Social Scene Understanding from First Person Cameras

Hyun Soo Park

June 28

Mechanical Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Yaser Sheikh, Chair (Carnegie Mellon University)
Jessica K. Hodgins (Carnegie Mellon University)
Levent Burak Kara (Carnegie Mellon University)
Kenji Shimada (Carnegie Mellon University)
Christoph Bregler (New York University)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

A social scene is a scene occupied by humans. In these scenes, humans frequently interact with each other while sending visible social signals, such as facial expressions, body gestures, and gaze movements. Such scenes are very common in our daily life and, increasingly, artificial agents are entering these spaces. Classic scene understanding has focused on understanding the *structure* of scenes, e.g., where a sofa is, how to navigate the room, or what an object is useful for. For artificial agents to cohabit these scenes with humans as collaborating team members, it is necessary that they understand social context, such as what people cognitively attend to, what people want to accomplish, and whom people are interacting with. The classic definition of scene understanding must be expanded to include interpreting what is *socially salient* in a scene. In this thesis, we establish a computational basis towards understanding the relationship between social motion and social saliency.

A first person camera is a wearable camera looking out at a scene from the perspective of the wearer — the camera sees what the wearer sees. First person cameras are ideally placed to image a social scene for two reasons. First, first person cameras naturally secure the best view because humans intelligently move to look at what they are interested in from the best view point. Second, the more socially salient an event is, the more first person views of the event are likely to be available. We exploit these advantages of first person cameras, as socially immersed sensors, to interpret visible social signals associated with social context.

3D reconstruction of motion: Reconstructing motion in 3D from an image sequence is an ill-posed problem because there is one dimension lost while projecting a 3D point onto an image plane. We apply a temporal constraint on a moving point to solve this problem by representing the trajectory of the point using a linear combination of basis trajectories. This enables us to produce a linear least squares system for the trajectory parameters. Our solution is robust against missing data and measurement noise. For human motion, trajectories on adjacent joints are also spatially constrained, i.e., the distance between adjacent joint trajectories remains constant across time instances. We apply temporal and spatial constraints simultaneously on the adjacent joint trajectories. This enables us to reconstruct an articulated trajectory in 3D from a single first person camera. We also characterize the fundamental limitation of trajectory reconstruction via geometric analysis.

3D reconstruction of social saliency: We reconstruct social saliency in 3D by estimating where people look from first person cameras. A gaze concurrence is a 3D point where multiple people's gaze directions converge. It is a socially salient point because the attention of multiple people is directly linked to that point. Although an individual's gaze indicates what he or she is subjectively interested in, a gaze concurrence encodes the consensus of multiple individuals; the agreement of multiple subjective interpretations produces a representation of social saliency that approaches objectivity. We model the gaze with a cone-shaped distribution emitted from the center of eyes. This model captures the variation of eye-in-head motion. We calibrate the gaze model with respect to the first person camera. The resulting gaze model produces a social saliency field in 3D and we seek the modes of the field using a mode-seeking algorithm. The number and 3D locations of the gaze concurrences in the social saliency field are automatically estimated.

3D reconstruction of socially salient motion (proposed work): Through 3D reconstruc-

tion of motion and socially saliency, we will show how the challenges regarding social scene understanding can be resolved. As proposed work, we will study the relationship between social motion and social saliency. Estimation of social motion in conjunction with gaze concurrences will allow us to localize what is socially significant in the scene and predict how it will move. We will reconstruct human motion in 3D and infer the motion that triggers group responses based on social saliency.

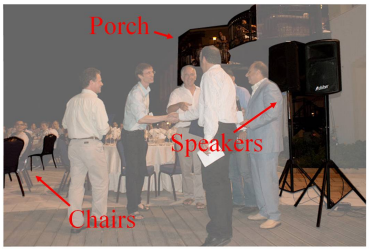
Throughout this thesis, we aim to understand a social scene by 3D reconstruction from socially immersed first person cameras. Our overarching goal is to develop algorithms that will enable an artificial agent to organically collaborate with us in our social spaces without continual prompting.

Contents

Chapter 1

Introduction

(a) Wedding reception



(b) Structural scene understanding

(c) Social scene understanding

Figure 1.1: (a) In a social scene, such as a wedding reception, people interact with others via visible social signals such as eye contact, gaze direction, and body motion. (b) Structural scene understanding algorithms interpret the structural context that characterizes geometric relationships in a scene, e.g., object recognition, structure from motion, and human affordance identification. (c) In a social scene occupied by many people, the definition of scene understanding must be expanded to include social context such as human interaction, interest, and motion. In this thesis, we aim to understand a social scene from first person cameras.

Suppose you are a waiter/waitress at a wedding reception, as shown in Figure 1.1(a). How would you interpret the scene? You see people socializing with others and participating in events such as the wedding toast, dancing, and cake cutting. In the process of interpreting the scene, you recognize where the main event happens and that you should not disturb what people want to see; you should not disturb them when they dance; when people form groups to socialize, you should avoid breaking into the group. You are trained to understand what people want to do via their visible social signals, such as eye contact, gaze direction, or body motion. Based on this understanding, you operate in accordance with *social context*, e.g., what people are cognitively attending to, what they are trying to accomplish, and whom they are interacting with. Consider how different your interpretation of the scene would be if you were the father of the bride, a jilted lover, or the wedding photographer. Given these many interpretations of the same scene, how should we represent social activity in a unified way so that artificial agents find it accessible? Could we produce such a unified interpretation of the scene if we had access to all the participants' views?

We define a *social scene* as a scene occupied by many people and where human interactions frequently take place, such as a wedding reception, a conference poster session, or a sporting event. These social scenes are very common in our daily life and, increasingly, artificial agents are entering these social spaces. Vacuum cleaning robots, for example, navigate rooms where humans reside and surgical robots help surgeons assisted by a medical staff. As they become integrated in our lives, we expect them to play a role no longer as tools that require prompting but as team members that organically interact with humans and accomplish tasks, seamlessly and safely. For artificial agents to be able to coexist with humans in a social scene, they need to understand social context and their tasks and strategies must be designed to respect social context based on this understanding.

Robotics and computer vision research has focused on understanding the *structural context* that characterizes the geometric relationship of a scene such as SLAM (Simultaneous Localization And Mapping) [102], object recognition [35], and image segmentation [10]. These enable the artificial agents to understand where a building is, how to navigate a room, or what an object is useful for, as shown in Figure 1.1(b). As a result, structural context about a scene is relatively well understood while the ability to understand social context (Figure 1.1(c)) is still limited. In this thesis, we aim to understand/represent social context that arises in a social scene.

Social context in a scene is often time-variant and it emerges in the form of the motion of the scene. Individual motions, such as facial expressions, gestures, and gaze movements are primary social signals that spontaneously arise during social interaction. These motions reflect the sender's emotion, intention, and attention [120]. At the wedding reception, for instance, the groom's friend may raise his champagne glass to propose a toast to the bride and groom. This motion conveys his intention about the wedding scene. The group's motion is inextricably interwoven by all such individual motions. Each individual motion may affect the motion of others and vice versa. For example, while you are talking to a group of people, you may instantly move your gaze to the particular person who just joined the group. Your gaze movement may trigger the gaze movements of the people who were paying attention to you, i.e., joint attention [75]. They may instantly focus on that person as well. Within this complex interaction between individuals, the group motion follows where their agreement is reached, e.g., what people are commonly interested in. The group motion reflects social agreement. Thus, individual and group motions are highly correlated with social context and motion estimation is a key component in understanding social context. We analyze individual and group motions evolving in a social scene from first person cameras.

Our overarching goal is to develop representations of social scenes to answer the following question: "what does it mean to understand a social scene?" As the first step to understand a social scene, we present a computational foundation of motion estimation associated with the social context from first person cameras (socially immersed cameras). This social understanding will address classic artificial intelligence and robotic task questions: how to navigate a social scene, how to anticipate group behavior, and how to communicate with people.

(a) Close up view

(b) Directionality

(c) Depth dependent uncertainty

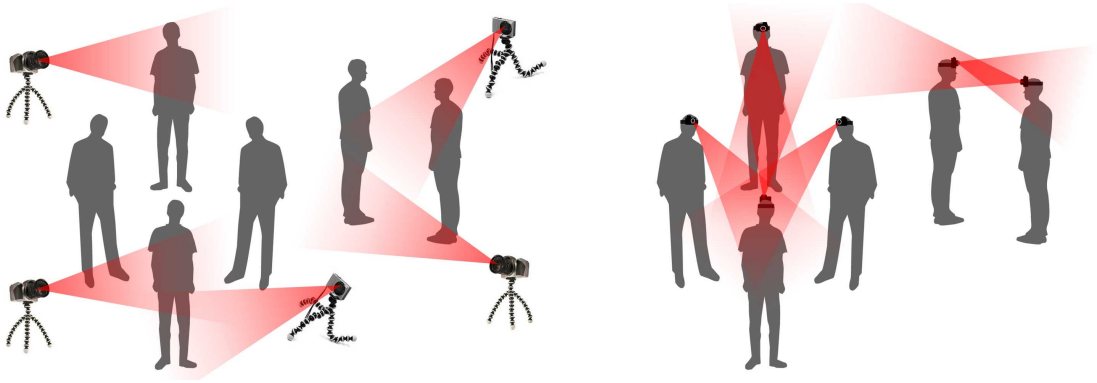
Figure 1.2: A camera inherently has two properties for measurement: directionality and depth dependent uncertainty. Given objects in (a), if the camera orientation is off, (b) some objects may become invisible. If the camera is placed far away from the objects, (c) the uncertainty of measurement is high. The number of pixels corresponding to the objects is small.

1.1 Why First Person Cameras?

A camera, in particular a first person camera, is an ideal sensor for social imaging. It provides rich information about a scene: it captures scene geometry, illumination, and texture, which no other single sensor can produce. It has been widely used to understand scene structure via 3D reconstruction [103], object recognition [35], and human affordance identification [45]. Consumer trends keep pushing down the cost of the cameras while camera performance progressively improves: small size, high resolution, low power consumption, and long duration of capture. Technical and economical improvements of the cameras have led to the wide proliferation of the cameras to the extent that most people carry at least one camera (such as a cell-phone camera) at all times.

Vision sensors have two properties: directionality and depth dependent uncertainty, as shown in Figure 1.2. The field of view is determined by camera position and orientation. To observe an object clearly, the camera must face the object. Uncertainty of the measurement from the camera is proportional to the depth of the object, i.e., distance between the object and the camera optical center. Objects near the camera appear larger and the number of pixels corresponding to the object is high. Details on the object can be clearly observed and measurements can be highly accurate. These two properties of vision sensors show that camera placement (position and orientation) is extremely important in measuring the scene. If we want to measure a specific object in the scene accurately, the camera must face the object and be placed as close as possible.

A social scene contains a few socially salient structures (what people are commonly interested in) that govern social context. In choosing the placement of cameras for social scene understanding, we need to consider two properties of social scenes. Socially salient structures are (1) sparsely distributed over the social scene, and (2) time-varying. Suppose people form several cliques in a party. People may focus on a particular person in each clique because he or she is famous or speaks loudly. While the space of the party may be large, most space is not occupied by these socially salient people. These socially salient structures also change over time. Some cliques are dissolved and some cliques are reformed in different places. These sparse and dynamic properties of the social salient structures are often observed in many scenes.



(a) Third person cameras (b) First person cameras

Figure 1.3: (a) Third person cameras sample a social scene from static points of view. This camera placement often cannot properly observe socially salient structures. (b) First person cameras sample the scene intelligently.

As one option for camera placement, third person cameras (outside cameras looking into a scene such as surveillance cameras) capture specific parts of a scene constantly, as shown in Figure 1.3(a). The vantage point is usually stationary unless there is human intervention. This does not reflect the properties of social scenes and vision sensors. To capture sparsely distributed socially salient structures, the third person cameras must be placed densely to cover the social space and to observe the structures as closely as possible. In many views from densely placed cameras, the structures may be invisible or imaged very small, which are useless measurements. More importantly, third person cameras are statically placed while scenes are usually dynamic. This limits the operating space. As shown in Figure 1.3(a), as it is usually impossible to predict social motion, third person cameras "sample" the scene from random static points of views.

A first person camera is a wearable camera looking out a scene from the first person perspective. It captures what the wearer sees. In contrast to third person cameras, first person cameras are ideally placed to image a social scene because their placement fully respects the properties of social scenes and vision sensors. First person cameras can naturally secure the best view. Humans intelligently find sources of interest and actively move themselves to see it from the best view point. This orients the camera attached to people directly to the source of social saliency. First person cameras can also minimize depth dependent uncertainty. Recognizing what happens in a clique is a hard task even for humans from outside of the cliques. First person cameras can observe the scene from a person who is actually involved in the interaction. Since people who interact with each other are likely located close by, the distance between the source of attention and the camera is minimized. These enable the first person cameras to capture moving and sparsely distributed socially salient structures with high accuracy and not suffer from limitations of the operating space. As shown in Figure 1.3(b), first person cameras sample the social scene intelligently, focusing on socially salient structures that are sparsely distributed and possibly moving from the best view. We leverage these advantages to estimate motion associated with social context.

1.2 Challenges

Classic scene understanding finds structural relationships in a scene. Many computer vision algorithms have addressed these tasks by applying geometric constraints on images. Unlike structural scene understanding, social scene understanding involves time-varying structure and subjective measurement. These two properties of social scenes make estimation of motion associated with social context difficult.

Time-varying structure: For static structures, measurements are timeless; what we measure now is consistent with what we measured in the past and what we will measure in the future. However, for time-varying structure, measurements vary across the time. There is only one opportunity to measure the structure at each time instant and there is no way to re-measure it because the structure changes across the time. Understanding the motion of time-varying structure is a challenging task because the structure must be fully captured by cameras at each time instant at once.

Subjective measurement: Different people have different interpretations of the same social scene. The interpretation is biased by their generation, culture, preference, profession, and background. Thus, what we measure from a person in the scene is subjective to the person. For instance, when you socialize with people at a wedding reception, you may join a clique because they may be your relatives or friends, they may share similar interests, or they may speak the same language. Even in the same clique, what people cognitively attend to is different from others. You may look at a person next to you because you think that he is interesting. A person on the other side may look at you because they find you attractive. Since each measurement is subjective, any result obtained from each measurement in isolation is subjective and cannot be directly used for an objective understanding of the social scene.

1.3 Our Approach

This thesis presents a method to estimate motion associated with social context (socially salient motion) from first person cameras. As addressed in Section 1.2, social scenes contain time-varying structure and its measurements are subjective. In Part I, we present a method to reconstruct time-varying structure in 3D from a series of 2D projections. We apply temporal and spatial constraints on the structure to estimate its motion. In Part II, we study how to derive an objective measurement from multiple subjective measurements. Even though a single social measurement is subjective, an interpretation approaches objectivity if many subjective measurements agree, statistically. We present an algorithm to reconstruct social saliency in 3D from these subjective measurements. As proposed work, we will integrate these studies of motion and social saliency to reconstruct socially salient motion in 3D in Part III. We will infer the causal relationships between social motion and social saliency and identify the motion that drives group behaviors. This reconstruction will provide the first step towards computationally understanding a social scene.

1.3.1 Part I: 3D Reconstruction of Motion

In Chapter 3, we present an algorithm to reconstruct a 3D trajectory of a moving point from its correspondence in a collection of 2D perspective images, given the 3D spatial pose and time of capture of the cameras that produced each image. Triangulation-based solutions do not apply, as multiple views of the point may not exist at each instant in time. A geometric analysis of the problem is presented and the problem is studied based on observability theory. For an observable system, a criterion, called reconstructibility, is defined to characterize the cases when reconstruction is accurate. The trajectory parameters are solved using linear least squares, and the estimate is refined by nonlinearly minimizing the reprojection error. A cross validation scheme is used to automatically select the number of basis vectors trajectories. This method can cope with missing data and uses a perspective camera model.

In Chapter 4, we present a method to reconstruct 3D trajectories specialized for human motion. An articulated trajectory is defined as a trajectory that remains at a fixed distance with respect to a parent trajectory. Spatial and temporal constraints are simultaneously applied in the form of a fixed 3D distance to the parent trajectory and smooth 3D motion. There exist two solutions that satisfy each instantaneous 2D projection and articulation constraint (a ray intersects a sphere at up to two locations) and we show that resolving this ambiguity by enforcing smoothness is equivalent to solving a binary quadratic programming problem.

1.3.2 Part II: 3D Reconstruction of Social Saliency

In Part II, we explore how to derive objective measurements about social saliency through the agreement of multiple subjective measurements. Although each social signal (head movement) is subjective to each person, the statistical agreement from multiple signals can produce objective measurements.

In Chapter 5, we present a method to estimate multiple gaze concurrences (socially salient points) in 3D from first person cameras based on a space-centric representation. First person cameras can capture what people look at and how they move. A gaze concurrence is a point in 3D where the gaze directions of multiple people intersect. It is a socially significant location because the attention of a clique is directly linked to that point. A 3D gaze ray is reconstructed by exploiting the fixed relationship between the primary gaze ray and the head-mounted camera pose, which is estimated via structure from motion. The variation of the eye orientation is modeled by a Gaussian distribution and the resulting gaze model produces a social saliency field in 3D. The number and 3D locations of the gaze concurrences via mode-seeking in the social saliency field are automatically estimated.

1.3.3 Part III: 3D Reconstruction of Socially Salient Motion (Proposed Work)

In Part III, we propose to reconstruct socially salient motion in 3D from first person cameras. Socially salient motion is motion emerging from social interactions, which often triggers group behavior. In Chapter 6, we propose a method to infer the relationship between social motion and social saliency and identify the socially salient motion. We will utilize 3D reconstruction

of motion in Part I and social saliency in Part II to reveal these relationships. We will find 3D human motion from 3D reconstructed trajectories by exploiting an articulation constraint of human body. Spatio-temporal representations of motion and saliency will be proposed and causal relationship between these representations will be estimated. This relationship will enable us to identify socially salient motion in 3D and predict how a group of people behave.

Chapter 2

Related Work

In this chapter, we review literature related to social scene understanding. A social scene involves two challenges as discussed in Section 1.2: time-varying structure and subjective measurement. This thesis proposes novel representations of social scenes for resolving these two challenges.

In particular, 3D reconstruction of scene geometry, called structure from motion discussed in Section 2.1, provides a computational basis to approach 3D reconstruction of motion associated with social context. In Section 2.2, we review a number of papers related to the first challenge; what is the fundamental ambiguity regarding time-varying structure and how it can be reconstructed in 3D. In Section 2.3, we explore how previous work was tackled the second challenge and how it is applied in a real world scenario.

2.1 Structural Scene Understanding: Structure from Motion

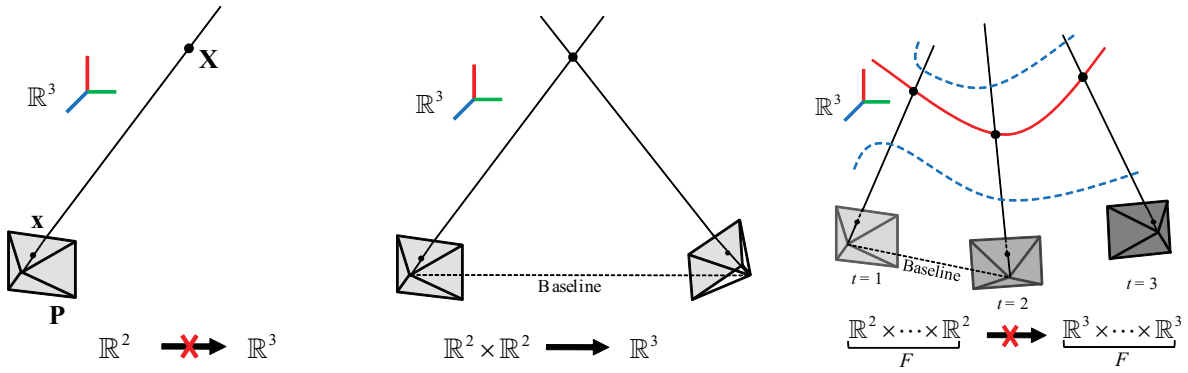
For a pinhole camera, a 3D point is perspectively projected onto a camera plane, which forms a 2D point as shown in Figure 2.1(a). The projection can be written as,

$$\lambda \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}, \tag{2.1}$$

where λ is a scalar, $\mathbf{X} \in \mathbb{R}^3$ is a 3D point, $\mathbf{x} \in \mathbb{R}^2$ is the corresponding 2D point measured in an image, and $\mathbf{P} \in \mathbb{R}^{3\times4}$ is a camera projection matrix. The camera projection matrix can be parameterized by $\mathbf{P} = \mathbf{KR}\begin{bmatrix} \mathbf{I}_3 - \mathbf{C} \end{bmatrix}$ where $\mathbf{I}_3$ is a 3 by 3 identity matrix, $\mathbf{R} \in \mathrm{SO}(3)$ is a camera rotation matrix, and $\mathbf{C}$ is a 3D camera position vector. $\mathbf{R}$ and $\mathbf{C}$ are called camera extrinsic parameters. $\mathbf{K}$ is a matrix of camera intrinsic parameters written as,

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.2}$$

where f_x and f_y are the focal lengths of the camera, and p_x and p_y are the image origin location. As shown in Equation (2.1), a 2D image measurement is formed by bilinear relationship between the 3D point and camera matrix. Also by projection, it loses 1 dimensional information, i.e., 3D$\rightarrow$2D, which appears in the form of the unknown scalar λ. There are an infinite number of 3D



(a) Single image (b) Static point (point in $\mathbb{R}^3$) (c) Moving point (trajectory in $\mathbb{R}^3$)

Figure 2.1: (a) Given $\mathbf{x}$, estimating $\mathbf{X}$ from a single image is fundamentally ambiguous because there are infinite number of solutions that satisfy the image measurement, $\mathbf{x}$. Any 3D point on the line between $\mathbf{X}$ and $\mathbf{x}$ can be a solution. (b) From two views, the 3D point can be triangulated without ambiguity. (c) From a series of images (projections), a point trajectory, $\mathbb{R}^3 \times \cdots \times \mathbb{R}^3$, in $\mathbb{R}^3$ also imaged to a series of points $\mathbb{R}^2$. Trajectory reconstruction is impossible without any constraint on the trajectory because any trajectory (dotted trajectories) passing through optical rays can be a solution. This is analogous to the fact that a static point reconstruction from single image is impossible without prior scene assumption.

points that satisfy the image measurement, $\mathbf{x}$. Any 3D point on the line between $\mathbf{x}$ and $\mathbf{X}$ can be a solution as shown in Figure 2.1(a). Therefore, given a single 2D image measurement, estimating the 3D point is impossible without prior assumptions about the scene. Structure from motion is to estimate 3D points and relative camera poses by exploiting multiple 2D images as shown in Figure 2.1(b). It enables us to reconstruct a 3D static scene and to understand the geometric relationship of the scene.

When correspondences are provided across 2D images in static scenes, the method proposed by Longuet-Higgins [66] estimated the relative camera poses and triangulates the point in 3D using epipolar geometry. He introduced the essential matrix, $\mathbf{E} \in \mathbb{R}^{3\times3}$, that constrains point correspondences between the images such that,

$$\begin{bmatrix} \mathbf{x}_1^\mathsf{T} & 1 \end{bmatrix} \mathbf{E} \begin{bmatrix} \mathbf{x}_2 \\ 1 \end{bmatrix} = 0, \tag{2.3}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are image measurements from image 1 and image 2, respectively, i.e., $\mathbf{x}_1$ in image 1 corresponds to $\mathbf{x}_2$ in image 2. This constraint holds for only 3D static points. Interestingly, the essential matrix encodes the 3D relative transform between two images and therefore, 3D camera pose can be extracted from the essential matrix. Once the camera pose is estimated, the 3D points can be triangulated as shown in Figure 2.1(b).

Tomasi and Kanade [109] addressed this problem in a different way. They decomposed bilinearly fused 2D measurement into camera pose and 3D point using a matrix factorization method based on orthographic projection. They concatenated all 2D measurements and factorize

to motion matrix and shape matrix as follows:

$$\begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1P} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{F1} & \cdots & \mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_F \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_P \end{bmatrix}, \quad \text{or}$$
$$\mathbf{W} = \mathbf{MS}, \tag{2.4}$$

where $\mathbf{W} \in \mathbb{R}^{2F \times P}$ is an image measurement matrix, $\mathbf{M} \in \mathbb{R}^{2F \times 3}$ is a matrix composed of all camera rotation matrices, and $\mathbf{S} \in \mathbb{R}^{3 \times P}$ is a shape matrix that contains all 3D points. P and F are the number of points and images, respectively. Only $\mathbf{W}$ is known. The rank of $\mathbf{W}$ is 3 and this rank constraint allows $\mathbf{W}$ to be factorized by $\mathbf{M}$ and $\mathbf{S}$ using singular value decomposition.

After these two seminal papers, structure from motion has been systematically developed. Triggs et al. [114] presented a bundle adjustment algorithm that simultaneously optimizes camera poses and 3D points, robustly and accurately. Also the SIFT feature descriptor proposed by Lowe [68] enables us to fully automate image matching. These two groundbreaking improvements in structure from motion allow us to apply large scale 3D reconstruction [2, 100, 103]. Geometric analysis regarding structure from motion is well summarized in [33, 50, 69].

2.2 3D Reconstruction of Time-Varying Structure

While a static point can be estimated by triangulation [66] as shown in Figure 2.1(b), in the case where the point may move between the capture of both images the triangulation method becomes inapplicable: the line segments mapped out by the baseline and the rays from each camera center to the point no longer form a closed triangle (Figure 2.1(c)). This problem is equivalent to the 3D point reconstruction from a single image as shown in Figure 2.1(a). Without prior assumption, reconstruction is impossible; there are infinite number of solutions. To disambiguate the solution, explicit constraints are necessary and 3D reconstruction of motion has been studied by applying various constraints. Constraints for a moving or deformable object can be classified by three domains: shape regularity, temporal regularity, and articulation regularity. Shape regularity approaches assume that nonrigid structure undergoes small regular deformations such as facial expression. Temporal regularity approach assumes that a point moves along a mathematically describable trajectory. Articulation regularity is applicable to point motion on rigid bodies articulated by connecting joints such as human body motion.

