Learning Dense 3D Models from Monocular Video

Rui Yu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Department of Computer Science
University College London

August 17, 2017

I, Rui Yu, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Reconstructing dense, detailed, 3D shape of dynamic scenes from monocular sequences is a challenging problem in computer vision. While robust and even real-time solutions exist to this problem if the observed scene is static, for non-rigid dense shape capture current systems are typically restricted to the use of complex multi-camera rigs, taking advantage of the additional depth channel available in RGB-D cameras, or dealing with specific shapes such as faces or planar surfaces. In this thesis, we present two pieces of work for reconstructing dense generic shapes from monocular sequences.

In the first work, we propose an unsupervised approach to the challenging problem of simultaneously segmenting the scene into its constituent objects and reconstructing a 3D model of the scene. The strength of our approach comes from the ability to deal with real-world dynamic scenes and to handle seamlessly different types of motion: rigid, articulated and non-rigid. We formulate the problem as a hierarchical graph-cuts based segmentation where we decompose the whole scene into background and foreground objects and model the complex motion of non-rigid or articulated objects as a set of overlapping rigid parts. To validate the capability of our approach to deal with real-world scenes, we provide 3D reconstructions of some challenging videos from the *YouTube Objects* and KITTI dataset, *etc.*

In the second work, we propose a direct approach for capturing the dense, detailed 3D geometry of generic, complex non-rigid meshes using a single camera. Our method makes use of a single RGB video as input; it can capture the deformations of generic shapes; and the depth estimation is dense, per-pixel and direct. We first reconstruct a dense 3D template of the shape of the object, using a short rigid sequence, and subsequently perform online reconstruction of the non-rigid mesh as it evolves over time. In our experimental evaluation, we show a range of qualitative results on novel datasets and quantitative comparison results with stereo reconstruction.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The recovery of 3D scene information from images is a fundamental problem in computer vision. Recent years have seen significant progress [51, 29, 114, 63, 87, 37, 84, 89] in the field of Structure from Motion(SfM), which can be defined as the problem of joint estimation of the motion of the cameras and the 3D structure of the captured scenes. SfM algorithms have reached an extremely high degree of maturity in which existing systems [87, 37, 89] are capable of performing real time camera tracking and dense 3D reconstruction from a hand-held camera. However, the fundamental assumption of existing robust solutions is that the scene is static, which limits their application to more general, real-world dynamic scenes.

The recent emergence of low cost depth sensors, has brought easy and fast acquisition of 3D geometry closer to reality. Systems such as KinectFusion [88] allow users to scan the detailed 3D shape of rigid scenes. The use of RGB-D sensors has also been extended to markerless capture of non-rigid shapes. The additional depth channel provides a strong geometric prior that has made the recovery of accurate 3D models of deformable shapes from depth sensors possible [73, 77] even in real time [144, 86, 56, 27]. At the same time, many multi-camera techniques for markerless high-end dynamic 3D shape acquisition have been developed over the last decade [31, 128, 33]. However, these multi-camera systems are typically confined to studio settings with complex calibrated camera setups and sophisticated lighting; are often model-based and specific to human motion capture.

In contrast, the acquisition of dense 3D models of generic deformable meshes from a monocular *RGB-only* video stream is significantly harder and continues to be an unsolved and challenging problem in computer vision and graphics. The ability to acquire time-varying dense shapes from monocular

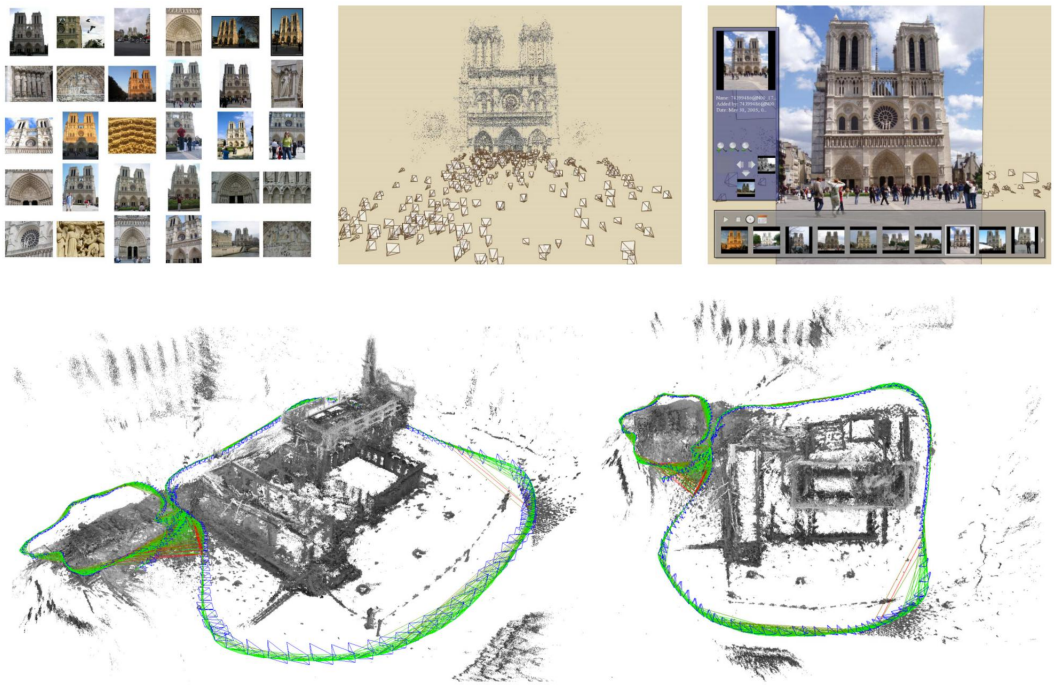


Figure 1.1: Reconstruction results of rigid scenes. Top row shows the 3D reconstruction results from online image searches [114]. From left to right, unstructured collections of photographs from online image search, 3D reconstruction results (including camera viewpoints) and novel ways of browsing the photos. Bottom rows shows accumulated pointclouds of all keyframes of a medium-sized trajectory (from a hand-held monocular camera), generated in real-time by Large-Scale Direct Monocular SLAM(LSD-SLAM) [37]. Images reproduced from [114] (top row) and [37] (bottom row).

RGB video would open the door to easy, lightweight non-rigid capture and, perhaps more importantly, from existing video footage or web-based video libraries such as YouTube.

In this thesis, we focus on the problem of reconstructing dense deformable shapes from a monocular video sequence.

1.1 Motivation

Although rigid SfM is now well understood [51] and has been widely used for commercial applications, the world is essentially dynamic. Compared to static environments, dynamic scenes are much more challenging to reconstruct in 3D, due to the algorithmic and computational difficulties in non-rigid deformation modelling. Multibody SfM and non-rigid structure from motion (NRSfM) have addressed some of the limitations of SfM and have seen sustained progress in dealing with dynamic scenes [91, 100, 97] or creating vivid life-like reconstruc-

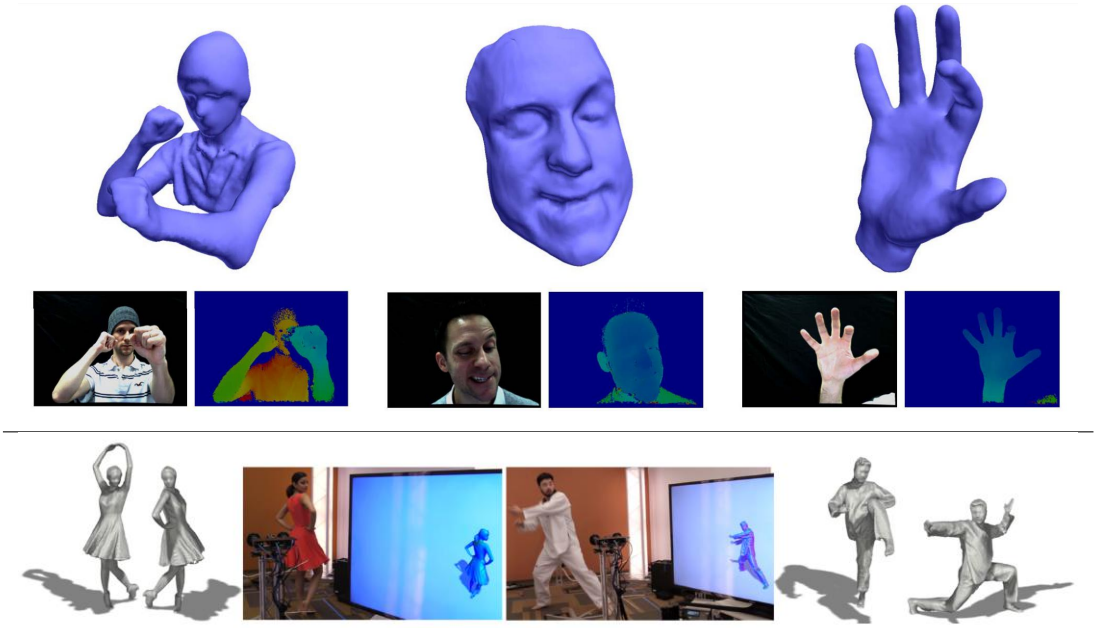


Figure 1.2: Reconstruction results of dynamic scenes with RGB-D camera. Top row shows real-time template-based non-rigid tracking results for 3 different sequences with a RGB-D camera from [144]. Bottom rows shows the results of a real-time high quality 4D (i.e. spatio-temporally coherent) performance capture system [33], allowing for incremental non-rigid reconstruction using noisy input from multiple RGB-D cameras. Images reproduced from [144] (top row) and [33] (bottom row).

tions of deformable objects [45, 119]. However, they remain far behind their rigid counterparts. The research of this thesis aims to bridge the gap between non-rigid reconstruction of dynamic scenes and its rigid counterparts. Our motivation may be understood from two perspectives, first the key role of scene reconstruction in computer vision research, and second the numerous potential applications that will benefit from this research.

In Marr's pioneering work [81], he described vision as the process of discovering useful information by looking at the outside world, e.g. what is present and where it is. The perception of 3D shape is one of the most important cues for our vision system, since it provides useful information for many vision tasks, such as, recognition, navigation, scene understanding, *etc*. It has long been recognized that 3D reconstruction of the outside world is essential for vision systems. In Marr's classic work [81], he introduced a three-stage representation framework for deriving 3D shape from images, namely, primal sketch, 2.5D sketch and 3D model. This 3D information would be better suited as input to recognition tasks than 2D cues as the 3D shape is a viewpoint

independent representation.

Despite the obvious advantages of using 3D information for action and object recognition tasks, recognition algorithms have traditionally focused on using 2D images to learn semantic models of the world. The reason for this can be mostly attributed to the lack of reliable and robust solutions for the problem of 3D reconstruction of *real-world dynamic scenes*.

Apart from computer vision research, this work is also motivated by numerous applications that have been made possible by progress in reconstructing dynamic scenes. Two of the most significant applications are:

- **Entertainment.** Motion capture has been widely used in industry, including gaming, movie, sports, media, *etc*. For example, the Microsoft motion sensing device Kinect [1] has been successfully used in its gaming console Xbox One. In the film industry, motion capture has a long history of being used for CG (Computer Graphics) effects. For instance, in the movie 'The Lord of the Rings: The Two Towers', the action of actor Andy Serkis was mapped to computer generated skin of Gollum/Smeagol as it was being performed in real time. Recently, Face2Face [121] has enabled real-time facial reenactment of a monocular target video sequence (e.g., Youtube video) and has been successfully used in real-time telepresence communication.

- **Medical Imaging.** Image-guided techniques have been used in minimally invasive surgery (MIS) to reduce the damage to human tissue when performing surgery. With the help of vision-based tracking of deforming tissues in real time, surgeons would be able to observe augmented views during the surgery and perform as small incisions as possible.

Although recent years have seen fantastic progress in motion capture, from early marker based systems to recent markerless motion capture systems, its wide application is still limited by the sophisticated setups. Existing systems typically require depth cameras or multiple synchronised cameras, and are often confined to studio settings and specifically tailored for human motion capture.

Monocular video cameras are probably the most common capturing device and have become more and more ubiquitous with the recent emergence of video cameras on phones and laptops. Compared to depth cameras such as Kinect, a single RGB camera setup is less expensive, more lightweight, less invasive and more widely applicable, meaning that it does not need to project specific patterns and can work in a much longer range than active depth sensors.

We believe that the ability to reconstruct dynamic deforming shapes from monocular RGB video would not only be useful for vision research but also for many industry applications.

1.2 Why is Dynamic Scene Reconstruction Challenging?

Compared to rigid scenarios, the reconstruction of dynamic scenes is a less constrained and much more challenging problem. Particularly, the difficulties of reconstructing a general dynamic scene include: successful segmentation of the scene into separate moving objects, 3D reconstruction of complex deformable objects, and computational challenges.

First, to reconstruct general dynamic scenes, in particular, scenarios where there are multiple moving objects, possibly articulated or non-rigid, we need to simultaneously segment the scene into individual objects and reconstruct the deformable shape of each object. The problem of simultaneous segmentation and reconstruction is a difficult chicken-and-egg problem, and even more so when the number of objects is unknown and the motion of objects is non-rigid. Specifically, segmentation needs to be performed based on the motion of the objects whose estimation in turn relies on the segmentation result.

Second, even with perfect segmentation, the problem of NRS*f*M is inherently ill-posed. In rigid S*f*M, given the observations from many different viewpoints, the problem is typically well constrained to guarantee a robust solution, while for NRS*f*M problem this is not the case. In NRS*f*M, the problem is essentially under constrained, with the number of measurements far fewer than the number of unknowns. Without any priors on the camera motion or the observed scene, the problem is equivalent to 3D reconstruction from a single image. As the same image can be generated by an infinite number of 3D shapes, the problem is ill-posed and has no unique solution. In order to solve the problem, we would have to introduce object deformation priors as regularizers. However, the deformation of generic objects is complex to model. The exact form of deformation depends not only on the physical properties of the object itself, but also on possible external forces. The shape may change arbitrarily and may even change topology, making it very challenging to come up with reasonable priors for generic deforming objects. Furthermore, as it is very difficult to capture 3D ground truth for dynamic scenes or to simulate real-world dynamics and create realistic synthetic sequences, the idea of using machine learning

techniques to learn the prior is also stillborn due to the lack of training data.

Last but not least, the computational requirements are another contributing factor that makes the problem challenging. Compared with the rigid case where the world is static and fixed, in dynamic scenes, the 3D geometry may change at every instant and the number of unknowns grows linearly with time, resulting in a significant computational challenge. In fact, it is not until recently [104, 45] that NRS*f*M approaches can only reconstruct sparse feature points. Even today most methods still work in batch mode and are far from real time.

1.3 Contributions

This thesis tackles the problem of dense 3D reconstruction from a monocular video. The main contributions made in the thesis can be summarized as:

- We offer a solution to the problem of scene reconstruction for real-world dynamic monocular videos that deals seamlessly with the presence of non-rigid, articulated or pure rigid motion. In an entirely unsupervised approach, we reorganise/segment the scene into a constellation of object parts, recognise which parts are likely to constitute objects, join them together, and reconstruct the scene. More specifically, we offer solutions to some of the problems of previous approaches to dynamic scene reconstruction: *(i)* Our approach is able to adapt the topology of the neighbourhood graph by breaking edges where necessary to preserve boundaries between objects. In this way our approach can deal with an entire scene where objects might occlude one another and not just pre-segmented objects; *(ii)* Our work results in a hierarchical approach to dynamic scene analysis. At the higher level of the hierarchy the scene is explained as a set of objects that are detached from the background and from each other. At the lower level of the hierarchy, each object could be explained as a set of overlapping parts that can model more complex motion. Figure 1.3 shows our segmentation and scene reconstruction results for three challenging monocular sequences.

- We introduce an end-to-end system that builds a dense template from an initial rigid subsequence and subsequently estimates the deformations of the mesh with respect to the 3D template by minimizing a robust photometric cost. Unlike previous template-based direct methods, we demonstrate our approach on a variety of generic complex non-planar meshes. Our work is the first template-based dense approach that only

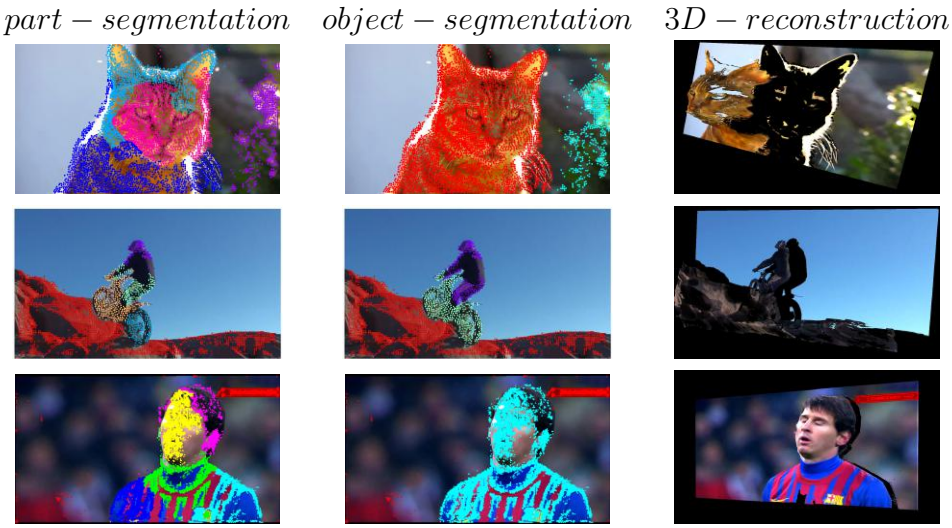


Figure 1.3: Segmentation and 3D reconstruction results of two dynamic sequences of the *Youtube-Objects* Dataset [96] and a football sequence downloaded from YouTube. **Left:** segmentation into *parts* (rigid models). **Centre:** segmentation into *objects*. **Right:** densified 3D video pop-up from a novel viewpoint. The *cat* sequence is a non-rigid sequence occluding a static background. The *motorbike* sequence, acquired with a moving camera, shows articulated motion. Bottom row shows a reconstruction of football footage.

uses monocular RGB data; is frame-to-frame; direct; and suitable for reconstructing generic shapes. To evaluate our system, we show a range of qualitative results on novel datasets and quantitative comparison results with stereo reconstruction. While our algorithm is not real-time, it is sequential and relatively fast, typically requiring 2 seconds per frame on a standard desktop machine to optimize a mesh with approximately 25,000 vertices. Figure 1.4 shows our direct and dense non-rigid reconstruction results on a face sequence.

1.4 Thesis Structure

Chapter 2 introduces the background for the thesis by discussing related techniques, including rigid SfM, multi-model fitting, higher order inference and dense tracking. In Chapter 3 we review existing related work in the literature, such as motion segmentation, multibody reconstruction, non-rigid structure from motion, shape-from-template and model-based non-rigid reconstruction approaches. Chapter 4 introduces our unified whole scene reconstruction framework which can seamlessly deal with unknown number of moving objects

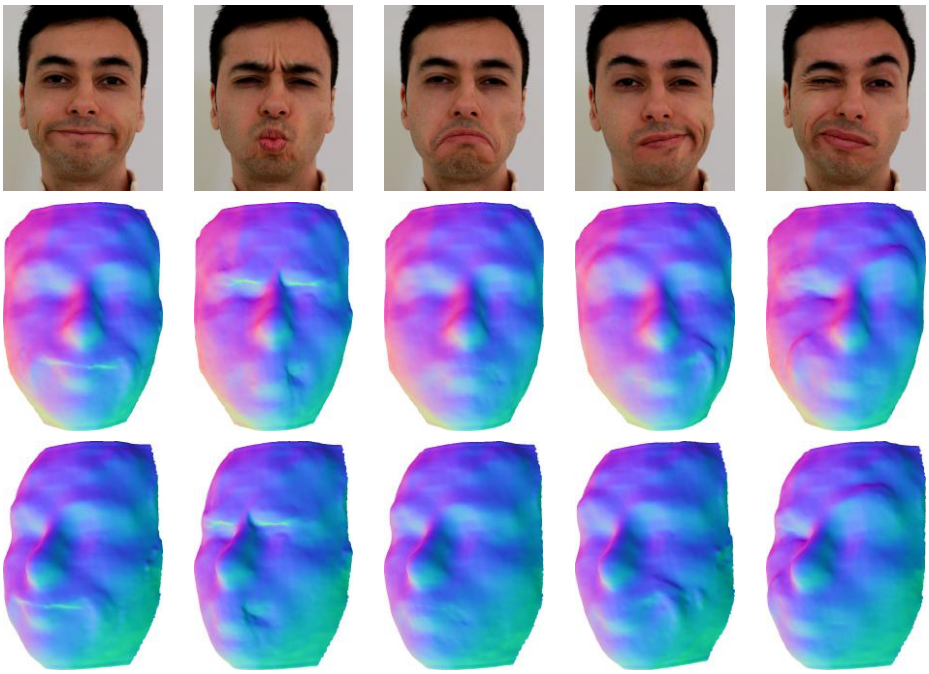


Figure 1.4: Direct deformable reconstruction of our algorithm on a face sequence. Each column corresponds to different views of the same frame.

and the presence of non-rigid, articulated or pure rigid motion. In Chapter 5, we discuss our first RGB-only direct, dense and deformable tracking system. Finally, we conclude the thesis and discuss future directions in Chapter 6.

1.5 Supporting Publications

The selected publications which support the chapters of this thesis are listed as below. The full publication list is available in Appendix A.

1) Chris Russell*, Rui Yu* and Lourdes Agapito. Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes. In *ECCV*, 2014. **(Main text of Chapter 4)**

2) Rui Yu, Chris Russell, Neill D. F. Campbell and Lourdes Agapito. Direct, Dense, and Deformable: Template-Based Non-Rigid 3D Reconstruction from RGB Video. In *ICCV*, 2015. **(Main text of Chapter 5)**

"*" means joint first authorship with equal contribution to the paper.

Chapter 2

Preliminaries

In this chapter, we discuss the background models and mathematical methods that will be used throughout the thesis. First, we introduce the problem of rigid structure from motion (SfM), including the camera projection model (perspective and orthographic), 3D reconstruction using orthographic factorization, *etc*. Then, we introduce the problem of multiple model fitting as well as some state-of-the-art solutions. Furthermore, we discuss higher order graph-cuts optimization techniques which will be used in the formulation of our scene reconstruction. Finally, we introduce the problem of dense photometric tracking and related optimization methods.

2.1 Rigid Structure from Motion

2.1.1 Camera Model

The projection of our 3D world onto a 2D image plane can be described by a pinhole camera model, performing a perspective projection operation, as shown in Figure 2.1. Without loss of generality, let the center of projection, e.g. the camera focus, be at the origin of the world coordinate system. Alternatively, we could apply a rigid transform to align it with the camera coordinate system.

As shown in Figure 2.1, given a 3D point $\mathbf{X} = [X, Y, Z]^T$, its projection on the image plane can be given as:

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{fX}{Z} \\ \frac{fY}{Z} \end{bmatrix} \tag{2.1}$$

where f is the focal length of the camera, x and y are the coordinates of the projected point on the image plane. Equation 2.1 assumes that the image coordinate frame is centered at the principal point. In practice, we usually use uv coordinate system (in pixels), where we define the top-left corner of the

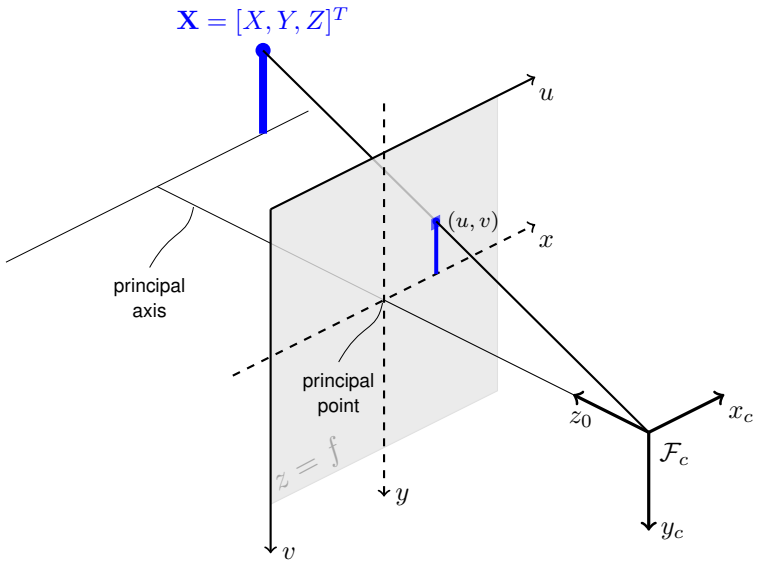


Figure 2.1: The projection of a point under pinhole camera model.

image plane as the origin:

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{fX}{Z} + u_o \\ \frac{fY}{Z} + v_o \end{bmatrix} \tag{2.2}$$

where, u_o and v_o are the coordinates of the offset of the principal point in the uv coordinate system. Using homogeneous coordinates and matrix formulation, equation 2.2 can be rewritten as:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_o & 0 \\ 0 & f & v_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.3}$$

or

$$\lambda \tilde{\mathbf{u}} = \mathbf{P}\tilde{\mathbf{X}} = \mathbf{K} \left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \end{array} \right] \tilde{\mathbf{X}} \tag{2.4}$$

where $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{X}}$ are the homogeneous representation of point $\mathbf{u}$ and $\mathbf{X}$, $\mathbf{P}$ is the projection matrix, $\mathbf{I}$ is a 3×3 identity matrix, $\mathbf{0}$ a 3×1 zero vector and $\mathbf{K}$ is the calibration matrix:

$$\mathbf{K} = \begin{bmatrix} f & 0 & u_o \\ 0 & f & v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

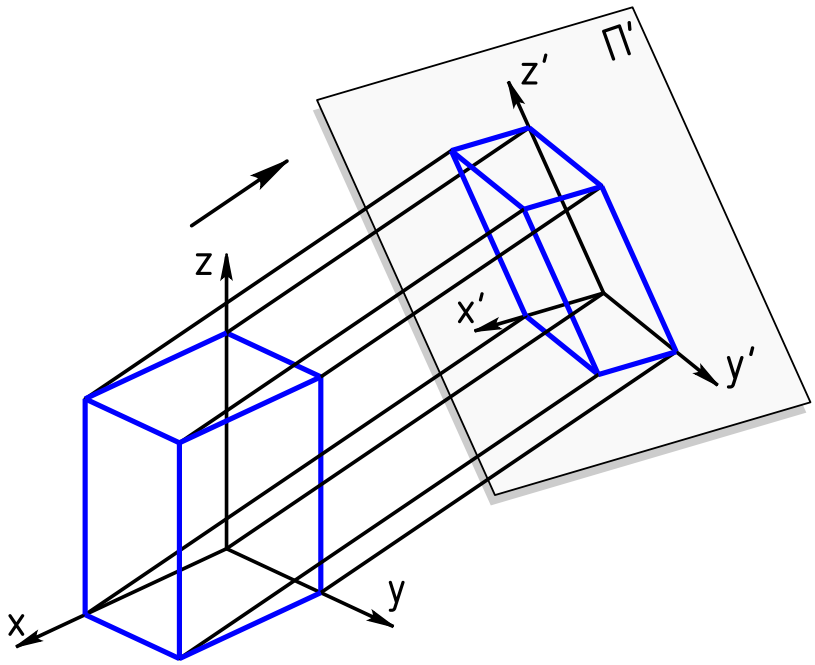


Figure 2.2: Parallel projection of an orthographic camera: a cube (in blue) is projected under orthography to the image plane of the camera.

In general, when the rigid transformation between the world and camera coordinate systems is given by rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, we have:

$$\lambda\tilde{\mathbf{u}} = \mathbf{P}\tilde{\mathbf{X}} = \mathbf{K}\left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \end{array}\right]\tilde{\mathbf{X}} \tag{2.6}$$

2.1.2 Orthographic Camera Model

The pinhole camera model introduced above is a perspective camera model, and can model perspective effects, i.e. the size of an object in the image plane is inversely proportional to its distance from the camera. However, when the camera is far away compared to the size of the object, perspective effects are negligible and the projection from the 3D world to the image plane can be well approximated by an orthographic camera model. As shown in Figure 2.2, in the orthographic case, the rays are parallel and orthogonal to the image plane. The camera projection matrix can be written as:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

For a general orthographic projection, we assume that the rigid transformation between the world and the camera coordinate systems is given by a rotation $\mathbf{R}$ and translation $\mathbf{t}$, therefore:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^T & t_1 \\ \mathbf{r}_2^T & t_2 \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{2.8}$$

where $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]^T$, $\mathbf{t} = [t_1, t_2, t_3]^T$. For a 3D point $\mathbf{X} = [X, Y, Z]^T$, its 2D projection $\mathbf{u} = [u, v]^T$ on the image plane can be computed as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \end{bmatrix} \mathbf{X} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \tag{2.9}$$

The above equation shows clearly that for an orthographic model, the 2D projection does not depend on the third component of the translation vector, meaning that an object moving in the direction of the depth will preserve the same projection.

2.1.3 Orthographic Factorization

Consider a set of 3D points $\{\mathbf{X}_j, j = 1, 2, ...P\}$ observed by an orthographic camera. The projection of the j^{th} point on the i^{th} image plane via orthography can be written as:

$$\mathbf{u}^i = \begin{bmatrix} u_j^i \\ v_j^i \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^{i\,T} \\ \mathbf{r}_2^{i\,T} \end{bmatrix} \mathbf{X}_j + \begin{bmatrix} t_1^i \\ t_2^i \end{bmatrix} = \mathbf{R}^i \mathbf{X}_j + \mathbf{t}^i \tag{2.10}$$

where $\mathbf{u}^i = [u_j^i, v_j^i]^T$ is the 2D projection and $\mathbf{R}^i$ and $\mathbf{t}^i$ are the truncated rotation matrix and translation vector of the orthographic camera respectively. Without loss of generality, we assume that the origin of the world coordinate system is defined at the centroid of the 3D points, therefore the translation vector $\mathbf{t}^i$ can be easily eliminated by subtracting the centroid of the 2D projections from $\mathbf{u}^i$. We can now formulate the projection of P points into F camera frames in matrix form:

$$\mathbf{W} = \begin{bmatrix} \mathbf{R}^1 \\ \mathbf{R}^2 \\ \vdots \\ \mathbf{R}^F \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_P \end{bmatrix} = \mathbf{RS} \tag{2.11}$$

where $\mathbf{W}$ is the so called measurement matrix:

$$\mathbf{W} = \begin{bmatrix} u_1^1 & u_2^1 & \cdots & u_P^1 \\ v_1^1 & v_2^1 & \cdots & v_P^1 \\ \vdots & \vdots & \vdots & \vdots \\ u_1^F & u_2^F & \cdots & u_P^F \\ v_1^F & v_2^F & \cdots & v_P^F \end{bmatrix} \tag{2.12}$$

Th measurement matrix $\mathbf{W}$ is the product of two low rank matrices: the motion matrix $\mathbf{R}(2F \times 3)$ and the structure matrix $\mathbf{S}(3 \times P)$. Based on this observation, Tomasi and Kanade [122] proposed a factorization method for orthographic reconstruction. The method has two main steps: low rank factorization and metric upgrade.

In the first step, the $2F \times P$ measurement matrix $\mathbf{W}$ is factorized into two low rank matrices, a $2F \times 3$ matrix $\hat{\mathbf{M}}$ and a $3 \times P$ matrix $\hat{\mathbf{S}}$, by singular value decomposition (SVD):

$$\mathbf{W} \approx \mathbf{U_3 \Lambda_3 V_3^T} = \mathbf{U_3 \Lambda_3^{1/2} \Lambda_3^{1/2} V_3^T} = \hat{\mathbf{M}}\hat{\mathbf{S}} \tag{2.13}$$

where $\mathbf{U_3}$ is a $2F \times 3$ matrix, $\mathbf{\Lambda_3}$ is a 3×3 diagonal matrix that contains the three largest singular values of $\mathbf{W}$, and $\mathbf{V_3}$ is a $P \times 3$ matrix. In general, the recovered $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ do not correspond to motion and structure matrices due to the ambiguity of the factorization:

$$\hat{\mathbf{M}}\hat{\mathbf{S}} = \hat{\mathbf{M}}\mathbf{Q}\mathbf{Q}^{-1}\hat{\mathbf{S}} = \mathbf{RS} \tag{2.14}$$

where $\mathbf{Q}$ could be an arbitrary 3×3 invertible matrix.

In the metric upgrade step, we solve for the rectification matrix $\mathbf{Q}$. For each rotation matrix $\mathbf{R}^i, i = 1, 2, \cdots, F$, we have the following orthonormality constraints:

$$\begin{aligned} {\mathbf{r}_1^i}^T \mathbf{r}_1^i &= 1 \\ {\mathbf{r}_2^i}^T \mathbf{r}_2^i &= 1 \\ {\mathbf{r}_1^i}^T \mathbf{r}_2^i &= 0 \end{aligned} \tag{2.15}$$

Therefore, in order to recover the correct Euclidean reconstruction, we need to find the rectification matrix $\mathbf{Q}$ such that $\mathbf{R} = \hat{\mathbf{M}}\mathbf{Q}$ satisfies constraints 2.15.

In other words, we have:

$$\begin{aligned} \hat{\mathbf{M}}_i \mathbf{Q}\mathbf{Q}^T \hat{\mathbf{M}}_i^T &= 1 \quad \forall i \in \{1, 2, \cdots, 2F\} \\ \hat{\mathbf{M}}_{2i} \mathbf{Q}\mathbf{Q}^T \hat{\mathbf{M}}_{2i-1}^T &= 0 \quad \forall i \in \{1, 2, \cdots, F\} \end{aligned} \tag{2.16}$$

where $\hat{\mathbf{M}}_i$ is the i^{th} row of $\hat{\mathbf{M}}$. This leads to $3F$ linear constraints in the elements of the symmetric Gram matrix $\mathbf{G} = \mathbf{Q}\mathbf{Q}^T$. With enough frames G can be estimated via least squares and Q may be recovered via cholesky decomposition.

It should be noted that in the presence of noise, the estimated 'motion matrix' $\mathbf{R}$ does not strictly satisfy the orthonormality constraints as equation 2.16 is only solved in least squares sense. In order to obtain a valid motion matrix, one could project matrix $\mathbf{R}$ onto the motion matrix manifold. To further refine the motion matrix $\mathbf{R}$ and the structure matrix $\mathbf{S}$, bundle adjustment could be performed to minimize the reprojection error.

2.1.4 Ambiguities of Orthographic Reconstruction

In general, there are two ambiguities which can not be resolved when performing orthographic reconstruction: the depth and flip ambiguities. The depth ambiguity is a per-frame ambiguity while the flip ambiguity is global.

