly estimate the shape of the subject from image data alone. This is naturally a significantly more difficult challenge and had existed only in a very limited state prior to the emergence of learned implicit functions. Earlier template-free methods used a voxelisation approach to shape estimation [74]. These approaches were largely limited by high memory requirements and thus finer details were often absent. Most notably for the context of VV, the PIFu framework by Saito *et al*. [6, 75] uses implicit functions to directly infer a volumetric model of a subject given only an RGB image with an impressive degree of fidelity. This pixel-aligned implicit function is part of a family of works which utilize the concept of *deep level-sets*. Some concurrent works included Occupancy Networks [76], DeepSDF [77] and others [78, 79]. A later work by Mustafa *et al*. would extend the concept to multi-person scenarios [80]. The implicit function idea expands on the classical concept of a level-set representation of surfaces [81] replacing the need for solving a differential equation with a deep learning framework. In this way the network is designed to model an embedded continuous function which can be discretized at inference at an arbitrary resolution. Another useful property of these networks in relation to VV content creation is that they can be supported by multi-view inputs allowing for higher fidelity reconstruction in multi-view studio setups.

As promising as these networks are they can still be difficult to train for human shape estimation, especially as high-quality datasets are relatively

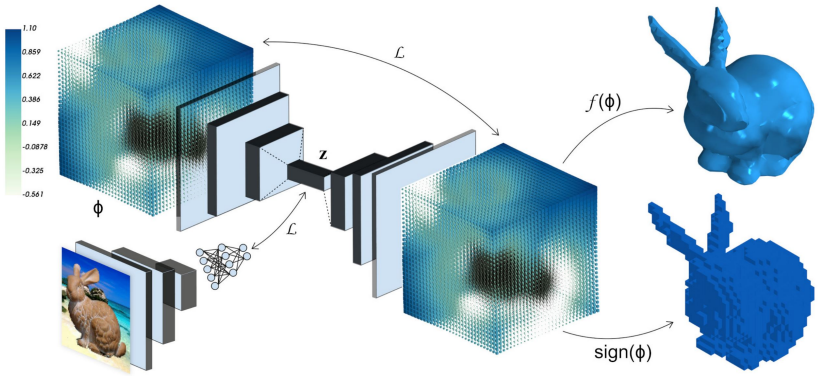

Figure 2.10: The deep Level-set Learning Architecture of Michalkiewicz *et al*. [78].During training, a 3D auto-encoder is used to project the shape into a z-dimensional latent space and back to 3D while a 2D predictor project image data into the same latent space. At test inference the encoder part is removed and only the 2D image data is used along with the 3D decoder to infer shape.

sparse among the research community. Among the available datasets, it is often seen that they are locked behind a paywall [82], low-resolution [83] or completely synthetic [84]. Additionally, while great improvements have been made in terms of scalability, these networks are still relatively expensive to train at high fidelity. Thus we have yet to see one which can perform some spatio-temporally consistent shape estimation for video sequence data. As a result the use of these network in a VV pipeline effectively still requires post-process tracking in order to reduce temporal noise.

Some notable works present spatio-temporally consistent results in that the general shape and motion is coherent. Caliskan *et al*. [85] propose a temporal-consistency loss for learning based shape estimation. Bozic *et al*. [86] propose a learnable deformation graph framework for temporally consistent reconstruction from RGBD data. It should be clarified that while these methods produce ST coherent shapes, they do not maintain the topological consistency required for compression and streaming. Hence, when we refer to ST coherent shape estimation we also assume some degree of topological consistency.

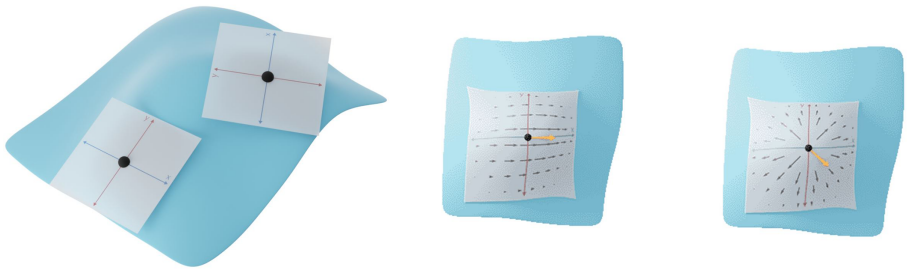

Figure 2.11: The work of Wiersma *et al*. [89] propose a framework which learns on surfaces via rotation-equivariant kernels. The learned features are robust to symmetry problems but rely on computationally expensive parallel transport operations in order to traverse the input.

2.3.3 Deep Learning on Surfaces

As new architectures make strides towards viable shape estimation for VV content creation we can continue to consider their application to other stages of the pipeline. As there currently exists no practical way of applying topologically consistent ST coherence to existing shape estimation networks, we should consider the post-processing approach. Sequential registration and tracking would be considered within the domain of shape analysis which leads us towards recent developments deep learning approaches to learning on surfaces. As mentioned in section 2.3.1, naively applying the success of convolution neural networks to polygon mesh data is inherently difficult due to the lack of shared properties between image data and mesh representations. While many point-based learning approaches exist [87, 60] they tend to be less accurate than surface-based methods and require extremely large datasets in order to achieve accurate results on shape deformation tasks [88].

Initial works which learn directly on surfaces tended toward generalizing convolution operations using local surface parameterization [90, 91]. Other works expand on this by introducing equivariance to rotation in order to address the lack of rotation invariance in the tangent plane of local parameterizations [89].

In a recent work, Sharp *et al*. [63] propose a DiffusionNet for learning on surfaces which employs a diffusion mechanism in place of established

parallel transport methods. Their semi-spectral approach demonstrates adaptability to generalized 3D data inclusive of point clouds, structured and un-structured polygon meshes. In chapter 5 we present a deep learning framework of sequential mesh registration that leverages DiffusionNet's capacity for learning robust features on meshes.

2.4 Summary

We conclude this chapter with a brief summary of the areas presented in the last few sections and highlight how that connects with our work in the next three chapters.

Some of the earliest works in VV were introduced as well as a brief introduction to the fundamental technologies that evolved and currently support VV. Among these we explored traditional photogrammetry, shape from sihouette and active depth sensing as well as how some of the issues they present individually.

From this basis we then went on to introduce contemporary graphics applications to temporally incoherent VV sequences for quality improvement and compression. Specifically we discussed non-rigid shape registration as a method for enforcing shared vertex connectivity across a sequence of incoherent meshes.

Finally we reviewed the latest developments in deep learning for shape estimation in relation to VV content creation and how deep learning on surfaces is still a growing area of research with distinct limitations that have yet to be overcome.

In the following three chapters we will rely on this knowledge to explore how ST analysis can be used to improve VV content in three domains. These include point cloud filtering and upsampling for pre-volumetric processes, autonomous tracking and registration for incoherent mesh sequences, and learning-based mesh sequence registration.

Chapter 3

Filtering and Upsampling Point Cloud Sequences

Figure 3.1: **Dense point clouds generated using an affordable VV capturing platform** [3]. Even after densification via multi-view stereo, the input clouds still exhibit large gaps in structure as well as patches of noise.

The generation and processing of point cloud sequences can be an essential component of modern VV content creation pipelines. Often times using sparse camera setups or unusual capture subjects (e.g. non-lambertian reflective objects) can lead to noisy point cloud data which must be filtered. Further, it would be highly beneficial for meshing algorithms if

the underlying point clouds were accurately densified. We have identified these concerns as a particularly useful application for ST solutions. Thus, in this chapter we present a self-regulating filter that is capable of performing accurate upsampling of dynamic point cloud data sequences captured using wide-baseline multi-view camera setups. This is achieved by using two-way temporal projection of edge-aware upsampled point clouds while imposing coherence and noise filtering via a windowed, self-regulating noise filter. We use a state of the art ST Edge-Aware scene flow estimation to accurately model the motion of points across a sequence and then, leveraging the ST inconsistency of unstructured noise, we perform a weighted Hausdorff distance-based noise filter over a given window. Our results demonstrate that this approach produces temporally coherent, upsampled point clouds while mitigating both additive and unstructured noise. In addition to filtering noise, the algorithm is able to greatly reduce intermittent loss of pertinent geometry. The system performs well in dynamic real world scenarios with both stationary and non-stationary cameras as well as synthetically rendered environments for baseline study. This work extends our prior work in Moynihan *et al*. [12] to which we will refer during the evaluation discussions.

3.1 Motivation

One of the fundamental processes in modern VV pipelines is ST consistency. Ensuring this consistency across the sequence of 3D models helps reduce the impact of small geometry differences among frames and surface artifacts, which result in temporal flickering when rendering the VV sequence. Most techniques apply a variation of on the non-rigid ICP algorithm [92, 40], such as the coherent drift point method [55], performing a geometric temporal constraint to align the meshes resulting from the 3D reconstruction process on a frame-by-frame basis [93, 94]. This works specially well when the 3D models acquired for every frame are detailed and accurate, as registration algorithms are not always robust to big geometry differences or loss of portions of the mesh (something that can often happen for human limbs). A good example of this is the system by Collet *et al*. [5]: they apply mesh tracking in the final processing stage, both to provide a smoother VV sequence and also to improve data storage efficiency as, between keyframes, only the vertex positions vary while face indices and texture coordinates remain the same. They achieve very appealing results by utilizing a sophisticated, very dense camera setup of over 100 sensors (RGB and IR), ensuring a high degree of accuracy for the reconstructed point clouds on a frame-to-frame basis.

This type of temporal consistency is also key in the methods proposed by Dou *et al*. [39, 42], where they are able to perform registration in real-time, using data coming from depth sensors. These methods ensure temporal consistency at the end of their pipeline, but differently to the method proposed, they do not address the loss of geometry in the capture stage, which can only be solved using temporal coherence at the point cloud generation.

The work of Cagniart *et al*. [95] improves on global ICP registration at mesh level by leveraging a patch-based registration approach while Budd *et al*. [96] improve global registration strategies by using shape similarity trees and non-sequential registration to minimise the global deformation required for registration.

Mustafa *et al*. [97] ensure temporal consistency of their VV sequences by

first, using sparse temporal dynamic feature tracking as an initial stage, followed by a shape constraint based on geodesic star convexity for the dense model. These temporal features are used to initialize a constraint which refines the alpha masks used in visual-hull carving and are not directly applied to the input point cloud. The accuracy of their results is not comparable with the methods mentioned above, but they show good performance with a reduced number of viewpoints and wide baseline. Mustafa *et al*. extended their work to include sequences that are not only temporally but also semantically coherent [98], and even light-field video [99].

An interesting way of pursuing ST consistency is by using optical flow. For example, Prada *et al*. [100] use mesh-based optical flow for adjusting the tracking drift when generating texture atlases for the VV sequence, adding an extra layer of ST consistency at the texturing step. It is possible to address temporal coherence by trying to use the scene flow to recover not only motion, but also depth. Examples of this are the works by Basha *et al*. [101] and Wedel *et al*. [102]. These techniques require a very dense and accurate motion estimation for every pixel to acquire accurate depth maps, together with a camera setup with a very narrow baseline. Alternatively, our system uses the temporally consistent flow proposed by Lang *et al*. [103] applied to multi-view sequences, allowing us to track dense point clouds across the sequence even with a wide baseline cameras configurations.

Other ways of improving incomplete 3D reconstructions, such as the ones acquired with wide baseline camera setups, include upsampling or densifying [104, 105, 106] them in a spatially coherent way. These systems are designed to perform upsampling for a single input point cloud, and not specifically a VV sequence, so they are unable to leverage any of the temporal information within a given sequence of point clouds. As a result, the use of such techniques alone will still suffer from temporally incoherent errors. Our system takes advantage of the geometric accuracy of the state of the art Edge-Aware Point Set Resampling technique proposed by Huang *et al*. [104] and supports it using the temporal information obtained from the inferred 3D scene flow along with some ST noise filtering. The reasoning behind this approach being that increasing the density

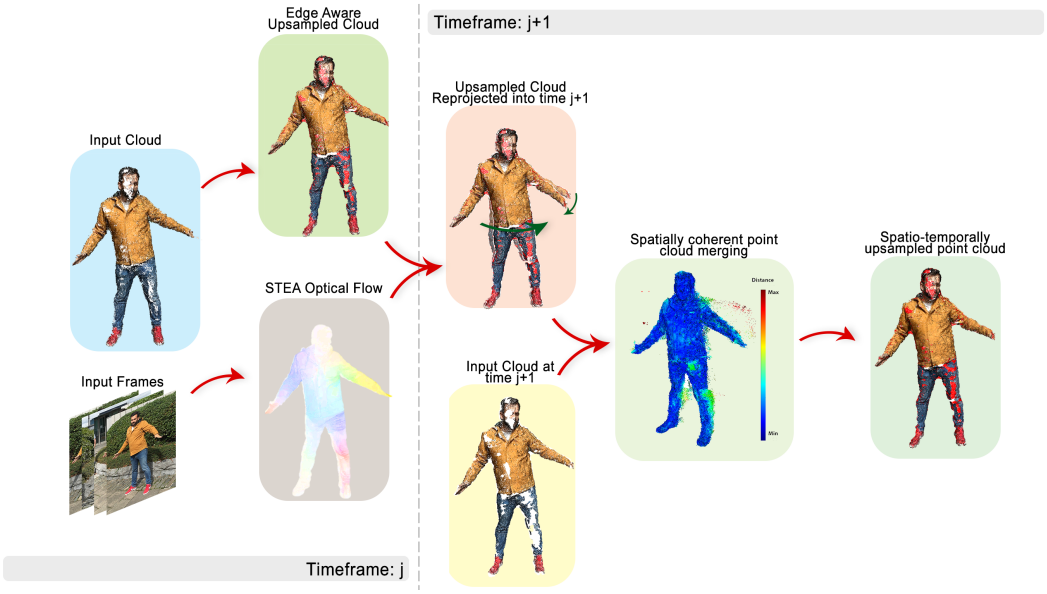

Figure 3.2: **Point Cloud Filtering Pipeline** The input to the algorithm requires a sequence of temporally independent point clouds along with the corresponding RGB images and calibration data. At timeframe j, the input cloud is upsampled and projected into the subsequent frame $t+1$. This is done via an edge-aware scene flow generated from the input RGB images. Expanding on [12], this is performed iteratively across a window of frames centered about the input frame i.e. we recursively project frames within the given window toward the center frame. The output consists of a spatio-temporally coherent merge and averaging system which upsamples the input point clouds and filters against temporal noise.

of coherent points improves the accuracy of surface reconstruction algorithms such as Poisson Surface Reconstruction [107] and thus, propagates visual improvement through the VV pipeline.

3.2 Filter Design

We use a low-cost VV pipeline similar to the system by [3] to generate the input clouds for the proposed algorithm. Such pipelines generally maximise the baseline between cameras in order to reduce the cost of extra hardware while still providing full coverage of the subject. The camera intrinsics are assumed to be known from prior calibration while extrinsics can be calculated automatically using sparse feature matching and in-

cremental structure-from-motion [108]. In some cases the cameras may be handheld, whereby more advanced techniques like CoSLAM [109] can be applied to better produce dynamic poses. The input sparse clouds are further densified using multi-view stereo. The examples presented within the context of our system use the sparse point cloud estimation system by Berjón *et al*. [110] and are then further densified by using the unstructured MVS system of Schönberger *et al*. [111]. Formally, we define $S = \{s_{i=1}, ..., s_m\}$ as the set of all m video sequences, where $s_i(j)$, $j \in \{1, ..., J\}$ denotes the jth frame of a video sequence $s_i \in S$, with J frames. Then for every frame j, there will be an estimated point cloud $\mathcal{P}_j$. In a single iteration, $\mathcal{P}_j$ is taken as the input cloud which is upsampled using Edge-Aware Resampling (EAR) [104]. This initializes the geometry recovery process with a densified point cloud prior which will be temporally projected into the next time frame $j + 1$ and geometrically filtered to ensure both temporal and spatial coherence. With the windowed filtering approach this iteration is performed recursively in such a way that each frame within the window is iteratively projected toward the center frame via it's respective intermediate frames.

3.2.1 Spatio-Temporal Edge-Aware Scene Flow

Accurately projecting geometry from between different timeframes is directly dependent on the accuracy of the scene flow used to achieve it. In the context of this paper the scene flow used is actually a dense, pseudo-scene flow which is generated from multi-view videos as opposed to directly extracting it from the clouds themselves. This scene flow is calculated as an extension to dense 2D flow, thus, for every sequence s_i we compute its corresponding scene flow f_i. This view-independent approach ensures that the system is robust to wide baseline input.

To retain edge-aware accuracy and reduce additive noise we have chosen a dense optical flow pipeline that guarantees ST accuracy:

- Initial dense optical flow is calculated from the RGB input frames using the Coarse to fine Patch Match(CPM) approach described in [112].
- The dense optical flow is then refined using a ST edge aware filter

Table 3.1: Investigation by [12] on the effect of STEA filter initialization on geometry recovered expressed as % increase in points. Tested on a synthetic ground-truth sequence. Flow algorithms tested: Coarse-to-Fine Patch Match (CPM) [112], Fast Edge-Preserving Patch Match (FEPPM) [115], Pyramidial Lukas-Kanade (PyLK) [116] and Gunnar-Farnebäck (FB) [113].

STEA Initialization	Area Increase (%)
CPM	**37.73**
FEPPM	34.9
PyLK	34.77
FB	29.7

based on the Domain Transform [103].

The CPM optical flow is used to initialize a ST edge aware (STEA) filter which regularizes the flow across a video sequence, further improving edge-preservation and noise reduction.

While the STEA can be initialized with most dense optical flow techniques such as the popular Gunnar-Farnebäck algorithm [113], the proposed system uses the coarse-to-fine patch match algorithm by [112] as recommended in [114]. Table 3.1 provides a breakdown of the amount of pertinent geometry recovered via different optical flow techniques.

The STEA filter consists of the following implementation as in [103]. This implementation further builds upon the Domain Transform [117] extending into the spatial and temporal domains given the optical flow as the target application:

1. The filter is initialized as in [114], using coarse-to-fine patch match [112]. The CPM algorithm estimates optical flow as a quasi-dense nearest neighbour field (NNF) using a subsampled grid.

2. The edges of the RGB input are then calculated using the Structure Edge Detection Toolbox [118].

3. Using the calculated edges, the dense optical flow is then interpolated using Edge-Preserving Interpolation of Correspondences [119].

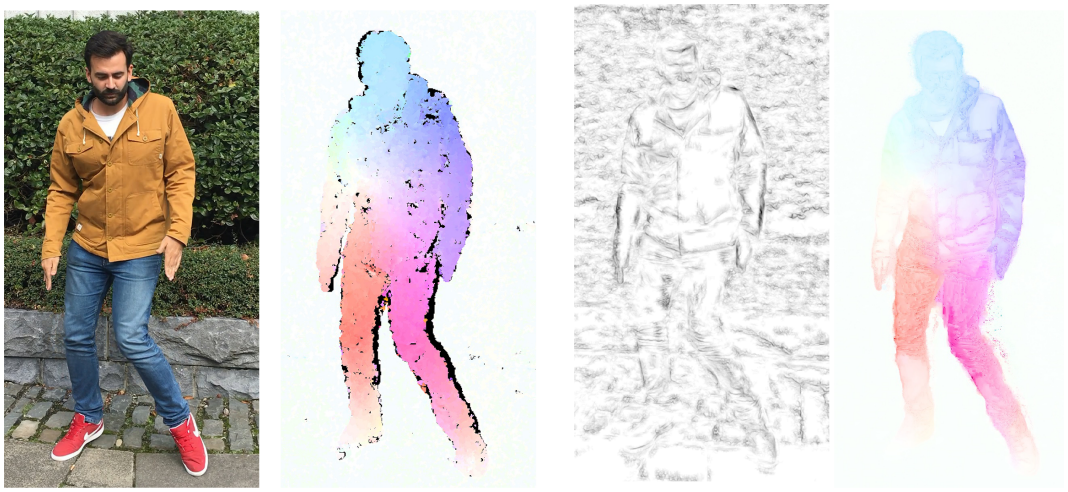

Figure 3.3: From left to right, dense optical flow calculation: For a particular viewpoint, the input RGB image, (1) nearest neighbour field estimate from CPM, (2) SED detected edges, (3) interpolated dense STEA output. Conventional colour coding has been used to illustrate the orientation and intensity of the optical flow vectors. Orientation is indicated by means of hue while vector magnitude is proportional to the saturation i.e. negligible motion is represented by white, high-speed motion is shown in highly saturated color [12]

This dense optical flow field is then regulated by the STEA filter via multiple ST domain iterations to reduce temporal noise. Figure 3.3 visualizes the intermediate stages of the flow processing pipeline.