2.2.1 Shape Regularity

Computer vision, graphics and computer aided design research has facilitated shape regularity to model/estimate a 3D structure [21, 27, 59, 60, 90, 108, 111, 113]. Many shapes, such as face and cloth, do not deform randomly in reality. They follow physics law of deformation; each point on the surface of an object is connected with adjacent points and the shape changes while minimizing deformation energy. We will study various types of shape constraints in this section.

The seminal work of Bregler *et al.* [21] introduced linear shape models as a representation for nonrigid 3D structures, and demonstrated their applicability within the factorization-based

reconstruction paradigm of Tomasi and Kanade [109]. From linear shape basis, various methods to refine the factorization have been proposed. Torresani *et al.* [111, 113] introduced a method of trilinear optimization (camera motion, basis, coefficients) in an alternating fashion, Brand [19] integrated the optimization with a sophisticated initialization by enforcing the rank constraint on motion and by allowing minimal deformation of the shape. Later on, Paladini *et al.* [85] proposed a robust metric upgrade method which iterates solving unconstrained least squares for the bilinear system (camera motion and shapes) and projecting the solution onto metric motion manifold.

Extracting a linear shape basis from the measurement matrix suffers from instability of estimation [4, 20, 130, 131]. To address such problems, subsequent work has proposed numerous constraints and techniques to specify shape priors. Xiao *et al.* [130, 131] added a shape basis constraint which maximizes the basis independence for disambiguation under weak-perspective projection leading to a closed-form solution. Brand [20] pointed out the fragility of the closed-form solution in the presence of noise in the measurement matrix. Later on, Akhter *et al.* [4] discovered that the rank constraint and the orthonormality constraint on the camera motion matrix are sufficient to reconstruct structure up to a rotation. Meanwhile, a number of papers constrained the shape basis based on priors. Torresani *et al.* [112] introduced an algorithm to learn shape assuming that a Gaussian distribution of learned shape can represent the deformation of the structure. Torresani and Bregler [110] and Olsen and Bartoli [82] proposed a temporal smoothness prior on the shape basis and camera parameters (spatio-temporal constraint). Del Bue *et al.* [29] proposed a prior based on the rigidity of the majority of points, Del Bue [28] proposed a pre-computed prior which produces reliable reconstruction where there is degeneracy of motion, and Bartoli *et al.* [15] introduced a way to build the shape basis in a coarse-to-fine manner by iteratively decreasing reprojection error. Recently, Taylor *et al.* [107] proposed locally rigid structure from motion by allowing minimal deformation of triangles formed by three adjacent points in 3D, and Fayad *et al.* [34] introduced piecewise reconstruction by dividing the surface into overlapping patches. The strong assumption of known correspondences was relaxed using a weak prior of structure modeled as a Gaussian mixture model [97] and by solving a mixed integer quadratic problem using the Branch and Bound method [98]. A detailed survey by Salzmann and Fua [96] summarizes subsequent work on nonrigid structure from motion.

2.2.2 Temporal Regularity

The principal work in 'triangulating' moving points from a series of images is by Avidan and Shashua [11], who coined the term *trajectory-triangulation*. They demonstrated two cases where a moving point can be reconstructed: (1) if the point moves along a line, or (2) if the point moves along a conic section. This inspired a number of approaches of geometrically constrained trajectory recovery. Han and Kanade [47] showed the factorization method of a moving object with constant velocity by exploiting the fact that the rank of the measurement matrix is six. Shashua and Wolf [99] and Wexler and Shashua [128] introduced homography tensors to represent a point moving on the plane. As an integration of the algebraic curve representation, Wolf and Shashua [129] classified different manifestations of related problems, analyzing them as projections from $\mathbb{P}^N$ to $\mathbb{P}^2$. Kaminski and Teicher [58] extended these ideas to a general trajectory represented by a family of hypersurfaces in the projective space $\mathbb{P}^5$. Sidenbladh et al. [51] ap-

plied a constant velocity model to constrain a smooth motion and Torresani and Bregler [110] applied spatial and temporal constraints via a rank constraint.

Similar to the factorization based approach of the shape regularity discussed in Section 2.2.1, Akhter *et al.* [3, 6] proposed analyzing each trajectory as a linear combination of basis trajectories. They proposed the use of the Discrete Cosine Transform (DCT) as a basis, and applied factorization techniques to estimate nonrigid structure. While shape based approaches have to estimate camera motion, basis, and coefficients simultaneously, Akhter *et al.* [3, 6] reduce the complexity of a trilinear problem into a bilinear problem using pre-defined trajectory basis which can represent an arbitrary trajectory compactly. Reduction of the problem complexity allows them to estimate the motion and coefficients more robustly.

2.2.3 Articulation Regularity

As humans are of particular interest, several papers consider priors based on the factorization method: Costeira and Kanade [25] proposed a factorization method for multiple rigid bodies using 2D trajectories in an image stream. Yan and Pollefeys [132] used the fact that the articulation subspace is the intersection of all rigid body subspaces and discussed the physical meaning of the articulation subspace. Using the articulation constraint, they devised an automatic algorithm for building a kinematic constraint by clustering moving points [133].

Human pose estimation from a single image by applying a spatial constraint (skeletal structure) was proposed by Taylor [106] (parameterization of limb lengths by a scalar), by Barron and Kakadiaris [14] (joint motion constraint from the anthropometric statistics), by Parameswaran and Chellappa [86] (camera pose estimation from head orientation and rigidity of torso), and by Agarwal and Triggs [1] (silhouette based regression).

Human motion estimation from an image sequence of a monocular camera has been studied as an extension of human pose estimation. Two popular approaches have been explored: the data driven approach and the physics based approach. Data driven approaches learn low dimensional subspace or latent variables that control underlying human skeletal motion fully using motion capture data or annotated video data. Sidenbladh *et al.* [51] applied a Bayesian framework for 3D human pose tracking using a generative model of the human body and a prior distribution defined by a temporal dynamics model. Howe *et al.* [53] showed Baysian learning, Choo and Fleet [24] sampled high dimensional training space from hybrid Monte Carlo method, and Urtasun *et al.* [115] used Principle Coordinate Analysis (PCA) for learning of specific motion (e.g. walking and golfing). Like Taylor's work [106], Wei and Chai [125] introduced a geometric solution of motion reconstruction using the bone symmetric constraint from biomechanical data. Valmadre and Lucey [116] discussed the validity of Wei and Chai [125]'s work and extended their algorithm using a structure from motion scheme. Recently, physics based approaches have received attention. Brubaker *et al.* [22] have shown reconstruction of a bipedal locomotion from a dynamical model and Vondrak *et al.* [122] have applied multibody dynamics simulation to infer the most plausible human motion in 3D. Wei and Chai [126] have built an interactive system that integrates a dynamical model to capture motion from a video.

2.2.4 Relation to Our Work

Most previous nonrigid structure from motion algorithms rely on factorizing the measurement matrix. The primary limitation of these factorization-based methods is: (1) the assumption of an orthographic camera, and (2) their inability to handle missing information. Several papers have relaxed the constraint of orthography, such as Hartley and Vidal [49], Vidal and Abretske [119], and Zhu *et al.* [135]. The work by Torresani *et al.* [113] can handle missing data using the rank constraint of the flow matrix. However, all these algorithms remain unstable and have been demonstrated to work only for constrained deformation of objects like faces.

In Chapter 3, we present a method to reconstruct a moving point from a series of 2D projections. Unlike previously proposed methods we do not pursue a factorization based solution. Instead we propose a linear solution to reconstruct a moving point inspired by the Direct Linear Transform algorithm [50]. It is able to handle problems like missing data (due to occlusion and matching failure) and estimation instability. An analysis is presented which geometrically describes the reconstruction problem as fundamentally restricted by the relation between the motion of the camera center, the motion of a scene point trajectory, and the trajectory basis.

In Chapter 4, we present a method to reconstruct human motion. Unlike previous methods based on the articulation regularity, our approach relies purely on a geometric interpretation of the articulation constraint by parameterizing a trajectory in a way that satisfies both spatial and temporal constraints simultaneously. It can reconstruct activity independent motion which other methods cannot.

2.3 3D Reconstruction of Social Saliency

Understanding how we socially interact with each other has been a long-standing focus of the social sciences. With the growth of computing and computer science, a significant research thrust has emerged in building computational models for understanding social interactions and their related dynamics, driven by efforts in psychology and sociology. Nevertheless, measuring social interaction is not a trivial task because it involves many subjective measurements. In this thesis, we claim that when many subjective measurements agree, the accordant measurement approaches objectivity. This implies that many measurements are required to understand social interaction from either multiple time instances (long term measurements) or multiple perspectives (crowd measurements). For example, by measuring how frequently you interact with a group, the measure of the bond strength in social networks can be objectively determined. By measuring how many people look at a person, instantaneous popularity can be objectively determined.

2.3.1 Long Term Measurement

Even if an instantaneous measurement is subjective, the time accumulated measurement can approach the objective measurement. Theories of social networks has been grounded on this measurement. They build a graphical model of social structure statistically from time accumulated social interaction. Pool and Kochen [93] proposed the small world conjecture that people

in the world are connected by six degrees of separation and Milgram [73] conducted experiments to verify this conjecture and examined the average path length for social networks of people in the United States. Based on the small world assumption, Watts and Strogatz [124] proposed a method to construct a social graph, mathematically, which preserves local and global properties. Newman et al. [80] adopted a random graph to model social networks and presented a method to solve for the model exactly given the degree distribution. Eubank et al. [31] introduce a method to infer social dynamics from spatio-temporal information of individuals interactions and Lauw et al. [62] also propose an algorithm to reconstruct a large-scale social network by mining spatio-temporal events. Recently, Gilbert and Karahalios [41] studied and predicted the strength of connections (tie-strength) over a social network. As noted, most of this work has expanded to much larger scales with the growth of social networking.

2.3.2 Crowd Measurement

When multiple subjective measurements agree, the accordant measurement also approaches objectivity. In a social scene, humans transmit and respond to many different social signals when they interact with others. Social signals such as facial expression, gesture, and gaze movement are measurable signals from vision inputs but the signals are highly subjective. Each person can transmit different signals given the same scene depending on gender, culture, preference, intelligent, role, and background. However, when many people are commonly interested in something, each subjective measurement collectively forms objective group measurement, e.g., most people pay attention to the bride when she marches in a wedding.

Among social signals, gaze direction is one of the most effective visual signals because it usually indicates what the individual is interested in. In this context, gaze direction estimation has been widely studied in robotics, human-computer interaction, and computer vision [12, 30, 40, 44, 52, 65, 76, 78, 81, 94, 95, 105, 123]. Gaze direction can be precisely estimated by the eye orientation. Wang and Sung [123] presented a system that estimates the direction of the iris circle from a single image using the geometry of the iris. Guestrin and Eizenman [44] and Hennessey and Lawrence [52] utilized corneal reflections and the vergence of the eye to infer the eye geometry and its motion, respectively. A head-mounted eye tracker is often used to determine the eye orientation [65, 105]. Although all these methods can estimate highly accurate gaze direction, either they can be used only in a laboratory setting or the device occludes the viewer's field of view.

While the eyes are the primary source of gaze direction, Emery [30] notes that the head orientation is a strong indication of the direction of attention. For head orientation estimation, there are two approaches: outside-in and inside-out [127]. An outside-in system takes as input a third-person view image from a particular vantage point and estimates face orientation based on a face model. Murphy-Chutorian and Trivedi [78] have summarized this approach. Geometric modeling of the face has been used to orient the head by Gee and Cipolla [40] and Ballard and Stockman [12]. Rae and Ritter [94] estimated the head orientation via neural networks and Robertson and Reid [95] presented a method to estimate face orientation by learning 2D face features from different views in a low resolution video. With these approaches a large number of cameras would need to be placed to cover a space large enough to contain all people. Also, the size of faces in these videos is often small, leading to biased head pose estimation depending

on the distance from the camera. Instead of the outside-in approach, an inside-out approach estimates head orientation directly from a head-mounted camera looking out at the environment. Munn and Pelz [76] and Takemura et al. [105] estimated the head-mounted camera motion in 3D by feature tracking and visual SLAM, respectively. Pirri et al. [91] presented a gaze calibration procedure based on the eye geometry using 4 head-mounted cameras. Our method leverages this approach which does not suffer from space limitations and biased estimation.

Gaze in a group setting has been used to identify social interaction or to measure social behavior. Stiefelhagen [104] and Smith et al. [101] estimated the point of interest in a meeting scene and a crowd scene, respectively. Bazzani et al. [16] introduced the 3D representation of the visual field of view, which enabled them to locate the convergence of views. Cristani et al. [26] adopted the F-formation concept that enumerates all possible spatial and orientation configurations of people to define the region of interest. Fathi et al. [32] showed how social interactions are detected and recognized from first person cameras.

2.3.3 Relation to Our Work

While long term measurement requires that the measurement must be consistent across time, a social salient structure in a social scene is often time-varying. Instead, in a social scene, many people are simultaneously involved and thus, measurements from multiple perspectives (crowd measurements) are a viable approach. Therefore, in Chapter 5, we apply crowd measurement to achieve objectivity. 3D gaze concurrences where multiple gaze directions converge are locations where multiple people are commonly interested in, i.e., subjectivity approaches objectivity. We use head orientation to estimate a gaze direction and find intersections of the gaze directions from many people. This enables us to estimate the socially salient region where people are interested in and its motion in 3D.

Part I

3D Reconstruction of Motion

Chapter 3

3D Reconstruction of a Moving Point from First Person Cameras

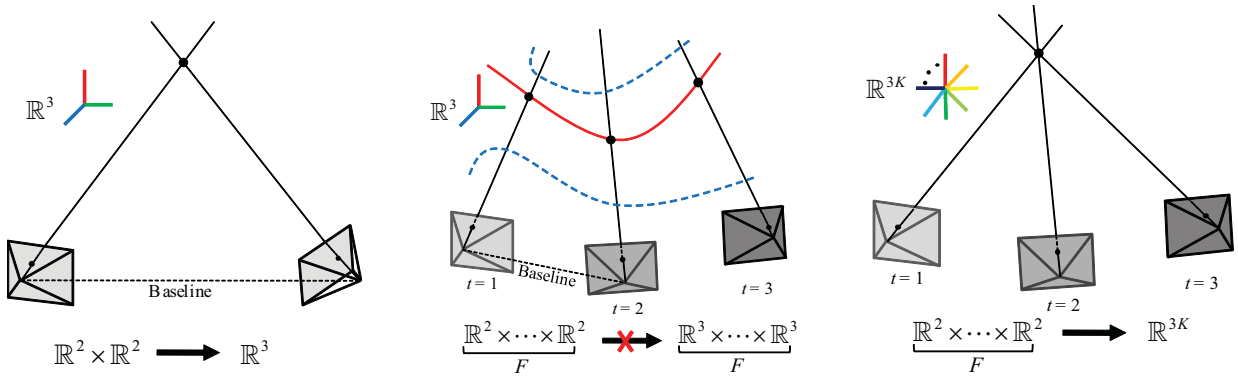
3.1 Introduction

In a social scene, a socially salient structure undergoes significant deformation across time, such as facial expressions and body motions. This time-varying property of socially salient structures makes social scene understanding difficult. In this chapter, we resolve this difficulty via 3D reconstruction of the time-varying structure from first person cameras. We represent each point motion on the time-varying structure as a trajectory and present an algorithm to reconstruct the 3D trajectory of a moving point from a collection of 2D perspective projections by applying a temporal constraint.

Without making prior assumptions about scene structure, it is impossible to reconstruct a 3D scene from a single image. Binocular stereoscopy is a solution used by both biological and artificial systems to localize the position of a point in 3D via correspondences in two views. Classic triangulation used in stereo reconstruction is geometrically well-posed as shown in Figure 3.1(a). The rays connecting each image location to its corresponding camera center intersect at the true 3D location of the point — this process is called triangulation as the two rays map out a triangle with the baseline that connects the two camera centers. The triangulation constraint does not apply when the point moves between image captures, as shown in Figure 3.1(b). This case abounds as most artificial vision systems are monocular and most real scenes contain moving elements.

The 3D reconstruction of a trajectory is directly analogous to monocular image reconstruction: it is impossible to reconstruct a moving point without making some assumptions about the way it moves. In this chapter, we represent the 3D trajectory of a moving point as a compact linear combination of a trajectory basis and demonstrate that, under this model, we can recover the 3D motion of the point from a series of perspective projections. By posing the problem in this way we generalize the problem of triangulation, which is a mapping from $\mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^3$ to 3D *trajectory* reconstruction, as a mapping $\mathbb{R}^2 \times \cdots \times \mathbb{R}^2 \rightarrow \mathbb{R}^{3K}$, where $3K$ is the number of trajectory basis vectors required to represent the 3D point trajectory[1]. The resulting optimization can be solved using linear least squares providing stable and accurate estimates in the presence

[1]Related observations have been made in [49, 99].



(a) Static point (point in $\mathbb{R}^3$) (b) Moving point (trajectory in $\mathbb{R}^3$) (c) Moving point (point in $\mathbb{R}^{3K}$)

Figure 3.1: (a) A point in $\mathbb{R}^3$ is mapped to $\mathbb{R}^2$. From two views, the 3D point can be triangulated. (b) From a series of images (projections), a point trajectory, $\mathbb{R}^3 \times \cdots \times \mathbb{R}^3$, in $\mathbb{R}^3$ also imaged to a series of points $\mathbb{R}^2$. Trajectory reconstruction is impossible without any constraint on the trajectory because any trajectory (dotted trajectories) passing through optical rays can be a solution. This is analogous to the fact that a static point reconstruction from single image is impossible without prior scene assumption. (c) The compact linear trajectory basis representation allows us to transform a point trajectory in $\mathbb{R}^3$ to a point in $\mathbb{R}^{3K}$.

of missing data.

The stability of classic triangulation is known to depend on the baseline between camera centers [50]. We characterize the instability encountered when interference occurs between the trajectory of the point and the trajectory mapped out by successive camera centers. We demonstrate that 3D trajectory reconstruction is fundamentally limited by the relationship between the trajectory of the point, the trajectory of successive camera centers, and the trajectory basis. We characterize the cases when trajectory reconstruction is possible by using observability theory. For an observable system, a measure called *reconstructibility* is defined, which describes the accuracy of reconstruction for a particular trajectory basis, given a 3D point trajectory and a 3D camera center trajectory.

Since different points may undergo different degrees of motion, we present a cross validation scheme to independently select the number of basis vectors for each trajectory. The reconstruction algorithm and the cross validation scheme are combined in a practical algorithm for the reconstruction of multiple 3D trajectories from a collection of non-coincidental images.

3.2 Method

3.2.1 Linear Reconstruction of a 3D Point Trajectory

For a static point in 3D projective space, correspondences across a pair of images enable us to triangulate as shown in Figure 3.1(a). Classic triangulation solves for a 3D point from an overconstrained system because there are three unknowns (3D coordinate of the point) while the number of equations is $2F$, where F is the number of images (projections). As was the case with

static point projection, if $2F \geq 3K$ where $3K$ is the number of 3D trajectory parameters, solving for a 3D trajectory becomes an overconstrained problem as shown in Figure 3.1(c). Using this observation, we develop a linear solution for reconstructing a point trajectory given the relative poses of the cameras and the time instances the images were captured.

For a given i^{th} camera projection matrix, $\mathbf{P}_i \in \mathbb{R}^{3\times4}$, let a point in 3D, $\mathbf{X}_i = \begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^\mathsf{T}$, be imaged as $\mathbf{x}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^\mathsf{T}$. The index i used represents the i^{th} time sample. This projection is defined up to scale,

$$\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \simeq \mathbf{P}_i \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix}, \text{ or } \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_i \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix} = \mathbf{0}, \tag{3.1}$$

where $[\cdot]_\times$ is the skew symmetric representation of the cross product [50]. This can be rewritten as an inhomogeneous equation,

$$\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_{i,1:3}\mathbf{X}_i \;=\; -\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_{i,4} \;,$$

where $\mathbf{P}_{i,1:3}$ and $\mathbf{P}_{i,4}$ are the matrices made of the first three columns and the last column of $\mathbf{P}_i$, respectively, or simply as $\mathbf{Q}_i\mathbf{X}_i = \mathbf{q}_i$, where,

$$\mathbf{Q}_i = \left(\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_{i,1:3} \right)_{1:2}, \; \mathbf{q}_i = -\left(\begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix}_\times \mathbf{P}_{i,4} \right)_{1:2},$$

and $(\cdot)_{1:2}$ is the matrix made of two rows from $(\cdot)$. By taking into account all time instances, a closed form for the 3D point trajectory, $\mathbf{X}$, can be formulated as,

$$\begin{bmatrix} \mathbf{Q}_1 & & \\ & \ddots & \\ & & \mathbf{Q}_F \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_F \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_F \end{bmatrix}, \text{ or } \mathbf{QX} = \mathbf{q}, \tag{3.2}$$

where F is the number of time samples in the trajectory. Since Equation (3.2) is an underconstrained system (i.e., $\mathbf{Q} \in \mathbb{R}^{2F\times3F}$), there are an infinite number of solutions for a given set of measurements (2D projections). There are many ways to constrain the solution space in which $\mathbf{X}$ lies. One way is approximating the point trajectory using a linear combination of any trajectory basis that can describe it as,

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1^\mathsf{T} & \cdots & \mathbf{X}_F^\mathsf{T} \end{bmatrix}^\mathsf{T} \approx \boldsymbol{\Theta}_1\beta_1 + \ldots + \boldsymbol{\Theta}_{3K}\beta_{3K} = \boldsymbol{\Theta}\boldsymbol{\beta}, \tag{3.3}$$

where $\boldsymbol{\Theta}_j \in \mathbb{R}^{3F}$ is a trajectory basis vector, $\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\Theta}_1 & \ldots & \boldsymbol{\Theta}_{3K} \end{bmatrix} \in \mathbb{R}^{3F\times3K}$ is the trajectory basis matrix, $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \ldots & \beta_{3K} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{3K}$ are the parameters or coefficients of a point trajectory, and K is the number of bases per coordinate.

If the trajectory basis are known *a priori* [3], as is the case with the DCT basis, this linear map between the point trajectory and basis enables us to formulate a linear solution. By plugging Equation (3.3) into Equation (3.2), we can derive an overconstrained system by choosing K such that $2F \geq 3K$,

$$\mathbf{Q}\boldsymbol{\Theta}\boldsymbol{\beta} \;=\; \mathbf{q}. \tag{3.4}$$

Equation (3.4) is a linear least squares system for reconstructing a point trajectory, $\boldsymbol{\beta}$, which provides an efficient, numerically stable, and globally optimal solution. $\boldsymbol{\beta}$ is the coefficient of the trajectory based on measurements and known camera poses embedded in $\mathbf{Q}$ and $\mathbf{q}$ and known trajectory basis, $\boldsymbol{\Theta}$.

If there are missing data by self-occlusion or measurement noise, corresponding rows of $\mathbf{Q}$ and $\mathbf{q}$ may be dropped in Equation (3.4). As long as resulting $\mathbf{Q}\boldsymbol{\Theta}$ matrix satisfies the least squares criterion, i.e., $2\widehat{F} > 3K$ where $\widehat{F}$ is the remaining number of measurements, the estimation of $\boldsymbol{\beta}$ is robust. This allows us to handle the problem of missing data.

3.2.2 Selection of The Number of Basis Vectors

Our approach uses a truncated DCT basis which requires the selection of the number of basis vectors, K. In [87], the number of basis vectors was manually tuned and all trajectories were reconstructed with the same number of basis. The number of basis vectors controls the complexity of the trajectory motion: for example, in the dance scene shown in Figure 3.9, if a point motion is complex (like the motion of a hand), it requires higher K; if a point motion is simpler (like a point on the torso) it requires lower K. If the number of basis vectors is too high, the algorithm overfits the trajectory in presence of measurement noise, and conversely, if it is too low, the reconstructed trajectory cannot describe the original point motion. In this section we present an approach to automatically select K_i for the i^{th} trajectory rather than manually setting a global value of K.

To select the number of basis vectors automatically and individually, we use an N-fold cross validation scheme to check the consistency[2] of the reconstructed trajectory. The 2D trajectory is divided into N sets such that each set contains F/N samples which are uniformly distributed in time across the 2D trajectory. When the j^{th} set is considered, the reprojection error, e_j, is evaluated from a 3D trajectory reconstructed from the rest of the $N-1$ sets for a given K_i. This is iterated until all N sets are tested. When K_i is too high, the trajectory overfits to measurement noise, which results in high reprojection error. When K_i is too low, the reprojection error is also high because of limited expressiveness of the basis. We choose the number of basis vectors for the i^{th} trajectory, which minimizes reprojection error, i.e.,

$$K_i^* \;=\; \underset{K_i}{\operatorname{argmin}} \sum_{j=1}^{N} e_j(K_i), \qquad K_i = 1, 2, \cdots, \lfloor 2F/3 \rfloor, \tag{3.5}$$

where $\lfloor \cdot \rfloor$ is the floor operator (the largest integer not greater than $\cdot$). Figure 3.5(c) illustrates an example of reprojection error as the number of basis vectors increases. When $K_i = 12$, the most consistent trajectory through all image measurements (minimum reprojection error) is achieved.

