DATA-DRIVEN 3D RECONSTRUCTION AND VIEW SYNTHESIS OF DYNAMIC SCENE ELEMENTS

Dinghuang Ji

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2017

Approved by:

Jan-Michael Frahm

Enrique Dunn

Tamara L. Berg

Marc Niethammer

Silvio Savarese

ABSTRACT

Dinghuang Ji: Data-driven 3D Reconstruction and View Synthesis of Dynamic Scene Elements
(Under the direction of Jan-Michael Frahm and Enrique Dunn)

Our world is filled with living beings and other dynamic elements. It is important to record dynamic things and events for the sake of education, archeology, and culture inheritance. From vintage to modern times, people have recorded dynamic scene elements in different ways, from sequences of cave paintings to frames of motion pictures. This thesis focuses on two key computer vision techniques by which dynamic element representation moves beyond video capture: towards 3D reconstruction and view synthesis. Although previous methods on these two aspects have been adopted to model and represent static scene elements, dynamic scene elements present unique and difficult challenges for the tasks.

This thesis focuses on three types of dynamic scene elements, namely 1) dynamic texture with static shape, 2) dynamic shapes with static texture, and 3) dynamic illumination of static scenes. Two research aspects will be explored to represent and visualize them: dynamic 3D reconstruction and dynamic view synthesis. Dynamic 3D reconstruction aims to recover the 3D geometry of dynamic objects and, by modeling the objects' movements, bring 3D reconstructions to life. Dynamic view synthesis, on the other hand, summarizes or predicts the dynamic appearance change of dynamic objects – for example, the daytime-to-nighttime illumination of a building or the future movements of a rigid body.

We first target the problem of reconstructing dynamic textures of objects that have (approximately) fixed 3D shape but time-varying appearance. Examples of such objects include waterfalls, fountains, and electronic billboards. Since the appearance of dynamic-textured objects can be random and complicated, estimating the 3D geometry of these objects from 2D images/video requires novel tools beyond the appearance-based point correspondence methods of traditional 3D

computer vision. To perform this 3D reconstruction, we introduce a method that simultaneously 1) segments dynamically textured scene objects in the input images and 2) reconstructs the 3D geometry of the entire scene, assuming a static 3D shape for the dynamically textured objects.

Compared to dynamic textures, the appearance change of dynamic shapes is due to physically defined motions like rigid body movements. In these cases, assumptions can be made about the object's motion constraints in order to identify corresponding points on the object at different timepoints. For example, two points on a rigid object have constant distance between them in the 3D space, no matter how the object moves. Based on this assumption of local rigidity, we propose a robust method to correctly identify point correspondences of two images viewing the same moving object from different viewpoints and at different times. Dense 3D geometry could be obtained from the computed point correspondences. We apply this method on unsynchronized video streams, and observe that the number of inlier correspondences found by this method can be used as indicator for frame alignment among the different streams.

To model dynamic scene appearance caused by illumination changes, we propose a framework to find a sequence of images that have similar geometric composition as a single reference image and also show a smooth transition in illumination throughout the day. These images could be registered to visualize patterns of illumination change from a single viewpoint.

The final topic of this thesis involves predicting the movements of dynamic shapes in the image domain. Towards this end, we propose deep neural network architectures to predict future views of dynamic motions, such as rigid body movements and flowers blooming. Instead of predicting image pixels from the network, my methods predict pixel offsets and iteratively synthesize future views.

ACKNOWLEDGEMENTS

My deepest gratitude is to my advisors Jan-Michael Frahm and Enrique Dunn, for the discussions and guidence they gave in countless meetings, for the patience and encouragement when I encounter failures, and for the endless support in my program study and job hunting.

I would also like to thank my committee members, Tamara L. Berg, Marc Niethammer, and Silvio Savarese, for their feedback and advice.

Additionally, I would like to thank my labmates, as their company and discussion made my time more fruitful and enjoyable:

Philip Ammirato, Akash Bapat, Sangwoo Cho, Marc Eder, Pierre Fite-Georgel, Yunchao Gong, Rohit Gupta, Shubham Gupta, Xufeng Han, Jared Heinly, Junpyo Hong, Hadi Kiapour, Hyo Jin Kim, Wei Liu, Licheng Yu, Vicente Ordóñez-Román, David Perra, True Price, Rahul Raguram, Patrick Reynolds, Johannes Schönberger, Meng Tan, Joseph Tighe, Sirion Vittayakorn, Ke Wang, Yilin Wang, Zhen Wei, Yi Xu, Hongsheng Yang, and Enliang Zheng.

I deeply appreciate my mentors during internships, for their kind support and supervision: Shiyu Song, Yong-Dian Jian, and Junghyun Kwon.

I want to thank my friends met in Chapel Hill, who really made me enjoy the life and taught me how to become a better person: Yi Hong, Wentao Li, Michael Wilson, Hongxiang Long, Jun Jiang, Qiuyu Xiao, Yu Meng, Yipin Zhou, Jingbo Wang, Wenhua Guan, Ye Zhao, Ruoyu Wu, Xiaoming Liu, Dawei Tang, Cheng Cao, Yueting Luo, Xiaodan Wang, Qingning Zhou, Qixin Wu, Wenshuai Li, Si Chen, and Mark Vance. I especially appreciate Yirong Jia, Baocai Cheng, Yu Cheng and James Wang for their long lasting friendship and supports.

Last but not least, I would like to thank my beloved parents Ying Chen and Liangui Ji, who encouraged me to learn, explore and create for my PhD study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

MRF	Markov Random Field
NCC	Normalized cross correlation
NRSFM	Non-rigid structure from motion
RANSAC	Random sample consensus
SfM	Structure from motion
SIFT	Scale invariant feature transform (Lowe, 2004)

CHAPTER 1: INTRODUCTION

Our living environment is vibrant and intriguing because of dynamism. At a large scale, the Earth itself rotates in the space, creating days and nights in a perpetual flow. At scales visible to humans, we see water flowing down from the top of mountains, a juggler performing in the market, and billboards flashing at night. The world is constantly transient, persistent only in small moments, and we watch our seemingly static selves and creations change with time.

Visual perception is an important component in the human perception of time, and visual representations are therefore one of the most effective ways to record and convey the dynamic world we inhabit. In ancient times, people used carvings or paintings to express what they saw; art and history have always been deeply intertwined. In the last two centuries, film and video introduced increasingly authentic means for documenting the human experience. In the present age, the wide-spread availability of digital cameras, along with the existence of social media websites like Facebook, Snapchat, and Twitter, has enabled ubiquitous capture and sharing of the visual world. The growth of imagery in the Internet Age is overwhelmingly prolific. And here, the excitement of dynamism rises again: As we approach a near-constant, near-global capture of our world, how can we best represent the moments in which we live and the experiences we have today? Answering this question is a major sub-focus in the vast, ever-expanding field of computer vision.

Today, technologies like 3D reconstruction and view synthesis have been adopted to generate more concise visual representations of large-scale 2D visual data. For example, many individual images from Google StreetView cameras are used to create long tracks of 360° panoramas (Zhang and Liu, 2014); aerial images are often used to create large-scale 2D and 3D maps (Wang et al., 2016a,b); ground-level photographs from many individuals have been leveraged to create 3D reconstructions of highly-photographed landmarks (Frahm et al., 2010; Agarwal et al., 2012; Heinly

et al., 2015); and high-quality motion capture applications rely on many synchronized video streams (Kanade et al., 1997; Joo et al., 2015). These creations, in turn, are widely adapted in applications like virtual tourism, virtual reality, autonomous driving, and special effects for movies.

However, the authentic representation of dynamic elements in our 3D world has yet to be completely attained, especially for the uncontrolled capture scenarios found in Internet-scale data. To bridge the gap in dynamic object representation from uncontrolled imagery, we explore research in two aspects: dynamic 3D reconstruction and dynamic view synthesis of scene elements.

Regarding 3D reconstruction, state-of-the-art crowd-sourced 3D reconstruction systems employ structure from motion (SfM) techniques that leverage large-scale imaging redundancy in order to generate photo-realistic models of scenes of interest. SfM (Frahm et al., 2010; Snavely et al., 2006; Zheng and Wu, 2015; Schönberger and Frahm, 2016; Heinly et al., 2015) is the process by which the 3D geometry (structure) of a scene is recovered via a set of images taken from different viewpoints (which constitute camera motion). The estimated 3D models reliably depict both the shape and appearance of the captured environment under the joint assumptions of shape constancy and appearance congruency, both of which are commonly associated with static structures. Accordingly, the resulting 3D models are unable to robustly capture dynamic scene elements not in compliance with the aforementioned assumptions. Applying SfM to dynamic objects requires two methodological considerations: how to determine correspondences between images given that the object's appearance may have changed, and how to model changes in the object's geometry, position, or pose.

In a dynamic reconstruction framework, dynamic scene elements can be determined through the observation of visual motion. Nelson and Polana (Nelson and Polana, 1992) categorized visual motion into three classes: activities, motion events, and dynamic (temporal) texture change. *Activities*, such as walking or juggling, are defined as motion patterns that are periodic in time; *motion events*, like opening a door, lack explicit temporal or spatial periodicity; *dynamic textures*, such as flowing water, exhibit statistical regularity but have uncertain spatial and temporal extent. Dynamic scenes may contain visual motions in any combination of these three categories. Another

criterion to categorize dynamic objects is whether the shapes or texture of the objects change, which classifies them into shape-deforming objects (Dynamic Shapes) and shape-constant objects with temporal appearance change (Dynamic Textures).

To reconstruct the 3D shape of dynamic textures, the geometry of those scene elements having time-varying appearance (e.g., active billboards, bodies of water, or building facades under varying illumination conditions) can be approximated by a single surface; in this thesis, a completely data-driven method that does not impose geometric or shape priors is proposed.

For objects having time-varying shape, several methods (Jiang et al., 2012; Joo et al., 2014, 2015; Mustafa et al., 2015; Zheng et al., 2015a, 2017; Russell et al., 2014; Garg et al., 2013) have been introduced over the last few years, with most of them assuming multi-view synchronized video sequences within controlled environments. Reconstruction with unsynchronized data captured in general scenes, such as multiple individuals filming a concert, is an unsolved problem that I tackle in this thesis. Taking as input a pair of unsynchronized video streams of the same dynamic scene, the method outputs a dense point cloud corresponding to the evolving shape of the commonly observed dynamic foreground. In addition to the 3D structures, the method estimates the temporal offset of the input pair of video streams, assuming a known frame-rate ratio between them.

Another general topic of this thesis involves dynamic view synthesis, wherein unseen novel views of a dynamic object are generated based on a set of available existing views. View synthesis in general has many appealing applications in computer vision and graphics, such as creating a continuous 2D viewpoint space for virtual 3D tours and photo editing with 3D object manipulation capabilities. However, the majority of existing view synthesis methods (Beier and Neely, 1992; Jones and Poggio, 1995; Katayama et al., 1995; Seitz and Dyer, 1996; Tatarchenko et al., 2016; Yang et al., 2015; J. Flynn and Snavely, 2015; Zhou et al., 2016) focus on rigid scene element synthesis and assume constant scene illumination or object geometry. In this thesis, we introduce new approaches for visualizing dynamic changes in illumination in a single view, and for synthesizing the possible future appearance of an object in motion.

First, considering the problem of modeling dynamic scene illumination, Internet photo-collections provide vast samples in the space of possible viewpoints and appearance configurations available for a given scene. Such images could be utilized to visualize the appearance change of a single viewpoint within a given time range. Here, we propose a method for augmenting a static image with the range of scene illuminations found in an Internet photo-collection, in a method combining geometry and appearance information.

Second, image-based motion prediction aims to generate plausible visualizations of the temporal evolution of dynamic scene elements. In addition to view synthesis, this problem is closely related to the problem of motion field estimation. Motion field estimation strives to determine dense pixel correspondences among a pair of image observations of a common scene. Given an input image and a motion field, it is straightforward to synthesize a novel image by simply locally shifting the image according to the 2D field. Conversely, given an input image and a synthesized image, there exists an abundance of methods to estimate the motion field. Inspired by the pioneer work of appearance flow network (Zhou et al., 2016), we propose to implicitly learn motion flow within visual prediction neural networks, which has the potential to generate images with more crisp textures than image prediction approaches.

1.1 Thesis Statement

Representations of the geometry and appearance of dynamic scene elements can be estimated from images and videos captured in uncontrolled settings by reformulating existing imagery content matching and registration frameworks to include: data-driven segmentation for shape correspondence of dynamic textures, local appearance matching in the spatio-temporal domain within unsynchronized videos, and the construction of augmented image representations combining geometry and appearance information for data-driven illumination transfer.

1.2 Outline of Contributions

This dissertation contributes significantly to advance the state-of-the-art techniques for the problems of 3D reconstruction of dynamic objects and data driven dynamic view synthesis, and it builds on our published works (Ji et al., 2014, 2016, 2015; Radenović et al., 2016; Ji et al., 2017).

3D Reconstruction of Dynamic Textures: Chapter 3 aims for a more complete and realistic 3D scene representations by addressing the 3D modeling of dynamic scene elements within the context of crowd-sourced input imagery. The input data to my proposed framework encompasses both online image and video collections capturing a common scene. Sparse reconstruction is first performed for the rigid scene elements. Then, video collection data is analyzed to reap video segments amenable for 1) registration to the existing rigid model and 2) coarse identification of dynamic scene elements.

The proposed method adopts these coarse estimates, along with the knowledge of the sparse rigid 3D structure, to pose the segmentation of dynamic elements within an image as a global two-label optimization problem. The attained dynamic region masks are subsequently fused through shape-from-silhouette techniques in order to generate an initial 3D shape estimate from the input videos. The preliminary 3D shape is then back projected to the original photo-collection imagery, and all image labelings are then recomputed and fused to generate an updated 3D shape. This process is iterated until convergence of the output photo-collection imagery segmentation process.

Dense Dynamic Scene Reconstruction with Unsynchronized Videos: Chapter 4 targets the problem of reconstructing the dense 3D geometry of dynamic objects from given unstructured video sequences. Existing motion capture techniques have typically addressed well-controlled capture scenarios, where aspects such as camera positioning, sensor synchronization, and favorable scene content are either carefully designed *a priori* or controlled online. Whereas multi-camera static scene reconstruction methods leverage photo-consistency across spatially varying observations, their dynamic counterparts must address photoconsistency in

the spatio-temporal domain. In this respect, the main challenges are 1) finding a common temporal reference frame across independent video captures, and 2) meaningfully propagating temporally varying photo-consistency estimates across videos.

In this work, we propose to address both of these challenges by enforcing the geometric consistency of optical flow measurements across spatially registered video segments. Moreover, the proposed approach builds on the thesis that maximally consistent geometry is obtained with minimal temporal alignment error, and vice versa. Towards this end, we posit that it is possible to recover the spatio-temporal overlap of two image sequences by maximizing the set of consistent spatio-temporal correspondences among the two video segments.

Appearance Analysis of Scenes Under Different Illuminations: Chapter 5 strives to address the organization and characterization of the image space by exploring the link between time-lapse photography and crowd-sourced imagery. Time-lapse photography strives to depict the evolution of a given scene as observed under varying image capture conditions. While the aggregation of a sequence of images into a video may be the most straightforward visualization for time-lapse photography, the integration of multiple images in the form of a mosaic provides a descriptive 2D representation of the observed scene's temporal variability. The problem of mosaic construction can be abstracted as a three-stage process of image registration, alignment, and aggregation. However, the representation of the appearance dynamics introduces the qualitative challenge of producing an aggregate mosaic that is both coherent with the original scene content and descriptive of the fine-scale appearance variations across time. We address these challenges by exploring the spectrum of capture variability available in Internet photo-collections and propose a novel framework to obtain illumination mosaics.

Visual predictions: We target the problem of synthesizing future motion sequences from input images. Previous methods tackled the problem in two manners: predicting the future image pixel values and predicting the dense time-space trajectory of pixels. The use of generative

encoder-decoder networks has been widely adopted in both kinds of methods. Pixel prediction via these networks has been shown to suffer from blurry outputs, since images are generated from scratch and there is no explicit enforcement of visual coherency. However, crisp details can be achieved by transferring pixels from the input image through trajectory prediction, but this requires pre-computed motion fields for training. To synthesize realistic movement of objects under weaker supervision, we propose a novel network structure, inspired by appearance flow networks (Zhou et al., 2016). Motion priors (sparse joint positions of rigid body movements) are further incorporated to enable more efficient appearance synthesis.

Following Chapter 2, which covers related works, the next four chapters describe each method in detail, and Chapter 7 concludes the dissertation with potential extensions to the works and possible future research directions.

CHAPTER 2: RELATED WORK

Related to the problem of dynamic 3D reconstruction and dynamic view synthesis, many approaches have been proposed to address issues relating to them. This section outlines several related efforts in each of these areas.

2.0.1 3D Reconstruction of Dynamic Objects

For static environments, very robust SfM systems (Agarwal et al., 2012; Heinly et al., 2015; Wu, 2013) and multi-view stereo (MVS) approaches (Furukawa and Ponce, 2010; Schönberger et al., 2016; Zheng et al., 2014) have shown much success in recovering scene geometry with high accuracy on a large variety of datasets. Modeling non-static objects with those frameworks, however, is considerably more difficult because the assumptions driving correspondence detection and 3D point triangulation in rigid scenarios cannot be directly applied to moving objects. To address these challenges, a wide array of dynamic scene reconstruction techniques have been introduced in the computer vision literature, in capture situations that are controlled or uncontrolled, synchronized or unsynchronized, single-view or multi-view, and model-based or model-free.

In general, highly controlled image capture scenarios have shown considerable success for non-static scene capture because they are able to leverage more powerful assumptions with respect to appearance and correspondence of scene elements. For example, Joo *et al*. (Joo et al., 2014, 2015) used a large-scale rig of 480 synchronized cameras arranged along a sphere to obtain high-quality, dense reconstructions of moving objects within the capture environment. This system strategically selects views that have good visibility and bypass the occlusion issue. However, it takes substantial efforts to setup the system e.g. camera synchronization, data storage and cable arrangements. For more general applications, Kim *et al*. (Kim et al., 2010) designed a synchronized, portable,

multi-camera system specifically tailored for dynamic object capture. These works, and others (Martin and Daniel, 2013; Oswald et al., 2014; Djelouah et al., 2015; Letouzey and Boyer, 2012; Wu et al., 2011; Guan et al., 2010; Cagniart et al., 2010), clearly indicate the strong potential for non-rigid reconstruction in controlled capture scenarios, and they highlight in particular the usefulness of multiple synchronized video streams toward this end.

3D reconstruction of dynamic scenes in uncontrolled environment is a challenging problem for computer vision research. Several systems have been developed for building multiview dynamic outdoor scenes. Jiang et al. (Jiang et al., 2012) and Taneja et. al. (Taneja et al., 2010) propose probabilistic frameworks to model outdoor scenes with handheld synchronized cameras. By incorporating depth consistency within depth maps and images, these frameworks could obtain smooth depth maps and 3D surfaces. Pollefeys et al. (Pollefeys et al., 2007) built a large scale 3D reconstruction system that combines GPS and inertial info with videos to generate a 3D mesh in urban scenes. Again, these systems all rely on a set of pre-calibrated or synchronized cameras. In this thesis, we propose two frameworks to recover 3D geometry of dynamic scene elements captured in uncontrolled environments, with Internet downloaded imagery which extensively vary in environment and camera parameters and hand-held unsynchronized video streams respectively.

Despite the large amount of crowd-sourced video data available on the Internet (for example, multiple video uploads from a live concert), only a few research works have focused on general dynamic 3D reconstruction from unsynchronized, non-concurrent capture. Zheng *et al*. (Zheng et al., 2015a) recently propose a solution to this interesting problem. The authors introduced a dictionary learning method to simultaneously solve the problem of video synchronization and sparse 3D reconstruction. In this method, the frame offsets of multiple videos are obtained by sparse representation of the triangulated 3D shapes, and the shapes are iteratively refined with updated sequencing information. However, this approach is not automatic, relying heavily on manually labeled correspondences on the rigid bodies, and the resulting reconstructions are relatively sparse (i.e. they represent a human using only 15 3D points). Their extended version (Zheng et al., 2017),

further asserts that both outlier correspondences and reduced/small temporal overlap will hinder the accuracy of the temporal alignment.

Besides of Zheng *et al*. (Zheng et al., 2015a, 2017), multi-view geometric reasoning has been employed for the problem of video synchronization. For example, Basha *et al*. (Basha et al., 2012, 2013) proposed methods for computing partial orderings for a subset of images by analyzing the movement of dynamic objects in the images. There, dynamic objects are assumed to move closely along a straight line within a short time period, and video frames are ordered to form a consistent motion model. Tuytelaars and Gool (Tuytelaars and Gool, 2004) proposed a method for automatically synchronizing two video sequences of the same event. They do not enforce any constraints on the scene or cameras, but rather rely on validating the rigidity of at least five non-rigidly moving points among the video sequences, matched and tracked throughout the two sequences. In (Wolf and Zomet, 2006), Wolf and Zomet propose a strategy that builds on the idea that every 3D point tracked in one sequence results from a linear combination of the 3D points tracked in the other sequence. This approach works with articulated objects, but requires that the cameras are static or moving jointly. Finally, Pundik and Moses (Pundik and Moses, 2010) introduced a novel formulation of low-level temporal signals computed from epipolar lines. The spatial matching of two such temporal signals is given by the fundamental matrix relating each pair of images, without requiring pixel-wise correspondences. In this thesis, a method computing spatio-temporal consistent correspondences are proposed to model rigid body movements, which could also be adopted to align unsynchronized video streams.

Single-view video capture can be considered as a dynamic reconstruction scenario inherently lacking the benefits of multi-view synchronization. On this front, the monocular method of Russell *et al*. (Russell et al., 2014) is germane to our approach. The authors employ automatic segmentation of rigid object subparts, for example 3D points on the arms, legs, and torso of a human, and solve the dynamic reconstruction problem by jointly computing hierarchical object segmentation and sparse 3D motion. Their notion of *spatial consistency* of rigid subparts is an important contribution that inspires our approach to unsynchronized multi-view reconstruction. A key distinction is that

our method utilizes perspective camera model, which is not recoverable using monocular input alone.

A critical problem of multiview 3D reconstruction is foreground segmentation, which generate the 2D shape of foreground objects. Many dynamic scene modeling methods only use controlled environments where the background is known or can be accurately estimated. Hasler et al. (Hasler et al., 2009) tackle body shape reconstructions with known background by adopting statistical human body models. While Ballan et. al.(Ballan et al., 2010a) model the background by looking for pixel consistency among multiple views. Taneja et al.(Taneja et al., 2010) propose a method to estimate scene dynamics without making any assumptions on the shape or the motion of elements to be reconstructed. They use the precomputed geometry of the static parts of the scene to transfer the current background appearance across multiply views. And 3D shapes of the dynamic foreground objects are obtained from multiview 2D foreground segmentations. To estimate more accurate dynamic object segmentations, Kim et. al.(Kim et al., 2010) propose a multiple view trimap(with foreground, background and unknown labels) propagation algorithms, which allows trimaps to be propagated across multiple views given a small number of manually specified key-frames trimaps in a single view. Jiang et al.(Jiang et al., 2012) propose a novel dense depth estimation method, which simultaneously solves bilayer segmentation and depth estimation in a unified energy minimization framework. Shape from silhouettes is one popular class of methods to estimate shape of scenes from multiview 2D segmentations. Most of these techniques compute the visual hull, which is the maximal volume consistent with a given set of silhouettes. It was first introduced by Baumgart(Baumgart, 1974), and extensively reviewed by Laurentini(Laurentini, 1994). The visual hull is usually in the format of a 3D volume, which is a subdivision of space into elementary regions, typically as voxels. Many 3D volume-based visual hull methods, including (Furukawa and Ponce, 2006)(Sinha and Pollefeys, 2005)(Bonet and Viola, 1999), are widely used in research works. Due to reasons like camera calibration error and foreground self-occlusion, traditional shape from silhouette is not competent enough to obtain a good reconstruction result, Franco and Boyer (Franco and Boyer, 2005) propose a sensor fusion method to modify this process and generate

more accurate model by accumulating view ray hits for voxels instead of simply carving. In order to address problems like occlusion inference and multi objects modeling, Guan et al.(Guan et al., 2008) further propose a Bayesian fusion framework.

While previous methods discussed in this section mostly belong to model-free methods, model based methods are widely adopted to recover the dynamic 3D geometry. Most of these methods require extra priors on the shape or camera matrices to resolve the ambiguities. (Park et al., 2010; Akhter et al., 2011; Bartoli et al., 2008) assume temporal smoothness by synthesizing motion trajectories with a pre-defined trajectory basis. (C. Bregler and Biermann, 2010; Garg et al., 2013; Zheng et al., 2015a, 2017) assume the 3D shape at any frame can be expressed as a linear combination of an unknown low-rank shape basis governed by time-varying coefficients. To reduce the problem complexity, (C. Bregler and Biermann, 2010; Garg et al., 2013) assume orthogonal camera projection to the image planes instead of projective projections. The methods proposed in this thesis require no shape priors and simpler camera model, instead we adopt motion priors inspired by rigidity.