The depth ambiguity arises from the fact that the depth component of the 3D translation vector $\mathbf{t}$ cannot be determined. It is not possible to recover the absolute value for the translation along the depth direction, as any movement in this direction results in the same 2D projection. In other words, the depth component could take any value without changing the measurement matrix.

To understand the flip ambiguity, we introduce a 3×3 matrix $\mathbf{A}$,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{2.17}$$

and rewrite equation 2.11 as

$$\mathbf{W} = \mathbf{R}\mathbf{S} = \mathbf{R}\mathbf{A}\mathbf{A}\mathbf{S} = \mathbf{R}'\mathbf{S}' \tag{2.18}$$

where $\mathbf{R}'$ and $\mathbf{S}'$ are flipped motion and structure matrices, with the third column/row flipped, respectively. To be more concrete, we assume $\mathbf{R}_1$ and $\mathbf{R}'_1$ are the 3D rotation matrix associated with frame 1 and its flipped version

respectively. Given a 3D point $\mathbf{X}_1$ and its flipped version $\mathbf{X}'_1$, we have

$$\mathbf{R}_1\mathbf{X}_1 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ d_1 \end{bmatrix} \tag{2.19}$$

$$\mathbf{R}'_1\mathbf{X}'_1 = \begin{bmatrix} r_{11} & r_{12} & -r_{13} \\ r_{21} & r_{22} & -r_{23} \\ -r_{31} & -r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ -z_1 \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ -d_1 \end{bmatrix} \tag{2.20}$$

Therefore, both the projections before and after flipping result in the same measurement matrix and only differ in the sign of their depth values.

These two ambiguities are intrinsic, i.e. they do not depend on the structure of the 3D scene or the motion of the camera, and cannot be resolved without additional information.

In certain scenarios, either because the object's 3D structure is close to plane or because it becomes front-parallel and all visible points lie approximately on a plane, an additional flip ambiguity may occur, with regard to every degenerate frame. Suppose the 3D shape of visible points in frame f is $\mathbf{S}^f$ and the corresponding 3D rotation is $\mathbf{R}^f$. Suppose $\mathbf{S}^f$ is rank 2 and degenerate, then we have:

$$\mathbf{W}^f = \mathbf{R}^f\mathbf{S}^f = \mathbf{R}^f\mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{R}^f\mathbf{U}\mathbf{A}\mathbf{D}\mathbf{V}^T = \underbrace{\mathbf{R}^f\mathbf{U}\mathbf{A}\mathbf{U}^T}_{\mathbf{R}'^f}\underbrace{\mathbf{U}\mathbf{D}\mathbf{V}^T}_{\mathbf{S}^f} = \mathbf{R}'^f\mathbf{S}^f \tag{2.21}$$

When the 3D shape is close to being planar, an ambiguity appears since $\mathbf{R}^f$ and $\mathbf{R}'^f$ would lead to similar projections. This rotational flip ambiguity could be resolved by enforcing temporal motion smoothness constraint, since it only affects frames for which the structure appears as planar. By formulating this as a dynamic programming problem over all the frames, this rotational flip ambiguity can be solved efficiently.

2.2 Multiple Model Fitting

Multiple model fitting is a common problem in many computer vision applications, such as motion segmentation, piecewise-rigid reconstruction, *etc.* It is a typical chicken-and-egg problem, where measurements need to be assigned to an unknown number of models whose parameters must be estimated at the same time. Existing multiple model fitting approaches can be classified into

four main categories: *(i)* greedy generalisations of RANSAC to multiple models, *(ii)* Hough transform based clustering methods, *(iii)* affinity based clustering methods, and *(iv)* energy-based multiple model fitting methods.

When there is only one model, RANSAC [40] is a well-known robust method which can deal with outliers. The basic idea is to generate a number of proposals by randomly sampling data points and select the one with the largest number of inliers (the fitting error is within a predefined threshold). Various generalisations of RANSAC [123, 62] to multiple models have been proposed in the literature. These methods usually run RANSAC sequentially and greedily extract the model with the largest number of inliers at each step. However, because of the greedy selection, these methods are fundamentally flawed and sensitive to noise.

Hough transform based approaches formulate the problem as clustering in the space of model parameters, where the identified modes in the parameter space correspond to the models we are searching for. Each data point votes for candidate models in the parameter space and the modes in the space correspond to the models with the highest vote. When the number of model parameters is large, this voting scheme suffers from the exponential nature of the algorithm complexity and it's sensitive to noise as the average number of votes in each parameter bin will be small.

Affinity based methods first compute the similarity between points and then perform clustering on the data based on the affinity matrix. Sparse representation based methods [35], residual histogram analysis [142] and ordered residual kernel methods [22] are amongst the most widely used similarity computation methods. In the second step, given the affinity matrix, spectral clustering [134] is often used for data points clustering. The performance of this approach heavily relies on the similarity measure between the data points. However, this two step pipeline approach is not optimal and moreover the affinity computation often involves using heuristics.

Recently, energy-based methods [58] have been successfully used for multiple model fitting problems. Compared to previous methods, this approach offers a more principled solution by minimizing a pre-defined global objective. Moreover, by modifying the objective function, this formulation allows to incorporate priors more straightforwardly. For example, in geometric multi-model fitting problems, neighbouring data points tend to belong to the same model. This smoothness prior could easily be integrated to the formulation by adding a smoothness term to the global energy.

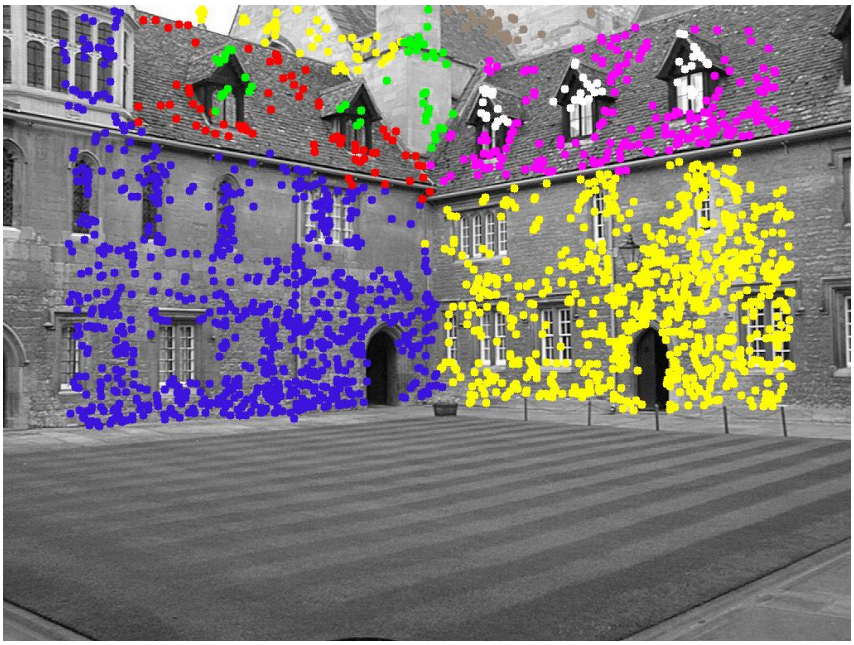


Figure 2.3: Multi-homography fitting result for stereo images from VGG (Oxford) Merton College I using PEaRL [58]. The spatial coherence prior is well justified by the regular planar structures of the building. Figure from [58].

2.2.1 Energy-based Geometric Multiple Model Fitting

In this section, we introduce the formulation for energy-based geometric multi-model fitting and discuss its applications in related computer vision problems. In particular, we introduce the PEaRL [58] algorithm, a general framework which combines RANSAC-style model sampling from data points with iterative re-estimation of inliers and model parameters by minimizing a global energy.

Assuming $\mathcal{L}$ is the set of all models, $\boldsymbol{\theta} = \{\theta_l \mid \forall\ l \in \mathcal{L}\}$ is the set of model parameters, $\mathcal{P}$ is the set of input data points, and $\mathbf{x} = \{x_1, x_2, \cdots, x_P\} \in \mathcal{L}^P$ is the labelling for all P input points, PEaRL proposes to minimize the following energy:

$$C(\mathbf{x}, \boldsymbol{\theta}) = \sum_{\forall p \in \mathcal{P}} \psi_p(\theta_{x_p}) + \sum_{(p,q) \in \mathcal{N}} \psi_{pq}(x_p, x_q) + \psi(\mathbf{x}) \tag{2.22}$$

where $\psi_p(\theta_{x_p})$ is the unary cost of assigning point p to model x_p, $\psi_{pq}(x_p, x_q)$ is the spatial smoothness cost between neighbouring points p and q on the neighbourhood graph $\mathcal{N}$, and $\psi(\mathbf{x})$ is the Minimum Description Length (MDL) cost, i.e. it penalizes the total number of models.

Note that although spatial smoothness priors are widely used in vision, they are not as common in geometric multi-model fitting methods. As this prior

Algorithm 1: PEaRL

Propose:
Generate model proposals by sampling from data points.
while *not converged* **do**
Expand:
Given the models, reassign the data points.
Re-estimate Labels:
Given the data points for each model, refit the models.
end

is not straightforward to encode in many other types of methods, it is usually ignored. However, for geometric fitting problems, many authors [58, 59] have argued that this prior should be exploited as its use gives better performance. This spatial coherence prior can be justified generatively because clusters of inliers are generated by regular objects, as shown in Figure 2.3.

PEaRL is an acronym for 'Propose Expand and Re-estimate Labels'. As shown in Algorithm 1, the algorithm first proposes a large number of candidate models (labels) by sampling the data points, and then it alternates between model expansion and parameter re-estimation.

To illustrate the PEaRL algorithm in more detail, we show its application to line fitting as a working example. Specifically, for multi-line fitting, the unary cost $\psi_p(x_p)$, i.e. the fitting error between each point p and its assigned model x_p, is the point-to-line distance. For point $p = (x, y)$, assuming the parameter of model x_p is given by $\theta_{x_p} = (a, b)$, we have:

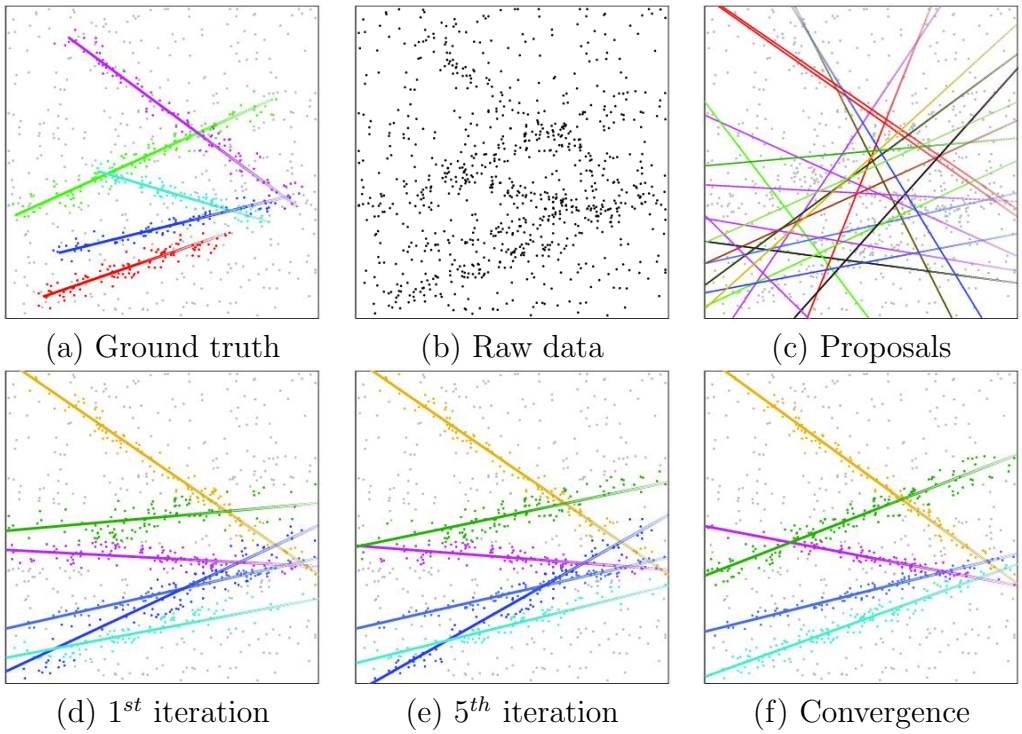
$$\psi_p(\theta_{x_p}) = \frac{|y - ax - b|^2}{\sqrt{a^2 + 1}} \tag{2.23}$$

Assuming a neighbourhood structure for the data points, which might be obtained by K-nearest neighbour or delaunay triangulation, the pairwise smoothness term $\psi_{pq}(x_p, x_q)$ encourages neighbouring points to take the same model:

$$\psi_{pq}(x_p, x_q) = \begin{cases} 0 & \text{if } x_p = x_q \\ \lambda \exp(-\frac{\|p-q\|^2}{\sigma^2}) & \text{otherwise} \end{cases} \tag{2.24}$$

The third term in 2.22 is a label cost term, which penalizes the number of models used to explain the input data:

$$\psi(\mathbf{x}) = \beta \sum_{\forall l \in \mathcal{L}} \delta_l(\mathbf{x}) \tag{2.25}$$



(a) Ground truth (b) Raw data (c) Proposals

(d) 1^{st} iteration (e) 5^{th} iteration (f) Convergence

Figure 2.4: Multi-line fitting result using PEaRL. Above shows 900 raw data points with 50% generated from 5 line intervals. Random sampling proposes a list of candidate lines (we show 20 out of 100). The algorithm converges after a few iterations. Figure from [32].

where the indicator function $\delta_l(\mathbf{x})$ is defined as:

$$\delta_l(\mathbf{x}) = \begin{cases} 1 & \exists p : x_p = l \\ 0 & \text{otherwise} \end{cases}$$

Figure 2.4 shows the result of using PEaRL for multi-line fitting. The data is generated from 5 line intervals with 50% outliers. In this example, 100 initial models are proposed by random sampling. Most of these models are removed and only 5 models are left after the first iteration. The algorithm converges in a few iterations and successfully recovers the 5 input lines.

PEaRL has been used in many geometric multi-model fitting problems, such as motion segmentation, homography estimation, fundamental matrix estimation, *etc*. It provides a powerful framework for these kinds of problems by combining data sampling with simultaneous model fitting and point assignment via minimization of a single unified global energy.

2.3 Higher Order Inference

The label cost term $\psi(\mathbf{x})$ in the above section is defined over the labelling of all the data points and is therefore a high order term, i.e. order greater than 2. Apart from this, a variety of higher order potentials have been proposed and successfully used in some interesting vision tasks. In comparison with unary and pairwise potentials, higher order potentials are able to model more complex interactions between variables. Because of their greater expressive power, it has been shown that adding higher order potentials can significantly improve performance in related vision applications, such as semantic segmentation [65], stereo reconstruction [72], *etc*.

However, minimizing energies with higher order potentials is challenging and their wider applications have been limited due to the lack of efficient optimization algorithms for the corresponding energy. On the one hand, researchers would like to use complex higher order potentials which could naturally model the properties of related problems, on the other efficient algorithms would need to be proposed for the resulting optimization problem. In this section, we introduce some successful higher order potentials which have proved useful in related applications and efficient optimization algorithms for dealing with these potentials.

2.3.1 Graph-Cuts

Graph-cuts [14] is a well-known graph-based fast energy minimization algorithm that has been successfully used for many problems in computer vision, such as image denoising, image segmentation and stereo reconstruction. In particular, it is efficient in solving a variety of labelling problems. Given a set of labels $\mathcal{L} = \{l_1, l_2, \cdots, l_k\}$, the task of a labelling problem is to assign a label $x_p \in \mathcal{L}$ for each point p such that the labelling $\mathbf{x} = \{x_1, x_2, \cdots, x_P\} \in \mathcal{L}^P$ achieves the minimum for the corresponding energy. In many interesting vision applications, we have the following energy:

$$C(\mathbf{x}) = \sum_{p \in \mathcal{P}} \psi_p(x_p) + \sum_{(p,q) \in \mathcal{N}} \psi_{pq}(x_p, x_q) \tag{2.26}$$

where $\psi_p(x_p)$ and $\psi_{pq}(x_p, x_q)$ are unary and pairwise potentials respectively. In the case of binary labels, i.e. $\mathcal{L} = \{0, 1\}$, if the pairwise potentials satisfy the following submodularity condition:

$$\psi_{pq}(0,1) + \psi_{pq}(1,0) \geq \psi_{pq}(1,1) + \psi_{pq}(0,0) \tag{2.27}$$

then this energy can be formulated as a graph-cuts problem and globally minimized with a minimum cut of the corresponding graph. Following max-flow min-cut theorem, this minimum cut solution could be found by solving an equivalent maximum-flow problem. The advantage of using graph-cuts is that for binary submodular energies, it gives the global optimal solution efficiently and when used for solving more general submodular energies with multiple labels, it achieves a locally optimal solution with good properties.

Consider a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ defined over nodes $\mathcal{V}$ and edges $\mathcal{E}$. The set of nodes $\mathcal{V} = \{v_1, v_2, \cdots, v_P, s, t\}$ consists of one node v_p for each point $p \in \mathcal{P}$ and two terminal nodes, i.e. s and t, also referred to as the source and sink nodes respectively. The set of edges $\mathcal{E}$ include two different kinds of edges, terminal edges and neighbourhood edges. Terminal edges connect nodes v_p with terminal nodes s or t, while neighbourhood edges connect two neighbouring nodes. Each edge $e = (i, j) \in \mathcal{E}$ is associated with a nonnegative weight w_{ij}. We define a cut on the graph $\mathcal{G}$ as a partition of the nodes into two disjoint sets S and T such that $s \in S$ and $t \in T$. The cost of a cut is defined as the sum of weights over edges between S and T:

$$\sum_{i \in S, j \in T, (i,j) \in \mathcal{E}} w_{ij} \tag{2.28}$$

It is straightforward to show that any cut of $\mathcal{G}$ could be described by a binary labelling of the points $\mathcal{P}$. Specifically, we set x_p to 0 if $v_p \in S$ and 1 if $v_p \in T$. Therefore, $\mathcal{G}$ represents an energy function which maps the labellings of the points to the cost of corresponding cuts. Moreover, minimizing this particular energy function is equivalent to computing the minimum cut on the graph $\mathcal{G}$. The question remains what energy functions could be represented by graph $\mathcal{G}$? In [68] it was shown that pairwise potentials are representable with graphs if and only if they are submodular.

In order to show the graph construction for a binary pairwise submodular energy 2.26, we rewrite it in the following form (refer to [68] for details):

$$C(\mathbf{x}) = \sum_{p \in \mathcal{P}} c_{sp} x_p + \sum_{p \in \mathcal{P}} c_{pt}(1 - x_p) + \sum_{(p,q) \in \mathcal{N}} c_{pq}(1 - x_p) x_q \tag{2.29}$$

where $c_{sp} \in \mathbb{R}_0^+, c_{pt} \in \mathbb{R}_0^+, \forall p \in \mathcal{P}$ and $c_{pq} \in \mathbb{R}_0^+, \ \forall (p, q) \in \mathcal{E}$. The graph construction for this energy is rather straightforward. For the first term we draw a directed edge with weight c_{sp} from source s to node v_p , while for the

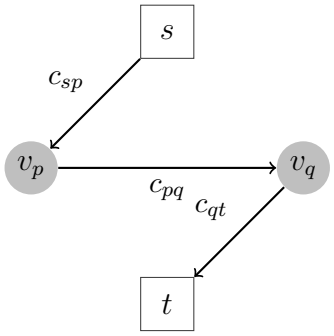


Figure 2.5: Graph construction result for a simple binary pairwise submodular energy $c_{sp}x_p + c_{qt}(1 - x_q) + c_{pq}(1 - x_p)x_q$. Here all the three coefficients c_{sp}, c_{qt} and c_{pq} are positive and we set x_p to 0 if the corresponding node v_p is connected to source s after a cut, otherwise to 1.

second term we draw a directed edge with weight c_{pt} from v_p to sink s. For the pairwise cost, we draw a directed edge with weight c_{pq} from node x_p to x_q. See Figure 2.5 for a simple example.

Alpha-Expansion Algorithm

Alpha-expansion [14] is a widely used algorithm for minimizing multi-labelling submodular energies. The main idea is to convert the difficult multi-labelling problem into a series of binary labelling problems, which can then be efficiently optimized with graph-cuts. Specifically, the algorithm iterates over a randomly selected label α and searches for a better local solution, also referred to as α-expansion move. In an *expansion move*, each variable is given a binary choice of either keeping its current label or switching to label α, encoded by moving variables $\mathbf{t} = \{t_1, t_2, \cdots, t_P\} \in \{0, 1\}^P$. In particular, we have:

$$x'_p = \begin{cases} x_p & \text{if } t_p = 0 \\ \alpha & \text{if } t_p = 1 \end{cases} \tag{2.30}$$

where x_p and x'_p are the labels before and after an expansion move respectively. The best expansion move $\mathbf{t}$ is found via graph-cuts and the current labelling is updated if the expansion move solution leads to a lower energy. The algorithm converges when no better α-expansion move can be found.

2.3.2 Higher Order Potentials

P^n Potts Model

The P^n Potts model is a higher order potential proposed by Kohli *et al.* [64].

It is an extension of the pairwise Potts model and can be defined as:

$$\psi_c(\mathbf{x}_c) = \begin{cases} \gamma_{c,l} & \text{if } \forall p \in c : x_p = l \\ \gamma_{c,\max} & \text{otherwise} \end{cases} \tag{2.31}$$

where $\gamma_{c,\max} > \gamma_{c,l} > 0, \forall l \in \mathcal{L}$ and $\mathbf{x}_c$ is the labelling of the variables in the clique c (a set of variables) of the potential. This potential encourages the labelling of the clique to be consistent and pays a maximum fixed penalty otherwise. In the binary case, this higher order potential can be rewritten in the following form:

$$\psi_c(\mathbf{x}_c) = \gamma_{c,\max} - k_c^0 \prod_{i \in c} \Delta(x_i = 0) - k_c^1 \prod_{i \in c} \Delta(x_i = 1) \tag{2.32}$$

where Δ is the Dirac function, k_c^0 and k_c^1 are defined as:

$$\begin{aligned} k_c^0 &= \gamma_{c,\max} - \gamma_{c,0} \\ k_c^1 &= \gamma_{c,\max} - \gamma_{c,1} \end{aligned} \tag{2.33}$$

By introducing an auxiliary variable z_0, we have:

$$-\prod_{i \in c} \Delta(x_i = 0) = \min_{z_0} \left(z_0 + \sum_{i \in c} x_i(1 - z_0) - 1 \right) \tag{2.34}$$

Similarly,

$$-\prod_{i \in c} \Delta(x_i = 1) = \min_{z_1} \left(-z_1 + \sum_{i \in c} (1 - x_i) z_1 \right) \tag{2.35}$$

Putting all these results together, from equations 2.32, 2.34 and 2.35, $\psi_c(\mathbf{x}_c)$ can be rewritten as:

$$\psi_c(\mathbf{x}_c) = \min_{z_0, z_1} \left(k_c^0 z_0 + \sum_{i \in c} k_c^0 x_i(1 - z_0) + k_c^1(1 - z_1) + \sum_{i \in c} k_c^1(1 - x_i) z_1 \right) + C \tag{2.36}$$

where $C = \gamma_{c,\max} - k_c^0 - k_c^1$ is a constant value that doesn't influence the solution of the energy minimization problem. Formula 2.36 has a similar form as 2.29, and the corresponding graph can be constructed straightforwardly, as shown in Figure 2.6(a).

The Robust P^n Model

The robust P^n model was later introduced by Kohli *et al.* [67] as a robustified

extension of the P^n model. Instead of having the same energy for any inhomogeneous labelling of the clique variables, this new cost increases with the amount of labelling inconsistency. For each label, the energy increases linearly with the number of different labelled points, and is then truncated with a fixed threshold. The cost of the clique is defined as the minimum over all the labels:

$$\begin{aligned}\psi_c(\mathbf{x}_c) &= \min\left(\gamma_{c,max}, \min_{l\in\mathcal{L}}(\gamma_{c,l} + k_c^l \sum_{i\in c}\Delta(x_i \neq l))\right) \\ &= \min_{l\in\mathcal{L}}\left(\min(\gamma_{c,max}, \gamma_{c,l} + k_c^l \sum_{i\in c}\Delta(x_i \neq l))\right) \\ &= \min_{l\in\mathcal{L}}(f_c^l(\mathbf{x}_c))\end{aligned} \tag{2.37}$$

where k_c^l is defined as:

$$k_c^l = \frac{\gamma_{c,max} - \gamma_{c,l}}{Q} \tag{2.38}$$

Here Q is a truncation parameter of the potential and satisfies the constraint $2Q < |c|$, i.e. the size of the clique. For each label $l \in \mathcal{L}$, we have:

$$\begin{aligned}f_c^l(\mathbf{x}_c) &= \min\left(\gamma_{c,max}, k_c^l \sum_{i\in c}\Delta(x_i \neq l) + \gamma_{c,l}\right) \\ &= \begin{cases} k_c^l \sum_{i\in c}\Delta(x_i \neq l) + \gamma_{c,l} & \text{if } \sum_{i\in c}\Delta(x_i \neq l) < Q \\ \gamma_{c,max} & \text{otherwise}\end{cases}\end{aligned} \tag{2.39}$$

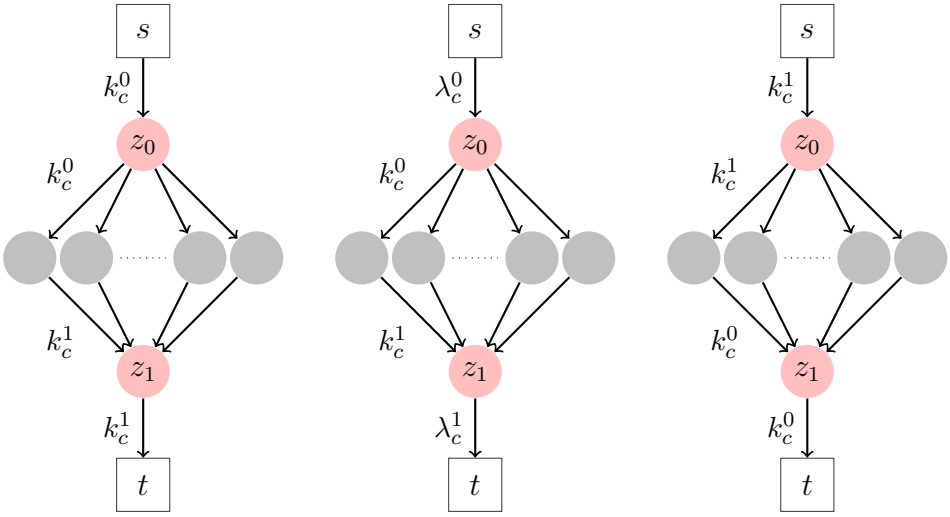
In the binary labelling case, by introducing auxiliary variables, we have:

$$\begin{aligned}f_c^0(\mathbf{x}_c) &= \min_{z_0}\left((\gamma_{c,max} - \gamma_{c,0})z_0 + k_c^0 \sum_{i\in c} x_i(1 - z_0) + \gamma_{c,0}\right) \\ f_c^1(\mathbf{x}_c) &= \min_{z_1}\left((\gamma_{c,max} - \gamma_{c,1})(1 - z_1) + k_c^1 \sum_{i\in c}(1 - x_i)z_1 + \gamma_{c,1}\right)\end{aligned} \tag{2.40}$$

Therefore, $\psi_c(\mathbf{x}_c)$ can be rewritten as:

$$\psi_c(\mathbf{x}_c) = \min(f_c^0(\mathbf{x}_c), f_c^1(\mathbf{x}_c)) = f_c^0(\mathbf{x}_c) + f_c^1(\mathbf{x}_c) - \gamma_{c,max}$$

The second equation is based on the constraint $2Q < |c|$, which indicates that $f_c^0(\mathbf{x}_c)$ and $f_c^1(\mathbf{x}_c)$ cannot be smaller than $\gamma_{c,max}$ at the same time. Plugging in



(a) P^n Potential (b) Robust P^n Potential (c) Label costs

Figure 2.6: From left to right: graph construction results for P^n potential, robust P^n potential and label cost terms. The three figures show the corresponding graphs for binary pairwise energies 2.36, 2.41, 2.44 respectively.

$f_c^0(\mathbf{x}_c)$ and $f_c^1(\mathbf{x}_c)$, and ignoring constant value, we have

$$\min_{z_0,z_1}\left(\lambda_c^0 z_0 + k_c^0 \sum_{i\in c} x_i(1-z_0) + \lambda_c^1(1-z_1) + k_c^1 \sum_{i\in c}(1-x_i)z_1\right) \tag{2.41}$$

where $\lambda_c^0 = \gamma_{c,max} - \gamma_{c,0}$ and $\lambda_c^1 = \gamma_{c,max} - \gamma_{c,1}$. The graph construction for the robust P^n potential term is shown in Figure 2.6(b).

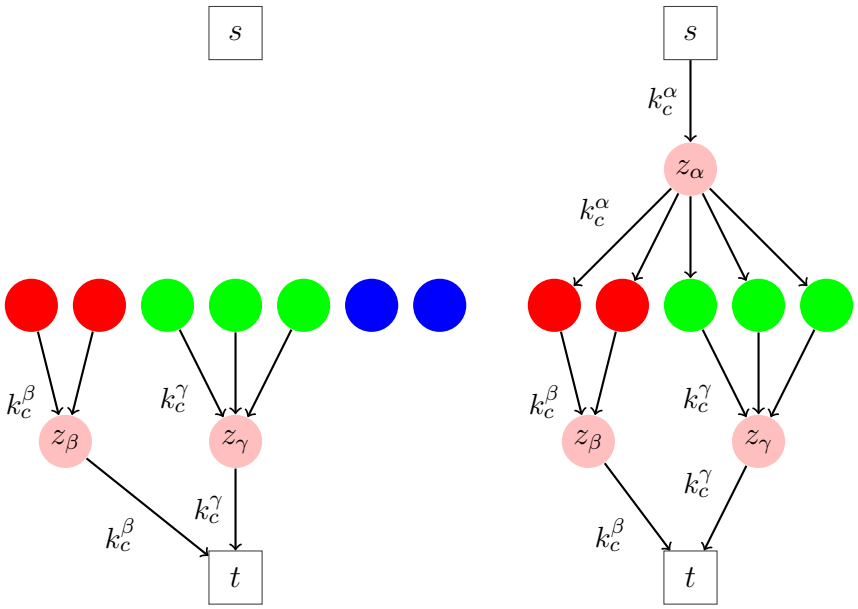
Label Costs

Label costs follows the principle of parsimony and encourages the data to be explained by as few models as possible. If two solutions are equally likely, then the simpler one, i.e. the one with fewer labels, is preferred. Specifically, we define label costs as the sum of the cost k_c^l over each used label l:

$$\psi_c(\mathbf{x}_c) = \sum_{l\in\mathcal{L}}\left(k_c^l(1 - \prod_{i\in c}\Delta(x_i \neq l))\right) \tag{2.42}$$

In binary cases, the above equation can be written as:

$$\psi_c(\mathbf{x}_c) = k_c^0 + k_c^1 - k_c^0\prod_{i\in c}\Delta(x_i = 1) - k_c^1\prod_{i\in c}\Delta(x_i = 0) \tag{2.43}$$



(a) Label α is present in current labelling (b) Label α not present

Figure 2.7: Graph construction for an α-expansion step of label costs term. (a): Currently used labels are α, β, γ. (b): Currently used labels are β, γ. Red, green, blue and pink nodes are variables with label β, γ, α and auxiliary variables respectively.

Based on the results from equations 2.34 and 2.35:

$$\psi_c(\mathbf{x}_c) = \min_{z_0, z_1} \left(k_c^1 z_0 + \sum_{i \in c} k_c^1 x_i (1 - z_0) + k_c^0 (1 - z_1) + \sum_{i \in c} k_c^0 (1 - x_i) z_1 \right) \tag{2.44}$$

Note that the above equation is very similar to 2.36, and the only difference is the coefficients. The graph construction for label costs is shown in Figure 2.6(c).

Alpha-Expansion for Label Costs

We now show how to construct the graph for one α-expansion step of label costs. Assuming the set of currently used labels for clique c is $\mathcal{L}_0$ and $\alpha \in \mathcal{L}_o$, the cost of move variables $\mathbf{t}_c$ is:

$$\psi_c(\mathbf{t}_c) = \sum_{\substack{l \in \mathcal{L}_0 \\ l \neq \alpha}} \left(k_c^l - k_c^l \prod_{i \in \mathcal{S}_l} \Delta(t_i = 1) \right) \tag{2.45}$$

where $\mathcal{S}_l = \{i : x_i = l\}$ is the set of variables currently labelled as l.

Using the same trick as above, $\psi_c(\mathbf{t}_c)$ can be written as:

$$\psi_c(\mathbf{t}_c) = \sum_{\substack{l \in \mathcal{L}_0 \\ l \neq \alpha}} \left(\min_{z_l} \left[k_c^l (1 - z_l) + \sum_{i \in \mathcal{S}_l} k_c^l (1 - t_i) z_l \right] \right) \tag{2.46}$$

Similarly, if label $\alpha \notin \mathcal{L}_0$, we have:

$$\begin{aligned} \psi_c(\mathbf{t}_c) &= k_c^\alpha + \sum_{l \in \mathcal{L}_0} k_c^l - \sum_{l \in \mathcal{L}_0} k_c^l \prod_{i \in \mathcal{S}_l} \Delta(t_i = 1) - k_c^\alpha \prod_{i \in c} \Delta(t_i = 0) \\ &= \min_{\substack{z_l, z_\alpha \\ l \in \mathcal{L}_0}} \left(\sum_{l \in \mathcal{L}_0} k_c^l (1 - z_l) + \sum_{l \in \mathcal{L}_0} \sum_{i \in \mathcal{S}_l} k_c^l (1 - t_i) z_l + k_c^\alpha z_\alpha + \sum_{i \in c} k_c^\alpha t_i (1 - z_\alpha) \right) \end{aligned}$$

Figure 2.7 shows the α-expansion move graph construction for label costs. Therefore, higher order label costs can be efficiently minimized with the α-expansion algorithm. As shown in section 2.2, these label costs have been used in PEARL for geometric multi-model fitting applications. Interestingly, a more general case of label costs, co-occurrence statistics cost, was proposed in parallel by Ladicky *et al.* [71], and successfully used in semantic segmentation.

Overlapping Models

Russell *et al.* [103] proposed the use of overlapping models as a way to enforce the constraint that adjacent points at the boundary of a model should be explained by both models, instead of just one (see figure 2.8). In the paper, this constraint is used as an alternative to pairwise smoothness to encourage smooth model transition between neighbouring points.