3.2.3 Trajectory Refinement

Trajectory reconstruction from Equation (3.4) minimizes the algebraic error [50]. However, the solution, $\boldsymbol{\beta}$, is not necessarily the maximum likelihood solution under Gaussian measurement

[2]Since we do not have labeled training trajectories in real scene, we look for the value of K_i which is the most consistent through all measurements.

noise. We refine the linearly reconstructed trajectory by minimizing the reprojection error, i.e.,

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{F} \left(\frac{\mathbf{P}_i^1 \mathcal{X}}{\mathbf{P}_i^3 \mathcal{X}} - x_i \right)^2 + \left(\frac{\mathbf{P}_i^2 \mathcal{X}}{\mathbf{P}_i^3 \mathcal{X}} - y_i \right)^2, \text{ where } \mathcal{X} = \begin{bmatrix} \boldsymbol{\Theta}(t_i)\boldsymbol{\beta} \\ 1 \end{bmatrix}, \tag{3.6}$$

t_i is the time instance when $\mathbf{P}_i$ is taken, $\boldsymbol{\Theta}(t_i)$ is the trajectory basis evaluated at t_i, and $\mathbf{P}^j$ is the j^{th} row of the matrix $\mathbf{P}$.

3.3 Geometric Analysis of 3D Trajectory Reconstruction

Empirically, the point trajectory reconstruction approaches the ground truth point trajectory when the camera motion is fast or random. Conversely, if the camera moves slowly or smoothly, the solution tends to deviate from the ground truth. In this section, we analyze stability of trajectory reconstruction from Equation (3.4) by considering the geometric relation between the trajectory basis, and point and camera trajectories. We link trajectory basis representation to linear dynamical models and categorize a solution as either *observable* or *unobservable*. Also within an observable system, we define a measure of reconstruction accuracy, *reconstructibility*, which enables us to precisely characterize when accurate reconstruction of a 3D trajectory is possible.

3.3.1 Geometry of Trajectory Basis, Point, and Camera Trajectories

Let $\mathbf{X}$ and $\widehat{\mathbf{X}}$ be a ground truth trajectory and an estimated point trajectory respectively. The camera matrix can, without loss of generality, be normalized by intrinsic and rotation matrices, $\mathbf{K}$ and $\mathbf{R}$, respectively, (as all camera matrices are known), i.e., $\mathbf{R}_i^{\intercal}\mathbf{K}_i^{-1}\mathbf{P}_i = \left[\begin{array}{c|c} \mathbf{I}_3 & -\mathbf{C}_i \end{array} \right]$, where $\mathbf{P}_i = \mathbf{K}_i\mathbf{R}_i \left[\begin{array}{c|c} \mathbf{I}_3 & -\mathbf{C}_i \end{array} \right]$, $\mathbf{C}_i$ is the camera center, and $\mathbf{I}_3$ is a 3×3 identity matrix. This follows from the fact that triangulation and 3D trajectory reconstruction are both geometrically unaffected by the rotation of the camera about its center. All $\mathbf{P}_i$ subsequently used in this analysis are normalized camera matrices, i.e., $\mathbf{P}_i = \left[\begin{array}{c|c} \mathbf{I}_3 & -\mathbf{C}_i \end{array} \right]$. Then, a measurement is a projection of $\mathbf{X}_i$ onto the image plane from Equation (3.1). Since Equation (3.1) is defined up to scale, the measurement, $\mathbf{x}_i$, can be replaced[3] as follows,

$$\left[\mathbf{P}_i \begin{bmatrix} \mathbf{X}_i \\ 1 \end{bmatrix} \right]_{\times} \mathbf{P}_i \begin{bmatrix} \widehat{\mathbf{X}}_i \\ 1 \end{bmatrix} = 0. \tag{3.7}$$

Plugging in $\mathbf{P}_i = \left[\begin{array}{c|c} \mathbf{I}_3 & -\mathbf{C}_i \end{array} \right]$ results in, $[\mathbf{X}_i - \mathbf{C}_i]_{\times} \left(\widehat{\mathbf{X}}_i - \mathbf{C}_i \right) = 0$, or equivalently,

$$[\mathbf{X}_i - \mathbf{C}_i]_{\times} \widehat{\mathbf{X}}_i = [\mathbf{X}_i]_{\times} \mathbf{C}_i. \tag{3.8}$$

To satisfy Equation (3.8), $\widehat{\mathbf{X}}_i$ has to lie in the space spanned by $\mathbf{X}_i$ and $\mathbf{C}_i$, or $\widehat{\mathbf{X}}_i = a_1\mathbf{X}_i + a_2\mathbf{C}_i$. It can be easily verified that $a_2 = 1 - a_1$ by substituting in Equation (3.8). Thus, the solution of Equation (3.8) is,

$$\widehat{\mathbf{X}}_i = a_i\mathbf{X}_i + (1 - a_i)\mathbf{C}_i, \tag{3.9}$$

[3]We assume that there is no measurement noise.

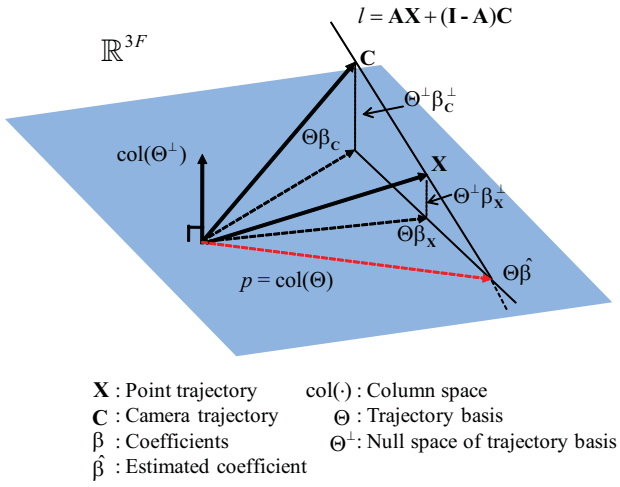


Figure 3.2: Geometric illustration of the least squares solution of Equation (3.4). The estimated trajectory $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$ is placed on the intersection between l containing the camera trajectory space and the point trajectory, and the p space spanned by the column space of the trajectory basis matrix, $\mathrm{col}(\boldsymbol{\Theta})$.

where a_i is an arbitrary scalar. Geometrically, Equation (3.9) is a constraint for the perspective camera model that enforces the solution to lie on the ray joining the camera center and the point in 3D. By generalizing the i^{th} point to a point trajectory, Equation (3.9) becomes,

$$\widehat{\mathbf{X}} = \mathbf{AX} + (\mathbf{I} - \mathbf{A})\mathbf{C}, \tag{3.10}$$

where $\mathbf{A} = \mathbf{D} \otimes \mathbf{I}_3$[4]. From Equation (3.3), Equation (3.10) can be rewritten as $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} \approx \mathbf{AX} + (\mathbf{I} - \mathbf{A})\mathbf{C}$ where $\widehat{\boldsymbol{\beta}}$ is the estimated parameter.

Figure 3.2 illustrates the geometry of the solution of Equation (3.4). Let the subspace, p, be the space spanned by the column space of the trajectory basis matrix, $\mathrm{col}(\boldsymbol{\Theta})$. The solution $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$, has to simultaneously lie on the hyperplane l, which contains the camera trajectory and the point trajectory, and must lie in $\mathrm{col}(\boldsymbol{\Theta})$. Thus, $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$ is the intersection of the hyperplane l and the subspace p. In the figure, note that the line and the plane are a conceptual 3D vector space representation for the $3F$-dimensional space. The camera center trajectory, $\mathbf{C} = \begin{bmatrix} \mathbf{C}_1^{\mathsf{T}} & \dots & \mathbf{C}_F^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$, and the point trajectory, $\mathbf{X}$, are projected onto $\mathrm{col}(\boldsymbol{\Theta})$ as $\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}}$ and $\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}}$, respectively. From this point of view, we want $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$ to be as close as possible to $\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}}$.

3.3.2 Relationship Between Trajectory Reconstruction and Linear Dynamical Systems

If the dynamics governing a 3D point motion is linear, the trajectory of the point can be modeled by a linear dynamical system. In this section, we briefly review the theory of linear dynamical systems and link it to trajectory reconstruction, with the goal of extending observability theory in linear dynamical systems to the trajectory basis representation.

[4] $\otimes$ is the Kronecker product and $\mathbf{D}$ is a diagonal matrix which consists of $\{a_1, \cdots, a_F\}$.

If a 3D point moves according to a discrete linear dynamical model under the first order Markov assumption,

$$\mathbf{X}_{i+1} = \mathcal{A}\mathbf{X}_i \tag{3.11}$$
$$\mathbf{x}_i = \mathcal{C}_i\mathbf{X}_i \tag{3.12}$$

where $\mathbf{X}_i \in \mathbb{R}^3$ and $\mathbf{x}_i \in \mathbb{R}^2$ are the state which is a 3D point and the measurement at i^{th} time instant, respectively, and $\mathcal{A} \in \mathbb{R}^{3\times3}$ and $\mathcal{C}_i \in \mathbb{R}^{2\times3}$ are a linear dynamical model matrix and affine camera matrix[5], respectively, if there is no control input. We begin with an affine camera model and will generalize to a perspective camera model. By stacking all measurement, it results in following linear system:

$$\begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_F \end{bmatrix} = \begin{bmatrix} \mathcal{C}_1 \\ \mathcal{C}_2\mathcal{A} \\ \mathcal{C}_3\mathcal{A}^2 \\ \vdots \\ \mathcal{C}_F\mathcal{A}^{F-1} \end{bmatrix} \mathbf{X}_1 = \begin{bmatrix} \mathcal{C}_1 & & \\ & \ddots & \\ & & \mathcal{C}_F \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 \\ \mathcal{A} \\ \mathcal{A}^2 \\ \vdots \\ \mathcal{A}^{F-1} \end{bmatrix} \mathbf{X}_1. \tag{3.13}$$

By inverting $\begin{bmatrix} \mathcal{C}_1^\mathsf{T} & (\mathcal{C}_2\mathcal{A})^\mathsf{T} & \dots & (\mathcal{C}_F\mathcal{A}^{F-1})^\mathsf{T} \end{bmatrix}^\mathsf{T}$, the initial state, $\mathbf{X}_1$, can be estimated given measurements when it is invertible.

When the systems follow the Kth order Markov assumption, Equation (3.11) can be written as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{X}_{i+1} \\ \vdots \\ \mathbf{X}_{i+K-1} \\ \mathbf{X}_{i+K} \end{bmatrix} &= \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_3 & & \\ & & \ddots & \\ & & & \mathbf{I}_3 \\ \mathcal{A}_1 & \mathcal{A}_2 & \dots & \mathcal{A}_K \end{bmatrix} \begin{bmatrix} \mathbf{X}_i \\ \mathbf{X}_{i+1} \\ \vdots \\ \mathbf{X}_{i+K-1} \end{bmatrix} \\ &= \mathbf{A}\mathbf{X}_{(3(i-1)+1):3(i+K-1)} \\ &= \mathbf{A}^2\,\mathbf{X}_{(3(i-2)+1):3(i+K-2)} \\ &= \mathbf{A}^i\,\mathbf{X}_{1:3K}, \end{aligned} \tag{3.14}$$

where $\mathbf{X}_{i:j}$ is a truncated vector from i^{th} element to j^{th} element of $\mathbf{X}$. Then,

$$\mathbf{X}_{i+K} = \begin{bmatrix} \mathbf{0}_{3\times3(K-1)} & \mathbf{I}_3 \end{bmatrix} \mathbf{A}^i\,\mathbf{X}_{1:3K} = \mathcal{I}\mathbf{A}^i\,\mathbf{X}_{1:3K}. \tag{3.15}$$

From this relation, Equation (3.13) can be written as,

$$\begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_F \end{bmatrix} = \begin{bmatrix} \mathcal{C}_1 & & \\ & \ddots & \\ & & \mathcal{C}_F \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3K} \\ \mathcal{I}\mathbf{A} \\ \vdots \\ \mathcal{I}\mathbf{A}^{F-K} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_K \end{bmatrix} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{X}_{1:3K}, \tag{3.16}$$

[5]Since there is camera motion, the measurement mapping, $\mathcal{C}_i$, changes over time but as long as $\mathcal{C}_i$ is known, the linear dynamical systems still holds.

where $\mathbf{X}_{1:3K}$ and $\mathbf{\Lambda}$ can be considered as a trajectory parameter and the trajectory basis, respectively, because

$$\mathbf{X} = \mathbf{\Lambda}\mathbf{X}_{1:3K}. \tag{3.17}$$

For a perspective camera model, we can replace $\mathbf{x}_i$ and $\mathcal{C}_i$ with $\mathbf{q}_i$ and $\mathbf{Q}_i$ in Equation (3.2). Then, Equation (3.16) is equivalent to Equation (3.4). This equivalent relationship between trajectory reconstruction from Equation (3.4) and linear dynamical systems implies that a point motion modeled by any linear dynamical system can be represented by the trajectory basis. It should be noted that the inverse is not necessarily true: trajectory basis representation cannot always be realized as linear dynamical systems.

3.3.3 Observability and Reconstructibility

In the theory of linear dynamical systems, a system is *observable* if and only if there exists a finite time such that the initial state can be determined from the observation history (measurement) given the control input [57]. Mathematically, $\mathbf{\Gamma\Lambda}$ in Equation (3.16) is the observability matrix of the linear dynamical system under the Kth order Markov assumption. For trajectory reconstruction from Equation (3.4) which can be represented by linear dynamical system, the corresponding observability matrix is $\mathbf{Q\Theta}$. We generalize the observability concept to 3D trajectory reconstruction using general trajectory basis. We overload the terminology, "observable system", to describe the degeneracy of a solution of Equation (3.4).

Definition 1. A system is observable if $\mathrm{rank}(\mathbf{Q\Theta}) = 3K$ (i.e., full column rank).

Unobservable system: When the system is unobservable, there is a space of solutions where trajectory estimation is ambiguous. We characterize such an unobservable system by the following theorem.

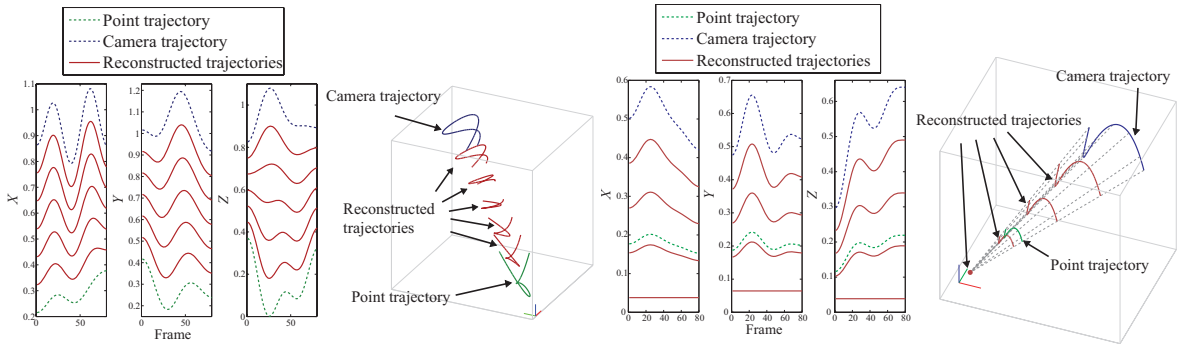
Theorem 1. Equation (3.4) is unobservable if
i) for given $\mathbf{X}$, $\mathbf{C}$, and $\mathbf{\Theta}$, $\mathbf{X}, \mathbf{C} \in \mathrm{col}(\mathbf{\Theta})$.
ii) for given $\mathbf{X}$ and $\mathbf{C}$, $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ where c is a nonzero scalar, $\mathbf{1}$ is a F dimensional vector whose entries are all ones, and $\mathbf{d} \in \mathbb{R}^3$ is an arbitrary vector.

Proof. *i*) If $\mathbf{X}, \mathbf{C} \in \mathrm{col}(\mathbf{\Theta})$, $\mathbf{X} = \mathbf{\Theta}\boldsymbol{\beta}_{\mathbf{X}}$ and $\mathbf{C} = \mathbf{\Theta}\boldsymbol{\beta}_{\mathbf{C}}$. Then,

$$\begin{aligned}
\mathrm{null}(\mathbf{Q\Theta}) &= \mathrm{null}\left(\begin{bmatrix} [\mathbf{\Phi}_1(\boldsymbol{\beta}_{\mathbf{X}} - \boldsymbol{\beta}_{\mathbf{C}})]_\times & & \\ & \ddots & \\ & & [\mathbf{\Phi}_F(\boldsymbol{\beta}_{\mathbf{X}} - \boldsymbol{\beta}_{\mathbf{C}})]_\times \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_1 \\ \vdots \\ \mathbf{\Phi}_F \end{bmatrix}\right) \\
&= \mathrm{null}\left(\begin{bmatrix} [\mathbf{\Phi}_1(\boldsymbol{\beta}_{\mathbf{X}} - \boldsymbol{\beta}_{\mathbf{C}})]_\times \mathbf{\Phi}_1 \\ \vdots \\ [\mathbf{\Phi}_F(\boldsymbol{\beta}_{\mathbf{X}} - \boldsymbol{\beta}_{\mathbf{C}})]_\times \mathbf{\Phi}_F \end{bmatrix}\right) = \boldsymbol{\beta}_{\mathbf{X}} - \boldsymbol{\beta}_{\mathbf{C}}
\end{aligned} \tag{3.18}$$

where $\mathbf{\Theta} = \begin{bmatrix} \mathbf{\Phi}_1^\mathsf{T} & \cdots & \mathbf{\Phi}_F^\mathsf{T} \end{bmatrix}^\mathsf{T}$. Since there exists a null space of $\mathbf{Q\Theta}$, $\mathrm{rank}(\mathbf{Q\Theta}) < 3K$.

ii) Let us consider two cases where $c \neq 1$ and $c = 1$.



(a) $\mathbf{C}, \mathbf{X} \in \text{col}(\mathbf{\Theta})$ (b) $\mathbf{C}, \mathbf{X} \in \text{col}(\mathbf{\Theta})$ and $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$

Figure 3.3: (a) Trajectory reconstruction is ambiguous when $\mathbf{C}, \mathbf{X} \in \text{col}(\mathbf{\Theta})$ because there exists $\text{null}(\mathbf{Q\Theta})$, which is an unobservable system. Reconstructed trajectories that satisfy Equation (3.4) are illustrated. (b) Reconstructed trajectories that satisfy Equation (3.4) when $\mathbf{C}, \mathbf{X} \in \text{col}(\mathbf{\Theta})$ and $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ are shown.

When $c \neq 1$, by plugging $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ into the Equation (3.8), it becomes,

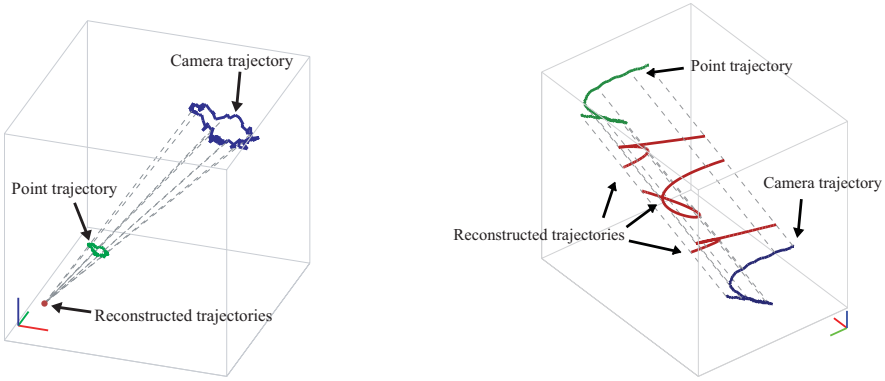
$$\begin{aligned} \left[(c-1)\mathbf{C}_i + \mathbf{d}\right]_\times \widehat{\mathbf{X}}_i &= \left[c\mathbf{C}_i + \mathbf{d}\right]_\times \mathbf{C}_i \\ &= \left[\mathbf{d}\right]_\times \mathbf{C}_i. \end{aligned} \tag{3.19}$$

From Equation (3.19), $\widehat{\mathbf{X}}_i = \alpha\mathbf{C}_i + (1-\alpha)\mathbf{d}/(1-c)$ where α is a scalar. When $\mathbf{C} \in \text{col}(\mathbf{\Theta})$, it is the case where the first condition *i*) holds, where the system is unobservable. When $\mathbf{C} \notin \text{col}(\mathbf{\Theta})$, $\alpha = 0$ because any component of $\mathbf{C}$ that cannot be expressed by the trajectory basis results in the residual error of Equation (3.3). Only $\mathbf{1} \otimes \mathbf{d}/(1-c)$ nullifies the residual error of Equation (3.4) but it is still a trivial solution (i.e., a reconstructed trajectory, $\widehat{\mathbf{X}} = \mathbf{1} \otimes \mathbf{d}/(1-c)$, is simply a stationary point even though the point undergoes motion.).

When $c = 1$, $\mathbf{d}/(1-c)$ term in $\widehat{\mathbf{X}}_i = \alpha\mathbf{C}_i + (1-\alpha)\mathbf{d}/(1-c)$ goes to infinity. It is the case where the camera moves exactly the same way the point moves with some offset and $\text{rank}(\mathbf{Q\Theta}) = 2K$ because from Equation (3.8) and $\mathbf{X} = \mathbf{C} + \mathbf{1} \otimes \mathbf{d}$,

$$\begin{aligned} \text{rank}(\mathbf{Q\Theta}) &= \text{rank}\left(\begin{bmatrix} [\mathbf{d}]_\times & & \\ & \ddots & \\ & & [\mathbf{d}]_\times \end{bmatrix}\begin{bmatrix} \mathbf{\Phi}_1 \\ \vdots \\ \mathbf{\Phi}_F \end{bmatrix}\right) \\ &= \text{rank}\left(\begin{bmatrix} \mathbf{0} & -d_3\theta_1 & d_2\theta_1 \\ \vdots & \vdots & \vdots \\ \mathbf{0} & -d_3\theta_F & d_2\theta_F \end{bmatrix}\right) + \text{rank}\left(\begin{bmatrix} d_3\theta_1 & \mathbf{0} & -d_1\theta_1 \\ \vdots & \vdots & \vdots \\ d_3\theta_F & \mathbf{0} & -d_1\theta_F \end{bmatrix}\right) = 2K, \end{aligned}$$

where $\mathbf{d} = \begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix}^\mathsf{T}$ and $\mathbf{\Phi}_i = \text{blkdiag}\{\theta_i, \theta_i, \theta_i\}$ where the trajectory basis for each coordinate (x, y, and z) is the same. Since the rank of the system is $2K$, the system is unobservable. □



(a) $c \neq 1$ (b) $c = 1$

Figure 3.4: (a) When $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ where $c \neq 1$, the solution of the system is always $\mathbf{1} \otimes \mathbf{d}/(1-c)$, which is trivial. (b) When $\mathbf{X} = \mathbf{C} + \mathbf{1} \otimes \mathbf{d}$, the system is unobservable because $\text{rank}(\mathbf{Q}\boldsymbol{\Theta}) = 2K$.

Figure 3.3 and 3.4 illustrate solutions of unobservable systems. For Theorem 1.*i*, Figure 3.3(a) shows an ambiguous solution of Equation (3.4) when $\mathbf{X}, \mathbf{C} \in \text{col}(\boldsymbol{\Theta})$. All reconstructed trajectories are a trajectory lying on one dimensional subspace $\boldsymbol{\beta}_{\mathbf{X}} - \boldsymbol{\beta}_{\mathbf{C}}$. When $\mathbf{X} = c\mathbf{C} + \mathbf{1} \otimes \mathbf{d}$ (i.e., Theorem 1.*ii*), the system is also unobservable. When $c \neq 1$, the solution is $\alpha\mathbf{C}_i + (1-\alpha)/(1-c)\mathbf{d}$. α can be nonzero only when $\mathbf{C} \in \text{col}(\boldsymbol{\Theta})$. Figure 3.3(b) shows the space of solutions by varying α. When $\mathbf{C} \notin \text{col}(\boldsymbol{\Theta})$, $\alpha = 0$ and the solution is always $\mathbf{1} \otimes \mathbf{d}/(1-c)$ (i.e., stationary point) which is a trivial solution as shown in Figure 3.4(a). Figure 3.4(b) shows trajectory reconstruction when $c = 1$, which results in $\text{rank}(\mathbf{Q}\boldsymbol{\Theta}) = 2K$. Any trajectory in K dimensional subspace (i.e., $\text{null}(\mathbf{Q}\boldsymbol{\Theta})$) is a solution lying on a surface made by the point trajectory and the camera trajectory, which is shown by gray dotted lines.

Observable system: Theorem 1 considers an unobservable system or a system resulting in a trivial solution due to the relation between the point trajectory, the camera trajectory, and the trajectory basis. For an observable system, Equation (3.4) can be solved without ambiguity in a least squares sense and there exists a unique solution, $\widehat{\boldsymbol{\beta}}$. However, the observable system does not guarantee the accuracy of the solution: How much $\widehat{\boldsymbol{\beta}}$ deviates from $\boldsymbol{\beta}_{\mathbf{X}}$. We observe accuracy of trajectory reconstruction depends on relationship between the camera trajectory, the point trajectory, and the trajectory basis. Given this observation, we characterize the case when reconstruction is accurate in the rest of this section.