3.2.2 Scene Flow Point Projection

Given known per-camera intrinsics ($C_{j_1}, ..., C_{j_m}$, at timeframe j), the set of scene flows ($f_{j_1}, ..., f_{j_m}$), and the set of point clouds ($\mathcal{P}_j, ..., \mathcal{P}_J$), the motion of any given point across a sequence can be estimated. To achieve this, each point is back-projected $\mathbf{P}_k \in \mathcal{P}_j$ to each 2D flow f_i at that specific frame j. We check the sign of the dot product between the camera pointing vector and the normal of the point $\mathbf{P}_k$ to prune any point projections which may otherwise have been occluded for the given view. Using the flow, we can predict the position of the back-projected 2D points $\mathbf{p}_{ik}$ in sequential frames, $\mathbf{p}'_{ik}$.

The set of projected 3D points $\mathcal{P}'_j$, at frame $j+1$, is then acquired by triangulating the flow-projected 2D points $\mathbf{p}'_{ik}$, using the camera parameters of frame $j+1$. This is done by solving a set of overdetermined homogeneous systems in the form of $H\mathbf{P}'_k = \mathbf{0}$, where $\mathbf{P}'_k$ is the estimated 3D point and matrix H is defined by the Direct Linear Transformation algorithm [120]. The reprojection error is minimized using a Gauss-Markov weighted non-linear optimisation [121].

3.2.3 Windowed Hausdorff Filter

The aforementioned point cloud projection framework can now be used to support the coherent merging and noise filtering process. For a given window of width w for frames $\left\{j_{(c-w/2)}...j_c...j_{(c+w/2)}\right\} \subset J$ where c is the center frame, we project the point cloud at each frame towards the center frame using the above method in a recursive manner. In this way structural information is retained and propagated. However, this also has the effect of accumulating any inherent noise within this window. For this reason we extend the two-way Hausdorf filter in [12] with the addition of an energy density term E_{dens}. This density term takes into account the average voxel density of the merged window of frames which is essentially the sum of the propagated clouds. Using density as a conditioning term takes advantage of the stochastic nature of noise in that statistically, persistently occupied space due to noise is far less common than occupancy due to pertinent geometry.

The coherent merged cloud $\mathcal{P}^*_{j+1}$ is given by the logical definition in equation 3.1 where $D_{\mathcal{P}'_j}$ is the summed result of projecting all point clouds within window w recursively toward the center frame j.

Given an ordered array of values $D_{\mathcal{P}'_j}$ such that $D_{\mathcal{P}'_j(k)}$ is the distance from point $\mathcal{P}_j(k)'$ to its indexed match in $\mathcal{P}_{j+1}$. We also define $D_{\mathcal{P}_{j+1}}$ as an array of distances in the direction of $\mathcal{P}_{j+1}$ to $\mathcal{P}'_j$. We then define the merged cloud to be the union of two subsets $M \subset \mathcal{P}'_j$ and $T \subset \mathcal{P}_{j+1}$ such that,

$$\begin{aligned} M \subset \mathcal{P}'_j \,\forall\, \mathcal{P}'_j(k) \;:\; D_{\mathcal{P}'_j(k)} < \; d_j,\; k \in \{1...j\}\,, \\ T \subset \mathcal{P}_{j+1} \,\forall\, \mathcal{P}_{j+1}(k) \;:\; D_{\mathcal{P}_{j+1}}(k) < \; d_j,\; k \in \{1...j\}\,, \\ \mathcal{P}^*_{j+1} = M \cup T \end{aligned} \tag{3.1}$$

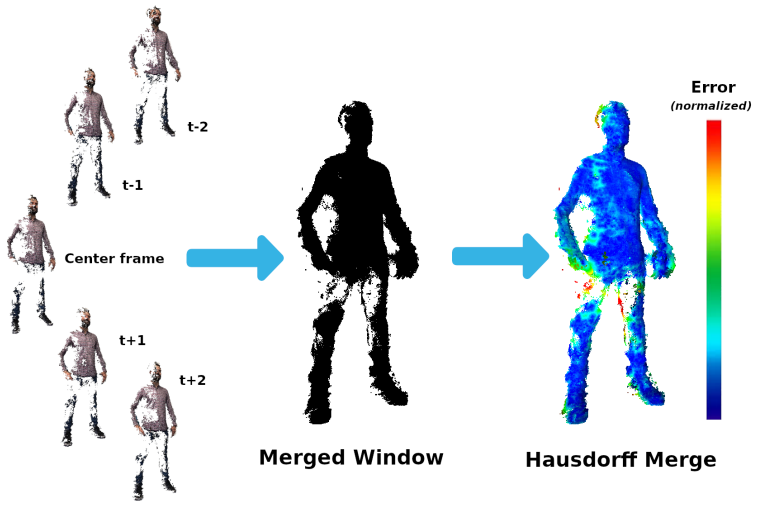

Figure 3.4: **The windowed merge process**. Left: a 5-frame window of input clouds, Middle: the cumulative merge of the upsampled and projected input clouds. Right: the filtered merge process visualized with normalized error given by distance of each point to it's corresponding match in the input cloud. This error term is then augmented with the energy terms $E_{dynamic}$ and E_{dens}.

By this definition, $\mathcal{P}^*_{j+1}$ contains only the points in $\mathcal{P}_{j+1}$ and $\mathcal{P}'_j$ whose distance to their nearest neighbour in the other point cloud is less than the computed threshold d_j. The intention of this design is effectively to remove any large outliers and incoherent points while encouraging consistent and improved point density. Figure 3.4 shows an example of how the coherent merge works.

3.2.4 Dynamic Motion Energy Term

Due to the distance-based nature of the Hausdorff-based filter, it is often observed that fast-moving objects are pruned after being projected into the next frame. This approach to filtering greatly reduces the amount of temporally inconsistent noise, but simultaneously, it over-filters dynamic objects due to the lack of spatial overlap between frames. This is especially true for sequences captured at 30fps or less, which is often the case for affordable VV setups where bandwidth and storage are concerned. To address this issue, we supplement the distance-based threshold term

with a dynamic motion energy which is designed to add bias towards fast-moving objects. This energy term is proportional to the average motion observed across the scene-flow estimates for a given timeframe. For faster-moving objects, higher confidence is assigned to clusters of fast-moving points. Given that $\mathcal{P}'_j$ is a prediction for frame $j+1$, we validate each predicted point by back-projecting $\mathcal{P}'_j$ into the respective scene flow frames for time $j+1$. The flow values for the pixels in each view are then averaged to calculate the motion for a given pixel at that time. As in section 3.2.2 we again filter out occluded points using the dot product of the camera pointing vector and the point normal.

3.2.5 Spatio-Temporal Density Term

The proposed system offers an expansion to the two-way Hausdorff-based filter presented in [12] by sampling a window of frames about the current timestamp. While the two-way filter is robust to temporal noise it isn't capable of recovering large sections of missing geometry over a spanning timeframe. As illustrated in figure 3.6, the two-way approach fails to recover much from the sequence where large patches are missing over a longer time period. To address this, the proposed system introduces a windowed approach which combines the projected information from multiple frames while retaining comparable noise filtering. In order to reduce the added noise we propose an additional energy term for the filtering threshold based on ST density within the given window. The new threshold score criteria is then given by:

$$E_{th} = d - (E_{dens} + E_{dynamic}) \tag{3.2}$$

The E_{dens} term is calculated as follows:

- For a window of width w we iteratively project each frame into the current timestamp such that a single point cloud object is created consisting of the points projected from the frame range $\{t_{(c-w/2)}...t_c...t_{(c+w/2)}\}$

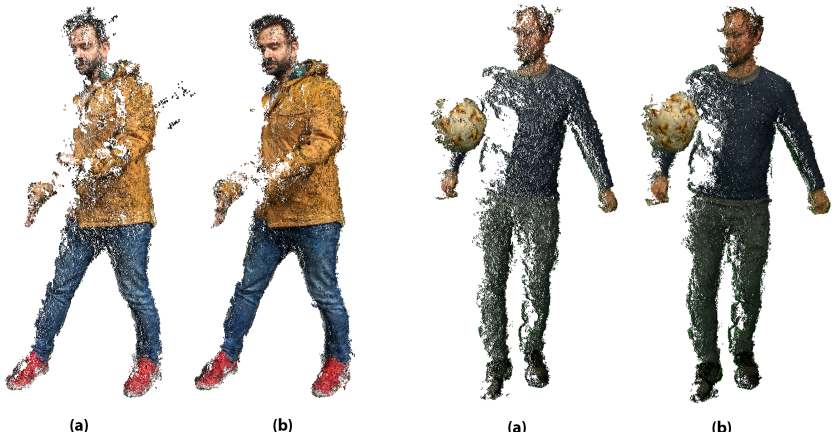

Figure 3.5: An example of the proposed upsampling and filtering system. Pictured left: a sequence captured outdoors with handheld devices. Pictured right: a sequence captured in a low-cost controlled studio environment with fast-moving objects. For both sequences, (a) corresponds to the input cloud prior to filtering while (b) represents the upsampled and filtered result [12].

- An octree-based occupancy grid is then constructed on this object where each leaf is assigned a normalized density score. This score is the E_{dens} term for any point given its index within the occupancy grid.

Figure 3.4 illustrates this process for any given window. The size of this window is variable but is limited by practical limitations of computation time and the trade-off of adding multiple sources of noise. For our purposes we concluded that a window size of $w = 5$ was within practical time constraints while still providing good results. As with any filtering or averaging algorithm, there is an inherent risk of over-smoothing data and thus, such decisions may differ for various sequences depending on the degree of dynamic motion.

3.3 Experiments

In figure 3.5 we demonstrate a side-by-side comparison of the process results vs unprocessed input for two challenging yet conventional scenarios. We evaluate the system on a number of sequences captured outdoors

with as little as 6 to 12 handheld devices (i.e. smartphones, tablets etc..) as well as a controlled green screen environment comprised of 12 high-end, rigidly mounted cameras (6 4K resolution, 6 Full HD). A ground-truth comparison is also presented by comparing reconstruction results against a known synthetic model within a virtual environment with rendered cameras.

3.3.1 Outdoor Handheld Camera Sequences

Shooting outdoors with heterogenous handheld devices can present a number of challenging factors including: non-uniform dynamic backgrounds, increased margin of error for intrinsics and extrinsics calculations, instability of automatic foreground segmentation methods and more. The cumulative effect of these factors results in temporal inconsistncies with the reconstructed point cloud sequence as well as the addition of structured noise and omission of pertinent geometry. Figure 3.5 (left model) shows the difference between using framewise reconstruction (a) and the proposed system (b). A significant portion of structured noise has been removed whilst also managing to fill-in gaps in the subject.

To further demonstrate the impact of our system targeting volumetric reconstruction, we present the effect of applying screened Poisson surface reconstruction (PSR) [107] to the input point cloud. In general, the direct application of PSR creates a fully closed surface which usually creates bulging or ”inflated-looking” surface meshes. Instead we use the input cloud to prune outlying faces from the PSR mesh such that the output surface mesh more accurately represents the captured data. Thus, in figure 3.6 the gaps in the input data can be visualized clearly. This figure also shows the appreciable increase in pertinent surface area after ST upsampling.

3.3.2 Indoor Studio Sequences

In general, sequences shot in controlled studio environments exhibit far less temporal noise and structural inconsistencies in comparison to ”in-the-wild” dynamic outdoor shots. To further test our system we introduce an extra degree of challenge in the form of multiple, fast-moving objects

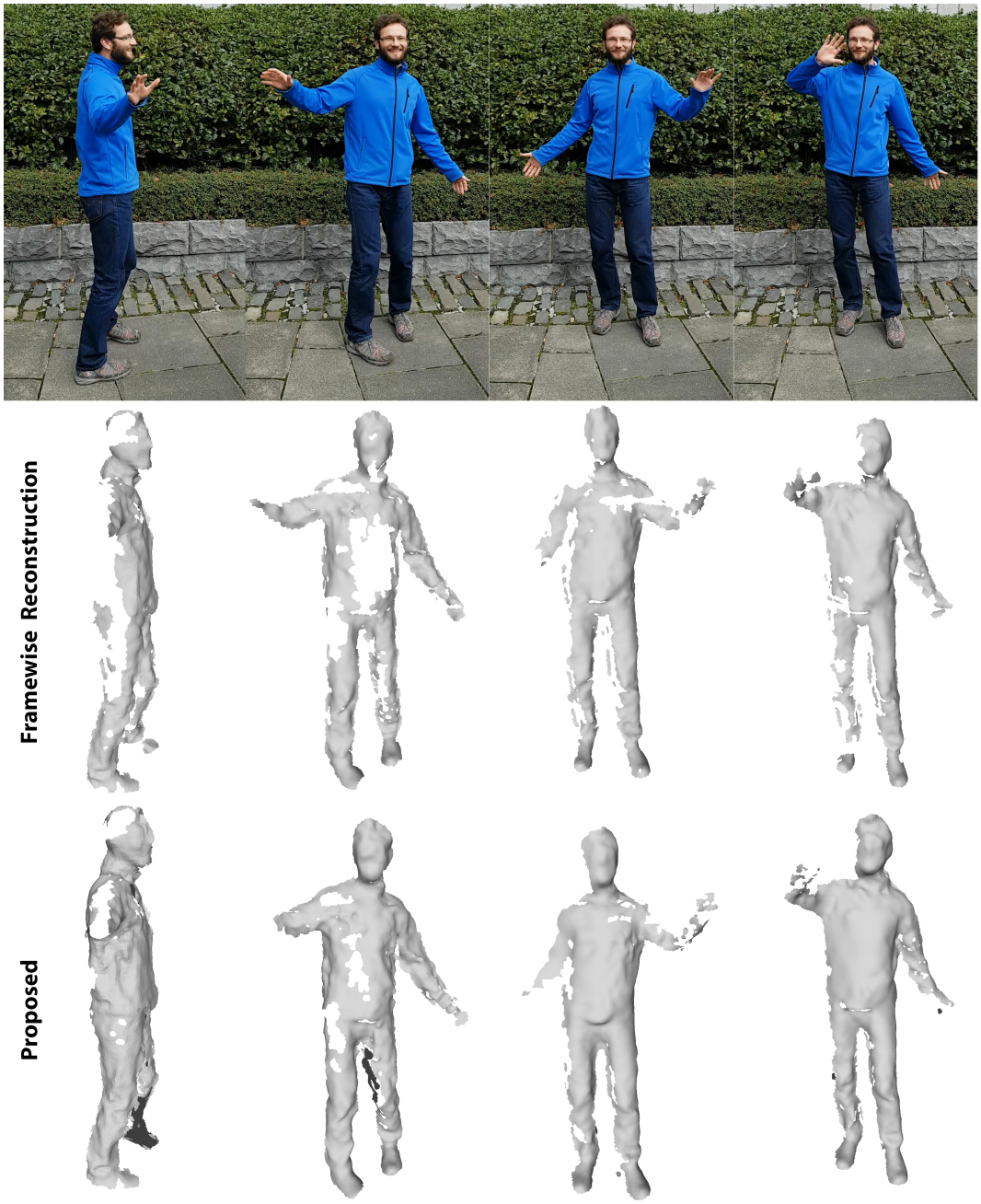

Figure 3.6: A non-sequential set of frames from an outdoor VV shoot using handheld cameras. (Top): The RGB input to the system. (Middle): The result of applying poisson reconstruction to the unprocessed, temporally incoherent point clouds. (Bottom): The same Poisson reconstruction method applied to the upsampled and filtered output of the propsed system [12]

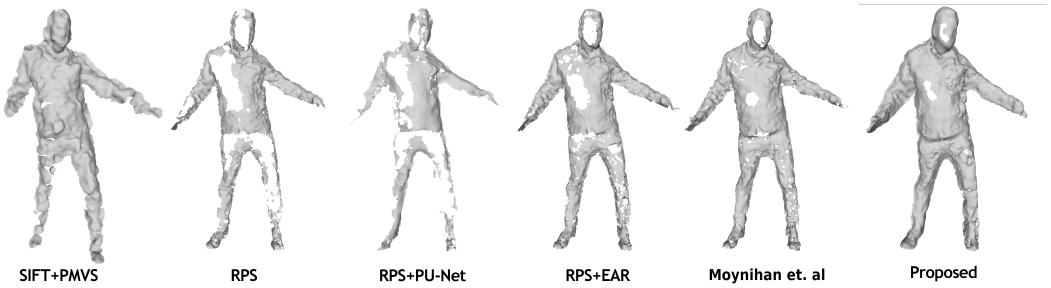

Figure 3.7: A qualitative comparison of surface areas recovered from PSR meshing of point clouds from comparable systems. All meshes were created using the same octree depth for PSR and same distance threshold for outlier removal. From left to right: SIFT+PMVS [122, 123], RPS [3], RPS+PU-Net [106], RPS+EAR [104], Proposed system applied in two-frame, forward direction only[12], the proposed system with windowed temporal filter centered on a window of 5 frames.

while still using no more than 12 cameras for full, 360-degree coverage. This introduces further difficulty due to occlusions caused when the ball passes in front of performer as well as testing the limits of the flow-based projection system. In spite of these challenges, the proposed system is still able to filter a lot of the noise generated and can recover a modest amount of missing geometry, Figure 3.5, (right model).

3.3.3 Synthetic Data Sequences

As a baseline for ground-truth quantitative benchmarking, we evaluate our system using a synthetic virtual scenario. This synthetic data consists of a short sequence featuring a human model performing a simple animated dance within a realistic environment. 12 virtual cameras were evenly spaced around a $180°$ arc centered about the animated character model. The images rendered from these virtual cameras provided the input to the VV systems for testing. Using this data we compare our results with those of temporally incoherent VV systems by applying PSR to the output point clouds and using the Hausdorff distance as an error metric. This is shown in Figure 3.9.

Figure 3.8: An animated character model within a realistic virtual environment to generate synthetic test data [124, 12]

We compare our results against similar framewise point cloud reconstruction systems, SIFT+PMVS [123] and RPS [3] as well as some state of the art upsampling algorithms for which we provide the method of RPS as input; PU-Net [106] and the Edge-Aware Resampling [104] method.

Benchmarking against RPS+EAR also provides a form of ablation study for the effect of the proposed method as this is the approach used to initialize the system.

The proposed system demonstrates an overall improvement in quality in Table 3.2 yet the synthetic dataset lacks the noise which would be inherent to data captured in a real-world scenario. We would expect further improvements in such a scenario where the input error for the framewise reconstruction systems would be higher. Figure 3.7 qualitatively shows the effect of applying the proposed system to much noisier input data.

3.3.4 Flow Initialization

While practically any dense optical flow approach can be used to initialize the STEA filter in section 3.2.1, improvements can be achieved by application-appropriate initialization. We show the results of initializing the STEA filter with CPM against other dense-flow alternatives in table 3.1. The advanced edge-preservation of CPM results in it out-performing the alternatives but comparable results can achieved using GPU-based alternatives which may somewhat trade off accuracy for speed [115].