Assuming a set of models $\mathcal{M}$, in contrast to standard labelling problem where each point is assigned only to one model, when using overlapping models each point p can be assigned to a subset of models $\mathbf{x}_p$. This assignment has to satisfy two constraints: *(i)* Each point has to belong to at least one model. *(ii)* Adjacent models must overlap, i.e. they must explain some of the same points. To formulate these overlapping constraints, the concept of interior points is introduced. A point p is an interior point of model α if and only if all the neighbours of p also belong to model α, *but not necessarily as interior points*. We use $\mathbf{I}$ to denote the interior model labelling for all the points, and we define I_α, the interior of model α, as the set of all points whose neighbours also belong to model α. With a slight abuse of notation we use I_p to refer to the interior label assigned to point p. Figure 2.8 gives an illustration of interior points.

The problem of seeking the best solution can be formulated as:

$$\underset{\mathbf{x}\in(2^{\mathcal{M}})^P}{\operatorname{argmin}} C(\mathbf{x}) = \sum_{p\in\mathcal{P}} \left(\sum_{\alpha\in\mathbf{x}_p} U_p(\alpha) \right) \tag{2.47}$$

subject to the constraints

$$\forall p \in \mathcal{P}\ \exists \alpha : p \in I_\alpha \tag{2.48}$$

and

$$\forall q \in \mathcal{N}_p \wedge q \in I_\alpha \Longrightarrow \alpha \in \mathbf{x}_p \tag{2.49}$$

where $U_p(\alpha)$ is the unary cost of assigning point p to model α. Russell *et al.* [103] proved that the above optimization problem can be reformulated as an optimization over the interior labelling $\mathbf{I}$. Moreover, for a minimal cost solution $\mathbf{x}$, the interior label assigned to each point (I_p) is unique. This leads to the following unconstrained minimization problem:

$$\underset{\mathbf{I}\in\mathcal{M}^P}{\operatorname{argmin}} C(\mathbf{I}) = \sum_{p\in\mathcal{P}} \left(\sum_{\bigcup_{q\in\mathcal{N}_p}\{\alpha:q\in I_\alpha\}} U_p(\alpha) \right) = \sum_{p\in\mathcal{P}} \psi_p(\mathbf{I}) \tag{2.50}$$

where $\psi_p(\mathbf{I})$ is the cost of interior labellings $\mathbf{I}$ over $\mathcal{N}_p$ and can be rewritten as:

$$\psi_p(\mathbf{I}) = \sum_{\bigcup_{q\in\mathcal{N}_p}\{\alpha:q\in I_\alpha\}} U_p(\alpha) = \sum_{\alpha\in\mathcal{L}} U_p(\alpha) \left(1 - \prod_{q\in\mathcal{N}_p} \Delta(I_q \neq \alpha) \right) \tag{2.51}$$

Note that the higher order cost $\psi_p(\mathbf{I})$ is a label cost over the neighbourhood $\mathcal{N}_p$. Unlike the label costs introduced in the last section, this cost is local, i.e. depends only on the interior labellings of the neighbourhood $\mathcal{N}_p$. For each label α present in the neighbourhood, we pay a cost $U_p(\alpha)$. It should be pointed out that the neighbourhood $\mathcal{N}_p$ defined here also includes point p itself.

The graph construction for an α-expansion step of this overlap cost is also straightforward. It is very similar to the case of label costs, as shown in 2.7, and the only difference is that $\psi_p(\mathbf{I})$ is defined within a local neighbourhood N_p, with model fitting cost $U_p(\alpha)$ as label cost for label α. Figure 2.9 shows the graph for an α-expansion step of the cost term $\psi_p(\mathbf{I})$.

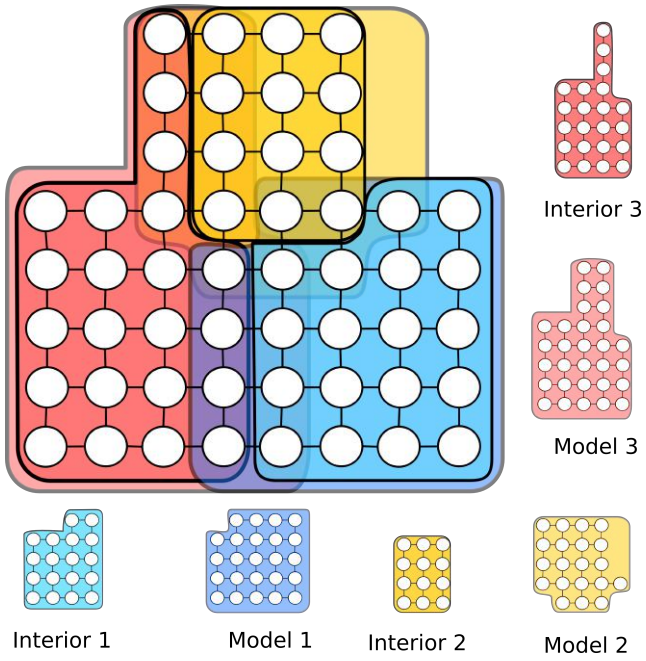
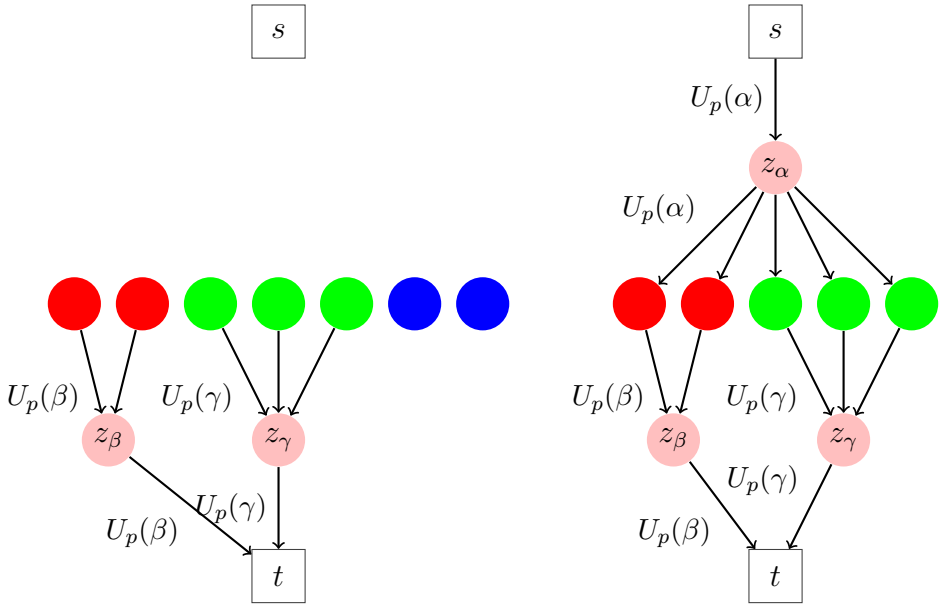


Figure 2.8: An illustration of model assignments that satisfies overlapping constraints 2.48 and 2.49. Figure from [103].



(a) Label α is present in local neighbourhood (b) Label α not present

Figure 2.9: Graph construction of an α-expansion step of the cost term $\psi_p(\mathbf{I})$. (a): Currently used labels within the neighbourhood $\mathcal{N}_p$ are α, β, γ. (b): Currently used labels within the neighbourhood $\mathcal{N}_p$ are β, γ. Note the similarity with the graph for label costs 2.7. The only difference is the weights of the edges of the graph.

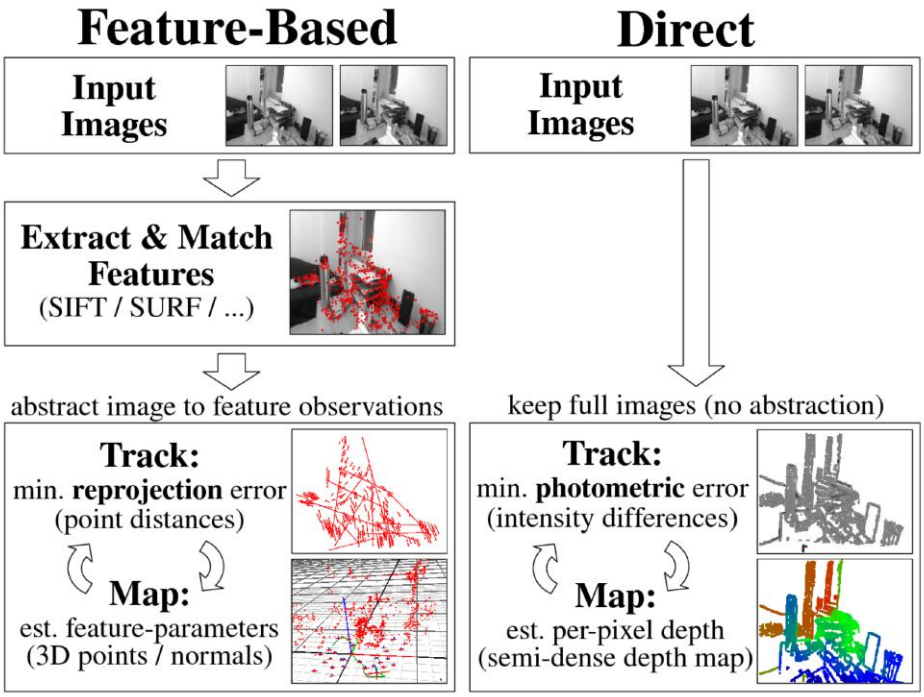


Figure 2.10: An comparison between feature-based and direct methods. Figure from [37].

2.4 Direct Dense Tracking

2.4.1 Direct Methods or Indirect Methods

Direct methods [57, 87, 88, 144, 37, 86, 36] refer to the approach of inferring unknown model parameters directly from raw image intensity measurements without any intermediate steps, while indirect methods [124, 114, 30, 63, 45, 84] usually decompose the problem into two separate steps, first solving an intermediate correspondence problem and then computing model parameters with the established correspondences. Note that for indirect methods the original input data is usually discarded in the second step and all the computation is based on pre-computed correspondences only.

Both of these two methods have been widely used by the computer vision community. Indirect methods typically involve extracting and matching a set of sparse feature points [84], and then working on these sparse feature correspondences while direct methods are often based on the minimization of a dense per-pixel cost function. However, other options exist and one could use dense correspondences within indirect methods [45] or use direct methods only on sparse points [36]. A comparison between feature-based and direct methods

is shown in Figure 2.10.

There has been a debate as to which of these two methods is better, dense direct methods or sparse feature-based methods. We refer readers to [57, 124, 126] for excellent discussions about this topic. Both methods have their own advantages and the answer will typically depend on the specific application. The main arguments favouring feature-based methods include: *(i)* invariant features are robust to illumination and viewpoint changes and therefore could be matched across a wide range of conditions; *(ii)* computation based on sparse features is often much more efficient than minimizing a dense per-pixel cost; *(iii)* no need for an appropriate initialization as the problem can be solved linearly in a lot of cases.

In contrast, dense direct methods are preferred because: *(i)* dense methods utilize all the input information and should be able to generate more accurate results; *(ii)* direct methods solve the model parameters and correspondences simultaneously instead of dividing them into two separate steps; *(iii)* the method still works when there are not enough feature points, for example when the image is less textured, and degrades gracefully under deteriorating visual conditions, such as blurring due to motion or out of focus.

However, there are two main criticisms to direct approaches. First, the common assumptions made about the model, such as the frequently used brightness constancy assumption, are not accurate. Second, the resulting minimization is non-convex and difficult to optimize. This optimization problem is usually solved with an iterative linearization strategy where the linearization is based on local gradient information. As such, the convergence radius of this problem is small and the method will often need a decent initialization for the algorithm to converge. This explains the fact that although direct methods have been widely used for tracking and real time SLAM systems recently, applications to unstructured image collection, such as structure from motion from web images, are still dominated by feature-based methods.

In fact, the advantages of these two approaches can be combined together by first computing an initial estimate using feature-based approaches and then refining the solution with more accurate direct methods.

In this section we are interested in describing the fundamentals of dense direct methods for monocular camera tracking by minimizing a photometric cost. In particular, we assume that a 3D model of the scene is available and the goal is to track the motion of the camera as it moves across the scene.

2.4.2 Dense Visual Tracking

Assuming a pre-computed 3D template mesh with vertices $\hat{\mathbf{V}} = \{\mathbf{V_i}\}$ and colours $\hat{\mathbf{I}} = \{\mathbf{I_i}\}$, we would like to track the camera motion in an input video sequence. For simplicity, we assume that the mesh is rigid and there is no non-rigid deformation over the whole sequence.

Direct methods formulate the problem of camera tracking as an energy minimization, where the energy is given by the sum of the squared differences between the colour of the template mesh vertex and the corresponding point in the input image, the so called photometric cost. In other words, we assume that the brightness constancy constraint is satisfied and minimize the colour discrepancy between the 3D model and the input image.

Let us denote $\tilde{\mathbf{V}}_i = [\mathbf{V}_i^T, 1]^T$ as the homogeneous representation of vertex $\mathbf{V}_i = [x_i, y_i, z_i]^T$ and $\mathbf{T}_{cr}$ as the transformation matrix between the reference frame and current frame, i.e.

$$\mathbf{T}_{cr} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.52}$$

First, for each frame, a visibility mask is computed for all the vertices by rendering the mesh using the tracking result from the previous frame. Only visible vertices are then considered to estimate the current camera pose. Assuming the set of visible vertices is V, the dense photometric cost can then be written as:

$$E(\mathbf{T}_{cr}) = \sum_{i \in V} \rho(\|\mathbf{I}(\pi(\mathbf{K}\mathbf{T}_{cr}\tilde{\mathbf{V}}_i)) - \mathbf{I}_i\|_2^2) \tag{2.53}$$

where $\mathbf{K}$ is the camera intrinsic matrix (a 3×4 matrix with an added zero fourth column), $\pi(\cdot)$ is a 3D to 2D projection operator such that:

$$\pi\left(\begin{bmatrix} u \\ v \\ w \end{bmatrix}\right) = \begin{bmatrix} \frac{u}{w} \\ \frac{v}{w} \end{bmatrix} \tag{2.54}$$

and $\rho(\cdot)$ is a robust loss function for handling outliers. For simplicity reasons, we assume that $\rho(\cdot)$ is an identity loss function in this section. We refer readers to Zach [140] for detailed discussions on handling different robust functions.

We utilize the commonly used axis-angle representation $\mathbf{w}$ for rotation $\mathbf{R}$. Then the relationship between $\mathbf{w}$ and $\mathbf{R}$ is described by the exponential

mapping:

$$\mathbf{R} = \exp(\hat{\mathbf{w}}) = e^{\hat{\mathbf{w}}} \tag{2.55}$$

where $\hat{\mathbf{w}}$ is the skew-symmetric matrix of vector $\mathbf{w}$:

$$\hat{\mathbf{w}} = [\mathbf{w}]_\times = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \tag{2.56}$$

Similarly, the rigid transformation $\mathbf{T}_{cr}$ can be described using the exponential mapping from rotation $\mathbf{w}$ and translation $\mathbf{t}$:

$$\mathbf{T}_{cr} = \exp(\begin{bmatrix} \hat{\mathbf{w}} & \mathbf{t} \\ 0 & 0 \end{bmatrix}) = e^{\hat{\xi}} \tag{2.57}$$

where ξ and $\hat{\xi}$ are defined as:

$$\xi = \begin{bmatrix} \mathbf{w} \\ \mathbf{t} \end{bmatrix} \qquad \hat{\xi} = \begin{bmatrix} \hat{\mathbf{w}} & \mathbf{t} \\ 0 & 0 \end{bmatrix} \tag{2.58}$$

With a slight abuse of notation, we also use $\exp(\xi)$, which should be understood as $\exp(\hat{\xi})$ when necessary.

Nonlinear Least-Squares Optimization

Minimization of the photometric cost, as shown in 2.53, is a standard nonlinear least-squares problem and is usually solved with the widely used Levenberg-Marquardt (LM) algorithm.

The LM algorithm is a popular approach for solving nonlinear least-squares problems. It employs an iterative minimization scheme and instead of solving the difficult original problem, it solves a sequence of approximate but easier problems. At each step, it linearizes the cost function based on the jacobian information at the current estimate and solves for the optimal incremental parameter update $\Delta\xi$ by solving a least-squares problem. LM can be seen as an interpolation between the gradient descent and Gauss-Newton optimization methods. When the parameters are far from the optimal value, it behaves like gradient descent methods. However, when the estimate is close to the solution, it acts like Gauss-Newton methods and converges quickly.

LM is used to minimize costs of the form:

$$E(\xi) = \|f(\xi) - \mathbf{y}\|_2^2 = \|\mathbf{r}\|_2^2 \tag{2.59}$$

where $\mathbf{r}$ denotes the residual vector between the prediction $f(\xi)$ and the measurement $\mathbf{y}$. At each iteration, the non-linear function $f(\cdot)$ is linearized at the current estimate ξ using $f(\xi + \Delta\xi) = f(\xi) + \mathbf{J}\Delta\xi$. This leads to the following approximate objective:

$$\underset{\Delta\xi}{\operatorname{argmin}} \|\mathbf{y} - f(\xi) + \mathbf{J}\Delta\xi\|_2^2 = \|\mathbf{J}\Delta\xi - \mathbf{r}\|_2^2 \tag{2.60}$$

where $\mathbf{J}$ and $\mathbf{r}$ are the Jacobian matrix and the residual vector at the current estimation respectively. This minimization problem is essentially a least-squares problem and can be solved via the so-called normal equations:

$$\mathbf{J}^T\mathbf{J}\Delta\xi = \mathbf{J}^T\mathbf{r} \tag{2.61}$$

Solving the normal equations is referred to as a Gauss-Newton iteration. However, the linear approximation may not be accurate when the estimate is far from the optimal solution and the iteration may not decrease the energy. To make the approach better behaved, the LM algorithm uses a trust region strategy to control the size of the update $\Delta\xi$, or equivalently adds a damping factor into $\mathbf{J^T J}$ matrix. In each step, LM method solves a damped normal equation:

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{D})\Delta\xi = \mathbf{J}^T\mathbf{r} \tag{2.62}$$

Where, λ is a coefficient adapted to the quality of current solution and $\mathbf{D}$ is the damping matrix, which is usually an identity matrix or the diagonal of the $\mathbf{J}^T\mathbf{J}$ matrix.

In practice, the algorithm will terminate when there is no significant decrease in the cost or the number of iterations is larger than a specified number. In general, LM has a fast convergence and the method will terminate after a few iterations.

Jacobian Computation

We now discuss how to compute the jacobian $\mathbf{J}$ for the introduced photometric cost $E(\mathbf{T}_{cr})$, as shown in 2.53. Given the current estimate ξ, we write the cost explicitly in terms of the update $\Delta\xi$:

$$E(\Delta\xi) = \sum_{i \in V} \|\mathbf{I}(\pi(\mathbf{K}\exp(\xi)\exp(\Delta\xi)\tilde{\mathbf{V}}_i)) - \mathbf{I}_i\|_2^2 = \sum_{i \in V} \|\mathbf{r}_i\|_2^2 \tag{2.63}$$

Let us denote $\mathbf{P}_i = \mathbf{K}\exp(\xi)\exp(\Delta\xi)\tilde{\mathbf{V}}_i$ and $\mathbf{u}_i = \pi(\mathbf{P}_i)$. The jacobian for i^{th}

point can be computed based on chain rule:

$$\mathbf{J}_i = \frac{\partial \mathbf{r}_i}{\partial \Delta\xi}\Big|_{\Delta\xi=0} = \frac{\partial \mathbf{I}(\mathbf{u}_i)}{\partial \mathbf{u}_i}\Big|_{\mathbf{u}_i=\pi(\mathbf{K}\exp(\xi)\tilde{\mathbf{V}}_i} \frac{\partial \mathbf{u}_i}{\partial \mathbf{P}_i}\Big|_{\mathbf{P}_i=\mathbf{K}\exp(\xi)\tilde{\mathbf{V}}_i} \frac{\partial \mathbf{P}_i}{\partial \Delta\xi}\Big|_{\Delta\xi=0} \tag{2.64}$$

The first term is the jacobian of image colour with respect to 2D projections of 3D points. In general, the projections are not necessarily integers, so bilinear interpolation is used to obtain image intensity values for arbitrary positions. Given a 2D projection $\mathbf{u}_i = [u_i, v_i]^T$, the corresponding image value at channel c is retrieved as:

$$\mathbf{I}^c(\mathbf{u}_i) = \sum_{m=1}^{H}\sum_{n=1}^{W} I^c_{mn} \max(0, 1 - |u_i - n|) \max(0, 1 - |v_i - m|) \tag{2.65}$$

where H and W are the image height and width respectively. I^c_{mn} is the c^{th} channel image value at integer pixel location (m, n). Differentiating $\mathbf{I}^c(\mathbf{u}_i)$ with respect to u_i and v_i, we have:

$$\frac{\partial \mathbf{I}^c(\mathbf{u}_i)}{\partial u_i} = \sum_{m=1}^{H}\sum_{n=1}^{W} I^c_{mn} \max(0, 1 - |v_i - m|) \begin{cases} 0 & \text{if } |u_i - n| > 1 \\ 1 & \text{if } (n-1) \leq u_i \leq n \\ -1 & \text{if } n < u_i \leq (n+1) \end{cases} \tag{2.66}$$

$$\frac{\partial \mathbf{I}^c(\mathbf{u}_i)}{\partial v_i} = \sum_{m=1}^{H}\sum_{n=1}^{W} I^c_{mn} \max(0, 1 - |u_i - n|) \begin{cases} 0 & \text{if } |v_i - m| > 1 \\ 1 & \text{if } (m-1) \leq v_i \leq m \\ -1 & \text{if } m < v_i \leq (m+1) \end{cases} \tag{2.67}$$

The second term is just the jacobian of projection operator $\pi(\cdot)$. From equation 2.54, it is straightforward to compute $\frac{\partial \mathbf{u}_i}{\partial \mathbf{P}_i}$:

$$\frac{\partial \mathbf{u}_i}{\partial \mathbf{P}_i}\Big|_{\mathbf{P}_i=[u,v,w]^T} = \begin{bmatrix} \frac{1}{w} & 0 & -\frac{u}{w^2} \\ 0 & \frac{1}{w} & -\frac{v}{w^2} \end{bmatrix} \tag{2.68}$$

The last term can be derived as:

$$\begin{aligned} \frac{\partial \mathbf{P}_i}{\partial \Delta\xi}\Big|_{\Delta\xi=0} &= \frac{\partial \mathbf{K}\exp(\xi)\exp(\Delta\xi)\tilde{\mathbf{V}}_i}{\partial \Delta\xi}\Big|_{\Delta\xi=0} \\ &= \mathbf{K}\exp(\xi)\frac{\partial \exp(\Delta\xi)}{\partial \Delta\xi}\Big|_{\Delta\xi=0}\tilde{\mathbf{V}}_i \\ &= \mathbf{K}\exp(\xi)[\mathbf{G}_1\tilde{\mathbf{V}}_i\ \mathbf{G}_2\tilde{\mathbf{V}}_i\ \mathbf{G}_3\tilde{\mathbf{V}}_i\ \mathbf{G}_4\tilde{\mathbf{V}}_i\ \mathbf{G}_5\tilde{\mathbf{V}}_i\ \mathbf{G}_6\tilde{\mathbf{V}}_i] \end{aligned} \tag{2.69}$$

where $\mathbf{G}_i, i = 1, \cdots, 6$ are the generators of the exponential map, defined as:

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{G}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{G}_2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\mathbf{G}_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{G}_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{G}_6 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.70}$$

Assuming point $\tilde{\mathbf{V}}_i = [x_i, y_i, z_i, 1]^T$, the multiplication between generators and $\tilde{\mathbf{V}}_i$ can be simply written as:

$$[\mathbf{G}_1\tilde{\mathbf{V}}_i \ \mathbf{G}_2\tilde{\mathbf{V}}_i \ \mathbf{G}_3\tilde{\mathbf{V}}_i \ \mathbf{G}_4\tilde{\mathbf{V}}_i \ \mathbf{G}_5\tilde{\mathbf{V}}_i \ \mathbf{G}_6\tilde{\mathbf{V}}_i] = \begin{bmatrix} 0 & z_i & -y_i & 1 & 0 & 0 \\ -z_i & 0 & x_i & 0 & 1 & 0 \\ y_i & -x_i & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.71}$$

Having computed the jacobian $\mathbf{J}_i$ for all the P visible points, the jacobian $\mathbf{J}$ is simply a stack of all the $\mathbf{J}_i$, i.e $\mathbf{J} = [\mathbf{J}_1^T, \mathbf{J}_2^T, \cdots, \mathbf{J}_P^T]^T$.

Parameter Update

After obtaining optimal incremental update $\Delta\xi$ by solving the linear equation 2.62, we could update current parameter ξ. It is important to note that this update should be computed by compositional exponential mapping and not simple addition, in other words we have:

$$\exp(\xi) \leftarrow \exp(\xi)\exp(\Delta\xi) \quad \text{or} \quad \xi \leftarrow \ln(\exp(\xi)\exp(\Delta\xi)) \tag{2.72}$$

Chapter 3

Literature Review

In this chapter we review the literature about 3D reconstruction of dynamic scenes in computer vision. We classify non-rigid capturing approaches into three different categories based on the amount of input information (single camera, RGB-D camera or multiple cameras) and prior knowledge (an offline trained class specific model or a 3D template shape) of the dynamic scene: *(i)* Non-rigid structure from motion (NRS*f*M) methods where the reconstruction is based on monocular sequence only, without assuming any scene-specific priors. *(ii)* Shape-from-template methods where a template shape of the scene to be reconstructed is available; *(iii)* Model-based methods where the reconstruction is based on a model tailored and trained for a specific class of objects.

3.1 Non-rigid Structure from Motion

3D reconstruction of dynamic scenes from monocular sequence is an extremely challenging problem in computer vision. Imagine a scene containing an unknown number of moving and possibly deforming objects, a successful reconstruction of this dynamic scene would need to first correctly segment the objects and then reconstruct the 3D shape of each object. This simultaneous segmentation and reconstruction problem can be naturally formulated as a multi-model fitting problem. However, accurately modelling the deformation of generic objects is very difficult. In this section, we review some of the state-of-the-art methods in NRS*f*M. In particular, we introduce video and motion segmentation methods first, then discuss methods for reconstructing a pre-segmented articulated or non-rigid object and methods that can deal with multiple objects or general scenes.

3.1.1 Motion Segmentation

Providing robust solutions to video and motion segmentation is a fundamental problem in computer vision and often a preliminary step towards 3D reconstruction. Most works in dynamic scene reconstruction [25, 41, 111, 91] follow a pipeline approach where feature point tracks or dense optical flow is estimated, followed by a motion segmentation step that clusters trajectories into different independent motions before reconstructing each of the objects independently in 3D.

As is the case with simultaneous segmentation and reconstruction, motion segmentation can also be formulated as a multi-model fitting problem. However, for the purpose of motion segmentation, we only consider segmenting the scene into different motion coherent objects, which can be done without estimating the 3D shape. In fact, most motion segmentation methods only use motion cues rather than the 3D shape of the object. Specifically, most widely used cues in motion segmentation include spatial proximity, motion similarity, motion subspace constraints, *etc*. Different motion segmentation methods tend to use different cues and the way how these cues are exploited may also differ.

As discussed in the multi-model fitting section 2.2 in previous chapter, multi-model fitting approaches can be broadly classified into four main categories. Motion segmentation methods also fall into these four categories. In this section we focus on the two categories that are most widely used in practice: spectral clustering based methods and energy based multi-model fitting methods. We now review the methods in each category.

Spectral Clustering based Methods

Spectral clustering is a widely used approach for clustering. It first computes the similarity between each pair of points and builds an affinity matrix, and then performs clustering based on the spectrum (eigenvalues and eigenvectors) of the affinity matrix.

One of the first motion segmentation methods for multi-rigid scenes was proposed by Costeira *et al.* [25]. They introduced the concept of Shape Interaction Matrix (SIM) $\mathbf{Q}$ by decomposing the measurement matrix W using SVD:

$$\mathbf{W} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathbf{T}} \quad \text{and} \quad \mathbf{Q} = \mathbf{V}\mathbf{V}^{\mathbf{T}} \tag{3.1}$$

Importantly, they found that the element of $\mathbf{Q}$ has the property that $\mathbf{Q}_{ij} = 0$ if point i and point j belong to two different objects. In other words, matrix $\mathbf{Q}$ encodes the motion membership of each point and can be used as affinity

matrix for spectral clustering. However, to construct the SIM matrix $\mathbf{Q}$ we need to know the number of objects a priori. Moreover, equation $\mathbf{Q}_{ij} = 0$ is very sensitive to noise and is only valid for independent motions. Recently, this method has been revisited and a more robust solution has been proposed by Pan *et al.* [60].

Generalized Principal Component Analysis (GPCA) [132] is an algebraic framework for modelling and segmenting data lying in different subspaces. It can be seen as a generalisation of Principal Component Analysis (PCA) to the case of multiple linear subspaces. The main idea behind GPCA is that a collection of n subspaces can be fit with a set of n polynomials, whose derivatives at a particular point encode the normal vector to the subspace containing that point. Therefore, by clustering these normal vectors we could segment the data into different subspaces. Specifically, one can define similarity measures between points using the normal vector or subspace principal angles and perform spectral clustering on the affinity matrix. However, although GPCA is computationally cheap when the number of points is small, its complexity increases exponentially with the number of subspaces and the method does not scale to large number of points.

Sparse subspace clustering (SSC) [35] is a method based on the idea that points lying in a subspace can be represented as a sparse linear combination of other points in the same subspace. By enforcing the sparsity constraint on the self-representation coefficient matrix, one could obtain a sparse representation where each point is represented as a linear combination of the points in the same subspace. This sparse representation matrix can then be used to build an affinity matrix for spectral clustering. This method is originally used for affine motion segmentation [99] and has later been extended to the perspective case [74].

Low rank representation (LRR) is a method very similar to SSC, except that it seeks a low rank representation rather than a sparse representation. Interestingly, it can be shown [131] that LRR provides a theoretical justification for the Costeira and Kanade algorithm [25].

Unlike previous methods which cast motion segmentation as a motion subspace clustering problem, approaches such as Brox and Malik's work [18] exploit the consistency of point trajectories over time and can deal with non-rigid motion. In particular, they define distance between trajectories as the maximum difference of their motion over time, build an affinity matrix from these pairwise distances and then run spectral clustering on this matrix. Note

that the problem of occlusion or missing data is naturally handled in this approach. Moreover, this kind of method is far more general than methods based on subspace clustering which can only handle rigid motions in theory.

Energy based Multi-model Fitting Methods

In contrast to spectral clustering based methods, energy based multi-model fitting methods formulate motion segmentation as an energy minimization problem. These methods seek the best solution by minimizing an energy which objectively encodes the "goodness" of a solution. The goal of these methods is to find the best motion model parameters and the assignment of points to these models. For motion segmentation applications, commonly used motion models include homography, fundamental matrix, linear subspace, *etc*. Particularly, homography and fundamental matrix are usually used for two frame motion segmentation, while motion subspace is often used for multi-frame case.

Two popular energy based methods for mixture models, Expectation Minimization (EM) and K-means, have also been used in geometric multi-model fitting problems in computer vision. However, these kinds of methods have some limitations. In general, they are sensitive to initializations and not robust to outliers. Moreover, EM methods are based on probabilistic model representations, which are not always well-suited for geometric problems. For instance, EM algorithm allow points to be softly assigned to multiple distributions simultaneously, while in the real-world geometric vision problems points typically are exclusively assigned to a single model. Also EM and K-means type methods deal with each point independently and do not exploit spatial smoothness priors during clustering. Furthermore, EM and K-means methods need the number of models to be known in advance.

Recently, PEaRL [58] was proposed as a principled framework for geometric multi-model fitting problems. As discussed in section 2.2 in the previous chapter, this method has been successfully used in many geometric multi-model fitting problems, such as line fitting, homography estimation, motion segmentation, *etc*. It provides a powerful framework for multi-model fitting problems by combining data sampling and simultaneous model fitting and point assignment. In contrast with EM and K-means, PEaRL naturally models geometric spatial regularity and encodes MDL (minimum description length) prior that encourages the number of used models to be small. Besides, by generating a large number of initial proposals using local sampling in the first step, PEaRL is insensitive to initialization and can converge to a good solution quickly and reliably.

More recently, Ranftl *et al.* [98] introduced an efficient convex formulation

for two frame motion segmentation problem. Their method is closely related with PEARL, but relaxes the discrete segmentation problem to a soft assignment and formulates it as a continuous convex optimization problem. Another important difference is that their convex formulation does not have label costs and as such their results are often over-segmented. Nonetheless, with an efficient GPU-based implementation, their method is much faster than PEARL and has been successfully used for segmenting dense optical flow.

3.1.2 NRSFM for a Single Object

Two main successful approaches dominate the modelling of deformable shapes in NRS*f*M literature: low-rank based methods and piecewise modelling. Low-rank based methods assume that the global deformation of an object can be well modelled by deformations of low rank. This prior is often used in two different ways, either the deformed shape at each frame is explicitly represented as a linear combination of basis shapes or by forcing the concatenation of shapes over all the frames to be a low-rank matrix. In constrast, piecewise methods model local deformation directly and assume that by dividing an object into smaller enough pieces, each part can be well approximated with a simple model, such as planes or rigid objects.