2.0.2 Appearance Analysis and Mosaics

In this thesis, we present a method to automatically generate image sequences showing smooth illumination change from night-time to day-time (shown in Fig. 5.1). There exists a large body of research on modeling the temporal order of images based on appearance. Wang *et al*. (Wang et al., 2006) propose low-dimensional manifolds to model the gradual appearance change of materials. In order to find smooth transitions between images of faces, Shlizerman *et al*. (Kemelmacher-Shlizerman et al., 2011) build a graph with faces as nodes and similarities as edges, and solve for walks and shortest paths on this graph. For natural scenes like the appearance of the sky, Tao *et al*. (Tao et al., 2009) analyze semantic attributes of sky images, train a classifier to categorize them, and find a smooth sequence of appearance change. To find intermediate images in the sequence, they build an image graph and connect images with their 200 nearest neighbors (in terms of color distance). Their method can also change the sky in an image with the help of interactive

image segmentation and global color transfer. Instead of the sky, we focus on generating the temporal change of more general scenes and adopt local color transfer techniques to better portray the color transition. Schindler and Dellaert (Schindler and Dellaert, 2007) propose a constraint-satisfaction method for determining the temporal ordering of images based on visibility reasoning of reconstructed 3D points. They further present a generalized framework (Schindler and Dellaert, 2010) for estimating temporal variables in SFM problems and obtaining the temporal order of images. Their methods work for images taken over decades of time. Palermo *et al*. (Palermo et al., 2012) extract features that are temporally discriminative and show outstanding results in temporal classification of historical images. Kim *et al*. (Kim et al., 2010) propose a non-parametric approach for modeling and analysis of the topical evolution for Internet images with time stamps. Jacobs *et al*. (Jacobs et al., 2007) created a large dataset of over 500 static web-cameras around the world and propose a method to analyze consistent temporal variations in these scenes. Our proposed method mines unorganized crowd-sourced data to identify a suitable visual datum and generate photo sequences from day to night. There has been tremendous progress in modeling unordered Internet image collections (Frahm et al., 2010; Agarwal et al., 2011; Heinly et al., 2015; Schonberger et al., 2015). The work of Snavely *et al*. (Snavely et al., 2008, 2006) enabled the spatially smooth traversal from Internet images of landmark scenes. Lee *et al*. (Lee et al., 2000) propose a system to "rephotograph" historical photographs. Xu *et al*. (Xu et al., 2008) use collections of images to infer the motion cycle of animals. Hays and Efros (Hays and Efros, 2007) propose an image completion algorithm which fills in empty areas by finding similar image regions in a large dataset.

In this thesis, with a different goal, we aim to visualize the temporal change of scenes by leveraging appearance transfer techniques. To create an appearance mosaic we compose the information from multiple images into a single photo. Agarwala *et al*. (Agarwala et al., 2004) adopt graphcut and gradient domain fusion to choose good seams between images and reduce visible artifacts in a composite image. To stitch a set of images, Levin *et al*. (Levin et al., 2004) introduce several formal cost functions for the evaluation of the quality of stitching. The most used one is evaluating boundary alignment consistency. Zhang and Liu (Zhang and Liu, 2014) propose a hybrid

alignment model that combines homography and content-preserving warping to provide flexibility for handling parallax. However, this method is not designed to align image sequences and did not show results to align images with very different illuminations.

2.0.3 View Synthesis and Visual Predictions

Long range motion flow. Optical flow estimation among successive frames is mainly used to generate motion flows (Black and Anandan, 1991; Elad and Feuer, 1998; Shi and Malik, 1998). Brox *et al*. (Brox et al., 2014; Papenberg et al., 2006) estimate optical flows simultaneously within multiple frames by adopting robust spatio-temporal regularization. Wills and Belongie (Wills and Belongie, 2004) estimate dense correspondences of image pairs using a layered representation initialized with sparse feature correspondences. Irani (Irani, 1999) describes linear subspace constraints for flow across multiple frames. Brand (Brand, 2001) applies a similar approach to non-rigid scenes. Sand and Teller (Peter Sand, 2006) propose to represent video motion using a set of particles, which are optimized by measuring point-based matching along the particle trajectories and distortion between the particles.

Future prediction. Future prediction has been used in various tasks such as estimating the future trajectories of cars (Walker et al., 2014), pedestrians (Kitani et al., 2012), or general objects (Yuen and Torralba, 2010) in images or videos. Given an observed image or a short video sequence, models have been proposed to predict a future motion field (Liu et al., 2009b; Pintea et al., 2014; Walker et al., 2015, 2016). Zhou and Berg (Zhou and Berg, 2015) frames the prediction problem as a binary selection task to determine the temporal sequence of two video clips. (Vondrick et al., 2015) trains a deep network to predict visual representations of future images with large amounts of unlabeled video data from the Internet.

View synthesis with CNN. Recent methods for synthesizing novel views, objects, or scenes under diverse view variations have been boosted by the ability of Convolutional Neural Networks (CNNs) to function as image decoders. Hinton *et al*. (Hinton et al., 2011) learned a hierarchy of capsules, computational units that locally transform their input, for generating small rotations to an

input stereo pair. Dosovitskiy *et al*. (Dosovitskiy et al., 2015) learned a generative CNN model to hallucinate chairs with respected to given input graphics codes i.e. identity, pose, and lighting. Inspired by this paper, Tatarchenko *et al*. (Tatarchenko et al., 2016) and Yang *et al*. (Yang et al., 2015) instead adopt an encoder-decoder network to implicitly learn graphics code from training image pairs or sequences. Tatarchenko *et al*. (Tatarchenko et al., 2016) proposed an approach to predict images and silhouette masks without explicit decoupling of identity and pose. Yang *et al*. (Yang et al., 2015) applied input transformation to the learned pose units of source images to obtain desired target images, and apply a recurrent network to enable synthesizing sequences with large viewpoint difference.

Since the above methods generate new pixels from scratch and thus the synthesized results will tend to be blurry. Zhou *et al*. (Zhou et al., 2016) propose to use the pixels of the input image as much as possible, by learning the pixel correspondences within given input images. This method can obtain synthesis with crisp texture and much less blurriness. However, since this method poses no constraints on the learned appearance flow, some of the generated synthesis has large texture distortions. Generative adversarial networks (GANs) have shown great promise for improving image generation quality (Goodfellow et al., 2014). GANs are composed of two parts, a generative model and a discriminative model, to be trained jointly. Some extensions have combined GAN structure with multi-scale laplacian pyramids to produce high-resolution generation results (Denton et al., 2015). Inspired by (Zhou et al., 2016), this thesis proposes a method to implicitly learn motion flow within visual prediction neural networks, which has the potential to generate images with better details than image prediction approaches e.g. (Zhou and Berg, 2015).

CHAPTER 3: 3D RECONSTRUCTION OF DYNAMIC TEXTURES IN CROWD SOURCED DATA

3.1 Introduction

State of the art crowd sourced 3D reconstruction systems deploy structure from motion (SfM) techniques leveraging large scale imaging redundancy in order to generate photo-realistic models of scenes of interest. The estimated 3D models reliably depict both the shape and appearance of the captured environment under the joint assumptions of shape constancy and appearance congruency, commonly associated with static structures. Accordingly, the attained 3D models are unable to robustly capture dynamic scene elements not in compliance with the aforementioned assumptions. In this work, we strive to estimate more complete and realistic 3D scene representations by addressing the 3D modeling of dynamic scene elements within the context of crowd sourced input imagery.

In our crowd sourced 3D modeling framework, dynamic scene content can only be determined through the observation of visual motion. Nelson and Polana (Nelson and Polana, 1992) categorized visual motion into three classes: activities, motion events and dynamic (temporal) texture. *Activities*, such as walking or swimming, are defined as motion patterns that are periodic in time; *motion events*, like opening a door, lack temporal or spatial periodicity; *dynamic textures*, i.e. fire, smoke and flowing water exhibit statistical regularity but have uncertain spatial and temporal extent. Dynamic scenes may contain visual motions in any combination these three categories. Our work focuses on modeling the 3D shape of scene elements belonging to the dynamic texture category, working under the assumption of a shape-fixed surface. Moreover, while our framework assumes the geometry of scene elements having time-varying appearance (i.e. such as active billboards or bodies of water) to be approximated by a single surface, our solution is completely data-driven and does not impose geometric or shape priors to perform our estimation.

First, we briefly give an overview of the functionality of our processing pipeline. The input data to our framework encompasses both online image and video collections capturing a common scene. We initially leverage photo-collection data to perform sparse reconstruction of the rigid scene elements. Then, the video collection is analyzed to reap video segments amenable for 1) registration to our existing rigid model and 2) coarse identification of dynamic scene elements. We use these coarse estimates, along with the knowledge of our sparse rigid 3D structure to pose the segmentation of dynamic elements within an image as a global two-label optimization problem. The attained dynamic region masks are subsequently fused through shape-from-silhouette techniques in order to generate an initial 3D shape estimate from the input videos. The preliminary 3D shape is then back projected to the original photo-collection imagery, all image labelings are recomputed and then fused to generate an updated 3D shape. This process is iterated until convergence of the output photo-collection imagery segmentation process. Figure 3.1 depicts an overview of the proposed pipeline.

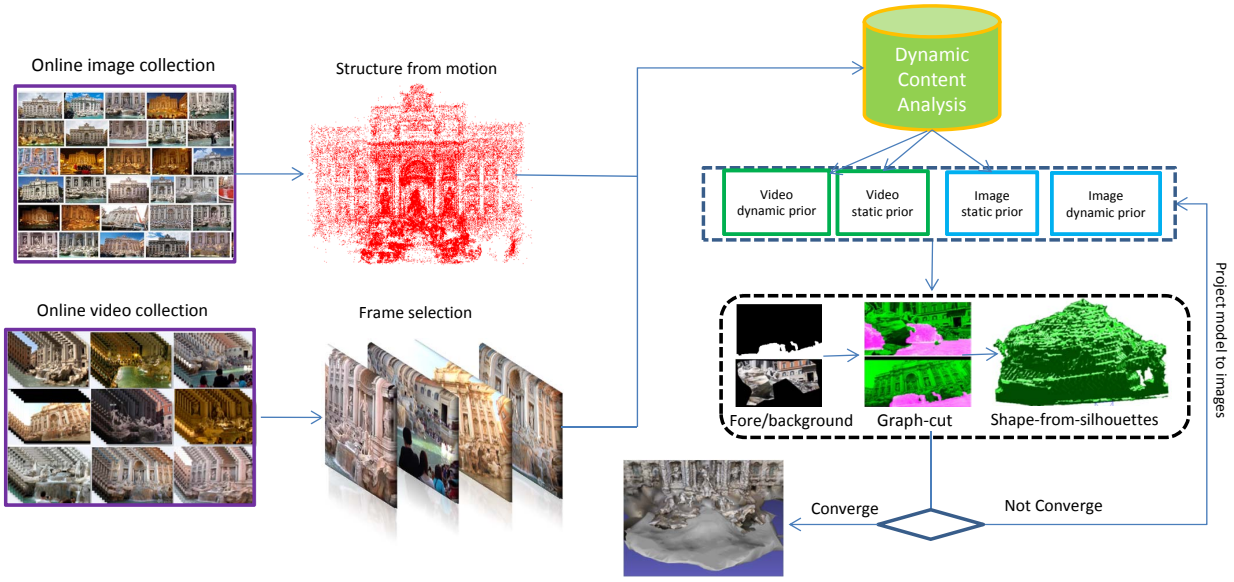


Figure 3.1: Workflow overview of the proposed framework.

Our developed system improves upon existing 3D modeling systems by increasing the coverage of the generated modeling, mitigating spurious geometry caused by dynamic scene elements and enabling more photo-realistic visualizations through the explicit identification and animation of

model surfaces having time varying appearance. The remainder of this chapter describes the design choices and implementation details of different modules comprising our dynamic scene content modeling pipeline.

3.2 Initial Model Generation

3.2.1 Static Reconstruction from Photo Collections

The first step in our pipeline is to build a preliminary 3D model of the environment using photo-collection imagery. To this end we perform keyword and location based queries to photo sharing websites Flickr & Panoramio. We perform GIST (Oliva, 2005) based K-means clustering to attain a reduced set images on which to perform exhaustive geometric verification. We take the largest connected component in the resulting camera graph, consisting of pairwise registered cluster centers, as our initial sparse model and perform intra-cluster geometric verification to densify the camera graph. The final set of images is fed to the publicly available VisualSfM module to attain a final sparse reconstruction. The motivation for using VisualSfM is the availability for direct comparison against two input compatible surface reconstruction modules PMVS2(Furukawa and Ponce, 2010) by Furukawa & Ponce and CMPMVS(Jancosek and Pajdla, 2011) by Jancosek & Pajdla. Once a static sparse model is attained the focus shifts to identifying additional video imagery enabling the identification and modeling of dynamic scene content.

3.2.2 Coarse Dynamic Textures Priors from Video

Video collections are the natural media to identify and analyze dynamic content. To this end we download videos from YouTube using tag queries of the scenes of interest . Our goal is to identify and extract informative video fragments within our downloaded set of videos. We consider as informative, those video subsequences where the dynamic texture content can be distinguished and reliably correlated with our existing sparse model of the scenes static structure.

Video Frame Registration. We temporally sample each video at a 1/50 ratio to obtain a reduced set of video frames for analysis. For illustration, a set of 500 videos generated little over 80K frame samples. We introduce into the video frame set a random subset of 30% of the registered cameras from the rigid scene modeling. We again perform GIST based clustering on the augmented image set and re-run intra cluster geometric verification to identify registered video frames. In principle, the entire process can be performed directly on the joint set of input photo-collection images and sampled video frames. However, we found the implemented two stage image and video registration to provide more robust performance and increased video frame registration.

Video Sub-sequence Selection. Given a reduced set of registered video frames we want to select compact frame sub-sequences having reduced camera motion in order to simplify the detection of dynamic scene content. Namely, we compute the HOG descriptor for the frames immediately preceding and following a registered video frame in the original sequence. We count the number of neighboring frames having an NCC value in the range $(0.9, 1)$ w.r.t. the registered frame and keep those sequences having cardinality above a given threshold τ_{seq_len}. We favor such image content based approach instead of pair-wise camera motion estimation due to the difficulty in defining suitable capture dependent thresholds (i.e. camera motion, lighting changes, varying zoom, etc.). Discarding fully correlated (i.e. NNC=1) pairwise measurements enables the elimination of duplicates. Moreover, we found measuring the NCC over the HOG descriptors to be robust against abrupt dynamic texture variation as long such changes were restricted to reduced image regions. Figure 3.2 describes the selection thresholds utilized for subsequence detection.

Barebones Dynamic Texture Estimation. In order to segment dynamic texture from static backgrounds on the selected short video sequences, we deploy basic frame differencing by accumulating the inter-frame pixel intensity differences. We compensate for (the reduced) camera motion by performing RANSAC based homography warping of all sub-sequence frames to the anchor (i.e. registered) video frame. The accumulated difference image is then binarized using non-parametric Otsu thresholding (Otsu, 1979). The attained mask is then modified by a sequence of closing-erosion morphological operations with respective window sizes of 2×2, 11×11 and 9×9

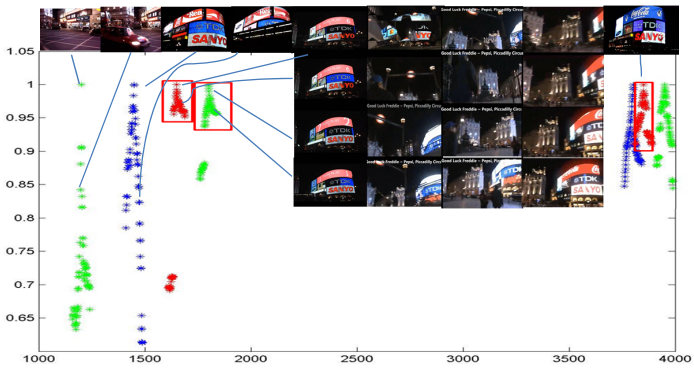


Figure 3.2: Keyframe selection for an input video. The plot shows the frame number count vs the NCC similarity of each frame's HOG descriptor. Red boxes indicate selected video fragments centered on sampled frames. Keyframe selection is a function of the plot density at the upper end of the NNC values.

for an input image of VGA resolutions. We sort the connected component of the binary output image w.r.t. their area and eliminate all individual components at the bottom 10% of total image area (shown in Figure 3.3).

Figure 3.3: Dynamic content priors from video fragments. Left to right: (a) Reference frame (b) Accumulated frame differencing (c) Result after post processing.

3.2.3 Coarse Static Background Priors from Video Frames

We leverage the dense temporal sampling within a single video sub-sequence in order to estimate a mask for static texture observed on all selected reference video frames. Instead of naively

using the complement of the precomputed dynamic texture mask for a given video frame, we strive to deploy a more data-driven approach. To this end we analyze the sparse feature similarity among the reference frame and one of its immediate neighbors. We retrieve the set of putative SIFT matches previously used for homography based stabilization of the video sequence and perform RANSAC based epipolar geometry estimation. We consider the attained set of inlier image features in the reference videoframe as a sparse sample of the observed static structure. To mitigate spurious dynamic features being registered due to low frequency appearance variations, we exclude from this set any features contained within the regions described by dynamic texture mask. From the final image feature set we compute the concave hull and use the attained 2D polygon as an area based prior for static scene content(shown in Figure 3.4).

Figure 3.4: Static content prior from video fragments. First and third columns depict SIFT features matches among neighboring frames as red dots. Second and fourth columns depict the concave hull defined by detected features not overlapping with the existing dynamic content prior.

3.2.4 Graph-cut based dynamic texture refinement

Once a preliminary set of segmentation masks for static and dynamic object regions are attained, they are refined trough a two label (e.g. foreground/background) graph-cut labeling optimization framework. We will denote static structure as background and dynamic content as foreground. The optimization problem in Graphcut is defined as:

$$\min E(f) = \sum_{u \in \mathcal{U}} D_{\mathcal{U}}(f_u) + \sum_{u,v \in \mathcal{N}} V_{u,v}(f_u, f_v) \tag{3.1}$$

where $f_u, f_v \in \{0, 1\}$ are the labels for pixels u and v, $\mathcal{N}$ is the set of neighboring pixels for u and $\mathcal{U}$ denotes the set of all the pixels with unknown labels. Similarly to the work of Jiangyu (Liu et al., 2009a), we use a Gaussian mixture model to compute the foreground/background membership probabilities of a pixel. Hence, the smoothness term is defined to be:

$$V_{u,v}(f_u, f_v) = |f_u - f_v| \exp(-\beta(I_u - I_v)^2), \tag{3.2}$$

where I_u, I_v denote the RGB values of pixels u and v, while $\beta = (2 \langle (I_u - I_v)^2 \rangle)^{-1}$, for $\langle \cdot \rangle$ denoting the expectation over an image sample. Conversely, the data term is defined as:

$$D_u(f_u) = \log \left(\frac{p(f_u = 1)}{p(f_u = 0)} \right), \tag{3.3}$$

$$\begin{aligned} p(f_u = 1) &= p(I_u|\lambda_1) = \textstyle\sum_{i=1}^{M} \omega_{i1} g(I_u|\mu_{i1}, \Sigma_{i1}) \\ p(f_u = 0) &= p(I_u|\lambda_0) = \textstyle\sum_{i=1}^{M} \omega_{i0} g(I_u|\mu_{i0}, \Sigma_{i0}) \\ \lambda_{1|0} &= \{\omega_{i1|0}, \mu_{i1|0}, \Sigma_{i1|0}\}, i \in \{1, 2, ..., M\} \end{aligned}$$

and $g(I_u|\mu_i, \Sigma_i)$ belongs to a mixture-of-gaussian model using $M = 3$, and we assume the labels for fore/background are 1/0.

Figure 3.5 exemplifies the result of graph-cut segmentation. Moreover, the output regions categorized as foreground (i.e. dynamic content)

3.2.5 Shape from silhouettes

We leverage the output of our graph-cut segmentation module to estimate the 3D visual hull of the dynamic texture through space carving methods. Namely, we utilize the refined dynamic content mask as an object silhouette, along with the corresponding camera poses and calibration

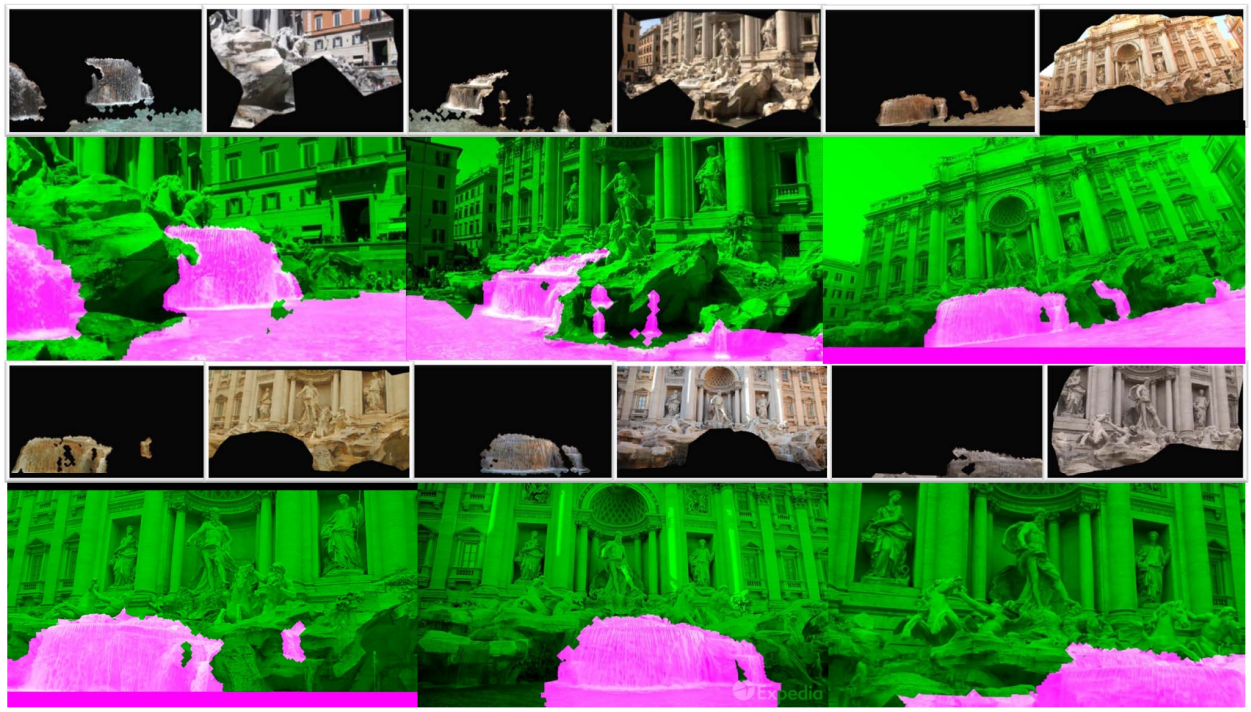


Figure 3.5: Graphh-cut label refinement. First and third rows depict (alternatively from left to right) single image dynamic and static content priors. Second and fourth rows depict the outputs of the label optimization, where green regions are dynamic textures.

estimate, to deploy a 3D fusion method estimating a volumetric shape representation in accordance to the steps described in Algorithm 1.

Algorithm 1: SHAPE FROM SILHOUETTES FUSION

Input: Sets of camera poses $\{\mathbb{C}_i\}$ and corresponding foreground silhouettes $\{\mathbb{M}_i\}$, where $i \in [1, \cdots, N]$, 3D occupancy grid O, threshold θ_1

Output: Labeled 3D occupancy grid V

Set all $O(x, y, z) = 0$

for $i \in [1, N]$ **do**

  for *pixel* $\mathbf{M}_{ij} \in \{\mathbb{M}_i\}$ **do**

    Find all voxels $O_{x,y,z}, \{x, y, z\} \in O_1 \subset O$,$Proj^i(O_1) = M_{ij}$

    $O_1 \leftarrow O_1 + 1$

$V = Find(\{x, y, z\}|O_{x,y,z} > \theta_1), \{x, y, z\} \in V \subset O$

Label voxels in V as occupied.

Algorithm 2: SHAPE FROM SILHOUETTES FUSION

Input: Sets of camera poses $\{\mathbb{C}_i\}$ and corresponding foreground silhouettes $\{\mathbb{M}_i\}$ and background silhouettes $\{\mathbb{M}'_i\}$, where $i \in [1, \cdots, N]$, 3D occupancy grid O, threshold θ_1, θ_2

Output: Labeled 3D occupancy grid V

Set all $O(x, y, z) = 0$
for $i \in [1, N]$ **do**
    for *pixel* $\mathbf{M}_{ij} \in \{\mathbb{M}_i\}$ **do**
        Find all voxels $O_{x,y,z}, \{x, y, z\} \in O_1 \subset O$,$Proj^i(O_1) = M_{ij}$
        $O_1 \leftarrow O_1 + 1$
$V = Find(\{x, y, z\}|O_{x,y,z} > \theta_1), \{x, y, z\} \in V \subset O$
Set all $V(x, y, z) = 0$
for $i \in [1, N]$ **do**
    for *pixel* $\mathbf{M}'_{ij} \in \{\mathbb{M}'_i\}$ **do**
        Find all voxels $V_{x,y,z}, \{x, y, z\} \in V_1 \subset V$,$Proj^i(V_1) = M'_{ij}$
        $V_1 \leftarrow V_1 + 1$
$V = Find(\{x, y, z\}|V_{x,y,z} < \theta_2), \{x, y, z\} \in V$
Label voxels in V as occupied.

3.3 Closed Loop 3D Shape Refinement

The preceding section described a video based approximation of the observed shape of dynamic texture within the scene. The motivation for exclusively using video keyframes until now has been the lack of a mechanism to estimate dynamic texture prior for static images. In this section, we describe an iterative mechanism to effectively transfer the labelings attained from video sequences to the available photo-collection imagery. Such label transferring will enable us to leverage and augmented imagery dataset offering 1) increased robustness through additional redundancy and viewpoint diversity as well as 2) increased level of detail afforded by larger available imaging resolutions.