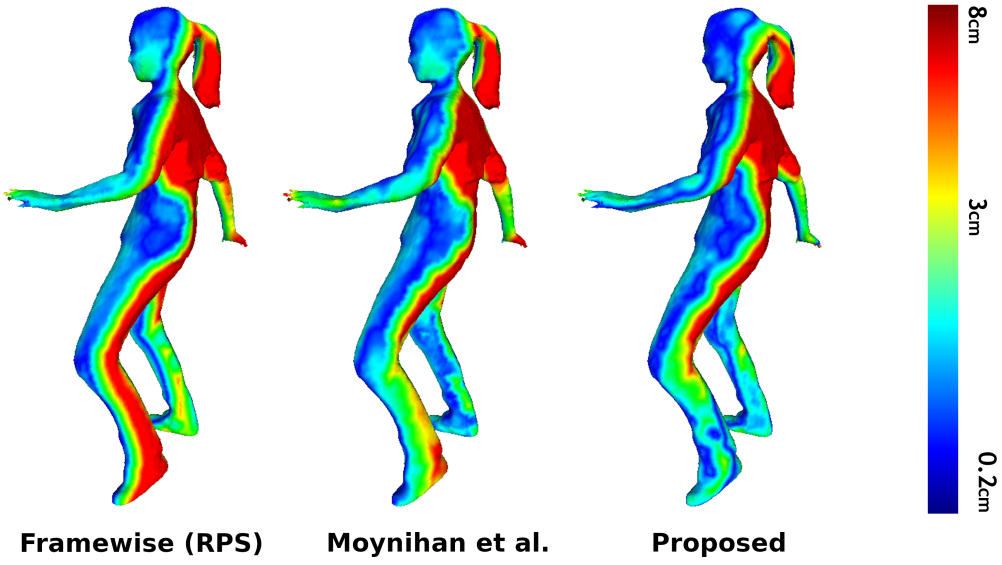

Figure 3.9: Ground-truth evaluation of the proposed system against the virtual reference model using Hausdorff distance as the error metric. The left model shows a frame generated using framewise reconstruction [3], the middle model is the forward-projection, two frame filter [12], while the right shows the proposed systems[12] for a filtering window of 5 frames.

Table 3.2: Synthetic baseline comparison between the proposed method and similar state of the art approaches. Figures represent the Hausdorff distance metric with respect to the bounding box diagonal of the ground truth (%) [12]

Method	Mean Error(%)	RMS Error(%)
SIFT+PMVS	6.18	8.09
RPS	2.17	3.27
RPS + PU-Net	2.44	3.50
RPS + EAR	2.40	3.64
Moynihan *et al.*	1.78	2.72
Proposed	**1.56**	**2.30**

3.4 Conclusions

Throughout this chapter we have demonstrated that the application of ST processes in the form of a self-regulating filter can have significant beneficial effects on early-stage VV production processes. Improvements to quality at this fundamental part of VV pipelines can propagate to subsequent stages yielding higher quality throughout. In particular we demonstrated that this improves the quality of meshes from poisson reconstruction of filtered point clouds .

Due to the temporal nature of the algorithm, it is not possible to directly parallelize the proposed system as the most accurate scene flow is generated by providing the full length of the video sequence. Yet, if parallelism is a necessity, a compromise can be achieved in the form of a keyframe-based system whereby the input timeline is divided in reasonably-sized portions. Future work may employ some automatic keyframe detection which could maximise inter-keyframe similarity.

We demonstrate a qualitative baseline by means of a synthetic reconstruction scenario, however it would be highly beneficial for future benchmarks to establish a common ground truth dataset for evaluation which are often sparse or non-existent within the VV community.

In the next chapter we will present a system which does exactly this for surface mesh sequences in order to apply ST via surface mesh registration.

Chapter 4

Spatio-Temporal Coherence for Mesh Sequences

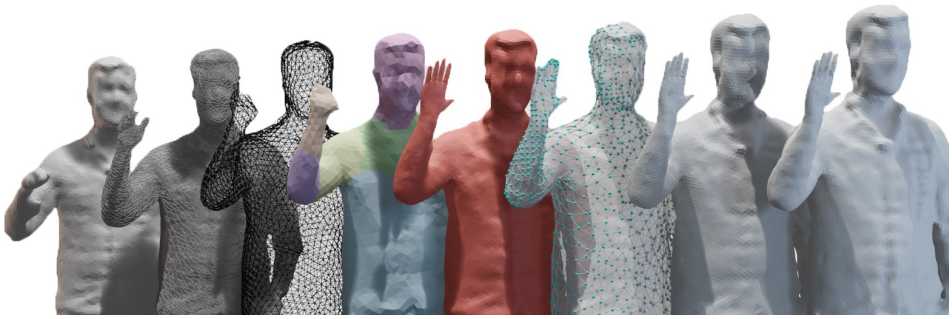

Figure 4.1: We present a robust, autonomous method for tracking volumetric sequences which can detect missing geometry and propagate user edits. Pictured left to right are step-by-step visualizations of the process. The input to our system is a temporally incoherent and noisy sequence of meshes. We perform pairwise registration using abstraction layers, volumetric segmentation and a keyframing system which allows for user edits, e.g. the hand recovered in red. We establish correspondences which maintain edits and propagate geometry throughout a graph-based deformation process.

Previously we discussed how ST analysis could support early stages in VV content creation pipelines for reducing this barrier to entry. In this chapter we support this goal further by presenting a system for robustly and autonomously performing temporally coherent tracking for volumetric

sequences, specifically targeting those from sparse setups or with noisy output. The proposed system can detect and recover missing pertinent geometry across highly incoherent sequences as well as provide users the option of propagating drastic topology edits. In this way, affordable multi-view setups can leverage temporal consistency to reduce processing and compression overheads while also generating more aesthetically pleasing volumetric sequences.

4.1 Motivation

To capture these realistic human performances in VV, one typically needs a multi-camera system that records the performer from different viewpoints, such as the one proposed by Collet *et al*. [5] or Guo *et al*. [10], which uses more than one hundred high-end cameras (including infra-red projectors and cameras) to achieve the best reconstruction possible in a very controlled environment. In these systems, 3D reconstruction algorithms are run on a per-frame basis and the output is a sequence of 3D models (i.e., an independent mesh and texture image per frame).

Some methods address this problem by enforcing temporal coherence in the 3D reconstruction process [13, 12, 97], however, to avoid storing large amounts of data per frame it becomes necessary to apply a mesh tracking algorithm that introduces temporal consistency in the sequence and enables the reuse of a significant amount of data. This compression can be facilitated by keeping the same topology for as long as possible throughout the sequence and updating only the mesh vertex positions. Furthermore, to enable heterogeneous sequences with variations in the mesh geometry and topology, it is necessary to split the sequence into regions controlled by keyframe meshes, similar to methods employed in video encoding. The current state of the art for mesh tracking in this manner works well when consecutive meshes are very similar to each other which is the case for high-end setups; however, they can fail when applied to capture methods which use sparser camera setups [125, 3] or even monocular systems [6, 75] where there is a significant amount of noise, or if geometry is lost (for example a hand or entire limb) due to the challenging capture conditions.

Our proposed approach prioritises generality and scalability by applying temporal coherence to an unstructured series of meshes in a completely autonomous fashion, requiring no system priors, and supporting the challenging conditions presented above. Lastly, our system allows for the recovery of missing geometry and enables the user to introduce geometry edits that can be seamlessly propagated through the sequence. In particular, the proposed system presents the following contributions towards tracking noisy volumetric data from sparse multi-view capture:

- An automatic, similarity-driven keyframe selection process based on spherical harmonics that minimises keyframes and supports varying geometry and topology.
- A volume-based segmentation and registration method for robust tracking of volumetric sequences.
- A tracking system that enables missing geometry recovery and realistic propagation of user edits.

Mesh Tracking. Mesh tracking and registration algorithms, especially when representing the shape and appearance of humans, are an essential part of VV processing pipelines. Such systems use variably dense arrays of RGB and depth cameras to perform per-frame 3D reconstruction [5, 10], while other methods use monocular RGBD sensors [126, 41, 127, 40] and online character template generation [128, 129, 41, 130]. For each of these systems mesh tracking and registration is a fundamental process, ensuring temporal coherence for visual appeal and reduction of data overheads.

The use of a template-driven method helps constrain the problem focus toward reliable pose estimation. With recent developments in monocular 3D pose algorithms [131, 132], similarly, single-camera performance capture systems can produce reliable results [7, 69]. However, even if one was to take pose estimation for granted, the template deformation can still become a challenging task and quite often the approach will be some amalgamation of a customised avatar fitted to a pre-defined parametric model such as SMPL [70]. While the use of a template generally produces robust results, these systems cannot capture dynamic changes in topology without the use of some adaptive surface deformation. Habermann *et al*. [7] present a hybrid of pose-driven template deformation as well as graph-based surface alignment driven by 2D keypoints. While this system is more capable of modelling the dynamic motion of clothing, it is still unable to capture drastic changes in topology which would stray from the input template such as the introduction of new objects or changing clothes.

Some approaches acknowledge this problem and instead opt for the use of an evolving, canonical model which is constructed over the course of

the capture [39, 38, 41]. These methods are well adapted to modelling temporally sensitive, high-frequency details and can faithfully produce temporally coherent models from noisy RGBD data. However, these systems are input-limited to the use of depth sensors which may not be as widely available or scalable as commodity RGB cameras.

For the proposed work we seek to improve content created from scalable studio setups, some of which employ multiple arrays of RGB and infra-red structured light sensors [5, 10] while others present extremely flexible and economical sparse arrays of commodity cameras only [125, 3]. Given a sequence of unstructured meshes generated from such setups, the general approach towards adding temporal coherence is to perform keyframe-based tracking of sequential mesh pairs.

Like many of the previously addressed tracking algorithms, this work also leverages the deformation graph of [46]. The correspondences which guide the deformation in such graph-based approaches are often based on constrained ICP variants [56] or supported by photometric data [133]. Few systems address the scenario of missing geometry [93] and even so, they require strong priors and robust skeleton estimation. In contrast the proposed work requires no priors and doesn't impose any constraints on the mesh topology or number of independent components.

Keyframe Detection. Many sequential tracking systems for unstructured mesh sequences rely on some form of keyframing system in order to select the ideal candidate frames to begin tracking. Collet *et al*. [5] propose a number of heuristics metrics for keyframe selection based on the genus, surface area and number of connected components. These metrics are combined to formulate a *feasibility score* which is used to drive the keyframe selection.This approach is reasonably suited to consistent, high-quality input which would be expected from the system presented in [5]. However, when applied to the highly inconsistent data typical of sparse setups, any metric directly dependant on the input topology becomes uninformative (e.g. the mesh genus can be wrongly represented if the mesh presents numerous small holes).

This same issue is present in the work by Guo *et al*. [10], which solves a discrete Markov Random Field inference problem to minimise the number of

keyframes and reduce artifacts, but relies on the error of a mesh deformation method that takes very detailed and accurate mesh sequences.

Huang *et al*. [93] present a keyframe selection system based on pose variance, however their approach relies on accurate skeleton fitting along with image and silhouette priors. While this approach works well for relatively high-quality data, when applied to the noisy data expected from sparse setups the skeleton-optimization approach becomes unreliable. Furthermore the joint-vertex skinning can suffer where the body shape is obscured by loose clothing.

Budd *et al*. [96] propose a global, non-sequential registration strategy which minimizes the deformation discrepancy. While this approach implies the use of a canonical template for deformation, the application of shape similarity is still very useful for keyframe-based approaches.

Our work opts for an autonomous keyframe system based on shape similarity via spherical harmonics descriptors. By using spherical harmonics as an abstract shape descriptor, a shape similarity map can be built that is robust to frequent and disruptive noise in the input sequence.

4.2 Geometry-Aware Tracking and Editing Framework

We propose a tracking system that applies ST coherence whilst also remaining faithful to the underlying motion and structure of the captured volumetric sequence. This is a challenging task as the input to such a system typically involves a lot of temporal noise, can present high-speed motion and may require demanding shape deformation, especially if the sequences are captured with sparse camera setups. We propose a system which requires no priors other than the input mesh sequence and can be equally evaluated on any VV platform which generates unstructured mesh sequences.

As abundant noise and irregularity can be expected, the proposed method seeks to generate simplistic representations of the input data for some steps of the system via abstraction layers, without the use of model fitting or templates in order to maintain generality. Abstraction layers are

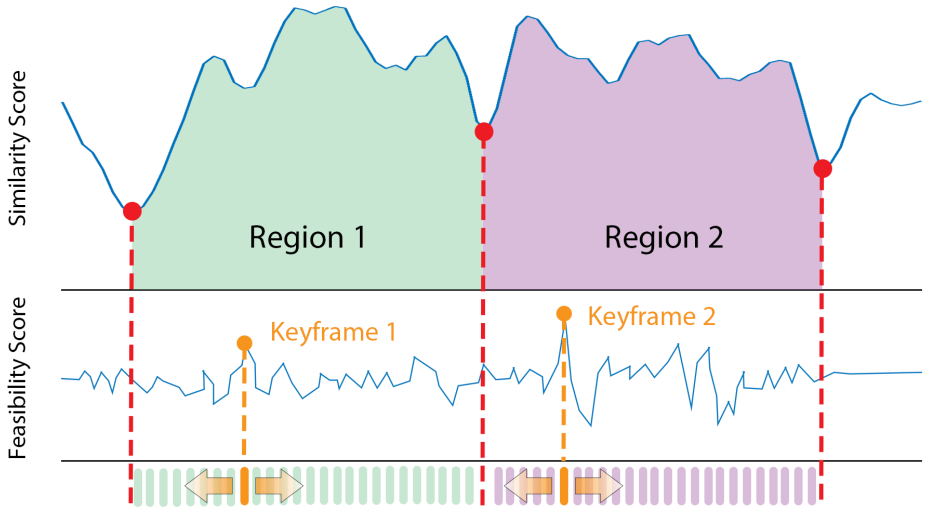

Figure 4.2: Shape similarity descriptors are used to generate a similarity score for each mesh which is used to define tracking regions. Keyframe meshes are selected by using a feasibility score within regions and tracking is then performed sequentially outwards from the keyframe mesh toward region boundaries.

generated by detaching the vertex data from the mesh, filtering outliers and small unconnected components, and applying an adaptive isotropic remeshing [134] which results in a quasi-uniformly distributed set of sample points with sufficient density. This creates an abstraction of the input mesh which supports some key aspects of our system such as the preliminary step of automatic keyframe mesh selection driven by shape-similarity (Section 4.3). They are also used in the following step for establishing dense volumetric correspondences capable of detecting missing geometry and propagating user edits (Section 4.3.1). These correspondences drive a sequential registration by means of a deformation graph (Section 4.3.2). Finally, we apply a post-processing step in the form of a dynamic 3D Kalman filter applied to mesh vertices tracked across a region (Section 4.3.2).

4.3 Similarity-Driven Automatic Keyframe Mesh Selection

The goal of the keyframing system is to simultaneously minimize the cumulative error from sequential tracking and select the minimum number of meshes, N, which can encapsulate the shape and motion represented by an unstructured sequence of meshes, $M_{\{1..T\}}$. With this goal in mind we propose a system which partitions $M_{\{1..T\}}$ into sequential groups based on shape similarity. Thus, given a shape-similarity score for all meshes in the sequence which indicates a per-frame similarity to the other meshes, we infer that highly dissimilar frames will introduce errors when attempting to track back against other meshes in the sequence.

The central metric exercised in this process is the shape-similarity score. In order to establish shape similarity in a computationally effective manner, rotation-invariant descriptors, d_i, are generated for each mesh using the spherical harmonic representation system by Kazhdan *et al*. [135]. With this metric, we compute a similarity matrix among all meshes, $[d_i d_j^\top]_{1 \leq i,j \leq T}$, where the value at $[d_i, d_j]$ is the dot product of d_i and d_j. Mesh similarity score is then defined as the mean value of the matrix per row. To reduce high frequency variance, this one dimensional signal can then be filtered using a moving average filter.

Figure 4.2 illustrates the process further by plotting a typical similarity score overlaid by the determined tracking regions and keyframe meshes determined as above. From these keyframes the framewise registration will be performed outwardly toward region boundaries. By defining region boundaries on frames with low similarity score we effectively isolate the error that would be introduced by attempting to force dissimilar frames to register to adjacent frames. Despite filtering high-frequency variance in the similarity score, we still employ a fixed minimum separation value λ_{min} between selected minima i.e. region boundaries, which maintains a minimum keyframe to frame ratio.

Within each region, a keyframe must be selected which produces the smallest cumulative error when tracked sequentially towards the region boundaries. Collet *et al*. [5] propose a *feasibility score* based on heuris-

tically determined characteristics of the mesh topology, specifically the surface area, genus and number of connected components. For noisy input this score is unreliable and incoherent. Instead we apply the score to abstracted representations of the input meshes which filters out high-frequency topology noise and provides coherent input. We further modify the equation to accommodate the larger impact of genus over surface area on keyframe selection and add a negative weight for region boundary proximity to discourage keyframe selection adjacent to tracking region boundaries.

4.3.1 Dense Volumetric Correspondences

Given a selection of keyframes and defined regions, the tracking process is performed outwardly from the keyframe up to the region boundaries as shown in Figure 4.2. Each pair-wise mesh registration is driven by robust, volumetric correspondences and a topologically coherent deformation graph. We use the abstraction-layer meshes on both the source and the target mesh, as a robust framework for matching reliably significant details. The use of abstraction means that the correspondence accuracy and cost is relatively constant regardless of the size of the input.

The abstraction layers are used as the basis to establish dense pairwise correspondences preserving robustness to missing geometry. This is done by volumetrically segmenting them, and performing a series of alignments from the source layer to the target layer via matching segments. To ensure a reasonable alignment there must be consistent segmentation between the source and target abstraction layers, so we need to segment the former and transfer that same segmentation to the latter.

Previous works have met success by using patch-based registration of meshes [95, 93]. However, these patches are often uniformly distributed or determined by surface geometry.

Our approach follows the idea of a pseudo-semantic segmentation, i.e., creating segments at sharp changes in volume which generally resemble the boundaries of joints and limbs. In comparison with traditional animation rigs, this approach is motivated by the idea that articulated motion tends to be most non-rigid at joints and less so along bones. Thus, we pri-

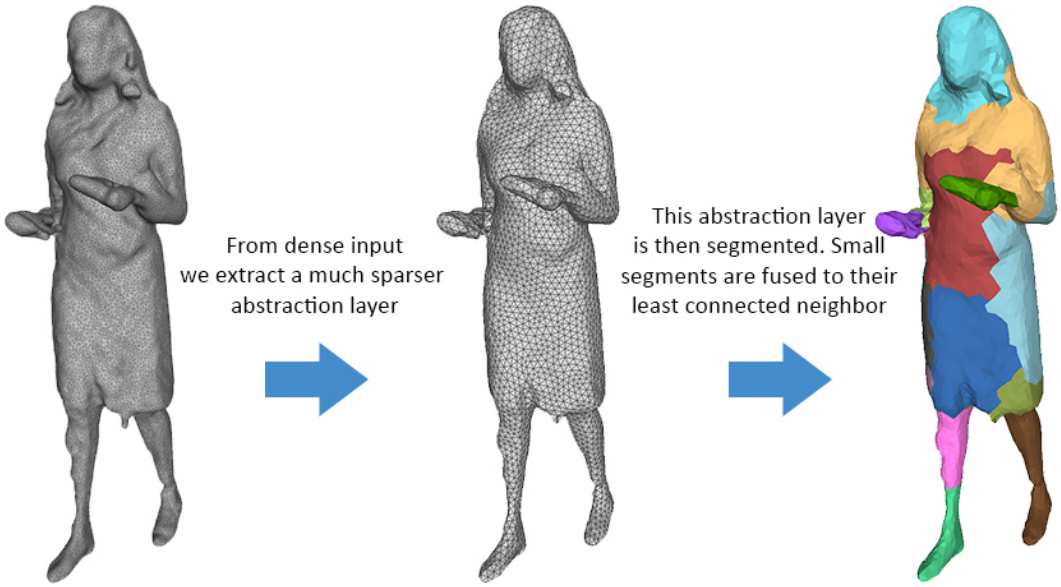

Figure 4.3: The abstraction and segmentation process as a precursor to segment-based alignment. A typical 25K vertex mesh is reduced to 4.5K and segmented.

oritise the semi-rigid parts of the mesh to drive the correspondences. The pseudo-semantic segmentation map is created using the shape diameter function as proposed by Shapira *et al*. [136] and it is organised in a hierarchy from least-connected to most-connected components as a guide for resolving segmentation issues. For example, if the segmentation creates many small components, they are fused to the least-connected neighboring segment. Thus, fusion tends to occur from limb-ends towards the central component. Figure 4.3 shows the abstraction layer creation for a typical mesh and the segmentation result.