Solving the least squares system, $\widehat{\mathbf{X}} = \boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$, minimizes the residual error,

$$\underset{\widehat{\boldsymbol{\beta}},\mathbf{A}}{\text{argmin}} \left\| \boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\mathbf{X} - (\mathbf{I} - \mathbf{A})\,\mathbf{C} \right\|. \tag{3.20}$$

Let us decompose the point trajectory and the camera trajectory into the column space of $\boldsymbol{\Theta}$ and that of the null space, $\boldsymbol{\Theta}^{\perp}$ as follows, $\mathbf{X} = \boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} + \boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}^{\perp}_{\mathbf{X}}$, $\mathbf{C} = \boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}} + \boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}^{\perp}_{\mathbf{C}}$, where $\boldsymbol{\beta}^{\perp}$ is the coefficient for the null space. Let us also define a measure of *reconstructibility*, η, of the 3D

point trajectory reconstruction,

$$\eta\left(\boldsymbol{\Theta}\right) \;=\; \frac{\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\|}{\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}\right\|}. \tag{3.21}$$

Reconstructibility enables us to define the accuracy of the trajectory reconstruction because as η approaches infinity, $\widehat{\boldsymbol{\beta}}$ approaches $\boldsymbol{\beta}_{\mathbf{X}}$. This can be proven as follows: from the triangle inequality, the objective function of Equation (3.20) is bounded by (when $\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}\right\| \rightarrow 0$),

$$\begin{aligned}
& \left\|\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} - \left(\mathbf{I}-\mathbf{A}\right)\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}} - \mathbf{A}\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp} - \left(\mathbf{I}-\mathbf{A}\right)\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\| && (3.22)\\
\leq\; & \left\|\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} - \left(\mathbf{I}-\mathbf{A}\right)\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}}\right\| + \left\|\mathbf{A}\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}\right\| + \left\|\left(\mathbf{I}-\mathbf{A}\right)\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\| &&\\
\leq\; & \left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\| \left(\frac{\left\|\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} - \left(\mathbf{I}-\mathbf{A}\right)\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}}\right\|}{\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\|} + \frac{\left\|\mathbf{A}\right\|}{\eta} + \left\|\mathbf{I}-\mathbf{A}\right\| \right), && (3.23)
\end{aligned}$$

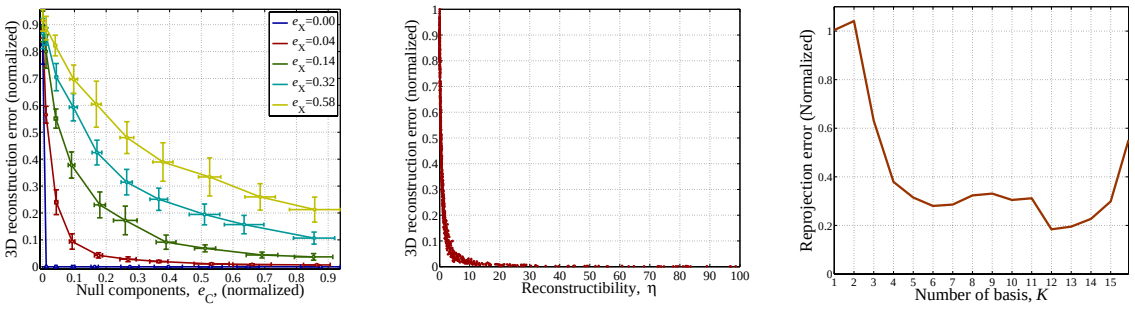
or when $\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\| \rightarrow \infty$,

$$\leq\; \left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}\right\| \left(\frac{\left\|\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}} - \mathbf{A}\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}} - \left(\mathbf{I}-\mathbf{A}\right)\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{C}}\right\|}{\left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}\right\|} + \left\|\mathbf{A}\right\| + \left\|\mathbf{I}-\mathbf{A}\right\|\eta \right). \tag{3.24}$$

As η approaches infinity, $\left\|\mathbf{A}\right\|/\eta$ in Equation (3.23) becomes zero or $\left\|\mathbf{I}-\mathbf{A}\right\|\eta$ in Equation (3.24) becomes infinity. In order to minimize either Equation (3.23) or Equation (3.24), $\mathbf{A}=\mathbf{I}$ because it leaves the last term zero and $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}_{\mathbf{X}}$ because it cancels the first term. This leads the minimum of Equation (3.23) or Equation (3.24) to be zero, which bounds the minimum of Equation (3.22). Thus, as η approaches infinity, $\widehat{\boldsymbol{\beta}}$ approaches $\boldsymbol{\beta}_{\mathbf{X}}$.

Figure 3.5(a) shows how reconstructibility is related to the accuracy of the 3D reconstruction error. In each reconstruction, the residual error (null components) of the point trajectory, $e_{\mathbf{X}} = \left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{X}}^{\perp}\right\|$, and the camera trajectory, $e_{\mathbf{C}} = \left\|\boldsymbol{\Theta}^{\perp}\boldsymbol{\beta}_{\mathbf{C}}^{\perp}\right\|$, are measured. Increasing $e_{\mathbf{C}}$ for a given point trajectory enhances the accuracy of the 3D reconstruction, while increasing $e_{\mathbf{X}}$ lowers accuracy. Even though we cannot directly measure the reconstructibility (we never know the true point trajectory in a real example), it is useful to demonstrate the direct relation with 3D reconstruction accuracy. Figure 3.5(b) illustrates that the reconstructibility is inversely proportional to the 3D reconstruction error.

Reconstructibility provides key insights into the fundamental relationship between the camera trajectory, the point trajectory, and the trajectory basis for trajectory reconstruction in 3D and it explains why a certain type of the camera motion produces high 3D reconstruction error. It is analogous to the baseline which connects two camera centers in classic triangulation as shown Figure 3.1(a). Stability or uncertainty of point reconstruction is dependent on the baseline between camera centers. If the baseline is wide, the uncertainty of the 3D reconstructed point is small and the stability of that is high. If the baseline is narrow, reconstructing the point is highly unstable (i.e., high uncertainty along the rays of projections) in the presence of Gaussian noise.



(a) Orthogonality (b) Reconstructibility (c) Number of basis

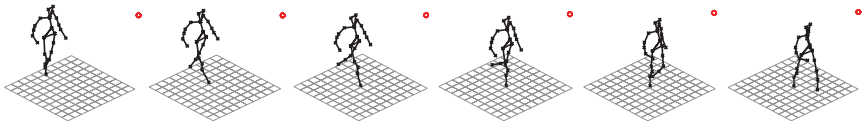
Figure 3.5: (a) As the null component of the camera trajectory, $e_{\mathbf{C}}$, decreases, the closed form solution of Equation (3.4) deviates from the real solution. (b) Reconstructibility, η, provides the degree of interference between the camera trajectory and the point trajectory. Reconstructibility is inversely proportional to 3D reconstruction error. (c) Using the cross validation scheme, the number of basis vectors is selected automatically such that it minimizes the reprojection error. Selected number of basis produces the most consistent trajectory in the presence of measurement noise. ($K = 12$)

Thus, the baseline provides a key insight of the stability of the reconstruction. Reconstructibility is the corresponding concept of the baseline for nonrigid structure from motion in trajectory space.

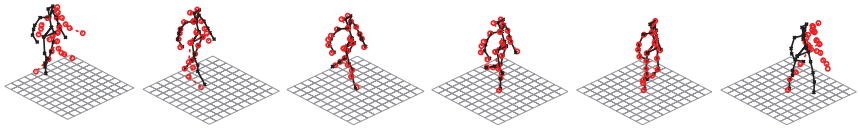
In practice, the infinite reconstructibility criterion is difficult to satisfy because the actual $\mathbf{X}$ is unknown. To enhance reconstructibility we can maximize $e_{\mathbf{C}}$ with constant $e_{\mathbf{X}}$. Thus, the best camera trajectory for a given trajectory basis matrix is the one that lives in the null space, $\mathrm{col}(\boldsymbol{\Theta}^{\perp})$. This explains our observation about slow and fast camera motion described at the beginning of this section. When the camera motion is slow, the camera trajectory is likely to be represented well by the DCT basis, which results in low reconstructibility and vice versa. However, for a given camera trajectory, there is no deterministic way to define a trajectory basis matrix because it is coupled with both the camera trajectory and the point trajectory. If one simply finds an orthogonal space to the camera trajectory, in general, it is likely to nullify space that also spans the point trajectory space. Geometrically, simply changing the surface of p in Figure 3.2 may result in a greater deviation between $\boldsymbol{\Theta}\boldsymbol{\beta}_{\mathbf{X}}$ and $\boldsymbol{\Theta}\widehat{\boldsymbol{\beta}}$.

3.4 Results

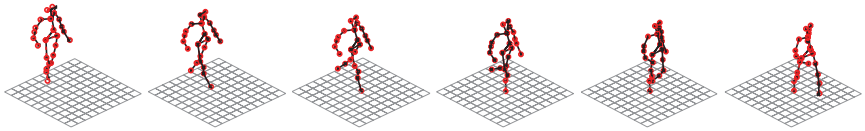
In this section, we evaluate 3D trajectory reconstruction quantitatively on motion capture data and qualitatively on real data. In all cases, the trajectory bases are the first K_i discrete cosine transform (DCT) basis in order of increasing frequency where K_i is determined by (3.5). The DCT basis has been shown that it provides the optimal performance to encode a signal under the first order Markov process [46] and demonstrated to accurately and compactly model point trajectories [3, 6]. If a 3D trajectory is continuous and smooth, the DCT basis can represent it accurately with relatively few low frequency components. We make the realistic assumption that



(a) Trajectory reconstruction at zero reconstructibility



(b) Trajectory reconstruction with low reconstructibility



(c) Trajectory reconstruction with high reconstructibility

Figure 3.6: Qualitative comparison of trajectory reconstruction from various reconstructibility. Black: ground truth, red: reconstructed trajectory. (a) Zero reconstructibility, $\eta = 0$. Camera trajectory is stationary and reconstructed trajectory is exactly the same as the camera trajectory. (b) Low reconstructibility, $\eta = 0.32$ results in mis-estimation of trajectories at the beginning and the end of the sequence. (c) All trajectories are reconstructed accurately under high reconstructibility, $\eta = 5.31$.

each point trajectory is continuous and smooth and use the DCT basis as the trajectory basis, $\mathbf{\Theta}$. Also for numerical stability of the system, we normalize 2D measurements of the each trajectory such that the mean of 2D measurements is $\mathbf{0}$ and the average distance from the origin is $\sqrt{2}$ before solving Equation (3.4) [48, 50]. The results, data, and the code of real data are available on the webpage, http://www.andrew.cmu.edu/user/hyunsoop/eccv2010/eccv_project_page.html.

3.4.1 Quantitative Evaluation

To quantitatively evaluate our method we generate synthetic 2D images from 3D motion capture data and test it in three perspectives: reconstructibility, handling missing data and low frame rate, and accuracy. For reconstructibility, we compare reconstruction by increasing the null component, $e_{\mathbf{C}}$, of the camera trajectory. For robustness, we test with missing data and lower frame rates. Finally, for accuracy, we compare our algorithm with state-of-the-art algorithms [3, 85, 113] while varying the perspectivity of projection. The results show our method outperforms others, particularly under perspective projection.

Reconstructibility: Earlier, we defined the reconstructibility of a 3D trajectory as the trade off between the ability of the chosen trajectory basis to accurately reconstruct the point trajectory vs. its ability to reconstruct the camera trajectory. To evaluate this effect empirically we generate camera trajectories by varying $e_{\mathbf{C}}$ and measure the error in point trajectory reconstruc-

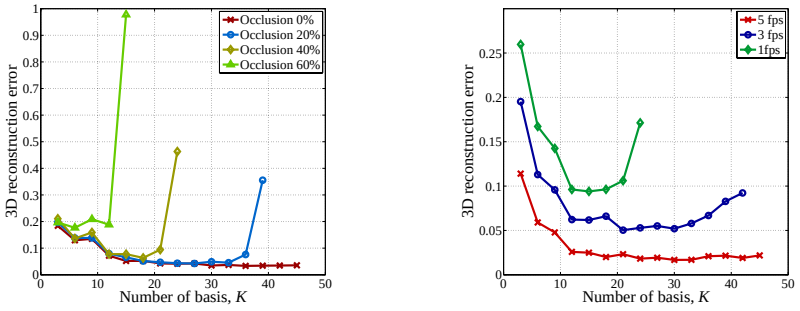
tion. Each trajectory is normalized to have zero mean and unit variance so that errors can be compared across different sequences. Figure 3.6 shows examples (walking sequences) of trajectory reconstructions under various reconstructibility. When reconstructibility is zero shown in Figure 3.6(a), reconstructed trajectories are exactly the same as the camera motions because the camera trajectory is the intersection of the hyperplane, l, and the basis space, $\mathrm{col}(\mathbf{\Theta})$, in Figure 3.2. When reconstructibility is low, $\eta = 0.32$, shown in Figure 3.6(b), the reconstruction deviates from the ground truth because there is interference from the camera trajectory. High estimation error can be observed at the beginning and the end of the sequence. If the reconstructibility is high, $\eta = 5.51$, reconstruction is very close to the ground truth.

Handling missing data and low frame rate: In this experiment, we test for the effects of missing data and low frame rate with high reconstructibility with missing 2D point samples. Missing samples occur in practice due to occlusion, self-occlusion, or measurement failure. Figure 3.7(a) shows the reconstruction error of a trajectory as the amount of occlusion varies (0%, 20%, 40%, and 60% of the sequence) for different numbers of the DCT basis, K. A walking motion capture sequence was used and each experiment was repeated 10 times with random occlusion. As long as the visibility of a point in a sequence is sufficient to overconstrain the linear system of equations, the closed form solution is robust to moderate occlusion. Figure 3.7(a) shows that our algorithm can handle relatively high number of missing data (40%) with $K = 19$. In general, as K increases, the 3D reconstruction error decreases because the high frequency components of a point trajectory can be described by the basis. However, when there is occlusion, reconstruction instability occurs by the trajectory overfitting. Figure 3.7(b) evaluates the robustness to the frequency of input samples, i.e., varying the effective frame rate of the input sequence. Visibility of moving points is important to avoid poor conditioning of the closed form solution, and intuitively more frequent visibility results in better reconstruction. The results confirm this observation. As was observed in the occlusion experiment, the higher the number of basis vectors, the less the reconstruction error but reconstruction instability can be observed when frame rate is low (1 fps).

Accuracy: We compare the accuracy of reconstructed trajectories against methods that use shape basis reconstruction proposed by Torresani *et al.* [113] and Paladini *et al.* [85] and the method that uses trajectory basis reconstruction proposed by Akhter *et al.* [3]. To validate that our closed form solution is independent of the camera projection model, we parameterize camera projection as the distance between the image plane and the camera center and evaluate across a range that moves progressively from projective at one end to orthographic at the other. Note that we are given all camera poses for the closed form trajectory solution, while the previous methods reconstruct both camera poses and point trajectories simultaneously. Figure 3.8 compares the normalized reconstruction accuracy for the walking scene under a random camera trajectory. The methods that assume orthographic camera projections are unable to accurately reconstruct trajectories in the perspective case.

3.4.2 Experiments with Real Data

The theory of reconstructibility states that it is possible to reconstruct 3D point trajectories using the DCT basis if a camera trajectory is random (non-smooth). An interesting real world example of this case occurs when many independent photographers take asynchronous images of the



(a) Occlusion (b) Frame rate

Figure 3.7: (a) While a high number of basis results in low 3D reconstruction error in general, reconstruction instability is observed when there is occlusion. Reconstruction instability results from overfitting of trajectories. Nevertheless, our algorithm can handle 40% missing data with 19 basis vectors, which results in relatively low 3D reconstruction error. (b) As frame rate increases, visibility of motion also increases, which results in low 3D reconstruction error.

Table 3.1: Parameters of real data sequences.

	F (sec)	# of photos	# of photographers
Rock climbing	39	107	5
Handshake	10	32	3
Speech	24	67	4
Greeting	24	66	4
Dance	16	49	4

same event from different locations. A collection of asynchronous photos can be interpreted as the random motion of a camera center. Using multiple photographers, we collected data in several 'media event' scenarios: a person *rock climbing*, a photo-op *hand shake*, a public *speech*, *greeting*, and *dance*. The static scene reconstruction is based on the structure from motion algorithm described in [103]. Keypoints are extracted by SIFT [68] and all possible pairs of images are considered to find matches using the fundamental matrix. To estimate camera poses, we apply structure from motion with incremental bundle adjustment to the image collection. From the first image pair, relative camera pose is estimated from the essential matrix, and then static points are triangulated. To estimate an additional camera pose we compare the keypoints registered in 3D space with new keypoints observed by the target camera. If there are unregistered keypoints which are also visible from any of the registered cameras their 3D locations are estimated through triangulation. This procedure is repeated until no image remains. Camera poses and static structures are also refined by sparse bundle adjustment [67] at each time a new camera is registered. We also extracted time and the focal length of each photo from its EXIF tag. Correspondences of moving points across images were obtained manually. Trajectory estimation is done linearly as described in Section 3.2.1. The number of basis vectors is chosen using the cross validation method individually and each linearly estimated trajectory is refined by the nonlinear optimization as described in Section 3.2.2 and in Section 3.2.3, respectively.

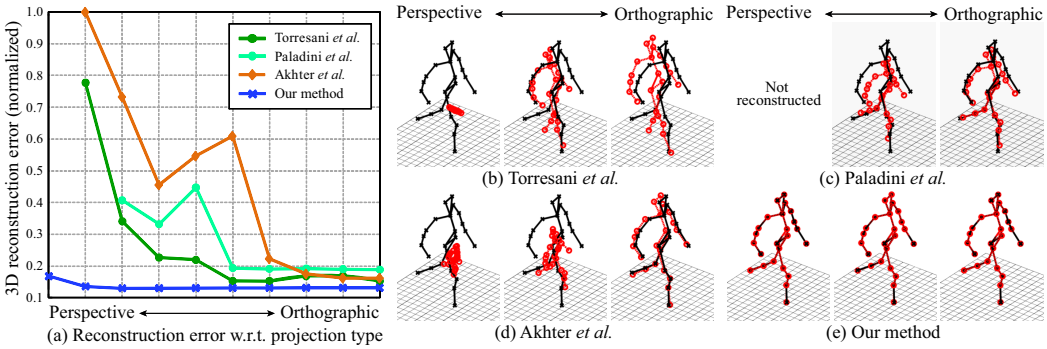


Figure 3.8: (a) Quantitative comparisons of reconstruction accuracy with previous methods regarding projection types, and qualitative comparisons of reconstruction errors using the DCT basis (blue) and the methods by Torresani *et al.* [113] (dark green), Paladini *et al.* [85] (light green) and Akhter *et al.* [3] (orange). (b-e): Qualitative comparison between the ground truth (black) and reconstructed trajectories (red) for each method.

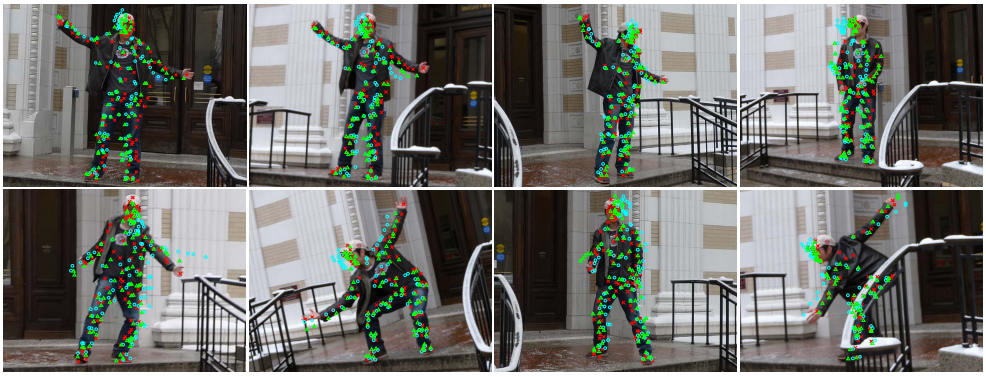


Figure 3.9: Reprojections of trajectories from manually selected K and automatically selected K_i are shown for the dance scene. Red cross: measurement, cyan circle: manually selected K, and green triangle: automatically and individually selected K_i. Trajectory from K_i has smaller reprojection error. (Average reprojections for K and K_i are 11.55 and 6.47, respectively.

To validate the proposed method of selecting the number of basis vectors described in Section 3.2.2, we tested on static points of real scenes where we know $K_i = 1$. As a result, static points of most scenes are classified as $K_i = 1$ ($> 96\%$) except for the speech scene ($> 70\%$). For the speech scene, since the baselines between photographers are very small uncertainty of the depth of points is relatively high. This causes some static points in the speech scene to be classified as moving points in depth direction.

Figure 3.9 shows results of automatic selection of the number of basis vectors for the dance scene. It is compared with manually and globally set $K = 14$. Automatic selection produces smaller reprojection error and it describes point motions better than manual selection.

The parameters for each scenario are summarized in Table 3.1. We were able to use the DCT basis for all scenes. The required number of basis implies the complexity of the trajectory. A long sequence such as the rock climbing scene requires generally higher number of basis than

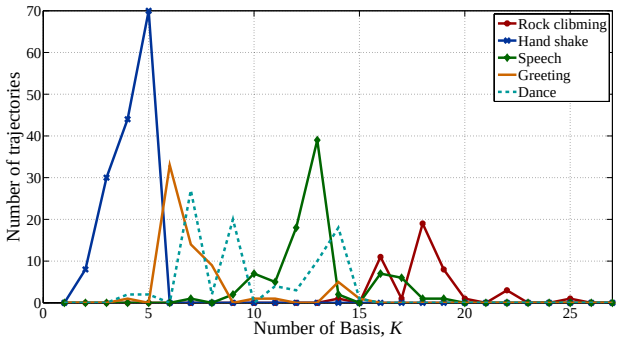


Figure 3.10: The distribution of the number of basis vectors. Scenes which are long or contain complex trajectories such as the rock climbing scene or the speech scene (complex hand motions) require high number of basis while short or simple motion scenes such as the hand shake scene or the greeting scene require low number of basis. In the greeting scene, there are several trajectories that exhibit a relatively high number of basis ($14 \sim 15$), which correspond to the hand motion (there is hand waving motion.).

a short sequence such as hand shake scene as shown in Figure 3.10. Figures 3.11, 3.12, 3.13, 3.14, and 3.15 show some of input images and reconstructed point trajectories (the number of basis vectors is color-coded into a trajectory). The reconstructed point trajectories look similar to postures of the person.

3.5 Discussion

We present an algorithm to robustly estimate the general motion of a 3D point from monocular perspective projections. The algorithm is stable in the presence of missing data and measurement noise. We rigorously analyze the cases when 3D reconstruction is possible and how accurate it can be, relating it to the concept of observability in linear dynamical systems. The algorithm presented by Park *et al*. [87] is extended to automatically select the number of trajectory basis vectors using a cross validation scheme. In addition, we refine the trajectories initialized by the least squares system by minimizing image reprojection error directly. Our algorithm takes as input the camera pose at each time instant, and a predefined trajectory basis. These requirements are met in practice when we reconstruct a dynamic scene from collections of images captured by a number of photographers. We estimate the relative camera pose by applying robust structure from motion to the static points in the scene. The Discrete Cosine Transform (DCT) is used as a pre-defined basis. Because the effective camera trajectory is quite discontinuous, we are able to obtain accurate 3D reconstructions of the dynamic scenes.

Since all points are reconstructed independently, when there is mis-matched correspondence or high depth ambiguity is observed because of small baseline, for instance, the speech scene in Figure 3.13, the trajectory can be reconstructed inaccurately. This can be resolved by applying spatial constraints on structure at a given time instant if prior information about 3D structure is available such as a human skeleton model. Future work can explore how spatial constraints may

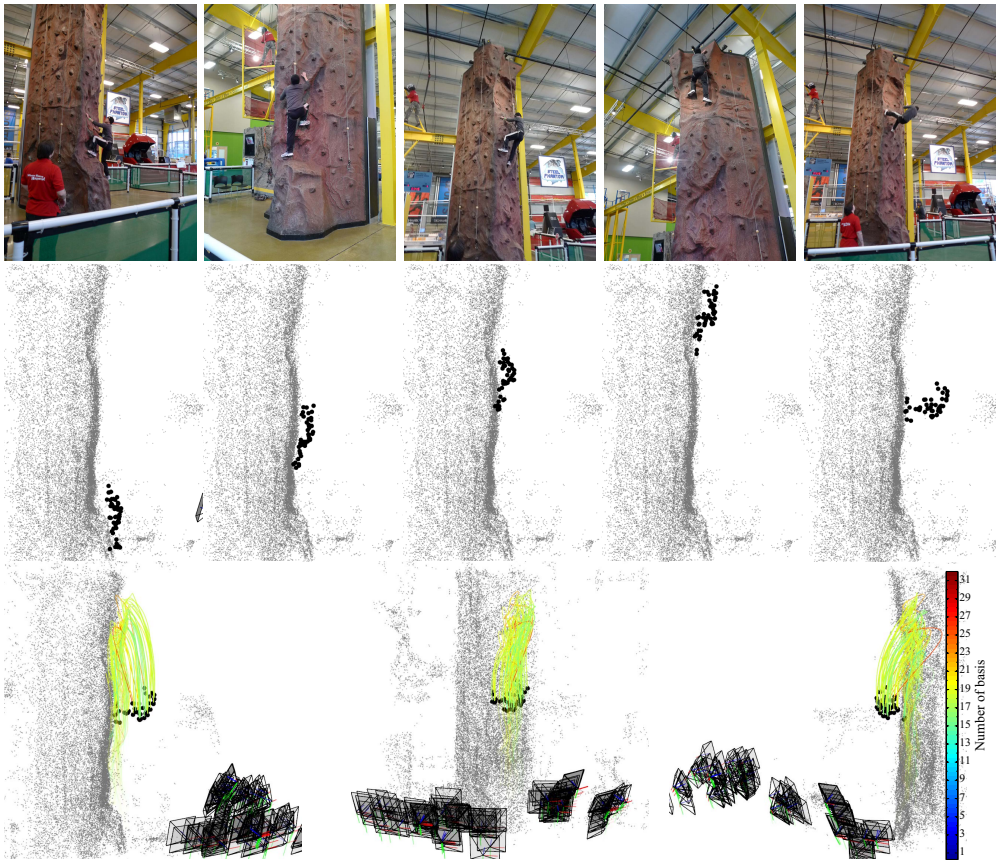


Figure 3.11: Results of the rock climbing scene. Top row: sampled image input, second row: five snap shots of 3D reconstruction of motion of the rock climber, and bottom row: reconstructed trajectories in different views. The number of basis vectors is color-coded.

correct trajectories effectively so that the system can reduce the ambiguity of motion.

Our algorithm assumes that the correspondences of moving points are given. We manually specified point correspondences across images for our experiments. From a practical stand point, this is undesirable. However, as camera optics and sensors improve, and more sophisticated point correspondence methods are developed, the ability to automatically achieve correspondences will likely become achievable. Future directions of this work include making the correspondence process entirely automatic, and applying the method to reconstruct longer sequences where the frequency of photographs, and therefore quality of reconstruction, varies within sequence. We are also interested in applying stronger priors to recognizable objects like people and faces to construct denser representations.