3.3.1 Geometry based Video to Image Label Transfer

In order to transfer dynamic content masks from videos into static images we leverage the estimated preliminary 3D volume. The process is as follows:

1. Generate static background priors for each image.

2. Project the preliminary 3D shape model to all registered images and use its silhouette as a dynamic foreground prior for each image.

3. Execute graph-cut based label optimization

4. Generate an updated 3D model using the shape from silhouettes module.

Steps 2 to 4 in the above method will iterate until convergence of the dynamic foreground prior mask. Note that in such a framework the static background priors are kept constant while the dynamic texture content is a function of an evolving 3D shape. In general, the preliminary model attained from videos sequences may suffer from a variability viewpoint coverage or be sensitive to errors in our video based video segmentation estimates. While the former may either under-constrain or bias the attained 3D shape, the latter may arbitrarily corrupt the estimate. Both of these challenges are addressed through the additional sampling redundancy afforded by image photo-collections. The remaining challenges consist then in robustly defining static content priors for single images and adapting the shape estimation framework to adequately handle the heterogeneous additional imaging data.

3.3.2 Mitigating Dynamic Texture in SfM Estimates

The variability in the temporal behavior and extent of dynamic textures may enable its spurious inclusion within SfM estimates. Namely, it is possible for changes in appearance to manifest themselves at time scales larger than those encompassed through short video subsequences or to present periodic behavior that would enable feature correspondence across multiple unsynchronized image captures. We evaluate the appearance variability of sparse reconstructed features across the imaging dataset to classify them having either persistent or sporadic color.

In principle, static 3D structure with constant appearance should provide consistent color throughout all images observing said structure. Conversely, features with sporadic color are mainly observed from dynamic structures, for example: flowing water from a fountain, moving waves in

ocean, flashing letters on a billboard etc. The existence of reconstructed features within a dynamic texture obeys mainly to the transient nature of their appearance. That is, while such appearance is observable at multiple different times, the same structure element may alternatively display appearance independent of the one used for matching.

Moreover, according to Lambert's cosine law, if the colors of a static structure remains constant, the observed pixels are linearly correlated to intensity of the incoming light, as described by

$$I_D = \mathbf{L} \cdot \mathbf{N} C I_L = C I_L \cos \alpha, \tag{3.4}$$

Where **L** and **N** are the normalized incoming light direction and the normalized normal for 3D object, C and I_L the color of the model and the intensity of incoming light respectively, making the reflection color I_D a linear function of I_L (with slope $\cos \alpha$). Given that robust features (e.g. SIFT, SURF) enable the robust detection of features even in the presence of such lighting variations, we can generally expect the color variability of a static feature to comply with such linear behavior. Based on this assumption, we propose a simple method for consistency detection. First we re-project each reconstructed feature to all cameras observing the same structure and record the observed RGB pixel color. Note we re-project to all cameras where the feature fall within the viewing frustum, not just those cameras where the feature was detected. We perform RANSAC based line fitting on the set of measured RGB values to determine the inlier ratio ϵ for a pre-specified distance $d_1 = 0.08$ in the RGB unit color cube. We consider any feature with an estimated inlier ratio below 0.6 to have sporadic color. Figure 3.6 shows the results running our method on a billboard dataset. Moreover, the set of features classified as having sporadic color will be subsequently used to filter sparse SfM estimates corresponding to static structure.

3.3.3 Building a Static Background Prior for Single Images

We leverage the dense spatial sampling within image photo-collections in order to estimate a mask for the static structure observed on all images registered by SfM. In order to achieve as

Figure 3.6: Identification of dynamic textures within existing SfM estimates.Top Row: birds-eye and fontal view of estimated sparse structure for Piccadilly Circus. Blue dots are 3D features with persitent color across the dataset. Red dots are 3D features determined to have sporadic color. The bottom row shows sample images in the dataset. We associate color persistance with predominantly linear variation in the RGB space.

dense as possible sampling of static structure within the image, we retrieve the set inlier feature matches previously attained by pairwise geometric verification to its closest registered neighbor in GIST-space. We then exclude from this feature set the subset of features having sporadic color across the entire dataset. There is a coverage to accuracy tradeoff in selecting the pairwise inlier feature set instead of the final reconstructed feature set for each image. In order to mitigate the effect of spurious dynamic texture features, we define a sparse background prior, where each feature location is dilated to define a background mask comprising multiple (possibly overlapping) blob structures. We note the contrast with the area-based static prior masks estimated from video (i.e. determined by the concave hull of features). Our rationale is that while the dense spatial sampling of video sequences affords strong spatial correlations, the viewpoint and temporal variability of sparse SfM features provides tightly localized correlations. Moreover, the elimination of features having sporadic color from the static prior enables more robust segmentation by the subsequent graph-cut label refinement.

3.3.4 Mitigating of Non-uniform Spatial Sampling

In order to generate accurate 3D shape models of dynamic scene elements through space carving methods wide spatial coverage of cameras is a requisite. In fact, this is the motivation of using photo-collection images. However, the availability of abundant images also presents challenges when said imagery is not uniformly distributed within the scene. Namely, we require a large number of viewing rays tangent to the shape's surface in order for the estimated visual hull to accurately approximate the observed surface. Moreover, our basic shape from silhouettes method will favor the identification of commonly observed image regions. For example, to accurately estimate the shape for the Piccadilly circus billboard (which is a round rectangled shape), we require cameras located in a range of nearly $270°$. Figure 3.7 shows the reconstruction of Piccadilly circus using 5800 iconic images(from more than 60,000 images). We can see the camera distribution is not uniform providing scarce coverage of the tangent views of the billboard. Densely distributed cameras dilute the weights of sparsely distributed cameras in (Algorithm 2). And the generated 3D shapes will deform towards the direction of densely distributed cameras (shown in Figure 3.7(c)). In order to deal with the uncontrolled viewpoint distribution we deploy a weighting mechanism (Algorithm 3) within our image base shape from silhouettes framework. The idea is to reduce relative weights of densely distributed cameras. To obtains new set of weights for cameras, we compute viewing ray angle between reference camera and other cameras, generating a histogram for the angles and set the weights of cameras located in each bin with the inverse of camera numbers. After applying weighting strategy, the computed models are visually more natural.

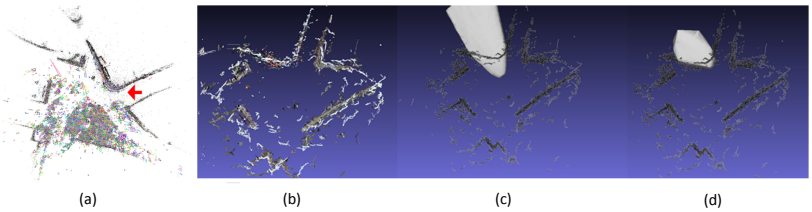


Figure 3.7: Mitigation of non uniform spatial sampling. Left to right: (a) Cameras in the red arrow direction are scarse in the SfM model (b) Quasi-dense output from PMVS (c) Dynamic Shape estimation with uniformly weighted carving. the reconstructed 3D volume will be towards the cmera centroid (d) Shape estimate with weighted carving.

Algorithm 3: camera weighting strategy

Input: A initial model M_0, camera centers $C_i, i \in [1, \cdots, N]$, cameras field-of-view angles $f_i, i \in [1, \cdots, N]$

Output: Space carving weight w_i for each camera

for $i \in [1, \cdots, N]$ **do**
- Direction vector of each camera center $v_i \leftarrow C_i - centroid(M_0)$
- Direction angle of each camera center $a_i \leftarrow \arccos \frac{v_i * v_{N/2}}{norm(v_i) norm(v_{N/2})}$
- $w_i = 1$

Discretize the direction angles into 5 bins histogram centered at $B_j, j \in [1, \cdots, 5]$, with frequency $H_j, j \in [1, \cdots, 5]$

for $i \in [1, \cdots, N]$ **do**
- $idx = find(j | B_j \leq a_i < B_{j+1})$
- $w_i \leftarrow w_i * min(H)/H_{idx}$
- $w_i \leftarrow w_i * min(f)/f_i$

3.4 Experiments

We downloaded 4 online datasets from the Internet, with videos attained from Youtube and images from Flickr. The statistics of our data associations are presented in Table 3.1. For all datasets, the set of registered images and the final sparse SfM was generated using visualSfM. Figure 3.9 shows our results combining PMVS quasi-dense model and our dynamic texture shape estimate.

Table 3.1: Composition of our downloaded crowd sourced datasets

Dataset	Videos Downloaded	Keyframes Extracted	Images Downloaded	Images Registered
Trevi Fountain	481	68629	6000	810
Navagio Beach	300	45823	1000	520
Piccadilly Circus Billboard	460	75983	5000	496
Mooney Falls	200	17850	1000	723

To illustrate the iterative space carving method, we show the segmented estimated visual hull result in each iteration using the Trevi Fountain dataset (Fig. 3.8). For the the first iteration we use an interaction count ratio of 0.10 and increment this value by 0.03 each iteration. To ensure convergence of the iteration, we choose a random subset of wide field-of-view images and test their segmentation change in each iteration.

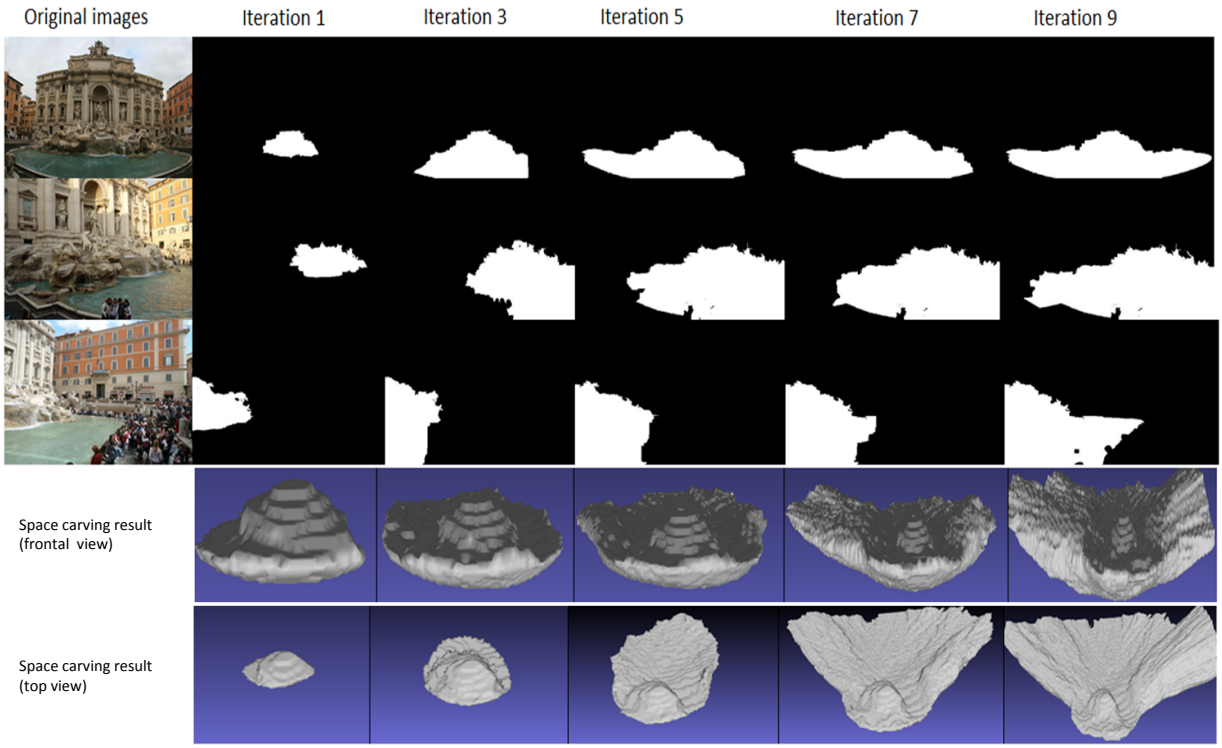


Figure 3.8: Evolution of estimated 3D dynamic content in Trevi Fountain model. The video based model only identified the water motion in the central part of the fountain. Iterative refinement extends the shape to the brim of the fountain. Top rows depict the evolving segmentation mask. Bottom rows depict the evolving 3D shape.

The efficacy of our weighted space carving method for photo collection imagery is illustrated for the Piccadilly Circus Billboard dataset in Figure 3.7 We can see in the absence of camera contribution weighting, the model will outstretch in the direction of greater camera density. The effect is effectively mitigated by our weighting approach. We also generate the textured 3D model and compare the results generated by the state-of-the-art method CMPMVS (Jancosek and Pajdla, 2011) (Fig. 3.9). For all the experiments, we use the same input dataset for comparison. Each dataset takes a approximately 24 hours of processing using both methods.

Figure 3.9: Top two rows: sample dataset imagery, respective outputs for PMVS, CMPMVS and our proposal. Bottom two rows: sample dataset imagery, respective outputs for PMVS and our proposal; CMPMVS failed to generate on the same input data.

To illustrate the generality of the proposed framework, we also considered a controlled capture scenario of an indoor scene containing a flat surface with varying illumination. Adapting our method to work with a single input video, instead of crowd sourced data, we were able to generate a 3D

approximation of the screen surface of an electronic tablet displaying dynamic texture(shown in Fig. 3.10). In practice, the inability to attain observations of the dynamic texture of a flat surface from completely oblique views yielded a piece-wise planar 3D surface with slight outside of plane protrusions. Nevertheless, our attained 3D model was amenable for video texture mapping yielding a realistic animation of the captured video.

Figure 3.10: From left to right: sample dataset imagery, respective outputs of PMVS, CMPMVS and our proposed method.

3.5 Conclusion

We proposed a crowd sourced 3D modeling framework encompassing scene elements having dynamic appearance but constant shape. By leveraging both online video and photo-collections we enable the analysis of scene appearance variability across different time scales and spatial layout. Building upon standard SfM, scene labeling and silhouette fusion modules our system can provide, in a fully automated way, more complete representations of captured landmarks containing dynamic elements such as bodies of water surfaces and active billboards. Moreover, the segregation of the scene content into static and dynamic elements enables compelling visualizations that incorporate the texture dynamics and effectively ”bring 3D models to life”.

CHAPTER 4: SPATIO-TEMPORALLY CONSISTENT CORRESPONDENCE FOR DENSE DYNAMIC SCENE MODELING

4.1 Introduction

Dynamic 3D scene modeling addresses the estimation of time-varying geometry from input imagery. Existing motion capture techniques have typically addressed well-controlled capture scenarios, where aspects such as camera positioning, sensor synchronization, and favorable scene content (*i.e.* fiducial markers or "green screen" backgrounds) are either carefully designed *a priori* or controlled online. Given the abundance of available crowd-sourced video content, there is growing interest in estimating dynamic 3D representations from uncontrolled video capture. Whereas multi-camera static scene reconstruction methods leverage photoconsistency across spatially varying observations, their dynamic counterparts must address photoconsistency in the spatio-temporal domain. In this respect, the main challenges are 1) finding a common temporal reference frame across independent video captures, and 2) meaningfully propagating temporally varying photo-consistency estimates across videos. These two correspondence problems – temporal correspondence search among unaligned video sequences and spatial correspondence for geometry estimation – must be solved jointly when performing dynamic 3D reconstruction on uncontrolled inputs.

In this work, we address both of these challenges by enforcing the geometric consistency of optical flow measurements across spatially registered video segments. Moreover, our approach builds on the thesis that *maximally consistent geometry is obtained with minimal temporal alignment error*, and *vice versa*. Towards this end, we posit that it is possible to recover the spatio-temporal overlap of two image sequences by maximizing the set of consistent spatio-temporal correspondences (that is, by maximizing the completeness of the estimated dynamic 3D geometry) among the two video segments.

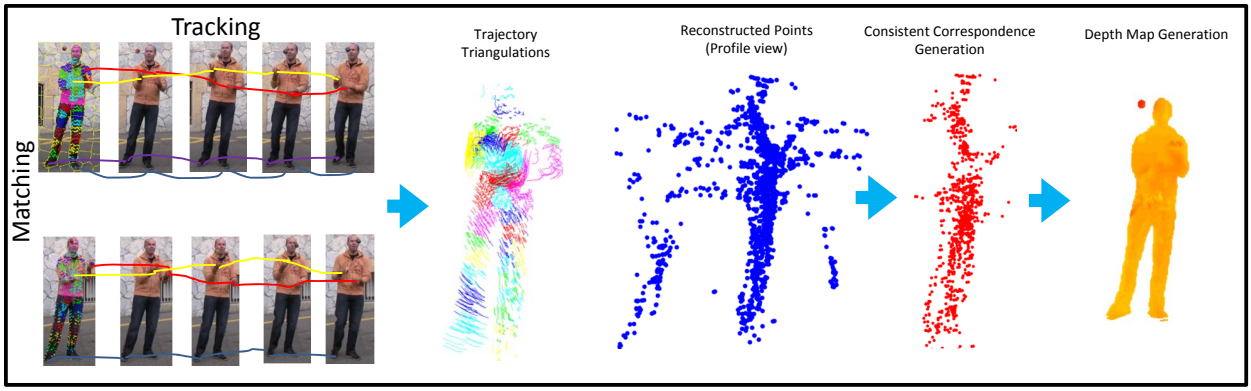


Figure 4.1: Overview of the proposed approach for dense dynamic scene reconstruction from two input video streams.

In practice, our approach addresses the spatio-temporal two-view stereo problem. Taking as input two unsynchronized video streams of the same dynamic scene, our method outputs a dense point cloud corresponding to the evolving shape of the commonly observed dynamic foreground. In addition to outputting the observed 3D structure, we estimate the temporal offset of a pair of input video streams with a constant and known ratio between their frame rates. An overview of our framework is shown in Fig. 4.1. Our framework operates within local temporal windows in a strictly data-driven manner to leverage the low-level concepts of local rigidity and non-local geometric coherence for robust model-free structure estimation. We further illustrate how our local spatio-temporal assumptions can be built to successfully address problems of much larger scope, such as content-based video synchronization and object-level dense dynamic modeling.

4.2 Spatio-Temporal Correspondence Assessment

Our goal is to analyze two spatially-registered video sub-sequences of equal length, in order to determine the largest set of spatio-temporally consistent pixel correspondences belonging to a commonly observed dynamic foreground object. In particular, we are interested in building two-view correspondence-based visual 3D tracks spanning the entire length of the sub-sequences and assessing the validity of the initial correspondences in terms of the geometric properties of the 3D tracks. Our goal has two complimentary interpretations: 1) to develop a spatio-temporal correspondence

filtering mechanism, and 2) to provide a measure of local spatio-temporal consistency among video sub-sequences in terms of the size of the valid correspondence set. We explore both these interpretations within the context of video synchronization and dense dynamic surface modeling.

4.2.1 Notation

Let $\{\mathcal{I}_i\}$ and $\{\mathcal{I}'_j\}$ denote a pair of input image sequences, where $1 \leq i \leq M$ and $1 \leq j \leq N$ are the single image indices. For each image $\mathcal{I}_k \in \{\mathcal{I}_i\} \cup \{\mathcal{I}'_j\}$, we first obtain via structure-from-motion (SfM) a corresponding camera projection matrix, $\mathbf{P}(\mathcal{I}_k) = \mathbf{K}_k \left[\mathbf{R}_k | -\mathbf{R}_k \mathbf{C}_k\right]$, where $\mathbf{K}$, $\mathbf{R}$, and $\mathbf{C}$ respectively denote the camera's intrinsic parameter matrix, external rotation matrix, and 3D position. Let $\mathbf{F}_{ij}$ denote the fundamental matrix relating the camera poses for images $\mathcal{I}_i$ and $\mathcal{I}'_j$. Furthermore, let $\mathcal{O}_i$ and $\mathcal{O}'_j$ denote optical flow fields for corresponding 2D points in consecutive images (*e.g.* $\mathcal{I}_i \rightarrow \mathcal{I}_{i+1}$ and $\mathcal{I}'_j \rightarrow \mathcal{I}'_{j+1}$) in each of the two input sequences. Finally, let $\mathbf{x}_{ip}$ and $\mathbf{X}_{ip}$ denote the 2D pixel position and the 3D world point, respectively, for pixel p in image $\mathcal{I}_i$ (and similarly $\mathbf{x}'_{jp}$ and $\mathbf{X}'_{jp}$ for image $\mathcal{I}'_j$).

Figure 4.2: (a) Background mask that has high color consistency. (b) Foreground mask with low color consistency. (c) Segmented result.

4.2.2 Pre-processing and Correspondence Formulation

Spatial Camera Registration. Our approach takes as input two image streams capturing the movements of a dynamic foreground actor, under the assumption of sufficient visual overlap that enables camera registration to a common spatial reference defined by a static background structure. Inter-sequence camera registration is carried out in a pre-processing step using standard SfM

methods (Wu et al., 2011) over the aggregated set of frames, producing a spatial registration of the individual images from each stream. Since the goal of this stage is simply image registration of the two sequences, the set of input images for SfM can be augmented with additional video streams or crowd-sourced imagery for higher-quality pose estimates; however, this is not necessarily required for our method to succeed.

Dynamic Foreground Segmentation. SfM simultaneously recovers the camera poses for the input images and reconstructs the 3D structure of the static background. The first step in our method is to build a reliable dynamic foreground mask for each image using the available 3D SfM output. At first blush, it seems that this task can be accomplished by simply reprojecting the SfM 3D points into each image and aggregating these projections into a background mask. However, this approach is less effective for automatic foreground segmentation primarily because it does not account for spurious 3D point triangulations of the dynamic foreground object. Hence, to identify the non-static foreground points in an image, we adopt a three-stage process: First, we perform RANSAC-based dominant 3D plane fitting on the SfM point cloud, under the assumption that large planar structures will be part of the background. We iteratively detect dominant planes until we have either included over 70% of available points or the estimated inlier rate of the current iteration falls below a pre-defined threshold. Second, for the remaining reconstructed 3D points not belonging to a dominant plane, we identify their set of nearest 3D neighbors and measure the photoconsistency of this set with their corresponding color projections into the image under consideration. We measure the normalized cross correlation (NCC) of these samples and threshold values above 0.8 as background and below 0.5 as foreground. Third, we perform a graph-cut optimization to determine a global foreground-background segmentation, where we use the points on the dominant planes along with photoconsistent reprojections as initial background seeds, while the non-photoconsistent pixels are considered foreground seeds. Fig. 4.2 illustrates an example of our segmentation output.

Correspondence Search Space. Consider two temporally corresponding image frames $\mathcal{I}_i$ and $\mathcal{I}'_j$. For a given pixel position $\mathbf{x}_{ip}$ contained within the dynamic foreground mask of image $\mathcal{I}_i$, we can readily compute a correspondence $\mathbf{x}'_{jp}$ in image $\mathcal{I}'_j$ by searching for the most photoconsistent candidate along the epipolar line $\mathbf{F}_{ij}\mathbf{x}_{ip}$. We can further reduce the candidate set $\Omega(\mathbf{x}_{ip}, \mathbf{F}_{ij}) \in \mathcal{I}'_j$ by only considering points along the epipolar line contained within the foreground mask of $\mathcal{I}'_j$. In this manner, we have $\Omega(\mathbf{x}_{ip}, \mathbf{F}_{ij}) = \{\mathbf{x}'_{jq} \mid \mathbf{x}_{ip}\mathbf{F}_{ij}\mathbf{x}'_{jq} = 0\}$. Henceforth, we shall omit the dependence on the pre-computed camera geometry and segmentation estimates from our notation, denoting the set of candidate matches for a given pixel as $\Omega(\mathbf{x}_{ip})$. We measure the NCC w.r.t. the reference pixel $\mathbf{x}_{ip}$ using 15×15 patches along the epipolar line, and all patches with a NCC value greater than 0.8 are deemed potential correspondences. Once $\Omega(\mathbf{x}_{ip})$ is determined, its elements $\mathbf{x}'_{jq}$ are sorted in descending order of their photoconsistency value. Fig. 5.4 provides an example of our epipolar correspondence search for an image pair.

Figure 4.3: (a) Local features in reference image. (b) Corresponding points are found along the epipolar lines in the second image.

Figure 4.4: Red stars: Feature point in reference frame. Blue stars: Matched feature points in the target frame. Green circles: Points with highest NCC values. In (a), the point with the highest NCC value is actually the correct correspondence. However, in (b), the green circle is indicating the wrong match. The other candidate is the correct correspondence and should be used for triangulation.

4.2.3 Assessment and Correction Mechanism

Based on the example shown in Fig. 5.4, we propose a method to discern wrong correspondences and correct them with alternative pixel matches. The steps of our method are as follows:

4.2.3.1 Step ❶: Building Motion Tracks

The set of 2D feature points $\{\mathbf{x}_{ip}\}$ and currently selected corresponding points $\{\mathbf{x}'_{jq}\}$ are updated with optical flow motion vectors computed between neighboring frames using the approach of Brox et al. (Brox et al., 2004). Thus we have $\{\mathbf{x}_{i+1,p}\} = \{\mathbf{x}_{i,p}\} + \mathcal{O}_i$ and $\{\mathbf{x}'_{j+1,q}\} = \{\mathbf{x}_{jq}\} + \mathcal{O}'_j$. We select the video with the higher frame rate as the target sequence, which will be temporally sampled according to the frame rate ratio α among the sequences. The reference sequence will be used at its native frame rate. Hence, given a temporal window of W frames, the reference video frames and their features will be denoted, respectively, by $\mathcal{I}_i$ and $\{\mathbf{x}_{i,p}\}$, where $1 \leq i \leq W$, denotes the frame index. Accordingly, the frames and features in the target video frames will be denoted by $\mathcal{I}'_j$ and $\{\mathbf{x}'_{j+w*\alpha,q}\}$, where j corresponds to the temporal frame offset between the two sequences, and $0 \leq w < W$. The size of the temporal window must strike a balance between building informative 3D tracks for spatial analysis and maintaining the reliability of the chain of estimated dense optical flows.