A global rigid ICP alignment is performed between the source and target abstraction layers prior to transferring the segmentation of the source layer to the target abstraction layer. Semi-sparse matches between target and segmented source are then calculated using ICP with strict normal alignment tolerance. Typically, in the case of missing geometry (e.g., a limb or other thin structure) there is a large mismatch in segment size. So we perform a coherence check to compare the size of a segment between the source and target abstraction layers and if a mismatch is detected, the segment is flagged to be fused with its nearest connected neighbor. The

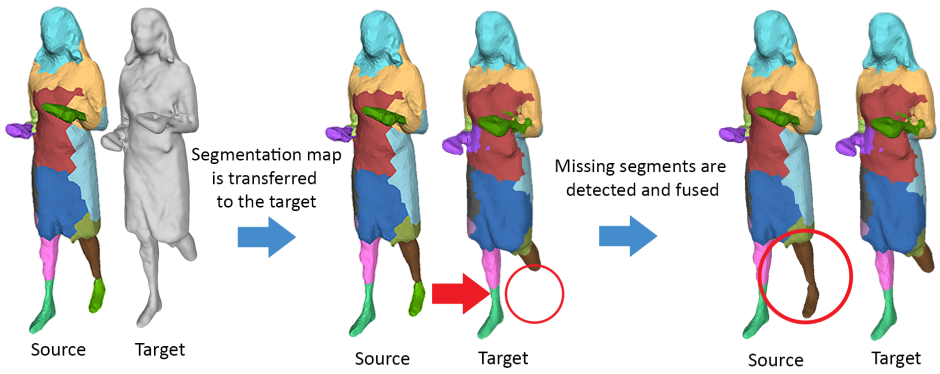

Figure 4.4: Segmentation map is transferred from the source abstraction layer to the target abstraction layer. Any missing segments are fused and flagged for rigid ICP.

flagged segments are recorded and will aligned differently so as to preserve the structure. Figure 4.4 illustrates typical segmentation transfer from source to target.

Once the segment map has been successfully transferred, a segment-wise alignment is performed using an augmented version of the Coherent Point Drift (CPD) algorithm [55], applied to the point cloud represented by the vertices of the meshes. In some cases large segments can be encountered, for example, the central chest region or instances of multiple fused segments. Instead of applying the standard CPD algorithm and encountering performance bottlenecks due to size, we provide the following adaptation to the CPD algorithm which allows for upscaling the alignment that would register two smaller point clouds. This effectively approximates the alignment of a large dataset for the computational cost of a significantly smaller one. If the source and target segment are relatively large clouds S and T respectively, then given some uniformly downsampled clouds s and t, the alignment via standard CPD is given as:

$$s' = s + G_{st}W \tag{4.1}$$

where the aligned cloud s' is calculated as the input cloud plus the affinity matrix G_{st} times a weighted transformation matrix W, which is solved in the main part of the CPD algorithm. Following this calculation, if G_{st}

is replaced by the affinity matrix between s and S i.e. GsS, the alignment can be upscaled to the original size of S by a second application of Equation 4.1:

$$S' = s' + G_{sS}W \tag{4.2}$$

Where W is the same transformation matrix solved for in Equation 4.1. This upscaling naturally simplifies the alignment calculated for W but requires much less computation time. Considering that at a segmentation level the alignment is approximately rigid, so any loss of accuracy due to scaling is negligible. This process is applied to all segments with the exception of those flagged with missing geometry. These segments instead undergo a purely rigid ICP alignment to prevent deforming a segment into a target which is significantly absent. This segment-based alignment of the source abstraction layer to the target abstraction layer can now be used to drive the deformation graph optimization.

4.3.2 Deformation Graph Construction and Application

After the first abstraction layer has been coarsely aligned with the target mesh via segment-based registration, a second layer of abstraction is created from the aligned first layer to assist in generating the structure for the deformation graph which will be used to smoothly reshape the source mesh towards the target. In brief, the deformation graph framework consists of a set of nodes evenly distributed about a mesh with edges connecting regions of influence. Each node n represents a rotation R_j and translation t_j for a set of nodes $n_j = n_1..n_J$. Thus, for any particular mesh M of vertices $v_m \epsilon M$, the transformed vertex $v_m^{'}$ is given by:

$$v_m^{'} = \sum_{n_j \epsilon N_{(v_m)}} w(v_j, n_j) \left[R_j(v_m - n_j) + n_j + t_j\right] \tag{4.3}$$

Where $N_{(v_m)}$ is the set of nodes which influence v_m and $w(v_j, n_j)$ is the skinning weight of a given node towards v_m, following the work of Li *et al.* [56]. The translations and rotations for each node are found by formulating them as a non-linear optimization problem. We model the optimization problem in this work on the cost function of Guo *et al.* [44], driven by the aforementioned correspondences.

Detail Synthesis

Regardless of tracking accuracy, the nature of keyframing will introduce popping artifacts as the topology changes across a region boundary. To address this issue, one could attempt to directly re-align the output topology to the temporally coherent fine details in the input sequence as in [92]. This approach works best when the input noise is relatively small and fine surface details deform slowly. Given that the input to our system may exhibit extremely large perturbations due to noise, this approach will produce incoherent results. Instead we opt for a boundary-blending interpolation technique, analogous to deblocking filters used in decompression [137]. Given region sets of $0 < r \leq R$ containing tracked frames r_t, for timesteps $t \in [0..T]$, we perform a boundary-crossing alignment of the last frame in $(r-1)_{t=T}$ to the first frame in $r_{t=0}$ as if it were a normal pairwise alignment. We then perform a highly non-rigid surface alignment by relaxing the rigidity parameters which creates a detail layer for synthesising surface level details. For each step between the final frame and the keyframe in $(r-1)$ we perform a LERP operation between the detail layer and coarse alignment in order to create a gradient between the deformations. Using cached transformations from the tracking process we can invert and accumulate them as needed to back-project the LERP states to each time step between the last frame in the region and the keyframe. This same process is repeated in the forward direction from $(r-1)$ to r. As this approach is applied directly on top of the registered mesh sequence, it can produce consistent results regardless of noisy surface perturbations in the input raw data. Another advantage of caching the forward registrations is the the reverse application is computationally 'free'.

Sequence Smoothing

Temporal noise may still be observed in the final result despite the smooth nature of the as-rigid-as-possible deformation framework. This noise usually takes the form of high frequency *flickering* of the vertex positions and can be visually unappealing. However, given a sequence of meshes which now share the same topology it becomes possible to filter the vertex positions over time against high frequency noise. To achieve this we apply a standard 3D Kalman filter [138] to the new vertex positions within the

calculated regions treating the keyframe as the initial position and each subsequent frame as a set of observations. The transition matrix used is a simple linear motion model for points in 3D Cartesian coordinates in order to maintain complete generality and avoid introducing constraints via any inherent assumptions of a more complex motion model. Regarding the model parameters, a small process noise Q and larger measurement noise R is used such that $R/Q \approx 1e2$, thus prioritizing smoother motion over observations.

In practice this Kalman filter can inhibit motion over time and lead to noticeably larger popping effects between keyframes. To reduce this we would like the Kalman filter to be most effective when underlying motion is small and to ignore vertices with large per-frame displacement vectors. To address this we perform an offline motion dynamics analysis per vertex and use the displacement deltas to negatively impact the model correction. To this effect we reduce the lag of "genuine motion" and apply the filter in an adaptive manner.

4.4 Experiments

In the following section we validate the proposed method with quantitative, qualitative and ablation studies. We evaluate the keyframe selection metric in comparison to the feasibility score heuristic presented by Collet *et al*. [5]. We also assess the accuracy of the proposed correspondence and deformation framework against the state of the art using numerous challenging sequences, free from temporal noise as a baseline for ground-truth evaluation. Furthermore, we perform qualitative evaluation of several sequences with different levels of noise and artifacts, captured with sparse multi-view setups. Finally, we demonstrate the application of the geometry recovery, edit propagation and smoothing aspects through realistic examples.

4.4.1 Keyframing

To evaluate our proposed method for keyframe selection we illustrate the results of the similarity score compared to the feasibility metric proposed

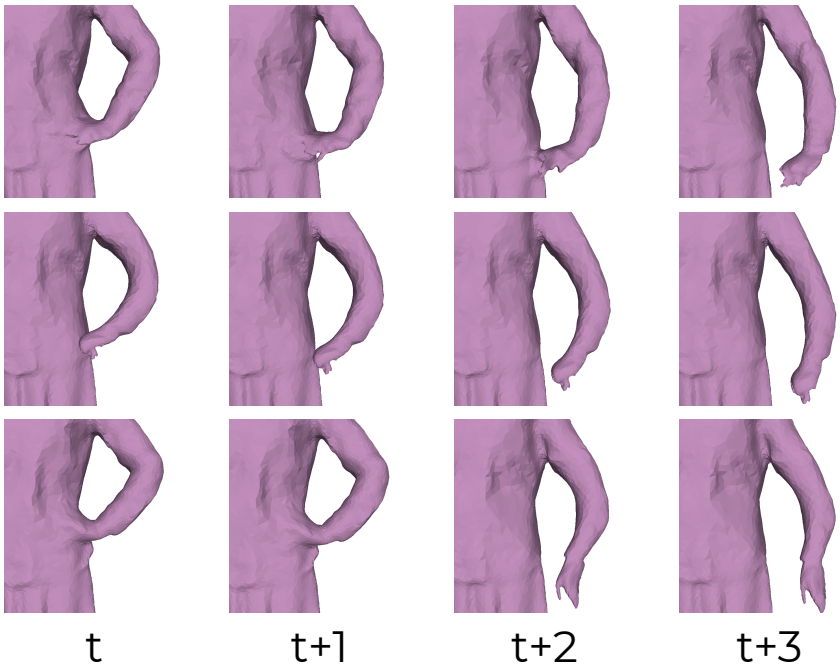

Figure 4.5: Autonomous keyframe selection: (top) input from a sequence featuring many similar topology changes. (mid) proposed algorithm which identifies a keyframe at t>t+3 and tracks from t>3 toward t. (bottom) the system of Collet *et al.* [5] which attempts to resolve the geometry change by stretching before eventually giving up and creating a new keyframe at t=t+2.

	Ours	Collet *et al.* [5]
Max Error	**0.0651**	0.0662
Median Error	**0.0205**	0.0208
# Keyframes	**13**	19

Table 4.1: Keyframe evaluation on twirl sequence containing 170 frames with large topological changes and fast motion. Errors correspond to Hausdorff distance in relative units.

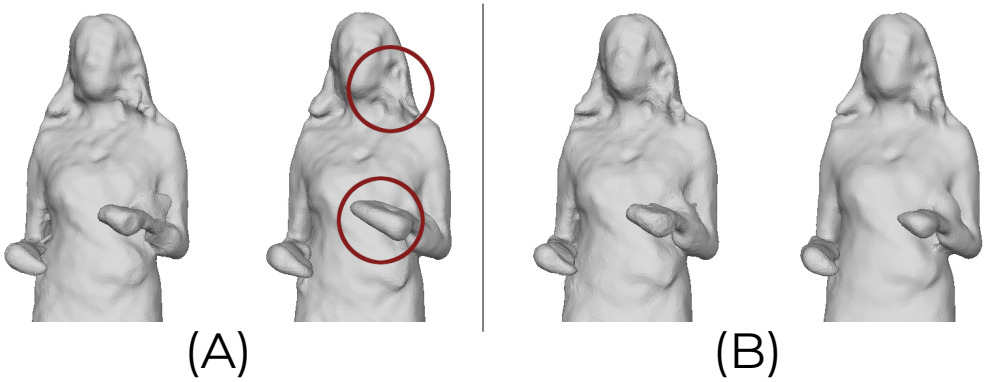

Figure 4.6: Detail Synthesis: (A) and (B) show a topology change where tracking regions meet. (A) uses the temporal detail synthesis of Li *et al.* [92] while (B) is the proposed method.

by [5] when applied to a challenging sequence with drastic topology changes, Figure 4.5. Furthermore, this sequence was captured in a budget studio using 12 RGB cameras and contains a lot of structured noise. We demonstrate the tracking results for this sequence using the proposed keyframe sequence against the greedy-selection algorithm proposed by Collet *et al.* [5]. The proposed approach produces smaller error while significantly reducing the number of keyframes needed, Table 4.1.

4.4.2 Tracking Evaluation

We evaluate the performance of our system against two state of the art approaches which best represent common techniques in surface-based non-rigid registration. The most general of which being Amberg *et al.* [54]

Sequence	Max Error			Median Error		
	Ours	[44]	[54]	Ours	[44]	[54]
Crane	0.0424	**0.3432**	0.3753	**0.0019**	0.0338	0.0278
Jumping	**0.2002**	1.4723	0.3549	**0.0019**	0.0382	0.0145
Bouncing	**0.0891**	0.9982	0.4234	**0.0027**	0.0565	0.0151
Handstand	**0.0054**	0.6450	0.1706	**0.0009**	0.0023	0.0032
Swing	0.2386	0.4298	**0.0813**	**0.0031**	0.0185	0.0074

Table 4.2: Ground-truth evaluation of tracking. Figures are relative to the scale of the input data. Results are given as Maximum Hausdorff Error (max) and Median Hausdorff Error (med).

which is applicable to any type of surface or motion and attempts to iteratively solve vertex positions globally with locally varying ”stiffness”. Lately, however more systems closely resemble that of [44], iteratively solving point-to-plane correspondence driven deformation graphs.

To objectively evaluate the performance of our method we use the dataset from Vlasic *et al*. [139] which features mesh sequences generated by animating a pre-defined template. In this way the input can be considered free from reconstruction artefacts which establishes a reliable reference point for common error metrics like Hausdorff distance [140].

We also present qualitative results of each approach applied to a mix of the above dataset as well as volumetric data captured from multi-view capture setups. Furthermore, we demonstrate the ability of our system to propagate user edits and recover lost geometry by conducting experiments which would replicate some expected user edits or volumetric capture failure modes.

Ground Truth Evaluation

For a fair evaluation of the tracking error introduced by each system, each dataset was given the same keyframes and tracking regions. In this way

the error metric provides a direct indication of the correspondence robustness and deformation fidelity.

The results of Table 4.2 shows that our system introduces fewer errors in multiple ground-truth sequences which exhibit highly dynamic and varying motions.

Qualitative Evaluation

It can be seen from Figure 4.7 that where fast motion is concerned, the proposed system shows robustness in both correspondence matching and large deformation. In contrast to [44] the use of volumetric correspondences over standard normal-constrained ICP methods allows for reliable matching along fast pose changes. The as-rigid-as-possible deformation constraint prevents any large pose changes in [44] despite the likely errors in correspondences resulting in either largely unchanged poses or extreme deformations where the solver struggled to converge. This is evident in (b) for all cases of Figure 4.7. In contrast, the naive global deformation of [54] exhibits very little robustness to bad correspondences and can compress thin structures due to fast motion. This is most clearly seen in the hands and feet in (c) where we see a larger range of motion has led to surface compression due to nearest-neighbour correspondences.

Persistent Geometry Evaluation

We demonstrate the ability of our system to recover and propagate pertinent features in some conventional and challenging sequences captured from multi-view volumetric systems. In particular Figure 4.8 illustrates a sequence which was highly occluded and contained a fast moving football being volleyed. Large sections of the mesh exhibit intermittent missing portions as well as difficulty reconstructing the ball, sometimes across many sequential frames. Our geometry aware system was able to retain important features including the ball, while still registering to the underlying motion. In comparison, template or skeleton-based approaches are simply unable to track foreign objects without manual intervention.

We further illustrate geometry propagation in Figure 4.9 as well as a sample case for user edits. In such a case the reconstruction failed to recover

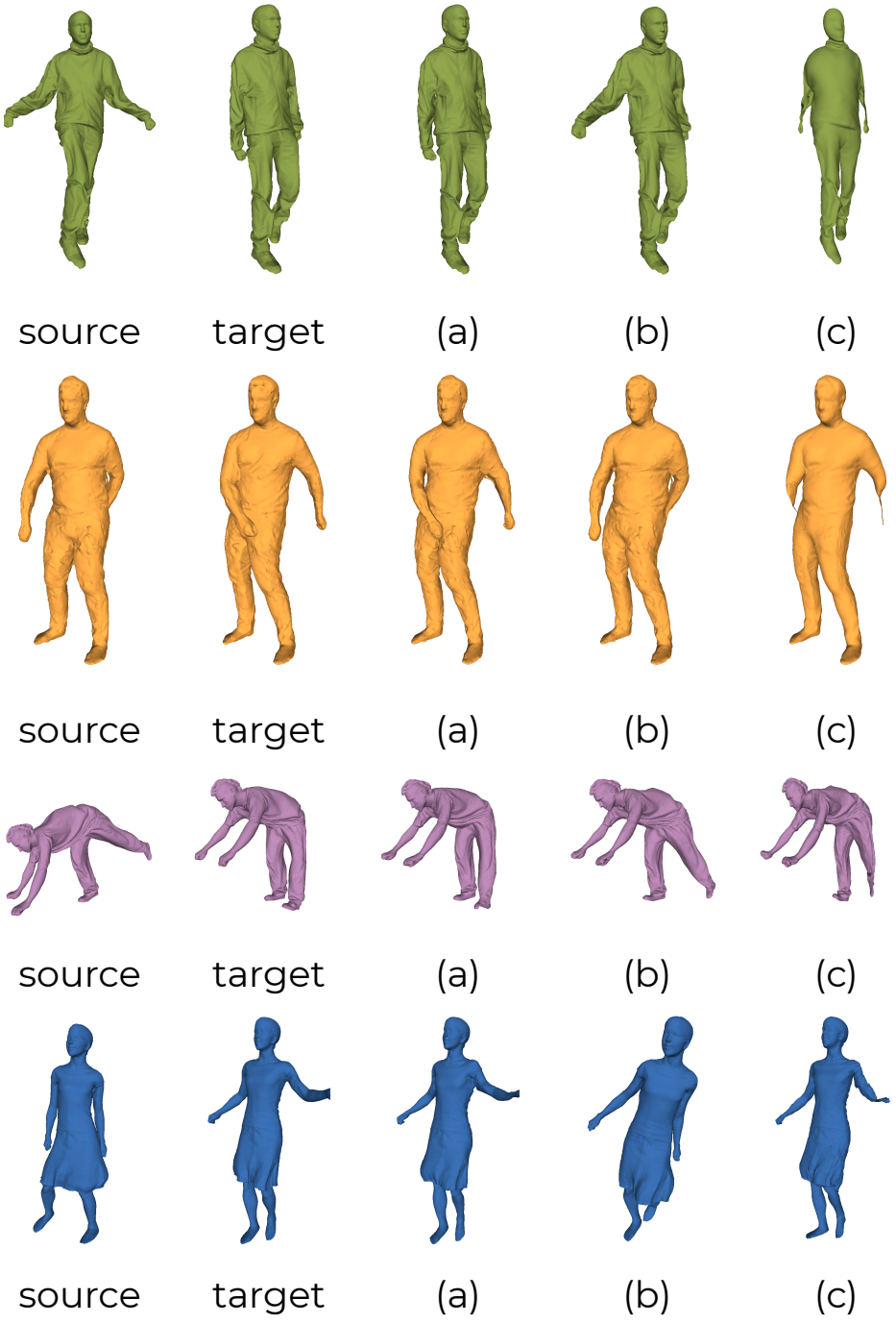

Figure 4.7: Qualitative results of some challenging sequences containing fast motion. Presented for each sequence are: the source, final target, (a): the proposed method, (b): Guo *et al*. [44], (c): Amberg *et al*. [54]. In each case the results are the output of successively tracking the frames between the source and target.

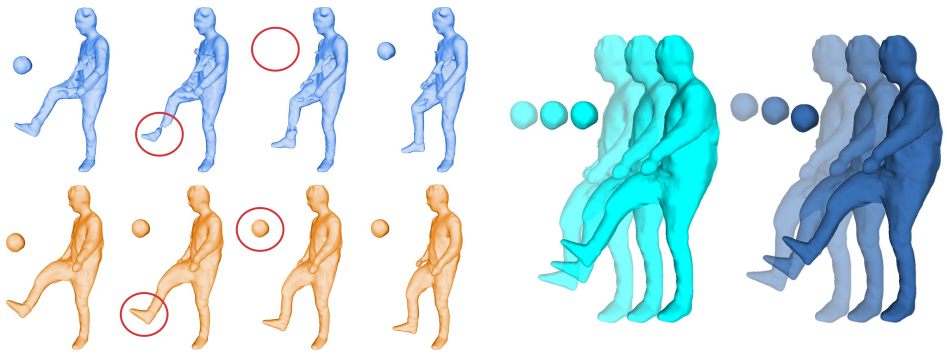

Figure 4.8: Fast moving objects can be lost or cause occlusions (left, top row). The proposed system can track multiple moving objects and provide geometry recovery (left, bottom row). Pictured right (light blue) are 3 successive frames tracked without motion smoothing. Pictured right (dark blue), the same 3 frames where interpolation has occurred as a result of motion smoothing.

the finger detail in the hand of the actor (top right). The user may edit the nearest keyframe(s) and manually restore the data in any 3D modelling software. Afterwards, the system inherently detects the absent geometry through the tracking process and will propagate the edit throughout the frames influenced by the given keyframe. The system is also capable of much larger edits such as the addition of props. The added geometry becomes rigidly tracked along with the nearest connected component and thus it realistically follows the underlying motion while maintaining intact structure.