Low-Rank Based Methods

The idea of using linear basis shape model for NRS*f*M was first introduced by Bregler *et al.* [17]. In their seminal paper, they proposed to represent the 3D deforming shape of the object over each frame as a linear combination of K basis shapes. In other words, we have:

$$\mathbf{S}_f = \sum_{k=1}^{K} l_k^f \mathbf{B}_k \tag{3.2}$$

where the $3 \times P$ matrix $\mathbf{S}_f$ is the 3D shape of the deforming object at frame f, $\mathbf{B}_k, k = 1, \cdots, K$ are the shape bases and l_k^f are the linear combination coefficients. Based on this assumption, we could rewrite the rigid factorisation equation 2.11 as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{R}^1 & & \\ & \ddots & \\ & & \mathbf{R}^F \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_F \end{bmatrix} = \underbrace{\begin{bmatrix} l_1^1\mathbf{R}^1 & \cdots & l_K^1\mathbf{R}^1 \\ \vdots & \ddots & \vdots \\ l_1^F\mathbf{R}^F & \cdots & l_K^F\mathbf{R}^F \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_K \end{bmatrix}}_{\mathbf{B}} \tag{3.3}$$

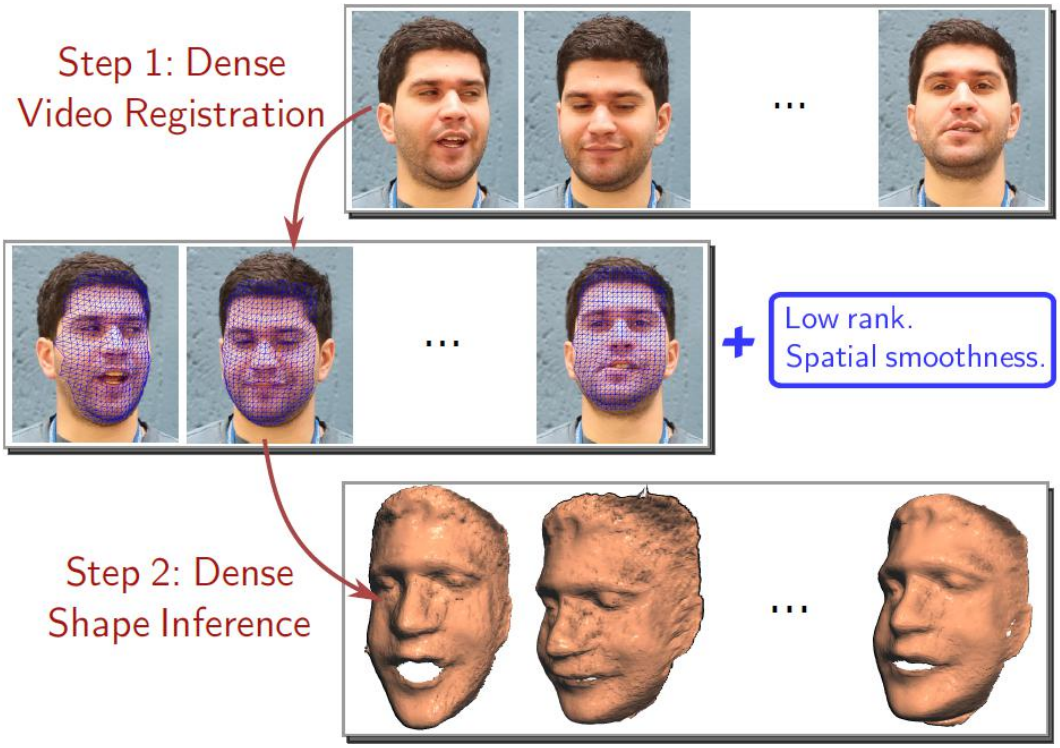
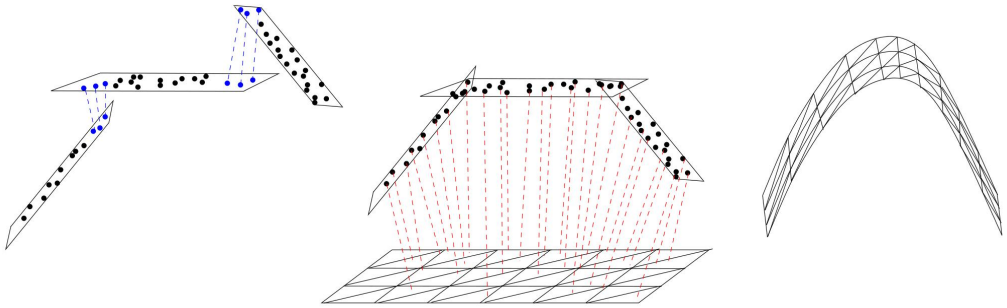


Figure 3.1: Proposed pipeline for dense NRS*f*M [45]. The pipeline consists of two steps: first dense long-term 2D trajectories are computed via a multi-frame optical flow method [46], then based on pre-computed dense trajectories, deforming dense 3D shape can be recovered by minimizing a continuous variational energy. Figure from [43].

Therefore, the $2F \times P$ measurement matrix $\mathbf{W}$ can be decomposed as the product of a $2F \times 3K$ matrix $\mathbf{M}$ and a $3K \times P$ matrix $\mathbf{B}$. In addition, the matrix $\mathbf{M}$ is constructed from rotation matrices $\mathbf{R}^f, f = 1, \cdots, F$ and has a repetitive structure. These observations form the basis of low-rank based methods for NRS*f*M. Similar to rigid case, non-rigid factorisation methods also consist of two steps. First, the measurement matrix $\mathbf{W}$ is decomposed into two low-rank matrices, a $2F \times 3K$ matrix $\hat{\mathbf{M}}$ and a $3K \times P$ matrix $\hat{\mathbf{B}}$. Note that this decomposition is not unique, since for any invertible matrix Q, we have $\mathbf{W} = \hat{\mathbf{M}}\hat{\mathbf{B}} = \hat{\mathbf{M}}\mathbf{Q}\mathbf{Q}^{-1}\hat{\mathbf{B}}$. In the second step, matrix $\hat{\mathbf{M}}$ is then upgraded to $\mathbf{M} = \hat{\mathbf{M}}\mathbf{Q}$ by finding the correction matrix $\mathbf{Q}$ that gives the correct form of $\mathbf{M}$. This second step is usually referred to as metric upgrade step, as the affine ambiguity in the decomposition is resolved by enforcing the constraints on matrix $\mathbf{M}$.

Let $3K \times 3$ matrix $\mathbf{Q}_k$ be the k^{th} triad of columns of the correction matrix $\mathbf{Q}$, i.e. $\mathbf{Q}_k = \mathbf{Q}_{:,3k-2:3k}$. From equation 3.3, the constraints on matrix $\mathbf{M}$ can



(a) (b) (c)

Figure 3.2: Algorithm pipeline. (a) Image patches are reconstructed independently up to a scale ambiguity. (b) Based on shared points between overlapping patches (blue points), they recover globally consistent scales for all patches and stitch all the patch reconstructions into a single globally consistent shape. (c) Finally, a triangulated mesh is fitted to the resulting 3D point cloud. Figure from [129].

be written as:

$$\begin{aligned} \hat{\mathbf{M}}_{2f-1:2f}\mathbf{Q}_k\mathbf{Q}_k^T\hat{\mathbf{M}}_{2f-1:2f}^T &= (l_k^f)^2\mathbf{I}_{2\times 2} \\ \hat{\mathbf{M}}_{2f-1:2f}\mathbf{Q}_i\mathbf{Q}_j^T\hat{\mathbf{M}}_{2f-1:2f}^T &= l_i^f l_j^f\mathbf{I}_{2\times 2} \end{aligned} \tag{3.4}$$

for any $f \in \{1, 2, \cdots, F\}$ and $i, j, k \in \{1, 2, \cdots, K\}, i \neq j$.

Many methods [15, 137, 16, 28] for finding the rectification matrix $\mathbf{Q}$ satisfying above constraints have been proposed during the last decade. In particular, Dai *et al.* [28] proposed a convex formulation for estimating the Gram matrix $\mathbf{G}_k = \mathbf{Q}_k\mathbf{Q}_k^T$ by relaxing the rank 3 constraint of $\mathbf{G}_k$ using its trace norm approximation. However, this method requires the number of basis shapes to be known in advance, and does not scale to large number of points. Later Garg *et al.* [45] proposed a novel energy minimization framework for NRS*f*M. Instead of representing the deforming shape as a linear combination of fixed number of basis shapes explicitly, their method encodes the low-rank prior implicitly as a trace norm term in the energy. Moreover, their variational formulation is able to incorporate an edge-preserving spatial smoothness prior into the low-rank framework and they are able to reconstruct a dense 3D shape with an efficient GPU implementation of their algorithm. Figure 3.1 shows the pipeline of Garg *et al.*'s dense non-rigid reconstruction method and the deforming mesh results obtained by this method.

Piecewise Modelling Methods

Low-rank based methods have been successfully used for modelling deforming objects, such as faces, moving backs, beating hearts, *etc*. However,

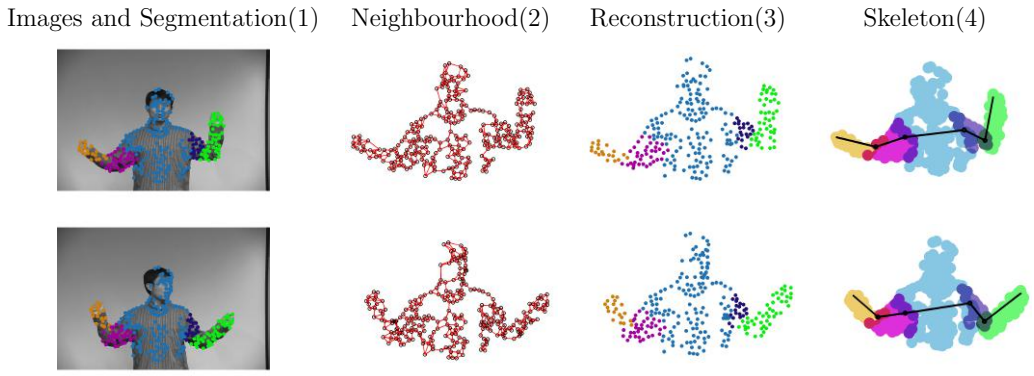


Figure 3.3: Reconstruction results of the 'dance' dataset [138]. From left to right (1) Original image and segmentation results. (2) Generated neighbourhood structure. (3) Segmentation results and estimated 3D reconstructions. (4) Estimated skeletal structure, and model assignment. Figure from [39].

the modelling power of these models is restricted by the global low-rank shape prior (either explicitly represented as a linear shape basis model or implicitly encoded as a soft constraint), and cannot deal with strong local deformations. To overcome this limitation, a new trend of deformation modelling methods, i.e. piecewise modelling methods, has emerged recently.

Varol *et al.* [129] proposed a piecewise planar method for reconstructing deforming meshes. In this work, they successfully reconstructed deforming planar objects, such as a piece of paper. They first manually divided the paper into rectangular overlapping regions, then reconstructed each individual patch independently and stitched them together into a complete object based on the common points between overlapping patches. Figure 3.2 shows the algorithm workflow of this method.

Later, Taylor *et al.* [120] introduced a locally rigid formulation for non-rigid reconstruction. Specifically, they first reconstructed every neighbouring triple of non-collinear points, so called 'triangle soup' and then glued these triangles together by solving depth and reflection ambiguities. Although locally rigid reconstruction based on triangle soups do not require manual patch division, this method suffers from computation complexity issues with increasing number of triangles. In particular, resolving the flip ambiguity is NP-hard with respect to the number of triangles.

In recent work by Russell *et al.* [103], deformable shape was modelled as overlapping piecewise quadratic models and the problem of non-rigid reconstruction was formulated as a multi-model fitting problem, where points

need to be assigned to local rigid models, whose parameters in turn need to be updated based on the assignment result. Unlike previous methods where points were manually divided into different patches, Russell *et al.*'s method provided a principled solution for the patch division problem. Moreover, this multi-model fitting problem is solved within an energy minimization framework, where the energy could conveniently incorporate spatial smoothness prior of the assignment and also an MDL prior. Later the idea of overlapping models was also used for articulated reconstruction [39] and dense NRS*f*M problems [105]. Figure 3.3 shows articulated reconstruction results [39] on the 'dance' dataset.

3.1.3 NRSFM for Multiple Objects

Reconstructing general dynamic scenes with an unknown number of objects, is a challenging simultaneous segmentation and reconstruction problem. When the scene contains many moving objects, a successful reconstruction would need to segment the scene into different independently moving objects and reconstruct them at the same time.

Most [15, 137, 16, 28, 45, 103] NRS*f*M methods assume that each deforming object has been pre-segmented before reconstruction, either manually or using a motion segmentation method. This pipeline approach has the drawback that motion segmentation and reconstruction tasks are separated and thus errors in the first segmentation step cannot be recovered from, leading to reconstruction failure.

In contrast, methods like [135, 101, 70] perform segmentation and 3D reconstruction simultaneously. Wang *et al.* [135] proposed a multibody reconstruction method for reconstructing dynamic scenes of rigid moving objects from pairs of images. They first established feature correspondences between the two images, and then used a energy minimization framework to optimize the assignment of points to different motion models, motion model parameters and 3D points simultaneously. Roussos *et al.* [101], introduced a similar framework for reconstructing multiple rigid objects from monocular sequences. In their framework, they formulated the problem of simultaneous segmentation, motion estimation and dense 3D reconstruction of dynamic scenes as the optimization of a single unified objective function. They showed that their method was able to produce detailed and accurate 3D reconstructions for Augmented Reality. Recently, Kumar *et al.* [70] extended classic NRS*f*M methods to multiple deforming objects. By minimizing an energy function that combines reprojection error, sparse self representation prior and a low-rank deformable shape prior,

their method could deal with multiple deforming objects at the same time.

Note that reconstructing each object in the scene independently does not guarantee a globally consistent scene reconstruction, as the inherent scale ambiguity of each object still remains unsolved. To put every object in the right scale, relation cues between neighbouring objects are often used. In particular, Ranftl *et al.* [98] proposed a monocular dense depth estimation method for reconstructing dynamic scenes by optimizing a global objective function that integrates reprojection error, spatial smoothness prior and occlusion cues between background and foreground objects. In this way, their method is able to obtain a consistent dense depth map for complex dynamic scenes.

3.2 Shape-from-template Methods

Unlike the generic reconstruction methods of NRS*f*M that assume no specific information about the scene being reconstructed, shape-from-template methods require that a template shape of the object is given and the goal is to track the deformation of the template mesh over a monocular sequence.

Most template-based methods [106, 108, 109, 107, 7] use an indirect approach to reconstruct the non-rigid shape. First either 2D-2D or 3D-2D correspondences are established via sparse feature matching or optical flow methods, then the template shape is deformed accordingly based on these correspondences. However, correspondence constraints only are not sufficient to constrain the 3D deformations and without additional priors the non-rigid tracking problem is ill-posed.

To make the problem well behaved, priors of the deformation of the object to be reconstructed need to be assumed. The most commonly used deformation priors for non-rigid objects are isometric constraint (distances between neighbouring mesh vertices remain constant during the deformation), smoothness prior (deformation of neighbouring vertices should be similar) and as rigid as possible prior (deformation can be approximated by local rigid body motion). For instance, Salzmann *et al.* [109] proposed a closed-form solution for template-based non-rigid tracking. Specifically, they assume that a reference image, i.e. the image of the template shape, is given and the 2D correspondences between reference image and current image have been computed. Based on these correspondences and inextensibility constraint (relaxed isometric constraint), they formulate the tracking problem as a set of quadratic equations, which could then be solved by Extended Linearization [26].

Recently, Bartoli *et al.* [7] introduced first-order methods by using both

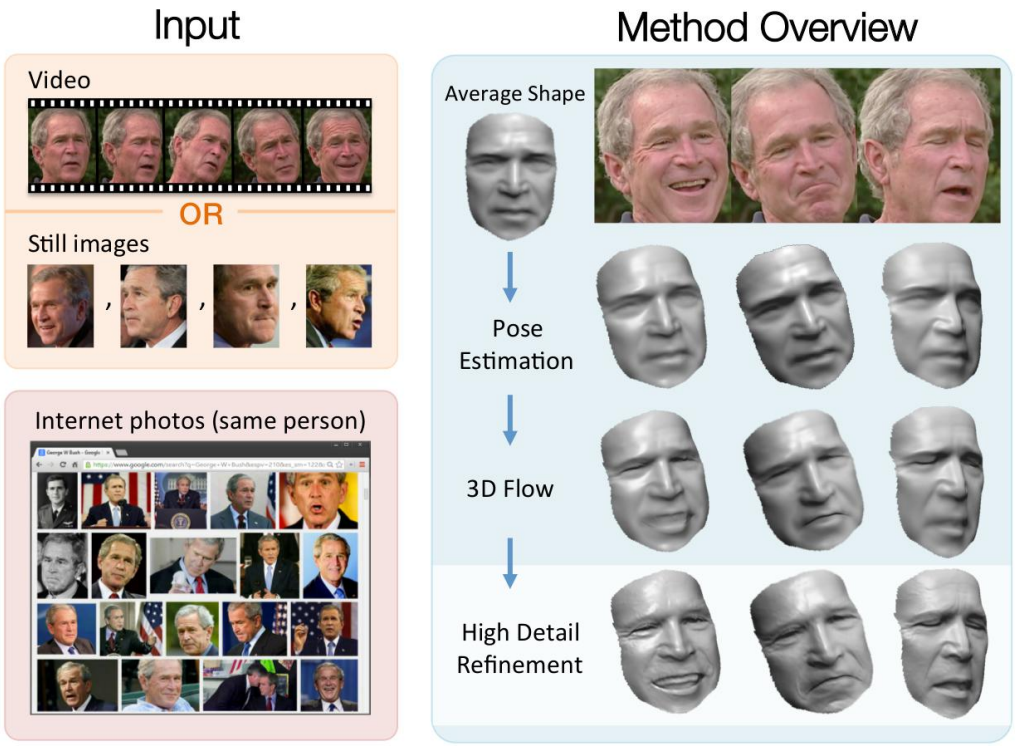


Figure 3.4: Overview of method [119]. Given a video sequence they first estimate 3D global pose, then compute dense 3D flow between the average model and the input expression, followed by high frequency details refinement using shading cues. Figure from [119].

image point locations and their first-order differential structure which can be computed from the warping between the template and input image. They formulated the problem of non-rigid tracking of isometric and conformal surfaces as a system of PDEs that they can solve analytically.

While most template approaches are feature-based and only reconstruct a small number of points, Malti *et al.* [78] proposed a direct pixel-based variational framework that exploits visibility constraints. However, their method was only demonstrated on flat isometric surfaces.

The recent work of Suwajanakorn *et al.* [119] reconstructs faces of celebrities from RGB-only videos. First they build a person-specific average face shape from a large collection of images of the same person. Then given a new video sequence, they could track the non-rigid deformation of the face by estimating dense 3D flow of the average model to match the input expression. After 3D flow computation, they recover high frequency details of facial expressions using shading cues. They formulate template-based non-rigid reconstruction

as a frame-to-frame energy minimization that optimizes a direct photometric cost. However, their method is limited to reconstructing human faces as their template reconstruction approach is specifically tailored to them.

3.3 Model-based Methods

Much like low-rank based NRS*f*M methods, model-based non-rigid reconstruction methods assume that the deformation of shapes lie on a low-dimensional manifold. However, there are two important differences here. First, in NRS*f*M, low-rank assumption is a generic assumption for arbitrary deforming shapes. Second, the linear shape bases are not fixed, but estimated together with the coefficients for every sequence during reconstruction. In contrast, for model-based methods, the low-dimensional manifold is built specifically for a particular class of objects from a large set of training examples. Moreover, the manifold is built offline and usually fixed during reconstruction.

Model-based methods have been successfully used for non-rigid reconstruction of deforming shapes, such as faces, human bodies, *etc*. For instance, Cootes *et al.* [24] proposed a 2D morphable face model, active shape models (ASM), for locating 2D feature points on human faces. This model was then extended to active appearance models (AAM) [34] by introducing appearance information into the ASM framework. As a linear shape and appearance model, AAM model is obtained by performing PCA on manually labelled 2D meshes and their corresponding images. During test time, the model parameters (the coefficients of the linear bases) are optimized by fitting AAM to input images, i.e. maximizing the match between the predicted image and input image. This fitting problem is a non-linear optimization problem, and usually solved with iterative optimization techniques [23, 34]. The AAM model was later revisited by Matthews *et al.* [82]. They proposed an efficient inverse compositional image alignment algorithm for AAM fitting, which outperformed previous approaches in terms of computational cost, convergence speed, and frequency of convergence.

3D morphable models (3DMM) were first used for face modelling by Blanz and Vetter [9]. In their seminal work, they built a 3D linear shape and apperance model by performing PCA on high resolution laser scans of 200 subjects, 100 male and 100 female. Based on a set of manually labelled scan examples, they learnt the mapping from facial attributes to morphable model parameters, i.e. shape and texture coefficients, and showed that by tuning these coefficients they could deform the face to exhibit certain attributes. They also showed

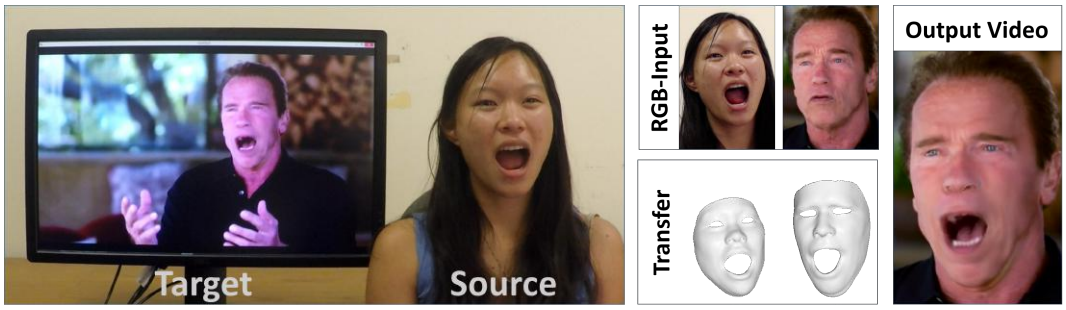


Figure 3.5: Proposed online reenactment setup for Face2Face [121] system: a monocular target video sequence (e.g., from Youtube) is reenacted based on the expressions of a source actor who is recorded live with a commodity webcam. Figure from [121].

how to perform 3D reconstruction from a single input image and registration between a given 3D scan and an existing model.

Matthews *et al.* [83] compared 2D and 3D face models in detail. In particular, they compared the representation power of 2D AAM and 3DMM. They proved that under scale orthographic camera model, these two are equivalent. However, 2D models require 6 times more parameters to represent the same 3D model. In general, 2D models are over expressive and can generate model instances that are impossible to generate with the corresponding 3D model.

Recently, Chen *et al.* [21] proposed a real-time facial tracking and animation system with a single video camera. By using a generic regressor learned from a set of public images with labelled landmarks, they could accurately infer 2D facial landmarks as well as the 3D facial shape from the input video frame. These inferred 2D landmarks were then used to update the camera matrix and the user blendshapes for next frame. The main novelty of this work was that they introduced a DDE (Displaced Dynamic Expression) model that, by combining a 3D parametric model with 2D landmark displacements, allowing them to track a face without knowing the user identity beforehand, and to update the estimated user identity on the fly.

More recently, Thies *et al.* [121] proposed an impressive real-time monocular face capture and reenactment system – Face2Face. Given a source video and a target video, the system could animate the facial expressions of the target video according to the source actor and re-render the synthetic video in a realistic fashion. The key of this system is the facial identity and expression recovery step performed by fitting a multi-linear PCA model to the input video, of which the first two dimensions represent facial identity (geometric shape and skin

Figure 3.6: 3D pose and shape estimation results on two example images from the Leeds Sports Pose Dataset [61]. From left to right: the input image, fitted model and the 3D shape rendered from a new viewpoint. Figure from [11].

reflectance) and the third dimension controls the facial expression. Specifically, all the unknown model parameters, including the face model, the illumination parameters, the rigid transformation and the camera parameters are optimized by minimizing a dense photometric energy. Then a novel expression transfer and mouth synthesis approach was used to generate realistic reenactment of target video. Although this system was completely based on RGB videos, they showed that the tracking accuracy was on par with the state-of-the-art, even for online tracking methods using depth data.

As well as faces, model-based methods were also used for capturing the deformation of human bodies [5, 75]. Loper *et al.* [75] introduced a Skinned Multi-Person Linear model (SMPL) for modelling the deformation of body shapes across a wide variety of subjects as well as different human poses. The parameters of the model, including template shape, pose-dependent blend shapes, identity-dependent blend shapes, a joint regressor and blend weights, are learned from thousands of aligned 3D high-resolution meshes of different people with different poses. They showed that the new model was more accurate than the state-of-the-art methods. Moreover, SMPL is compatible with existing rendering engines and is available for use in various animation softwares, such as Maya, Blender, Unreal Engine and Unity.

Based on SMPL, Bogo *et al.* [11] proposed a method for automatically recovering human pose and shape from a single RGB image. Given an input image, they first extract 2D body joint locations using DeepCut [95], and then fit a SMPL model to these extracted 2D joints by minimizing a cost that penalizes the difference between the projected 3D model joints and 2D joints.

Chapter 4

Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes

4.1 Introduction

Substantial progress has been made in multibody SfM and non-rigid structure from motion (NRSfM) for dealing with dynamic scenes [91, 100] or creating vivid life-like reconstructions of deformable objects [45]. However, dynamic scene reconstruction still remains a significant challenging problem and is far from solved. On the one hand, multibody SfM approaches can segment the scene into multiple **rigid** moving objects, but they cannot deal with the presence of deformable or articulated objects in the scene. On the other hand, although NRSfM algorithms can reconstruct a single pre-segmented **deformable** surface moving in front of a camera [45, 129], they require segmentation of the scene into background and foreground objects. To be able to deal with complex dynamic scenes, which may contain a mixture of rigid objects, non-rigid motion and articulated objects, a powerful framework is needed which combines the advantages of both multibody SfM and NRSfM. In other words, a framework is needed that can segment the scene into different moving objects and reconstruct them separately, in a unified way, regardless of whether objects are rigid, non-rigid or articulated.

Recently, piecewise methods have emerged as a promising technique for seamlessly handling general dynamic scenes by modelling all objects (rigid, non-rigid or articulated) in the scene as a combination of rigid parts. Piecewise approaches to non-rigid and articulated reconstruction have been successfully applied to explain the complex motion of 2D tracks on a single non-rigid

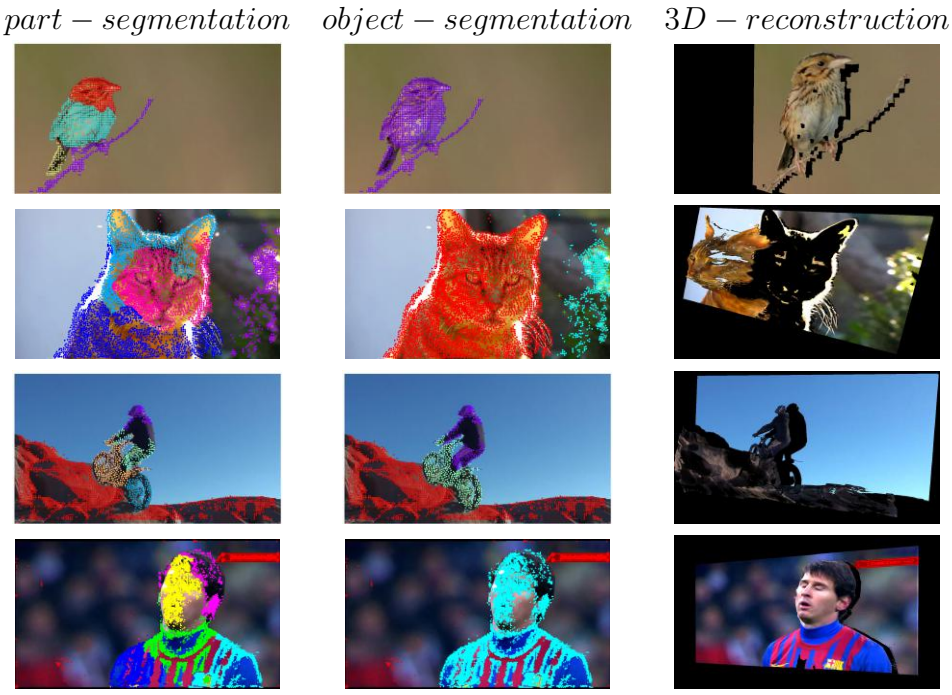


Figure 4.1: Segmentation and 3D reconstruction results of three dynamic sequences of the *Youtube-Objects* Dataset [96] and a football sequence downloaded from YouTube. **Left:** segmentation into *parts* (rigid models). **Centre:** segmentation into *objects*. **Right:** densified 3D video pop-up from a novel viewpoint. The *bird* and *cat* sequence are non-rigid sequences occluding a static background. The *motorbike* sequence, acquired with a moving camera, shows articulated motion. Bottom row shows a reconstruction of football footage.

surface or an articulated object as a network of *overlapping* parts [39, 103, 129]. However, if naively applied to an entire scene with foreground/background objects occluding one another, depth boundaries between objects would not be respected and neighbouring models in the image would be forced to overlap irrespective of whether or not they belong to the same physical object.

The main contribution of the work we described in this chapter is to offer a solution to the problem of *scene reconstruction* for real-world dynamic monocular videos that deals seamlessly with the presence of multiple non-rigid, articulated or pure rigid motion. In an entirely unsupervised approach, we reorganise/segment the scene into a constellation of object parts, recognise which parts are likely to constitute objects, join them together, and reconstruct the scene. We offer solutions to some of the problems of previous approaches to dynamic scene reconstruction: *(i)* Our approach is able to adapt the topology of the neighbourhood graph by breaking edges where necessary to preserve

boundaries between objects. In this way our approach can deal with an entire scene where objects might occlude one another and not just pre-segmented objects; *(ii)* Our work results in a hierarchical approach to dynamic scene analysis. At the higher level of the hierarchy the scene is explained as a set of ***objects*** that are detached from the background and from each other. At the lower level of the hierarchy, each *object* can be explained as a set of overlapping ***parts*** that can model more complex motion. Figure 4.1 shows our *part*, *object* segmentation and 3D reconstruction results for four different sequences.

Our approach is closely related to the paradigm of *multiple model fitting* where feature tracks, that might contain outliers, belong to an unknown number of models. The assignment of tracks to models and the estimation of model parameters are optimized simultaneously [58, 103] to minimize a geometric cost subject to the constraint that neighbouring tracks must belong to the same model. The energy also incorporates a minimum description length (MDL) cost that prefers sparse solutions. The cost function is optimized by alternating between a discrete graph-cuts algorithm to solve the labelling problem and a continuous optimization to update the model parameters. This approach has previously been applied to computer vision problems such as stereo [10]; motion segmentation [58]; 3D reconstruction of non-rigid [103] and articulated objects [39]; and multi-body reconstruction [100].

Our approach departs from previous work in geometric multiple model fitting in multiple ways: *(i)* Our model offers segmentation at two granularities: object-level and part-level. At the object-level, we segment the scene into a small number of disjoint *objects*. At the part-level, objects are further divided into a set of overlapping parts; *(ii)* Our model uses a combination of appearance and geometry cues for segmentation which encourages salient foreground objects to be separated accurately from the background even when the motion is not distinctive enough; *(iii)* Our geometric cost uses a perspective camera model and is able to deal with perspective effects and incomplete tracks.

4.2 Problem Formulation

We consider a monocular video sequence, possibly downloaded from the web, captured by a single camera observing a complex dynamic scene that contains an unknown mixture of multiple moving and possibly deforming objects. First, we extract a set $\mathcal{T} = [1, \ldots, T]$ of feature point tracks using Sundaram *et al.*'s publicly available code [118]. Although the tracker aims to provide long-term video correspondences, the length of tracks is variable and not all points tracked

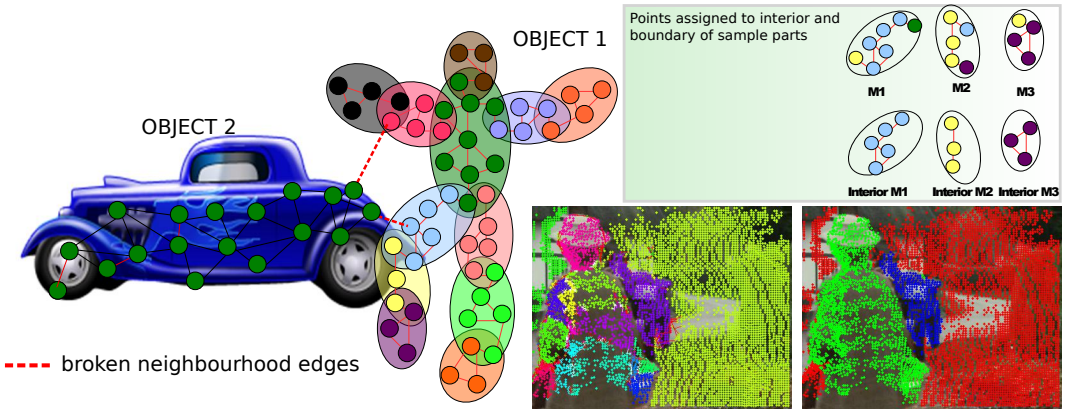


Figure 4.2: **Left:** Conceptual illustration of our approach to 3D reconstruction of complex dynamic scenes. The image shows a person occluding a car. In the original neighbourhood graph, some point-tracks on the car are path connected with tracks on the person. Our approach reasons about object boundaries by adapting the neighbourhood, and breaking edges where necessary to detach parts from other occluding objects. **Top Right:** Illustration of the concept of overlapping models and *interior points* [103]. A tracked point belongs to the interior of a model (points with the same colour) if all its neighbours also belong to that model (though not necessarily as interior points). **Bottom right:** real-world example of segmentation into parts (left) and two objects and background (right).

are visible in all the frames. We make no assumptions about the number of objects or their motions which could be rigid, articulated or non-rigid. Our goal is to estimate the 3D coordinates for all feature points in every frame.

4.2.1 Piecewise Overlapping Models

The works [103, 39] proposed a novel piecewise approach to the problem of 3D reconstruction of non-rigid objects. Rather than attempting to reconstruct objects by fitting a global low-rank shape model [125, 92] that is sufficiently expressive to capture deformations, but also sufficiently low-rank to discourage overfitting, they automatically segmented the object to be reconstructed into a set of parts, each of which could be expressed by a simple model – either local rigid reconstructions [39] or local quadratic deformations [103]. By forcing these parts to overlap, and to agree about the reconstruction of the region of overlap, per part depth/scale and sign-flip ambiguities can be resolved. Figure 4.2 shows an illustration of the segmentation of an articulated object into overlapping rigid parts.

The problem was formulated as a labelling one where the assignment of tracks to models and the fitting of models to tracks were jointly optimized

to minimize a geometric fitting cost subject to the spatial constraint that neighbouring tracks should also belong to the same model.

4.2.1.1 Assignment of point tracks to models

Let $\mathcal{T}$ refer to a set of point tracks and $\mathcal{M}$ a set of models. We use the notation $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_T\}$ to refer to a labelling, where $\mathbf{x}_i$ is the *set* of models assigned to track i. Assuming a known topology, or graph, which connects tracks together in a neighbourhood structure $\mathcal{N}$, the following objective was proposed by [39, 103]

$$C(\mathbf{x}) = \sum_{i \in T} \sum_{m \in \mathbf{x}_i} U_i(m) + \text{MDL}(\mathbf{x}) \tag{4.1}$$

where tracks are allowed to belong to multiple models in $\mathcal{M}$. The unary term $U_i(m)$ is the cost of assigning track i to model m and the term $\text{MDL}(\mathbf{x})$ is a label cost that encourages sparse solutions. In [39] local rigid models were used where each model was parameterized with the rotation and translation associated with a rigid motion and the unary cost $U_i(m)$ was defined as the image reprojection error under orthographic projection for that point given the model parameters. The optimization of 4.1 was subject to the constraint that each track must be an interior point of some model, i.e. that for every track there is a model such that that track and all its neighbours belong to that model (Figure 4.2 illustrates the concept of *interior point*), or more formally:

$$\forall i, \exists \alpha : \alpha = I_i \quad \text{and} \quad I_i = \alpha \rightarrow \forall j \in N_i, \alpha \in \mathbf{x}_j \tag{4.2}$$

where $\mathbf{I} = \{I_1, I_2 \dots I_T\}$ refers to the assignment of each track i to the interior of one model I_i and N_i is the neighbourhood of track i. Russell *et al.* [103] showed how this problem could be formulated as a labelling problem over the assignment of tracks to the interior of models and efficiently solved using a novel variant of α-expansion. Starting from an excess of models the optimization followed a hill climbing approach that alternates between assigning tracks to models, and refitting the models to minimize the geometric error (image reprojection error).