The initial set of correspondence estimates $\{\mathbf{x}_{ip}\}$, $\{\mathbf{x}'_{jq}\}$ are temporally tracked through successive intra-sequence optical flow estimates, and their updated locations are then used for two-view 3D triangulation. Namely, for each point $\mathbf{x_{ip}}$ selected at frame p, we have a 3D track $\mathbf{T}_i = \{\mathbf{X}_{iw}\}$ comprised of $1 \leq w \leq W$ 3D positions determined across the temporal sample window.

4.2.3.2 Step ❷: Enforcing Local Rigidity

Local rigidity assumes a pair of nearby 3D points in the scene will maintain a constant Euclidean distance throughout our temporal observation window. Assuming a correct spatio-temporal inter-sequence registration and accurate intra-sequence optical flow estimates, deviations from this assumption are attributed to errors in the initial correspondence estimation. More specifically, tracks

having incorrect initial correspondences will present inconsistent motion patterns. Accordingly, the key component of our rigidity estimation is the scope of our locality definition. To this end, we use the appearance-based super-pixel segmentation method proposed in (Achanta et al., 2012) to define relatively compact local regions aligned with the observed edge structure. The SLIC scale parameter is adaptively set such that the total of superpixels contained within the initial segmentation mask is 30. The output of this over-segmentation of the initial frame in the reference sequence is a clustering of our 3D tracks into disjoints partitions $\{\mathcal{C}_c\}$, where $1 \leq c \leq 30$.

Having defined disjoint sets of 3D tracks, we independently evaluate the rigidity of each track cluster. We measure this property in terms of the largest consensus set of constant pairwise distances across successive frames. Although this set can be identified through exhaustive evaluation of all pairwise track distances, we instead take a sampling approach for efficiency. We iteratively select one of the tracks in $\mathcal{C}_c$ and compare the temporal consistency against all other tracks. We then store the track with the largest support within $\mathcal{C}_c$. An outline of our sampling method is presented in Algorithm 4. Our local rigidity criteria decides if two trajectories are consistent based on the accumulated temporal variation of point-wise distance of two 3D tracks over time:

$$\sum_{i=2}^{W} \Big| \|\mathbf{X}_{m,i-1} - \mathbf{X}_{n,i-1}\|_2 - \|\mathbf{X}_{m,i} - \mathbf{X}_{n,i}\|_2 \Big|, \ \mathbf{T}_n, \mathbf{T}_m \in \mathcal{C}_c \tag{4.1}$$

Once the consensus track set has been identified, all its members are considered inliers to the rigidity assumption, while all tracks not belonging to the consensus set are labeled as outliers.

4.2.3.3 Step ❸: Enforcing Structural Coherence

Local rigidity in isolation is unable to determine systematic errors caused by motion correlation among content having similar appearance. A particular challenge is the presence of poorly textured and (nearly) static scene elements, as both appearance and motion cues are ill-defined in this scenario. For example, in Fig. 4.6(a), some correspondences are located on the left leg, while the true correspondences should be on the right leg. In order to make our correspondence estimation

Algorithm 4: SAMPLING FOR LOCAL RIGIDITY TRACK CONCENSUS

Input: 3D trajectories $\mathbf{T}_i(m)$, $1 \leq m \leq |\mathcal{C}_i(c)|$
Output: Inliers trajectories set $\{\hat{\mathcal{C}}_i(c)\}$

```
iterations = 0
Ĉ_i(c) = NULL
while iterations ≤ |C_i(c)|/5 do
    C'_i(c) = NULL
    Draw a random trajectories T_i(m)
    for k ∈ [1, ||C_i(c)||] do
        decide if T_i(m) and T_i(k) are consistent
        if consistent then
            add k into C'_i(c); if C'_i(c) = C_i(c) then
                return
    if C'_i(c) ≥ Ĉ_i(c) then
        Ĉ_i(c) = C'_i(c)
```

more robust, we further enforce the assumption of geometric coherence within local structure estimates deemed to be locally rigid. We consider two types of non-local coherence violations:

1. **Track-Bundle Consistency** 3D Tracks emanating from a common compact image region should also correspond to a compact set of 3D trajectories. We observe that small subsets of inlier (*i.e.* mutually rigid) 3D tracks can be spatially disjoint from the remaining tracks belonging to the same initial cluster (Fig. 4.6(b)). We measure this behavior by analyzing the results of individual pairwise 3D point sampling used in step ❷ for rigidity consensus estimation. We aggregate all the sampled $N = \|\mathcal{C}_c\|$ pairwise rigid distances of the inlier set into a single vector $S_c \in \mathbf{R}^N$ and sort the elements by increasing distance. We then scan for an inflection point depicting the largest pairwise deviation among successive bins in S_c and threshold on both the relative magnitude and the percentile of the inflection point location within the histogram. Inflection points found in the top and bottom 10% quantiles are to be discarded. If an inflection point is found in the histogram, the corresponding distance value is used as a distance consistency threshold. Tracks exhibiting an average distance to other tracks greater than the consistency threshold are removed from the inlier set $\mathcal{C}_c$. Fig. 4.5 illustrates

the behavior of the distance histogram for different 3D track bundle scenarios. The above framework operates under the assumption that locally inconsistent tracks represent a small fraction of a cluster's track bundle.

2. **Inter-Cluster Consistency** The scenario where the majority (or all) of the mutually rigid tracks within a cluster are structured outliers is extremely uncommon but cannot be identified through track-bundle consistency (Fig. 4.6(c)). To address this challenge, we impose thresholds on the spatial divergence between the average 3D position of a given track and a fixed global 3D reference representative of the estimated structure across the entire image. We define this reference to be the 3D centroid of the 3D tracks of all other clusters. This approach is aimed at identifying gross outliers within the context of a single foreground dynamic object and is to be considered a special-purpose noise filtering technique. In practice, 3D tracks away from the moving body are identified and singled out as correspondence outliers.

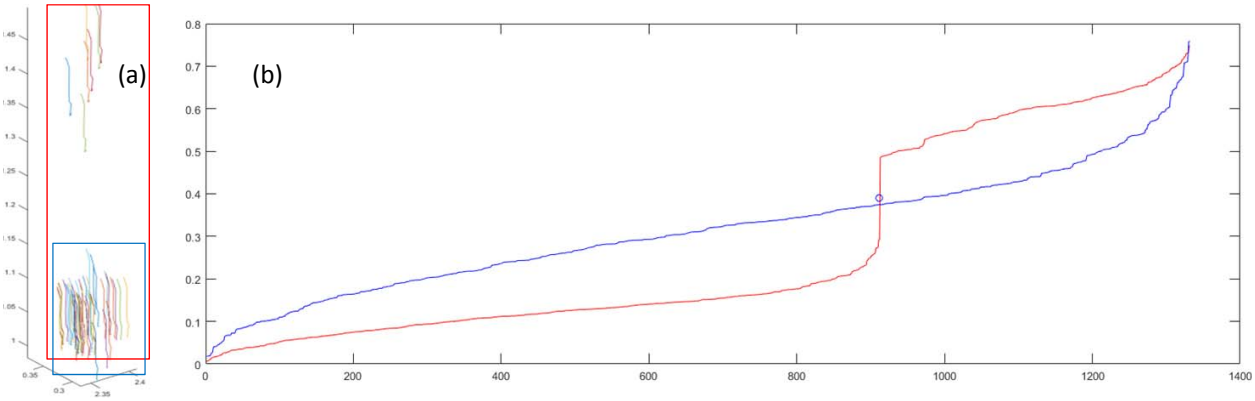


Figure 4.5: In (a), trajectories from wrong correspondences deviate away from the inlier trajectories (outlined in blue). (b) The sorted pairwise distance array of all inliers has no abrupt gradient in the middle, sorted pairwise distance array of all trajectories will have those cutting edge when outlier trajectories are present.

4.2.3.4 Step ❹: Track Correction

The set of 3D tracks determined to be outliers by our preceding validation steps are assumed to occur due to an outlier feature correspondence $\mathbf{x}_{ip} \leftrightarrow \mathbf{x}_{jq}$. Accordingly, to correct this erroneous initial assignment, we revisit the sorted set of correspondence candidates $\Omega(\mathbf{x}_{ip})$ lying on the

Figure 4.6: Corresponding points in image pairs. Red dots (crosses): Feature (inlier) points within one super-pixel in the reference frame. Blue dots (crosses): Correspondence (inlier) points found in the target frame. In (a), outliers on the left leg are detected because they located in different rigid parts. In (b), outliers on the right waist are removed because they are far away from majority of other trajectories. In (c), correct correspondences are the minority (there might be repetitive correspondences in the target frame). The wrong correspondences are removed by the depth constraints.

epipolar line. We will replace the initial assignment with the next-most photo-consistent element of $\Omega(\mathbf{x}_{ip})$ and evaluate the local rigidity of the updated 3D track across the temporal sampling window. We can now modify the correspondence to regenerate the 3D track (i.e. step ❶) and re-run our original rigidity sampling procedure (i.e. step ❷) over the entire cluster to account for possible changes to the consensus set. In practice, it is more efficient to verify the rigidity of each updated track against a small sample of the current consensus/inlier (i.e. locally rigid) set of tracks. The process is repeated until each original feature has either 1) been determined to be an inlier or 2) exhausted the candidate set.

4.2.4 Applications to Stream Sequencing and 3D Reconstruction

We have described a framework to determine and enhance the spatio-temporal consistency of two-view pixel correspondences across a time window. Our image-wide active correspondence correction framework effectively maximizes the number of locally consistent 3D tracks. The relevance of this functionality lies in the insight that, given an unknown temporal offset between

two spatially overlapping video sequences, scanning a short video segment from one sequence over the entirety of the other sequence can be used to identify the temporal offset between those sequences. Figure 4.7(b) shows the average correspondences with different offsets (computed over 50 consecutive frames from one of our datasets), we can see our method obtains the highest value on the 0 offset point, which means accurate alignment. The criteria to determine alignment is, intuitively, the offset results in maximal locally rigid (e.g. inlier) 3D tracks. Conversely, determining a robust and dense set of inter-sequence correspondences, directly provides the observed 3D geometry given knowledge of the imaging geometry. A straightforward way to generate depthmaps under our framework is to perform bi-linear 2D interpolation on each sequence frame for all inlier 3D tracks. Figure 4.7(a), illustrates the depthmap generated by our approach without any post-processing corrections.

Figure 4.7: (a) show depth map generated from raw correspondences (Left) and the corrected correspondences (Right). (b)Average correspondences with different offsets(red curve), the green boundary should the plus minus standard deviation.

4.3 Experiments

Experimental Setup. All reported experiments considered a temporal window size of $W = 6$, and unless stated otherwise, the initial correspondence set is comprised of all putative pixel

Name	# Video frames	GT 3D Points	Synchronized	Moving Cameras	Outdoor Scene
ETH	200	No	Yes	Yes	Yes
CMU	160	Yes	Yes	No	No
UNC	150	No	No	Yes	Yes

Table 4.1: Composition of our datasets.

correspondences along the epipolar line with an NCC value above 0.8. Epipolar constraints don't hold for unsynchronized frames (Albl et al., 2017), searching along epipolar lines between two views generate more feasible correspondences when frames are closely aligned. We evaluated our method on three datasets: the ETH juggler (Ballan et al., 2010b), the CMU bat (Joo et al., 2014), and the UNC juggler (Zheng et al., 2015a). For the ETH dataset (6 cameras) and the UNC dataset (4 cameras), we select the pair of cameras having the smallest baseline. For the CMU dataset, we select two neighboring cameras facing the front of the players. The CMU dataset provides reconstructed 3D points which are used as ground truth to evaluate the accuracy of our estimated 3D triangulations and depth maps. The UNC dataset is not synchronized; hence, we adopt the synchronized result from (Zheng et al., 2015a) as sequencing ground truth. Details for each of the three considered datasets are provided in Table 5.1.

Synchronization Evaluation. In order to evaluate synchronization accuracy, we carried out experiments with temporal offsets between the reference and the target sequence in the range of $[-15, 15]$ with step size 3. We considered the following scenarios: (1) common frame with varying pixel sampling density, and (2) one sequence having double the frame rate of the other. Fig. 4.8(a-c) shows respectively the results for ETH, UNC, and CMU datasets under varying pixel densities. By controlling the density of considered pixels within each local neighborhood (i.e. SLIC-based superpixel segmentation) we can directly control the computational burden of our sampling rigidity framework. Alternatively, we may perform KLT-based feature selection. For efficiency reasons, we simply select in these experiments a fixed number of random pixels as features for correspondence analysis within a local neighborhood $\mathcal{C}_c$. We experimented with pixel densities of 2%, 2.5%, and 3.3%. The results illustrated in Fig. 4.8(a-c) highlight the positive effect of increased pixel densities towards synchronization accuracy. We observe that, in addition to segments exhibiting reduced

motion or poorly textured content, repetitive motion was a source of synchronization ambiguity leading to potential errors. Fig. 4.8(d) shows the alignment results with the target sequence at twice the frame rate of reference sequence. We use 3.3%, 1.25%, and 5% sampling density, and the results are very close to the equal-frame-rate test, with a decrease in average accuracy of 9%. In Fig. 4.8(e) we show more synchronization results with variable sampling rates for video streams.

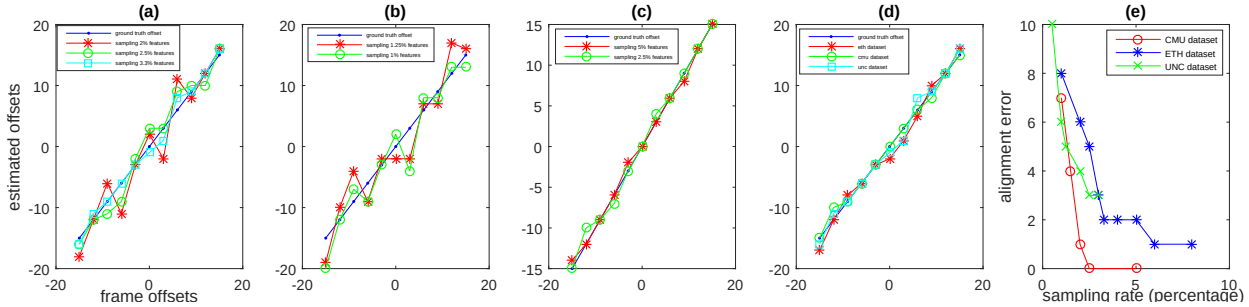
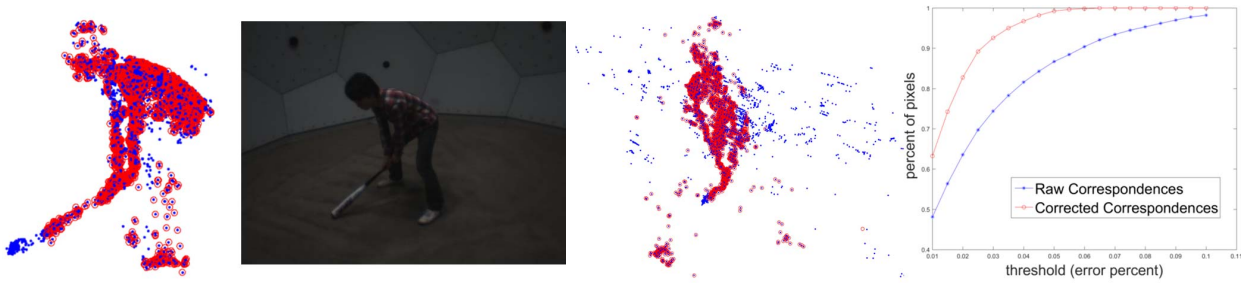


Figure 4.8: Accuracy of our synchronization estimation across different datasets scenarios.

Dense Modeling Evaluation. We explored the effectiveness of our correspondence correction functionality when applied for 3D reconstruction. Given that the CMU dataset provides groundtruth 3D structure values, we include the reconstruction error of our 3D reconstructions. In Fig. 4.9(a)(c), we show the front and back view of the estimated 3D points. We observe our method's ability to effectively remove outlier 3D structure. In Fig. 4.9(d), we quantitatively evaluate the accuracy of our depth map, in terms of the percentage of pixels falling within variable accuracy thresholds. Fig. 4.10 shows some qualitative comparisons of our interpolated depth maps obtained from correspondence-corrected 3D points against the depthmaps interpolated from raw correspondence output (e.g. in the absence of corrections). Since (Mustafa et al., 2015) does not consider motion consistency nor temporal alignment, their depth maps correspond to "raw correspondences" in our method given synchronized input frames.

4.4 Discussion and Conclusion

We have presented a local spatio-temporal correspondence verification and correction method, and used it to develop a bottom-up solution for video synchronization and dense dynamic modeling.



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

Figure 4.9: Results of corrected point cloud on the CMU dataset. Left: Blue 3D points depict the originally reconstructed 3D points from initial correspondences, while red points denote the 3D points obtained through corrected correspondences. Left middle: Corresponding reference image. Right center: A side view of the same structure. Right: Accuracy for both original and corrected point sets.

The underlying assumption of local geometric consistency as a guide for spatio-temporal overlap has been proven to be informative across an expanded spatio-temporal scope.

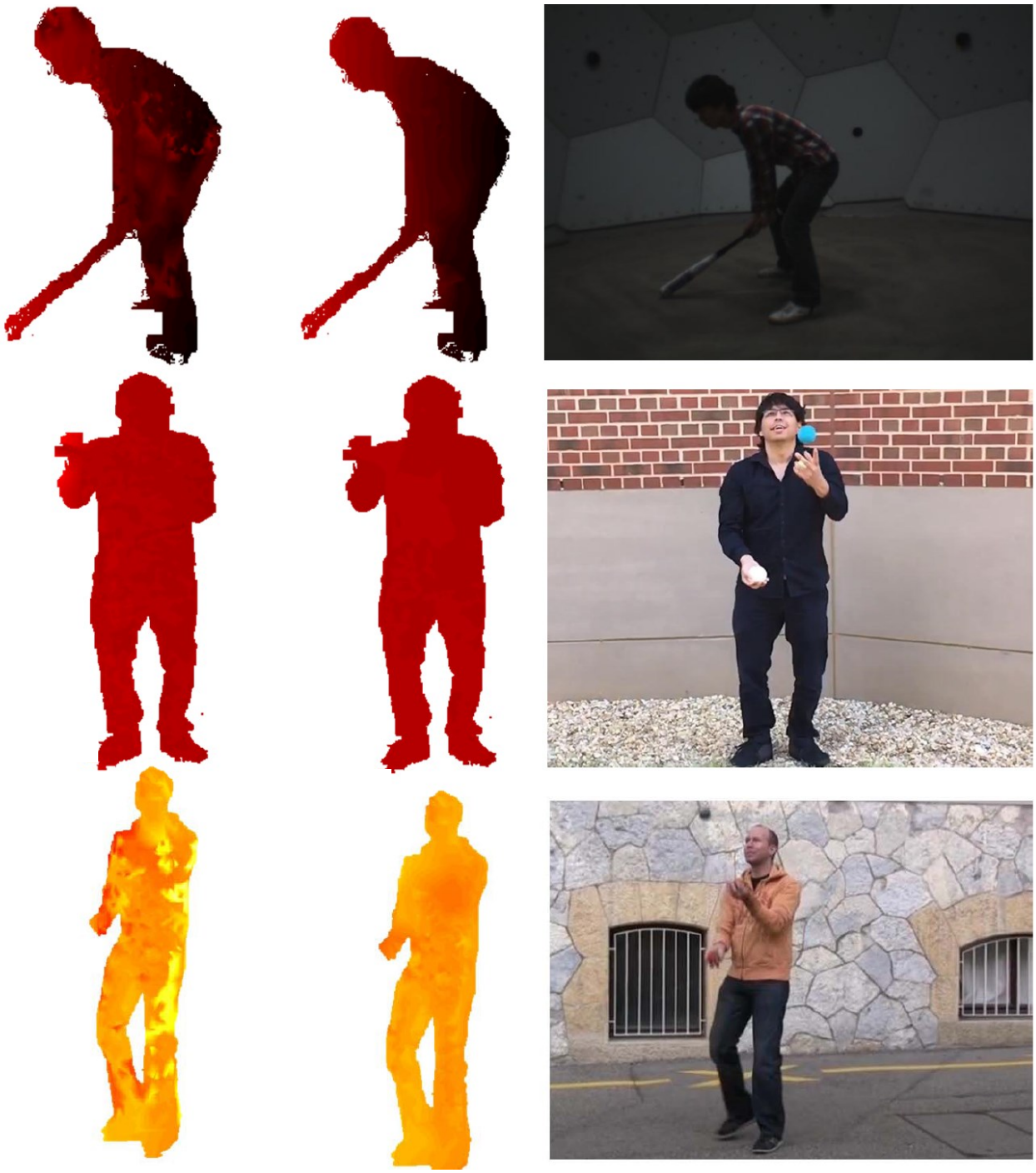


Figure 4.10: Qualitative results illustrating the effectiveness of our correspondence correction functionality.

CHAPTER 5: SYNTHESIZING ILLUMINATION MOSAICS FROM INTERNET PHOTO-COLLECTIONS

5.1 Introduction

Internet photo-collections can provide a vast sample of the space of possible viewpoint and appearance configurations available for a given scene. This chapter addresses the organization and characterization of this image space by exploring the link between time-lapse photography and crowd-sourced imagery. Time-lapse photography strives to depict the evolution of a given scene as observed under varying image capture conditions. While the aggregation of a sequence of images into a video may be the most straightforward visualization for time-lapse photography, the integration of multiple images in the form of a mosaic provides a descriptive 2D representation of the observed scene's temporal variability. We denote these time-lapse mosaics as *illumination mosaics* and show an example in Fig. 5.1.

The problem of mosaic construction can be abstracted as a three-stage process of image registration, alignment, and aggregation. However, the representation of the appearance dynamics introduces the qualitative challenge of producing an aggregate mosaic that is both coherent with the original scene content and descriptive of the fine-scale appearance variations across time. The associated technical challenges addressed in this work are 1) identify within an unorganized image set an image sequence depicting the desired content appearance transition and 2) construct an illumination mosaic that accurately depicts the observed appearance variability while mitigating scene artifacts due to changes in scene content and capture parameters.

We address these challenges by exploring the spectrum of capture variability available in Internet photo-collections and propose a novel framework to obtain illumination mosaics. We briefly summarize the functionality of our processing pipeline. The input data to our framework are

Figure 5.1: Example time-lapse image of the Coliseum, the top image is automatically generated by our method, and the bottom is manually made by a photographer (courtesy of Richard Silver).

a reference image depicting the desired image composition to be used to generate the illumination mosaic and a crowd-sourced image collection of the scene of interest. We initially use semantically-aware global image features characterizing an imaged scene's composition and ambient illumination properties in order to determine the scope of the variation to be represented in the mosaic. Then, a limited connected graph is built based on image similarities, from which we find a smooth path between two nodes, defining an ordered set of images to be used for mosaicing. Our subsequent image alignment and stitching leverages 2D warping, segmentation, and color mappings to achieve smooth image transition while mitigating scene aberrations. We demonstrate our method on several landmark datasets, and show both qualitative and quantitative results.

5.2 Illumination Mosaic Generation

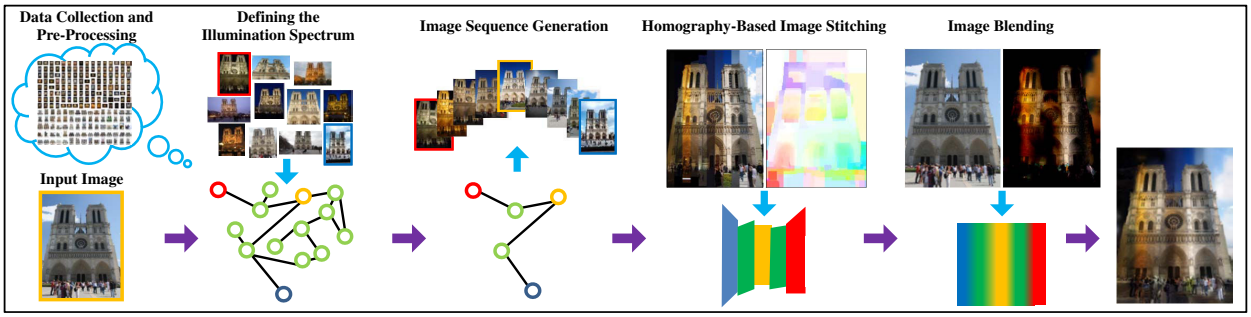


Figure 5.2: Framework of our method. Given an input image I, our method determines an appearance neighborhood $\mathcal{N}_{GIST}(I)$ within a photo collection. We identify two extremum elements of $I^- \in \mathcal{N}_{GIST}(I)$ and $I^+\mathcal{N}_{GIST}(I)$ to determine a path within an appearance similarity graph, which corresponds to image sequence used for mosaic integration. We perform robust homography-based region warping to aggregate a mosaic. Finally, we transfer color from the mosaic into our reference image.