4.4.3 Detail Synthesis

We compare our detail synthesis approach to that of Li *et al*. [92] which was subsequently used by Guo *et al*. [44] and present the results in Figure 4.6 of a noisy sequence from a sparse camera studio setup. The benefits of the proposed boundary-aware detail synthesis can be seen as a smoother transition across frames while the approach of Li *et al*. [92] produces a sharp boundary transition with large topology changes, resulting in noticeable popping effects. In addition, the proposed method is robust to input noise as it only seeks to smooth tracking region boundaries while the synthesis of Li *et al*. [92] manifests input noise in the hands and hair.

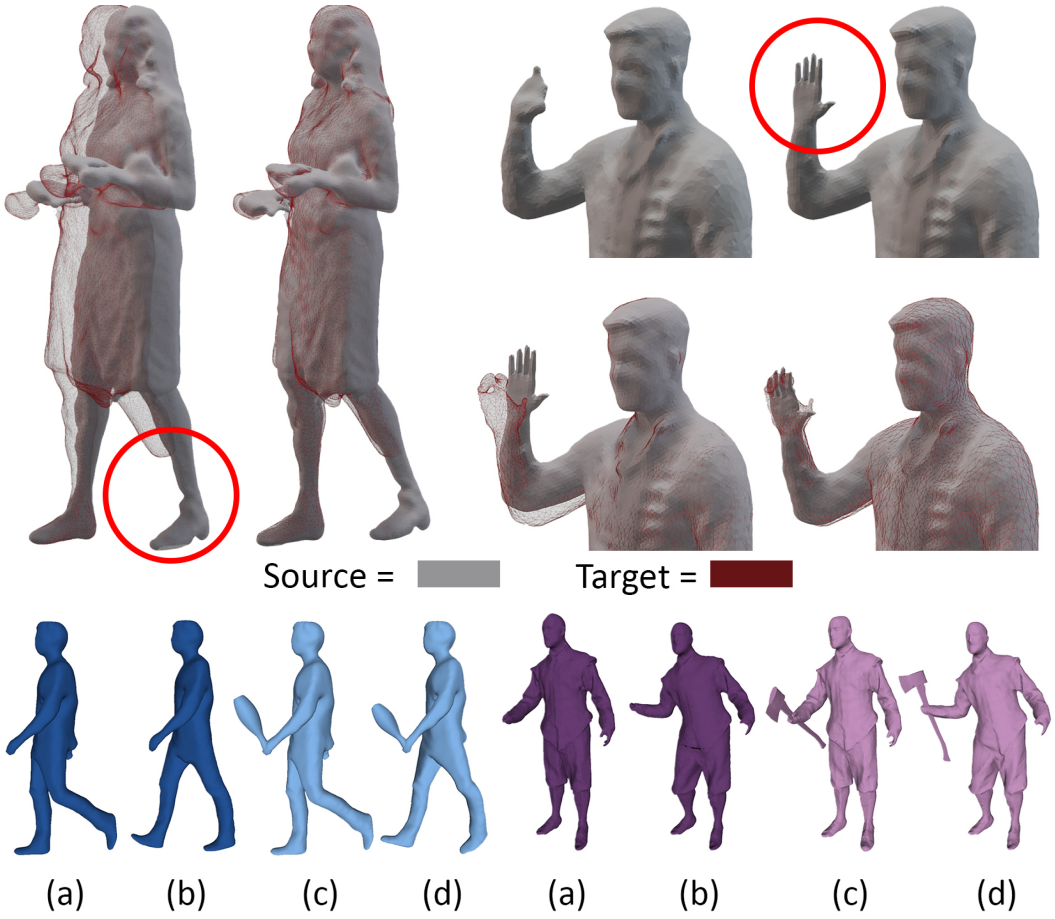

Figure 4.9: Geometry recovery & propagation. Top left: A missing leg is recovered from a walking sequence. Top right: A user manually restores the hand to a keyframe which is then propagated. Bottom: Mesh data from Casas *et al*. [141](blue) and Vologrames [8] (purple), user edits may also be extreme additions such as props.(a) source, (b) target, (c) edited source (d) propagated to target over multiple frames

4.4.4 Smoothing

Smoothing not only helps to reduce high-frequency, *flickering* motions, it also improves the quality of propagated user edits and recovered geometry without the need for expensive 3D flow. Referring back to the recovered fast-moving football in Figure 4.8 (Right, light blue), the motion of the ball becomes static in the recovered frames from having no connected reference segment to propagate to. The smoothing filter helps to interpolate the motion between the static frames and the next observation of the ball. Figure 4.8 (Right, dark blue) illustrates the ablation results where the smoothing process can help interpolate the missing motion. Thus, the smoothing and interpolating motion greatly improves the temporal coherence of the end result.

It is important to note that missing geometry is not only flagged to be excluded by pointwise correspondence matching, but also we ignore the velocity-dependant smoothing for recovered geometry and instead opt for a static covariance noise in the kalman filter in order to allow for motion interpolation of recovered data as is seen in Figure 4.8 of the main paper.

4.5 Conclusions

In this chapter we presented a robust autonomous tracking algorithm which can detect discrepancies in input data and can propagate pertinent geometry. The system outperforms the state of the art for available datasets and requires no priors of the input sequence. Dense volumetric correspondences through shape abstraction provide an indiscriminate shape registration framework which is robust to large or fast motions. Furthermore, our system allows for drastic alterations of the input mesh which can be reliably integrated with the underlying motion, enabling a new domain for creative freedom and post-production.

In conclusion we have demonstrated that the correct application of ST analysis can be used to improve the quality of incoherent VV sequences, provide a degree of post-production alteration and allow for better compression potential.

Chapter 5

A Deep Learning Framework for Volumetric Sequence Registration

Deep learning for shape estimation has seen significant developments in recent years. Latest adaptations of implicit functions to deep learning frameworks now enable efficient and high quality shape estimation even from monocular RGB images. There has even been recent applications of shape estimation towards VV content, that is to say, the estimation of entire shape sequences.

As we have seen in previous chapters, in traditional VV applications it has been commonplace to perform shape analysis and ST registration in order to improve the visual quality as well as generate some temporal redundancy for compression and streaming. The state of the art for learned analysis on surfaces is either limited or constrained to consistent data formats such as RGBD. In this section we present a study on existing architectures for learning on surfaces as well as a proposed pipeline for generalized volumetric sequence registration in a deep learning framework. We will also provide extensive comparisons with the work presented in the prior chapter henceforth referred to as *"AutoTracker"*.

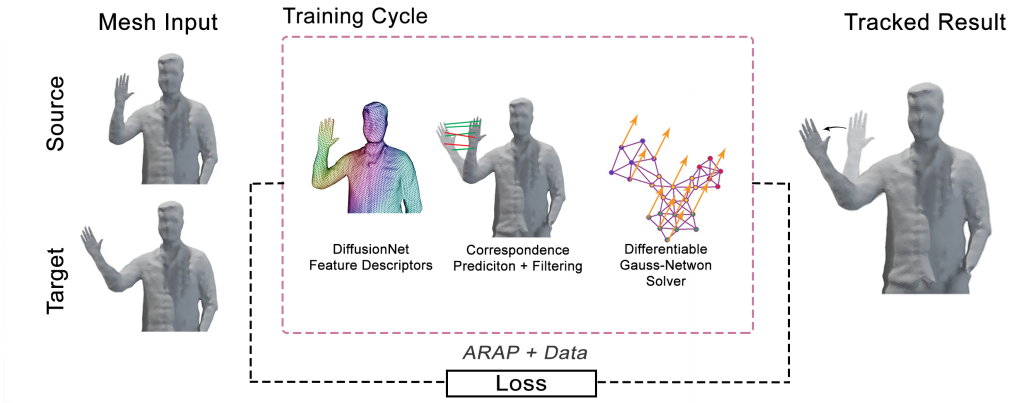

Figure 5.1: An overview of the proposed system: Given as input as surface mesh, we use the robust surface-agnostic feature descriptors of DiffusionNet as part of a labelling classifier to correctly identify correspondences between the input and source template. A differentiable Gauss-Newton solver for 3D deformation is then used to deform the target to the source mesh. Our system is end-to-end optimized to enable training on mesh sequence datasets.

5.1 Motivation

As mentioned in previous chapters, traditional VV content creation can be prohibitively expensive and complicated, requiring large studio spaces with dozens of sensors [5, 10]. Some works have addressed this problem by allowing more budget-friendly, sparse setups [3, 125]. However, the ultimate convenience and ease of access would be exhibited by a VV platform requiring only a single smart-device camera.

With recent advances in learned implicit functions [6, 75], deep-learning for shape estimation has matured to a point that single-camera learning-based frameworks can now be considered for consumer-grade VV content creation. Such single-frame shape estimation frameworks may be applied to video sequences to enable sequential shape estimation but they remain temporally noisy and incoherent. We require temporal coherence in order to provide improvements to visual quality as raw data tends to exhibit temporally incoherent structured noise, largely varying topology and other high frequency visual disturbances that greatly de-

tract from the visual quality of the sequence. Further, as volumetric data has a large footprint, it becomes desirable to try to exploit some temporal redundancies and apply compression or encoding. In particular, as we expect the subject matter to exhibit geometric similarity on a frame by frame basis, it would be ideal to retain a shared topology in so far as possible.

A naive solution could be to mimic some of the success seen in other video-based learning frameworks whereby the architecture itself is temporally receptive. This may be a valid consideration, however the practicality is limited by the exponential increase in computational costs inferred by extending existing architectures into the temporal dimension. In general, it is significantly more practical to explore the domain of learning-based shape analysis to infer properties such as scene-flow [142].

Traditional pipelines for volumetric sequence registration [44, 14] can address this issue but they are often slow and prone to error in replicating challenging scenarios such as evolving topologies and fast motion.

An appealing system would ideally leverage the efficiency and robustness of deep learning frameworks to perform sequential surface registration and minimize some of the failures seen in traditional approaches. In this section we propose a system which consists of a generalizable learning-based mesh tracking and registration framework which aims to improve on the shortcomings of traditional approaches. In this regard we propose a generalized framework for tracking on surfaces under the following novelties:

- Combining the concept of surface agnostic learning with a differentiable optimizer
- A new differentiable optimizer which removes prior constraints on data formats towards more generalizable 3D data

Registration Frameworks

Traditionally, tracking and registration of incoherent mesh sequences had been achieved using deformation-based frameworks largely inspired by the Deformation Graph works of Sumner *et al*. [46] and the As-Rigid-As-

Possible deformation policy of Li *et al*. [92]. These frameworks consist of sequential, correspondence-driven deformations of keyframe meshes within a temporal region [5, 44, 14]. The deformation graph provides a vehicle for experimentation with various registration and deformation policies that have been explored by prior research. The L0-regularization of Guo *et al*. [44] more accurately models the deformation of joints leading to less puckering when deforming clothes or elbows and knees in human shapes. The segment-based correspondence matching of AutoTracker is more robust to correspondence outliers and even provides a degree of editing freedom if one wishes to track geometry edits across keyframes.

In general, many of the expanding works on the deformation graph framework have offered improvements to correspondence matching, keyframing or deformation but as of yet not many works have proposed a learnable deformation framework. Existing works largely rely on some variants of traditional non-rigid ICP for correspondence estimation making them susceptible to outliers and often costly to compute on higher density meshes.

The work of Li *et al*. [143] propose a differentiable Gauss-Newton solver to enable a learning-based deformation framework on RGBD inputs. Bozic *et al*. [144] extend this work to propose more robust correspondences with the addition of learnable correspondence weights and again by proposing a learnable deformation graph for reconstruction [86]. These systems produce compelling results on RGBD benchmarks [145] however, they are limited in application by being constrained to RGBD data. This constraint eliminates the use cases of conventional surface meshes and does not exploit connectivity properties of topology representations.

We propose an extension to this learnable deformation framework which releases the constraint on RGBD data input and explores the domain of shape analysis using state of the art networks for learning on surfaces.

Learning-based Surface Analysis

A key component for the proposed system is a network architecture which facilitates learning descriptive features on surface mesh data. To this end,

there have been recent attempts to provide learning frameworks on various types of 3D data. For pointcloud data, there exists many successful approaches stemming from the seminal PointNet [60] and PointNet++ [146], namely PointCNN[147], DGCNN [148] and KPConv [64]. While point-based methods have the advantages of simplicity, practicality and robustness, they can lack the accuracy of approaches which utilize the connectivity of surface meshes. Even more so, these approaches are not well-suited to shape analysis on deformable data. As such, they make a poor choice of feature descriptor for the shape analysis proposed in this work.

In contrast, there exists some works which attend to surface mesh data and provide more suitable intuition as a feature-descriptor candidate. Methods which learn directly on the surface via some local parameterization such as that of Boscaini *et al*. [91] or Wiersma *et al*. [89] can be well suited to modelling deformable data, however, they often come with the cost of expensive geodesic methods such as parallel transport. Thus, such approaches are not often scalable or practical to implement on larger datasets. Furthermore, they tend to heavily rely on consistent mesh structure and don't generalize well to varying topologies.

An alternative to re-defining convolution operations directly on surface components is to exploit the connection between convolution and mesh operations in a computed basis e.g. Laplace-Beltrami or Fourier. The problem with this approach is that it is non-trivial to transfer features learned from one spectral basis to that of another mesh. Thus, the architecture we have chosen as our feature detector, DiffusionNet, is not spectral in nature but it does leverage the spectral acceleration to efficiently evaluate diffusion along a mesh. The DiffusionNet of Sharp *et al*. [63] proposes a learnable diffusion method for propagating rotation-invariant spatial features and has been shown to perform well on surface-agnostic tasks.

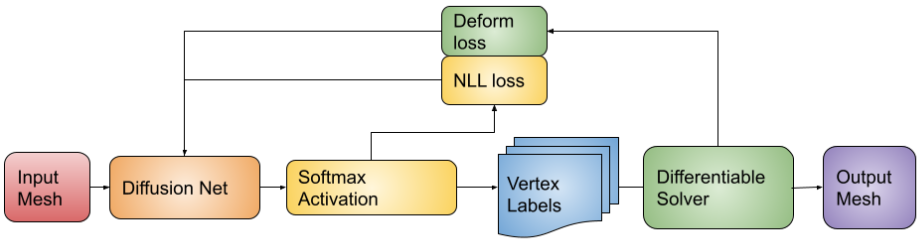

Figure 5.2: An Overview of the system architecture showing the flow of data. As input we take the target mesh and compute the DiffusionNet features at a per-vertex level. Following a softmax activation function we assign per-vertex labels to identify correspondences. These correspondences drive the deformation solver which reduces the energy function representing the disparity between the shapes. Negative Log Likelihood (NLL) is used to evaluate the correspondence labels while the deform loss evaluates the overall deformation quality.

5.2 A Generalizable Deep Learning Framework for Volumetric Sequence Registration

Non-Rigid Deformation Notation

The non-rigid registration task can be viewed as the estimation of a warp field $\mathcal{F} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ which minimizes the nearest point-to-point error between a source and target shape. For our task, we seek to estimate the deformation of a common data format as opposed to device-limited formats like RGBD. As such, we define the source and target shapes as meshes consisting of vertices and faces given by $S\{S_v, S_f\}$ and $D\{D_v, D_f\}$ respectively. In order to perform a robust deformation of the source mesh S we use an embedded deformation graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ is a set of nodes with positions $\mathbf{v_i} \in \mathbb{R}^3$ and $\mathcal{E}$ is the set of edges which define the node-graph connectivity as defined in the work of Sumner *et al.* [46].

Using this embedded framework, we must estimate for every node in $\mathcal{V}$ a translation vector $\mathbf{t_{v_i}} \in \mathbb{R}^3$ and rotation matrix $\mathbf{R_{v_i}} \in \mathbb{R}^{3\times3}$. The graph motion is then denoted as $\mathcal{T} = (\mathbf{t_{v_1}}, \mathbf{R_{v_1}}, \ldots, \mathbf{t_{v_i}}, \mathbf{R_{v_i}}) \in \mathbb{R}^{N\times9}$ for N number of nodes. Thus, the deformation of a given point s_i in S is given by the

following equation:

$$s_i^{'} = \sum_{v_i \in \mathcal{V}} w\left(s_i, v_i\right)\left[R_{v_i}\left(s_i - v_i\right) + v_i + t_i\right] \tag{5.1}$$

where $w\left(s_i, v_i\right)$ is the influence or skinning weight of the node v_i toward s and are calculated as in [41]. The use of a skinning-weights approach allows for the deformation of each point to be an weighted interpolation of a given number of neighbouring graph nodes. The deformation graph is driven by a correspondence map $\mathcal{C}$ as estimated by a neural network Φ with parameters ϕ.

Deformation Estimation

To estimate $\mathbf{t_{v_i}}$ and $\mathbf{R_{v_i}}$ for each node we seek to minimize the following energy term:

$$E_{total} = \lambda_{rigid} E_{rigid} + \lambda_{smooth} E_{smooth} + \lambda_{data} E_{data} \tag{5.2}$$

The components E_{rigid} and E_{smooth} seek to ensure that the deformation is as-rigid-as-possible and that the variance between nodes is minimal respectively. They are formulated as in AutoTracker and [44]. We expand further on these in Appendix B.2.2 to avoid repetition. The E_{data} term is the main component which motivates the deformation and derives from the point-to-point distances of the correspondence estimates from our neural network.

Minimising the total energy across each node for $\mathbf{t_{v_i}}$ and $\mathbf{R_{v_i}}$ forms a non-linear system which can be solved via Gauss-Newton minimization. Thus, our framework comprises of two main components:

- A surface-agnostic correspondence estimator to produce a correspondence map driving the deformation graph framework
- A differentiable Gauss Newton solver that allows the system to be optimized, producing robust correspondence maps that minimize deformation error.

We present an overview of this framework in figure 5.2.

5.2.1 Surface Agnostic Feature Descriptors

In order for the proposed system to be generalizable i.e. independent of input data structure, it must be able to learn robust, descriptive features without restrictive, data-dependant mechanisms. The DiffusionNet architecture of Sharp *et al*. [63] learns descriptive, rotation-invariant features for any shape on which a spectral basis can be derived. The Laplacian operator is most often used as it can be sufficiently descriptive with a relatively low number of eigenvectors and can be generalized to common 3D formats such as meshes and point clouds. For each point on the input shape a *pointwise perceptron* is used to transform the input features into a pointwise function $f : \mathbb{R}^D \mapsto \mathbb{R}^D$ for D input scalar features. While these pointwise multi-layer perceptrons may define arbitrary functions at vertex level, they cannot encapsulate spatial information. Another advantage of this architecture is that it propagates learned features using a learnable diffusion time-step parameter. This parameter is modelled from the classical heat equation,

$$\frac{d}{dt}u_t = \Delta u_t \tag{5.3}$$

where Δ is the *Laplace-Beltrami* operator. Initially, for when $t = 0$, the H_t heat operator is the identity map, however as $t \to \infty$ it tends towards the domain average. By allowing t as a learnable parameter it enables the network to learn spatial influence for a given learned feature. In this way it can also generalize the concept of a receptive field to 3D data formats without the need for restrictive message-passing operations as seen in typical graph nets.

For our task we implement the DiffusionNet architecture as a classifier network given the task of assigning vertex labels. This is done by scaling the last linear layer up to the number of output vertex labels and applying a softmax activation function. As it often occurs in template-based registration tasks, if $D_v > S_v$ we filter out duplicate labels. We show in our experiments that this approach is able to learn robust correspondences across multiple datasets.