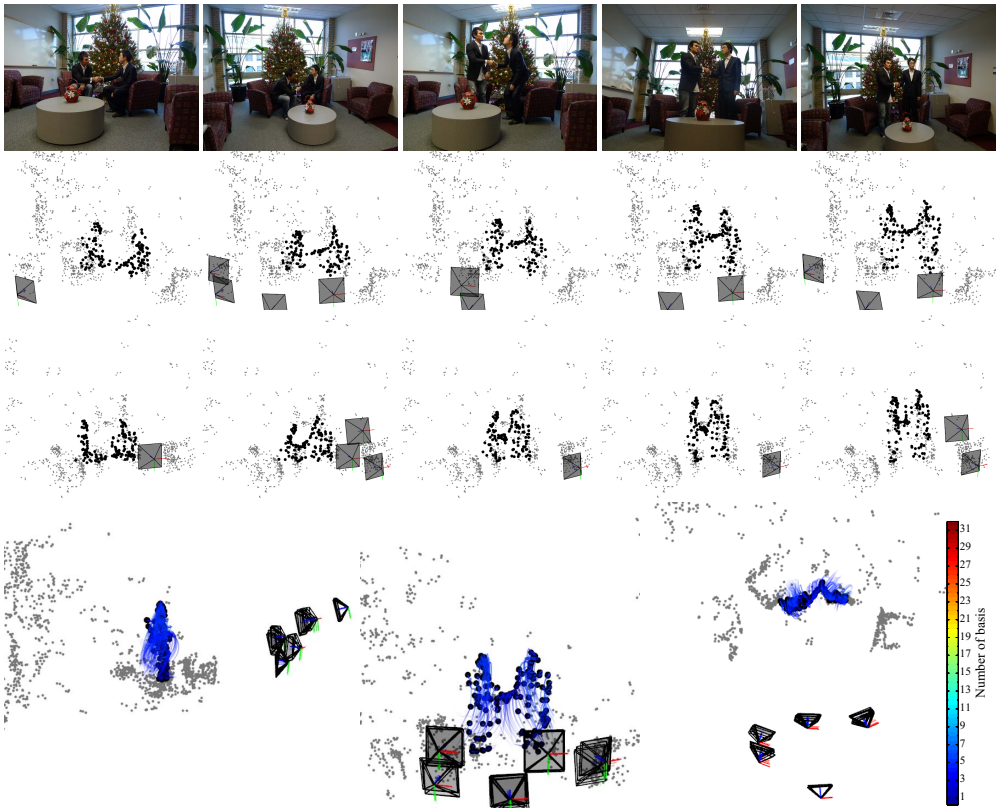


Figure 3.12: Results of the handshake scene. Top row: sampled image input, second and third row: five snap shots of 3D reconstruction in different views, and bottom row: reconstructed trajectories. The number of basis vectors is color-coded.

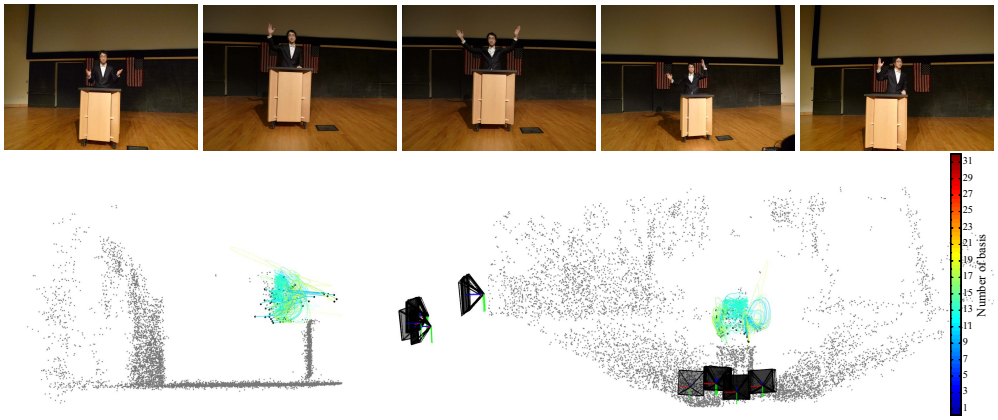


Figure 3.13: Results of the speech scene. Top row: sampled image input, and bottom row: reconstructed trajectories in different views. The number of basis vectors is color-coded.

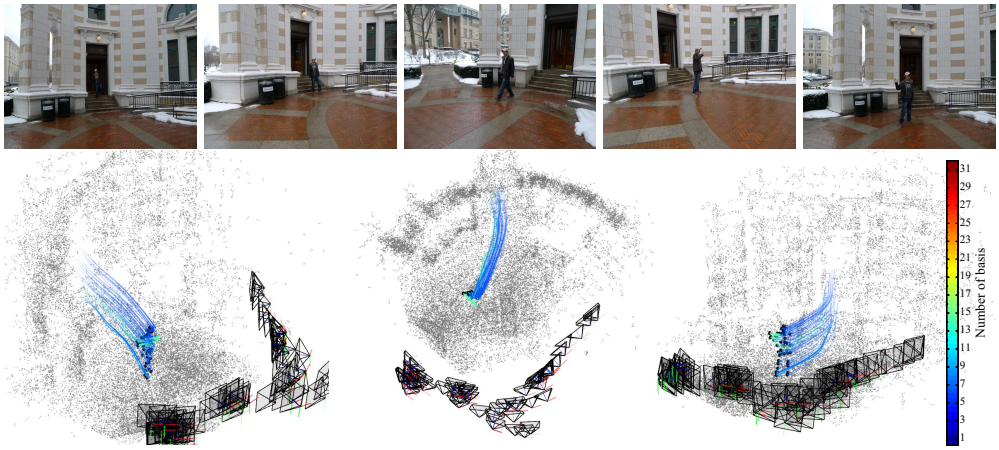


Figure 3.14: Results of the greeting scene. Top row: sampled image input and bottom row: reconstructed trajectories in different views. The number of basis vectors is color-coded.

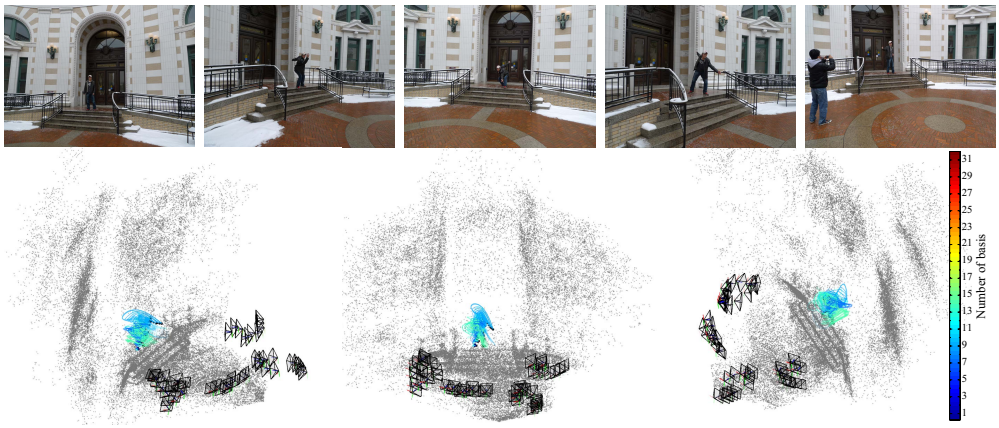


Figure 3.15: Results of the dance scene. Top row: sampled image input, and bottom row: reconstructed trajectories in different views. The number of basis vectors is color-coded.

Chapter 4

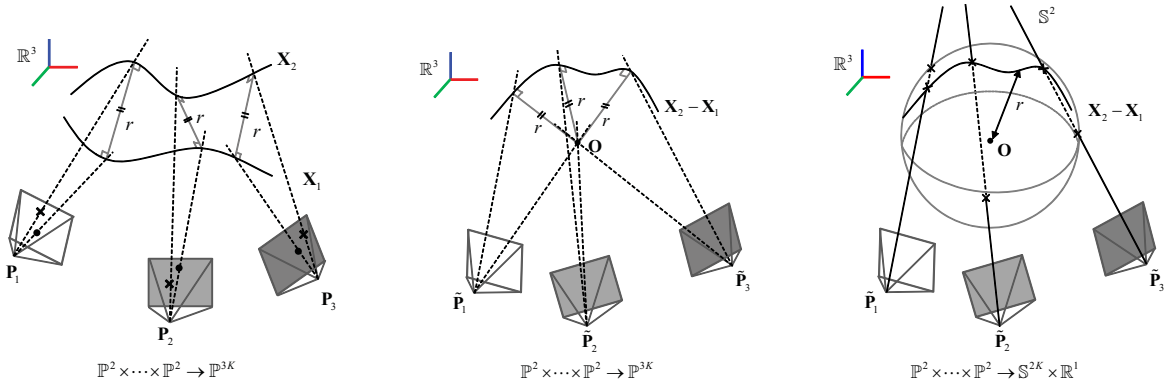
3D Reconstruction of Human Motion from a Single First Person Camera

4.1 Introduction

In Section 3, we studied 3D reconstruction of a time-varying structure from first person cameras by applying a temporal constraint. In that reconstruction, there was no spatial regularity between trajectories, i.e., each trajectory is reconstructed independently. In a social scene, many socially salient structures are associated with humans, i.e., people are interested in human motion, such as gestures. Human body is an articulated structure and the distance between two adjacent joint is fixed, e.g., distance between the elbow and shoulder joints remains constant across time instances. In this chapter, we present an algorithm for 3D reconstruction of human motion by applying temporal and spatial constraints simultaneously.

Reconstructing a moving point in three dimensions from a sequence of two dimensional projections is an ill-posed problem; any point on the line of projection connecting the camera's optical center and an image measurement can be a solution. Yet, humans can effortlessly perceive depth if the 2D points correspond to articulations of a known skeleton [56]. We study the conjecture that if 3D points move smoothly with a known articulation structure, then it is possible to reconstruct their 3D locations from their 2D projections — without any activity-specific prior. The reconstruction of an *articulated* trajectory has a fundamental ambiguity because there are two intersecting points that satisfy an articulation constraint and an image measurement at each time instant [63]: for a 2D trajectory of F frames, there are 2^F 3D trajectories that remain at fixed distance to a parent trajectory[1]. The reconstruction of a *smooth* trajectory without spatial constraints is also known to be fundamentally ambiguous when the camera trajectory is smooth [84, 87]. We present an algorithm to reconstruct a smooth articulated trajectory in 3D by *simultaneously* applying articulation and smoothness constraints. The algorithm takes as input 2D projections of the trajectory, its parent trajectory in 3D, and the camera pose at each time instant. We present a measure of reconstructibility of an articulated trajectory which characterizes the stability of estimation under articulation and smoothness constraints.

[1]The parent trajectory in a skeleton hierarchy is the proximal trajectory to the root trajectory and the child trajectory is the distal trajectory.



(a) Fixed length trajectory (b) Relative trajectory (c) Trajectory on a sphere

Figure 4.1: (a) An articulated trajectory is defined as a trajectory $\mathbf{X}_2$ which preserves distance from its parent trajectory $\mathbf{X}_1$ across all time instances. (b) The articulated trajectory is transformed to the relative trajectory, $\mathbf{X}_2 - \mathbf{X}_1$, by collapsing $\mathbf{X}_1$ to the origin. (c) The articulated trajectory lies on a sphere of radius r. There are two intersecting points at each time instant between the sphere and the ray connecting the camera's optical center and an image measurement, which allow 2^F possible 3D trajectories.

Each trajectory is parameterized by coefficients of a trajectory basis in the spherical coordinate system to enforce smoothness and articulation constraints. We show that if a trajectory is embedded in the trajectory basis and articulation constraints are applied, the reconstruction problem is equivalent to a binary quadratic program which is known to be NP-hard [39]. A number of algorithms exist that produce an approximate solution [72, 83, 92] and we use a branch-and-bound method to produce an initialization. We refine the articulated trajectories by minimizing reprojection error. The results are smooth, length preserved 3D trajectories. We have applied our algorithm to recursively reconstruct the 3D motion of a human given the 3D motion of its root. Two general approaches have been explored in prior literature to reconstruct human articulated body motion. Data-driven approaches use repositories of exemplars to overcome the ambiguity [24, 53, 115, 116, 125] and physics-based approaches use dynamical models of the human body to fit to the image stream [22, 122, 126]. Unlike these approaches, our approach reconstructs human motion from *purely geometric constraints*. Thus, the target motion is not confined to predefined activities or view points.

4.2 Geometry of an Articulated Trajectory

A point trajectory in 3D without any constraint can be represented by a series of points:

$$\mathbf{X} = \left(\begin{bmatrix} x_1 \\ \vdots \\ x_F \end{bmatrix}, \begin{bmatrix} y_1 \\ \vdots \\ y_F \end{bmatrix}, \begin{bmatrix} z_1 \\ \vdots \\ z_F \end{bmatrix} \right), \tag{4.1}$$

where (x_i, y_i, z_i) is the Cartesian coordinate of a point at i^{th} time instant and F is the number of frames. If a trajectory is smooth, it is known that the trajectory can be expressed by a linear combination of a compact trajectory basis [5], i.e.,

$$\mathbf{X} = (\mathbf{\Theta}\mathbf{a}_x, \mathbf{\Theta}\mathbf{a}_y, \mathbf{\Theta}\mathbf{a}_z) \tag{4.2}$$

where $\mathbf{\Theta}$ is a $F \times K$ matrix composed of a collection of linear trajectory basis, $\mathbf{a}$ is the coefficients or the parameters of a trajectory, and K is the number of basis.

If two trajectories, $\mathbf{X}_1$ and $\mathbf{X}_2$, are articulated, the distance between trajectories remains constant across all time instances as shown in Figure 4.1(a), i.e.,

$$\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2 = r^2, \quad i = 1, \cdots, F, \tag{4.3}$$

where $\Delta\mathbf{X} = \mathbf{X}_2 - \mathbf{X}_1$ is the relative trajectory.

When a perspective camera captures these two trajectories, points on the trajectories at the time instant are projected onto the camera plane. The camera representation in this chapter is a 3×4 projection matrix, $\mathbf{P}_i = \mathbf{K}\mathbf{R}_i \left[\begin{array}{c|c} \mathbf{I}_3 & -C_i \end{array} \right]$ where $\mathbf{I}_3$, $\mathbf{K}$, $\mathbf{R}_i$, and C_i are a 3×3 identity matrix, the upper triangular intrinsic matrix, the camera rotation matrix, and the camera's optical center vector at the i^{th} time instant, respectively.

If we transform one of the trajectories, $\mathbf{X}_1$, to the origin, $\mathbf{O}$, the other trajectory, $\mathbf{X}_2$, maps to the relative trajectory, $\Delta\mathbf{X}$, and a camera, $\mathbf{P}_i$, maps to the relative camera pose, $\widetilde{\mathbf{P}}_i$ with respect to $\mathbf{X}_1$ as shown in Figure 4.1(b). The transformed relative trajectory lies on a sphere with radius r. There are two points intersecting the sphere and the ray connecting the camera's optical center and an image measurement at each time instant as shown in Figure 4.1(c). All intersecting points are candidate 3D points which the relative trajectory passes and thus, there are 2^F possible relative trajectories.

The representation of a relative trajectory between the articulated trajectories from Equation (4.2) (Cartesian coordinate representation) has to meet the additional quadratic equality constraints of Equation (4.3). Instead of the Cartesian coordinate representation, we introduce the spherical coordinate representation for a relative trajectory to control the distance between trajectories, explicitly, i.e.,

$$\Delta\mathbf{X} = (\mathbf{\Theta}\mathbf{a}_\theta, \mathbf{\Theta}\mathbf{a}_\phi, r)\,, \tag{4.4}$$

where θ is inclination from the z axis, ϕ is azimuth from the x axis in the xy plane, and r is the radius . This representation enables us to describe an articulated trajectory precisely because it satisfies the temporal constraint and the length constraint simultaneously regardless of parameters by setting the radius constant explicitly. It also enforces that all imputed points between frames satisfy the articulation constraint while the Cartesian representation does not. For a topological point of view, the reconstruction from the spherical coordinate system is the mapping of $\mathbb{P}^{2F} \rightarrow \mathbb{S}^{2K} \times \mathbb{R}^1$ while the reconstruction from the Cartesian coordinate system is the mapping of $\mathbb{P}^{2F} \rightarrow \mathbb{P}^{3K}$ as shown in Figure 4.1(c).

4.3 Method

In this section, we present an algorithm for recovering a trajectory which satisfies spatial and temporal constraints using the spherical coordinate representation of a relative trajectory presented

in the previous section.

4.3.1 Objective Function of 3D Reconstruction

From the spherical coordinate representation, we reconstruct smooth articulated trajectories which minimize the reprojection errors:

$$\underset{\Delta\mathbf{X}_1,\cdots,\Delta\mathbf{X}_P}{\operatorname{argmin}} \sum_{i,j}^{F,P} d\left(\mathbf{x}_{ij}, \hat{\mathbf{x}}_{ij}\right), \tag{4.5}$$

where $\Delta\mathbf{X}_j$ is the j^{th} articulated (or relative) trajectory parameterized by $(\mathbf{\Theta a}_{\theta,j}, \mathbf{\Theta a}_{\phi,j}, r_j)$, $d(\cdot,\cdot)$ is the L_2 distance between two arguments, P is the number of articulated points, and $\mathbf{x}_{ij}$ and $\hat{\mathbf{x}}_{ij}$ are a 2D image measurement and a reprojection of the j^{th} point trajectory at the i^{th} time instant, respectively.

If articulated trajectories are sequentially linked, the trajectories are

$$\mathbf{X}_j = f(\mathbf{X}_R; \Delta\mathbf{X}_1, \cdots, \Delta\mathbf{X}_{j-1}), \tag{4.6}$$

where $f(\cdot)$ is the forward kinematic function that takes the root trajectory, $\mathbf{X}_R$, and all parent relative trajectories, $\Delta\mathbf{X}_1, \cdots, \Delta\mathbf{X}_{j-1}$, and outputs the j^{th} trajectory, $\mathbf{X}_j$, in the Cartesian coordinate system. The reprojection, $\hat{\mathbf{x}}_{ij}$ is

$$\hat{\mathbf{x}}_{ij} = \left(\frac{\mathbf{P}_i^1 \widetilde{\mathbf{X}}_j(i)}{\mathbf{P}_i^3 \widetilde{\mathbf{X}}_j(i)}, \frac{\mathbf{P}_i^2 \widetilde{\mathbf{X}}_j(i)}{\mathbf{P}_i^3 \widetilde{\mathbf{X}}_j(i)} \right), \tag{4.7}$$

where $\mathbf{P}_i^l$ is the l^{th} row of the camera projection matrix at the i^{th} time instant and $\widetilde{\mathbf{X}}_j(i)$ is the homogeneous representation of the i^{th} point in the j^{th} trajectory, $\mathbf{X}_j(i)$.

4.3.2 Initialization of Equation (4.5)

The objective function of Equation (4.5) is highly nonlinear and direct optimization falls into a local minimum. Therefore, a good initialization of trajectory parameters is necessary. When the parent joint position and the length between trajectories are known, there are two intersecting points between a sphere whose origin is the parent joint position, X_p, and a line connecting an image measurement and camera optical center, C, at each time instant as shown in Figure 4.2(a). A point lying on the line is $C + s\mathbf{v}$ where s is an unknown scalar and $\mathbf{v}$ is the direction of the projection, i.e.,$\mathbf{v} = \mathbf{R}^\mathsf{T}\mathbf{K}^{-1}\left[\ \mathbf{x}^\mathsf{T} 1\ \right]^\mathsf{T}$. Then, the intersecting points are

$${}^1X = C + s_1\mathbf{v}, \quad {}^2X = C + s_2\mathbf{v}, \tag{4.8}$$

where

$$s_{1,2} = \frac{-\mathbf{v}^\mathsf{T}\Delta C \pm \sqrt{\left(\mathbf{v}^\mathsf{T}\Delta C\right)^2 - \|\mathbf{v}\|^2\left(\|\Delta C\|^2 - r^2\right)}}{\|\mathbf{v}\|^2} \tag{4.9}$$

and $\Delta C = C - X_p$. For each time instant, we have two candidate 3D points through which the reconstructed trajectory must pass. Across all time instances, there are 2^F possible trajectories which satisfy the image measurements. Among those trajectories, we look for the trajectory best described by the trajectory basis.

Let χ be the relative direction vector with respect to the parent point as shown in Figure 4.2(a). For each time instant, χ_i takes either ${}^1\chi_i$ or ${}^2\chi_i$, i.e.,

$$\begin{aligned} \chi_i &= {}^1\chi_i b_i + {}^2\chi_i(1-b_i), \\ &= ({}^1\chi_i - {}^2\chi_i)b_i + {}^2\chi_i, \quad \text{where } b_i \in \{0,1\}. \end{aligned} \tag{4.10}$$

Then, all possible trajectories can be represented as:

$$\begin{aligned} \begin{bmatrix} \chi_1 \\ \vdots \\ \chi_F \end{bmatrix} &= \begin{bmatrix} \Delta\chi_1 & & \\ & \ddots & \\ & & \Delta\chi_F \end{bmatrix} \mathbf{b} + \begin{bmatrix} {}^2\chi_1 \\ \vdots \\ {}^2\chi_F \end{bmatrix} \\ \text{or} \quad \boldsymbol{\chi} &= \mathbf{E}\mathbf{b} + \mathbf{F}, \end{aligned} \tag{4.11}$$

where $\mathbf{b}$ is a binary variable vector, ${}^1\chi_i$ and ${}^2\chi_i$ are two relative direction vectors, and $\Delta\chi_i = {}^1\chi_i - {}^2\chi_i$. Finding the best trajectory is equivalent to finding the binary vector, $\mathbf{b}$, which minimizes the following cost,

$$\begin{aligned} \mathbf{b}^* = \underset{\mathbf{b}}{\operatorname{argmin}} & \left\| \left(\boldsymbol{\Theta}\boldsymbol{\Theta}^\mathsf{T} - \mathbf{I}\right)(\mathbf{E}\mathbf{b} + \mathbf{F}) \right\|^2, \\ \text{subject to} \quad & \mathbf{b} \in \{0,1\}^F. \end{aligned} \tag{4.12}$$

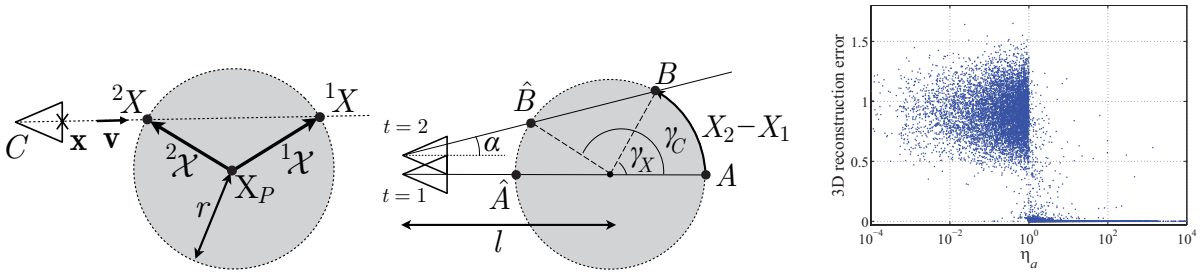
Note that $\boldsymbol{\Theta}\boldsymbol{\Theta}^\mathsf{T} - \mathbf{I}$ is the projection operation onto the null space of the trajectory basis, $\boldsymbol{\Theta}$. Equation (4.12) is a quadratic problem over binary variables.

A binary quadratic programming problem is NP-hard in general. The structure of our problem does not fall into one of the solvable cases; our quadratic matrix has positive off-diagonal elements [89], is a non-singular matrix [7, 36], and cannot be represented by a tri-/five-diagonal matrix [43]. Also, the underlying graph structure is not series parallel [13]. Thus, in theory, this is an intractable problem. However, a number of approaches have been proposed to approximate a solution of the problem efficiently using spectral or semidefinite relaxation. A branch-and-bound routine[2] with binary relaxation is one technique for global optimization. Since our quadratic matrix is positive definite, the objective function behaves convexly in a branched rectangle, which enables us to define a tight lower bound of the rectangle in polynomial time.

Once $\mathbf{b}^*$ is recovered, we project $\boldsymbol{\chi} = \mathbf{E}\mathbf{b}^* + \mathbf{F}$ onto the trajectory basis space of the spherical coordinate system to produce low dimensional parameters, i.e., $\Delta\mathbf{X} = (\boldsymbol{\Theta}\mathbf{a}_\theta, \boldsymbol{\Theta}\mathbf{a}_\phi, r)$. This yields an accurate initialization which can be refined by nonlinear optimization of Equation (4.5).

When the relative trajectory, $\Delta\mathbf{X}$, passes a singular point in the spherical coordinate system in the process of projecting $\boldsymbol{\chi}$ onto the spherical coordinate system, a discontinuity of angular trajectory occurs. For example, when ϕ passes from $\epsilon > 0$ to $2\pi - \epsilon$, this results in a discontinuity of the angular trajectory because ϕ is defined in the interval $[0, 2\pi)$. To deal with discontinuous trajectories, we find the best angular representation among all spherical representations of χ which preserves local continuity by allowing the domains of θ and ϕ to be $(-\infty, \infty)$.

[2] http://www.dii.unisi.it/~hybrid/tools/miqp/



(a) Intersection (b) Relative trajectory (c) Reconstructibility

Figure 4.2: (a) There are two solutions, ${}^1X$ and ${}^2X$ which satisfy the articulation constraint and an image measurement. (b) Articulated trajectory and the camera pose are transformed with respect to the parent trajectory. (c) The accuracy of the reconstruction is high when η_a is greater than 1 where the trajectory basis spans the ground truth trajectory better than the impostor trajectory.