4.2.2 Obstacles to reconstruction in the wild

Although these multiple model fitting approaches based on overlapping models do provide a robust approach to non-rigid [103] and articulated [39] reconstruction, they have shortcomings. First, they cannot deal with whole scenes

in which the neighbourhood graph maintains connections between tracks of different objects (see Figure 4.2) – the constraints 4.2 combined with a bad neighbourhood structure can force parts to straddle multiple objects, leading to an error that can not be recovered from. Secondly, the unary terms of [39, 103] minimize a geometric cost based on multiview affine factorization. Therefore, they have difficulty dealing with incomplete tracks. In real-world videos, tracks are likely not to persist for a large number of frames. Finally, a further limitation of the above approaches comes from the fact that only motion cues are used for the segmentation. Combining motion and appearance cues is useful to encourage object boundaries to be respected. Besides, these cues complement each other particularly if there are frames in the sequence with small motion.

The main contribution of our work is to offer solutions to these three limitations: *(i)* Our approach adapts the topology of the neighbourhood graph by breaking edges where necessary to preserve boundaries between objects. This allows our approach to deal with complete video footage where objects might occlude one another and not just single pre-segmented objects. *(ii)* Our geometric unary cost is based on frame-to-frame fundamental matrices, an approach than can naturally handle incomplete tracks. *(iii)* Our data term combines geometric and appearance costs. We use the saliency score provided by [113] to encourage parts of similar saliency to belong to the same object.

4.3 Scene Reconstruction

We propose a novel cost that allows us to modify the topology of the original neighbourhood by deleting edges between point tracks that belong to different physical objects, and should not overlap. Our new cost has four terms

$$C(\mathbf{x}) = E_{data} + E_{break} + E_{sparse} + E_{mdl} \tag{4.3}$$

$$= \sum_{i \in \mathcal{T}} \sum_{m \in \mathbf{x}_i} U_i(m) + \sum_{i \in \mathcal{T}} \sum_{j \in N_i} d_{i,j} \Delta(j \notin N_i') \tag{4.4}$$

$$+ \sum_{m \neq n \in \mathcal{M}} \Delta(\exists i : I_i = m, n \in \mathbf{x}_i) + \mathrm{MDL}(\mathbf{x}) \tag{4.5}$$

where as before $\mathbf{x}_i$ is the *set* of models that point i belongs to; $\Delta(\cdot)$ is the indicator function, taking value 1 if the statement is true and 0 otherwise; and N_i' the modified neighbourhood of track i. This optimization is subject to the constraints that neighbouring tracks also belong to the interior model, or more

formally

$$\forall i, I_i = \alpha \rightarrow \forall j \in N_i', \alpha \in \mathbf{x}_j \tag{4.6}$$

We now describe in detail each term of our cost function.

4.3.1 Unary Costs (E_{data})

Our unary term is the sum of two costs i.e. $U_i(m) = G_i(m) + P_i(m)$, that encourage tracks that both move consistently as a rigid object and have similar saliency scores, to belong to the same model. The geometric term $G_i(m)$ evaluates the cost of assigning track i to a rigid model m as the deviation from the epipolar geometry across all pairs of consecutive frames. The second term $P_i(m)$ computes a saliency score for each pixel in every frame and encourages tracks with similar saliency scores, to belong to the same model.

4.3.1.1 Rigidity term G_i

Given a set of point tracks assigned to the same rigid part, we parameterize the rigid model m associated with them as a set of $F-1$ fundamental matrices $\mathbf{F}_m = \{\mathbf{F}_m^{1,2}, \ldots, \mathbf{F}_m^{f,f+1}, \ldots, \mathbf{F}_m^{F-1,F}\}$ for every pair of consecutive frames in the sequence $f = \{1, \ldots, F-1\}$. The cost of associating track i to a specific rigid model m is the Sampson error [51] added over all pairs of fundamental matrices

$$G_i(m) = \sum_{f<F} \gamma^{-1} ({u_i^{f+1}}^T \mathbf{F}_m^{f,f+1} u_i^f)^2 \tag{4.7}$$

where u_i^f encodes the homogeneous image coordinates of track i in frame f and u_i^{f+1} its corresponding position in frame $f+1$ and γ is the Sampson weight [51]. This cost is summed over all frames in which the track is visible. To estimate the fundamental matrices, we use the eight-point algorithm embedded in a RANSAC scheme followed by non-linear refinement of 4.7. This fitting cost has several clear advantages over the affine factorization cost used by [39]. First, it allows to model perspective effects which are often present in unconstrained videos and to perform perspective reconstruction given an estimate of the camera calibration matrix. Second, it behaves better in the presence of missing data or short tracks, as it computes frame-to-frame geometric costs only for the frames where the track is visible rather than the multiframe factorization cost of [39].

4.3.1.2 Saliency Term

The work [113] provides a fully unsupervised method for object detection in an image I, using a novel saliency map S_I. The main idea of this paper is that as the appearance of salient object is less frequent than background, therefore by sampling from images, salient regions should be less probable. While [113] made use of both the statistics taken from a large corpus of unlabelled images, and from the image itself, we only make use of the statistics of the single image (this measure is termed *within image saliency* in [113]). We compute saliency maps S_{I_f} for each frame f in the video sequence and define the saliency cost $P_i(m)$ of point i belonging to model m as the distance from the mean saliency of model m

$$P_i(m) = \lambda_s \sum_{f \leq F} (S_{I_f}(i) - \bar{S}_m)^2 \tag{4.8}$$

where $\bar{S}_m$ is the mean saliency of all tracks that currently belong to model m, $S_{I_f}(i)$ is the saliency score of point i in frame f and λ_s a weight on the importance of this term.

The second row of figure 4.5 shows the saliency detection results on our synthetic dancer sequence, where brighter means more salient. It can be seen that the saliency maps are very discriminative between the lower part of the person and the background. By incorporating saliency term 4.8 into our unary cost, we are able to successfully separate the moving person and the static background, which is challenging for just motion cues as the left foot of the person is on the ground plane and does not leave the ground during the whole sequence. As shown in figure 4.5, without saliency cues the algorithm is not able to segment the dancer from the background.

4.3.2 Topologically Adaptive Neighbourhood (E_{break})

The cost 4.1 proposed in [103] was internally represented as a local MDL prior defined over the set of interior labels present in a local neighbourhood, and took the form

$$\sum_{i \in \mathcal{T}} \sum_{m : \exists j \in N_i \cap m = I_j} U_i(m) \tag{4.9}$$

As discussed, in order to separate connected objects from one another, we wish to discard edges from the neighbourhood N_i with a per edge cost $d_{i,j}$. As such, the new cost will be of the form

$$\sum_{i \in \mathcal{T}} \sum_{m : \exists j \in N_i \cap m = I_j} \min \big(\sum_{j : I_j = m} d_{i,j}, U_i(m) \big) \tag{4.10}$$

where $d_{i,j}$ is a varying cost modulated by a sigmoid function of the form

$$\mathrm{d}_{i,j} = \frac{M_c}{1+e^{M_{ij}-\frac{M_a}{M_b}}} \tag{4.11}$$

Here M_{ij} is computed as the pairwise neighbourhood distance between points i and j in the image and velocity spaces. M_a, M_b and M_c are parameters of the sigmoid function.

4.3.3 Overlap Sparsity Term (E_{sparse})

By itself, discarding edges from the neighbourhood graph improves the quality of the parts found, and allows more objects to be found. However, it does not correctly separate objects from the background. In almost all sequences, we find that one or two ambiguous tracks exist that could be easily explained as either object or background parts. These ambiguous tracks act as junctions, or regions of overlap between foreground and background objects, connecting the two and making it impossible to distinguish between foreground and background.

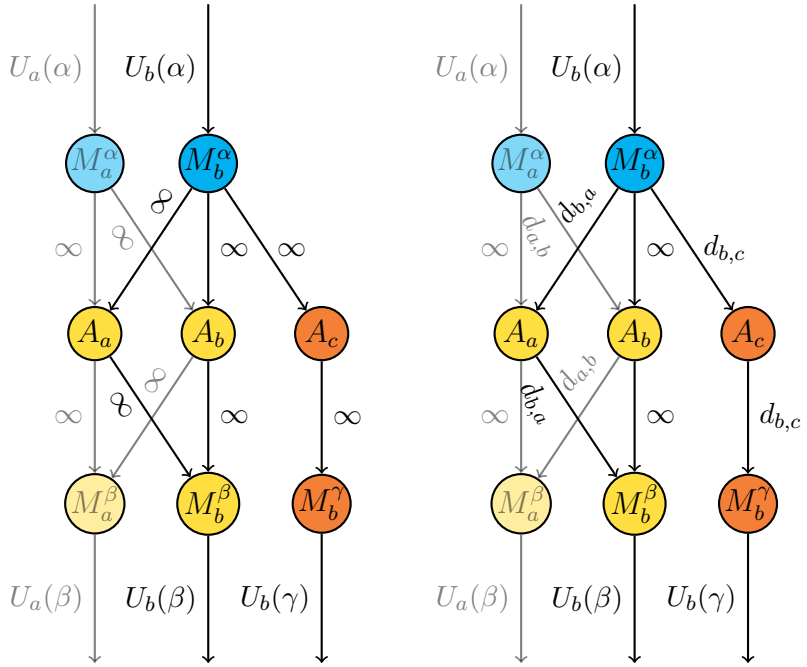
To eliminate this leaking, we introduce a novel sparsity term that penalizes the total number of models that overlap and encourages regions with limited overlap to disconnect. We formulate this penalty as a count of the number of pairs of models (m, n) such that there exists a track belonging to the interior of model m and also to model n, i.e.

$$\sum_{m \neq n \in \mathcal{M}} \Delta(\exists i : I_i = m, n \in \mathbf{x}_i) \tag{4.12}$$

As this cost does not depend on the number of tracks in the region of overlap, it dominates in small regions of overlap or where the cost of discarding edges is small, and is ignored elsewhere.

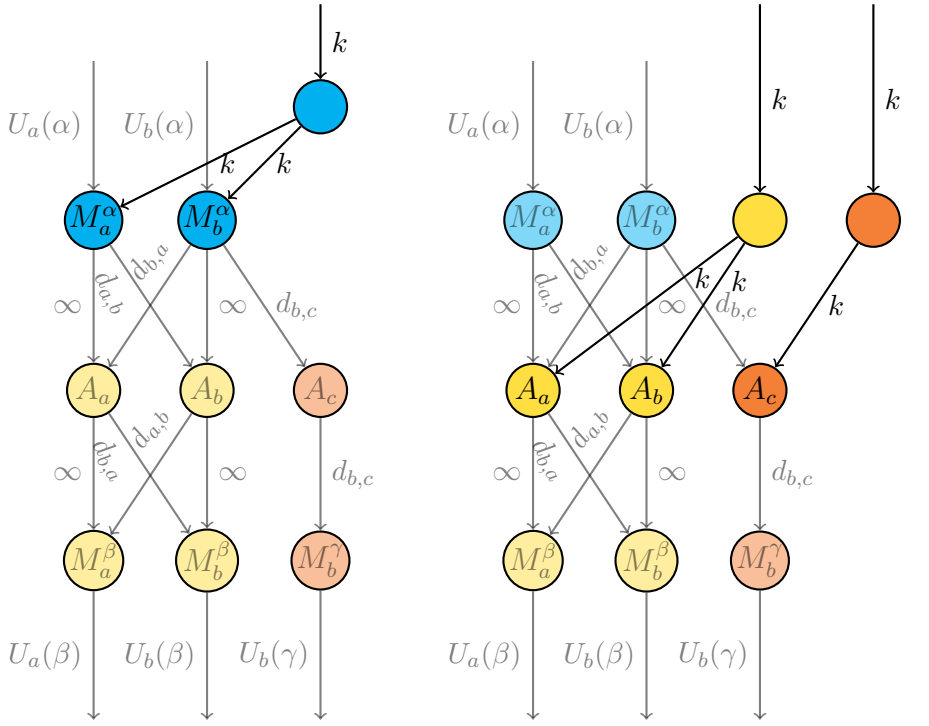
4.4 Efficient Optimization

As with other multiple model fitting approaches [39, 58, 103], we initialize with an excess of models which are generated by sampling randomly neighbouring groups of ten feature tracks and computing the frame-to-frame fundamental matrices using the eight-point algorithm [51]. We then optimize the cost 4.3 using a hill-climbing approach alternating between: *(i)* fixing the parameters of $\mathbf{F}_m$ and optimizing the labelling that assigns tracks to a set of parts (models) $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_T\}$ and *(ii)* fixing the labelling and optimizing $\mathbf{F}_m$ for all models.



(a) Without edge breaking (b) With edge breaking

Figure 4.3: Comparison of graph constructs between overlapping cost [103] and our new edge breaking cost 4.10. Left figure shows the graph construct for overlapping cost over the neighbourhood of point b (a, b and c, taking label β, β and γ respectively). Transparent nodes indicate the contribution of overlapping neighbourhoods. Right figure shows the modified graph construct for edge breaking cost, allowing point a to be dropped from N_b with cost $d_{b,a}$. The top row contains auxiliary variables indicating if tracks, a, b or c belong to model α. The middle row contains the standard expansion variables which govern whether or not a variable transitions to the interior of model α, while the bottom row shows auxiliary variables indicating if a variable belongs to model β or γ. Here we show the case when label α is not present in the neighbourhood. To understand the differences between these two figures, we could have a look at the point a from neighbourhood N_b. In particular, there are two edge connections, $M_b^\alpha \to A_a$ and $A_a \to M_b^\beta$. In the left figure, these two edges cannot be broken (with connection weights ∞), indicating that α is not a model of point b ($M_b^\alpha = 0$) while α is an interior model of point a ($A_a = 1$) cannot happen, and similarly β is not a model of point b ($M_b^\beta = 1$) while β is an interior model of point a ($A_a = 0$) cannot happen as well, both due to the overlapping constraint. In the right, we allow both these two cases to happen, each with a penalty $d_{b,a}$.



(a) $k\Delta(\mathbf{M}^\alpha \cap \mathbf{I}^\beta_{\text{last}} \neq \emptyset)$ (b) $k\Delta(\mathbf{M}^\beta_{\text{last}} \cap \mathbf{I}^\alpha \neq \emptyset)$, $k\Delta(\mathbf{M}^\gamma_{\text{last}} \cap \mathbf{I}^\alpha \neq \emptyset)$

Figure 4.4: Graph construct for overlap sparsity term. Transparent graph shows the graph construction result of edge breaking cost, as shown in Figure 4.3. Here we show the cost over the neighbourhood of point b (a, b and c, taking label β, β and γ respectively). We assume the edge between point b and c is currently broken, so $\mathbf{M}^\beta_{\text{last}} = \{a, b\}$ and $\mathbf{M}^\gamma_{\text{last}} = \{c\}$.

Alpha expansion [14] finds a local optimum of a difficult to optimize cost function by iteratively moving from a current labelling to the lowest-cost solution obtained by relabelling some of the variables as α. Finding an optimal move is formulated as a pseudo Boolean optimization [12] and solved using graph-cuts [13]. We follow work [103] in considering expansion moves over the interior of labels. We use $A \in 2^T$ to refer the found expansion move, with A_i taking value 1 if variable I_i transitions to label α in the move, and 0 otherwise. Unlike [103] we will need to explicitly keep track of whether or not tracks belong to models at all (either as interior or boundary tracks) and for a particular expansion move on label α this will be done by means of binary variables $M_i^\alpha = 1$ if $\alpha \in \mathbf{x}_i$ and a complementary set of variables M_i^β, such that $\beta \neq \alpha$ and $M_i^\beta = 0$ if $\beta \in \mathbf{x}_i$.

Optimization of the costs E_{data} and E_{mdl} can be done using the techniques

of [103]. We now deal with the modifications to the optimization required by the terms E_{break} and E_{sparse}. Although exact optimization of either of these costs is straightforward, optimizing both together is challenging, and we make use of the convex-concave procedure (CCP) [139, 85], and find an optimizable cost that is tight at the current location, but an over-estimate elsewhere.

4.4.1 Exactly Optimizing E_{break}

We can rewrite cost 4.10 in terms of the auxiliary variables

$$\sum_{i\in\mathcal{T}} \sum_{\substack{\beta\in\mathcal{M}\\ \beta\neq\alpha}} \min_{M_i^\beta} \Big(\sum_{\substack{j\in N_i\\ j:I_j=\beta}} ((1-A_j)d_{i,j}M_i^\beta) + U_i(\beta)(1-M_i^\beta)\Big) \tag{4.13}$$

$$+\sum_{i\in\mathcal{T}} \min_{M_i^\alpha} \Big(\sum_{\substack{j\in N_i\\ I_j\neq\alpha}} (A_j d_{i,j}(1-M_i^\alpha)) + \sum_{\substack{j\in N_i\\ I_j=\alpha}} (d_{i,j}(1-M_i^\alpha)) + U_i(\alpha)M_i^\alpha\Big) \tag{4.14}$$

This change can been seen as a robustification of the local co-occurrence potentials of [103] analogous to the robust P^n model [66]. As with the P^n potentials, it can be formulated as a graph-cuts problem simply by adjusting the used edge weights. Note that d_{ij} is defined as ∞ when $j = i$. Figure 4.3 shows an example when no track in the neighbourhood takes label α. Please see the caption of the figure for a detailed explanation of the graph construction and differences between overlapping cost and edge breaking cost. For neighbourhood containing α, the graph construction is similar. Note that we always assume that $d_{i,i} = \infty$, indicating that the interior model of a track is always a model for itself.

4.4.2 Approximately Minimising E_{sparse}

For the following section it is more convenient to use sets to describe which points belong to which models. We use $\mathbf{M}^\beta$ to refer to the set of points belonging to model β, $\mathbf{I}^\beta$ for the interior of model β, $\mathbf{M}^\beta_{\text{last}}$ for the region (fixed throughout the move) that was assigned to model β by the previous move, and $\mathbf{I}^\beta_{\text{last}}$ for points previously belonging to the interior of model β. Performing an expansion move on label α, we have three cases to consider: *(i)* the cost is a direct function of the interior labels $\mathbf{I}^\alpha$; *(ii)* the cost depends on tracks belonging to the boundary of α: $\mathbf{M}^\alpha \setminus \mathbf{I}^\alpha$; *(iii)* the cost is not a function of α and depends on: $\mathbf{I}^\beta \cap \mathbf{M}^\gamma$, where $\beta, \gamma \neq \alpha$.

For an expansion move on label α, $\mathbf{I}^\alpha$ is monotone increasing while $\mathbf{I}^\beta$ is monotone decreasing. If one of either the sparsity costs, or the edge breaking

of the previous subsection was not used, the labelling of $\mathbf{M}^\alpha$ and $\mathbf{M}^\beta$ would also be guaranteed to be monotone increasing/decreasing, but together the situation is more complex. In the following discussion, we artificially constrain the set of possible moves of $\mathbf{M}^\beta$ to be monotone decreasing, and allow $\mathbf{M}^\alpha$ to change arbitrarily. Let us deal with these costs by turn:

Interior of α cost: We consider the localised MDL costs

$$\Delta(\mathbf{M}^\beta_{\text{last}} \cap \mathbf{I}^\alpha \neq \emptyset) + \Delta(\mathbf{I}^\beta \neq \emptyset) - 1 \tag{4.15}$$

This cost is 1 if $\mathbf{I}^\alpha$ expands into $\mathbf{M}^\beta_{\text{last}}$ without completely removing model β (which can only be done by making sure no tracks belong to the interior of model β) and 0 otherwise. Clearly this is an over-estimate as the true $\mathbf{M}^\beta$ in the set of all moves considered is always smaller than $\mathbf{M}^\beta_{\text{last}}$, and tight at the current location. As this cost is simply two MDL costs defined over subregions of the graph, it can be optimized using the techniques of [71]. As these move costs satisfy the CCP criteria, they reduces the original cost function.

Boundary of α cost: A similar argument can be made for the above cost. Instead of directly optimizing it, we solve the over-approximation

$$\Delta(\mathbf{M}^\alpha \cap \mathbf{I}^\beta_{\text{last}} \neq \emptyset) + \Delta(\mathbf{I}^\beta \neq \emptyset) - 1 \tag{4.16}$$

This can be formulated as a local MDL prior over the auxiliary variable of the previous section and an MDL cost over label β.

Costs not dependent on α: The local co-occurrence potentials considered here, fall into the class of potentials that can not be exactly optimized by an expansion move over label α. Instead we follow the strategy of [71] and optimize the cost

$$0.5\Delta(\mathbf{I}^\gamma \cap \mathbf{M}^\beta_{\text{last}} \neq \emptyset) + 0.5\Delta(\mathbf{I}^\gamma_{\text{last}} \cap \mathbf{M}^\beta \neq \emptyset) \tag{4.17}$$

4.4.3 Merging Parts into Objects

The final result of our scene segmentation algorithm is the labelling $\mathbf{x}$ which assigns each feature track to a set of rigid parts. Figure 4.6 shows some results of the part segmentation (second row) for five videos of the Berkeley Motion Segmentation Dataset [18]. To segment the scene into objects we label connected components of overlapping parts as object detections.

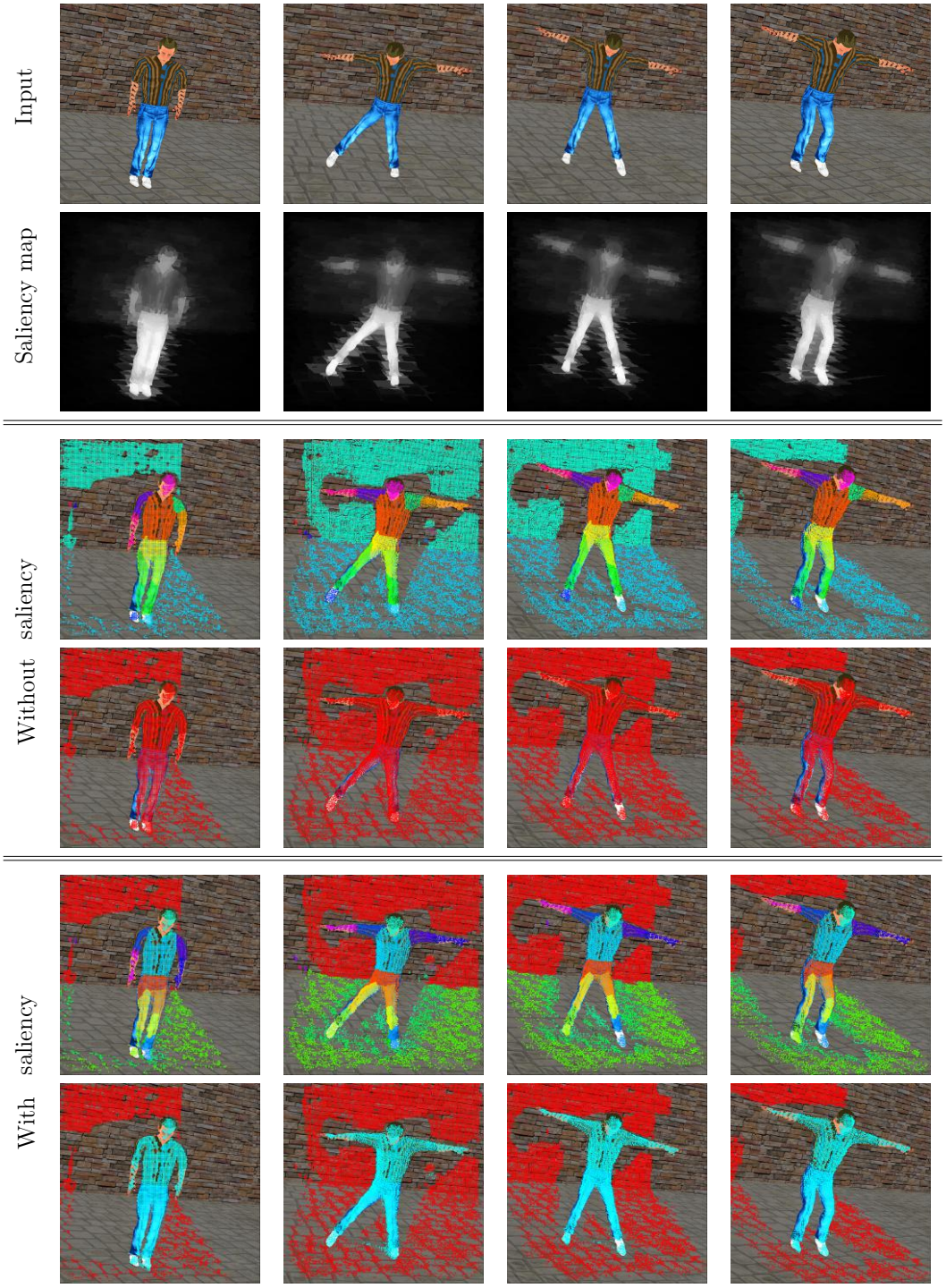


Figure 4.5: Comparison of part and object segmentation results on dancer sequence with and without using saliency. The first two rows are the input sequence and saliency results respectively. The third and fourth rows show the part and object segmentation results without using saliency, while the last two rows show corresponding results with saliency. It can be seen that as saliency cues are discriminative between the person and the background, adding a saliency term into unary cost can help break the connections between the left foot and the ground floor, resulting in a correct object segmentation.

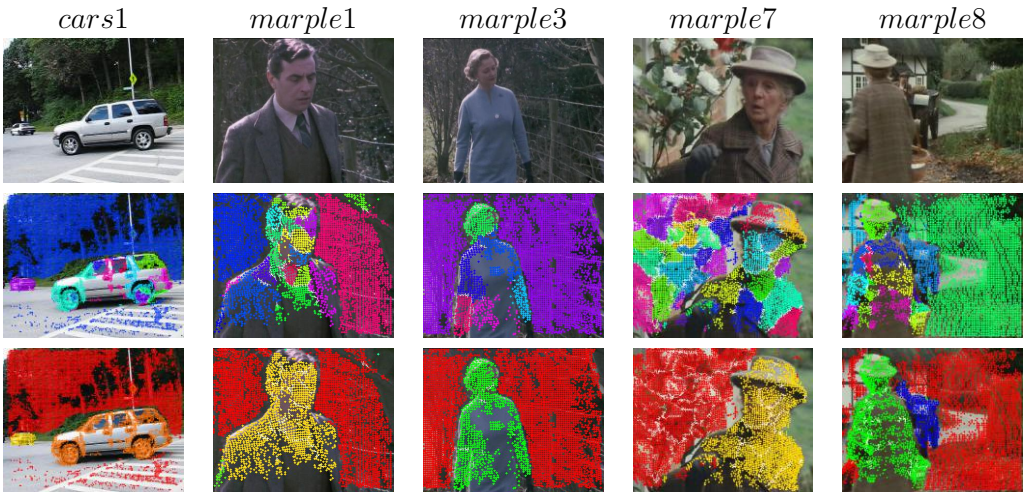


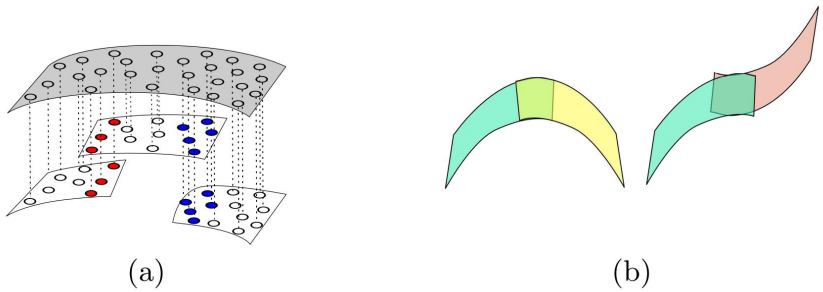
Figure 4.6: Motion segmentation results on five sample sequences of the Berkeley Motion Segmentation Dataset [18]. **Second row:** Part segmentation. **Third row:** Object segmentation.

Figure 4.7: Reconstruction results for a cat sequence of the *Youtube-Objects* Dataset [96]. **Second row** and **third row** are reconstruction results.

4.5 3D Reconstruction

The optimization of our cost function results in the labelling of rigid models or parts. Using the information about the regions of overlap, we also have a decomposition of the scene into different objects.

The 3D reconstruction of each object is then carried out using a piecewise rigid reconstruction approach. For each object we have a list of its constituent parts and a rigid model (set of fundamental matrices) for each part.



(a) (b)

Figure 4.8: Two ambiguities of orthographic reconstruction for each patch. (a) Reconstruction of shared points in different patches differ by a translation in the depth direction. (b) Representation of the ambiguity on the sign of the Z coordinate of the reconstructions. Figure from [38].

4.5.1 Piecewise Rigid Orthographic Reconstruction

When the average depth of an object is much larger than its depth relief, the 2D projections of the object can be well approximated with an orthographic camera model. In general, scenes captured with large focal length and small field of view tend to satisfy these conditions. In our case, the *bird*, *cat*, *bike* and *messi* sequences all belong to this category.

For these sequences, given the part and object segmentation results, we reconstruct each part independently with an orthographic factorization approach [80]. Then parts belonging to the same object are stitched together to obtain a consistent object reconstruction result.

In general, there are two ambiguities that cannot be resolved when performing orthographic reconstruction for each patch, the depth and flip ambiguities. Please see section 2.1.4 for a detailed discussion. In this section, we assume that temporal flip ambiguity has been resolved for each patch independently and we focus on solving the depth and flip ambiguity.

Figure 4.8 illustrates the two ambiguities for orthographic reconstruction. As patches are reconstructed separately, the ambiguities of overlapping patches do not necessarily agree with each other. To register these patches together to form a consistent object reconstruction, we need to solve the ambiguities between all overlapping patches.

Note that there is always a global depth and flip ambiguity that cannot be resolved, which is equivalent to picking one of the patches as reference and fixing its depth and flip. Although depth ambiguity can be resolved trivially given the solution to the flip, the problem of solving the flip ambiguity is

NP-hard. Fortunately, in most cases, as the number of patches within an object is relative small we can afford a brute-force search over all possibilities.

In particular, we create a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ for overlapping patches where each node denotes a patch, and each edge denotes the overlap between a pair of patches. We fix a patch as reference and pre-compute the cost for each edge. Assuming an edge e_{ij} between patch i and patch j, we compute two costs, c_{ij}^1 and c_{ij}^2, associated with this edge:

$$\begin{aligned} c_{ij}^1 &= \|\mathbf{d}_i - \mathbf{d}_j - \overline{\mathbf{d}_i - \mathbf{d}_j}\|_2^2 \\ c_{ij}^2 &= \|\mathbf{d}_i + \mathbf{d}_j - \overline{\mathbf{d}_i + \mathbf{d}_j}\|_2^2 \end{aligned} \tag{4.18}$$

where $\mathbf{d}_i$ and $\mathbf{d}_j$ are the $F \times P$ depth value matrices for all the common points over all frames of patch i and patch j respectively, $\overline{\mathbf{d}_i - \mathbf{d}_j}$ is the per frame average of the difference between these two depth matrices. Therefore c_{ij}^1 represents the alignment cost when two overlapping patches have the same flip, while c_{ij}^2 denotes the cost when the two patches take different flips.

We then search over all possible flips and find the configuration $\hat{\mathbf{f}}$ with the lowest overall cost:

$$\hat{\mathbf{f}} = \underset{\mathbf{f}}{\operatorname{argmin}} \sum_{(i,j)\in\mathcal{E}} \left(c_{ij}^1 \cdot \Delta(f_i = f_j) + c_{ij}^2 \cdot \Delta(f_i \neq f_j) \right) \tag{4.19}$$

where the flip value f_i takes 1 if patch i is flipped, otherwise 0.

When there is enough overlap between neighbouring patches and the points of the overlapping area have enough curvature to differentiate between flipped and non-flipped shapes, this depth-alignment based stitching approach works robustly. However, when the overlapping region is not discriminative enough, this method fails. In practice, we also use an alternative stitching strategy based on spatial rotation similarity, assuming that the rotations of neighbouring patches on the same object are similar. The only difference between these two methods is in the way we compute edge costs. Instead of computing edge costs using the depth-alignment error, we compute the rotation difference cost, between two neighbouring rotations for both values of f_i.

After solving the flips, we then use a greedy strategy to compute the depth of each patch and align all the overlapping pieces together. Specifically, we greedily select the pair with most common points and merge these two patches keeping the larger patch fixed until we have only one patch left.

Table 4.1: Evaluation results on the Berkeley Motion Segmentation Dataset using the metrics of [18]. *Fayad et al.* shows performance without discarding edges, using the same optimization as in [39, 103].

	Density	overall error	average error	over-segmentation	extracted objects
First 10 frames(26 sequences)					
Brox Malik	3.34%	7.75%	25.01%	0.54	24
Fayad *et al.*	3.28%	15.23%	51.89%	0.23	7
Our method	3.28%	8.00%	25.46%	1.00	22
First 50 frames(15 sequences)					
Brox Malik	3.27%	7.13%	34.76%	0.53	9
Fayad *et al.*	3.25%	24.95%	63.67%	0.20	0
Our method	3.25%	5.93%	27.84%	3.70	13
First 200 frames(7 sequences)					
Brox Malik	3.43%	7.64%	31.14%	3.14	7
Fayad *et al.*	3.42%	28.81%	66.78%	0.29	0
Our method	3.42%	13.28%	39.86%	8.60	4

4.5.2 Piecewise Rigid Perspective Reconstruction

When the depth range of the scene is not negligible compared to the average depth (for instance, sequences from KITTI dataset [49]), we use a perspective camera model and perform piecewise perspective reconstruction. Similar to orthographic case, we do rigid perspective reconstruction on each individual part and then stitch connected parts into different objects. For our piecewise rigid reconstruction, we assume the calibration is known, and the only existing ambiguity between parts is the depth/scale ambiguity, which can then be resolved by enforcing the constraint that tracks belonging to two or more parts should be reconstructed at the same depth by each part model.

We use an incremental method to reconstruct the 3D shape of each rigid part. First we initialize from two frames using the five-point algorithm [116], and then keep adding new frames into the process, triangulating new points and doing incremental bundle adjustment until all the frames are reconstructed.