In order to depict the illumination spectrum of a scene, our method relies on building discrete representations of the image appearance space through connectivity graphs defined over a pairwise image distance function. To generate illumination mosaics, we want to select an image sequence which 1) shares similar spatial composition, 2) features a smooth color transition between the images, and 3) conveys a large variety of scene appearances. We now detail our proposed framework

for identifying the appearance variability in a photo collection, and subsequently using it to build illumination mosaics. Fig. 5.2 shows an overview of our pipeline.

5.2.1 Data Collection and Pre-Processing

To obtain the image data for different landmarks, we first perform a keyword-based query to the Flickr photo sharing website. In order to remove unrelated images, we employ the iconic image selection pipeline proposed in (Frahm et al., 2010). We perform GIST-based((Oliva, 2005)) image clustering and discard images that cannot establish a pairwise epipolar geometry to the cluster center. We then perform K-means clustering enforcing an approximate average cluster size of 50 images. Given that all non-discarded images can be registered to the cluster center, it is possible to estimate a local 3D model of the scene. However, for efficiency purposes, we do not perform full dataset geometric verification, but instead rely on pairwise image registration to determine 2D image alignments.

5.2.2 Defining the Illumination Spectrum

The composition of our illumination mosaics requires us to specify both the desired spatial composition of the image output and the range of appearance variability to be depicted. We take as input (from the user) a reference image I that will define the spatial layout/composition of our output illumination mosaic and will be used to define subsequent image alignment and warping operations. Next, we identify, within our registered image set, elements that define the scope of our displayed appearance variation. We select a local appearance neighborhood to the reference image, which is comprised of the nearest K=300 images in terms of the Euclidean distance of their corresponding GIST descriptors. That is, we compute the GIST descriptor for the input reference, and by leveraging the pre-computed GIST descriptors for our registered dataset, we determine an image set $\mathcal{N}_{GIST}(I)$ of its K nearest neighbors. The motivation for initially focusing on a reduced local neighborhood is to ensure spatial content similarity among images, which will facilitate subsequent image alignment and warping.

In order to exploit the diversity of image capture characteristics found in a crowd-sourced photo collection we need to identify image measurements that are discriminative w.r.t. the variations we want to portray in our mosaics. We focus on a specific type of global appearance variations: the transitions between dark and bright ambiance. To enable this characterization we leverage image statistics of disjoint semantic elements within a scene to define an aggregate scene descriptor. More specifically, we perform foreground and sky segmentation on the input image and compute histogram statistics for each of the disjoint image segments.

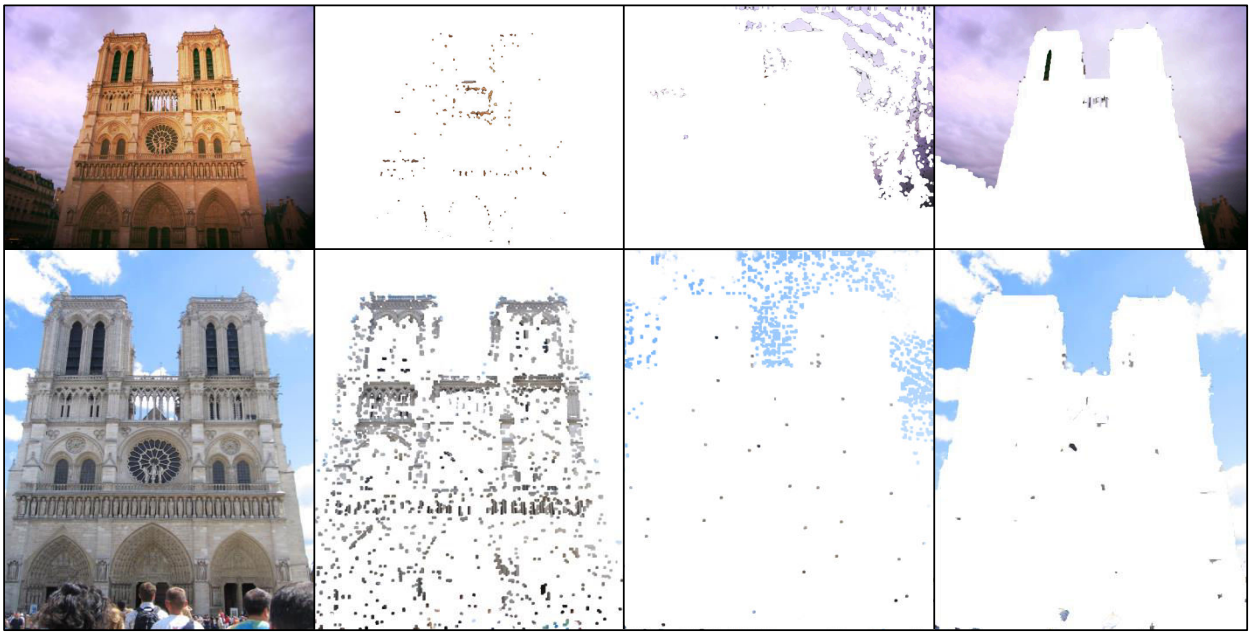
Sky Segmentation. Empirically we found that using the sky detector proposed in (Hoiem et al., 2005) to extract the sky region provides unreliable results for images captured at night.

For each image we estimate an homography-based warp to its nearest GIST-neighbor. We then compute local NCC for the two images, where local patches with NCC values larger than 0.5 will be deemed to belong to foreground buildings, and patches with NCC values less than 0.2 are labeled as background. The intuition is that static structure will have consistent NCC even in different illuminations while sky regions and transient objects will not. Graphcut is adopted to generate a more complete segmentation for the building and sky (shown in Fig. 5.3).

Quantifying Image Intensity. For the pixels contained in the sky segment we compute a 100-bin intensity histogram $\mathcal{H}_b$ of the blue color channel. We compute the intensity values (i.e. histograms bins) corresponding to the top 5 frequencies and select their median as our intensity measure for that image, given that image histogram will usually have multiple peaks. We choose images I^+ and I^- having the highest and lowest intensity values within $\mathcal{N}_{GIST}(I)$ as the two respective extremes of our illumination spectrum.

5.2.3 Image Sequence Generation

The goal of this step is to find an image sequence that depicts the gradual variation between the previously selected pair of images, I^- and I^+, which define the scope of our output illumination spectrum. We build this path by determining and concatenating an image sequence $I^- \rightarrow I$ and an image sequence $I \rightarrow I^+$, where all the aforementioned images are elements of our registered



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

Figure 5.3: Sky/building segmentation. (a) Original images, (b) Foreground mask, (c) Background mask, (d) Sky segmentation.

camera set. Henceforth, we will consider the $I^- \rightarrow I$ transition, but it is to be understood that the same steps apply to the second half of the image transition sequence.

Aggregated Image Appearance Descriptor. We combine a global image GIST feature descriptor to capture the image composition, a color histogram to represent the sky color, and a histogram of the dark channel prior image to choose photos that contain well-illuminated images. We restrict our color histogram to sky regions to account for landmarks which may be arbitrarily illuminated at night. We use all three color channels to enable more fine-grained discrimination of ambient illumination among subsequent images. These three features are normalized and concatenated to form a global image feature representation.

Image Similarity Graph. Based on our global image descriptor we define a discrete representation of our appearance space based on image pairwise similarity. We incrementally build a graph where each image is treated as a node, similar to (Wang et al., 2006), we use both k-rule and ϵ-rule to construct a neighborhood graph. The edge weights connecting two nodes are computed by L^2 distance of image features. To find a balance between path descriptiveness and compactness, we

iteratively augment the local image neighborhoods around both I^- and I^+ until we attain a single connected component from which to attain a minimum-length path between the nodes corresponding to I^- and I^+. Moreover, at each iteration k(which starts from 1), each image in the registered camera set is only connected to its k nearest neighbors. Outliers in the graph are reduced using the ϵ-rule, which removes edge connections that have weights (i.e. descriptor distance) more than $\epsilon = 1.3d_p$, where d_p is the average edge distance in the graph. Once a k-connected graph is defined at each iteration, we search for a connecting path between I^-, I and I, I^+ by using Dijkstra's method.

5.2.4 Homography-Based Image Stitching

Our scene warping is a two-stage process that leverages pairwise homography transfers between elements of our image sequence. First, we compute a homography warping $\mathbf{H}_j$ between every image I_j in the generated sequence and the input image I, which transfers the local surface appearance characteristics under a local planarity approximation, i.e. $I_j^{'} = \mathbf{H}_j(I_j)$. Second, we apply dense SIFT Flow (Liu et al., 2009b) warping to the homography-warped image to compensate for fine-scale scene parallax not modeled by the local planarity assumption, i.e. $I_j^{''} = \mathbf{S}(\mathbf{H}_j(I_j))$.

Robust Homography Chains. If the homography matrix H_{ba} aligns I_b to I_a, according to the chain rule, the homography matrix that aligns a third image I_c through I_b to I_a is $H_{ca} = H_{ba} \cdot H_{cb}$. Likewise, if we have N images and want to register the n_{th} image to the first one, the homography matrix could be written as $H_{1,N} = \Pi_{i=1}^{N} H_{i,i+1}$.

However, in our experiments we found computing feature-based homographies directly between neighboring images is unreliable, especially for images captured at night. Since we only extract color features from the sky, the colors on the building facades between neighboring images can be very different (i.e. in Fig. 5.4). While simplifying image alignment to a homography model provides a more inclusive geometry fitting framework (i.e. less constraints) we observed that reliably building an homography chain across the entire input sequence was still elusive. As mitigation we explored the use of bridge images to attain pairwise homography estimates through transitivity Fig. 5.5(c).

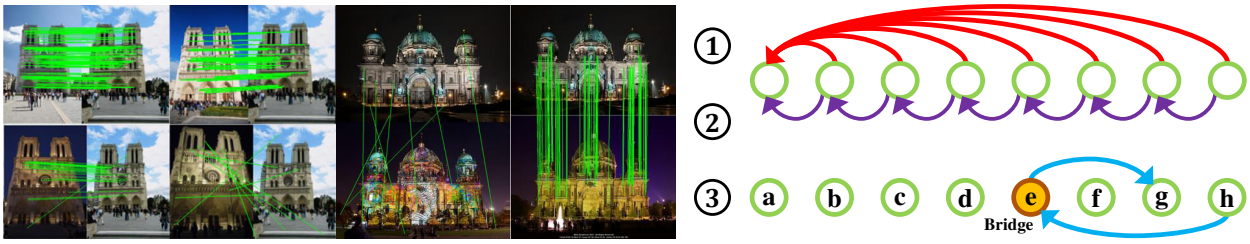


Figure 5.4: Motivation for robust homography chains. (left) The reliability of direct pairwise homography estimation of an entire image sequence to a single reference image is not uniform across the sequence. Moreover, neighboring images may exhibit drastic appearance variation (especially at night), hindering direct homography chains. Green lines depict RANSAC inlier matches. (right) Schematic representation of (1) direct pairwise estimation, (2) direct homography chains , and (3) our proposed bridge-based homography estimation.

We measure the confidence for our homography estimation based on the output of the pairwise RANSAC estimation process. We measure the number of inlier matches $m_{i,j}$ between images I_i and I_j and the image area $a_{i,j}$ of the convex hull of the attained inlier set normalized by total image size. Note that $m_{i,j}$ is symmetric while $a_{i,j}$ is not. Using these values we define a pairwise homography confidence score between images I_i and I_j as

$$C_{i,j} = m_{i,j} \cdot (a_{i,j} + a_{j,i}) \tag{5.1}$$

and use it to search for an alternative intermediate *bridge* image between every adjacent image pair in the sequence. The motivation is to omit unreliable adjacent estimates through the transitivity of a third image. Given an image I_i, the bridge image I_k is selected as the non-adjacent image with highest confidence path to the adjacent image. The bridge I_k image will be used to join two successive images I_i and I_{i+1} whenever the following condition is satisfied

$$C_{i,i+1} < \max_{k \neq i}(r_{i,k} \cdot C_{i,k} + r_{i+1,k} \cdot C_{k,i+1})/2 \tag{5.2}$$

in which $r_{i,k}$ is the area ratio of image i and k, and this is used to regularize cases when image k has higher resolution than image i. Similarly, we use a confidence threshold to eliminate images in the sequence that do not attain reliable homography estimations, and reconnect the sequence through the same bridge image search process as before.

Stitching & Refinement. Upon establishing a robust local homography chain across the entire sequence $\{I_j\}$, we warp all the images into the reference image I. Next, we apply dense SIFT Flow warping (Liu et al., 2009b) to the homography-warped images to compensate for fine-scale scene parallax not modeled by the local planarity assumption. Finally, we form a mosaic by sequentially aggregating equal-sized vertical stripes from each of the images in the sequence to form a single, combined image. It is constructed such that the first (leftmost) vertical stripe is obtained from the first image in the sequence, the second stripe from the second image, and so forth. In this manner, the mosaic depicts a single, recognizable view of a scene, but is composed of stripes taken from different images (see Fig. 5.5(b)). The length of the output sequence is data-dependent as it is a function of both the size and composition of the image set used to determine our illumination spectrum. However, replacing Dijkstra shortest path search in our implementation with Yen's k-shortest path algorithm (Yen, 1970) would enable the user to set sequence length a priori.

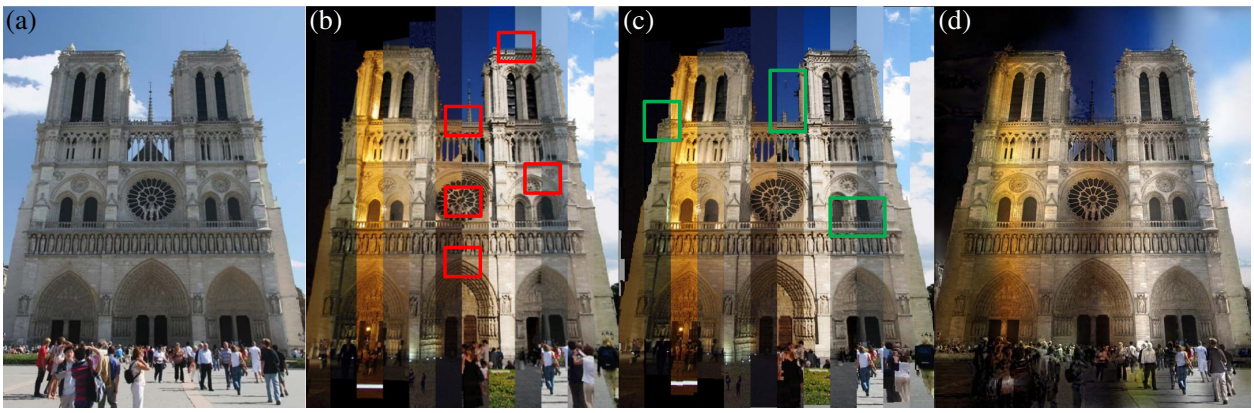


Figure 5.5: Mitigation of mosaicing artifacts. (a) Input reference image (b) Homography-based image stitching (red rectangles highlight alignment problems). (c) SIFT-flow dense registration refinement partially resolves alignment issues, at the expense of small-scale structure aberrations (highlighted green boxes) (d) Output image after transfering color from the mosaic to the reference image.

5.2.5 Image Blending

We note that the generated stitched mosaic M may have strong color and structural artifacts among adjacent mosaic segments, see Fig. 5.5(b). The reason for these artifacts include: 1) Inconsistent foreground objects, i.e. pedestrians, cars, or other transient objects. These transient

objects cannot find correspondences in other images and will cause registration artifacts. 2) Uneven resolutions for different stripes. Our generated image sequence does not enforce a common resolution for all images. When warping low-resolution images to high-resolution images, up-sampling will introduce blur artifacts. 3) Artifacts caused by dense registration. Although SIFT Flow generally works well for aligning static structures, sometimes it fails in texture-less regions (such as windows and tower top). Also, if the appearance or structure of the foreground elements changes dramatically, dense registration may introduce artifacts.

Color Transfer. In order to keep the fine-grained details of the mosaic, while at the same time conveying a large range of scene appearance, we decide to transfer the color from the image mosaic M to the reference image I. Shih *et al*. (Shih et al., 2013) propose a locally linear model learned from time-lapse video, allowing them to synthesize new color data while retaining image details. Moreover, for the image pair (M,I) we want to estimate local transformations which characterize the color variations between two images. The locally linear model proposed by (Shih et al., 2013) is used to relate the color of pixels in M to the color of pixels in I. We denote the patch centered on pixel p_k in the match image by $\mathbf{v}_k(M)$, and $\mathbf{v}_k(I)$ is the corresponding patch in the target image. Both are represented as $3\times N$ matrices in RGB color space; using patches of $N = 5\times 5$ pixels. The local linear transform applied to patch k is represented by a 3×3 matrix $\mathbf{A}_k$, and is estimated with a least-squares minimization:

$$\arg\min_{\mathbf{A}_k} \|\mathbf{v}_k(I) - \mathbf{A}_k\mathbf{v}_k(M)\|_F^2 + \gamma\|\mathbf{A}_k - \mathbf{G}\|_F^2 \tag{5.3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The second term regularizes A_k with a global linear matrix G estimated on the entire image (using a small weight $\gamma = 0.008$ in all tests). We obtain the optimal transform A_k in closed form:

$$\mathbf{A}_k = \left(\mathbf{v}_k(I)\mathbf{v}_k(M)^T + \gamma\mathbf{G}\right)\left(\mathbf{v}_k(M)\mathbf{v}_k(M)^T + \gamma\mathbf{I}_3\right)^{-1} \tag{5.4}$$

According to (Shih et al., 2013), the estimated local affine matrices $\mathbf{A}_k$ explain the color variations between the generated mosaics M and input image I. To generate an image with the same structure as the input and exhibiting the same color change as seen in the time-lapse mosaic, the output image $\bar{M}$ should be locally affine to I, and explained by the same affine models $\mathbf{A}_k$. Straight-forward solution compute each affine model $\mathbf{A}_k$ as a regression between the k-th patch of $\mathbf{v}_k(M)$ and $\mathbf{v}_k(I)$, then independently apply $\mathbf{A}_k$ to the patch of I for each k. With this method, the boundary between any two patches of $\bar{M}$ would not be locally affine with respect to I, and would make $\bar{M}$ have a different structure from I. This problem is instead formulated as a least-squares optimization that seeks local affinity *everywhere* between M and I.

Since the mosaic and reference image are already aligned, there is no need to compute a correspondence map between them. We adopt the linear equation system proposed in (Laffont et al., 2014) to solve the color transfer problem. Fig. 5.5(d) shows the color transfer results, compared to Fig. 5.5(b), and the artifacts highlighted in green are gone, and there is no detail loss from the reference image.

Local Stripe Reordering (optional). The image sequence is generated through global image appearance descriptors. However, there can be local appearance variations in the images, resulting in color inconsistencies among adjacent elements within the mosaic. Examples include clouds, partial foreground occlusions, or reduced overlap with the reference image. Addressing this contingency within the image sequencing step of our mosaic generation would entail an explosive growth of our image similarity graph, as each stripe needs to be connected to every other stripe in all other images within the appearance neighborhood. Accordingly, our approach is to resolve this issue through a post-processing step. We propose a method to locally reorder the stripes in the final mosaic to make the sky transitions look more natural by only reordering the contents of the sky regions. To this end, we leverage our existing sky segmentation and extract a sky-only intensity color histogram for each stripe. We sort the stripes by the median of their top 5 frequencies in the intensity histogram. We then transfer color from each image in the new sequence into the sky regions of the output mosaic. We repeat the process until the sequence convergences.

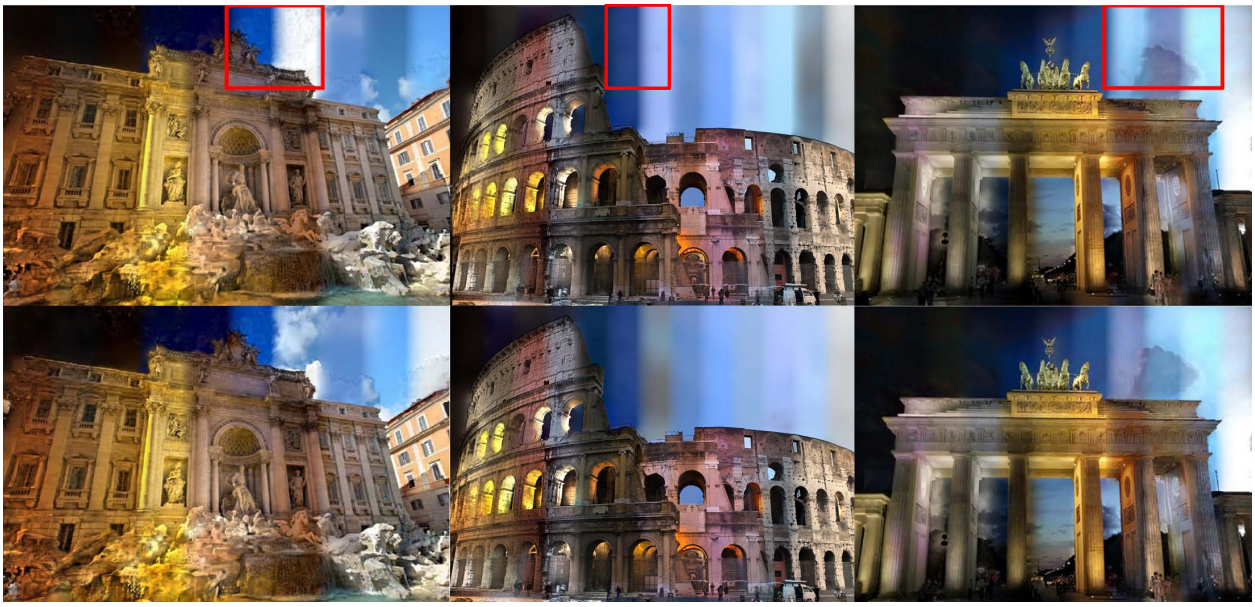


Figure 5.6: Sky reordering. Top: mosaics before reordering, red rectangles highlight the inconsistent stripes. Bottom: reordered mosaics, the sky appearance inconsistencies are mitigated.

5.3 Experiments

Data Acquisition. We downloaded 10 online datasets from Flickr, and the statistics of our system's data associations are presented in Table 5.1. We categorize images with average intensity of their sky regions below 100 as night images (intensity value range from 0 to 255).

Homography Chain Evaluation. To evaluate the effectiveness of our bridge-based image stitching method, we design a metric to quantitatively compare alternative stitching methods. We first compute an edge map for the reference image and all warped images used to form the output mosaic, using Canny edge detection (Canny, 1986). Using these edge-maps, we then compute the average per-pixel NCC values between each stripe in the reference image and its corresponding warped region in the mosaic using a 5×5 aggregation window. To focus on the inter-stripe alignment accuracy, we restrict our evaluation to edge pixels found in the boundaries between mosaic stripe elements. We compare our image stitching method (Bridge + SIFT Flow) with three methods: (1) Align image to neighbor, (2) Align with bridge, and (3) Align with SIFT Flow. From Table 5.2, we can see that most datasets benefit from bridge-based image stitching compared with the "Align to next" strategy. Moreover, many of the"Align to next" outputs suffer from incorrect homography

Name	# Downloaded	# Clustered # (night / day)	Stripe Reordering
Notre Dame Cathedral	60291	3615 / 5260	No
Berliner Dom	51892	2197 / 3986	Yes
Brandenburg Gate	63796	2671 / 5198	Yes
Mount Rushmore	53612	583 / 2423	Yes
Coliseum, Rome	49220	910 / 1027	Yes
Trevi Fountain	94370	1612 / 3689	Yes
Manarola	54535	1023 / 4058	Yes
Potala Palace	25039	450 / 1996	Yes
Tiananmen Square	70384	658 / 3142	Yes
St. Peter's Cathedral	91060	2557 / 3297	Yes

Table 5.1: Composition of our downloaded image datasets. The number of clustered images corresponds to images that were able to register through geometric verification to their cluster center. In most cases (˜90%), stripe reordering is applied to generate smoother appearance transition (For Notre Dame dataset, stripe reordering didn't change its original sequence).

estimates (due to highly different illumination conditions) which render severely distorted mosaics. Note that using robust homography chains in conjunction with dense SIFT Flow refinement provides enhanced accuracy when compared to either of them in isolation.

Color Transfer Results. We compare with three methods to create illumination mosaics: two previous works (Reinhard et al., 2001)(a), (Shih et al., 2013)(b), and our method without bridge homography connections(c). Method (a) adopts the same image sequence used in our method as input, and transfers color from all images to the reference image in the sequence using the approach proposed in (Reinhard et al., 2001). Method (b) implements the method in (Shih et al., 2013) using the same reference image and the video datasets created by the original paper as input. We randomly select frames from all videos, extract their GIST and color features, compute the nearest neighbors w.r.t. the input image, and use that video as the input time-lapse source. We then manually select a temporal sequence from the video and transfer the color with the pipeline proposed in (Shih et al., 2013). Method (c) also uses the same image sequence as input. We warp the sequence using only SIFT Flow, and transfer the color using the locally affine method proposed in (Shih et al., 2013). The comparative results in Fig. 5.7 illustrate both the wide range of appearance variation achieved by our approach as well as the recovered fine-scale chromatic and scene structure details.