4.4 Geometric Analysis of 3D Articulated Trajectory Reconstruction

We now explore the reconstruction ambiguity of an articulated trajectory and analyze configurations in which the reconstruction is accurate. Let $\mathbf{X}_1$ be a known parent trajectory and $\mathbf{X}_2$ be an articulated child trajectory which are observed at two time instances as shown in Figure 4.2(b). The ground truth relative trajectory between $\mathbf{X}_1$ and $\mathbf{X}_2$ moves from A to B. $\hat{A}$ and $\hat{B}$ are impostor points that satisfy the image measurements as well as the articulation constraint. In this section, we show that the relationship between the true trajectory and the impostor trajectory inherently determines the reconstruction accuracy.

We define a measure of *reconstructibility of an articulated trajectory*, η_a, as a criterion to characterize reconstruction accuracy where

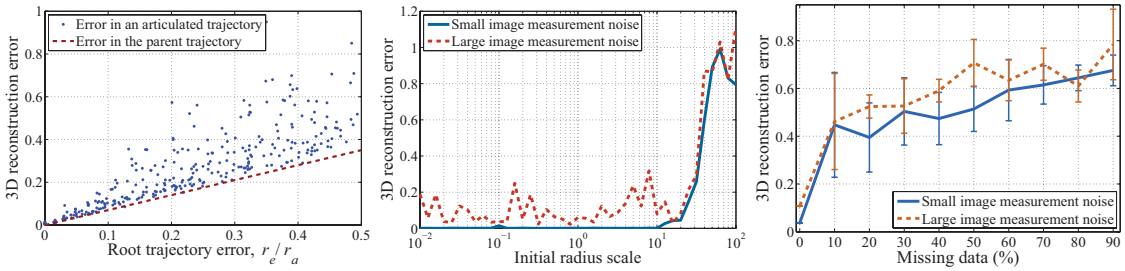
$$\eta_a = \frac{\left\| \mathbf{\Theta}^{\perp} \mathbf{a}^{\perp}_{\boldsymbol{\gamma}_\mathbf{C}} \right\|}{\left\| \mathbf{\Theta}^{\perp} \mathbf{a}^{\perp}_{\boldsymbol{\gamma}_\mathbf{X}} \right\|}, \tag{4.13}$$

$\boldsymbol{\gamma}_\mathbf{X} = \mathbf{\Theta}\mathbf{a}_{\boldsymbol{\gamma}_\mathbf{X}} + \mathbf{\Theta}^{\perp}\mathbf{a}^{\perp}_{\boldsymbol{\gamma}_\mathbf{X}}$, $\boldsymbol{\gamma}_\mathbf{C} = \mathbf{\Theta}\mathbf{a}_{\boldsymbol{\gamma}_\mathbf{C}} + \mathbf{\Theta}^{\perp}\mathbf{a}^{\perp}_{\boldsymbol{\gamma}_\mathbf{C}}$, and $\mathbf{\Theta}^{\perp}$ is the null space of the trajectory basis. If the reconstructibility of an articulated trajectory goes to infinity, there exists a unique solution and it corresponds to the ground truth trajectory. This can be proven by the following. For each time instant, there are two intersecting points and an estimation should be one of them:

$$\hat{\gamma} = (1-b)\gamma_X + b\gamma_C, \quad b = 1 \text{ or } 0 \tag{4.14}$$

where $\hat{\gamma}$ is an estimated angle. For an estimated angular trajectory,

$$\hat{\boldsymbol{\gamma}} = (\mathbf{I} - \mathbf{B})\,\boldsymbol{\gamma}_\mathbf{X} + \mathbf{B}\boldsymbol{\gamma}_\mathbf{C}, \tag{4.15}$$



(a) Error in the root trajectory (b) Initialization error of radius (c) Effect of missing data

Figure 4.3: (a) Performance of our algorithm against error in the root trajectory, (b) the initialization error of the radius, (c) amount of missing data are illustrated.

where $\mathbf{B}$ is a diagonal matrix whose entries takes either 1 or 0. The best trajectory represented by the trajectory basis minimizes following:

$$\underset{\hat{\mathbf{a}},\mathbf{B}}{\operatorname{argmin}} \left\|\boldsymbol{\Theta}\hat{\mathbf{a}} - \hat{\boldsymbol{\gamma}}\right\|^2 \tag{4.16}$$

$$= \underset{\hat{\mathbf{a}},\mathbf{B}}{\operatorname{argmin}} \left\|\boldsymbol{\Theta}\hat{\mathbf{a}} - (\mathbf{I}-\mathbf{B})\,\boldsymbol{\gamma}_{\mathbf{X}} - \mathbf{B}\boldsymbol{\gamma}_{\mathbf{C}}\right\|^2 \tag{4.17}$$

$$= \underset{\hat{\mathbf{a}},\mathbf{B}}{\operatorname{argmin}} \left\|\boldsymbol{\Theta}\hat{\mathbf{a}} - (\mathbf{I}-\mathbf{B})\,\boldsymbol{\Theta}\mathbf{a}_{\boldsymbol{\gamma}_{\mathbf{X}}} - \mathbf{B}\boldsymbol{\Theta}\mathbf{a}_{\boldsymbol{\gamma}_{\mathbf{C}}} - (\mathbf{I}-\mathbf{B})\,\boldsymbol{\Theta}^{\perp}\mathbf{a}^{\perp}_{\boldsymbol{\gamma}_{\mathbf{X}}} - \mathbf{B}\boldsymbol{\Theta}^{\perp}\mathbf{a}^{\perp}_{\boldsymbol{\gamma}_{\mathbf{C}}}\right\|^2. \tag{4.18}$$

Reconstructibility of an articulated trajectory goes to infinity when $||\boldsymbol{\Theta}^{\perp}\mathbf{a}^{\perp}_{\boldsymbol{\gamma}_{\mathbf{C}}}|| \to \infty$ or $||\boldsymbol{\Theta}^{\perp}\mathbf{a}^{\perp}_{\boldsymbol{\gamma}_{\mathbf{X}}}|| \to 0$. For either case, $\mathbf{B}$ has to approach $\mathbf{0}$ to eliminate the residual of the null components in Equation (4.18), which leads to $\hat{\mathbf{a}} \to \mathbf{a}_{\boldsymbol{\gamma}_{\mathbf{X}}}$.

From the method of Park *et al.* [87], if the camera motion is slow or stationary, there is no way to reconstruct an accurate trajectory using the trajectory basis because it spans the camera trajectory well. The reconstructibility of an articulation states that if the parent trajectory is independent of the camera trajectory, the trajectory reconstruction is still possible because mixed motion between the camera and the parent motions induces $\boldsymbol{\alpha}$ motion where $\boldsymbol{\alpha}$ is the trajectory of viewing angles from a camera, α, as shown in Figure 4.2(b). Even when camera and parent motions are stationary, the reconstruction is possible if $\boldsymbol{\gamma}_{\mathbf{X}} \in \boldsymbol{\Theta}$ because each α is a nonlinear function of γ_X, i.e., $\alpha = \tan^{-1}\left(\sin\gamma_X/(l+\cos\gamma_X)\right)$ where l is the distance between the parent trajectory and camera trajectory, and thus $\boldsymbol{\alpha} \notin \boldsymbol{\Theta}$ and $\boldsymbol{\gamma}_{\mathbf{C}} \notin \boldsymbol{\Theta}$ unless $l = 0$ or $l = \infty$ (i.e., orthographic projection) as shown in Figure 4.2(b).

Figure 4.2(c) shows the distribution of 3D reconstruction error with respect to reconstructibility of an articulated trajectory, η_a, from the CMU motion capture data[3]. A trajectory initialized by binary quadratic programming is the best fitted trajectory by the trajectory basis. When η_a is high ($\gg 1$), 3D reconstruction error of an articulated trajectory is low because the ground truth trajectory is well described by the trajectory basis and the ground truth trajectory and the impostor trajectory are well separable. In contrast, when η_a is low ($\ll 1$), our solution converges to the impostor trajectory because the trajectory basis spans the impostor trajectory better.

[3] http://mocap.cs.cmu.edu/

4.5 Results

To validate our method, we tested it with the HumanEva-II dataset, synthesized trajectories, and the CMU motion capture data quantitatively and with real human motion examples taken by video cameras qualitatively. We use the first K Discrete Cosine Transform (DCT) basis[4] in order of increasing frequency and the number of basis is chosen manually to span the trajectory well.

4.5.1 Quantitative Evaluation

We compare our method with the state-of-art human pose estimation [8, 17, 54, 88] using the HumanEva-II dataset[5]. Subject S2 with camera C1 is used to reconstruct the articulated trajectories. Our method results in 128.8mm of 3D mean error with 17.75mm standard deviation. This error is comparable to the error of the state-of-art pose estimation algorithms (82mm$\sim$211.4mm). It should be noted that while all methods rely on activity specific training data to reconstruct motions, our approach uses only activity independent geometric constraints.

We generate synthetic 2D perspective projections from synthetic data and the CMU motion capture data and evaluate for three aspects: error in the root trajectory, error in radius of an articulated trajectory, and missing data. For evaluation of errors in the root trajectory and radius, we set the camera stationary and vary error of the root trajectory and radius error while the root position is moving. For the evaluation of missing data, we artificially remove 2D projections randomly.

We measure 3D reconstruction error of an articulated trajectory by varying the ratio between the average distance error of the root trajectory, r_e, and the radius of the articulated trajectory, r_a, as shown in Figure 4.3(a). The error in the parent trajectory is a lower bound on the reconstruction error of the articulated trajectory. While the variance of the distribution for small root trajectory error (< 0.2) is low, i.e., the reconstruction can be done reliably, the reconstruction from high root trajectory error (> 0.3) causes high error in the child trajectory as well.

For the evaluation of the error in radius, we measure 3D reconstruction error for erroneous radii multiplied by scale[6]. Figure 4.3(b) illustrates robustness to erroneous initialization. Even though the initial scale is small (i.e., $10^{-2} \sim 10^{0}$), the 3D reconstruction can be done reliably because before solving the binary quadratic programming, we adjust the radius of the sphere to intersect with the line of projection at one point at least. When the initial scale is high ($> 10^{1}$), however, the reconstruction becomes unreliable because the ray intersects with the sphere at all time instances and the optimization falls into a local minimum around a mis-estimated trajectory.

We also test with the CMU motion capture data for the evaluation of missing data caused by occlusion or measurement failure. When there are missing data, our spatial and temporal constraints enable us to impute missing points. For this experiment, we artificially introduce length errors, image measurement noise, and root trajectory error while the camera is stationary. Our algorithm produces an average error[7] of 13% for 5% missing data as shown in Figure 4.3(c).

[4]Hamidi and Pearl [46] have shown that the DCT provides the optimal performance to encode the signal under the first order Markov processes. Ahkter *et al.* [5] have empirically justified its optimality on motion capture data.

[5]`http://vision.cs.brown.edu/humaneva/`

[6]Initial radius scale error 1 means the ground truth.

[7]error = $||\mathbf{X} - \hat{\mathbf{X}}||/||\mathbf{X}||$, where $\mathbf{X}$ is the ground truth trajectory and $\hat{\mathbf{X}}$ is the estimated trajectory.

4.5.2 Experiments with Real Data

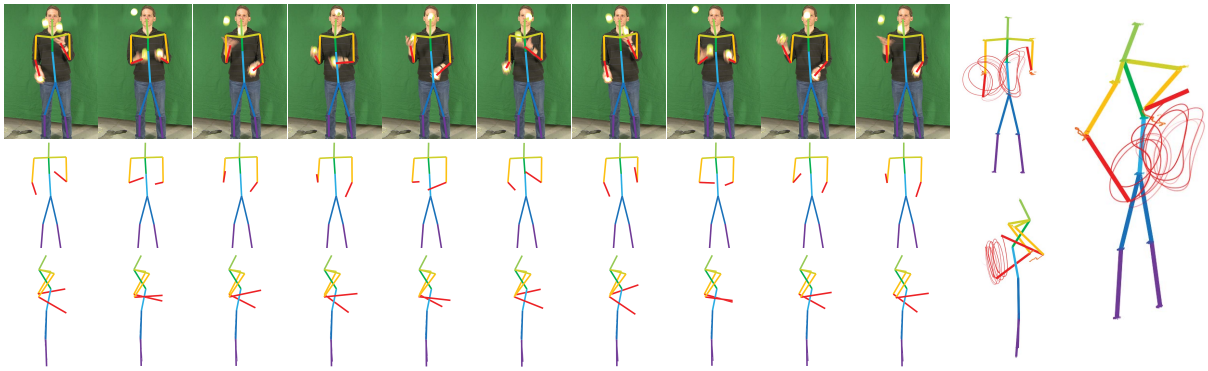
We apply our algorithm to reconstruct human body motion in 3D from 2D perspective projections. Reconstruction from a stationary camera and a moving camera are tested and the statistical anthropometric length ratio of the human body is used for the initialization of length ratio with some modifications for accurate skeleton estimation purpose. The scale of the skeleton is roughly initialized and we manually label image measurements for articulated points.

Figure 4.4(a) and Figure 4.4(b) show the reconstruction of the *juggling motion* and the *motion in front of a webcam*, respectively, in 3D from a stationary video camera. We project the 2D root trajectory to the unit depth plane and use it as the 3D root trajectory because the depth of the root trajectory is underdetermined from a stationary camera. For both experiments, we use the torso as the root. From the root trajectory, all articulated trajectories are reconstructed recursively.

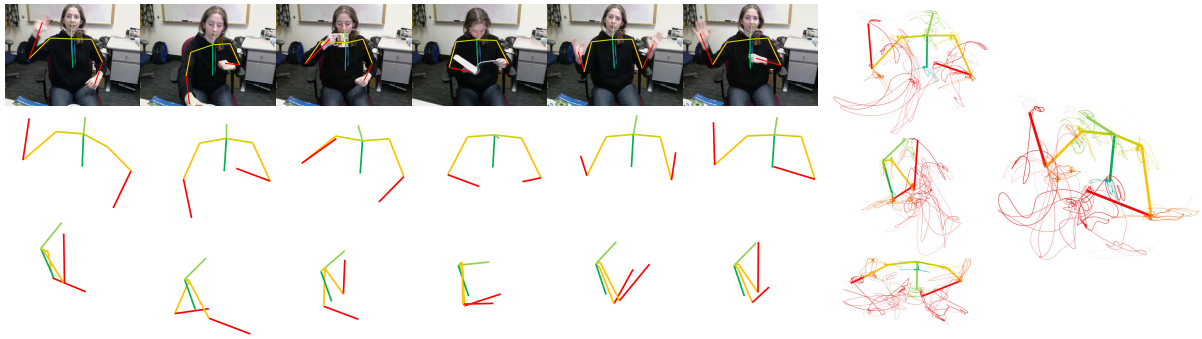
We also apply our method to data captured from a moving camera to recover the *playing card motion* and the *yoga motion* as shown in Figure 4.4(c) and Figure 4.4(d), respectively. Both camera trajectories are smooth and well spanned by the trajectory basis. For the reconstruction of the root trajectory, we choose a relatively rigid part of human body through a sequence and reconstruct them using the structure from motion algorithm. Once relative camera poses are estimated from the rigid part of the human body, we estimate the similarity transform between the relative camera poses and the original camera poses estimated by 3D static structure. Head and torso are used as the root for playing card motion and yoga motion, respectively.

4.6 Discussion

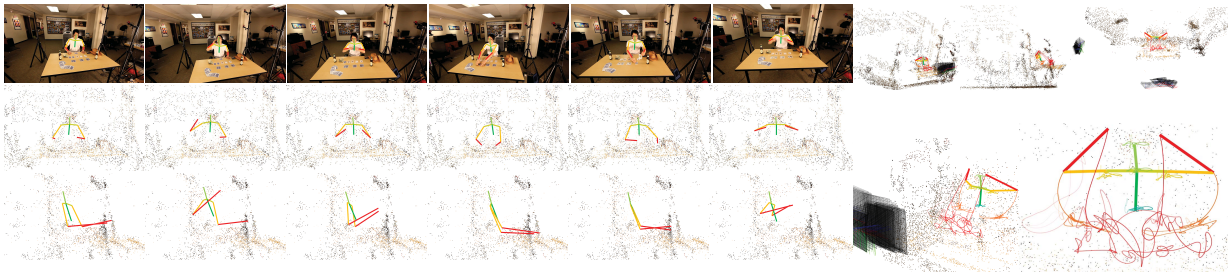
In this chapter, we study an articulated trajectory which remains at a constant distance with respect to the parent trajectory. The relative trajectory is a trajectory on a sphere and there are 2^F trajectories that meet the spatial constraint and image measurements. Among those trajectories, we look for the best trajectory spanned by the trajectory basis and we identify that this is equivalent to solving a binary quadratic programming problem. The relative trajectory obtained by the binary quadratic program is parameterized by a compact trajectory basis in the spherical coordinate system, which satisfies spatial and temporal constraints, simultaneously. We optimize the trajectory by minimizing reprojection error. Reconstruction of the articulated trajectory is fundamentally limited by the motion induced by the camera and the parent trajectory and we propose a measure of reconstructibility of an articulated trajectory, which characterizes the reconstruction accuracy. Our results show that we are able to reconstruct highly articulated human motions from a stationary camera and a moving camera.



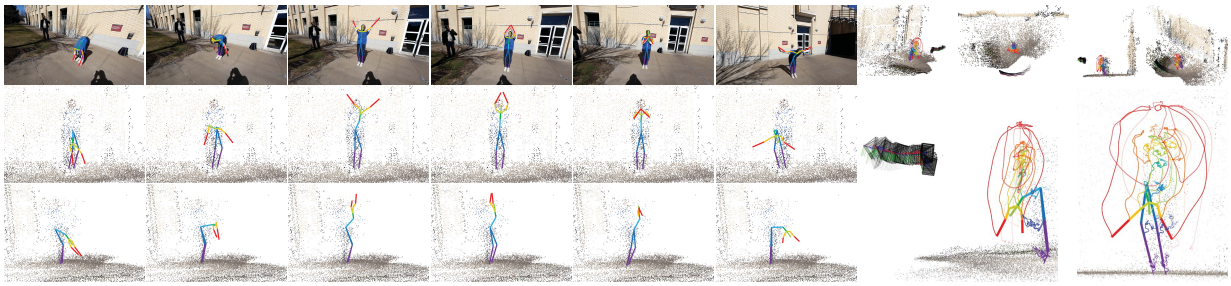
(a) Juggling motion from a stationary camera



(b) Motion in front of a webcam from a stationary camera



(c) Playing motion from a moving camera



(d) Yoga motion from a moving camera

Figure 4.4: (a) Juggling motion, (b) motion in front of the webcam from a stationary camera, (c) playing card motion, and (d) yoga motion from a moving camera. Image measurements are superimposed on images in the top row and 3D reconstruction of the motion corresponding to the images are shown from different views in the second and the third rows. The right-most figures summarize motion by showing whole trajectories.

Part II

3D Reconstruction of Social Saliency

Chapter 5

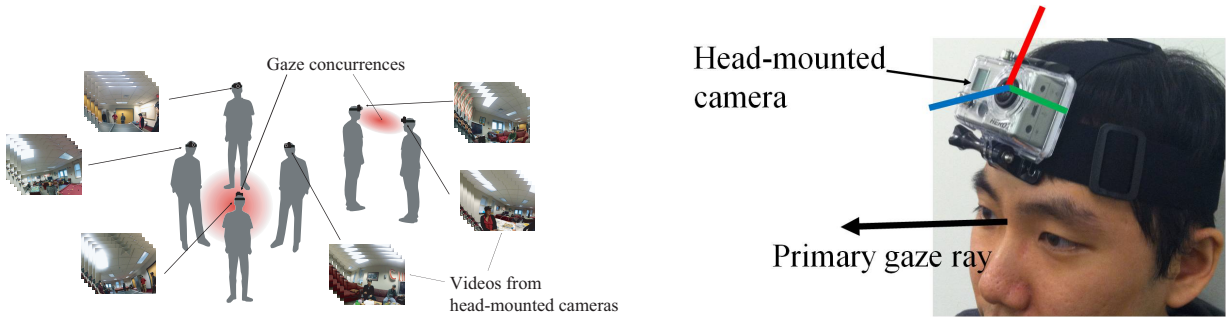
3D Reconstruction of Social Saliency from First Person Cameras

5.1 Introduction

Scene understanding approaches have largely focused on understanding the physical structure of a scene: "what is where?" [71]. In a scene occupied by people, this definition of understanding has to be expanded to include interpreting what is *socially salient* in that scene, such as whom people interact with, where they look, and what they cognitively attend to. Where classic structural scene understanding is an objective interpretation of the scene (e.g., 3D reconstruction [103], object recognition [35], or human affordance identification [45]), social scene understanding is subjective as it depends on the particular group of people and their particular relationships. For example, when we walk into a room, we quickly look at different people and the groups they have formed and choose which group we wish to join. Consider instead, an artificial agent such as a social robot that enters the same room: how should it interpret the social dynamics of the environment? The subjectivity of social environments makes the identification of quantifiable and measurable representations of social scenes difficult.

Humans transmit visible social signals about what they find important and these signals are powerful cues for social scene understanding [120]. For instance, humans spontaneously orient their gaze to the target of their attention. When multiple people simultaneously pay attention to the same point in three dimensional space, e.g., an object or a person, their *gaze rays*[1] converge to a point that we refer to as a *gaze concurrence*. Gaze concurrences are a first approximation of social saliency in a scene. It is an effective approximation because although an individual's gaze indicates what he or she is subjectively interested in, a gaze concurrence encodes the consensus of multiple individuals. Thus, social understanding tends towards objectivity when it is derived from the consensus of multiple interpretations. In this chapter, we present a method to identify and reconstruct gaze concurrences in 3D from videos taken by head-mounted cameras on multiple people (Figure 5.1(a)). Our method automatically finds multiple gaze concurrences that may occur as people form cliques in a social environment.

[1]A gaze ray is a three dimensional ray emitted from the center of eyes and oriented to the point of regard as shown in Figure 5.2(a).



(a) Input and output (b) Head-mounted camera

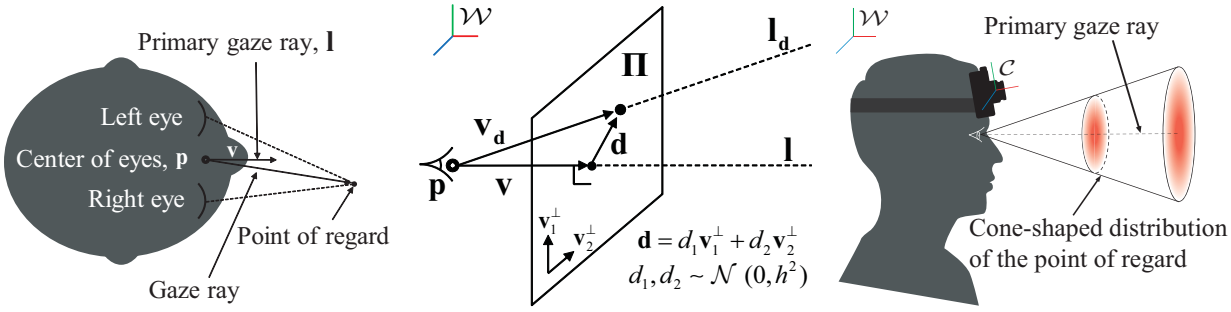
Figure 5.1: (a) In this chapter, we present a method to reconstruct 3D gaze concurrences from videos taken by head-mounted cameras. (b) A head-mounted camera provides head orientation information.

Head-mounted cameras are poised to enter our social spaces [18]. For certain specialized tasks, such as search and rescue [77], surgery [70], and empathetic interaction [79], humans would benefit from collaboration with artificial agents. To operate in these tasks, as a genuine team member, without prompting, it is necessary for the artificial agents to be able to interpret the social signals of other team members. During collaboration, the regions of social activity continually shift, split, and merge in 3D. Video data from third-person viewpoints would be biased by the camera placement and the operating space would be spatially limited. On the other hand, our method relies on head-mounted cameras, as shown in Figure 5.1(b), and therefore, it is not subject to the same limitations [100]. Furthermore, 3D pose estimation of head-mounted cameras accurately provides the primary gaze ray[2], i.e., where people are looking.

Our method takes, as input, a collection of videos captured by the head-mounted cameras and outputs the 3D gaze concurrences in a common coordinate frame with the 3D static structure, as shown in Figure 5.1(a). The head-mounted camera uses the static structure in the scene to recover the camera pose in 3D at each time instant using structure from motion. We learn the gaze parameters and the variance of the eye orientation with respect to the camera as part of the gaze ray calibration procedure. The reconstructed camera poses in conjunction with the gaze ray calibration enables us to build a 3D social saliency field as shown in Figure 5.4(c). The number and 3D locations of multiple gaze concurrences are automatically estimated via mode-seeking in the social saliency field.

The core contribution is an algorithm to estimate the 3D social saliency field of a scene and its modes from head-mounted cameras. We present a method to calibrate the primary gaze ray with respect to the head-mounted camera in 3D, and to detect multiple gaze concurrences from the gaze ray estimates via mode-seeking. To handle the variation of the eye-in-head motion, we use a cone-shaped 3D distribution. We evaluate our algorithm using motion capture data quantitatively, and apply it to real world scenes where social interactions frequently occur, such as meetings, parties, and attending theatrical performances.

[2]Eye-in-head motion contributes to the local fast gaze shift (saccade) but once the motion of the point of regard is stabilized, the eye orientation does not vary significantly from the primary gaze ray [9, 61, 74].



(a) Head top view (b) Gaze ray model (c) Gaze distribution

Figure 5.2: (a) The primary gaze ray is a fixed 3D ray with respect to the head coordinate system and the gaze ray can be described by an angle with respect to the primary gaze ray. (b) The variation of the eye orientation is parameterized by a Gaussian distribution of the points on the plane, $\mathbf{\Pi}$, which is normal to the primary gaze ray, $\mathbf{l}$ at unit distance from $\mathbf{p}$. (c) The gaze ray model results in a cone-shaped distribution of the point of regard.