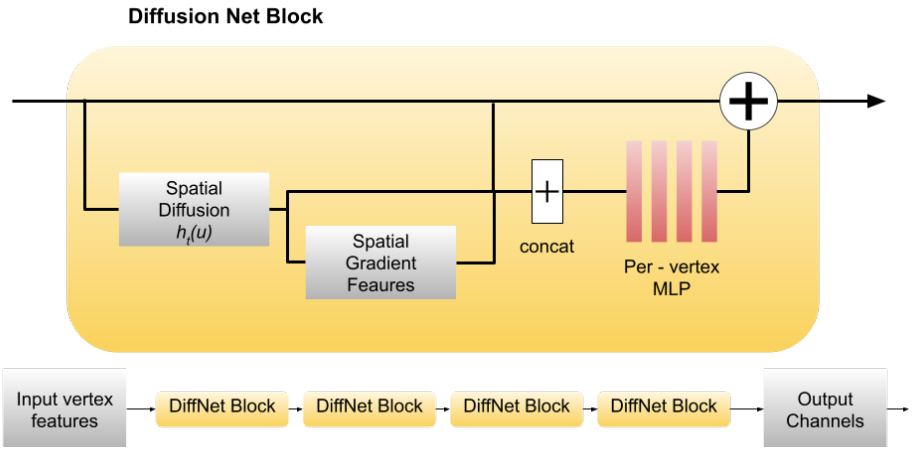

Figure 5.3: A simplified illustration of the DiffusionNet Architecture [63]

A Differentiable Solver for 3D data

Given correspondences between the source and target mesh $\mathcal{C}$, we require a deformation framework to warp the source mesh in such a way as to minimize the aforementioned energy term in equation 5.2. In order to do this within a learning context the deformation solver must be differentiable. We define the differentiable solver Ω as follows:

$$\Omega : \mathbb{R}^{N\times 3} \rightarrow \mathbb{R}^{N\times 9}, (\mathcal{C}, \mathcal{V}) \mapsto \Omega(\mathcal{C}, \mathcal{V}) = \mathcal{T} \tag{5.4}$$

That is, we define the solver Ω to estimate the motion vector $\mathcal{T}$ of graph $\mathcal{G}$ with N nodes and 3D node positions $\mathcal{V}$. The motion vector $\mathcal{T}_0$ is initialized such that the translation components are zero and the rotation components are the identity matrix. For each iteration n of the solver, we compute the respective Jacobian matrix $\mathbf{J}_n$ and residual energy $\mathbf{r}_n$ for the current $\mathcal{T}_n$ and use them to solve a linear system of the form:

$$\mathbf{J}_n^T \mathbf{J}_n \Delta\mathcal{T} = -\mathbf{J}_n^T \mathbf{r}_n \tag{5.5}$$

We use the pytorch implementation of LU decomposition in order to solve this linear system. The system is solved to compute an increment $\Delta\mathcal{T}$ which updates the current motion vector as $\mathcal{T}_{n+1} = \mathcal{T}_n + \Delta\mathcal{T}$. Most of the computations in this process consist of either matrix-vector or matrix-

matrix multiplications which are easily differentiable. One major exception to traditional Gauss-Netwon solvers must be made in that the number of solver iterations must be fixed. That is, conventionally one would set a convergence condition which is checked upon each iteration of the solver and make an "if/else" evaluation on whether the solver should proceed or return the current $\mathcal{T}_n$. Naturally, this operation is not differentiable so instead a fixed number of 3 solver iterations is allowed as heuristically this was found to offer the best trade-off between computational cost and error minimization. It should also be noted that while the optimizer is fully differentiable, it contains no learnable parameters, thus the learning capacity of the network resides fully within the correspondence estimator.

The differentiable framework algorithm is described below:

Algorithm 1 Gauss-Newton Optimization

$\mathcal{C} \leftarrow \Phi\left(T_v\right)$ ▷ Estimate Correspondence map
function SOLVER($\mathcal{C}, \mathcal{V}$)
  $\mathcal{T} \leftarrow 0$
  for $n \leftarrow 0$ to `max_iter` **do**
    J,$r \leftarrow$ ComputeJacobianandResidual($\mathcal{C}, \mathcal{V}, \mathcal{T}, \mathcal{S}_v, \mathcal{D}_v$)
    $\Delta\mathcal{T} \leftarrow$ LUDecomposition($J^T J\, \Delta\mathcal{T} = -J^T r$) ▷ Solve linear system
    $\mathcal{T} \leftarrow \mathcal{T} + \Delta\mathcal{T}$ ▷ Update motion vector
return your-text

5.2.2 Optimization

Given the parameters ϕ of the model Φ, we use three loss functions combined to optimize ϕ as follows:

$$\arg\min_{\Phi} \sum_{\mathcal{X}_{s,d}} \lambda_{data}\mathcal{L}_{data} + \lambda_{warp}\mathcal{L}_{warp} + \lambda_{graph}\mathcal{L}_{graph} \tag{5.6}$$

Correspondence Loss

We train the system under a two-stage learning curriculum, beginning by establishing good correspondences for only $\mathcal{L}_{data}$ before opening the network to the full deformation pipeline i.e. $\lambda_{warp} = \lambda_{graph} = 0$ and the solver is skipped. That is, we begin by training the correspondences as a

vertex-label classifier using negative log likelihood to motivate the correct label for each target vertex.

Warp Loss

The warp loss is used to minimize the error between the input S deformed by the warp field $\mathcal{T}$ and the input S as transformed by the ground-truth scene-flow $\tilde{\mathcal{S}}$:

$$\mathcal{L}_{warp} = \left\| \mathcal{Q}\left(S, \mathcal{T}\right) - \left(S + \tilde{\mathcal{S}}\right) \right\|_2^2 \tag{5.7}$$

Graph Loss

The graph loss is given as the L_2-loss of the graph node translations $\mathcal{T}_t$ and the ground-truth node translations $\tilde{t}$.

$$\mathcal{L}_{graph} = \left\| \mathcal{T}_t - \tilde{t} \right\|_2^2 \tag{5.8}$$

5.3 Experiments

In order to evaluate the proposed work we provide a series of experiments which demonstrate the performance of the system against the state of the art on common benchmarks.

Implementation

We implement the above framework in PyTorch [149] using the Adam optimizer [150] and a learning rate of 0.001 for all training schemes. Training was performed on a single NVIDIA RTX 2070 GPU.

5.3.1 DFAUST Evaluations

We perform our baseline evaluations on the Dynamic FAUST or "DFAUST" dataset [151]. The dataset consists of 10 minimally-clothed human subjects (5-masculine, 5-feminine) performing ∼10-14 action sequences with varying degrees of dynamic motion. Each subject also varies regarding body mass and exhibits different motion physics. For our evaluations we leave

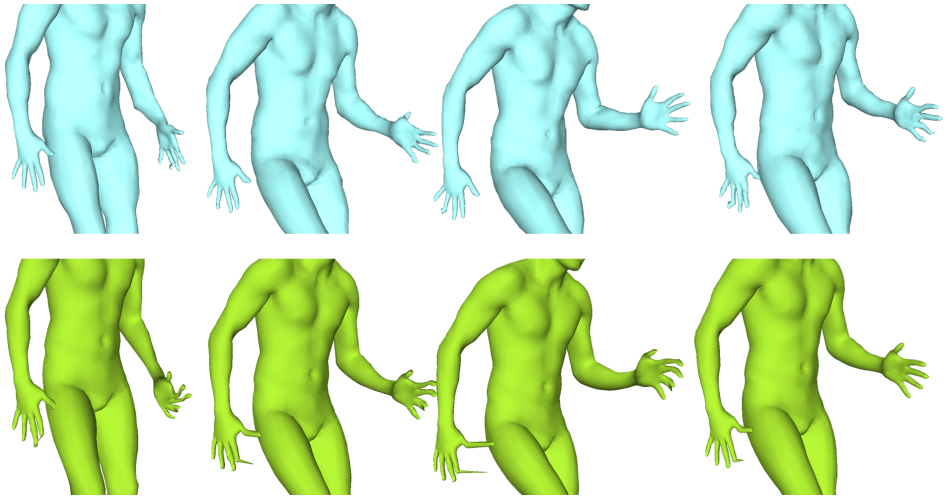

Figure 5.4: Tracking failure during the DFAUST evaluations for subject 50027, sequence "Running On Spot". Above is the proposed system which shows some signs of stress when tracking the fingers, but still manages to recover the motions of the arm. Below is the tracking failure of AutoTracker where the correspondences have failed on some of the extremities causing the process to halt.

out one masculine and one feminine subject for testing, including all of their respective activity sequences.

In table 5.1 we evaluate the accuracy of our approach on four sequences from the DFAUST test set. "Running On Spot" in particular demonstrates fast-paced dynamic motion which likely leads to the tracking failure of AutoTracker for one of the subjects. We show in figure 5.4 an example of this failure illustrating that the high speed motion likely led to correspondence failure due to insufficient overlap between the source and target meshes.

Sequence	Max Error(mm)		Median Error(mm)	
	Ours	AutoTracker	Ours	AutoTracker
50025: Running On Spot	**0.2563**	5.0083	**0.0800**	0.1789
50025: Jiggle On Toes	**0.1053**	0.4083	0.0480	**0.0315**
50027: Running On Spot	**0.3474**	N/A	**0.0738**	N/A
50027: Jiggle On Toes	**0.1492**	0.2915	**0.0405**	0.0419

Table 5.1: State of the art evaluation on DFAUST dataset. Error is calculated as the Hausdorff distance in mm. N/A signifies that tracking failed and the chosen method was unable to complete the sequence.

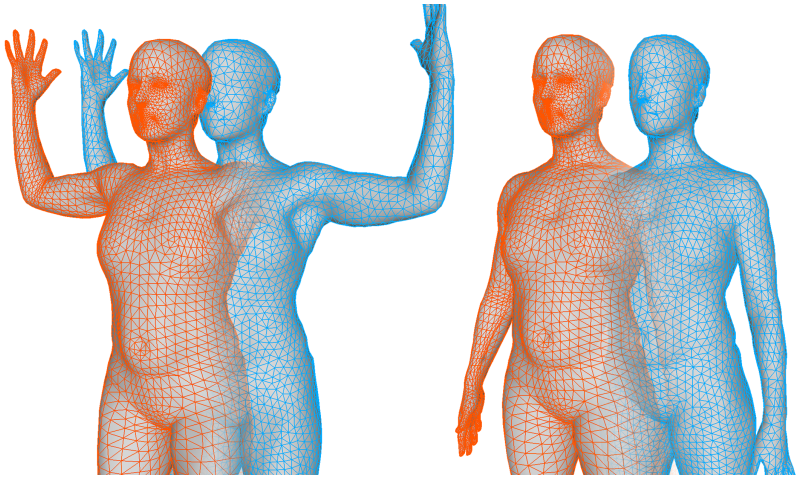

Figure 5.5: This figure highlights the difference between the template and remeshed registration tasks. In orange we see the mesh-graph structure of the template, while blue is the isoptropic remeshing. The two poses are used to show how the structure of the template remains constant in time while the remeshed targets vary.

Task	EPE (mm)	Graph Error (mm)
Template Correspondence	**2.2743**	**3.4976**
Remeshed Correspondence	3.4413	5.5278

Table 5.2: Template registration task vs remeshed target registration. Error is given as the average End-Point-Error(EPE) that is, the average L2 distance measured between ground truth and output after deformation. Graph error is the average L2 distance between predicted graph node translations and ground truth. Averages were calculated over all sequence from subjects 50025 and 50027 of the DFAUST dataset.

The results on DFAUST show that not only is the proposed method more robust to high speed motion, it is also more accurate under most criteria. We see little improvement on the ”Jiggle On Toes” sequences, likely as this sequence doesn’t exhibit much dynamic motion.

In order to demonstrate the ability of the proposed system to generalise beyond learning the mesh-graph structure we show in table 5.2 the results of alternating the learning task from estimating template to template labels and template to remeshed target labels. In previous works, the correspondence task is evaluated such that the input is simply a sequence of meshes derived from the same annotated template, differing

only in vertex positions [152, 65]. It has been suggested that such approaches are over-fitting to the graph structure of the input [63]. For this reason we demonstrate that DiffusionNet's ability to generalize beyond mesh structure is maintained by our deformation framework by observing a minimal increase in error when the target mesh structure has been effectively randomized. In order to randomize the target meshes, we take the template sequences and process them through isotropic remeshing, effectively removing any consistency in mesh structure and presenting a more realistic task.

5.3.2 Vlasic Evaluations

We provide further evaluation against the state of the art on the limited dataset available from Vlasic *et al*. [139]. This dataset consists of 10 sequences in total performed by 3 subjects. In comparison to DFAUST with ∼50K mesh samples, the Vlasic dataset contains ∼3K. Nonetheless, the Vlasic data varies greatly from the DFAUST subjects in that Vlasic subjects are clothed and perform very dynamic motions not necessarily contained to one spot i.e. "Crane" and "March" sequences involve a subject moving about in a circular path. For this test we compare our approach to the state-of-the-art on limited data by training on 8 sequences from Vlasic and presenting the results on the remaining test sequences. We compare our work to that of the previous chapter as well as the state-of-the-art prior. We show in table 5.3 that while our approach does not outperform all state-of-the-art , it produces competitive results given limited training data.

In figure 5.6 we provide a visual example of the proposed method alongside the state-of-the-art . As expected from this challenging dataset we can see that the proposed method shows signs of difficulty in high frequency areas such as the face and wrinkles of the clothes. In comparison we see that Guo *et al*. deforms more naturally but fails to register the arm where motion was fastest. Amberg *et al*. shows severe surface deformation in the same arm as it sacrifices structural integrity in order to resolve the correspondence energy. In this particular instance we see no erroneous deformations from AutoTracker.

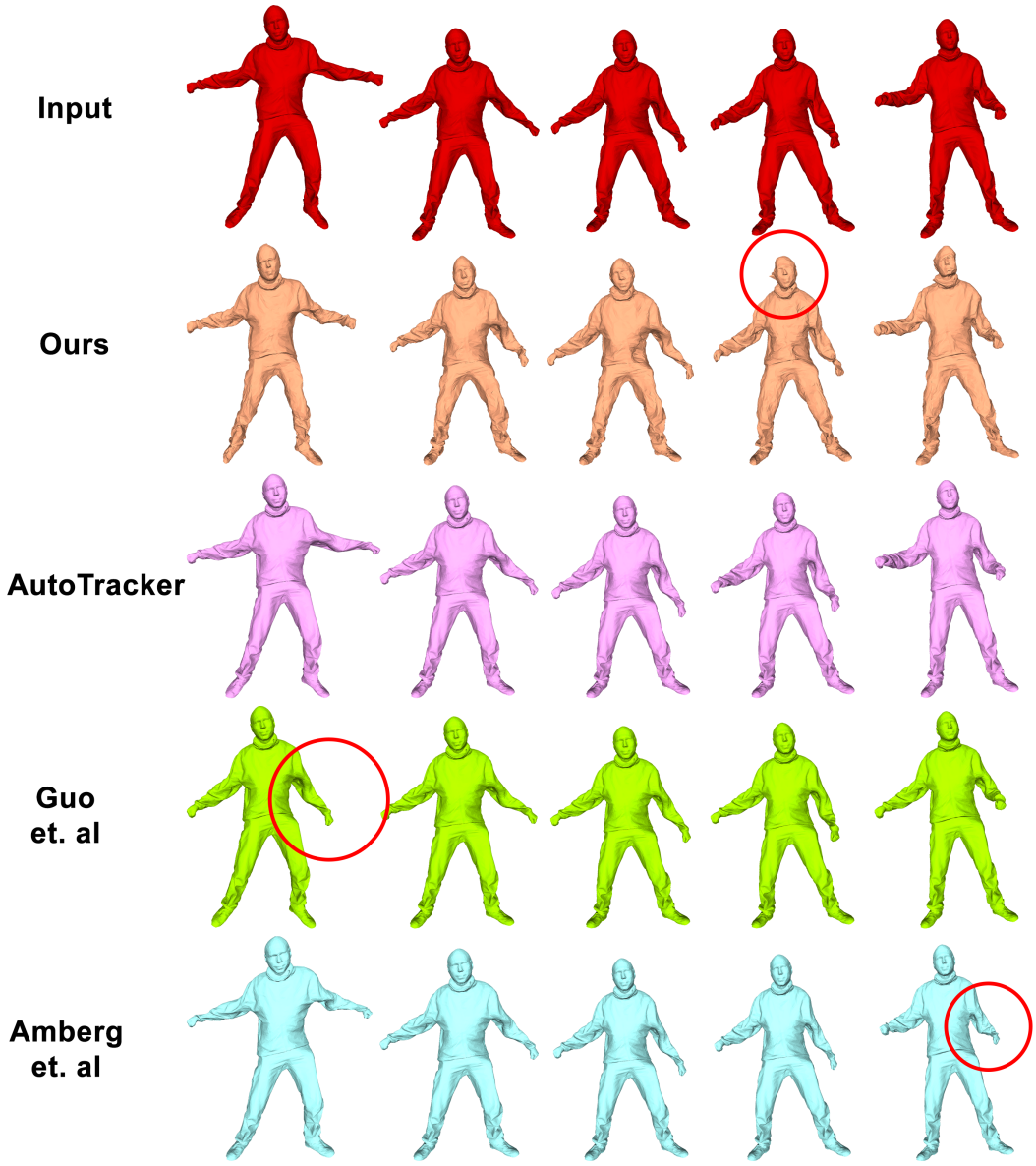

Figure 5.6: Qualitative comparison on Vlasic ”Jumping” sequence. Each row depicts from top to bottom a selection of frames from: The input sequence [139], the proposed system, AutoTracker, Guo *et al*. [44] and Amberg *et al*. [54]. Highlighted in red circles are notable errors. The proposed system shows some difficulty tracking the facial structure. Guo *et al*. show large tracking errors for the arm. Amberg *et al*. shows shrinking errors in the tracked limb.

Sequence	Max Error				Median Error			
	Ours	AutoTracker	[44]	[54]	Ours	AutoTracker	[44]	[54]
Crane	0.9914	0.0424	**0.3432**	0.3753	0.0192	**0.0019**	0.0338	0.0278
Jumping	1.2235	**0.2002**	1.4723	0.3549	0.0302	**0.0019**	0.0382	0.0145

Table 5.3: State of the art evaluation on Vlasic dataset [139]. The error is calculated as the Hausdorff error in relative units.

We also note that inference time for correspondences for the proposed system is <1ms as opposed to 20s for AutoTracker or 5s for [54] on 5k meshes. This significant improvement on correspondence estimation time removes one of the more cumbersome bottlenecks of previous works and demonstrates the type of advantages such deep learning frameworks can produce.

5.4 Conclusions

In this work we presented some of the current challenges in learning on surfaces without data conformity. We show that our system is able to provide a more generalized framework for learning sequential shape registration on meshes and can out-perform the state-of-the-art on large benchmarks. We also demonstrate that it performs in sparse training scenarios, where training data is limited. While this work demonstrates the ability of the proposed system to perform learnable mesh registration on the given datasets, it would be highly beneficial to re-evaluate this work on a dataset which mimics more realistic VV capture data. In general, the meshes from DFAUST and Vlasic, albeit remeshed, are still highly structured and devoid of the noise and artefacts one would expect from real-world captured data. We present the work in this chapter as a step toward a realistic system which could be implemented at such a time when a realistic VV dataset exists to provide a benchmark.

Chapter 6

Conclusion

This brings us to the end of the dissertation. Throughout this dissertation we studied and evaluated the use of ST analysis as applied to a selection of VV content creation contexts. In this chapter we will conclude the dissertation with a summary of the contributions made and how they address the research question. We will then finalize the chapter with a discussion on where future investigations may expand on this work.

6.1 Summary

In Chapter 1, we discussed the motivation behind this dissertation whereby the medium of VV was introduced along with the complications it brings as an emerging platform. With this we presented the motivation for this dissertation, the research question and the structure by which we address it.

In Chapter 2, we reviewed the state of the art for the relevant topics, giving preface to allow latter chapters be explored in depth with sufficient knowledge for insightful discussion. Specifically we looked at early VV content creation, modern VV capture, research trends for affordable systems and their limitations. We followed this with relevant exploration of important works in computer graphics in relation to VV content, namely deformation frameworks shape analysis techniques. We concluded this chapter with a review of 3D deep learning frameworks, their evolution from early challenges adapting 2D networks to 3D data and how recent breakthroughs may enable deep learning for VV content creation.