Dataset	Align to next	Bridge	SIFT Flow	Align to next + SIFT Flow	Bridge + SIFT Flow
Notre Dame	0.4179	0.4152	0.3509	0.4387	**0.6152**
Berliner Dom	0.3634	0.4539	0.3398	0.3812	**0.5529**
Trevi Fountain I	0.3967	0.4159	0.4123	0.6141	**0.6503**
Trevi Fountain II	0.4420	0.4292	0.3889	**0.6020**	0.5752
Forbidden City	0.3595	0.3969	0.3554	**0.4513**	0.4431
Mount Rushmore	0.4223	0.4563	0.2973	0.5257	**0.5708**
Brandenburg Gate	0.4095	0.5352	0.4130	0.4791	**0.5875**
Manarola	0.3415	0.4105	0.3306	0.4776	**0.5429**
Potala Palace	0.4251	0.5254	0.3875	0.5025	**0.5683**
Coliseum, Rome I	0.4085	0.4253	0.3169	0.6219	**0.6873**
Coliseum, Rome II	0.3468	0.4416	0.4152	0.5758	**0.7048**

Table 5.2: For each dataset, we create three sequences with different reference images and compute our predefined values. For Trevi Fountain I&II and Coliseum, Rome I&II, they differ in the viewing angle. Bold-font numbers highlight the best matching score, eight out of the ten datasets achieve the best results using our method. For the other two datasets, we are very close to the best scores.

Moreover, from Fig. 5.7 we can see that method (a) cannot generate a smooth color transition sequence. Method (b) can generate a smooth color transition, however a lack of drastic color change makes it surreal. Better results may be obtained if we enlarge the time-lapse video dataset and include more scenes. While method (c) generates reasonable color transitions overall, it suffers from severe local artifacts (i.e. the sky at night, blue regions on the building, etc.). Our method (d) can both keep the fine-grained details in the image and obtain smooth sky color transitions.

Qualitative Results. The generality and robustness of our approach is highlighted by applying our method to several image collections as shown in Fig. 5.8. Challenging appearance variations, such as drastic texture appearance changes (i.e. Berliner Dom), are addressed by leveraging the spatial composition similarity among images. Note that while our method relies on local homography-based structure transfer, deviations from non-planar scene structure (i.e. Mt. Rushmore) are mitigated by SIFT Flow refinement.

Quantitative Discussions. In the experiments, we observe a change in the color and smoothness in the color-transferred image by tuning the regularization factor γ. To make a convincing conclusion how γ influences the quality of the final images, we devise two metrics to quantitatively evaluate smoothness and color change. The first is a *smoothness ratio*, where we compute a sum

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

Figure 5.7: Comparative results for baseline color transfer methods. Column (d) is generated by our color transfer method, refer to the text for specification of baselines.

of the image's horizontal gradients near the stripe boundaries and denote it as V_s. For the original mosaic M, this value is the largest, since no smoothing is applied. We then compute the smoothness ratio for every image as $V_s^{\gamma_i}/V_s^M$, where $V_s^{\gamma_i}$ is the smoothness of the γ_i-modified image, and V_s^M is the smoothness of the original mosaic. To describe a change in the color, we measure *color deviation* as the color histogram difference of the original mosaic and γ_i-modified image in Euclidean space. As we can see in Fig. 5.10, when the value of γ increases, the image is overall smoother, but it contains higher color deviation (i.e. notice the top left corner of the coliseum, where the red pattern fades away with increasing γ). In Fig. 5.11 we show the plots for the smoothness ratio and color deviation as γ increases. With increasing γ, the smoothness ratio keeps decreasing, i.e. the transition is smoother, and the trend is to converge to a value that is equal to V_s^{Ref}/V_s^M, where V_s^{Ref} is the smoothness of the input image. The color deviation will also converge if γ goes to infinity, since the color transfer will be dominated by global linear matrix $\mathbf{G}$ (as shown in Eq. 5.4). One interesting thing to point out is when γ continues to decrease, the color-transferred image will contain increasingly many artifacts as without the regularization term, Eq. 5.3 the estimation will not be stable.

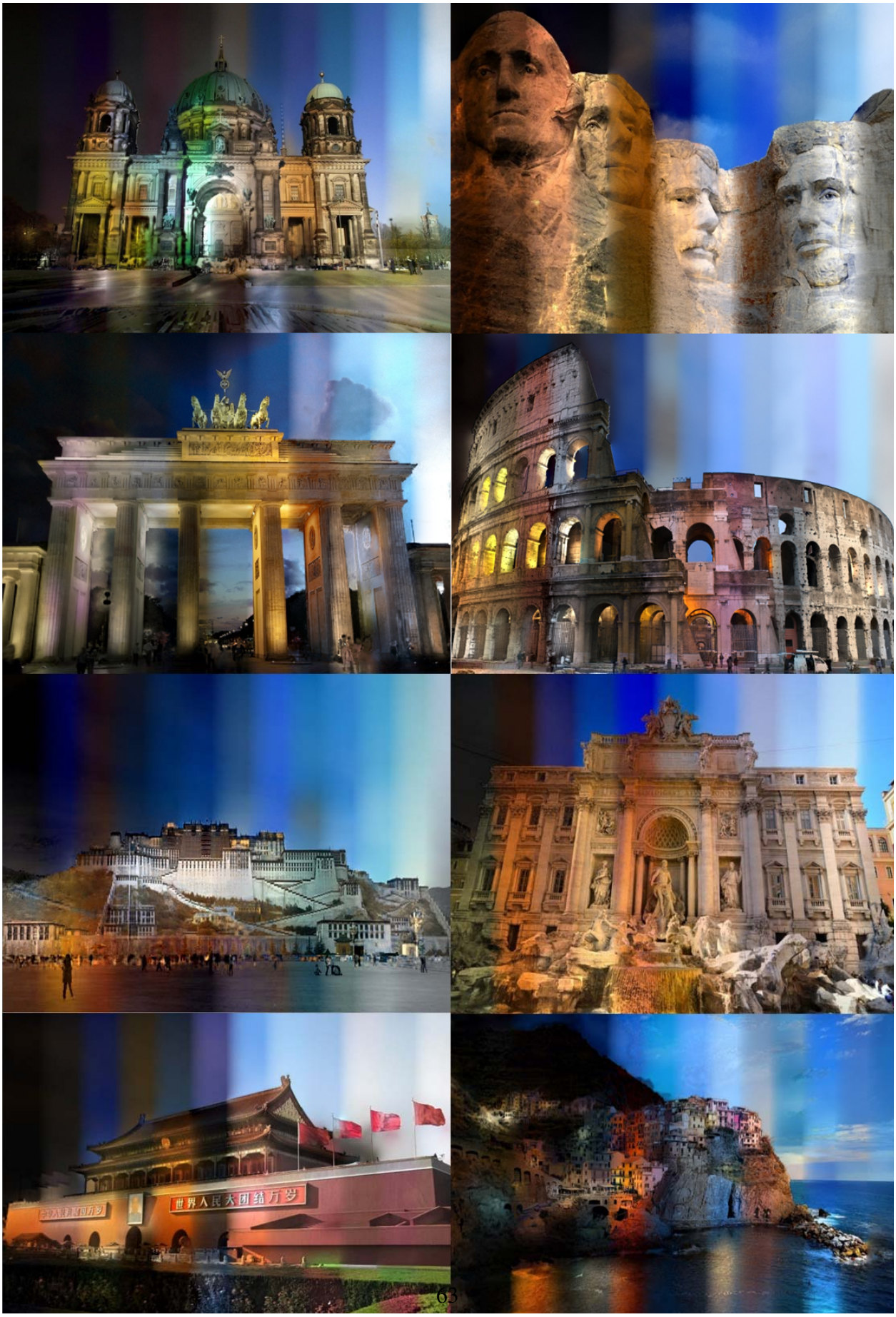


Figure 5.8: Illumination mosaics for eight downloaded datasets.

Figure 5.9: Failed cases for our method. Artifacts appear mainly on the domes and round facades which deviate from planar surfaces.

Figure 5.10: Color-transferred images with different γ, (left) $\gamma = 0.008$, and (right) $\gamma = 0.08$.

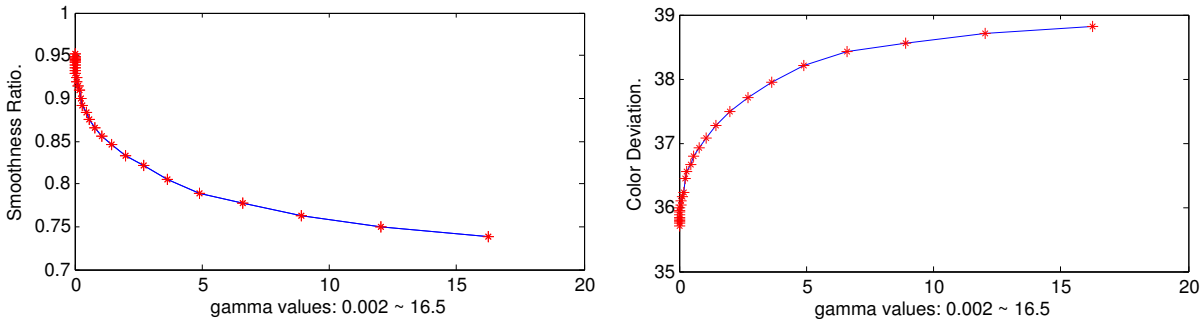


Figure 5.11: The effects of γ on the final mosaic: (left) smoothness ratio, and (right) color deviation.

CHAPTER 6: DYNAMIC VISUAL SEQUENCE PREDICTION WITH MOTION FLOW NETWORKS

6.1 Introduction

Image-based motion prediction aims to generate plausible visualizations of the temporal evolution of an observed scene. In principle, a set of multiple images of the scene of interest may enable geometry-based view synthesis through direct prediction of the variation in scene content and/or viewing parameters (e.g. model-based rendering). However, the problem of direct appearance-based prediction of image motion is heavily ambiguous as the relationship between the scene and the observer is not uniquely defined. The problem becomes even more challenging when the scope of the desired visualization encompasses multiple time steps into the future. In this context, motion prediction can be seen as a pair of complementary problems: view synthesis and motion field estimation. View synthesis strives to render an image observation given partial specification of the scene contents and the observation parameters. Motion field estimation strives to determine dense pixel correspondences among a pair of image observations of a common scene. Given an input image and a motion field, it is straightforward to synthesize a novel image. Conversely, given an input image and a synthesized image, there is an abundance of methods to estimate the motion field. To the best of our knowledge, no supervised learning methods have been deployed to address the motion prediction problem by leveraging the complementary nature of these problems. In this chapter, we attack the motion prediction problem within an image synthesis framework, so as to predict the motion flow and appearance simultaneously.

Predicting pixel values. View synthesis networks are naturally adopted to approach the visual prediction problem. To resolve motion ambiguity, Xue *et al*. (Xue et al., 2016) adopts a variational autoencoder framework to model the uncertainty of predicting the next state of a single input image.

They propose a Cross Convolutional Network to encode image and motion information as feature maps and convolutional kernels, respectively. The network directly outputs future image pixels, while a probabilistic model within the network makes it possible to sample and synthesize many possible future frames from a single input image. However, Zhou *et al*. (Zhou et al., 2016) shows that this kind of model suffers from heavy blurriness when directly outputting pixels. Instead of predicting pixels, Walker *et al*. (Walker et al., 2016) adopt a variational autoencoder to generate a distribution of possible trajectories. They use the output of (Wang and Schmid, 2013) as ground truth for dense pixel trajectories among the source and target images used to train their network. However, there is no evidence that the CNN network can improve upon the given ground truth dense trajectories, potentially imposing systematic biases into the prediction. In our proposed framework, we expect the network to learn the dense motion flows by minimizing the synthesis error through a weakly-supervised encoder-decoder architecture.

Increasing the predictive scope. Predicting images for more than one time step in the future has been previously addressed by Walker *et al*. (Walker et al., 2015) and Zhou *et al*. (Zhou and Berg, 2016). Walker *et al*. take an input image and predicts motion vectors with discretized directions and magnitudes. Recurrent networks are adopted to generate longer sequences. The method proposed in (Zhou and Berg, 2016) generates future image sequences within a generative adversarial network (GAN), which has greatly improved the image generation quality compared to a baseline auto-encoder network. However, the GAN may suffer from systematic appearance artifacts correlated to the training set appearance distribution. We generate multiple output predictions through an iterative network that internally accumulates sequential pairwise pixel motion fields.

Modeling Scene Dynamics. Zhou *et al*. (Zhou et al., 2016) propose "Appearance Flow" to learn dense pixel correspondences between different camera views under weak supervision, this method showed impressive success on static objects. However, predicting the motion of dynamic (and potentially non-rigid) objects is a heavily under-constrained problem. Directionally constrained correspondence prediction was recently addressed by Ji *et al*. (Ji et al., 2017) by learning the epipolar geometric constraints between two views and reducing the 2D flow search to a 1D search.

Their experimental results outperform the traditional 2D appearance flow search (Zhou et al., 2016). However, for dynamic objects, no geometric clues have been adopted to assist the correspondences search. Along these lines, the convolutional pose machine(CPM) (Wei et al., 2016) is recently widely used to detect human body pose, this network is trained with large datasets of labeled human joint positions and achieves astonishing speed and accuracy on 2D human pose estimation. We develop a pair of image synthesis networks: one a general appearance-based predictor, the other a capture-specific pose-constrained predictor.

Our Contributions In this chapter, we propose two motion flow-based view synthesis networks to tackle the visual prediction problem for dynamic scene content. The first network (MotionFlow) predicts 2D motion flows between multiple time steps, while the second network (PoseFlow) constrains the motion flow computation through domain-specific estimated directional priors. The novelty of our work can be summarized as:

- We propose the first weakly-supervised framework to model motion flow for the dynamic sequence synthesis problem.
- We incorporate sparse human body pose estimates to constrain dense motion flow prediction.

6.2 Our Approach

We address two main challenges in the learning-based prediction of extended motion from input images: 1) enhancing visual coherence, while simultaneously 2) reducing the supervision required for training. To this end, we generate future views with two motion flow networks (shown in Fig. 6.1 and 6.4) implemented with encoder-decoder networks. The core idea is to deploy an iterative predictive network to estimate dense correspondence fields across multiple time steps in the future. Since the direct output of the encoder-decoder network are motion fields, the synthesized views are comprised of pixels mapped from the input image instead of pixels directly synthesized by the decoder.

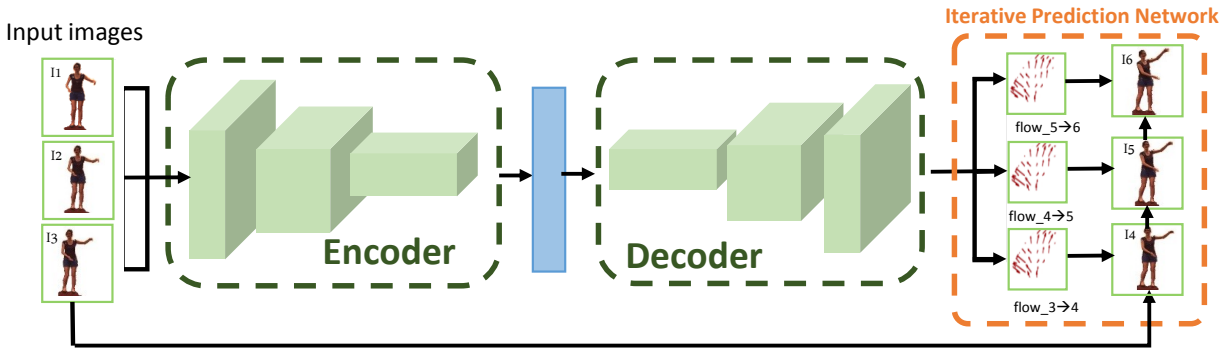


Figure 6.1: MoFlow Network. In this example network, three input images are concatenated as input for encoder network, the decoder network output three motion flows. Pixels of input image 3 are borrowed with learned motion flows to synthesize image in future timesteps so as to minimize the pixel reconstruction errors. The network iteratively borrows pixels from synthesized images to generate future images.

6.2.1 MotionFlowNet: Appearance Flow Estimation for Sequence Synthesis

The goal of an appearance flow network is to synthesize an output target image I_t by sampling pixels from an input source image I_s. The process of pixel sampling is guided by a dense 2D motion flow (e.g. pixel-wise displacement) field. The output of the network is a flow field $f = (f_x^{(i)}, f_y^{(i)})$, defined over the (i) pixels in the input image and yielding an image formation process of the form

$$g(I_s) = I_t(x^{(i)}, y^{(i)}) = I_s(x^{(i)} + f_x^{(i)}, y^{(i)} + f_y^{(i)}), \tag{6.1}$$

In general, learning pairwise correspondence fields requires a set of N source and target image pairs $\langle I_s, I_t \rangle^n \in \mathcal{D}$ are given during the training session. The learning is formalized as minimizing the pixel-wise reconstruction error (i.e. intensity difference):

$$\sum_{\langle I_s, I_t \rangle \in \mathcal{D}} \|I_t - g(I_s)\|_p, \tag{6.2}$$

where $\mathcal{D}$ is the set of training pairs, $g(.)$ refers to the motion-based image from the neural network whose weights we wish to estimate, $\|.\|_p$ denotes the L_p norm. Since the predicted motion fields are in sub-pixel coordinates, the synthesized view is obtained through bi-linear interpolation:

$$I_t^{(i)} = \sum_{q \in \mathcal{B}\left(x^{(i)}, y^{(i)}\right)} I_s^{(q)} (1 - |x^{(i)} - x^{(q)}|) \cdot (1 - |y^{(i)} - y^{(q)}|), \tag{6.3}$$

where $\mathcal{B}\left(x^{(i)}, y^{(i)}\right)$ denotes the set of four integer pixel positions bounding (i.e. top-left, top-right, bottom-left, bottom-right) the real-valued pixel coordinates of a given pixel $(x^{(i)}, y^{(i)})$, which is the corresponding positions for the ith pixel in I_t. To create back-propagation learning framework, the (sub)-gradient of this operations with respect to its inputs can be efficiently computed by the following equations:

$$\frac{\partial I_t^{(i)}}{\partial I_s^{(i)}} = \sum_{q \in \mathcal{B}\left(x^{(i)}, y^{(i)}\right)\}} (1 - |x^{(i)} - x^{(q)}|) \cdot (1 - |y^{(i)} - y^{(q)}|) \tag{6.4}$$

$$\frac{\partial I_t^{(i)}}{\partial x^{(i)}} = \sum_{q \in \mathcal{B}\left(x^{(i)}, y^{(i)}\right)\}} \begin{cases} 1, \text{ if } y^{(i)} <= y^{(q)} \\ -1, \text{ if } y^{(i)} > y^{(q)} \end{cases} \cdot I_s^{(q)} \cdot (1 - |y^{(i)} - y^{(q)}|) \tag{6.5}$$

$$\frac{\partial I_t^{(i)}}{\partial y^{(i)}} = \sum_{q \in \mathcal{B}\left(x^{(i)}, y^{(i)}\right)\}} \begin{cases} 1, \text{ if } x^{(i)} <= x^{(q)} \\ -1, \text{ if } x^{(i)} > x^{(q)} \end{cases} \cdot I_s^{(q)} \cdot (1 - |x^{(i)} - x^{(q)}|) \tag{6.6}$$

To generate multi-frame sequences, the decoder network outputs multiple 2D motion flows, and iteratively takes pixels from the synthesized images to generate future images. Our training

objective is based on pixel-wise prediction over all time steps for training sequences:

$$\sum_{k \in M, \cdots, N} \| I_k - g^{(k-M+1)}(I_{M-1}) \|_2 \} \tag{6.7}$$

In this formulation, for each motion sequence instance, we are given an ordered ground truth image set $\{I_n\}$, partitioned into input motion observations and target image predictions to be used within our supervised learning framework. More specifically, $I_{1 \geq j < M}$ are used as input images depicting the start of a motion sequence, and we aim to predict a sequence of images corresponding to $I_{M \geq k \leq N}$, which depicting the observation at immediately subsequent timesteps. In our notation, $g^{(n)}$ refers to the output image associated with the accumulated n-th motion flow defined over the last available image observation I_{M-1} of the input motion. Accordingly, the direct output of our encoder-decoder network is a set of $N - M$ total predicted pixel motion flows between successive timesteps and having the same pixel dimension as the input imagery.

6.2.2 PoseFlowNet: Appearance Flows with Constrained Directions

Motion flow estimation on dynamic objects is a challenging problem, as there are no geometric constraints (like epipolar constraints learned in (Ji et al., 2017)) that can be leveraged to reduce the motion flow search space. Hence, the correspondence search space for each pixel, into the next frame, spans the whole image. To ease the correspondence problem, we focus on human motion sequences and adopt an off-the-shelf pose estimator (Cao et al., 2016) to reliably determine subject landmarks across our input motion image sequence. We then leverage these detected sparse joint location estimates to 1) make predictions on future pose configurations, and 2) enforce consistency of the estimated dense motion field to these predicted poses. In practice, the geometry-based generalization of sparse local motion estimates is not robust to fine-grain appearance-based cues and leads to strong visual artifacts. Accordingly, the computation of motion flow prediction is decoupled into a directional component estimated from sparse pose predictions and a magnitude component that is estimated from input image observation

Feature Guided Correspondence Computations. The pose estimator outputs sparse joint positions (18 points) for each detected person in the image (shown in Fig. 6.2(a)(b)). If the subject shows up in profile view, some joint points will be missed. We fill these null values with symmetric joint positions. The human body movements are complex as each local part (left arm, right leg *etc.*) moves independently. Beier *et al*. (Beier and Neely, 1992) propose a method to compute how points around line segments move accordingly given line segment movements. With this method, given input human poses, we can obtain dense motion flow between consecutive frames.

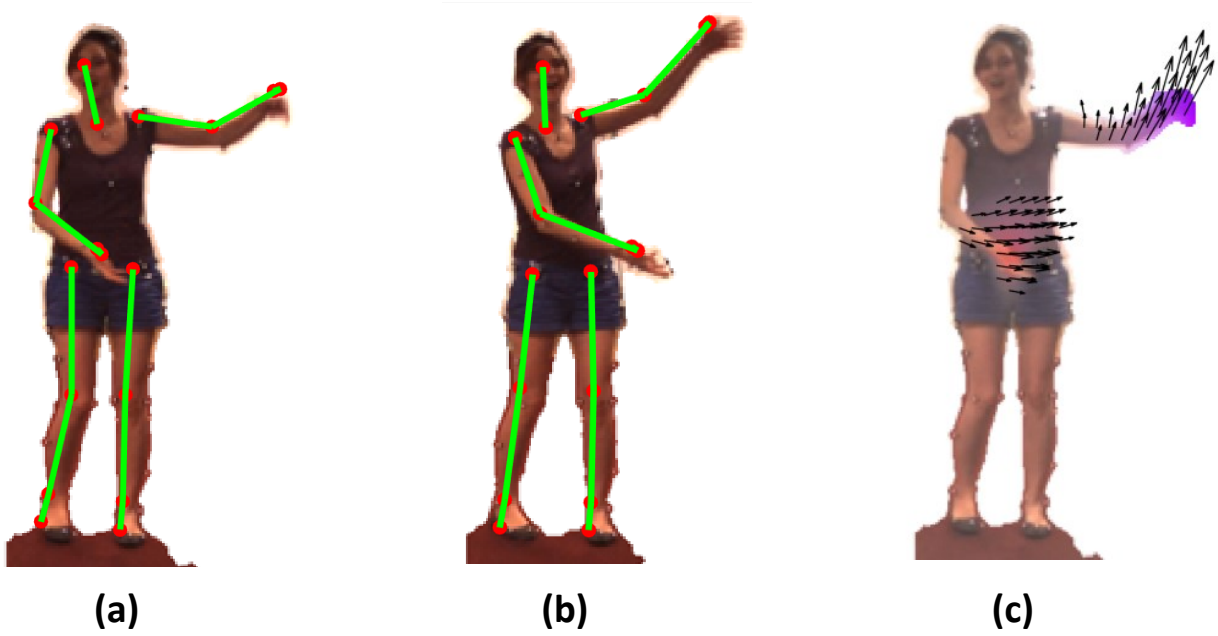


Figure 6.2: (a)(b) Pose estimation results for images within a motion sequence. (c) Computed motion flow with method (Beier and Neely, 1992).