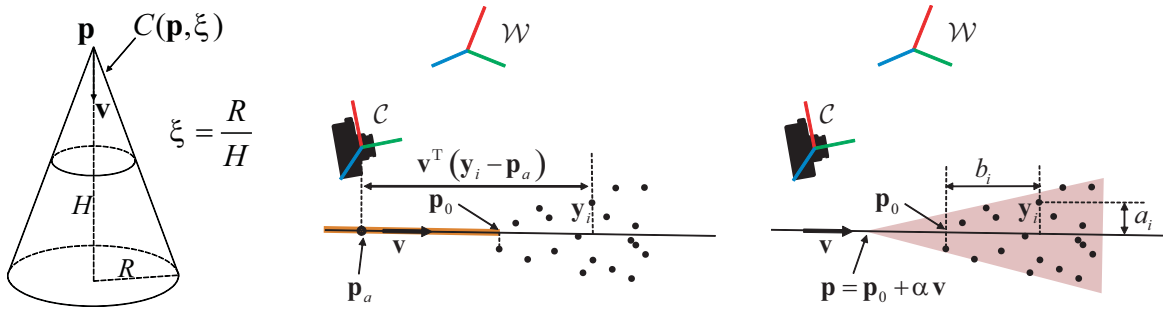
5.2 Method

In this section, we introduce an algorithm to reconstruct the gaze concurrences in 3D using head-mounted cameras. The videos from the head-mounted cameras are collected and reconstructed in 3D via structure from motion. Each person wears a camera on the head and performs a predefined motion for gaze ray calibration based on our gaze ray model (Section 5.2.1). After the calibration (Section 5.2.2), they may move freely and interact with other people. From the reconstructed camera poses in conjunction with the gaze ray model, we estimate multiple gaze concurrences in 3D via mode-seeking (Section 5.2.3).

Our camera pose registration in 3D is based on structure from motion described in [50, 100, 103]. We first scan the area of interest (for example, the room or the auditorium) with a camera to reconstruct the reference structure. The 3D poses of the head-mounted cameras are recovered relative to the reference structure using a RANSAC [37] embedded Perspective-n-Point algorithm [64]. When some camera poses cannot be reconstructed because of lack of features or motion blur, we interpolate the missing camera poses based on the epipolar constraint between consecutive frames.

5.2.1 Gaze Ray Model

We represent the direction of the viewer's gaze as a 3D ray that is emitted from the center of the eyes and is directed towards the point of regard, as shown in Figure 5.2(a). The center of the eyes is fixed with respect to the head position and therefore, the orientation of the gaze ray in the world coordinate system is a composite of the head orientation and the eye orientation (eye-in-head motion). A head-mounted camera does not contain sufficient information to estimate the gaze ray because it can capture only the head position and orientation but not the eye orientation. However, when the motion of the point of regard is stabilized, i.e., when the point of regard is stationary or slowly moving with respect to the head pose, the eye orientation varies by a small



(a) Cone (b) Apex candidate (c) Cone estimation

Figure 5.3: (a) We parameterize our cone, C, with an apex, $\mathbf{p}$, and ratio, ξ, of the radius, R, to the height, H. (b) An apex can lie on the orange colored half line, i.e., behind $\mathbf{p}_0$. Otherwise some of the points are invisible. (c) An apex can be parameterized as $\mathbf{p} = \mathbf{p}_0 - \alpha\mathbf{v}$ where $\alpha > 0$. Equation (5.2) allows us to locate the apex accurately.

degree [9, 61, 74] from the primary gaze ray[3]. We represent the variation of the gaze ray with respect to the primary gaze ray by a Gaussian distribution on a plane normal to the primary gaze ray. The point of regard (and consequently, the gaze ray) is more likely to be near the primary gaze ray.

Let us define the primary gaze ray, $\mathbf{l}$, by the center of the eyes, $\mathbf{p} \in \mathbb{R}^3$, and the unit direction vector, $\mathbf{v} \in \mathbb{R}^3$ in the world coordinate system, $\mathcal{W}$, as shown in Figure 5.2(a). Any point on the primary gaze ray can be written as $\mathbf{p} + \alpha\mathbf{v}$ where $\alpha > 0$.

Let $\mathbf{\Pi}$ be a plane normal to the primary gaze ray, $\mathbf{l}$, at unit distance from $\mathbf{p}$, as shown in Figure 5.2(b). The point, $\mathbf{d}$, in $\mathbf{\Pi}$ can be written as $\mathbf{d} = d_1\mathbf{v}_1^{\perp} + d_2\mathbf{v}_2^{\perp}$ where $\mathbf{v}_1^{\perp}$ and $\mathbf{v}_2^{\perp}$ are two orthogonal vectors to $\mathbf{v}$ and d_1 and d_2 are scalars drawn from a Gaussian distribution, i.e., $d_1,\, d_2 \sim \mathcal{N}(0, h^2)$. This point, $\mathbf{d}$, corresponds to the ray, $\mathbf{l}_\mathbf{d}$, in 3D. Thus, the distribution of the points on the plane maps to the distribution of the gaze ray by parameterizing the 3D ray as $\mathbf{l}_\mathbf{d}(\mathbf{p}, \mathbf{v}_\mathbf{d}) = \mathbf{p} + \alpha\mathbf{v}_\mathbf{d}$ where $\mathbf{v}_\mathbf{d} = \mathbf{v} + \mathbf{d}$ and $\alpha > 0$. The resulting distribution of 3D points of regard is a cone-shaped distribution whose central axis is the primary gaze ray, i.e., a point distribution on any normal plane to the primary gaze ray is a scaled Gaussian centered at the intersection between $\mathbf{l}$ and the plane as shown in Figure 5.2(c).

5.2.2 Gaze Ray Calibration

When a person wears a head-mounted camera (Figure 5.1(b)), it may not be aligned with the direction of the primary gaze ray. In general, its center may not coincide with the center of the eyes, either, as shown in Figure 5.2(c). The orientation and position offsets between the head-mounted camera and the primary gaze ray must be calibrated to estimate where the person is looking.

The relative transform between the primary gaze ray and the camera pose is constant across time because the camera is, for the most part, stationary with respect to the head, $\mathcal{C}$, as shown

[3]The primary gaze ray is a fixed eye orientation with respect to the head. It has been shown that the orientation is a unique pose, independent of gravity, head posture, horizon, and the fusion reflex [55].

in Figure 5.2(c). Once the relative transform and camera pose have been estimated, the primary gaze ray can be recovered. We learn the primary gaze ray parameters, $\mathbf{p}$ and $\mathbf{v}$, with respect to the camera pose and the standard deviation, h, of eye-in-head motion.

We ask people to form pairs and instruct each pair to look at each other's camera. While doing so, they are asked to move back and forth and side to side. Suppose two people A and B form a pair. If the cameras from A and B are temporally synchronized and reconstructed in 3D simultaneously, the camera center of B is the point of regard of A. Let $\mathbf{y}^{\mathcal{W}}$ (the camera center of B) be the point of regard of A and $\mathbf{R}$ and $\mathbf{C}$ be the camera orientation and the camera center of A, respectively. $\mathbf{y}^{\mathcal{W}}$ is represented in the world coordinate system, $\mathcal{W}$. We can transform $\mathbf{y}^{\mathcal{W}}$ to A's camera centered coordinate system, $\mathcal{C}$, by $\mathbf{y} = \mathbf{R}\mathbf{y}^{\mathcal{W}} - \mathbf{R}\mathbf{C}$. From $\{\mathbf{y}_i\}_{i=1,\cdots,n}$ where n is the number of the points of regard, we can infer the primary gaze ray parameters with respect to the camera pose. If there is no eye-in-head motion, all $\{\mathbf{y}_i\}_{i=1,\cdots,n}$ will form a line which is the primary gaze ray. Due to the eye-in-head motion, $\{\mathbf{y}_i\}_{i=1,\cdots,n}$ will be contained in a cone whose central axis is the direction of the primary gaze ray, $\mathbf{v}$, and whose apex is the center of eyes, $\mathbf{p}$.

We first estimate the primary gaze line and then, find the center of the eye on the line to completely describe the primary gaze ray. To estimate the primary gaze line robustly, we embed line estimation by two points in the RANSAC framework [37]. This enables us to obtain a 3D line, $\mathbf{l}(\mathbf{p}_a, \mathbf{v})$ where $\mathbf{p}_a$ is the projection of the camera center onto the line and $\mathbf{v}$ is the direction vector of the line. The projections of $\{\mathbf{y}_i\}_{i=1,\cdots,n}$ onto the line will be distributed on a half line with respect to $\mathbf{p}_a$. This enables us to determine the sign of $\mathbf{v}$. Given this line, we find a 3D cone, $C(\mathbf{p}, \xi)$, that encapsulates all $\{\mathbf{y}_i\}_{i=1,\cdots,n}$ where $\mathbf{p}$ is the apex and ξ is the ratio of the radius, R, to height, H, as shown in Figure 5.3(a).

The apex can lie on a half line, which originates from the closest point, $\mathbf{p}_0$, to the center of the eyes and orients to $-\mathbf{v}$ direction, otherwise some $\mathbf{y}$ are invisible. In Figure 5.3(b), the apex must lie on the orange half line. $\mathbf{p}_0$ can be obtained as follows:

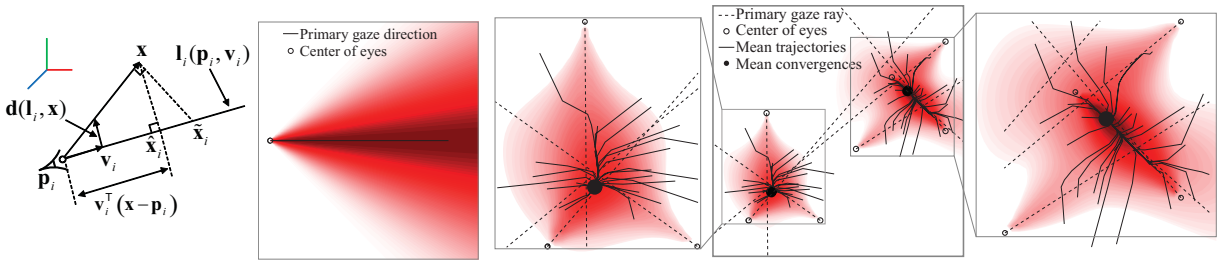
$$\mathbf{p}_0 = \mathbf{p}_a + \min\{\mathbf{v}^{\mathsf{T}}(\mathbf{y}_1 - \mathbf{p}_a), \cdots, \mathbf{v}^{\mathsf{T}}(\mathbf{y}_n - \mathbf{p}_a)\}\mathbf{v}. \tag{5.1}$$

Then, the apex can be written as $\mathbf{p} = \mathbf{p}_0 - \alpha\mathbf{v}$ where $\alpha > 0$, as shown in Figure 5.3(c).

There are an infinite number of cones which contain all points, e.g., any apex behind all points and $\xi = \infty$ can be a solution. Among these solutions, we want to find the tightest cone, where the minimum of ξ is achieved. This also leads a degenerate solution where $\xi = 0$ and $\alpha = \infty$. We add a regularization term to avoid the $\alpha = \infty$ solution. The minimization can be written as,

$$\begin{aligned} &\underset{\alpha}{\text{minimize}} \;\; \xi + \lambda\alpha \\ &\text{subject to} \quad \tfrac{a_i}{b_i+\alpha} < \xi, \quad \forall\, i = 1, \cdots, n \\ &\qquad\qquad \alpha > 0 \end{aligned} \tag{5.2}$$

where $a_i = \left\|(\mathbf{I} - \mathbf{v}\mathbf{v}^{\mathsf{T}})(\mathbf{y}_i - \mathbf{p}_0)\right\|$ and $b_i = \mathbf{v}^{\mathsf{T}}(\mathbf{y}_i - \mathbf{p}_0)$ (Figure 5.3(c)), which are all known once $\mathbf{v}$ and $\mathbf{p}_0$ are known. $a_i/(b_i+\alpha) < \xi$ is the constraint that the cone encapsulates all points of regard $\{\mathbf{y}_i\}_{i=1,\cdots,n}$ and $\alpha > 0$ is the condition that the apex must be behind $\mathbf{p}_0$. λ is a parameter



(a) Geometry (b) Gaze model (c) Social saliency field and mean trajectories

Figure 5.4: (a) $\widehat{\mathbf{x}}_i$ is the projection of $\mathbf{x}$ onto the primary gaze ray, $\mathbf{l}_i$, and $\mathbf{d}$ is a perspective distance vector defined in Equation (5.5). (b) Our gaze ray representation results in the cone-shaped distribution in 3D. (c) Two gaze concurrences are formed by seven gaze rays. High density is observed around the intersections of rays. Note that the maximum intensity projection is used to visualize the 3D density field. Our mean-shift algorithm allows any random points to converge to the highest density point accurately.

that controls how far the apex is from $\mathbf{p}_0$. Equation (5.2) is a convex optimization problem[4]. Once the cone $C(\mathbf{p}, \xi)$ is estimated from $\{\mathbf{y}_i\}_{i=1,\cdots,n}$, h is the standard deviation of the distance, $\|\mathbf{d}(\mathbf{l}, \mathbf{y}_i)\|$, and will be used in Equation (5.4) as the bandwidth for the kernel density function.

5.2.3 Gaze Concurrence Estimation via Mode-seeking

3D gaze concurrences are formed at the intersections of multiple gaze rays not at the intersection of multiple primary gazes (see Figure 5.2(a)). If we knew the 3D gaze rays, and which of rays share a gaze concurrence, the point of intersection could be directly estimated via least squares estimation, for example. In our setup, neither one of these are known, nor do we know the number of gaze concurrences. With a head-mounted camera, only the primary gaze ray is computable; the eye-in-head motion is an unknown quantity. This precludes estimating the 3D gaze concurrence by finding a point of intersection, directly. In this section, we present a method to estimate the number and the 3D locations of gaze concurrences given primary gaze rays.

Our observations from head-mounted cameras are primary gaze rays. The gaze ray model discussed in Section 5.2.1 produces the distribution of points of regard for each primary gaze ray. The superposition of Gaussian distributed gaze rays yields a 3D social saliency field. We seek modes in this saliency field via a mean-shift algorithm. The modes correspond to the gaze concurrences. The mean-shift algorithm [38] finds the modes by evaluating the weights between the current mean and observed points. We derive the closed form of the mean-shift vector directly from the observed primary gaze rays. While the observations are rays, the estimated modes are

[4]The problem can be rewritten as

$$\underset{\alpha}{\text{minimize}} \quad \max\left\{\frac{a_1}{b_1+\alpha}, \cdots, \frac{a_n}{b_n+\alpha}\right\} + \lambda\alpha, \qquad \text{subject to} \quad \alpha > 0. \tag{5.3}$$

Equation (5.3) is a convex optimization problem because the first term of the objective function is the sum of a pointwise maximum of convex functions, $a_i/(b_i+\alpha)$, which is convex in α.

points in 3D.

For any point in 3D, $\mathbf{x} \in \mathbb{R}^3$, a density function (social saliency field), f, is generated by our gaze ray model. f is the average of the Gaussian kernel density functions, K, which evaluate the distance vector between the point, $\mathbf{x}$, and the primary gaze rays, $\mathbf{l}_i$, as follows:

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{N}\sum_{i=1}^{N} K\left(\frac{\mathbf{d}(\mathbf{l}_i,\mathbf{x})}{h_i}\right) = \frac{c}{N}\sum_{i=1}^{N}\frac{1}{h_i}k\left(\frac{\|\mathbf{d}(\mathbf{l}_i,\mathbf{x})\|^2}{h_i^2}\right) \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h_i\sqrt{2\pi}}\exp\left(-\frac{1}{2}\frac{\|\mathbf{d}(\mathbf{l}_i,\mathbf{x})\|^2}{h_i^2}\right), \end{aligned} \tag{5.4}$$

where h_i is a bandwidth set to be the standard deviation of eye-in-head motion obtained from the gaze ray calibration (Section 5.2.2) for the ith gaze ray. k is the profile of the kernel density function, i.e., $K(\cdot) = ck(\|\cdot\|^2)/h$ and c is a scaling constant. $\mathbf{d} \in \mathbb{R}^3$ is a perspective distance vector defined as

$$\mathbf{d}(\mathbf{l}_i(\mathbf{p}_i,\mathbf{v}_i),\mathbf{x}) = \begin{cases} \frac{\mathbf{x}-\widehat{\mathbf{x}}_i}{\mathbf{v}_i^{\mathsf{T}}(\mathbf{x}-\mathbf{p}_i)} & \text{for } \mathbf{v}_i^{\mathsf{T}}(\mathbf{x}-\mathbf{p}_i) \geq 0 \\ \infty & \text{otherwise,} \end{cases} \tag{5.5}$$

where $\widehat{\mathbf{x}}_i = \mathbf{p}_i + \mathbf{v}_i^{\mathsf{T}}(\mathbf{x}-\mathbf{p}_i)\mathbf{v}_i$, which is the projection of $\mathbf{x}$ onto the primary gaze ray as shown in Figure 5.4(a). $\mathbf{p}_i$ is the center of eyes and $\mathbf{v}_i$ is the direction vector for the ith primary gaze ray. Note that when $\mathbf{v}_i^{\mathsf{T}}(\mathbf{x}-\mathbf{p}_i) < 0$, the point is behind the eyes, and therefore is not visible. This distance vector directly captures the distance between $\mathbf{l}$ and $\mathbf{l_d}$ in the gaze ray model (Section 5.2.1) and therefore, this kernel density function yields a cone-shaped density field (Figure 5.2(c) and Figure 5.4(b)). Figure 5.4(c) shows a social saliency field (density field) generated by seven gaze rays. The regions of the density are the gaze concurrences. Note that the maximum intensity projection of the density field is used to illustrate a 3D density field.

The updated mean is the location where the maximum density increase can be achieved from the current mean. Thus, it moves along the gradient direction of the density function evaluated at the current mean. The gradient of the density function, $f(\mathbf{x})$, is

$$\begin{aligned} \nabla_{\mathbf{x}} f(\mathbf{x}) &= \frac{2c}{N}\sum_{i=1}^{N}\frac{1}{h_i^3}k'\left(\left\|\frac{\mathbf{d}(\mathbf{l}_i,\mathbf{x})}{h_i}\right\|^2\right)\mathbf{d}(\mathbf{l}_i,\mathbf{x})^{\mathsf{T}}\left(\nabla_{\mathbf{x}}\mathbf{d}(\mathbf{l}_i,\mathbf{x})\right) \\ &= \frac{2c}{N}\left[\sum_{i=1}^{N} w_i\right]\left[\frac{\sum_{i=1}^{N} w_i\widetilde{\mathbf{x}}_i}{\sum_{i=1}^{N} w_i} - \mathbf{x}\right]^{\mathsf{T}}, \end{aligned} \tag{5.6}$$

where

$$w_i = \frac{g\left(\left\|\frac{\mathbf{d}(\mathbf{l}_i,\mathbf{x})}{h_i}\right\|^2\right)}{h_i^3\left(\mathbf{v}_i^{\mathsf{T}}(\mathbf{x}-\mathbf{p}_i)\right)^2}, \quad \widetilde{\mathbf{x}}_i = \widehat{\mathbf{x}}_i + \frac{\|\mathbf{x}-\widehat{\mathbf{x}}_i\|^2}{\mathbf{v}_i^{\mathsf{T}}(\mathbf{x}-\mathbf{p})}\mathbf{v}_i,$$

and $g(x) = -k'(x)$. $\widetilde{\mathbf{x}}_i$ is the location that the gradient at $\mathbf{x}$ points to with respect to $\mathbf{l}_i$, as shown in Figure 5.4(a). Note that the gradient direction at $\mathbf{x}$ is perpendicular to the ray connecting $\mathbf{x}$

and $\mathbf{p}_i$. The last term of Equation (5.6) is the difference between the current mean estimate and the weighted mean. The new mean location, $\mathbf{x}^{j+1}$, can be achieved by adding the difference to the current mean estimate, $\mathbf{x}^j$:

$$\mathbf{x}^{j+1} = \frac{\sum_{i=1}^{N} w_i^j \widetilde{\mathbf{x}}_i^j}{\sum_{i=1}^{N} w_i^j}. \tag{5.7}$$

Figure 5.4(c) shows how our mean-shift vector moves random initial points according to the gradient information. The mean-shift algorithm always converges as shown in the following theorem.

Theorem 2. *The sequence $\{f(\mathbf{x}^j)\}_{j=1,2,\cdots}$ provided by Equation (5.7) converges to the local maximum of the density field.*

Proof. $f(\mathbf{x})$ is a bounded function because it is the sum of finite bounded kernel density functions. To prove the theorem, it is sufficient to show that the sequence $\{f(\mathbf{x}^j)\}_{j=1,2,\cdots}$ is strictly monotonic increasing, i.e., $f(\mathbf{x}^j) < f(\mathbf{x}^{j+1})$, if $\mathbf{x}^j \neq \mathbf{x}^{j+1}$.

From Equation (5.4),

$$f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j) = \frac{c}{N} \sum_{i=1}^{N} \frac{1}{h_i} \left(k\left(\frac{\|\mathbf{d}(\mathbf{l}_i, \mathbf{x}^{j+1})\|^2}{h_i^2} \right) - k\left(\frac{\|\mathbf{d}(\mathbf{l}_i, \mathbf{x}^{j})\|^2}{h_i^2} \right) \right). \tag{5.8}$$

The profile, $k(x)$, of the Gaussian kernel density function is convex and therefore, it satisfies the following convexity condition:

$$k(x_2) - k(x_1) \;\geq\; k'(x_1)(x_2 - x_1) = g(x_1)(x_1 - x_2). \tag{5.9}$$

Note that $g(x) = -k'(x)$. The perspective distance vector function, $\mathbf{d}(\mathbf{l_i}, \mathbf{x})$, is convex in $\mathbf{x}$ by Lemma 1 and therefore, it also satisfies the convexity condition:

$$\begin{aligned} \left\|\mathbf{d}(\mathbf{l}_i, \mathbf{x}^j)\right\|^2 - \left\|\mathbf{d}(\mathbf{l}_i, \mathbf{x}^{j+1})\right\|^2 &\geq 2\mathbf{d}(\mathbf{l}_i, \mathbf{x}^j)^{\mathsf{T}} \left(\mathbf{d}(\mathbf{l}_i, \mathbf{x}^j) - \mathbf{d}(\mathbf{l}_i, \mathbf{x}^{j+1}) \right) \\ &\geq 2\mathbf{d}(\mathbf{l}_i, \mathbf{x}^j)^{\mathsf{T}} \left(\left(\nabla_{\mathbf{x}} \mathbf{d}(\mathbf{l}_i, \mathbf{x}^j) \right) \left(\mathbf{x}^{j+1} - \mathbf{x}^j \right) \right). \end{aligned} \tag{5.10}$$

Then, Equation (5.8) can be rewritten as,

$$f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j) \;\geq\; \frac{c}{N} \sum_{i=1}^{N} \frac{1}{h_i^3} g\left(\left\| \frac{\mathbf{d}(\mathbf{l}_i, \mathbf{x}^j)}{h} \right\|^2 \right) \left[\left\|\mathbf{d}(\mathbf{l}_i, \mathbf{x}^j)\right\|^2 - \left\|\mathbf{d}(\mathbf{l}_i, \mathbf{x}^{j+1})\right\|^2 \right] \tag{5.11}$$

$$\geq \frac{2c}{N} \sum_{i=1}^{N} \frac{1}{h_i^3} g\left(\left\| \frac{\mathbf{d}(\mathbf{l}_i, \mathbf{x}^j)}{h} \right\|^2 \right) \mathbf{d}(\mathbf{l}_i, \mathbf{x}^j)^{\mathsf{T}} \left(\nabla_{\mathbf{x}} \mathbf{d}(\mathbf{l}_i, \mathbf{x}^j) \right) \left(\mathbf{x}^j - \mathbf{x}^{j+1} \right) \tag{5.12}$$

$$= \frac{2c}{N} \sum_{i=1}^{N} w_i^j \left(\mathbf{x}^j - \widetilde{\mathbf{x}}_i \right)^{\mathsf{T}} \left(\mathbf{x}^j - \mathbf{x}^{j+1} \right) \tag{5.13}$$

$$= \frac{2c}{N} \sum_{i=1}^{N} w_i^j \left(\|\mathbf{x}^j\|^2 + \widetilde{\mathbf{x}}_i^{\mathsf{T}} \mathbf{x}^{j+1} - \widetilde{\mathbf{x}}_i^{\mathsf{T}} \mathbf{x}^j - (\mathbf{x}^j)^{\mathsf{T}} \mathbf{x}^{j+1} \right). \tag{5.14}$$

Inequality (5.11) and (5.12) is derived by Inequality (5.9) and (5.10), respectively. Equation (5.13) can be derived by Equation (6).

Equation (5.7) yeilds $\sum_{i=1}^{N} w_i^j \mathbf{x}^{j+1} = \sum_{i=1}^{N} w_i^j \widetilde{\mathbf{x}}_i^j$ and if we substitute it in Equation (5.14),

$$f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j) \;\geq\; \frac{2c}{N}\|\mathbf{x}^j - \mathbf{x}^{j+1}\|^2 \sum_{i=1}^{N} w_i^j. \tag{5.15}$$

Since the profile, $k(x)$, is monotonically decreasing, $k'(x) < 0$ and thus, $g(x) > 0$. This leads the weight w_i^j to be strictly positive. As a result, the right hand side of Inequality (5.15) is strictly positive if $\mathbf{x}^j \neq \mathbf{x}^{j+1}$. Thus, $f(\mathbf{x}^{j+1}) - f(\mathbf{x}^j) > 0$. □ □

Lemma 1. $\mathbf{d}(\mathbf{l}, \mathbf{x})$ *is convex in* $\mathbf{x}$.