4.5.3 Depth-map Densification

Our reconstruction algorithm is based on sparse feature tracks. To densify the 3D reconstruction, we apply Gaussian filtering on the sparse 3D tracks in xy-RGB image space using the fast implementation of [3] that performs filtering using the permutohedral lattice. Regions of the video far from any tracks in the xy-RGB space are assigned to a flat background billboard.

4.6 Experimental Results

Since we recover both a segmentation of the scene into multiple moving objects and a 3D model for each object, we evaluate both of these steps independently.

4.6.1 Evaluation of the Motion Segmentation Step

We evaluate the results of our *object-level* segmentation on the Berkeley Motion Segmentation Dataset using the tracks and evaluation tool proposed in [18]. Table 4.1 shows a comparison between the scores of our approach and the results from Brox and Malik's motion segmentation algorithm [18]. The results show that our method exhibits comparable performance to [18]. While our *over-segmentation* error is higher than [18], the *overall error* and *average error* are very close, and in some cases lower. Although our algorithm can be used for motion segmentation exclusively, it is geared towards 3D reconstruction of complex dynamic scenes. Providing object boundaries are respected, our 3D reconstruction method is unharmed by a slight over-segmentation given that we perform piecewise reconstruction. The same set of parameters was used for all the experiments. The results of Fayad *et al.* [39] show how our algorithm would perform without the novel edge breaking and sparsity terms. Objects are never discovered in sequences longer than 10 frames, and in the majority of the 10 frame long sequences no objects are discovered.

4.6.2 Evaluation of Orthographic Reconstruction

We demonstrate our approach on videos from the *Youtube-Objects* Dataset [96]. These are unconstrained real-world videos downloaded from YouTube, with the purpose of object detection in video [96]. Figure 4.1 shows reconstructions of a *bird*, a *cat*, a *motorbike* and a *footballer*. We show the decomposition into *parts*, *objects* and a 3D model of the objects from a novel viewpoint for one frame. Figures 4.7 and 4.10 show 3D reconstructions for further frames of the four sequences. Our algorithm shows a good segmentation of the scenes and a convincing 3D reconstruction of these challenging videos.

However, our approach also has some limitations. As our method is purely geometric, to achieve an accurate 3D reconstruction, there needs to be enough 3D information to be exploited in the sequence. In other words, for each object, the parts should have non-planar 3D shapes, not be degenerate and have enough out of plane rotations within the sequence. Figure 4.9 shows a failure example of our algorithm. In this dancer sequence, we perform reconstruction on the segmentation results with saliency cues, as shown in Figure 4.5. This sequence

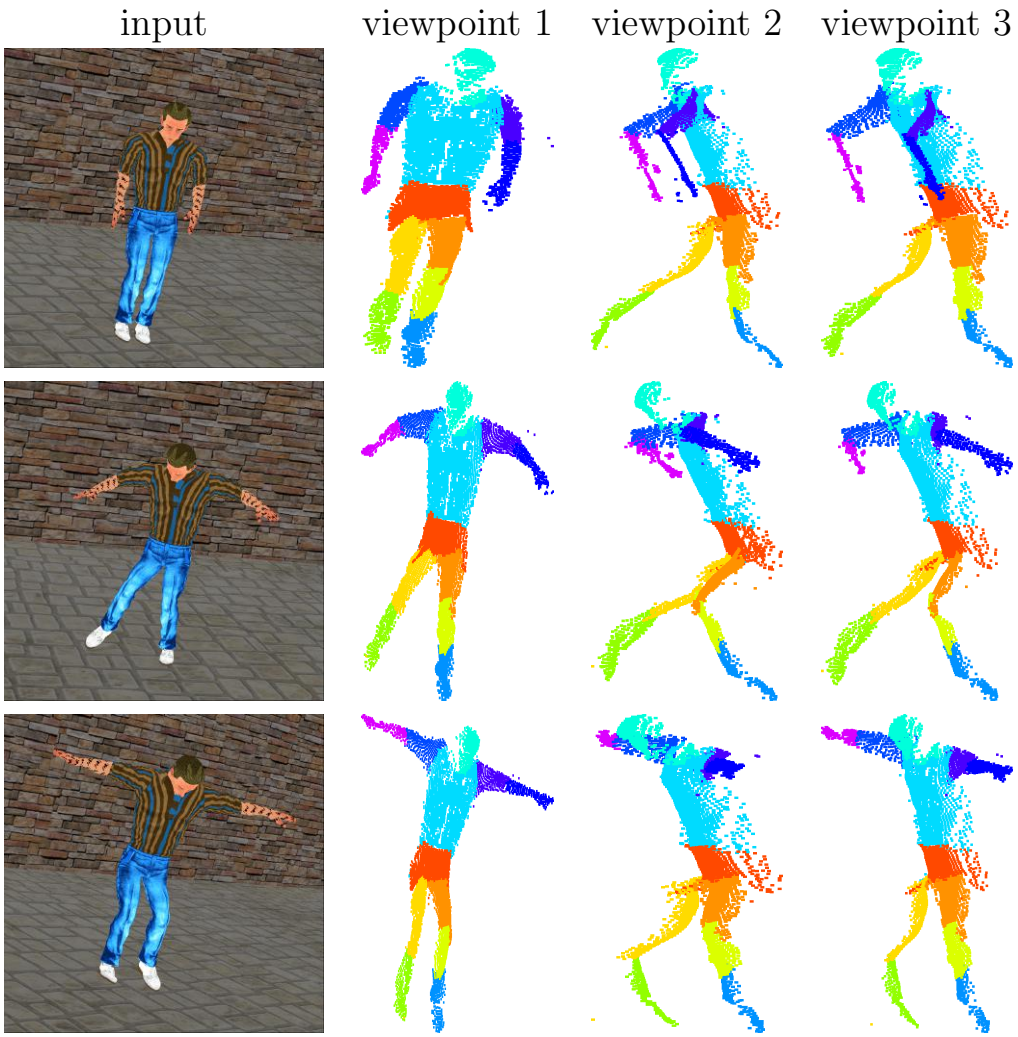


Figure 4.9: Failure orthographic reconstruction results on dancer sequence. From left to right, we show input image and reconstruction results from three different viewpoints. Results are generated by performing orthographic reconstruction on the segmentation results with saliency cues, as shown in Figure 4.5. This sequence is very challenging as the rigid parts are small, and our method fails to resolve the flip ambiguities of small and nearly planar parts, such as the arms and legs.

is very challenging as the rigid parts are small and some of them are nearly planar and degenerate. Our method has difficulties resolving both the temporal flip ambiguity (the flip of a single part across different frames) and spatial flip ambiguity (the flip between overlapping parts). For instance, as shown in the figure, the flip of both arms and legs are wrong, despite that the 2D projections on the input image (viewpoint 1 shows the result from camera viewpoint) look good.

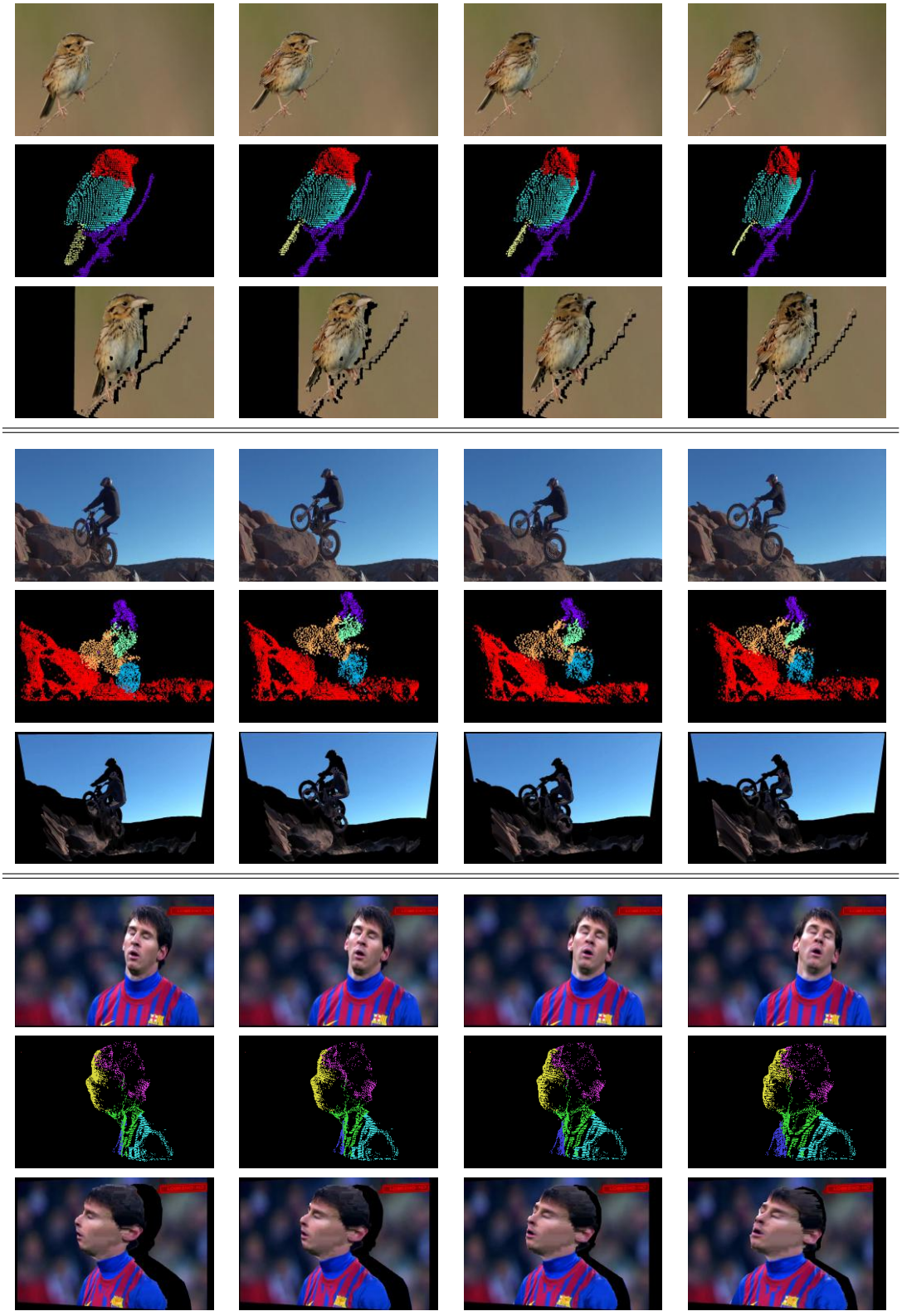


Figure 4.10: **Top:** and **Middle:** Reconstruction results for a bird sequence and a motorbike sequence from *Youtube-Objects* Dataset [96]. **Bottom:** Reconstruction results of a football footage. For all three sequences, we show registered sparse reconstruction of rigid parts and the dense results after densification.

4.6.3 Evaluation of Perspective Reconstruction

We evaluate our perspective reconstruction method on sequences from the KITTI dataset [49] both quantitatively and qualitatively. As sequences from KITTI mainly consist of static background and moving rigid objects, mostly cars, we turn off the overlapping between different parts since each rigid part readily corresponds to an individual object.

We show two different use cases of our approach, dense monocular depth estimation using two consecutive frames and monocular 3D sparse point cloud reconstruction using multiple frames.

4.6.3.1 Monocular Depth Map Estimation using Two Frames

In this experiment, we reconstruct a dense depth map from two consecutive frames taken from a monocular sequence of KITTI. First we obtain a sparse reconstruction of the background (we assume that the object with most points is the background) and of each moving object separately (by triangulating segmented sparse point tracks). Scale ambiguities between individual reconstructions are then resolved based on the assumption that all dynamic objects are always located on the background and occlude the static environment. We fix the scale of the background, and reconstruct the entire scene up to this fixed scale. For evaluation purposes, the unknown global scale will be estimated by registering the estimated depth values to the ground truth depth map. The final dense depth map is computed by linearly interpolating the sparse depth values of point tracks. In Figure 4.12, we show the obtained dense depth map after linear interpolation.

In order to compute the relative scales between foreground moving objects and the static environment, we first obtain a dense motion segmentation result based on superpixels and then enforce the constraint that the depth values of the boundary superpixels between dynamic objects and the static scene should be close. For dense segmentation, we simply divide the input image into superpixels [2], and assign all the pixels inside each superpixel segment the label with most support from the point tracks within the superpixel. Figure 4.11 shows an example of our sparse and dense segmentation results.

To compare with other state-of-the-art approaches, we implement the reconstruction algorithm recently proposed by Ranftl *et al.* [97]. In their paper, they focus on estimating dense depth maps of dynamic scenes from two consecutive monocular frames. Similarly to our approach, theirs also consists of two steps, motion segmentation and 3D reconstruction. For a fair comparison

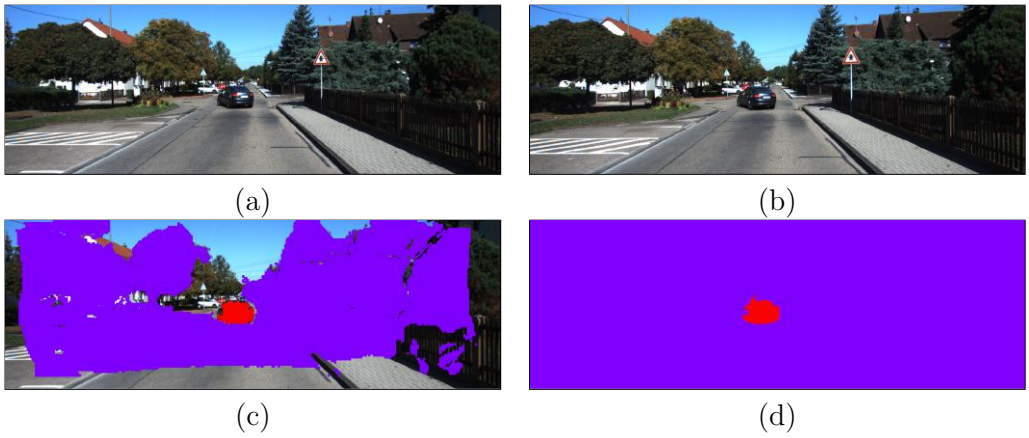


Figure 4.11: Motion segmentation results on two consecutive images from the KITTI dataset. (a)-(b) two consecutive images. (c) sparse motion segmentation result on point tracks. (d) dense segmentation result.

	Ours	Ranftl *et al.*
Mean relative error (MRE)	0.196	0.148

Table 4.2: Quantitative comparison between our method and Ranftl *et al.* [97] on the KITTI odometry set [49]. Our result is the mean error over 427 randomly sampled frames out of 43552.

between both reconstruction methods, we use our dense segmentation result as input for both 3D reconstruction approaches. In their 3D reconstruction step, they first estimate an initial reconstruction result by triangulation and then they perform a global optimization to obtain the final result. In fact, our method can be thought of as an initialization for their global optimization step. Figure 4.12 shows a qualitative comparison between our method and the approach from Ranftl *et al.*. For quantitative evaluation, we compute the mean relative error (MRE) with regard to the sparse laser scan ground truth, as used in Ranftl *et al.* [97]. As shown in table 4.2, our approach has an error of 0.196, compared with 0.148 from Ranftl *et al.*.

4.6.3.2 3D Point Cloud Reconstruction using Multiple Frames

To further evaluate our approach, we show multi-frame piecewise rigid perspective reconstruction results on sequences from the KITTI dataset [49] and from Zhang *et al.*'s [141] dataset. In this experiment, we first use an incremental method to reconstruct each rigid part separately and then assemble them together to achieve dynamic scene reconstruction results. We initialize from two frames with wide baseline and then perform incremental reconstruction.

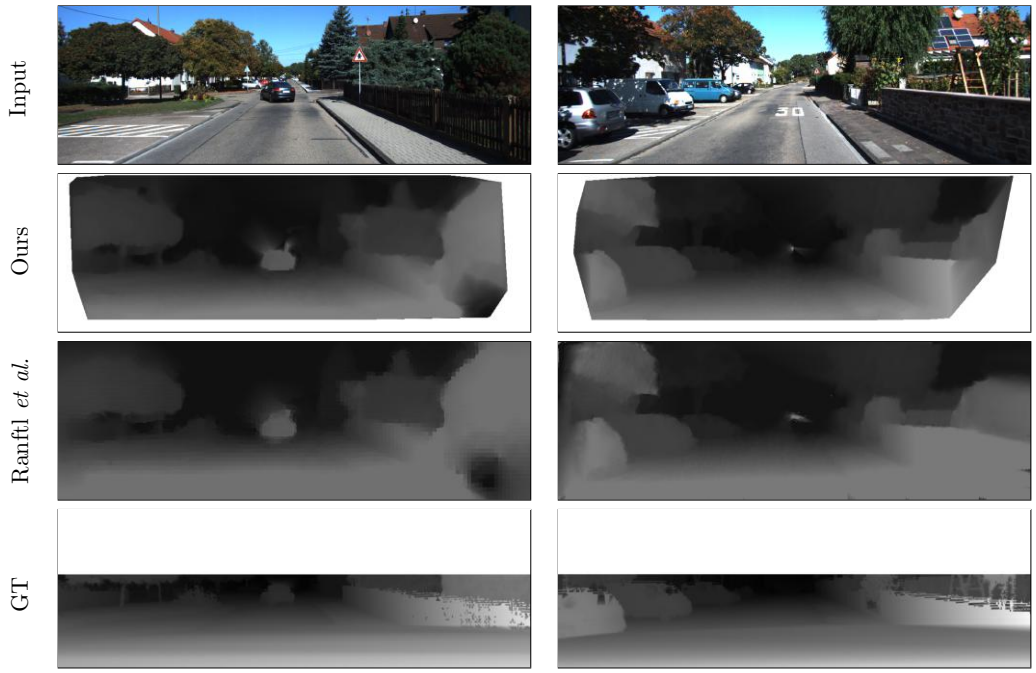


Figure 4.12: Depth map reconstruction results on two frames from the KITTI dataset. From top to bottom, we show input images, our dense reconstruction results (linear interpolation results from sparse reconstruction), our implementation of the reconstruction method from Ranftl *et al.* [97], and the ground truth (linear interpolation from sparse values). Note that the value of upper part is missing from the ground truth.

Specifically, in our implementation, we use the open source structure from motion library OpenSfM [47] for rigid perspective reconstruction. As each object is reconstructed independently, we need to solve the scale ambiguity to put everything together to create a dynamic scene reconstruction. To make it easier to register everything together, we create a graphical user interface where one could adjust the scale of each object manually. Examples shown in figure 4.13 and figure 4.14 are our scene reconstruction results after manual scaling.

Figure 4.13 shows our 3D reconstruction results for a short sequence with 15 consecutive frames, taken from KITTI. As shown in the figure, our method successfully segments the scene into the moving car and the static environment and reconstructs them accordingly. Figure 4.14 shows our reconstruction on the two-men sequence from Zhang *et al.* [141]. In this 30 frame long sequence, the two men move rigidly towards each other. From our result, the two people and the static background are correctly segmented and reconstructed.

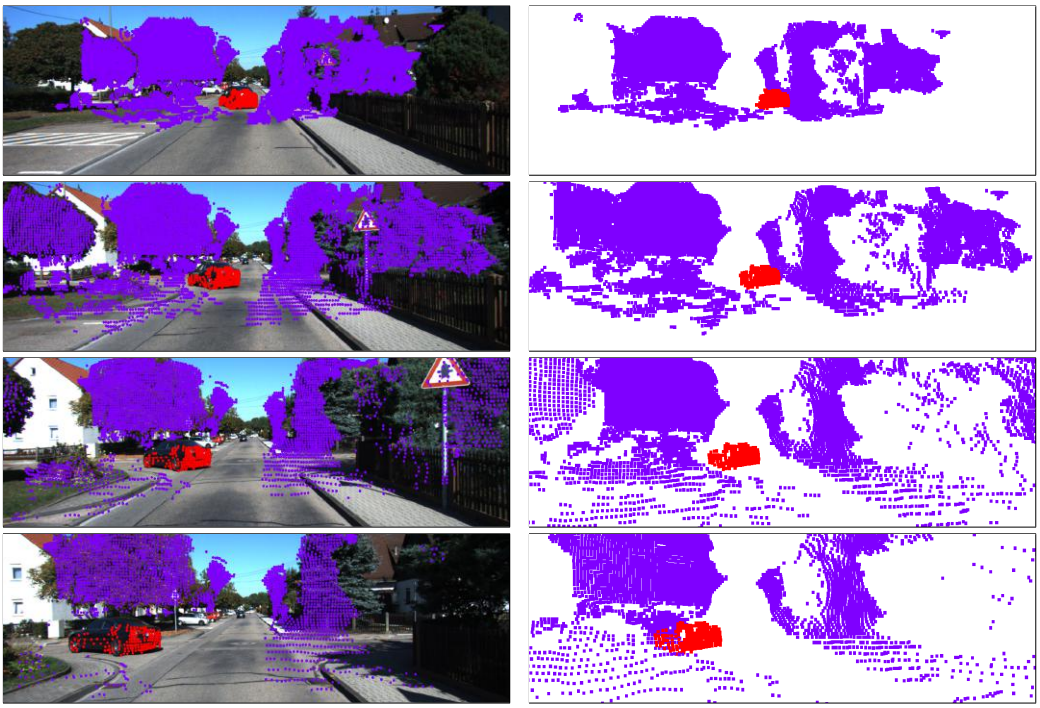


Figure 4.13: Multi-frame dynamic scene reconstruction results on a 15 frame short sequence from the KITTI dataset. From left to right, we show input images overlaid with segmented sparse point tracks, and corresponding reconstruction results (both the moving car and the static environment).

4.7 Conclusion

In this work we propose a fully unsupervised approach to the challenging problem of simultaneously segmenting a dynamic scene into its constituent objects and reconstructing a 3D model of the scene. We focus on the reconstruction of real-world videos downloaded from the web or acquired with a single camera observing a complex dynamic scene containing an unknown mixture of multiple moving and possibly deforming objects. Our method consists of two steps, motion segmentation and 3D reconstruction. The motion segmentation step segments the dynamic scene into rigid parts and combines overlapping parts into objects. In the subsequent 3D reconstruction step, we reconstruct each rigid part separately and stitch these part reconstruction results into a consistent whole scene reconstruction. Our results show examples of successful orthographic reconstruction on videos from the *Youtube Objects* dataset and perspective reconstruction on KITTI sequences, *etc*.

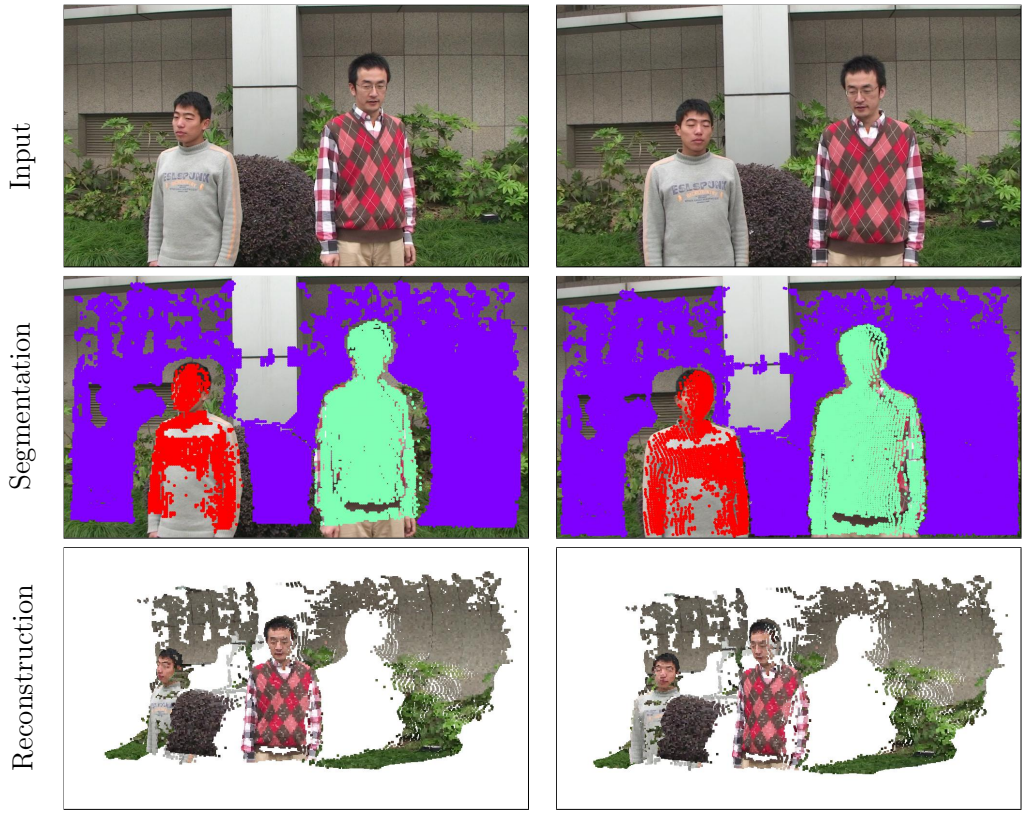


Figure 4.14: Segmentation and reconstruction results on two-men sequence from Zhang *et al.* [141]. We successfully segment the scene into three rigid components, the static background and two moving person. Last row shows the corresponding reconstruction results.

Chapter 5

Dense, Direct, Deformable: Template-Based Non-Rigid 3D Reconstruction from RGB Video

5.1 Introduction

Two common limitations remain with most NRS*f*M and *shape-from-template* formulations: *(i)* they are typically feature-based which leads to sparse reconstructions or failure with low-textured surfaces and *(ii)* estimation of 2D correspondences and 3D shape inference are decoupled and not solved simultaneously in a direct approach. So far the problem of jointly estimating dense point correspondences and non-rigid 3D geometry from monocular video has received very little attention. Garg *et al*. [45] demonstrated a dense per-pixel NRS*f*M approach but it required dense 2D correspondences to be pre-computed using a multi-frame optical flow method. Pixel-based approaches to template-based reconstruction have been proposed by Malti *et al*. [78] and Suwajanakorn *et al*. [119] but they were only demonstrated on planar surfaces (cloth or paper) [78] or worked exclusively for faces [119].

In this chapter we describe a template-based direct approach to deformable shape reconstruction from monocular sequences. Our contribution is an end-to-end system that builds a dense template from an initial rigid subsequence and subsequently estimates the deformations of the mesh with respect to the 3D template by minimizing a robust photometric cost. Unlike previous template-based direct methods [78, 119] we demonstrate our approach on a variety of

	Zollhofer *et al.* [144]	Malti *et al.* [78]	Suwajanakorn *et al.* [119]	Garg *et al.* [45]	Newcombe *et al.* [86]	Dou *et al.* [33]	Ours
Template-free	✗	✗	✗	✓	✓	✓	✗
Direct	✓	✓	✓	✗	✓	✓	✓
RGB-only	✗	✓	✓	✓	✗	✗	✓
Monocular	✗	✓	✓	✓	✓	✗	✓
Perspective camera	✓	✓	✗	✗	✓	✓	✓
Frame-to-frame	✓	✓	✓	✗	✓	✓	✓
Generic shapes	✓	✗	✗	✓	✓	✓	✓
Closed mesh with self-occlusion handling	✓	✗	✗	✗	✓	✓	✓

Table 5.1: Comparison of our approach with other dense competitors for reconstructing deformable shapes. Ours is the only template-based dense approach that only uses monocular RGB data; is frame-to-frame; direct; and suitable for reconstructing generic shapes.

generic complex non-planar meshes. While our algorithm is not real-time, it is sequential and relatively fast, typically requiring 2 seconds per frame on a standard desktop machine to optimize a mesh with approximately 25,000 vertices. Table 5.1 shows a comparison between our approach and state of the art methods. Ours is the only template-based approach that satisfies all the properties listed in the table.

5.2 Related Work

Very few methods attempt dense and direct reconstruction of non-rigid shapes from monocular sequences. There are three areas of research that have inspired and influenced our work: *non-rigid structure from motion*, *shape-from-template* and *RGB-D based non-rigid capture*. We now describe the most related approaches from each of these fields.

Although clearly inspired by the advances in **non-rigid structure from motion** methods [28, 92, 125], which can typically reconstruct non-rigid surfaces of generic shapes from monocular video while learning a low rank model that explains the deformations, our approach departs from them significantly. In

particular, NRS*f*M formulations are batch and require (usually a small number of) point correspondences to be given as input. In contrast, the distinguishing features of our approach are that it is direct, dense, and frame-to-frame.

The most related NRS*f*M method to ours is the dense monocular non-rigid reconstruction algorithm by Garg *et al.* [45]. Although their algorithm reconstructs dense per-pixel models, noticeably, it is a batch process that requires multi-frame optic flow over the entire sequence as an input. No attempt was made to solve the dense correspondence and reconstruction problems simultaneously. As such, if the flow generation fails, a good reconstruction is not possible.

Our method also shares strong similarities with work in the area of **shape from template** [6, 108, 90, 110]. Many approaches have been proposed mostly taking advantage of the constraints imposed by isometric or conformal deformations [6, 79, 109]. While most template approaches are feature-based and only reconstruct a small number of points, Malti *et al.* [78] departs by proposing a direct pixel-based variational framework that exploits visibility constraints. However, their method was only demonstrated on flat isometric surfaces. The recent work of Suwajanakorn *et al.* [119] reconstructs RGB-only videos of faces of celebrities. Similarly to our method, they formulate template-based non-rigid reconstruction as a frame-to-frame energy minimization that optimizes a direct photometric cost. However, their method is limited to reconstructing human faces as their template reconstruction approach is specifically tailored to them. In contrast, our template reconstruction step uses a dense volumetric multiview stereo formulation that is generic and can be used for any type of shape. In addition, our energy makes use of robust norms for the data and regularization terms; explores more sophisticated smoothness priors, such as local rigidity (as-rigid-as-possible [115]); and imposes temporal smoothness. Also related is the monocular face capture system of Garrido *et al.* [48]. While their work also minimizes a photometric cost and the deformations with respect to a template model, theirs is a sophisticated blend-shape model specifically built to capture the deformations of human faces.

Our work has been largely inspired by recent advances in non-rigid tracking **using depth cameras** [33, 86, 144]. Zollhofer *et al.*'s [144] is the most related approach since their setup is directly comparable to ours — a multi-scale template is built first from a rigid sub-sequence, followed by dense non-rigid monocular tracking. However, while their method uses both the depth and the RGB channels, ours only uses RGB images as input and can be seen as its

RGB-only equivalent. More recently, DynamicFusion [86] takes only the depth point cloud from a Kinect as input and estimates a warp back into a canonical reference scene, where a model is progressively denoised and completed. While [144] makes use of image data to help with frame-to-frame alignment [86] makes no use of any image data. However, since DynamicFusion system uses a fixed reference frame where the volumetric model is incrementally updated, it cannot deal with fast motion, major shape deformation or topology changes. To overcome these limitations, Dou *et al.* [33] proposed a new multi-view real time performance capture system for challenging scenes. By periodically resetting the reference frame to adapt to shape changes over time and robustly fusing data and reference volumes based on correspondence estimation and alignment error, their new Fusion4D system can robustly handle large frame-to-frame motion and topology changes.

While our work is related to and certainly inspired by these depth-based formulations, the underlying estimation problems are fundamentally different. The availability of a depth image for each frame turns the problem of 3D estimation of non-rigid geometry into one of denoising or fusion, while our monocular RGB-only reconstruction problem must infer the 3D deformations of a template purely from 2D image motion data.

Table 5.1 summarizes our main contributions and the differences with respect to the six most closely related approaches, namely the dense NRS*f*M approach of Garg *et al.* [45], the direct template-based monocular reconstruction approach of Malti *et al.* [78], the total face reconstruction system of Suwajanakorn *et al.* [119], real-time RGB-D non-rigid reconstruction system of Zollhofer *et al.* [144], DynamicFusion system of Newcombe *et al.* [86] and the multi-view Fusion4D system of Dou *et al.* [33].

In summary, ours is the only RGB-only, template-based, monocular, dense and direct approach to non-rigid reconstruction that is sequential and suitable for generic shapes and closed meshes.

5.3 Problem Formulation

We consider a perspective RGB camera with known internal calibration observing a non-rigid mesh deforming over time. The goal of our algorithm is to estimate, at each time-step t, the current 3D coordinates of the N vertices of the dense non-rigid mesh $\boldsymbol{S}^t = [\dots \boldsymbol{s}_i^t \dots]$, $i = 1..N$, as well as the overall rigid rotation and translation $(\boldsymbol{R}^t, \boldsymbol{t}^t)$ that align the deformed shape and a reference 3D template.

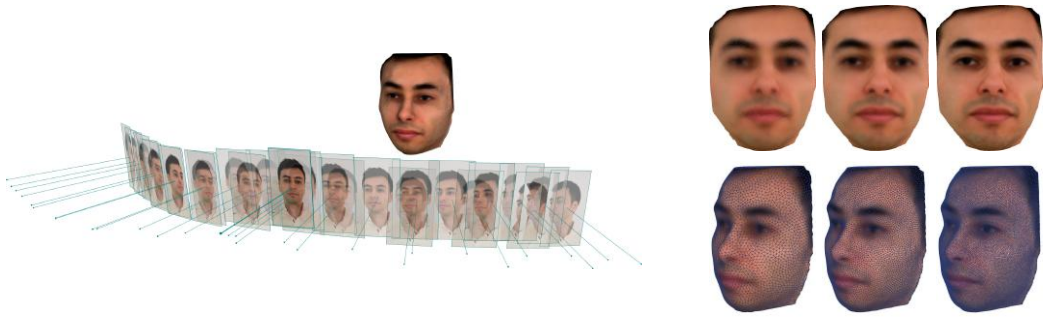


Figure 5.1: Template acquisition step. **Left:** A volumetric representation is generated from stereo depth maps taken over a rigid subsequence. This is then transformed into a coloured mesh. **Right:** The three scales of the template used to robustly estimate deformations.

The only inputs to our algorithm are the current RGB image $\boldsymbol{I}^t(x, y)$ observed at time t and a template shape $\widetilde{\mathbf{S}} = [\dots \widehat{\mathbf{s}}_i \dots]$, $i = 1..N$, which is acquired automatically in a preliminary template acquisition step using the multi-view stereo dense volumetric approach of [19]. Typically the user acquires a short rigid sequence to capture the 3D coordinates of the template mesh which is then subsampled to create a multi-resolution hierarchy of coarse-to-fine templates. This template acquisition step is described in more details in section 5.4. The template is then converted to a triangular mesh, consisting of N vertices and M edges.

Once the template has been acquired, our system turns to perform frame-to-frame non-rigid alignment of the 3D shape given only the current frame as input. Although optimization is initialized using the shape from the previous frame $\boldsymbol{S}^{t-1}$, once the template has been generated, the optimization objective does not depend on any other frames. As such, unlike most approaches to non-rigid structure from motion [28, 45, 92, 125], it scales to the streaming of long sequences, with the complexity of optimization guaranteed to grow linearly to the number of frames.