In a 2D image (Fig. 6.3 left), the coordinate mapping of a point X on a line segment MN are represented as (u, v), which are computed by Eq. (6.8),(6.9). If in the next time step (Fig. 6.3 right), position of MN changed to $M^{'}N^{'}$, then the new position of point X would be $X^{'}$ which is computed by Eq. (6.10).

$$u = \frac{(X - M) \cdot (N - M)}{\|N - M\|^2} \tag{6.8}$$

$$v = \frac{(X - M) \cdot Perpendicular(N - M)}{\|N - M\|} \tag{6.9}$$

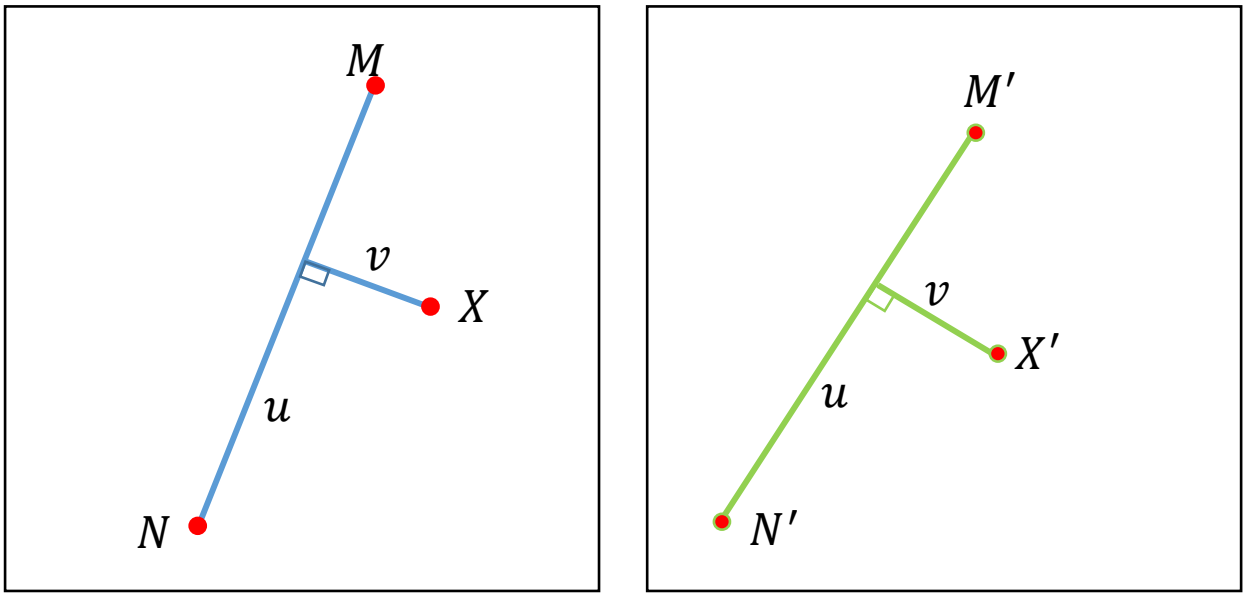


Figure 6.3: Between left and right image, endpoints of line segment MN are changed to $M^{'}N^{'}$.

$$X^{'} = M^{'} + u \cdot (N^{'} - M^{'}) + v \cdot \frac{v \cdot Perpendicular(N^{'} - M^{'})}{\|N^{'} - M^{'}\|} \tag{6.10}$$

Here function $Perpendicular(N - M)$ obtains vector perpendicular to $N - M$, which has the same length as $N - M$. In this coordinate system, value u defines the position along the line, and v is the distance from pixel X to the line MN. The value range of u is 0 to 1 as the pixel moves from M to N, and is less than 0 or greater than 1 outside that range. The value for v is the perpendicular proportional distance from pixel X to the line MN. If there is just one line pair, the transformation of the whole image proceeds as Eq. (6.8),(6.9),(6.10). Since the human body is composed of multiple line segments (we define 14 local parts on the human body.), pixels should naturally move in compliance to its nearest line segment. Since the assignment of pixels to local parts is unknown, a weighting strategy of the coordinate transformations for each line is performed, for each line segment a position $X_i' = (u_i, v_i)$ is computed for each pixel X. To calculate the weighted average of those displacements we follow

$$
\begin{aligned}
w_i =& \frac{1}{(a + dist)^b} \\
X' =& X + \sum_i \frac{w_i}{\sum_i w_i} * (X_i' - X)
\end{aligned}
\tag{6.11}
$$

Here a is a constant to prevent illegal division, variable b decides the displacement of a pixel along with different line segments. If b is large, every pixel will be affected only by the line nearest to it. If b is zero, each pixel will be affected by all lines equally. We set $b = 1.5$ in all experiments. A sample motion flow field is visualized in Fig. 6.2(c) which highlights the motion vectors between Fig. 6.2(a) and Fig. 6.2(b). It can be observed that motion estimates make no distinction between on pixel on a moving limb and nearby pixels not belonging to the limb (e.g. pixels on the torso). We address this challenge by estimating a per-pixel motion magnitude based on the appearance of the input motion sequence.

Sequence Synthesis with Constrained Correspondence Search. We propose the PoseFlow network (shown in Fig. 6.4), which takes images along with detected poses as input. Input poses are fed to a pose prediction network to predict future poses, and generate the dense motion flow fields (with Eq. (6.8),(6.9) and (6.10)) from the predictions. The pose prediction network is composed of four fully connected layers and outputs pose offsets with respect to the previous frame. The detailed network structure is listed in Table 6.1.

The encoder-decoder network has the same configuration as MotionFlowNet. However, instead of predicting 2D motion flows, the output of our decoder is the magnitude of motion flows, the final output of the network is the multiplication of the predicted motion flows and the magnitude fields. By learning appropriate magnitude fields, some mistakenly computed motion flows can be mitigated. For example, in Fig. 6.2(c), we observe motion vectors on torso above the right arm, caused by the proximity to the moving right arm. However, between Fig. 6.2(a) and Fig. 6.2(b), pixels on the torso are actually not moved. We expect the network to optimize magnitudes to mitigate this problem, i.e. the learned magnitudes on torsos would be near zeros.

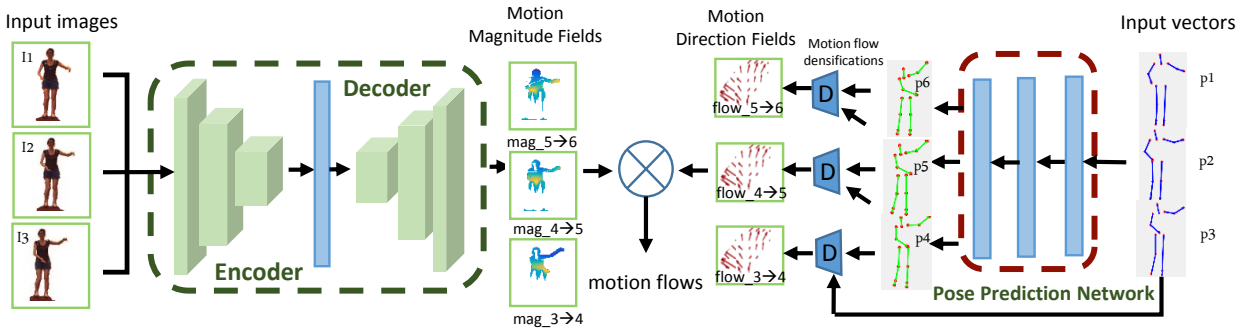


Figure 6.4: PoseFlow network. Left part of the network output pixel-wise predictions of motion flow magnitude, and the right part is a fully connected network predicting the future sparse poses that are densified into directional flow fields.

6.2.3 Implementation details

We trained the network parameters using the ADAM optimization method (Kingma et al., 2014). For different datasets, the input sequences may contain different numbers of images to reduce the motion prediction ambiguity. For our base implementation we train using three stacked images as input motion observations and output three predicted images as a single stack. We present the CNN architecture details of each subnetwork in Table 6.1. "FC2-1" is the CNN feature that is fed into the decoders. The decoder first processes the output of "FC2-1" with five deconvolution layers to perform upsampling of the CNN features. A convolution layer at the end ("DC6" MoFlowNet for and "DC7" for PoseFlowNet) finally outputs the 2D dense motion flow fields or 1D magnitude fields.

6.3 Experiments

Datasets. We adopt two datasets to verify our method, the synthetic *Sprites* dataset and the real image dataset *Human3.6M* (Ionescu et al., 2014; Catalin Ionescu, 2011).

The Sprites Dataset. This dataset consists of 672 unique characters, and for each character there are 5 rigid-body movements from 4 different viewpoints. Each animation ranges from 6 to 13 frames. The image contains a single character, with the original pixel size of 60 $\times$ 60, we resize it to 224 $\times$ 224 to fit our network architecture. In our experiments, our training and testing sequences

Table 6.1: All convolution layers are followed by ReLU. FC1 layer is followed by ReLU and dropout. k: kernel size (kxk). s: stride in horizontal and vertical directions. c: number of output channels. h: number of output heights. w: number of output widths. d: output spatial dimension. Conv: convolution. Deconv: deconvolution. IP: InnerProduct.

Encoder Network						Decoder Network						Pose Prediction Network				
Name	Type	k	s	c	d	Name	Type	k	s	c	d	Name	Type	c	h	w
I_s	Input	.	.	9	224	FC2-1	Input	.	.	64	8	V_s	Input	3	2	18
EC1	Conv	3	2	8	112	DC1	DeConv	9	1	256	15	IP1	IP	.	.	4000
EC2	Conv	3	2	16	56	DC2	DeConv	3	2	128	29	IP2	IP	.	.	2000
EC3	Conv	3	2	32	28	DC3	DeConv	3	2	64	57	IP3	IP	.	.	1000
EC4	Conv	3	2	64	14	DC4	DeConv	3	2	32	113	IP4	IP	.	.	108
EC5	Conv	3	2	128	7	DC5	DeConv	2	2	16	225	IP4-1	Reshape	3	2	18
EC6	Conv	3	2	256	4	DC6	DeConv	2	1	6	224	.	.	.	.	.
FC1	IP	.	.	.	4096	DC7	DeConv	2	1	3	224	.	.	.	.	.
FC2	IP	.	.	.	4096	DC6-1	Reshape	.	.	.	100352	.	.	.	.	.
FC2-1	Reshape	.	.	64	8	DC7-1	Reshape	.	.	.	50176	.	.	.	.	.

have length 6. For animations longer than 6 frames, we take sequences with 5 overlapping frames. For example, an 8 frame animation can generate 3 subsequences with length 6, with frame indices 1-6,2-7,and 3-8. We use 600 characters for training, 72 for testing and collect 12,642 sequences for training, and 2000 sequences for testing.

The Human3.6M Dataset. (Ionescu et al., 2014; Catalin Ionescu, 2011) is collected for tasks like 3D reconstruction of body movements, motion recognition and semantic segmentation. It's acquired by recording the performance of 5 female and 5 male subjects, under 4 different viewpoints. Overall, it has 3.6 million 3D human poses and corresponding images, consisting of 17 scenarios (discussion, smoking, posing, talking on the phone *etc.*). Since the subject number is very limited, we adopt 9 of them for training and 1 for testing. Since "Posing" sequences contains variety of motions, we generate the training and testing sequences from them. With each video, we take 6 consecutive frames as a sequence, the selected sequences have no overlaps, which gives us 10,125 training sequences and 1,600 testing sequences.

Baseline Methods. We compare our methods with a state-of-the-art pixel generating based sequence prediction method **ECCV16** (Zhou and Berg, 2016), which adopt a generative adversarial network to improve the image qualities. We trained both this model and ours on the same datasets. To evaluate the effectiveness of our PoseFlow network, we synthesize the predicted images trough

the method described in (Beier and Neely, 1992) (**SIG92**) using the pose parameters estimated on the ground truth imagery.

Qualitative Evaluations. To illustrate the effectiveness of our method, Figure 6.5 plots the synthesized images from the trained network and compare it with the baseline methods. The third row of Fig. 6.5 shows some artifacts (highlighted in red) generated with (Beier and Neely, 1992), this is caused by inaccurate motion flows of the torso pixels. Our network can learn appropriate magnitudes along the motion directions to mitigate this artifact. Compared with **MoFlowNet**, **PoseFlowNet** has less blurriness (highlighted in green boxes), and more accurate shape deformations (shown in Table 6.4.

In Fig. 6.5, we compare the synthesized images with the baseline methods. **ECCV16** outputs a sequence of 64×64 images, we resize them to be 224×224. While poses can be reasonably predicted, the synthesized appearance can differ strongly from the input image. This can be attributed to the GAN network mimicking the test results by sampling from training samples, instead of borrowing pixels specifically from the input test images. Since the *Sprites* dataset contains synthetic Emoji characters, the pose detector cannot detect poses from them, so we only compare our **MoFlowNet** with **ECCV16** (shown in Fig. 6.6). Again, **ECCV16** can generate correct poses as the groundtruth, however the color is distorted, while our method generates more similar and crisp appearance, especially on the static regions.

Quantitative Evaluations. As an error metric, we use the mean squared error (MSE) between the synthesized output and ground truth summed over all pixels. In Tab. 6.2, we show the MSE for synthesized 3 frames tested on the *Human3.6m* and the *Sprites* dataset. We can see for the *Human3.6M* dataset, the MoFlowNet and PoseFlowNet achieve on par synthesis errors along the sequences, and outperform the baseline methods by big margins. **MoFlowNet** reduce the synthesis errors by half than **ECCV16** on the *Sprites* dataset.

We adopt CPM (Cao et al., 2016) on synthesized images and their groundtruth to compare the estimated pose difference in terms of relative angle (RelAng) and lengths (RelLen). To measure the accuracy of our motion predictions, we compare against the baseline motion for points sampled

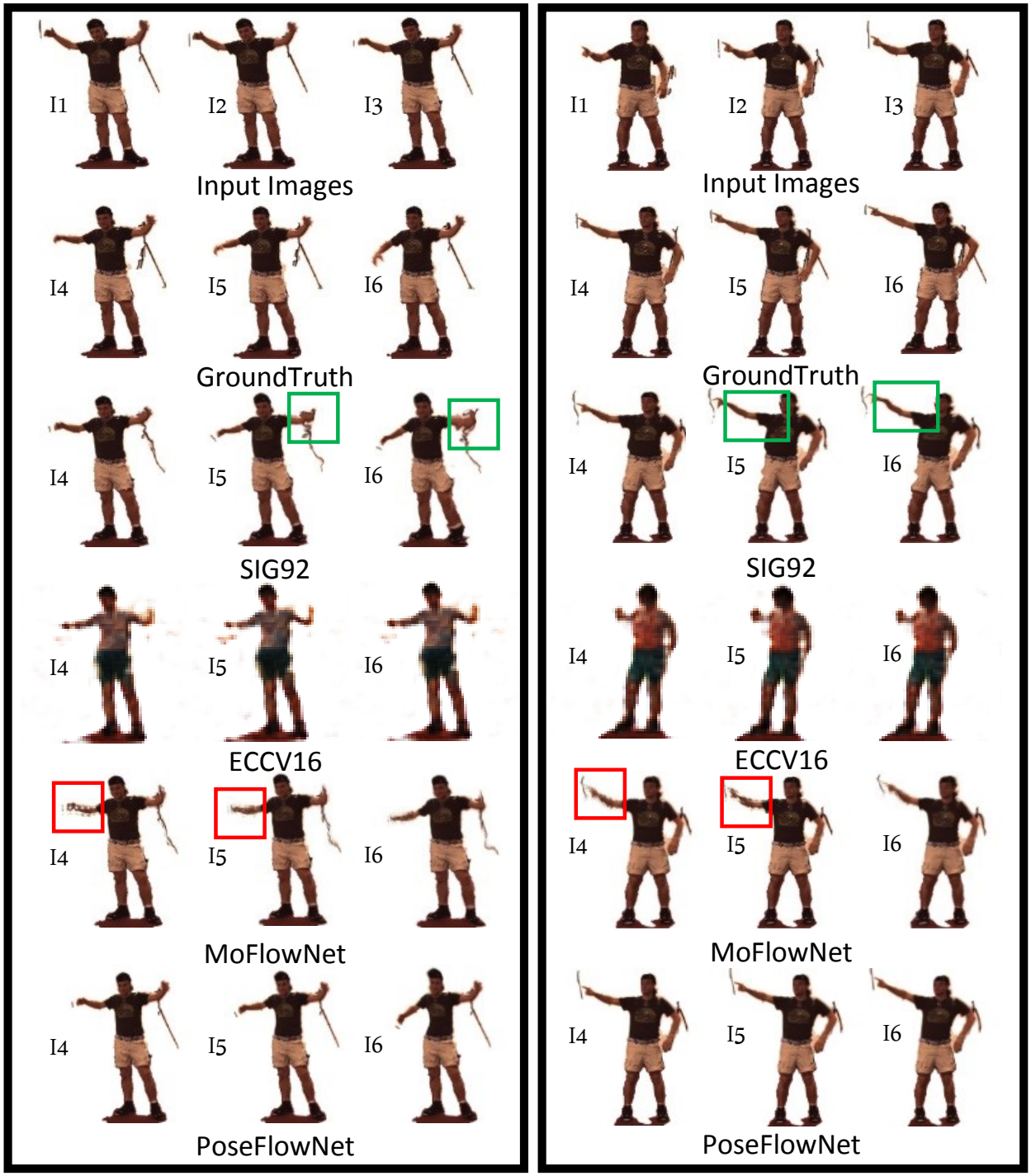


Figure 6.5: Two testing sequences for *Human3.6m* dataset, compare results generated by **SIG92**, **ECCV16**, **MoFlowNet** and **PoseFlowNet**.

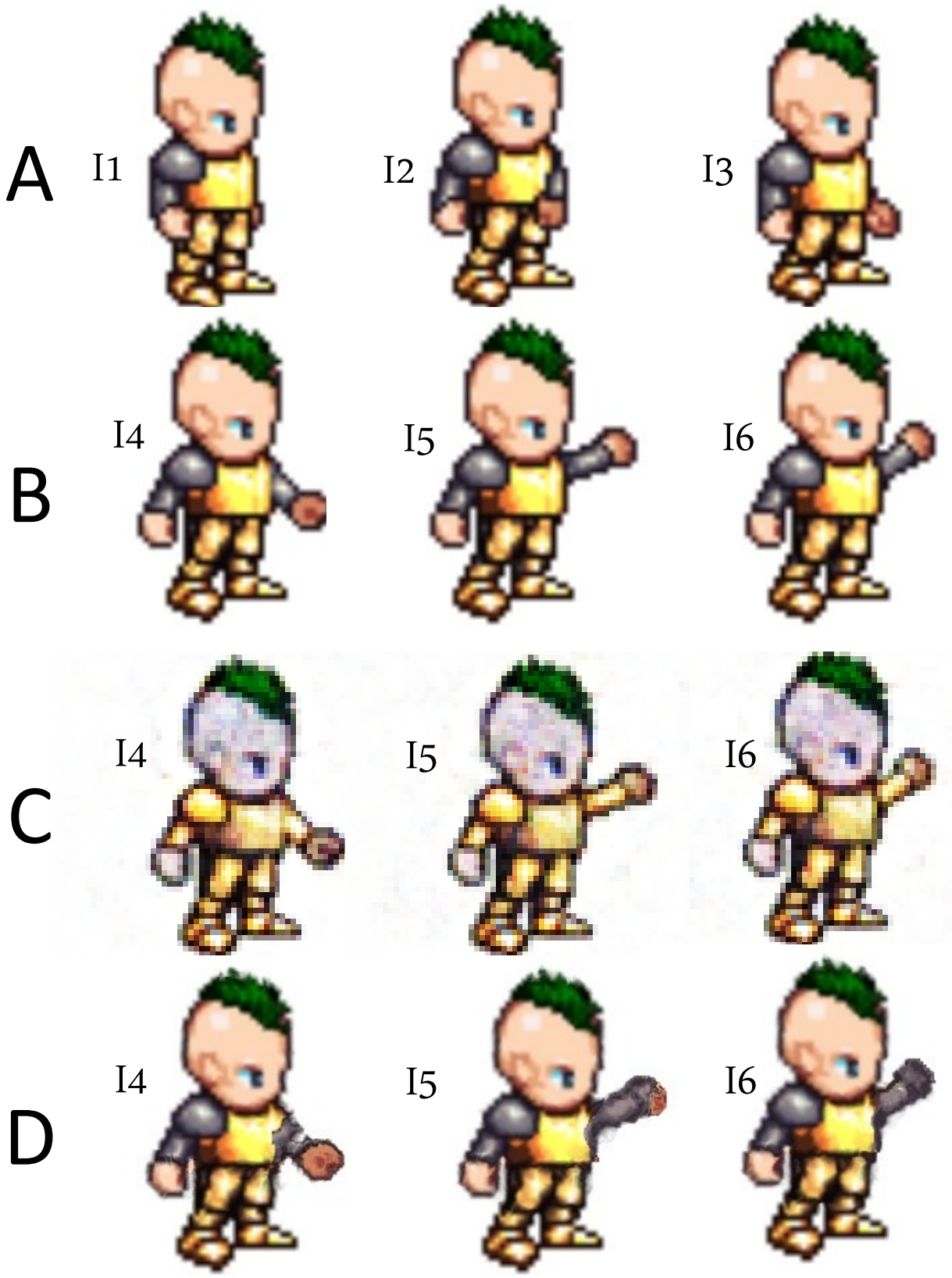


Figure 6.6: Testing sequences for *Sprites* dataset (Row A: input frames, row B: ground truth output frames), and compare results generated by ECCV16 (row C) and MoFlowNet (row D).

Method	Frame 4	Frame 5	Frame 6
SIG92	235.6	561.2	932.5
ECCV16	4602.2	4737.9	4993.1
MoFlowNet	185.1	380.5	850.5
PoseFlowNet	197.6	365.1	796.1
ECCV16	53.9320	54.1431	54.8665
MoFlowNet	27.0103	27.7398	27.9549

Table 6.2: MSE testing error for different frames in human3.6m (top four rows) and Sprites (bottom two rows) dataset.

Method	Frame 4	Frame 5	Frame 6
PosePred	3.59 – 3.55	5.72 – 4.23	6.66 – 5.33
ECCV16	22.78 – 15.20	25.67 – 13.17	33.32 – 18.15
MoFlowNet	1.91 – 3.39	3.90 – 5.26	5.03 – 4.17
PoseFlowNet	**1.54 – 2.84**	**2.11 – 3.23**	**4.54 – 4.32**

Table 6.3: End positions – Motion flow direction prediction error for different frames in human3.6m dataset. The values are in the unit of pixels and degrees.

along the straight-line segments detected on subsequent synthesized and ground truth images (shown in Table 6.3).

To highlight the effectiveness of **PoseFlowNet**'s decoupled motion flow estimation, we compare against the geometry-only flow estimate (**PosePred**) attained from densifying our sparse pose motion predictions. Table. 6.4 shows how **PoseFlowNet** consistently outperforms **ECCV16**, MoFlowNet and the geometry-based motion field estimation.

To verify how the length of input sequences affect the synthesis process, we adopt input length $1-4$ on the *Sprites* dataset and show the first two prediction errors (in Table. 6.5)

To compare with the flow generating network, we take the public model trained for (Walker et al., 2015) and predict the next frame given input images ((Walker et al., 2015) test with one image, PoseFlowNet is tested with the same image and its two previous frames, since our method requires three images as input). The public model only predicts the motion flow of the input image, we visualize the motion flows generated by (Walker et al., 2015) and PoseFlowNet. We adopt the optical flow method proposed by (Liu, 2009) as ground-truth. The red boxes in Fig. 6.7 show our

Method	Frame 4	Frame 5	Frame 6
PosePred	4.82 – 2.11	5.31 – 4.06	5.91 – 6.39
ECCV16	26.22 – 9.51	21.91 – 8.08	20.08 – 7.24
MoFlowNet	3.47 – 1.51	5.29 – 1.93	7.38 – 2.40
PoseFlowNet	**2.78 – 1.32**	**3.93 – 1.69**	**4.82 – 1.88**

Table 6.4: RelAng – RelLen testing error for different frames in human3.6m dataset. The values are in the unit of degrees and pixels.

Input images #	First Prediction	Second Prediction
1	60.2	73.5
2	35.2	41.8
3	27.7	28.0
4	**23.5**	**25.4**

Table 6.5: MoFlowNet testing errors with different input images for frame 5 and 6 on Sprites dataset.

flow direction is closer to the ground-truth. By measuring the direction error on non-white pixels, within the test set, PoseFlowNet and (Walker et al., 2015) achieve 6.3 and 26.8 degree errors.

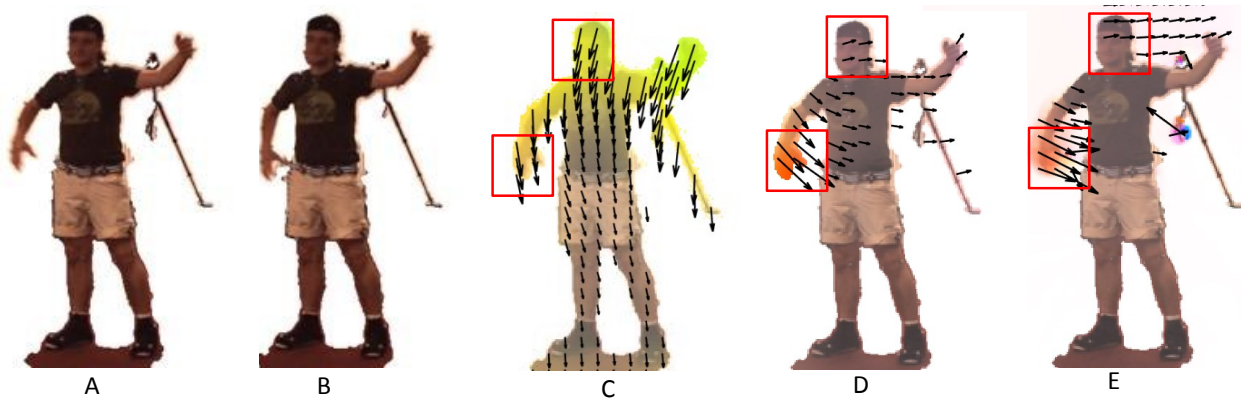


Figure 6.7: Motion flow prediction evaluation (A: input image, B: next frame, C: Flow by (Walker et al., 2015), D: Flow by PoseFlowNet, E: Groundtruth flow)

PoseFlowNet learns magnitude field, which acts like masks. To verify the effectiveness of learned magnitude fields, we compare with the network that fills masks with all 1s. From Fig. 6.8 C, we can see that without learning magnitude fields, the synthesized images (highlighted in green boxes) will have severe distortions. PoseFlowNet prevents pixels from moving into the wrong direction with the help of the learned magnitude fields.