In Chapter 3, we demonstrated how ST analysis can be applied to early stage VV processes. Specifically we looked at how one could improve point cloud sequences and presented a system for simultaneously filtering and upsampling sparse point clouds. We showed how this could be achieved by leveraging ST edge-aware scene flow along with projected motion filtering to improve the quality of sparse point cloud sequences, leading to improved foundations for typical VV content creation pipelines.

In Chapter 4, we studied ST shape analysis in the graphics domain and how it could be applied to registration and tracking to allow geometry recovery and some post-production editing. We proposed a system which uses ST analysis to provide automatic keyframe selection and temporal detail synthesis. We also expand on existing deformation frameworks to provide robust correspondence matching and evaluate our system against the state-of-the-art .

In Chapter 5, we explored similar ST shape analysis under a deep learning framework. We showed that using deep learning can lead to compelling applications in VV shape analysis and presented a framework that can

be used to learn on generic mesh sequence data. Where previous works have been restricted to RGBD data, we proposed a framework which can learn robust, descriptive features on surface meshes. Furthermore we expanded on a differentiable solver which was then adapted for our mesh sequence data to produce an end-to-end learnable deformation framework for registration of incoherent mesh sequences.

6.2 Outlook and Future Work

Research Question Revisited

Broadly, we sought to investigate — "***How can spatio-temporal processes improve Volumetric Video content creation?***".

In this context, we studied three different applications:

- Filtering and Upsampling Point Cloud Sequences.
- Tracking Mesh Sequences to allow Compression and Editing.
- Learnable Frameworks for Mesh Tracking and Registration.

How well *overall*, did spatio-temporal coherence work within the applications explored?

Throughout the dissertation we presented a number of applications of ST techniques with the key intention of highlighting the benefits for VV content creation. In chapter 3 we showed that the use of ST analysis in the form of a filtering process greatly improves the quality of early-stage VV creation leading to more coherent and visually pleasing meshes. In particular we were able to demonstrate that more significant gains can be achieved for sparse, noisy VV capture setups which in turns reduces the barrier to entry for low-budget creators in this space. From a more conceptual point of view, the relaxation on constraints for demanding resources could potentially free up resources for later stages in VV content pipelines such as additional mesh filtering or improved resolution from using wider camera baselines.

Our findings from chapter 4 support the argument for applied ST techniques by providing a fully autonomous tracking and registration pipeline

for VV mesh sequences. In particular, the benefits of ST analysis were at the fore of the keyframe selection process and the detail synthesis contributions.

Within the context of deep learning we found in chapter 5 that ST analysis is challenging to apply to generalized surfaces meshes however, the proposed system was competitive with the state of the art on small datasets and outperforms on larger ones. This work shows that deep learning frameworks have the capacity for ST analysis on surface meshes and that novel architectures may further exploit this in the future. Overall, in three separate scenarios we see very clear benefits for considering ST processes within volumetric video content creation systems.

Future Work

Throughout this dissertation we have presented some applications of ST to VV content creation. We showed how the mesh tracking and registration concepts of chapter 4 could be expanded with deep learning frameworks in chapter 5. A similar exploration could be done in applying ST upsampling and filtering of pointclouds for VV. While methods already exist for upsampling pointclouds [153, 154] they do so in a single-instance manner. While some methods suggest temporally-receptive learning on dynamic point cloud sequences [155, 156, 157], it would be interesting to see an extension to this which targets upsampling point cloud sequences from image-based reconstruction.

In chapter 4 we demonstrated a mesh sequence registration system which allowed the user some degree of editing capability regarding the topology. In a practical VV content creation setting one would also require some method of inferring new texture data on top of edited geometry. A model which could infer extended texture maps in a ST manner for edited mesh sequences would be useful in this application.

The deep learning framework we propose in chapter 5 demonstrates the capacity for surface-mesh learning architectures to model deformation tasks. While this approach is spatio-temporally consistent by nature of sequential processing, the correspondence inference is performed in a frame-wise manner. The architecture of the descriptor network could

potentially be extended to accommodate or predict more sequential information in a similar way to how this is done in sparse 3D pose estimation methods [158, 159]. Furthermore, while the proposed system performs well on large datasets, we see that it's accuracy suffers with limited data. In order to leverage learned features from multiple datasets it could be worth investigating the use of deep functional maps [88] in place of classifier-style vertex labelling.

Perspectives

It is the hope that this dissertation inspires more long-term work to consider the benefits of a ST approach. Though often limited by the scalability of expanding to the temporal domain, it has been demonstrated throughout this work that the benefits are worth consideration. It would be interesting to see how future algorithms and architectures adapt to further consider sequential 3D as they have for images and video. In particular we saw a natural extension of image-based architectures towards video data during the early days of deep learning research. It would be greatly beneficial to the research community if 3D deep learning architectures continue to grow in a similar fashion and tackle the difficult task of learning on 3D sequences.

With new datasets emerging that specifically target temporal 3D reconstruction and deformation [160, 161, 162], we predict that these areas of research will continue to drive compelling new breakthroughs in years to come. However, as evidenced by need for synthetic data in Chapter 3 and the concluding statement of Chapter 5, it is clear that there is a lack of VV datasets which can provide accurate benchmarks for data captured in a realistic VV capture studio. Especially as research trends further towards learnable solutions, the necessity for realistic ground truth baselines on actual captured data has become incredibly apparent. We have demonstrated the performance of the proposed works in this thesis when applied, for the most part, to synthetic or "clean" data with some qualitative evaluation on real VV data. It is hoped that if a more challenging dataset consisting of realistic, noisy data were created that it could bring such works closer to practical applications in real VV scenarios by means of better testing.

Appendix A

Abbreviations

Short Term	Expanded Term
6DOF	6 Degrees of Freedom
AR	Augmented Reality
DL	Deep Learning
FVV	Free-Viewpoint Video
IR	Infra-red
ICP	Iterative Closest Point
MVS	Multi-view Stereo
NRR	Non-Rigid Registration
RGB	Red, Green, Blue
SfM	Structure from Motion
SfS	Shape from Silhouette
ST	Spatio-Temporal
VR	Virtual Reality
VV	Volumetric Video

Appendix B

Chapter 4, Supplemental Info

This appendix serves as additional material to support the work presented in Chapter 4 including some visual aids and explanation of the mathematical foundations.

B.1 Overview

B.2 Implementation Details

The proposed system was implemented in C++ on Ubuntu 18.04.4 LTS on a laptop with an Intel i7-9750H CPU. On average, our system solves for 20k vertices per mesh in 45s per sequential alignment in comparison to 60s for [44] and 30s for [54]. While the proposed system does not outperform regarding speed, it is still competitive while producing significantly better results as demonstrated in our experiments and video.

B.2.1 Keyframing

We modify the feasibility score of Collet *et al*. [5] as the following:

$$S_i = \sum_{c \in C(i)} \left(1 + 2 * (g_{max} - g_c) + \frac{A_c}{2 * (A_{max} + 1)}\right) * \lambda_i \tag{B.1}$$

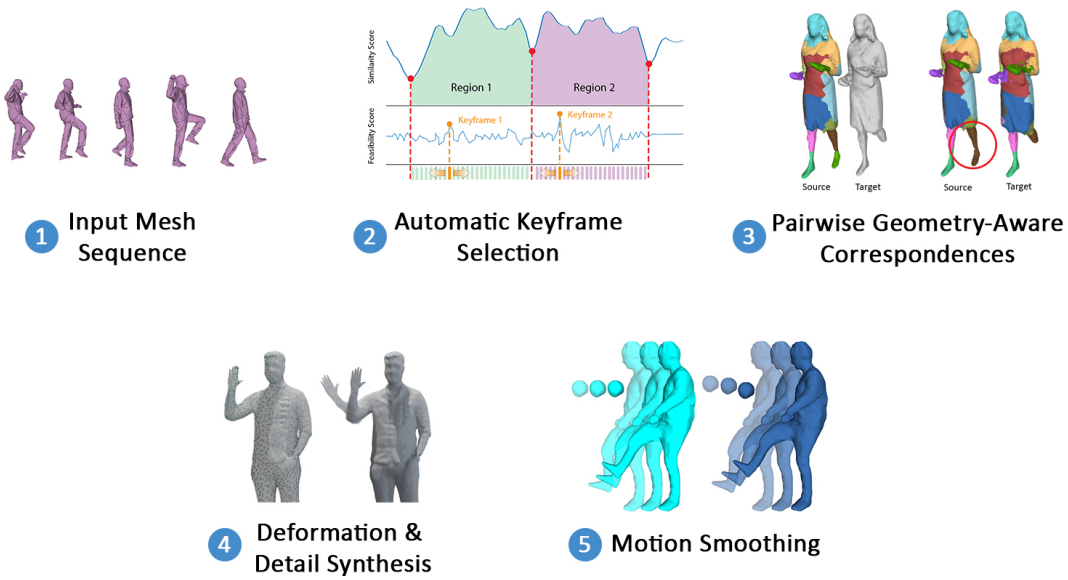

Figure B.1: 1. The input is an incoherent sequence of meshes with independent topology. 2. Using shape similarity and abstraction proxies, a combination of similarity and feasibility scores are used to select keyframes which approximate the optimum selection of frames that will lead to successful pairwise registration across the sequence. 3. Between pairwise registrations, correspondences are found using volumetric segmentation and geometry-aware correspondences which support the recovery of missing geometry and allows for user-defined editing. 4. Using the correspondences, a deformation graph deforms the source mesh to the target. Detail synthesis is then performed to recover high-frequency details and reduce keyframe "popping" effects. 5. 3D motion smoothing is applied to further improve the temporal coherence of the output tracked mesh sequence.

Where S_i is the feasibility score for frame i, C is the number of connected components and λ_i is the boundary weight which discourages keyrames at region edges and is formulated similarly to an activation function where x is the number of frames from the region boundary:

$$\lambda_i = 1 - \frac{1}{1 + x^2} \tag{B.2}$$

Empirically we found that surface area has less impact on the effect of tracking for noisy input data, especially considering that the proposed algorithm is designed to accommodate missing geometry. Genus has a

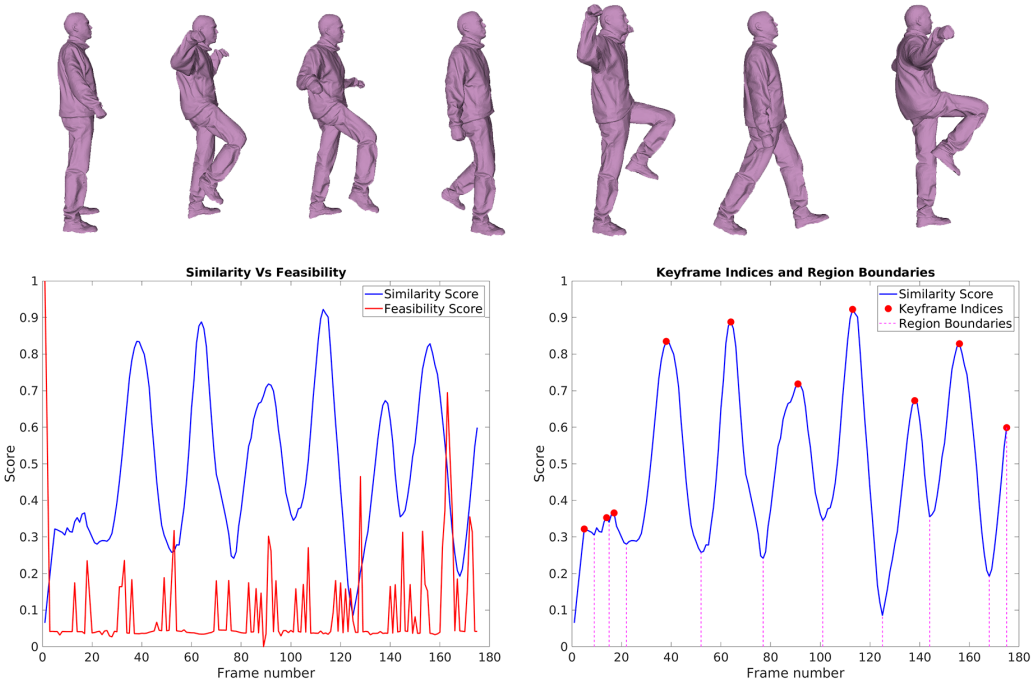

Figure B.2: Analysis of Shape similarty vs Feasibility score. From the Vlasic dataset [139] the above image shows the plot of feasibility score alongside shape similarity for each frame in the sequence (left). After selecting region boundaries via shape similarity, the feasibility score is used to elect keyframe candidates (right).

significantly large impact on the appearance of "chewing gum" stretching artifacts which are still a major concern. We show in figure B.2 a visual analysis of the shape similarity score vs feasibility score. It shows how feasibility score alone is not particularly informative yet combined with similarity regions it can be used to isolate candidates from a region.

Guo *et al*. [44] use an L0-regularization to determine anchor frames, however, attempts to replicate this on real studio data failed as their system requires a template mesh to initialize as well as relatively noise-free target meshes for tracking. For these reasons we were unable to provide a similar analysis against their approach.

B.2.2 Correspondence Conditioning and Alignment

In this section we provide extra implementation details about the correspondence estimation process. For segmentation transfer we initialize the process with global rigid alignment followed by a two-way ICP match with normal-constrained alignment. In the event that some vertices in the target mesh do not have a segment match with the moving mesh we use a k-connected (k=2) region with a majority voting system to assign values to the highly sparse, unmatched vertices.

We also provide an extended description of the key data terms in the deformation graph equation. Equation B.3, describes the main minimization equation as in Guo *et al.* [44]

$$E_{total} = E_{data} + \alpha_{rigid} E_{rigid} + \alpha_{smooth} E_{smooth} \tag{B.3}$$

Where E_{data} is the data term which expands to describe the point-to-point and point-to-plane correspondence error for a vertex v_j which has a matching vertex in the set of correspondences C:

$$E_{data} = \sum_{v_j \epsilon C} \alpha_{point} \left\| v_j' - c_j \right\|_2^2 + \alpha_{plane} \left| n_{c_j}^T \left(v_j' - c_j \right) \right|^2 \tag{B.4}$$

The E_{rigid} term encourages as-rigid-as-possible deformation and is constructed as:

$$\begin{aligned} E_{rigid} = \sum_j ((a_{j1}^T a_{j2})^2 + (a_{j2}^T a_{j3})^2 + (a_{j3}^T a_{j1})^2 + \\ (1 - a_{j1}^T a_{j1})^2 + (1 - a_{j2}^T a_{j2})^2 + (1 - a_{j3}^T a_{j3})^2) \end{aligned} \tag{B.5}$$

Where a_{j1}, a_{j2}, a_{j3} are the column vectors of R_j. The final term, E_{smooth} penalises abrupt variance between adjacent nodes and is given by:

$$E_{smooth} = \sum_{n_j} \sum_{n_i \epsilon N_{(n_j)}} w(n_i, n_j) \left\| R_i(n_i - n_j) + n_j + t_j - (n_i + t_i) \right\|_2^2 \tag{B.6}$$

This is formulated in our Gauss-Newton solver which is built using the

Eigen[1] libraries and CHOLMOD[2] for Supernodal Sparse Cholesky Factorization and converges in less than 5 iterations under the criteria that $\Delta E_{total} < 1e-6$ between successive iterations. We use ≈ 2.5K nodes and ≈ 3K constraints on each deformation. We use $\alpha_{rigid} = 500$, $\alpha_{smooth} = 500$, $\alpha_{point} = 0.1$ and $\alpha_{plane} = 1.0$ similar to those values as recommended by Guo *et al.* [44].

B.2.3 Detail Synthesis

We extended the description of detail synthesis in the text with figure B.3 which provides a more visual guide to the process.

[1] http://eigen.tuxfamily.org/
[2] https://developer.nvidia.com/cholmod

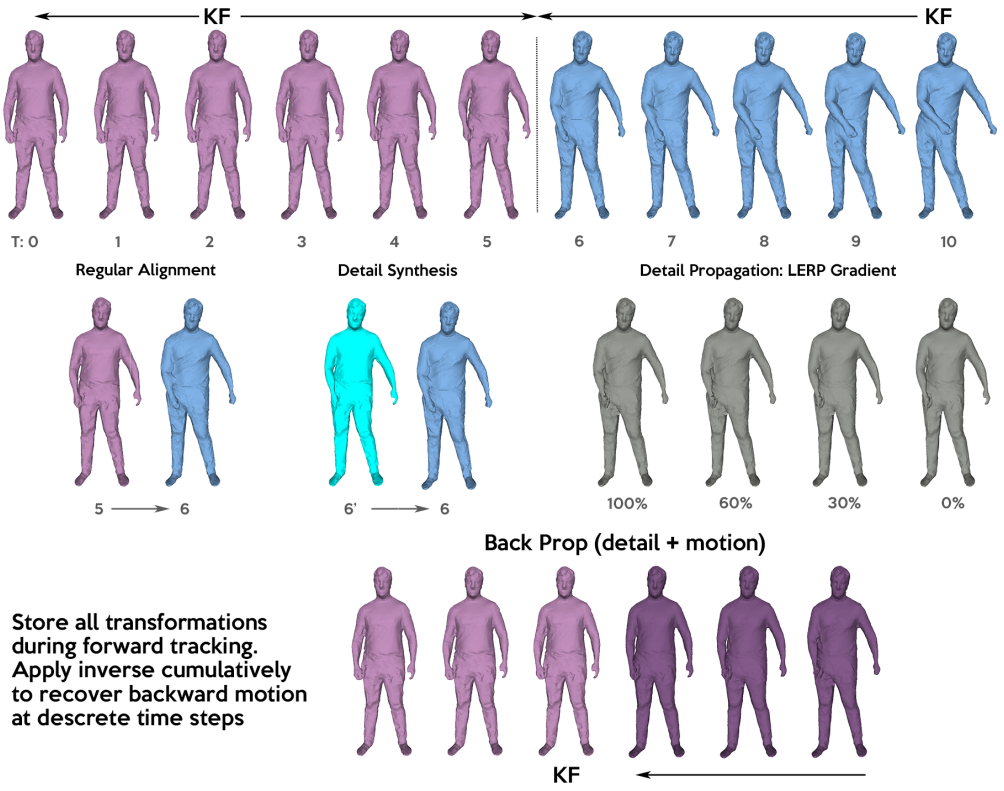

Figure B.3: Detail synthesis. Given two regions pictured top where T is a global frame index. We have keyframes at $T = 2$ and $T = 10$, from which tracking is performed to 5 and 6 respectively. For resolving details from $2 \longrightarrow 5$ we track 5 towards 6 as a normal framewise alignment giving $6'$. We then track $6'$ to 6 which greatly relaxed rigidity parameters in order to synthesis surface details. This detail layer is then linearly interpolated (LERP) for n intervals between 0% and 100% where n is the number of frames between the keyframe and the boundary. We finally use the cached transformations from the original tracking to propagate the LERP intervals to their respective frames i.e. in the above example 5(100%), 4(60%) etc... We apply the same process for detail synthesis from $6 \longrightarrow 10$.

Bibliography

[1] A. Smolic, "An overview of 3d video and free viewpoint video," in *International Conference on Computer Analysis of Images and Patterns*, pp. 1–8, Springer, 2009.

[2] A. Smolic, "3d video and free viewpoint video—from capture to display," *Pattern recognition*, vol. 44, no. 9, pp. 1958–1968, 2011.

[3] R. Pagés, K. Amplianitis, D. Monaghan, J. Ondřej, and A. Smolić, "Affordable content creation for free-viewpoint video and vr/ar applications," *Journal of Visual Communication and Image Representation*, vol. 53, pp. 192–201, 2018.

[4] C. Ozcinar, A. De Abreu, S. Knorr, and A. Smolic, "Estimation of optimal encoding ladders for tiled 360° vr video in adaptive streaming systems," in *The 19th IEEE International Symposium on Multimedia (ISM 2017)*, IEEE, 2017.

[5] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan, "High-quality streamable free-viewpoint video," *ACM Transactions on Graphics (ToG)*, vol. 34, no. 4, pp. 1–13, 2015.

[6] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2304–2314, 2019.

[7] M. Habermann, W. Xu, M. Zollhofer, G. Pons-Moll, and C. Theobalt, "Deepcap: Monocular human performance capture using weak su-

pervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5052–5063, 2020.