5.4 Step 1: Template Shape Acquisition

The first stage in our process is to obtain a rigid template mesh of the shape. We denote the whole shape as a $3 \times N$ matrix $\widehat{\mathbf{S}}$, and $\widehat{\mathbf{s}}_i$ as the i^{th} vertex on the mesh. We require a set of M images (we used $M \sim 30$) of the shape under a rigid transformation. These are obtained by subsampling a set of frames from a short video where either the object is static and the camera moves or the camera is static and the object is moved under a rigid transformation.

Figure 5.1 provides an example of the output of this process. As shown in the figure, this step takes sampled images as input and generates a set of coloured coarse-to-fine meshes.

The process of the template acquisition is an application of an existing multi-view stereo (MVS) technique [19]; consequently we provide only an overview of the process with appropriate references to the methods used.

Extrinsic Calibration The collection of frames from the video were calibrated automatically using an implementation (VisualSFM [136]) of standard rigid structure-from-motion (SfM). This was observed to be robust to some incompatible motion in the background. If there is too much background clutter in the image then an automatic segmentation of the foreground can be attempted using a fixation condition (that the center of the image fixates on the object of interest) [20].

Depth-Map Extraction Once we have a calibrated set of frames, we extract a depth-map using the stereo method of [19]. For each (reference) image, we take the two closest viewpoints as neighbouring images and extract the best $K = 9$ normalized cross-correlation (NCC) scores matching with 13×13 pixel windows. These are then filtered to provide a single depth estimate (or unknown label) using the default filtering parameters as specified in [19].

Mesh Estimation The last stage is to extract the template mesh by combining all the individual depth-maps in a single global optimization. As suggested in [19], we combine the depth-maps to recover a single watertight mesh $\widetilde{\mathbf{S}}$ using the volumetric fusion technique of [133] combined with the probabilistic visibility approach of [55].

Template Hierarchy The output of the fusion stage is a watertight mesh $\widetilde{\mathbf{S}}$. From this we build a multi-scale representation of the mesh as shown in Figure 5.1 (right). This is achieved by iteratively down-sampling and refining the template mesh using the isotropic surface remeshing method (and implementation) of Fuhrmann *et al.* [42]. Finally, a colour $\widehat{\mathbf{I}}_i$ is associated to each vertex i; this is the median colour over all the frames in the rigid subsequence in which the projected vertex is visible.

To avoid aliasing when colouring the low resolution meshes, we blur each of the input images with a length-scale given by the median mesh edge length projected into the corresponding camera view. Figure 5.2 shows a triangulated example of the multi-scale coloured mesh representation.

Figure 5.2: An example of our multi-scale template meshes generated by iterative mesh down-sampling and refinement. **Top:** From left to right the meshes contain approximately 5, 10, and 25 thousand vertices respectively. **Bottom:** The highest levels of the templates for the dog and ball sequences.

5.5 Step 2: Non-Rigid Model Tracking

5.5.1 Our Energy

Our objective is made of a balanced combination of five terms: *(i)* a *photometric error* which captures the expected colour of each *visible* vertex in the template; *(ii)* a *total variation term* on the gradient of the 3D displacements with respect to the template; *(iii) as rigid as possible* local regularization – this term allows the mesh to rotate locally without imposing a penalty; *(iv)* a *rotation total variation term* on the gradient of the local 3D rotations of each vertex with respect to the template and *(v)* a *temporal smoothness* term that penalizes strong frame-to-frame deformations.

The per-frame objective takes the form:

$$\begin{aligned} E(\boldsymbol{S}, \{\boldsymbol{A}_i\}, \boldsymbol{R}, \boldsymbol{t}) = & E_{\text{data}}(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{t}) + \lambda_r E_{\text{reg}}(\boldsymbol{S}, \{\boldsymbol{A}_i\}) \\ & + \lambda_a E_{\text{arap}}(\boldsymbol{S}) + \lambda_{rr} E_{\text{reg\_rot}}\{\boldsymbol{A}_i\} \\ & + \lambda_t E_{\text{temp}}(\boldsymbol{S}). \end{aligned} \tag{5.1}$$

where λ_r, λ_a , λ_{rr} and λ_t denote the relative weights between the terms. These terms are all required. The first term guarantees that the deformations of the template follow the image; the second term encourages locally smooth deformations while allowing sharp discontinuities which are needed to transition from parts of the object that deform strongly to those that do not; the third term approximates elastic deformation and encourages the deformation to be locally rigid; the fourth term encourages large articulation changes in the template shape. Finally, temporal smoothness is needed to avoid flickering.

For simplicity's sake, we drop temporal super-scripts where appropriate as much of the formulation does not depend on any other frames. We now define each of the terms of the energy in detail.

5.5.1.1 Photometric Data Term E_{data}

The **data term** E_{data} encourages a shape such that projection of the vertices into the current image has similar appearance to the template shape. In other words, minimization of this photometric cost encourages brightness constancy with respect to the colours $\widehat{\mathbf{I}}=\{\widehat{\mathbf{I}}_i\}$ of the mesh, built by back-projecting the images used to build the reference template $\widehat{\mathbf{S}}=\{\widehat{\mathbf{s}}_\mathbf{i}\}$ onto the vertices of the template. As we directly reconstruct closed meshes where much of the object is self-occluded, we first make an initial pass where we estimate the visibility of each vertex in the mesh. For additional robustness, we use a Huber loss.

$$E_{\text{data}}(\boldsymbol{S}, \boldsymbol{R}, \boldsymbol{t}) = \sum_{i \in \mathcal{V}} |\widehat{\mathbf{I}}_i - \boldsymbol{I}(\pi(\boldsymbol{R}(\boldsymbol{s}_i) + \boldsymbol{t}))|_\epsilon \tag{5.2}$$

where $\widehat{\mathbf{I}}_i$ is the colour of vertex $\widehat{\mathbf{s}}_i$ on the template mesh, $\boldsymbol{I}$ is the current image frame, $\mathcal{V}$ is the set of estimated visible vertices in the frame[1], $\{\widehat{\mathbf{s}}_i\}_1^N$ are the 3D vertices of the template, $\{\boldsymbol{s}_i\}_1^N$ are the 3D vertices of the shape in the current frame, $\pi(\cdot)$ is again the projection from 3D points to image coordinates, known

[1]This is generated by realigning the deformed mesh of the previous frame to minimize photometric error (see section 5.5.2.1), and z-buffering.

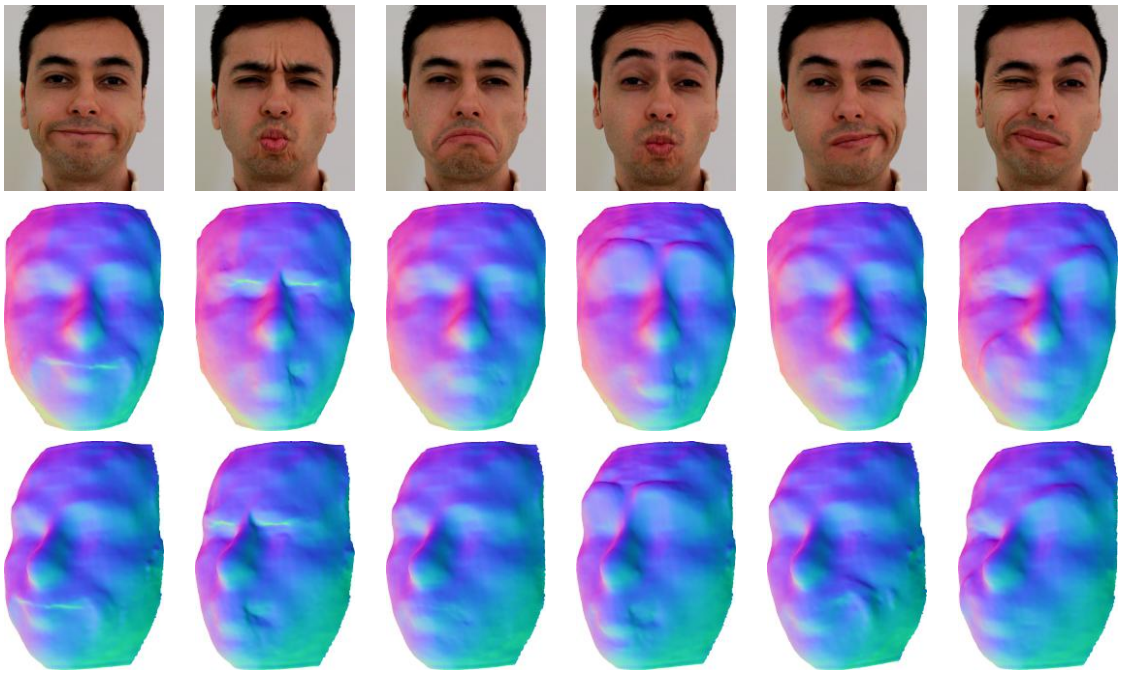


Figure 5.3: Direct deformable reconstruction from our algorithm on face1 sequence. Each column corresponds to different views of the same frame.

from camera calibration, and $|\cdot|_\epsilon$ denotes the Huber loss, which takes the form

$$|x|_\epsilon = \begin{cases} x^2/(2\epsilon) & \text{if } x^2 \leq \epsilon \\ |x| - \epsilon/2 & \text{otherwise.} \end{cases} \tag{5.3}$$

5.5.1.2 Spatial Regularization Term E_{reg}

The **regularization term** E_{reg} is a pairwise term that encourages spatially smooth deformations of the shape $\boldsymbol{S}$ with respect to the template $\widehat{\mathbf{S}}$.

$$E_{\text{reg}}(\boldsymbol{S}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \|(\boldsymbol{s}_i - \boldsymbol{s}_j) - (\widehat{\mathbf{s}}_i - \widehat{\mathbf{s}}_j)\|_\epsilon \tag{5.4}$$

Here $\mathcal{N}_i$ is the neighbourhood of i, and $\|\cdot\|_\epsilon$ is the vector analogue of the Huber loss formed by summing the standard Huber loss over all dimensions.

5.5.1.3 As Rigid as Possible Deformation Term E_{arap}

This cost was first proposed in [115] to allow deformable tracking of an initial mesh against a depth map. It takes the form

$$E_{\text{arap}}(\boldsymbol{S}, \{\boldsymbol{A}_i\}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \|(\boldsymbol{s}_i - \boldsymbol{s}_j) - \boldsymbol{A}_i(\widehat{\mathbf{s}}_i - \widehat{\mathbf{s}}_j)\|_2^2 \tag{5.5}$$

where the variables A_i are per-point local rotations. Essentially this cost allows for local rotations to take place in the mesh without penalty so long as the relative locations between points in the neighbourhood of i remain constant. It can be interpreted as a prior that allows for elastic style deformations of meshes. This cost has been widely used in non-rigid motion modelling [144, 86, 33].

5.5.1.4 Spatial Rotation Regularization Term $E_{\text{reg\_rot}}$

As E_{reg} penalizes the gradient of 3D displacements, in the case of large articulation motion, this cost will be relatively large due to the fact that the gradient of 3D displacements will be approximately constant in the whole articulated region. Therefore this term will penalize strong articulated motions. To allow large articulations, we introduce a new regularization term $E_{\text{reg\_rot}}$ on local rotations in addition to the 3D displacements.

$$E_{\text{reg\_rot}}(\{\boldsymbol{A}_i\}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \|(\boldsymbol{A}_i - \boldsymbol{A}_j) - (\widehat{\mathbf{A}}_i - \widehat{\mathbf{A}}_j)\|_\epsilon \tag{5.6}$$

where $\boldsymbol{A}_i$ is the local arap rotation of vertex i. Unlike E_{reg}, $E_{\text{reg\_rot}}$ will be small in the articulated region, only taking nonzero values around the joints, and therefore encouraging large articulated motion.

5.5.1.5 Temporal Smoothness E_{temp}

The temporal regularization encourages smooth deformations from frame to frame and can be formulated as

$$E_{\text{temp}}(\boldsymbol{S}, \boldsymbol{t}) = \|\boldsymbol{S} - \boldsymbol{S}^{t-1}\|_{\mathcal{F}}^2 + \|\boldsymbol{t} - \boldsymbol{t}^{t-1}\|_2^2 \tag{5.7}$$

where $\boldsymbol{S}^{t-1}$ and $\boldsymbol{t}^{t-1}$ are the shape and the translation in the previous frame and $\|\cdot\|_{\mathcal{F}}$ denotes the Frobenius norm of a matrix. The need for this term is most apparent when viewing a video of the reconstruction. Although a small amount of temporal regularization only alters the shape a little, it substantially reduces frame-to-frame flickering, while the temporal smoothness in the translation

prevents explaining deformations as perspective effects.

5.5.2 Energy Optimization

For reasons of robustness and efficiency, optimization is performed in a two step form over rotation and translation first, and shape secondly, and using a 3-layer spatial pyramid.

5.5.2.1 Initialization

We optimize this objective in a two step form: first the rotation and translation are estimated using the shape from the previous frame.

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} |\widehat{\mathbf{I}}_i - \boldsymbol{I}(\pi(\boldsymbol{R}(\boldsymbol{s}_i^{t-1}) + \boldsymbol{t}))|_\epsilon \tag{5.8}$$

Then, holding the global rotation and translation constant, $\boldsymbol{s}^t$ is estimated. $\boldsymbol{R}$, $\boldsymbol{t}$, and $\boldsymbol{S}^t$ (at the coarsest level of the pyramid) are initialized using the solution taken from the previous frame, and optimization is performed using the conjugate gradient based solver from Ceres [4].

5.5.2.2 Coarse-to-fine optimization and Deformation Graph

Both the rotation and translation cost 5.8, and the shape cost 5.1 are optimized over a set of 3-level coarse-to-fine images and shape templates, with each layer of the pyramid being two times larger than the coarser layer directly above it. As we move down the pyramid from coarse to fine, the 3D vertices are propagated to the next level of the hierarchy using a prolongation step as described in Sumner *et al.* [117]. The weights are pre-computed when the template mesh is created.

$$w_k(i) = \exp(-\|\widehat{\mathbf{s}}_i - \widehat{\mathbf{s}}_k\|_2^2 / 2\sigma^2) \tag{5.9}$$

where $\widehat{\mathbf{s}}_i$ is the 3D position of vertex i on the finest level template mesh, while $\widehat{\mathbf{s}}_k$ is the position of vertex k on the coarse mesh. σ is given by the largest distance between all the K nearest nodes in the coarse mesh and vertex i, and the weights $w_k(i)$ are then normalized to sum up to 1.

Based on the coarse level mesh $\{\boldsymbol{s}_k\}$, local rotations $\{\boldsymbol{A}_k\}$ and weights $w_k(i)$, we estimate the location of the vertices on the fine level mesh with:

$$\boldsymbol{s}_i = \sum_{k=1}^{K} w_k(i)(\boldsymbol{A}_k(\widehat{\mathbf{s}}_i - \widehat{\mathbf{s}}_k) + \boldsymbol{s}_k) \tag{5.10}$$

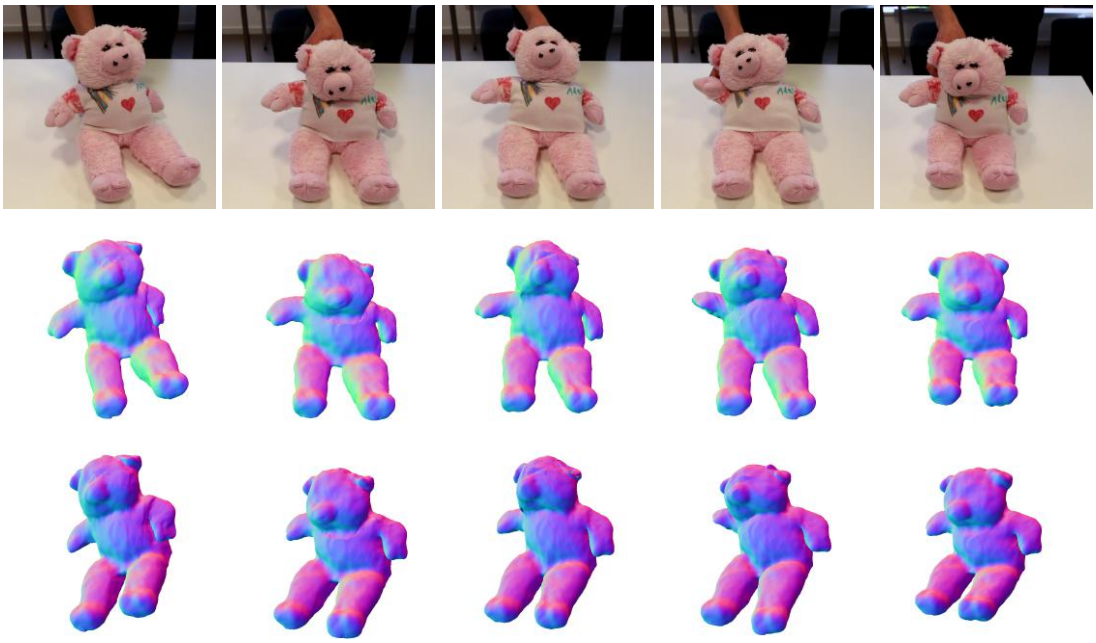


Figure 5.4: Direct deformable reconstruction on the pig sequence. Notice that our method is able to capture the motion of the right hand, the deformation of the head and the large articulation between the body and legs.

Prolongation is also applied to the arap rotations $\{\boldsymbol{A}_i\}$ by estimating the best local rotations between the fine template mesh and the current mesh.

In our energy formulation 5.1, the number of variables of the optimization problem is $6N + 6$, where N is the total number of vertices in the mesh. In our experiments, we use a 3-level mesh pyramid with 5k, 10k and 25k vertices, which gives rise to 30k, 60k and 150k variables respectively. To compute the mesh deformations more efficiently, we could use upper level mesh vertices (not necessarily the one directly above) as deformation nodes and compute the vertex position $\boldsymbol{s}_i$ of the fine level mesh via prolongation 5.10. Notice that in this case, K is the number of neighbouring deformation nodes of vertex i, $w_k(i)$ is the interpolation weight of node k on vertex i. Assuming there are M deformation nodes, the number of variables of the energy will be $6M + 6$ and therefore the algorithm will be much more efficient when M is much smaller than N.

These deformation nodes could be introduced in both data term and other regularization terms. Furthermore, we could use different deformation nodes for the data term and regularization terms. To induce long range regularization over the deformation nodes, we could use a hierarchical deformation graph.

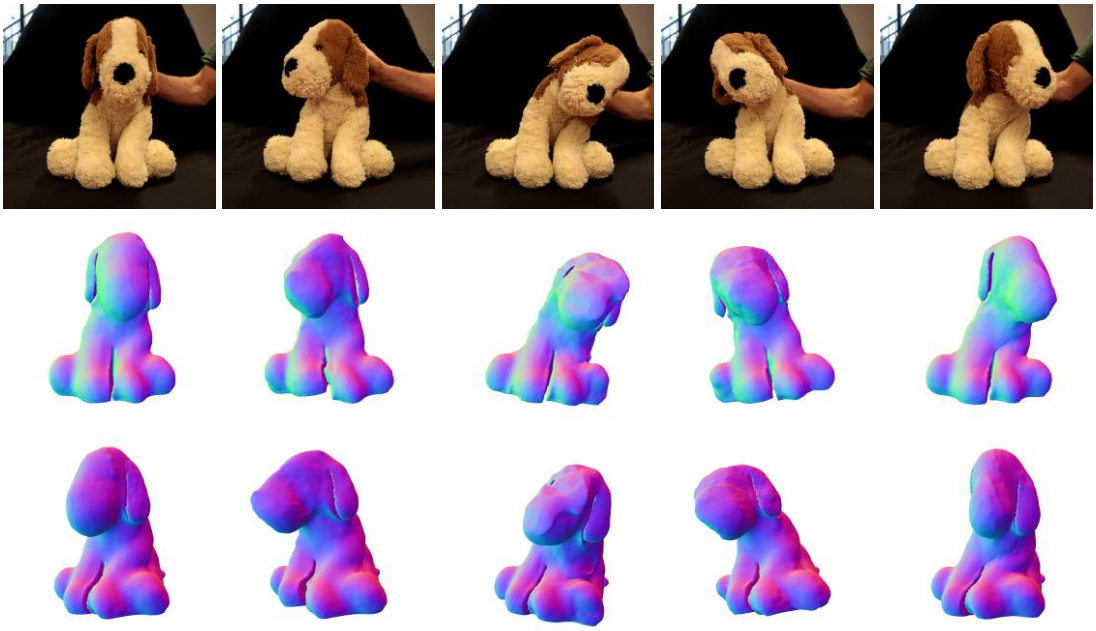


Figure 5.5: Direct deformable reconstruction on the dog sequence. Despite the large deformation created by the person's hand, our method successfully tracks the motion of the dog's head and the deformation of the neck.

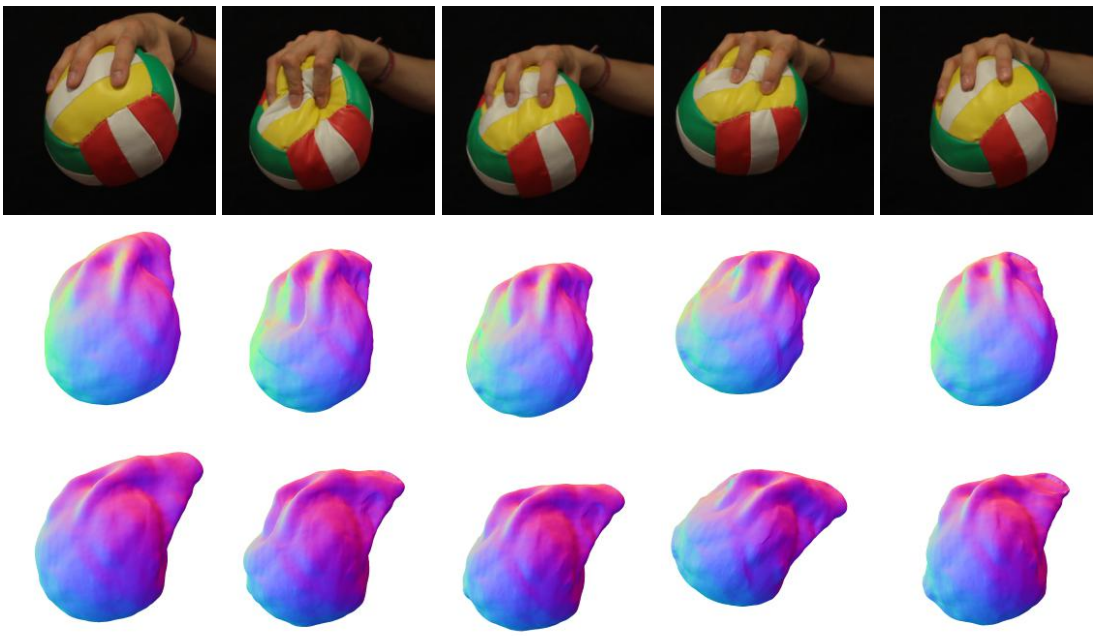


Figure 5.6: Direct deformable reconstruction on the ball sequence. Our method is able to track the motion of the ball as well as the deformation induced by hand pressing.

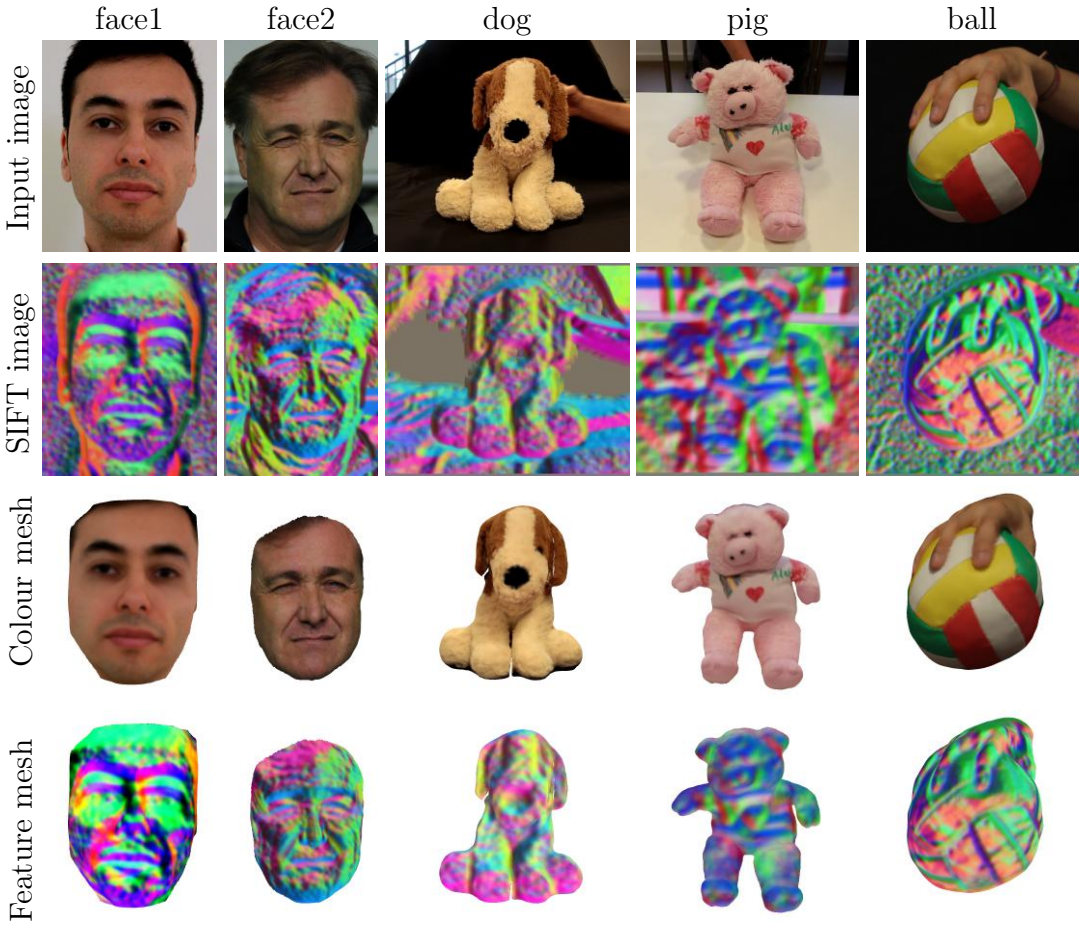


Figure 5.7: Input image, dense SIFT feature image, colour mesh and SIFT feature mesh of 5 different sequences. To show SIFT feature image and mesh, we use the first 3 principal components of 128 dimensional SIFT features as RGB values.

5.6 Robust Data Term

The photometric data term E_{data} from section 5.5.1 is based on the *brightness constancy assumption*, i.e. the corresponding point in the image should have similar colour and brightness as the template mesh vertex. In our scenario, this assumption will be violated when there are either illumination changes or shading effects caused by strong local mesh deformations. As shown in Figure 5.8, there is significant intensity change around the eye and mouth region when the face deforms over the whole sequence and intensity based tracking fails to capture the mesh deformation. In this section, we introduce two new data terms to robustly deal with appearance changes in the tracking.

5.6.1 SIFT Data Term

Feature descriptors, such as SIFT [76], SURF [8] and ORB [102], have been shown to be robust to large illumination and viewpoint changes. In order to overcome the shortcomings of intensity based tracking, we propose to use the SIFT feature descriptor image for tracking instead of the RGB values.

Specifically, we compute dense SIFT feature images offline using the VLFeat library [130]. Due to memory limitations, we perform PCA (Principal Component Analysis) on the 128 dimensional SIFT features and only keep the first 3 principal dimensions. We compute feature images for the rigid and non-rigid parts of the sequences. The rigid frames are used for template building while the non-rigid one for online tracking. In the template creation stage, instead of attaching RGB colours to mesh vertex, we attach 3-channel SIFT features.

Figure 5.7 shows the input image, dense SIFT feature image, colour mesh and SIFT feature template mesh for face1, face2, dog, pig and ball sequences.

5.6.2 NCC Data Term

Normalized Cross-Correlation(NCC) is a widely used distance measure in template matching due to its simplicity and robustness to lighting changes. It is a distance metric between two image patches. Specifically, for two patches $\mathbf{I_p}$ and $\mathbf{I_q}$, the NCC score is defined as follows:

$$NCC(\mathbf{I_p}, \mathbf{I_q}) = \frac{\mathbf{I_p} - \overline{\mathbf{I_p}}}{\|\mathbf{I_p} - \overline{\mathbf{I_p}}\|} \cdot \frac{\mathbf{I_q} - \overline{\mathbf{I_q}}}{\|\mathbf{I_q} - \overline{\mathbf{I_q}}\|} \tag{5.11}$$

where $\overline{\mathbf{I_p}}$ and $\overline{\mathbf{I_p}}$ are the mean values of patch $\mathbf{I_p}$ and $\mathbf{I_q}$ respectively. NCC measures the similarity of intensities in two neighbourhood regions, and is invariant to the change in average value or intensity range in the regions.

As an alternative to the intensity based photometric data term E_{data}, we compute the NCC score between local template mesh region and corresponding 2D projections on the input image.

5.7 Frame-to-frame data term

The formulation we introduced in section 5.5.1 is a frame-to-model tracking method, where the data term is based on the matching between a fixed template and an input frame. However, using a fixed template could fail to handle appearance changes, for example, sudden changes in the environment lighting or shading changes due to local deformations, which might be critical as our

goal is to track mesh deformation accurately. In order to adapt to possible changes in the intensity over time, we compute the data term based on the difference between the intensities of projections on the previous frame and the current frame. In other words, we update the colours of template mesh vertices for each frame:

$$\widehat{\mathbf{I}}_i = \boldsymbol{I}^{t-1}(\pi(\boldsymbol{R}^{t-1}(\boldsymbol{s}_i^{t-1}) + \boldsymbol{t}^{t-1})) \tag{5.12}$$

Similarly, we could compute frame-to-frame data terms using NCC and SIFT features by updating the intensities or features based on the projections of tracking results from the previous frame.

5.8 Experimental Results

In this section we show qualitative results of our method on a variety of non-planar 3D meshes; a qualitative comparison between the results with different combinations of regularization terms and a quantitative evaluation on the face2 sequence from Valgaerts *et al.* [128]. Our results can be best viewed in the video.[2]

Qualitative results on non-planar meshes We show results on some new sequences acquired with a handheld camera. Example sequences include a *face* (Figure 5.3) , two soft toys – a *pig* (Figure 5.4) and a *dog* (Figure 5.5), and a *ball* being squeezed by a hand (Figure 5.6). These sequences show a wide range of deformations of a varying set of shapes, with different degrees of elasticity. The reconstructions and deformations generated are convincing.

Qualitative results of using different regularization terms To justify the effectiveness of each regularization term, we compared with different combinations of regularizers. Figure 5.10 and Figure 5.11 show the tracking results with and without the arap term E_{arap}, spatial rotation regularization $E_{\text{reg\_rot}}$ term and temporal smoothness term E_{temp}.

As shown in the left of figure 5.10, using only the spatial regularization term E_{reg} does not allow large deformations from the template and cannot capture the large articulation movement when the dog turns sideways its head. While the arap term E_{arap} allows large deformations, it offers too much freedom that the left ear gets curly when the dog rotates its head, as shown in the middle. With the right combination of all three regularization terms (E_{reg}, E_{arap} and $E_{\text{reg\_rot}}$) we show that the energy is able to capture large deformation while not too flexible to induce unnecessary deformations.

[2]Please see `http://visual.cs.ucl.ac.uk/pubs/ddd/` for video.

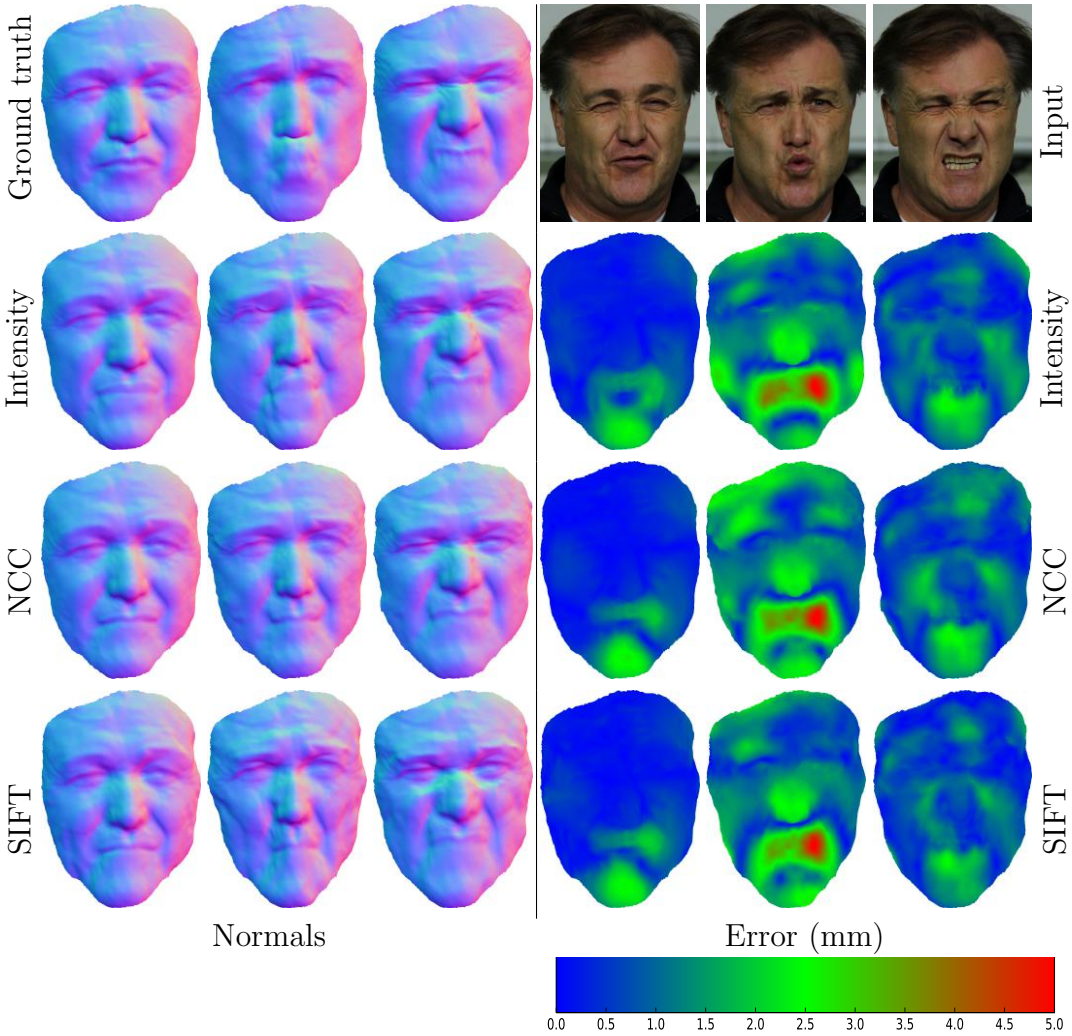


Figure 5.8: Comparison results between intensity, NCC and SIFT feature based tracking with frame-to-model data term. Left figure shows the reconstruction results compared with ground truth mesh, right figure shows the error heatmap. For the error heatmap, blue corresponds to low error while red means high error.