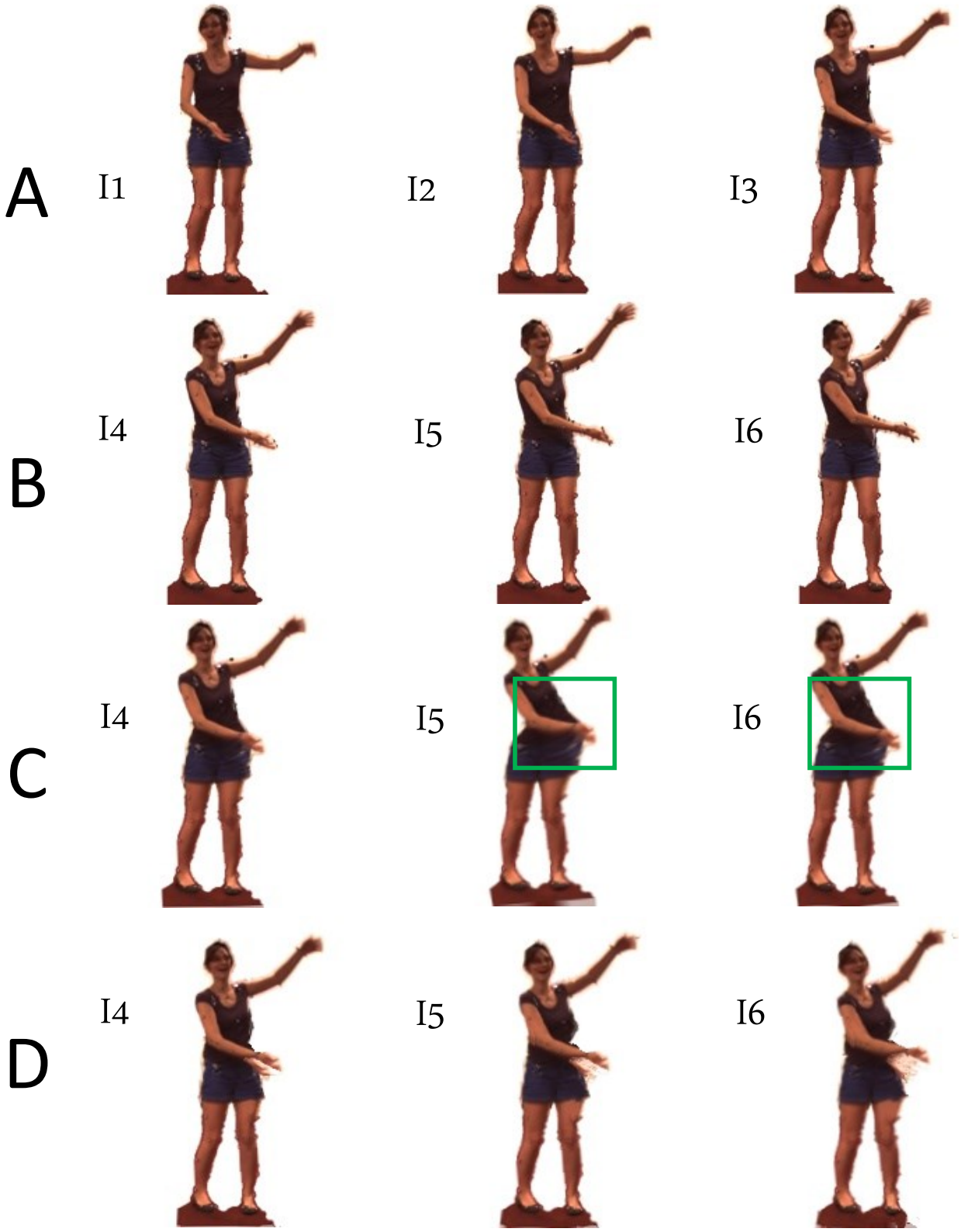


Figure 6.8: One sample test sequence for Human3.6M dataset (Row A: input frames, row B: ground truth output frames), and compare results generated PoseFlowNet without learning the magnitude field (row C) and PoseFlowNet (row D).

CHAPTER 7: DISCUSSION

This dissertation presents four novel approaches for the problems in dynamic 3D reconstruction and dynamic view synthesis. In Chapter 3, we propose a framework to recover geometry of dynamic textures from Internet videos and images. This work verifies that the 2D silhouettes and 3D shapes of dynamic texture could be estimated simultaneously, and be used to update each other until convergence. Chapter 4 aims to recover dynamic rigid body movements from unsynchronized video streams. Inspired by local rigidity, spatio-temporal consistent correspondences are computed, which could be used as clue for video stream synchronization. A novel method to synthesize dynamic illumination transition within mosaics is presented in Chapter 5 to substitute the manual process. Chapter 6 proposes a method for visual motion prediction task in a deep neural network. And during the experiments we found the dynamic appearance synthesis could benefit from some sparse motion priors. In this section, we discuss the possible extensions of our works, as well as the potential future research directions.

7.1 Future work

7.1.1 Extensions to 3D Reconstruction of Dynamic Texture

State-of-the-art SFM methods rely on accurate point correspondences which are difficult to obtain within the regions of dynamic texture. Our method adopts multiview 2D shape correspondences to obtain 3D shapes in the 3D space, and bypasses the reliance of point correspondences.

Though our method on dynamic texture reconstruction (in Chapter 3) outperforms dense reconstruction methods on 3D reconstruction of dynamic textures, one important limitation of this work is the segregation of 3D reconstruction and image segmentation. In our method, 3D geometry and image masks are separately updated, some parameters in GraphCut segmentation

and shape-from-silhouettes are independently adjusted. However, they are closely correlated and should be optimized together in a unified framework. Recently, semantic image segmentation has achieved great success using deep neural networks (Long et al., 2015; Zheng et al., 2015b), 3D operations like projective transformations have been incorporated in deep learning frameworks (Handa et al., 2016), it's possible to unify the two processes in a single deep learning framework. Another advantage of adopting deep neural networks is to learn the parameters automatically, which enable a fully automatic framework for dynamic texture reconstruction. Appearance of dynamic texture like waterfall or fountain water are not totally random, (Doretto et al., 2003a) proposes to represent dynamic texture as auto-regressive moving average process, which is a form of lower dimensional linear representations. (Doretto et al., 2003b) and (Saisan et al., 2001) adopt this model and applied on appearance synthesis, segmentation and recognition tasks. Deep neural networks could automatically learn to parameterize the dynamic texture and reshape the problem in a more formulated way.

Another limitation of our method is the rigid outputs. 3D models generated by our method have fixed shapes, it would be feasible to leverage physical based modeling to create more details like splashing water or shape deformations.

7.1.2 Extensions to 3D Reconstruction of Dynamic Shapes

Previous methods on dynamic shape reconstruction adopt motion priors (C. Bregler and Biermann, 2010; Garg et al., 2013; Zheng et al., 2015a, 2017) or simplified camera models (C. Bregler and Biermann, 2010; Garg et al., 2013). In this thesis, we utilize a different motion prior, namely local rigidity in the 4D trajectory space and refine dense feature correspondences between viewpoints by detecting trajectory consistency within local clusters.

The method on dynamic shape reconstruction (presented in Chapter 4) uses local rigidity as a clue to find spatial-temporal consistent correspondences between different camera viewpoints, and each local rigid part could be computed independently. A more efficient implementation with GPU could be incorporated in the future work, in which each computation unit handle one local

cluster. In motion capture based system on synchronized cameras (Joo et al., 2014, 2015), more cameras give more visibility coverage. However, camera synchronization and data storage are not trivial tasks as the system scales up, causing huge computational expenses and manual efforts. Thus, motion capture with unsynchronized cameras have huge advantages compared with its counterpart. Although our method proposed in the thesis works in two camera basis, it's straight-forward to scale up. With multiple video cameras, we first find neighboring camera pairs that have smallest baseline, adding camera pairs in the optimization formulations and maximize 3D trajectories consistency visible by at least two cameras. Cycle consistency among cameras should be considered to enforce neighborhood consistency constraints.

7.1.3 Extensions to View Synthesis

The method on illumination synthesis (proposed in Chapter 5) is a novel way to synthesize illumination mosaics which visualize appearance changes over time in a single viewpoint. Image sequences are found within a graph characterizing the illumination attributes. A more generative extension would be including more attributes like weather and seasons. The image sequences obtained by this method could be adopted to texture 3D model and visualize appearance transition in the 3D space.

Previous methods on visual prediction model visual views with pixel generating networks (Zhou and Berg, 2015; Xue et al., 2016) or predicting two dimensional motion field (Walker et al., 2016, 2015). In this thesis, we propose to adopt sparse motion prior to assist motion flow estimation, reducing the problem from 2D to 1D. Experimental results on multiple datasets are shown to support the efficiency. The network architecture proposed in Chapter 6 predicts motion sequences in the 2D domain, in the future work, we expect to directly learn 3D motion sequences from monocular/muti-view image sequences.

BIBLIOGRAPHY

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, page 2274.

Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., and Seitz, S. (2011). Building rome in a day. *Communications of the ACM*, 54(10):105–112.

Agarwal, S., Snavely, N., Simon, I., Seitz, S., and Szeliski, R. (2012). Building rome in a day. *IEEE International Conference on Computer Vision (ICCV)*.

Agarwala, A., Dontcheva, A., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M. (2004). Interactive Digital Photomontage. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*.

Akhter, I., Sheikh, Y., Khan, S., and Kanade, T. (2011). Trajectory space: A dual representation for nonrigid structure from motion. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Albl, C., Kukelova, Z., Fitzgibbon, A., Heller, J., Smid, M., and Pajdla, T. (2017). On the two-view geometry of unsynchronized cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ballan, L., Brostow, G., Puwein, J., and Pollefeys, M. (2010a). Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*.

Ballan, L., Brostow, G. J., Puwein, J., and Pollefeys, M. (2010b). Unstructured video-based rendering: Interactive exploration of casually captured videos. In *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*, volume 29, page 87. ACM.

Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., and Sayd, P. (2008). Coarse-to-fine low-rank structure-from-motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Basha, T., Moses, Y., and Avidan, S. (2012). Photo Sequencing. *European Conference on Computer Vision (ECCV)*.

Basha, T., Moses, Y., and Avidan, S. (2013). Space-Time Tradeoffs in Photo Sequencing. *IEEE International Conference on Computer Vision (ICCV)*.

Baumgart, B. (1974). Geometric modeling for computer vision. *Ph. D. Thesis (Tech. Report AIM-249), Stanford University*.

Beier, T. and Neely, S. (1992). Feature-based image metamorphosis. In *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*.

Black, M. and Anandan, P. (1991). Robust dynamic motion estimation over time. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bonet, J. S. D. and Viola, P. A. (1999). Roxels: Responsibility weighted 3d volume reconstruction. *IEEE International Conference on Computer Vision (ICCV)*, 1:418.

Brand, M. (2001). Morphable 3d models from video. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. *European Conference on Computer Vision (ECCV)*.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2014). High accuracy optical flow estimation based on a theory for warping. *European Conference on Computer Vision (ECCV)*.

C. Bregler, A. H. and Biermann, H. (2010). Recovering non-rigid 3d shape from image streams. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cagniart, C., Boyer, E., and Ilic, S. (2010). Probabilistic deformable surface tracking from multiple videos. In *European Conference on Computer Vision (ECCV)*, pages 326–339. Springer.

Canny, J. (1986). A computational approach to edge detection. i. *pami*, page 679.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2016). Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*.

Catalin Ionescu, Fuxin Li, C. S. (2011). Latent structured models for human pose estimation. In *International Conference on Computer Vision*.

Denton, L, E., Soumith, C., Arthur, S., and Rob, F. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems 28*, pages 1486–1494. Curran Associates, Inc.

Djelouah, A., Franco, J.-S., Boyer, E., Le Clerc, F., and Pérez, P. (2015). Sparse multi-view consistency for object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 37(9):1890–1903.

Doretto, G., Chiuso, A., Wu, Y., and Soatto, S. (2003a). Dynamic textures. *International Journal of Computer Vision (IJCV)*, 51(2):91.

Doretto, G., Cremers, D., Favaro, P., and Soatto, S. (2003b). Dynamic texture segmentation. *IEEE International Conference on Computer Vision (ICCV)*.

Dosovitskiy, A., Springenberg, J., and Brox, T. (2015). Learning to generate chairs with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Elad, M. and Feuer, A. (1998). Recursive optical flow estimation adaptive filtering approach. *Visual Communication and Image Representation*.

Frahm, J.-M., Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y., Dunn, E., Clipp, B., Lazebnik, S., and Pollefeys, M. (2010). Building rome on a cloudless day. *European Conference on Computer Vision (ECCV)*.

Franco, J.-S. and Boyer, E. (2005). Fusion of multi-view silhouette cues using a space occupancy grid. *IEEE International Conference on Computer Vision (ICCV)*, 2:1747.

Furukawa, Y. and Ponce, J. (2006). Carved visual hulls for image based modeling. *European Conference on Computer Vision (ECCV)*.

Furukawa, Y. and Ponce, J. (2010). Towards internet-scale multi-view stereo. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1434.

Garg, R., Roussos, A., and Agapito, L. (2013). Dense variational reconstruction of non-rigid surfaces from monocular video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets.

Guan, L., Franco, J.-S., and Pollefey, M. (2008). Multi-object shape estimation and tracking from silhouette cues. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Guan, L., Franco, J.-S., and Pollefeys, M. (2010). Multi-view occlusion reasoning for probabilistic silhouette-based dynamic scene reconstruction. *International Journal of Computer Vision (IJCV)*, 90(3):283–303.

Handa, A., Bloesch, M., Pătrăucean, V., Stent, S., McCormac, J., and Davison, A. (2016). gvnn: Neural network library for geometric computer vision. In *ECCV Workshop on Geometry Meets Deep Learning*.

Hasler, N., Rosenhahn, B., Thormahlen, T., Wand, M., Gall, J., and Seidel, H. (2009). Markerless motion capture with unsynchronized moving cameras. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 224.

Hays, J. and Efros, A. (2007). Scene Completion Using Millions of Photographs. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*.

Heinly, J., Schnberger, J., Dunn, E., and Frahm, J.-M. (2015). Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hinton, G., Krizhevsky, A., and Wang, S. (2011). Transforming autoencoders. In *In Proc. ICANN*.

Hoiem, D., Efros, A., and Hebert, M. (2005). Geometric Context from a Single Image. *IEEE International Conference on Computer Vision (ICCV)*.

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339.

Irani, M. (1999). Multi-frame optical flow estimation using subspace constraints. *IEEE International Conference on Computer Vision (ICCV)*.

J. Flynn, I. Neulander, J. P. and Snavely, N. (2015). Deepstereo: Learning to predict new views from the world's imagery. In *IEEE International Conference on Computer Vision (ICCV)*.

Jacobs, N., Roman, N., and Pless, R. (2007). Consistent Temporal Variations in Many Outdoor Scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jancosek, M. and Pajdla, T. (2011). Multi-view reconstruction preserving weakly-supported surfaces. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 3121.

Ji, D., Dunn, E., and Frahm, J. (2014). 3d reconstruction of dynamic textures in crowd sourced data. In *European Conference on Computer Vision (ECCV)*.

Ji, D., Dunn, E., and Frahm, J. (2015). Synthesizing illumination mosaics from internet photo-collections. In *European Conference on Computer Vision (ECCV)*, pages 3988–3996.

Ji, D., Dunn, E., and Frahm, J. (2016). Spatio-temporally consistent correspondence for dense dynamic scene modeling. In *European Conference on Computer Vision (ECCV)*.

Ji, D., Kwon, J., McFarland, M., and Savarese, S. (2017). Deep view morphing. *Computer Vision and Pattern Recognition (CVPR)*.

Jiang, H., Liu, H., Tan, P., Zhang, G., and Bao, H. (2012). 3d reconstruction of dynamic scenes with multiple handheld cameras. *European Conference on Computer Vision (ECCV)*.

Jones, M. and Poggio, T. (1995). Model-based matching of line drawings by linear combination of prototypes. In *IEEE International Conference on Computer Vision (ICCV)*.

Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., and Sheikh, Y. (2015). Panoptic studio: A massively multiview system for social motion capture. *IEEE International Conference on Computer Vision (ICCV)*.

Joo, H., Park, H., and Sheikh, Y. (2014). Map visibility estimation for large scale dynamic 3d reconstruction. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kanade, T., Rander, P., and Narayanan, P. (1997). Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*.

Katayama, A., Tanaka, K., Oshino, T., and Tamura, H. (1995). A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images. In *Stereoscopic Displays and Virutal Reality Systems*.

Kemelmacher-Shlizerman, I., Shechtman, E., Garg, R., and Seitz, S. (2011). Exploring Photobios. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*, 30.

Kim, H., Sarim, M., Takai, T., Guillemaut, J., and Hilton, A. (2010). Dynamic 3d scene reconstruction in outdoor environments. *International Conference on 3D Vision (3DV)*.

Kingma, D., Ba, J., and Gamow, G. (2014). Adam: A method for stochastic optimization. In *Technical report, arXiv:1412.6980*.

Kitani, K., Ziebart, B., Bagnell, J., and Hebert, M. (2012). Activity forecasting. *European Conference on Computer Vision (ECCV)*.

Laffont, P., Ren, Z., Tao, X., Qian, C., and Hays, J. (2014). Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*, 33(149).

Laurentini, A. (1994). The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 16(2):150.

Lee, K., Luo, S., and Chen, B. (2000). Rephotography Using Image Collections. *Pacific Graphics*.

Letouzey, A. and Boyer, E. (2012). Progressive shape models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 190–197. IEEE.

Levin, A., Zomet, A., Peleg, S., and Weiss, Y. (2004). Seamless Image Stitching in the Gradient Domain. *European Conference on Computer Vision (ECCV)*.

Liu, C. (2009). Beyond pixels: Exploring new representations and applications for motion analysis.

Liu, J., Sun, J., and Shum, H. (2009a). Paint selection. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*, 28(3):69.

Liu, J., Sun, J., and Shum, H.-Y. (2009b). Paint selection. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional models for semantic segmentation.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91.

Martin, O. and Daniel, C. (2013). A convex relaxation approach to space time multi-view 3d reconstruction.

Mustafa, A., Kim, H., Guillemaut, J., and Hilton, A. (2015). General dynamic scene reconstruction from multiple view video. *IEEE International Conference on Computer Vision (ICCV)*.

Nelson, R. and Polana, R. (1992). Qualitative recognition of motion using temporal texture. *CVGIP: Image Understanding*, 56:78.

Oliva, A. (2005). Gist of the scene. *Neurobiology of attention*, 696:251.

Oswald, M., Stühmer, J., and Cremers, D. (2014). Generalized connectivity constraints for spatio-temporal 3d reconstruction. pages 32–46. IEEE.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, 9(1):62.

Palermo, F., Hays, J., and Efros, A. (2012). Dating Historical Color Images. *European Conference on Computer Vision (ECCV)*.

Papenberg, N., Bruhn, A., Brox, T., Didas, S., and Weickert, J. (2006). Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision (IJCV)*.

Park, H. S., Shiratori, T., Matthews, I., and Sheikh, Y. (2010). 3d reconstruction of a moving point from a series of 2d projections. In *European Conference on Computer Vision (ECCV)*.

Peter Sand, S. T. (2006). Particle video: Long-range motion estimation using point trajectories.

Pintea, S. L., van Gemert, J. C., and Smeulders, A. W. (2014). Dejavu: Motion prediction in static images.

Pollefeys, M., Nister, D., Frahm, J.-M., and et. al. (2007). Detailed real-time urban 3d reconstruction from video. *European Conference on Computer Vision (ECCV)*.

Pundik, D. and Moses, Y. (2010). Video synchronization using temporal signals from epipolar lines. *European Conference on Computer Vision (ECCV)*, pages 15–28.

Radenović, F., S., J., Ji, D., Frahm, J.-M., Chum, O., and Matas, J. (2016). From dusk till dawn: Modeling in the dark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P. (2001). Color transfer between images. *Computer Graphics and Application*, 21(5):3441.

Russell, C., Yu, R., and Agapito, L. (2014). Video pop-up: Monocular 3d reconstruction of dynamic scenes. *European Conference on Computer Vision (ECCV)*.

Saisan, P., Doretto, G., Wu, Y., and Soatto, S. (2001). Dynamic texture recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schindler, G. and Dellaert, F. (2007). Inferring Temporal Order of Images From 3D Structure. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schindler, G. and Dellaert, F. (2010). Probabilistic Temporal Inference on Reconstructed 3D Scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schonberger, J., Radenovic, F., Chum, O., and Frahm, J.-M. (2015). From Single Image Query to Detailed 3D Reconstruction). *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Schönberger, J. L., Zheng, E., Pollefeys, M., and Frahm, J.-M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*.

Seitz, S. and Dyer, C. (1996). View morphing. In *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*.

Shi, J. and Malik, J. (1998). Motion segmentation and tracking using normalized cuts. *IEEE International Conference on Computer Vision (ICCV)*.

Shih, Y., Paris, S., Durand, F., and Freeman, W. (2013). Data-driven Hallucination for Different Times of Day from a Single Outdoor Photo. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*, 32(6).

Sinha, S. N. and Pollefeys, M. (2005). Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. *IEEE International Conference on Computer Vision (ICCV)*.

Snavely, N., Garg, R., Seitz, S., and Szeliski, R. (2008). Finding Paths through the World's Photos. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*, 27(3):11–21.

Snavely, N., Seitz, S., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*, 25(3):835–846.

Taneja, A., Ballan, L., and Pollefeys, M. (2010). Modeling dynamic scenes recorded with freely moving cameras. *European Conference on Computer Vision (ECCV)*.

Tao, L., Yuan, L., and Sun, J. (2009). SkyFinder: Attribute-based Sky Image Search. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*, 28(4).

Tatarchenko, M., Dosovitskiy, A., and Brox, T. (2016). Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision (ECCV)*.

Tuytelaars, T. and Gool, L. (2004). Synchronizing video sequences. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 762–768.

Vondrick, C., Pirsiavash, H., and Torralba, A. (2015). Anticipating the future by watching unlabeled video. *CoRR*.

Walker, J., Doersch, C., Gupta, A., and Hebert, M. (2016). An uncertain future: Forecasting from static images using variational autoencoders. *European Conference on Computer Vision (ECCV)*.

Walker, J., Gupta, A., and Hebert, M. (2014). Patch to the future: Unsupervised visual prediction. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Walker, J., Gupta, A., and Hebert, M. (2015). Dense optical flow prediction from a static image. *IEEE International Conference on Computer Vision (ICCV)*.

Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*, Sydney, Australia.

Wang, J., Tong, X., Lin, S., Pan, M., Wang, C., Bao, H., Guo, B., and Shum, H. (2006). Appearance Manifolds for Modeling Time-Variant Appearance of Materials. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH*, 25(4).

Wang, K., Stutts, C., Dunn, E., and Frahm, J. M. (2016a). Efficient joint stereo estimation and land usage classification for multiview satellite data. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9.

Wang, Y., Wang, K., Dunn, E., and Frahm, J.-M. (2016b). Stereo under sequential optimal sampling: A statistical analysis framework for search space reduction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wills, J. and Belongie, S. (2004). A feature-based approach for determining dense long range correspondences. *European Conference on Computer Vision (ECCV)*.

Wolf, L. and Zomet, A. (2006). Wide baseline matching between unsynchronized video sequences. *International Journal of Computer Vision (IJCV)*, 68:43–52.

Wu, C. (2013). Towards linear-time incremental structure from motion. *Proceedings of the international symposium on 3D data processing, visualization and transmission (3DPVT)*.

Wu, C., Varanasi, K., Liu, Y., Seidel, H.-P., and Theobalt, C. (2011). Shading-based dynamic shape refinement from multi-view video under general illumination. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1108–1115. IEEE.

Xu, X., Wan, L., Liu, X., Wong, T., Wang, L., and Leung, C. (2008). Animating Animal Motion from Still. *ACM Transactions on Graphics (TOG), Proceedings of SIGGRAPH Asia*.

Xue, T., Wu, J., Bouman, K. L., and Freeman, W. T. (2016). Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Yang, J., Reed, S., Yang, M., and Lee, H. (2015). Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems (NIPS)*.

Yen, Y. (1970). An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of Applied Mathematics*, 27:526.

Yuen, J. and Torralba, A. (2010). A data-driven approach for event prediction. *European Conference on Computer Vision (ECCV)*.

Zhang, F. and Liu, F. (2014). Parallax-tolerant Image Stitching. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zheng, E., Dunn, E., Jojic, V., and Frahm, J.-M. (2014). Patchmatch based joint view selection and depthmap estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zheng, E., Ji, D., Dunn, E., and Frahm, J. (2015a). Sparse dynamic 3d reconstruction from unsynchronized videos. *IEEE International Conference on Computer Vision (ICCV)*, pages 4435–4443.

Zheng, E., Ji, D., Dunn, E., and Frahm, J. (2017). Self-expressive dictionary learning for dynamic 3d reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Zheng, E. and Wu, C. (2015). Structure from motion using structureless resection. In *IEEE International Conference on Computer Vision (ICCV)*.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. (2015b). Conditional random fields as recurrent neural networks. In *IEEE International Conference on Computer Vision (ICCV)*.

Zhou, T., Tulsiani, S., Sun, W., Malik, J., and Efros, A. A. (2016). Single-view modelling of free-form scenes. In *European Conference on Computer Vision (ECCV)*.

Zhou, Y. and Berg, T. (2015). Temporal perception and prediction in ego-centric video. *IEEE International Conference on Computer Vision (ICCV)*.

Zhou, Y. and Berg, T. (2016). Learning temporal transformations from time-lapse videos. *European Conference on Computer Vision (ECCV)*.