[8] "Vologrames." `https://www.vologrames.com/`. Accessed: 2021-12-20.

[9] "MPEG 136." `https://www.mpegstandards.org/meetings/mpeg-136/`. Accessed: 2021-12-27.

[10] K. Guo, P. Lincoln, P. Davidson, J. Busch, X. Yu, M. Whalen, G. Harvey, S. Orts-Escolano, R. Pandey, J. Dourgarian, *et al.*, "The relightables: volumetric performance capture of humans with realistic relighting," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–19, 2019.

[11] O. Schreer, I. Feldmann, S. Renault, M. Zepp, M. Worchel, P. Eisert, and P. Kauff, "Capture and 3d video processing of volumetric video," in *2019 IEEE International conference on image processing (ICIP)*, pp. 4310–4314, IEEE, 2019.

[12] M. Moynihan, R. Pagés, and A. Smolic, "Spatio-temporal upsampling for free viewpoint video point clouds.," in *VISIGRAPP (5: VISAPP)*, pp. 684–692, 2019.

[13] M. Moynihan, R. Pagés, and A. Smolic, "A self-regulating spatio-temporal filter for volumetric video point clouds," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics*, pp. 391–408, Springer, 2019.

[14] M. Moynihan, S. Ruano, R. Pagés, A. Smolic, *et al.*, "Autonomous tracking for volumetric video sequences," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1660–1669, 2021.

[15] T. Kanade, P. Rander, and P. Narayanan, "Virtualized reality: Constructing virtual worlds from real scenes," *IEEE multimedia*, vol. 4, no. 1, pp. 34–47, 1997.

[16] T. Matsuyama and T. Takai, "Generation, visualization, and editing of 3d video," in *Proceedings. First International Symposium on 3D*

Data Processing Visualization and Transmission, pp. 234–245, IEEE, 2002.

[17] S. Wurmlin, E. Lamboray, O. G. Staadt, and M. H. Gross, "3d video recorder," in *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.*, pp. 325–334, IEEE, 2002.

[18] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3d video and free viewpoint video-technologies, applications and mpeg standards," in *2006 IEEE International Conference on Multimedia and Expo*, pp. 2161–2164, IEEE, 2006.

[19] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel, "Free-viewpoint video of human actors," *ACM transactions on graphics (TOG)*, vol. 22, no. 3, pp. 569–577, 2003.

[20] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 600–608, 2004.

[21] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 4, pp. 353–363, 1993.

[22] A. Grunnet-Jepsen, J. N. Sweetser, and J. Woofill, "Stereo depth cameras for mobile phones." `https://dev.intelrealsense.com/docs/stereo-depth-cameras-for-phones`.

[23] S. Ullman, "The interpretation of structure from motion," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 203, no. 1153, pp. 405–426, 1979.

[24] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM siggraph 2006 papers*, pp. 835–846, 2006.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[26] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.

[27] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 1, pp. 519–528, IEEE, 2006.

[28] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "OpenMVG: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*, pp. 60–74, Springer, 2016.

[30] 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*, vol. 24, no. 6, pp. 381–395, 1981.

[31] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*, pp. 298–372, Springer, 1999.

[32] B. G. Baumgart, *Geometric modeling for computer vision*. Stanford University, 1974.

[33] K.-m. G. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette across time part i: Theory and algorithms," *International Journal of Computer Vision*, vol. 62, no. 3, pp. 221–247, 2005.

[34] R. Pagés, D. Berjón, and F. Morán, "Automatic system for virtual human reconstruction with 3d mesh multi-texturing and facial enhancement," *Signal Processing: Image Communication*, vol. 28, no. 9, pp. 1089–1099, 2013.

[35] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving,"

International journal of computer vision, vol. 38, no. 3, pp. 199–218, 2000.

[36] G. Slabaugh, R. Schafer, and M. Hans, "Image-based photo hulls," in *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*, pp. 704–862, IEEE, 2002.

[37] S.-P. Yang, Y.-H. Seo, J.-B. Kim, H. Kim, and K.-H. Jeong, "Optical mems devices for compact 3d surface imaging cameras," *Micro and Nano Systems Letters*, vol. 7, no. 1, pp. 1–9, 2019.

[38] 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*, pp. 343–352, 2015.

[39] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, *et al.*, "Fusion4d: Real-time performance capture of challenging scenes," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–13, 2016.

[40] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, *et al.*, "Real-time non-rigid reconstruction using an RGB-D camera," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 4, p. 156, 2014.

[41] T. Yu, Z. Zheng, K. Guo, J. Zhao, Q. Dai, H. Li, G. Pons-Moll, and Y. Liu, "Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7287–7296, 2018.

[42] M. Dou, P. Davidson, S. R. Fanello, S. Khamis, A. Kowdle, C. Rhemann, V. Tankovich, and S. Izadi, "Motion2fusion: Real-time volumetric performance capture," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 246, 2017.

[43] "Google Draco." `https://github.com/google/draco`. Accessed: 2021-12-20.

[44] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai, "Robust non-rigid motion tracking and surface reconstruction using l0 regularization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3083–3091, 2015.

[45] W. Morgenstern, A. Hilsmann, and P. Eisert, "Progressive non-rigid registration of temporal mesh sequences," in *European Conference on Visual Media Production*, pp. 1–10, 2019.

[46] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," in *ACM SIGGRAPH 2007 papers*, pp. 80–es, ACM New York, NY, USA, 2007.

[47] U. Castellani and A. Bartoli, "3d shape registration," in *3D Imaging, Analysis and Applications*, pp. 353–411, Springer, 2020.

[48] F. Bernardini and H. Rushmeier, "The 3d model acquisition pipeline," in *Computer graphics forum*, vol. 21, pp. 149–172, Wiley Online Library, 2002.

[49] I. K. Park, M. Germann, M. D. Breitenstein, and H. Pfister, "Fast and automatic object pose estimation for range images on the gpu," *Machine Vision and Applications*, vol. 21, no. 5, pp. 749–766, 2010.

[50] J. W. Tangelder and R. C. Veltkamp, "A survey of content based 3d shape retrieval methods," *Multimedia tools and applications*, vol. 39, no. 3, pp. 441–471, 2008.

[51] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611, pp. 586–606, International Society for Optics and Photonics, 1992.

[52] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[53] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp.," in *Robotics: science and systems*, vol. 2, p. 435, Seattle, WA, 2009.

[54] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid icp al-

gorithms for surface registration," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[55] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[56] H. Li, R. W. Sumner, and M. Pauly, "Global correspondence optimization for non-rigid registration of depth scans," *Computer graphics forum*, vol. 27, no. 5, pp. 1421–1430, 2008.

[57] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 82–90, 2016.

[58] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2088–2096, 2017.

[59] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

[60] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

[61] G. Gkioxari, J. Malik, and J. Johnson, "Mesh r-cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9785–9795, 2019.

[62] H. Wei, S. Liang, and Y. Wei, "3d dense face alignment via graph convolution networks," *arXiv preprint arXiv:1904.05562*, 2019.

[63] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov, "Diffusion is all you need for learning on surfaces," *arXiv preprint arXiv:2012.00888*, 2020.

[64] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6411–6420, 2019.

[65] Q. Li, S. Liu, L. Hu, and X. Liu, "Shape correspondence using anisotropic chebyshev spectral cnns," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14658–14667, 2020.

[66] S. Gong, L. Chen, M. Bronstein, and S. Zafeiriou, "Spiralnet++: A fast and highly efficient mesh convolution operator," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[67] A. A. Osman, T. Bolkart, and M. J. Black, "Star: Sparse trained articulated human body regressor," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pp. 598–613, Springer, 2020.

[68] M. Habermann, W. Xu, M. Zollhöfer, G. Pons-Moll, and C. Theobalt, "Livecap: Real-time human performance capture from monocular video," *ACM Trans. Graph.*, vol. 38, pp. 14:1–14:17, Mar. 2019.

[69] W. Xu, A. Chatterjee, M. Zollhöfer, H. Rhodin, D. Mehta, H.-P. Seidel, and C. Theobalt, "Monoperfcap: Human performance capture from monocular video," *ACM Transactions on Graphics (ToG)*, vol. 37, no. 2, pp. 1–15, 2018.

[70] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[71] R. Vidaurre, I. Santesteban, E. Garces, and D. Casas, "Fully convolutional graph neural networks for parametric virtual try-on," in *Computer Graphics Forum*, vol. 39, pp. 145–156, Wiley Online Library, 2020.

[72] I. Santesteban, M. A. Otaduy, and D. Casas, "Learning-based animation of clothing for virtual try-on," in *Computer Graphics Forum*, vol. 38, pp. 355–366, Wiley Online Library, 2019.

[73] S. Saito, J. Yang, Q. Ma, and M. J. Black, "Scanimate: Weakly supervised learning of skinned clothed avatar networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2886–2897, 2021.

[74] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid, "Bodynet: Volumetric inference of 3d human body shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 20–36, 2018.

[75] S. Saito, T. Simon, J. Saragih, and H. Joo, "Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 84–93, 2020.

[76] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

[77] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

[78] M. Michalkiewicz, J. K. Pontes, D. Jack, M. Baktashmotlagh, and A. Eriksson, "Deep level sets: Implicit surface representations for 3d shape inference," *arXiv preprint arXiv:1901.06802*, 2019.

[79] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5939–5948, 2019.

[80] A. Mustafa, A. Caliskan, L. Agapito, and A. Hilton, "Multi-person implicit reconstruction from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14474–14483, 2021.

[81] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: integrating color, texture, mo-

tion and shape," *International journal of computer vision*, vol. 72, no. 2, pp. 195–215, 2007.

[82] "RenderPeople." `https://renderpeople.com/`. Accessed: 2021-12-20.

[83] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu, "Deephuman: 3d human reconstruction from a single image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7739–7749, 2019.

[84] "Microsoft RocketBox Library." `https://github.com/microsoft/Microsoft-Rocketbox`. Accessed: 2021-12-20.

[85] A. Caliskan, A. Mustafa, and A. Hilton, "Temporal consistency loss for high resolution textured and clothed 3d human reconstruction from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1780–1790, 2021.

[86] A. Bozic, P. Palafox, M. Zollhofer, J. Thies, A. Dai, and M. Nießner, "Neural deformation graphs for globally-consistent non-rigid reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1450–1459, 2021.

[87] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[88] N. Donati, A. Sharma, and M. Ovsjanikov, "Deep geometric functional maps: Robust feature learning for shape correspondence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8592–8601, 2020.

[89] K. H. Ruben Wiersma, Elmar Eisemann, "Cnns on surfaces using rotation-equivariant features," *Transactions on Graphics*, vol. 39, July 2020.

[90] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proceedings of the IEEE international conference on computer vision workshops*, pp. 37–45, 2015.

[91] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers,

"Anisotropic diffusion descriptors," in *Computer Graphics Forum*, vol. 35, pp. 431–441, Wiley Online Library, 2016.

[92] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," in *ACM Transactions on Graphics (ToG)*, vol. 28, p. 175, ACM, 2009.

[93] C.-H. Huang, E. Boyer, N. Navab, and S. Ilic, "Human shape and pose tracking using keyframes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3446–3453, 2014.

[94] M. Klaudiny, C. Budd, and A. Hilton, "Towards optimal non-rigid surface tracking," in *European Conference on Computer Vision*, pp. 743–756, 2012.

[95] 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*, pp. 1339–1346, IEEE, 2010.

[96] C. Budd, P. Huang, M. Klaudiny, and A. Hilton, "Global non-rigid alignment of surface sequences," *International Journal of Computer Vision*, vol. 102, no. 1, pp. 256–270, 2013.

[97] A. Mustafa, H. Kim, J. Y. Guillemaut, and A. Hilton, "Temporally coherent 4d reconstruction of complex dynamic scenes," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4660–4669, June 2016.

[98] A. Mustafa and A. Hilton, "Semantically coherent co-segmentation and reconstruction of dynamic scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 422–431, 2017.

[99] A. Mustafa, M. Volino, J.-Y. Guillemaut, and A. Hilton, "4d temporally coherent light-field video," in *2017 International Conference on 3D Vision (3DV)*, pp. 29–37, IEEE, 2017.

[100] F. Prada, M. Kazhdan, M. Chuang, A. Collet, and H. Hoppe, "Spatiotemporal atlas parameterization for evolving meshes," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 58, 2017.

[101] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation: A view centered variational approach," *International journal of computer vision*, vol. 101, no. 1, pp. 6–21, 2013.

[102] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3d motion understanding," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 29–51, 2011.

[103] M. Lang, O. Wang, T. O. Aydin, A. Smolic, and M. H. Gross, "Practical temporal consistency for image-based graphics applications.," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 34–1, 2012.

[104] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang, "Edge-aware point set resampling," *ACM Transactions on Graphics*, vol. 32, pp. 9:1–9:12, 2013.

[105] S. Wu, H. Huang, M. Gong, M. Zwicker, and D. Cohen-Or, "Deep points consolidation," *ACM Transactions on Graphics (ToG)*, vol. 34, no. 6, p. 176, 2015.

[106] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-net: Point cloud upsampling network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2790–2799, 2018.

[107] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, p. 29, 2013.

[108] P. Moulon, P. Monasse, and R. Marlet, "Adaptive structure from motion with a contrario model estimation," in *Asian Conference on Computer Vision*, pp. 257–270, Springer, 2012.

[109] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 2, pp. 354–366, 2013.

[110] D. Berjón, R. Pagés, and F. Morán, "Fast feature matching for detailed point cloud generation," in *Image Processing Theory Tools and Applications (IPTA), 2016 6th International Conference on*, pp. 1–6, IEEE, 2016.

[111] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision*, pp. 501–518, Springer, 2016.

[112] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patchmatch for large displacement optical flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5704–5712, 2016.

[113] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian conference on Image analysis*, pp. 363–370, Springer, 2003.

[114] M. Schaffner, F. Scheidegger, L. Cavigelli, H. Kaeslin, L. Benini, and A. Smolic, "Towards edge-aware spatio-temporal filtering in real-time," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 265–280, 2018.

[115] L. Bao, Q. Yang, and H. Jin, "Fast edge-preserving patchmatch for large displacement optical flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3534–3541, 2014.

[116] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas-kanade feature tracker," *Intel Corporation*, 2001.

[117] E. S. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," in *ACM Transactions on Graphics (ToG)*, vol. 30, p. 69, ACM, 2011.

[118] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 1841–1848, IEEE, 2013.

[119] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1164–1172, 2015.

[120] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[121] T. Luhmann, S. Robson, S. Kyle, and I. Harley, *Close range photoGrammetry*. Wiley, 2007.

[122] D. G. Lowe, "Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image," Mar. 23 2004. US Patent 6,711,293.

[123] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[124] L. Hoyet, K. Ryall, K. Zibrek, H. Park, J. Lee, J. Hodgins, and C. O'sullivan, "Evaluating the distinctiveness and attractiveness of human motions on realistic virtual bodies," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, pp. 1–11, 2013.

[125] Z. Huang, T. Li, W. Chen, Y. Zhao, J. Xing, C. LeGendre, L. Luo, C. Ma, and H. Li, "Deep volumetric video from very sparse multi-view performance capture," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 336–354, 2018.

[126] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi, "3d scanning deformable objects with a single rgbd sensor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 493–501, 2015.

[127] Z. Zheng, T. Yu, H. Li, K. Guo, Q. Dai, L. Fang, and Y. Liu, "Hybridfusion: Real-time performance capture using a single depth sensor and sparse imus," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 384–400, 2018.

[128] 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*, pp. 1386–1395, 2017.

[129] M. Slavcheva, M. Baust, and S. Ilic, "Sobolevfusion: 3d reconstruc-

tion of scenes undergoing free non-rigid motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2646–2655, 2018.

[130] Q. Zhang, B. Fu, M. Ye, and R. Yang, "Quality dynamic human body modeling using a single low-cost depth camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 676–683, 2014.

[131] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1812.08008*, 2018.

[132] D. Xiang, H. Joo, and Y. Sheikh, "Monocular total capture: Posing face, body, and hands in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10965–10974, 2019.

[133] M. Dou, H. Fuchs, and J.-M. Frahm, "Scanning and tracking dynamic objects with commodity depth cameras," in *2013 IEEE international symposium on mixed and augmented Reality (ISMAR)*, pp. 99–106, IEEE, 2013.

[134] N. Pietroni, M. Tarini, and P. Cignoni, "Almost isometric mesh parameterization through abstract domains," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 4, pp. 621–635, 2009.

[135] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3 d shape descriptors," in *Symposium on geometry processing*, vol. 6, pp. 156–164, 2003.

[136] L. Shapira, A. Shamir, and D. Cohen-Or, "Consistent mesh partitioning and skeletonisation using the shape diameter function," *The Visual Computer*, vol. 24, no. 4, p. 249, 2008.

[137] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 614–619, 2003.

[138] R. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng., Trans. ASME, D*, vol. 82, pp. 35–45, 1960.

[139] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," in *ACM SIGGRAPH 2008 papers*, pp. 1–9, 2008. Accessed June 2020 `http://people.csail.mit.edu/drdaniel/mesh_animation/`.

[140] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh: Measuring errors between surfaces using the hausdorff distance," in *Proceedings. IEEE international conference on multimedia and expo*, vol. 1, pp. 705–708, IEEE, 2002.

[141] D. Casas, M. Tejera, J.-Y. Guillemaut, and A. Hilton, "4d parametric motion graphs for interactive animation," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 103–110, 2012.

[142] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Occupancy flow: 4d reconstruction by learning particle dynamics," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5379–5389, 2019.

[143] Y. Li, A. Bozic, T. Zhang, Y. Ji, T. Harada, and M. Nießner, "Learning to optimize non-rigid tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4910–4918, 2020.

[144] A. Bozic, P. Palafox, M. Zollöfer, A. Dai, J. Thies, and M. Nießner, "Neural non-rigid tracking," in *NeurIPS*, pp. 18727–18737, 2020.

[145] A. Bozic, M. Zollhofer, C. Theobalt, and M. Nießner, "Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7002–7012, 2020.

[146] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017.

[147] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, pp. 820–830, 2018.

[148] A. V. Phan, M. Le Nguyen, Y. L. H. Nguyen, and L. T. Bui, "Dgcnn: A convolutional neural network over large-scale labeled graphs," *Neural Networks*, vol. 108, pp. 533–543, 2018.

[149] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[150] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[151] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black, "Dynamic FAUST: Registering human bodies in motion," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[152] M. Fey, J. E. Lenssen, F. Weichert, and H. Müller, "Splinecnn: Fast geometric deep learning with continuous b-spline kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 869–877, 2018.

[153] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Pu-gan: a point cloud upsampling adversarial network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7203–7212, 2019.

[154] W. Zhang, Q. Yan, and C. Xiao, "Detail preserved point cloud completion via separated feature aggregation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pp. 512–528, Springer, 2020.

[155] G. Wang, M. Chen, H. Liu, Y. Yang, Z. Liu, and H. Wang, "Anchor-based spatio-temporal attention 3d convolutional networks for dynamic 3d point cloud sequences," *arXiv preprint arXiv:2012.10860*, 2020.

[156] X. Liu, M. Yan, and J. Bohg, "Meteornet: Deep learning on dynamic 3d point cloud sequences," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9246–9255, 2019.

[157] H. Fan, Y. Yang, and M. Kankanhalli, "Point 4d transformer networks

for spatio-temporal modeling in point cloud videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14204–14213, 2021.

[158] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3d human pose estimation with spatial and temporal transformers," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[159] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[160] A. Chatzitofis, L. Saroglou, P. Boutis, P. Drakoulis, N. Zioulis, S. Subramanyam, B. Kevelham, C. Charbonnier, P. Cesar, D. Zarpalas, *et al.*, "Human4d: A human-centric multimodal dataset for motions and immersive media," *IEEE Access*, vol. 8, pp. 176241–176262, 2020.

[161] Y. Li, H. Takehara, T. Taketomi, B. Zheng, and M. Nießner, "4dcomplete: Non-rigid motion estimation beyond the observable surface," *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[162] A. Caliskan, A. Mustafa, E. Imre, and A. Hilton, "Multi-view consistency loss for improved single-image 3d reconstruction of clothed people," in *Proceedings of the Asian Conference on Computer Vision*, 2020.