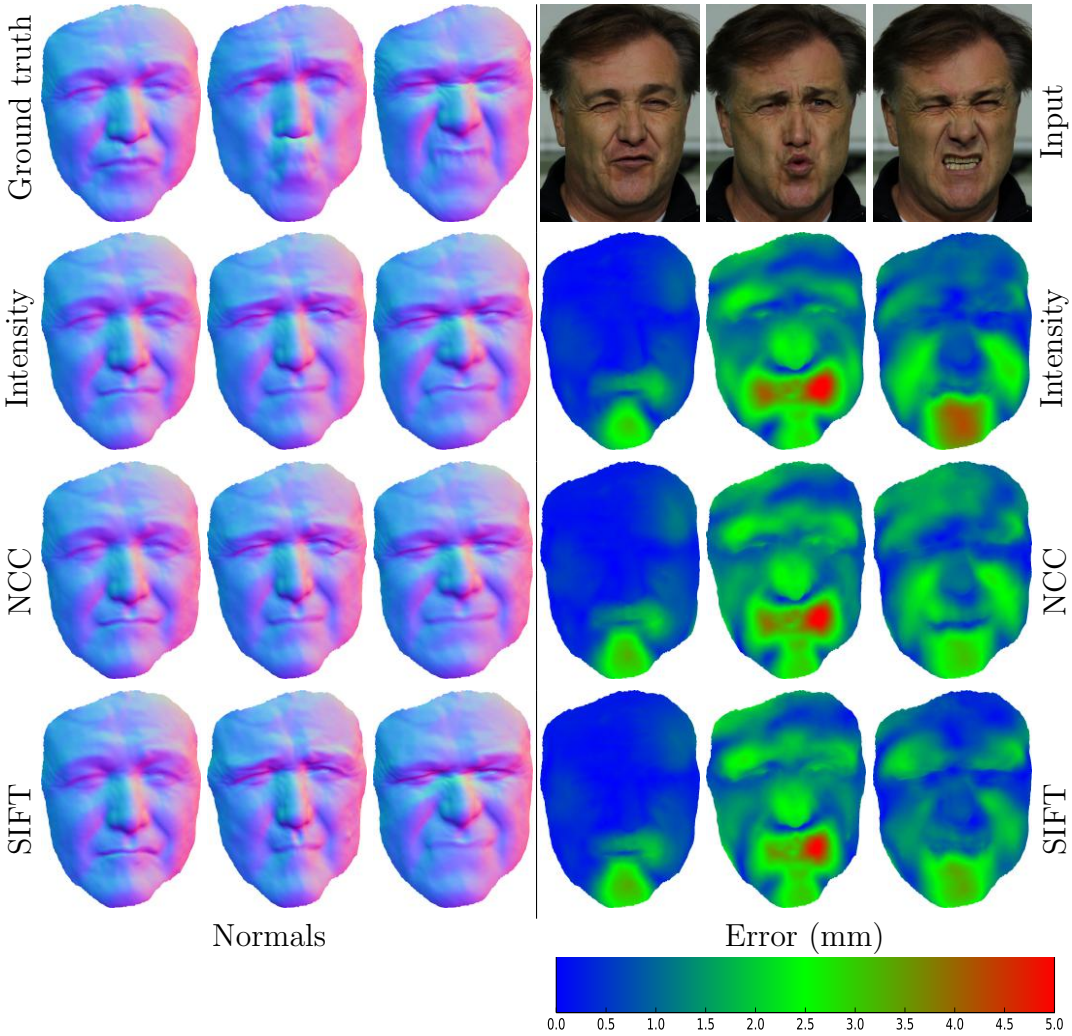


Figure 5.9: Comparison results between intensity, NCC and SIFT feature based tracking with frame-to-frame data term. Left figure shows the reconstruction results compared with ground truth mesh, right figure shows the error heatmap. Different from Figure 5.8, for frame-to-frame data term, NCC gives worse result than intensity based tracking.

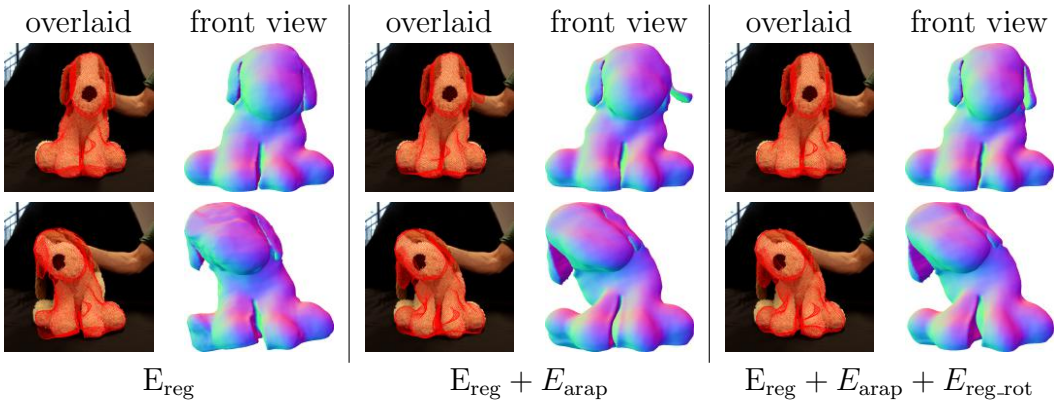


Figure 5.10: Comparison results using different combinations of regularization terms, including spatial regularization E_{reg}, arap E_{arap} and rotation regularization $E_{\text{reg\_rot}}$. From left to right, we show the results of using E_{reg}, $E_{\text{reg}} + E_{\text{arap}}$ and $E_{\text{arap}} + E_{\text{arap}} + E_{\text{reg\_rot}}$ respectively. In comparison, the spatial regularization term E_{reg} alone does not allow large deformations from the template and cannot capture the large articulation movement when the dog turns its head sideways. While the arap term E_{arap} allows large deformations, it provides too much freedom. Notice that the dog's ear bends upwards incorrectly. With the right combination of all three regularization terms, we show that the energy is able to capture large deformations while not allowing too much flexibility so as to induce unnecessary deformations.

Figure 5.11 illustrates the effectiveness of the temporal smoothness term. Due to the ambiguity in the depth direction, it can be seen that when there is no temporal smoothness term, the mesh tends to move forwards and backwards(as shown in the side view images on the left) without changing the 2D projections much. As shown in the overlaid image in the left, although the 2D projection of the mesh fits well to the input image, it is clear from the side view that the 3D mesh has moved substantially along the depth direction. In contrast, as shown in the right figure, when adding temporal smoothness, i.e. penalising the movement in the depth direction, the mesh tends to stay in place and does not jump forwards and backwards.

Quantitative evaluation with the face sequence from Valgaerts *et al.* [128] We evaluate our results quantitatively by taking the publicly available accurate stereo reconstruction results from Valgaerts *et al.* [128] as ground truth 3D shape.

Figure 5.8 shows the comparison between ground truth and tracking results as well as the corresponding error heatmaps for intensity, NCC and SIFT feature data terms. Intensity based tracking has high errors as well as more artefacts around the mouth and eyes regions. While NCC generates

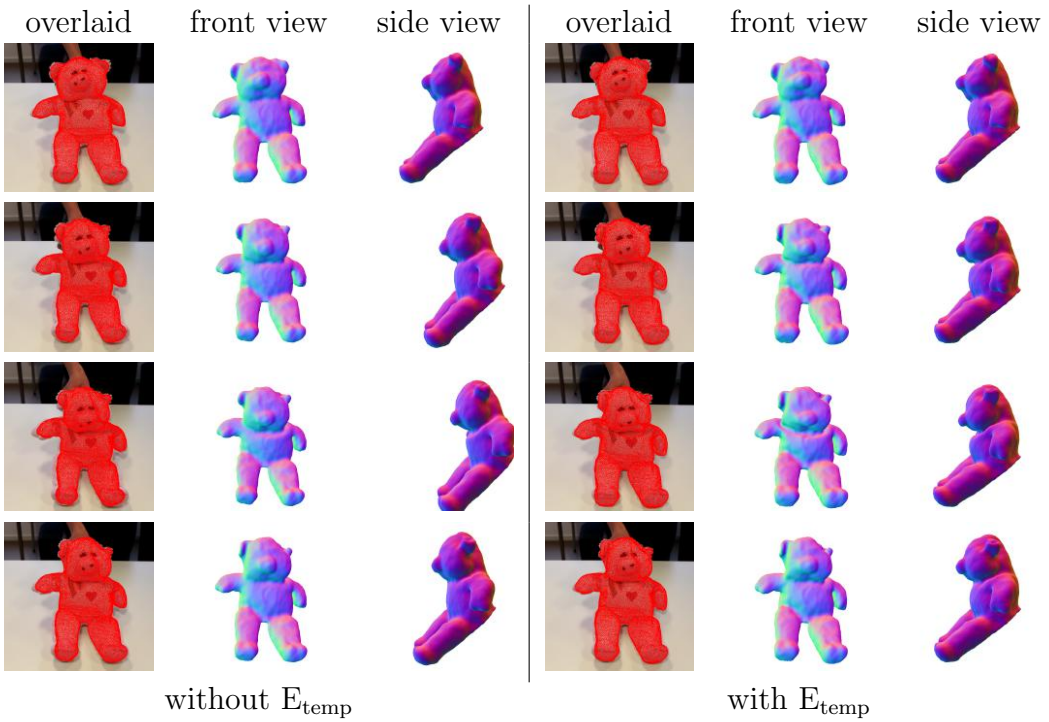


Figure 5.11: Comparison results of with and without temporal smoothness regularization term E_{temp}. **Left:** tracking results without using E_{temp}, including input image overlaid with mesh projections, front view and side view of mesh normals. **Right:** corresponding tracking results with E_{temp}. Due to the ambiguity in the depth direction, it can be seen that when there is no temporal smoothness term, the mesh tends to move forwards and backwards (as shown in the side view images of in the left) without changing the 2D projections much. In contrast, in the right we fix this problem by adding temporal smoothness term, in particular by penalising the movement in the depth direction.

smoother tracking results than using SIFT, it fails to capture the deformation of the mouth and the details of the deformations. Table 5.2 shows the average 3D tracking errors using different data terms compared with the stereo ground truth. It is clear that using more robust data terms, such as NCC or SIFT, can improve the tracking performance. The tracking error decreases from 2.38mm to 2.31mm(NCC) and 2.22mm(SIFT).

Figure 5.9 shows the comparison between frame-to-frame tracking with intensity, NCC and SIFT data terms. Compared to frame-to-model tracking (Figure 5.8), frame-to-frame gives smoother results. However, due to accumulated errors, its performance is worse compared to frame-to-model tracking results. Table 5.2 shows a comparison of 3D tracking errors. We can see that in all three cases, the frame-to-frame tracking error is higher than frame-to-model tracking.

	Intensity	NCC	SIFT
Model based tracking error(mm)	2.38	2.31	2.22
Frame to frame tracking error(mm)	2.84	2.76	2.53

Table 5.2: Average tracking errors using intensity, NCC and SIFT features evaluated on the face2 sequence. All the errors are computed with respect to stereo reconstruction results from Valgaerts *et al.* [128] as ground truth. In all cases, frame-to-frame tracking gives higher error due to accumulated errors.

5.9 Conclusion

In this chapter we have presented a novel approach to template driven capture of dense detailed non-rigid deformations from video sequences. Our method solves simultaneously 2D dense registration problem and 3D shape inference using RGB-video and a pre-acquired template as only input. An additional advantage is that our approach is sequential in nature and can therefore be applied to arbitrarily long sequences. Unlike many other template based methods, our approach can deform complex generic meshes and is not restricted to planar surfaces. We have shown results on real-world novel video sequences captured with a hand-held camera which demonstrate the validity of our approach; we study the use of different data terms with frame-to-frame as well as frame-to-model formulations; and perform a quantitative evaluation against stereo reconstruction results.

Chapter 6

Conclusion

This thesis presents two novel pieces of work on capturing dense deformable shapes from monocular video. In chapter 4, we presented a completely unsupervised solution for reconstructing real-world dynamic scenes. To evaluate the capability of our method, we have shown successful reconstruction results on challenging videos from the *YouTube Objects* dataset and the KITTI dataset [49]. In chapter 5, we presented a template based non-rigid reconstruction approach for tracking dense, generic and complex deforming meshes. We first compute a dense 3D template shape of the object, using a short rigid sequence, and subsequently perform online reconstruction of the non-rigid mesh as it deforms over time.

In this chapter, we conclude the thesis by summarizing the contributions of our work, discussing the limitations and pointing out possible directions for future work.

6.1 Dynamic Scene Reconstruction

In chapter 4, we offered a solution to the problem of scene reconstruction for real-world dynamic monocular videos that deals seamlessly with the presence of non-rigid, articulated or pure rigid motion. In an entirely unsupervised approach, we reorganise/segment the scene into a constellation of object parts, recognise which parts are likely to constitute objects, join them together, and reconstruct the scene. We offered solutions to some of the problems of previous approaches to dynamic scene reconstruction: *(i)* Our approach is able to adapt the topology of the neighbourhood graph by breaking edges where necessary to preserve boundaries between objects. In this way our approach can deal with an entire scene where objects might occlude one another and not just pre-segmented objects; *(ii)* Our work results in a hierarchical approach to dynamic

scene analysis. At the higher level of the hierarchy the scene is explained as a set of objects that are detached from the background and from each other. At the lower level of the hierarchy, each object can be explained as a set of overlapping parts that can model more complex motion.

However, our work has several limitations. First of all, our method is an indirect and pipeline approach, i.e. it takes long-term point trajectories as input, performs motion segmentation first and then uses a piecewise rigid approach to reconstruct the scene. One of the main limitation of this approach is that our reconstruction heavily relies on the motion segmentation step for reconstructing separated objects. Although we have shown that our approach works on several challenging videos, the object segmentation step will fail in more complex scenes when there are multiple moving objects and the boundary connections between these objects become tricky. The second limitation is that the reconstruction step is not stable as rigid parts of a deformable object are usually small and the reconstruction of small part is often degenerate and not well behaved. Either the reconstruction of part will fail or the quality is so bad that successful stitching is impossible because of the ambiguities.

To improve our system, one may try to use dense trajectories or use a direct approach and perform simultaneously segmentation and reconstruction. Here, we discuss some future directions for extending this work.

For object segmentation, we think exploring semantic information would be very useful as it provides complementary cues for what an object is. Recently, deep learning has proven extremely successful in instance segmentation [93, 94, 53]. If we could incorporate these semantic information into our segmentation framework, our object segmentation will benefit from semantic cues and therefore will be much more robust.

For the reconstruction step, we suggest that, to better deal with the reconstruction ambiguities, 3D reconstruction of each deforming object should be formulated as a global optimization problem, i.e. all parts of the object are optimized simultaneously, with overlapping constraints enforced so that rigid parts are consistent with each other. Or even better, we could formulate the whole scene reconstruction problem as a single global optimization, as proposed in recent work [98]. We could reconstruct the whole scene all at once, i.e. reconstruct all the objects within it and at the same time make sure the scales between objects are consistent. In other words, the scales of objects have to satisfy the usual assumptions of real-world scenes, e.g. objects are supported by the static environment and therefore they occlude the background.

6.2 RGB-only Monocular Non-rigid Tracking

In chapter 5, we introduced the first RGB-only, template-based, monocular, dense and direct approach to non-rigid reconstruction that is sequential and suitable for generic shapes. In particular, we first build a template shape of the object via a rigid subsequence and then start tracking the non-rigid motion of the template mesh. An apparent limitation of this approach is that it needs to build a rigid template for every sequence, which is an unnecessary requirement because ideally we would like to build the template shape for the same object only once. If the object starts in the same configuration (in the same shape but might have a different pose), we could register the template shape to the first frame of the non-rigid sequence and start tracking onwards.

We studied the use of robust data terms in our energy minimization framework, such as using SIFT feature images or NCC terms rather than RGB images for tracking. We found that using robust features can overcome some of the limitations of brightness constant assumption and generates better results than using RGB values.

Although our tracking system is not real-time, it is relatively fast and takes about 2s for a mesh with about $20k$ vertices. As current implementation is based on Ceres Solver [4], which only supports CPU at the moment, we expect the same system to run in real time when ported to GPU.

Another interesting future direction is to extend the current system to the template-free case, i.e. without requiring a rigid template to start with. Similar to DynamicFusion [86], the system should be able to update the reference mesh on the fly. However, this is much more challenging than DynamicFusion, as we would need to infer the 3D information of new scenes purely from 2D image data, without any depth input. This is an exciting research direction and remains completely open.

6.3 Future Directions: Deep Learning Based Dynamic Reconstruction

Although this thesis has focused exclusively on optimization-based approaches to the 3D reconstruction of dynamic scenes from video, the recent prevalence of deep learning approaches for solving 3D computer vision problems cannot be ignored. Despite that the use of deep learning techniques for dynamic scene reconstruction is still at its infancy, in this section we explore possible avenues that could be explored in the near future to take advantage of deep learning

techniques in the context of deformable 3D shape reconstruction.

Although traditional geometric computer vision approaches have been very successful [52], they are difficult to integrate with data-driven priors. Unlike rigid reconstruction, these priors are essential for non-rigid reconstruction, as non-rigid problem is usually highly ambiguous and requires powerful priors to overcome ambiguities in order to guarantee a stable solution.

In contrast, deep learning techniques [69, 112, 54] have proven very powerful in learning useful 3D priors from large datasets. Recent work from Ummenhofer *et al.* [127] has shown that we could learn rigid structure from motion using end-to-end deep learning techniques. They showed that this learning based approach outperforms well-established classic SfM methods. The main advantage of this approach is that it provides a principal way to combine geometry with data-driven priors. One interesting example they showed in the paper is that with small baseline images, or even two identical images as input, their network still generates reasonable result, while traditional SfM will become degenerate. This clearly shows that their network has not only learnt two frame SfM algorithm, but also learnt to extract powerful priors from the appearance.

However, deep learning, usually requires large datasets for training. Although deep learning has been very successful in many computer vision applications, such as, image recognition, object segmentation, object detection [69, 112, 54], the success in 3D reconstruction is not as significant so far. The main reason is that 3D data is much more expensive and there has not been as many large datasets as in 2D areas. To this end, the work in this thesis, unsupervised learning of 3D shape from monocular video, could be helpful in at least two ways. First, we could first use the method proposed in the thesis to perform 3D reconstruction and then use the obtained results as proxy ground truth to train deep networks. Second, the proposed formulation will still be relevant when we design neural network architecture and loss function for non-rigid 3D reconstruction problems. The knowledge about 3D geometric vision should be embedded into the architecture and be explored to make the learning process more efficient, requiring less data.

One trend of current research direction is about how we could explore our domain knowledge to help reduce the amount of data needed for training. In particular, recent work has shown promising directions by exploiting domain knowledge to design architectures that are weakly-supervised or self-supervised. For example, Garg *et al.* [44] and Godard *et al.* [50] have proposed methods to learn depth map from a single image without relying on any depth ground truth.

More recently, Zhou *et al.* [143] proposed an unsupervised approach for learning depth map and ego-motion from monocular video. We believe that combining deep learning and traditional geometric approaches is a promising direction and we would need much less labelled data to train deep neural networks.

For instance, the work from Ummenhofer *et al.* [127] could be extended to dynamic scenes, i.e. given two consecutive monocular images of dynamic scenes we could estimate the depth map of the first image and its dynamic motion. This two-frame result can then be further extended to multiple frames, or the framework could be reformulated to perform online reconstruction. Another interesting direction is fusion based reconstruction and tracking on top of the output of the network, namely the depth map and dynamic motion.

Appendix A

Full List of Publications

Rui Yu has published the following conference papers during his Ph.D.

1) Chris Russell*, Rui Yu* and Lourdes Agapito. Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes. In *ECCV*, 2014. **(Chapter 4)**

2) Rui Yu, Chris Russell, Neill D. F. Campbell and Lourdes Agapito. Direct, Dense, and Deformable: Template-Based Non-Rigid 3D Reconstruction from RGB Video. In *ICCV*, 2015. **(Chapter 5)**

3) Rui Yu, Chris Russell and Lourdes Agapito. Solving Jigsaw Puzzles with Linear Programming. In *BMVC*, 2016.

4) Qi Liu-Yin, Rui Yu, Andrew Fitzgibbon, Lourdes Agapito and Chris Russell. Better Together: Joint Reasoning for Non-rigid 3D Reconstruction with Specularities and Shading. In *BMVC*, 2016.

"*" means joint first authorship with equal contribution to the paper.

Bibliography

[1] Kinect. `https://en.wikipedia.org/wiki/Kinect`. Accessed: 2016-12-16. 16

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 84

[3] A. Adams, J. Baek, and A. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Eurographics*, 2010. 80

[4] S. Agarwal, K. Mierle, et al. Ceres solver. `http://ceres-solver.org`. 99, 113

[5] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005. 62

[6] A. Bartoli, Y. Gerard, F. Chadebecq, and T. Collins. On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In *CVPR*, 2012. 91

[7] A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-template. *IEEE transactions on pattern analysis and machine intelligence*, 37(10):2099–2118, 2015. 58

[8] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006. 103

[9] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999. 60

[10] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR*, 2010. 65

[11] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. *arXiv preprint arXiv:1607.08128*, 2016. 62

[12] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, page 155 – 225, 2002. 73

[13] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *PAMI*, 26(9):1124–1137, 2004. 73

[14] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. 23, 2001. 32, 34, 73

[15] M. Brand. Morphable models from video. In *CVPR*, 2001. 55, 57

[16] M. Brand. A direct method for 3D factorization of nonrigid motion observed in 2D. In *CVPR*, 2005. 55, 57

[17] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR*, June 2000. 53

[18] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010. 51, 75, 77, 80, 81

[19] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, 2008. 93, 94

[20] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Automatic object segmentation from calibrated images. In *CVMP*, 2011. 94

[21] C. Cao, Q. Hou, and K. Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on Graphics (TOG)*, 33(4):43, 2014. 61

[22] T.-J. Chin, H. Wang, and D. Suter. Robust fitting of multiple structures: The statistical learning approach. In *2009 IEEE 12th International Conference on Computer Vision*, pages 413–420. IEEE, 2009. 28

[23] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. 5th European Conference on Computer Vision, Freiburg, Germany*, volume 2, pages 484–498, 1998. 60

[24] T. F. Cootes and C. J. Taylor. Active shape models. In *Proc. British Machine Vision Conference*, pages 265–275, 1992. 60

[25] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *ICCV*, pages 1071–1076, 1995. 50, 51

[26] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 392–407. Springer, 2000. 58

[27] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *arXiv preprint arXiv:1604.01093*, 2016. 13

[28] Y. Dai, H. Li, and M. He. A simple prior-free method for non rigid structure from motion factorization. In *CVPR*, 2012. 55, 57, 90, 93

[29] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, 2003. 13

[30] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007. 42

[31] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *SIGGRAPH*, 2008. 13

[32] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *CVPR*, 2010. 31

[33] 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)*, 35(4):114, 2016. 13, 15, 90, 91, 92, 98

[34] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Interpreting face images using active appearance models. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, pages 300–306, 1998. 60

[35] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, 2009. 28, 51

[36] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *arXiv preprint arXiv:1607.02565*, 2016. 42

[37] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer International Publishing, 2014. 13, 14, 42

[38] J. Fayad, L. Agapito, and A. Del Bue. Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *European Conference on Computer Vision ECCV 2010, Crete, Greece*, volume 6314 of *Lecture Notes in Computer Science*, pages 297–310. Springer Berlin / Heidelberg, September 2010. 78

[39] J. Fayad, C. Russell, and L. Agapito. Automated articulated structure and 3d shape recovery from point correspondences. In *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011. 56, 57, 64, 65, 66, 67, 68, 69, 71, 80, 81

[40] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In M. A. Fischler and O. Firschein, editors, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, pages 726–740. Los Altos, CA., 1987. 28

[41] A. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3-d reconstruction of independently moving objects. In *European Conference on Computer Vision (ECCV)*, Dublin, Ireland, 2000. 50

[42] S. Fuhrmann, J. Ackermann, T. Kalbe, and M. Goesele. Direct resampling for isotropic surface remeshing. In *Proceedings of Vision, Modeling and Visualization 2010, Siegen, Germany*, 2010. 94

[43] R. Garg. *Dense Motion Capture of Deformable Surfaces from Monocular Video*. PhD thesis, Queen Mary University of London, 2013. 54

[44] R. Garg and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. *arXiv preprint arXiv:1603.04992*, 2016. 114

[45] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *CVPR*, 2013. 15, 18, 42, 54, 55, 57, 63, 89, 90, 91, 92, 93

[46] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *International Journal of Computer Vision (IJCV)*, 2013. 54

[47] P. Gargallo, Y. Kuang, et al. Opensfm. `https://github.com/mapillary/OpenSfM`. 86

[48] P. Garrido, L. Valgaerts, C. Wu, and C. Theobalt. Reconstructing detailed dynamic face geometry from monocular video. In *SIGGRAPH Asia*, 2013. 91

[49] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013. 80, 84, 85, 111

[50] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *arXiv preprint arXiv:1609.03677*, 2016. 114

[51] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 13, 14, 69, 71

[52] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 114

[53] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 112

[54] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 114

[55] C. Hernández, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *CVPR*, 2007. 94

[56] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. *arXiv preprint arXiv:1603.08161*, 2016. 13

[57] M. Irani and P. Anandan. About direct methods. In *International Workshop on Vision Algorithms*, pages 267–277. Springer Berlin Heidelberg, 1999. 42, 43

[58] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision (IJCV)*, 97(2), 2012. 9, 28, 29, 30, 52, 65, 71

[59] H. N. Isack. Energy based multi-model fitting and matching problems. 2014. 30

[60] P. Ji, M. Salzmann, and H. Li. Shape interaction matrix revisited and robustified: Efficient subspace clustering with corrupted and incomplete data. In *ICCV 2015*, 2015. 51

[61] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. doi:10.5244/C.24.12. 62

[62] Y. Kanazawa and H. Kawakami. Detection of planar regions with uncalibrated stereo using distributions of feature points. In *BMVC*, pages 1–10. Citeseer, 2004. 28

[63] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, 2007. 13, 42

[64] P. Kohli, M. P. Kumar, and P. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007. 34

[65] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008. 32

[66] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008. 74

[67] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009. 35

[68] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2):147–159, 2004. 33

[69] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 114

[70] S. Kumar, Y. Dai, and H. Li. Multi-body non-rigid structure-from-motion. *arXiv preprint arXiv:1607.04515*, 2016. 57

[71] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Graph-cut based inference with co-occurrence statistics. In *ECCV*, 2010. 39, 75

[72] L. Ladicky, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clockson, and P. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. *BMVC*, 2010. 32

[73] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *SIGGRAPH Asia*, 2009. 13

[74] Z. Li, J. Guo, L.-F. Cheong, and Z. Zhou. Perspective motion segmentation via collaborative clustering. In *ICCV*, 2013. 51

[75] 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)*, 34(6):248, 2015. 62

[76] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 103

[77] C. Malleson, M. Klaudiny, A. Hilton, and J.-Y. Guillemaut. Single-view rgb-d based reconstruction of dynamic human geometry. In *4DMOD Workshop at ICCV*, 2013. 13

[78] A. Malti, A. Bartoli, and T. Collins. A pixel-based approach to template-based monocular 3d reconstruction of deformable surfaces. In *4DMOD Workshop at ICCV*, 2011. 59, 89, 90, 91, 92

[79] A. Malti, R. Hartley, A. Bartoli, and J.-H. Kim. Monocular template-based 3d reconstruction of extensible surfaces with local linear elasticity. In *CVPR*, 2013. 91

[80] M. Marques and J. P. Costeira. Estimating 3D shape from degenerate sequences with missing data. *CVIU*, 2008. 78

[81] D. Marr. Vision: A computational investigation into the human representation and processing of visual information. 1982. 15

[82] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004. 60

[83] I. Matthews, J. Xiao, and S. Baker. 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. *International journal of computer vision*, 75(1):93–113, 2007. 61

[84] R. Mur-Artal, J. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 13, 42

[85] M. Narasimhan and J. A. Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. *arXiv preprint arXiv:1207.1404*, 2012. 74

[86] R. Newcombe, D. Fox, and S. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015. 13, 42, 90, 91, 92, 98, 113

[87] R. Newcombe, S. Lovegrove, and A. Davison. DTAM: Dense Tracking and Mapping in Real-Time. In *ICCV*, 2011. 13, 42

[88] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 13, 42

[89] P. Ondrúška, P. Kohli, and S. Izadi. Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones. *IEEE transactions on visualization and computer graphics*, 21(11):1251–1258, 2015. 13

[90] J. Östlund, A. Varol, D. T. Ngo, and P. Fua. Laplacian meshes for monocular 3d shape recovery. In *Computer Vision–ECCV 2012*, pages 412–425. Springer, 2012. 91

[91] K. Ozden, K. Schindler, and L. van Gool. Multibody structure-from-motion in practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2010. 14, 50, 63

[92] M. Paladini, A. Del Bue, J. Xavier, L. Agapito, M. Stosic, and M. Dodig. Optimal metric projections for deformable and articulated structure-from-motion. *IJCV*, 2012. 66, 90, 93

[93] P. H. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. *CoRR*, abs/1506.06204, 2015. 112

[94] P. H. O. Pinheiro, T. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. *CoRR*, abs/1603.08695, 2016. 112

[95] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. *arXiv preprint arXiv:1511.06645*, 2015. 62

[96] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors fromweakly annotated video. In *CVPR*, 2012. 19, 64, 77, 81, 83

[97] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 14, 84, 85, 86

[98] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 52, 58, 112

[99] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(10):1832–1845, 2010. 51

[100] A. Roussos, C. Russell, R. Garg, and L. Agapito. Dense multibody motion estimation and reconstruction from a handheld camera. In *ISMAR*, 2012. 14, 63, 65

[101] A. Roussos, C. Russell, R. Garg, and L. Agapito. Dense multibody motion estimation and reconstruction from a handheld camera. In *ISMAR*, 2012. 57

[102] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011. 103

[103] C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *CVPR*, 2011. 39, 40, 41, 56, 57, 64, 65, 66, 67, 68, 70, 71, 72, 73, 74, 80

[104] C. Russell, J. Fayad, and L. Agapito. Dense non-rigid structure from motion. *3dimPVT*, 2012. 18

[105] C. Russell, J. Fayad, and L. Agapito. Dense non-rigid structure from motion. In *3DIMPVT*, October 2012. 57

[106] M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. In *CVPR*, 2009. 58

[107] M. Salzmann and P. Fua. Deformable surface 3d reconstruction from monocular images. *Synthesis Lectures on Computer Vision*, 2(1):1–113, 2010. 58

[108] M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3-d tracking. In *CVPR*, 2007. 58, 91

[109] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-Form Solution to Non-Rigid 3D Surface Registration. In *ECCV*, 2008. 58, 91

[110] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3d shape recovery. In *CVPR*, 2008. 91

[111] K. Schindler, D. Suter, and H. Wang. A model selection framework for multibody structure-and-motion of image sequences. *International Journal of Computer Vision (IJCV)*, 79(2):159–177, 2008. 50

[112] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 114

[113] P. Siva, C. Russell, T. Xiang, and L. Agapito. Looking beyond the image: Unsupervised learning for object saliency and detection. In *CVPR*, 2013. 68, 70

[114] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press. 13, 14, 42

[115] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, 2007. 91, 98

[116] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:284–294, June 2006. 80

[117] R. W. Sumner, J. Schmid., and M. Pauly. Embedded deformation for shape manipulation. In *SIGGRAPH*, 2007. 99

[118] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 65

[119] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. M. Seitz. Total moving face reconstruction. In *ECCV*, 2014. 15, 59, 89, 90, 91, 92

[120] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *CVPR*, 2010. 56

[121] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. 16, 61

[122] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *IJCV*, 1992. 25

[123] P. H. S. Torr. Geometric motion segmentation and model selection [and discussion]. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 356(1740):pp. 1321–1340, 1998. 28

[124] P. H. S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In *International workshop on vision algorithms*, pages 278–294. Springer Berlin Heidelberg, 1999. 42, 43

[125] L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 2008. 66, 90, 93

[126] B. Triggs, A. Zisserman, R. Szeliski, H. Sawhney, S. Peleg, M. Irani, P. Torr, J. Knight, J. Malik, and A. P. Discussion for direct versus features session, 2000. 43

[127] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. DeMoN: Depth and Motion Network for Learning Monocular Stereo. *arXiv:1612.02401 [cs]*, December 2016. 114, 115

[128] L. Valgaerts, A. Bruhn, H.-P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *SIGGRAPH*, 2013. 13, 104, 107, 109

[129] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *ICCV*, 2009. 55, 56, 63, 64

[130] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472. ACM, 2010. 103

[131] R. Vidal. Subspace clustering. *Signal Processing Magazine, IEEE*, 28(2):52–68, 2011. 51

[132] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). In *CVPR*, pages 621–628, 2003. 51

[133] G. Vogiatzis, C. Hernández, P. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. *PAMI*, 2007. 94

[134] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 28

[135] T. Y. Wang, P. Kohli, and N. J. Mitra. Dynamic sfm: Detecting scene changes from image pairs. In *Computer Graphics Forum*, volume 34, pages 177–189, 2015. 57

[136] C. Wu. Visualsfm: A visual structure from motion system. `http://ccwu.me/vsfm/`, 2011. 94

[137] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. *IJCV*, 2006. 55, 57

[138] J. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):865–877, 2008. 56

[139] A. L. Yuille and A. Rangarajan. The concave-convex procedure (cccp). *NIPS*, 2002. 74

[140] C. Zach. Robust bundle adjustment revisited. In *European Conference on Computer Vision*, pages 772–787. Springer International Publishing, 2014. 44

[141] G. Zhang, J. Jia, and H. Bao. Simultaneous multi-body stereo and segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, November 2011. 85, 86, 88

[142] W. Zhang and J. Ksecká. Nonparametric estimation of multiple structures with outliers. In *Dynamical Vision*, pages 60–74. Springer, 2007. 28

[143] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. *arXiv preprint arXiv:1704.07813*, 2017. 115

[144] M. Zollhofer, M. Niessner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger. Real-time non-rigid reconstruction using an rgb-d camera. *SIGGRAPH*, 2014. 13, 15, 42, 90, 91, 92, 98