Proof. $\mathbf{d}(\mathbf{l}, \mathbf{x})$ is convex in $\mathbf{x}$ because for $\theta \in [0, 1]$,

$$\begin{aligned}\mathbf{d}(\mathbf{l}, \theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2) &= \frac{(\mathbf{I} - \mathbf{v}\mathbf{v}^\mathsf{T})(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2 - \mathbf{p})}{\mathbf{v}^\mathsf{T}(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2 - \mathbf{p})} \\ &= \mu\mathbf{d}(\mathbf{l}, \mathbf{x}_1) + (1-\mu)\mathbf{d}(\mathbf{l}, \mathbf{x}_2)\end{aligned}$$

where

$$\mu = \frac{\theta\mathbf{v}^\mathsf{T}(\mathbf{x}_1 - \mathbf{p})}{\theta\mathbf{v}^\mathsf{T}(\mathbf{x}_1 - \mathbf{p}) + (1-\theta)\mathbf{v}^\mathsf{T}(\mathbf{x}_2 - \mathbf{p})}.$$

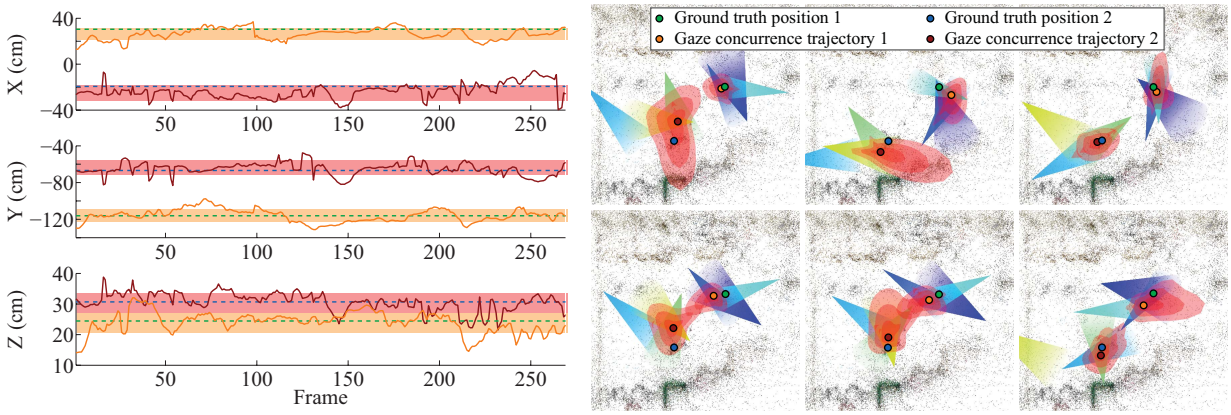
μ is still in $[0, 1]$. Thus, $\mathbf{d}(\mathbf{l}, \mathbf{x})$ is convex in $\mathbf{x}$. □ □

5.3 Result

We evaluate our algorithm quantitatively using a motion capture system to provide ground truth and apply it to real world examples where social interactions frequently occur. We use GoPro HD Hero2 cameras (www.gopro.com) and use the head mounting unit provided by GoPro. We synchronize the cameras using audio signals, e.g., a clap. In the calibration step, we let a pair of people move back and forth and side to side at least three times to allow the gaze ray model to be accurately estimated. For the initial points of the mean-shift algorithm, we sample several points on the primary gaze rays. This sampling results in convergences of the mean-shift because the local maxima form around the rays. If the weights of the estimated mode are dominated by only one gaze, we reject the mode, i.e., more than one gaze rays must contribute to estimate a gaze concurrence.

5.3.1 Quantitative Evaluation

We compare the 3D gaze concurrences estimated by our result with ground truth obtained from a motion capture system (capture volume: 8.3m×17.7m×4.3m). We attached several markers on a camera and reconstructed the camera motion using structure from motion and the motion capture system simultaneously. From the reconstructed camera trajectory, we recovered the similarity



(a) Quantitative result

(b) Social saliency field in 3D

Figure 5.5: (a) We compare our result with motion capture data (ground truth). The solid lines (orange and red) are the trajectories of the gaze concurrences and the dotted lines (green and blue) are the ground truth marker positions. Mean error is 10.1cm with 5.73cm standard deviation. The colored bands are one standard deviation wide and are centered at the trajectory means. (b) There are two gaze concurrences with six people. Orange and red points are estimated gaze concurrences and green and blue points are ground truth position. The confidence region (pink region) where a high density is achieved always contains the ground truth.

transform (scale, orientation, and translation) between two reconstructions. We placed two static markers and asked six people to move freely while looking at the markers. Therefore, the 3D gaze concurrences estimated by our algorithm should coincide with the 3D position of the static markers.

Figure 5.5(a) shows the trajectories of the gaze concurrences (solid lines) overlaid by the static marker positions (dotted lines). The mean error is 10.1cm with 5.73cm standard deviation. Figure 5.5(b) shows the gaze concurrences (orange and red points) with the ground truth positions (green and blue points) and the confidence regions (pink region) where a high value of the saliency field is achieved (region which has higher than 80% of the local maximum value). The ground truth locations are always inside these regions.

5.3.2 Experiments with Real Data

We apply our method to reconstruct 3D gaze concurrences in three real world scenes: a meeting, a musical, and a party. Figures 5.6, 5.7, and 5.8 show the reconstructed gaze concurrences and the projections of 3D gaze concurrences onto the head-mounted camera plane (top row). 3D renderings of the gaze concurrences (red dots) with the associated confidence region (salient region) are drawn in the middle row and the cone-shaped gaze ray models are also shown. The trajectories of the gaze concurrences are shown in the bottom row. The transparency of the trajectories encodes the timing.

Meeting scene: There were 11 people forming two groups: 6 for one group and 5 for the other group as shown in Figure 5.6. The people in each group started to discuss among themselves at

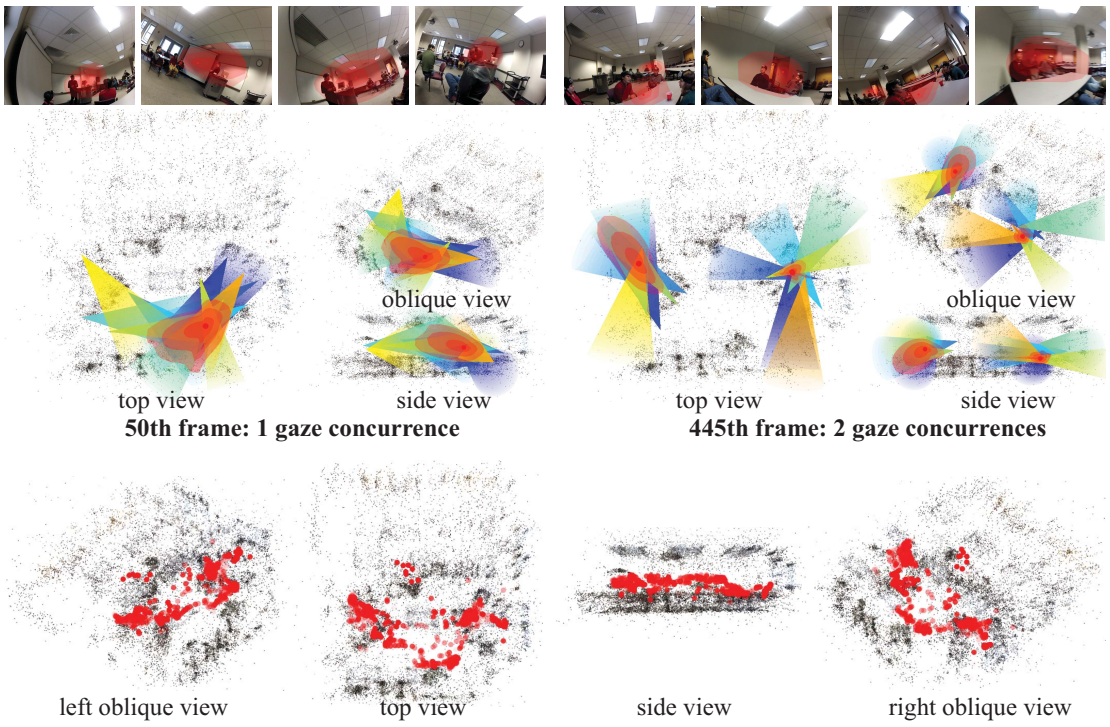


Figure 5.6: We reconstruct the gaze concurrences for the meeting scene. 11 head-mounted cameras were used to capture the scene. Top row: images with the reprojection of the gaze concurrences, middle row: rendering of the 3D gaze concurrences with cone-shaped gaze ray models, bottom row: the trajectories of the gaze concurrences.

the beginning (2 gaze concurrences). After a few minutes, all the people faced the presenter in the middle (50th frame: 1 gaze concurrence), and then they went back to their group to discuss again (445th frame: 2 gaze concurrences) as shown in Figure 5.6.

Musical scene: 7 audience members wore head-mounted cameras and watched the song, "Summer Nights" from the musical *Grease*. There were two groups of actors, "the pink ladies (women's group)" and "the T-birds (men's group)" and they sang the song alternatingly as shown in Figure 5.7. In the figure, we show the reconstruction of two frames when the pink ladies sang (41st frame) and when the T-birds sang (390th frame).

Party scene: there were 11 people forming 4 groups: 3 sat on couches, 3 talked to each other at the table, 3 played table tennis, and 2 played pool (178th frame: 4 gaze concurrences) as shown in Figure 5.8. Then, all moved to watch the table tennis game (710th frame: one gaze concurrence). Our method correctly evaluates the gaze concurrences at the location where people look. All results are best seen in the videos of the supplementary material.

5.4 Discussion

We present an algorithm that estimates 3D gaze concurrences from head-mounted cameras. The 3D gaze concurrences are locations where groups of people cognitively attend. We reconstruct

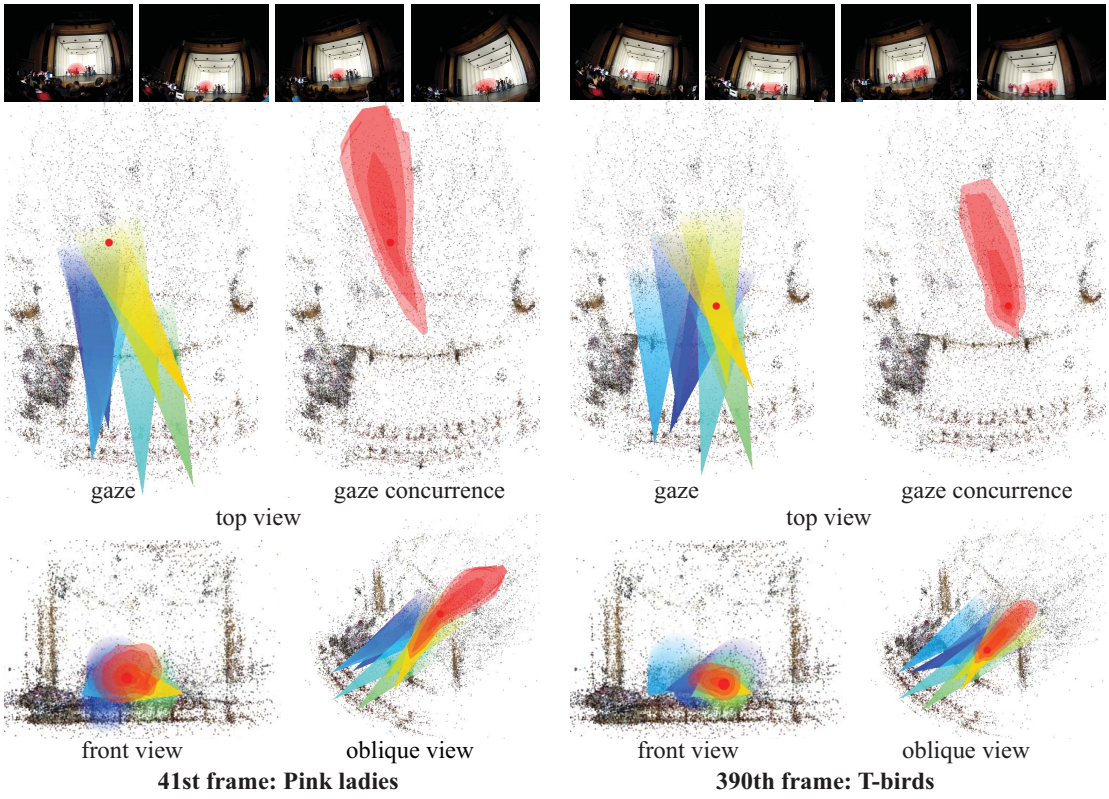


Figure 5.7: We reconstruct the gaze concurrences from musical audiences. 7 head-mounted cameras were used to capture the scene. Top row: images with the reprojection of the gaze concurrences, bottom row: rendering of the 3D gaze concurrences with cone-shaped gaze ray models.

the head-mounted camera poses in 3D using structure from motion and estimate the relationship between the camera pose and the gaze ray. The variation of the eye-in-head motion is modeled by a Gaussian distribution and it results in a 3D social saliency field. Our mode-seeking algorithm finds the gaze concurrences which are the local maxima in the social saliency field. We show that our algorithm can accurately estimate the gaze concurrences in 3D.

When people's gaze rays are almost parallel, as in the musical scene (Figure 5.7), the estimated gaze concurrences become poorly conditioned. The confidence region is stretched along the direction of the primary gaze rays. This is the case where the point of regard is very far away while people look at the point from almost the same vantage point. For such a scene, more head-mounted cameras from different points of views can help to localize the gaze concurrences more precisely.

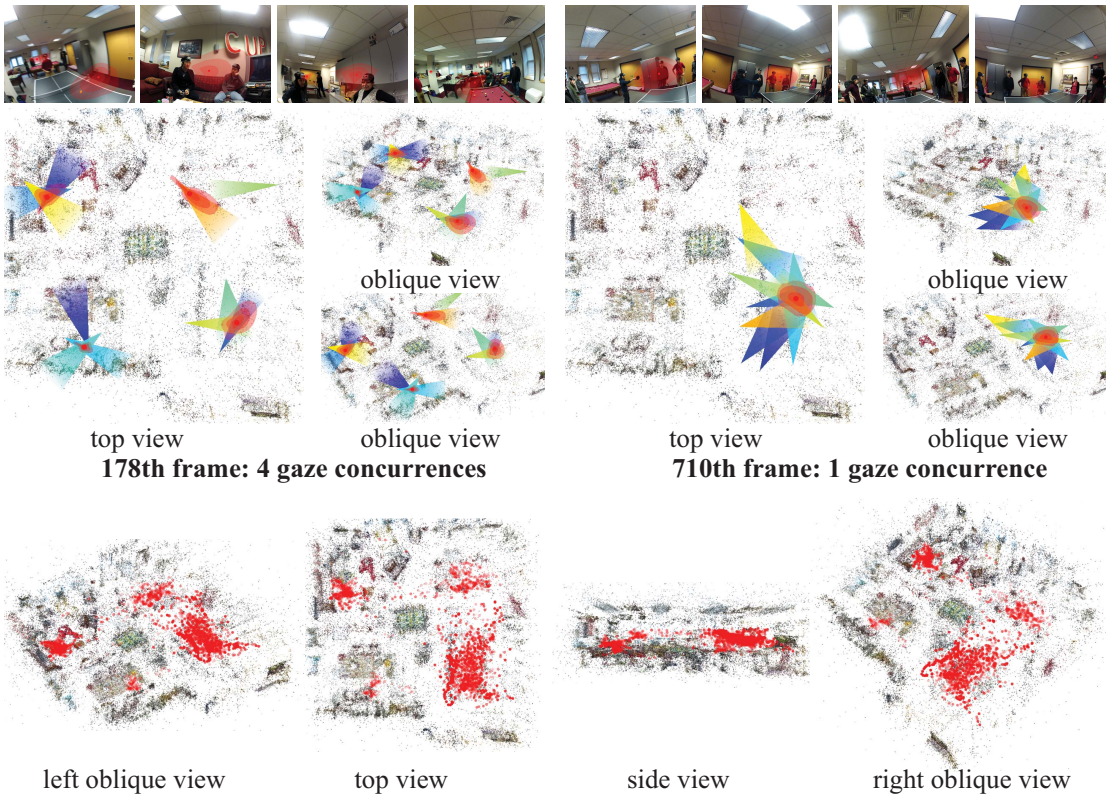


Figure 5.8: We reconstruct the gaze concurrences for the party scene. 11 head-mounted cameras were used to capture the scene. Top row: images with the reprojection of the gaze concurrences, middle row: rendering of the 3D gaze concurrences with cone-shaped gaze models, bottom row: the trajectories of the gaze concurrences.

Part III

3D Reconstruction of Socially Salient Motion

Chapter 6

3D Reconstruction of Socially Salient Motion from First Person Cameras

6.1 Introduction

A social scene typically includes many human interactions. Human interactions arise in the form of motion such as body gesture. Among motion associated with human interactions, *socially salient motion* strongly drives the holistic motion of the group of people. Imagine a lecturer in a class pointing at an equation on a blackboard with her hand. The pointing gesture triggers students' attention to move towards the equation on the board. The lecturer is a socially salient structure in the class because many students pay attention to her and her gesture is socially salient motion that directs the students' group behavior. Thus, socially salient motion is closely linked to how a group of people socially behave and it is a key component to understand a social scene.

Videos taken by first person cameras include information about a scene (exo-motion) and the wearer (ego-motion). As the wearers observe a scene, first person cameras capture how other people move in the scene, i.e., exo-motion. In Part I, we have shown how to estimate exo-motion from first person cameras; we reconstructed trajectories in 3D. Ego-motion information (how the camera, and in effect the wearer, moves in 3D) is included in first person videos [100]. The 3D relative transforms with respect to the background static structure tells us the ego-motion of the camera. 3D reconstruction of social saliency in Part II utilizes the ego-motion information of first person cameras to localize gaze concurrences. Integrating the exo- and ego-motion information of first person cameras can generate significant synergy to understand a social scene. As proposed work, we will integrate to reconstruct social motion and saliency in 3D and infer the relationship between them to identify socially salient motion.

Reconstructing socially salient motion in 3D includes two scientific challenges. First, human or point motion must be reconstructed in 3D from unstructured video data. Unlike trajectory reconstruction in Part I, first person video data does not explicitly provide point correspondences that are required to be established before reconstruction. What first person cameras see are different because what people are interested in is subjective to each person. Although we have shown that gaze concurrences can determine where people look in Part II, it is not necessary that gaze concurrences belong to physically meaningful structures, such as human body, where we can find

correspondences. Even if people look at the same structure, people are often widely distributed in the social space. This introduces the correspondence problem of wide baseline images, which has not been solved in computer vision. Unlike classic point correspondence, we will tackle the trajectory correspondence problem by exploiting motion information of a moving point. 3D reconstruction of motion must be annotated in terms of physically meaningful structures, e.g., hand gesture and head motion. This annotation of motion will allow us to build a set of motion candidates for socially salient motion. Second, the relationship between social motion and social saliency must be inferred from 3D motion reconstruction and gaze concurrences. Motion and saliency are spatio-temporal quantities that are correlated. Social saliency is usually driven by some social motion. In order to infer their relationship, they must be interpreted in computationally representable forms in the spatio-temporal domain. Based on these representations, a measure of the relationship should be defined so that we can identify motion that influences social behaviors.

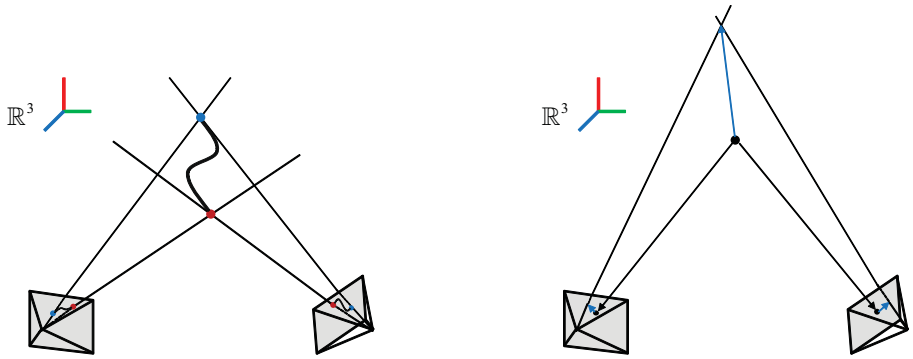
Reconstructing 3D socially salient motion is critical to understanding a social scene. A key benefit of this understanding will be to allow artificial agents to play a important role as team members without prompting in a social scene. Artificial agents will be able to process visible social signals in human interactions and recognize the signals that drives other people's behaviors. They will identify where socially salient motion occurs, focus on the motion, and anticipate how people around the socially salient structures behave or respond to them. This will enable them to organically interact with people and perform their tasks in accordance with social group behavior. By doing so, they will also learn to respond empathetically and observe social protocol in a scene. Understanding the relationship will enable investigations into social behaviors, such as group dynamics, hierarchies, or interactions: how particular motion influences group behaviors, how the rank of the society reflects social saliency, or how information propagates through social interactions. Also it will facilitate empirical study on behavioral disorder, such as autism and allow objective analysis of animal social behavior without introducing anthropomorphic bias.

6.2 Approach

Our algorithm will take, as input, a collection of videos captured by first person cameras and output prediction of 3D motion that drives social saliency. We will recognize which motion triggers group behavior via social saliency (gaze concurrences). We address challenges in reconstructing 3D socially salient motion and propose methods to resolve them.

6.2.1 3D Motion Reconstruction

In fluid dynamics and deformable solid mechanics, there are two reference frames to describe motion: the Lagrangian and Eulerian reference frames. In the Lagrangian reference frame, an observer measures flow properties by following a particle of the fluid, i.e., the reference frame is attached to the particle. For instance, the trajectory of the particle is measured by tracking the position of the reference frame across the time instances. In the Eulerian reference frame, an observer stays at a fixed position in the flow and measures flow properties. For example, the pressure change of the flow can be measured by placing many stationary barometers in the



(a) Lagrangian approach (b) Eulerian approach

Figure 6.1: (a) Classical motion reconstruction approaches from videos use the Lagrangian approach where they track 2D points that corresponds to each image and triangulate the 2D points to reconstruct the 3D point. (b) We will investigate a Eulerian approach that estimates the motion at a certain 3D point by measuring displacement of the projected 2D points.

flow. Methods to reconstruct 3D motion from videos can be also viewed in the Lagrangian and Eulerian frames. We study these approaches and their challenges.

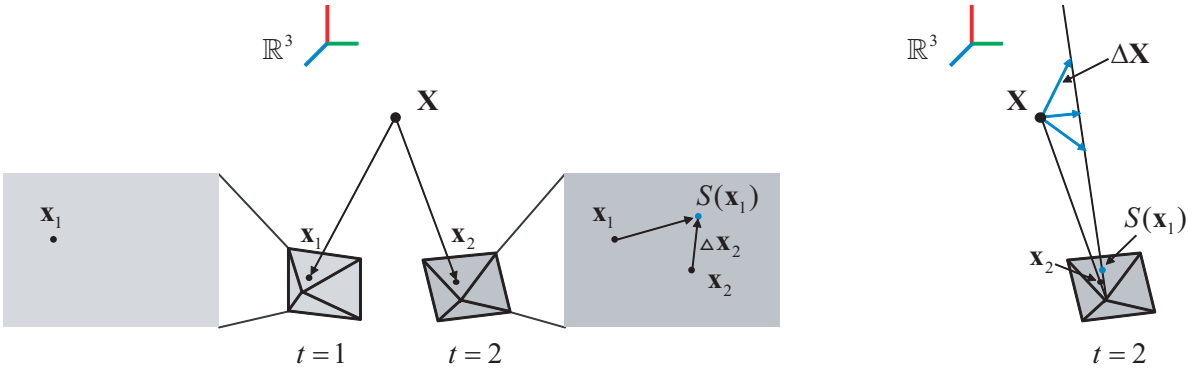
Lagrangian Approach

Classical 3D motion reconstructions from multiple videos have utilized the Lagrangian approach [66, 87, 103, 109, 114]. They track a point of interest (a particle in the Lagrangian frame) in each videos and then, triangulate the point in 3D as shown in Figure 6.1(a). This approach requires point correspondences between videos in advance. Finding point correspondences between two wide baseline images is a difficult task because local image information taken from different points of view look different. State-of-the-art matching algorithms cannot produce reliable matching results without making strong assumption about 3D points. Specifically for deformable objects, finding point correspondences is more difficult because motion introduces self-occlusion, local shape deformation, and illumination change.

Instead of finding point correspondences directly, we will find *trajectory correspondences* by utilizing point motion information encoded in a 2D trajectory. A 2D trajectory in a video is a projection of a 3D point trajectory. The corresponding 2D points from videos must undergo particular motion. For example, two corresponding points from different videos must satisfy an epipolar constraint as follows:

$$\mathbf{x}_1(t)^\mathsf{T}\mathbf{F}(t)\mathbf{x}_2(t) = 0, \tag{6.1}$$

where $\mathbf{x}(t)_1$ and $\mathbf{x}(t)_2$ are the corresponding points at time t from video 1 and 2, respectively, and $\mathbf{F}(t)$ is a fundamental matrix at time t. This constrains that two trajectories in 2D are projected from one 3D trajectory. As discussed in Part I, we can also apply a temporal constraint on a 3D trajectory by assuming the 3D point travels along a smooth trajectory. In conjunction with these constraints, motion of a local spatial descriptor, such as SIFT [68], can be a descriptor of a trajectory. As a tracked point in 2D moves, the local image descriptor changes. The



(a) Scene flow from a moving camera

(b) Ambiguity

Figure 6.2: (a) A 3D point, $\mathbf{X}$ is projected onto the camera plane at time instant 1 and 2, which forms 2D projections $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. The projected displacement, $\Delta\mathbf{x}_2 = S(\mathbf{x}_1) - \mathbf{x}_2$, is the measured displacement of $\mathbf{X}$ at $t = 2$ where $S(\mathbf{x}_1)$ is optical displacement of $\mathbf{x}_1$. (b) For a single camera, 3D displacement (scene flow), $\Delta\mathbf{X}$, is ambiguous given $\Delta\mathbf{x}_2$. To resolve the ambiguity, more than two cameras are required.

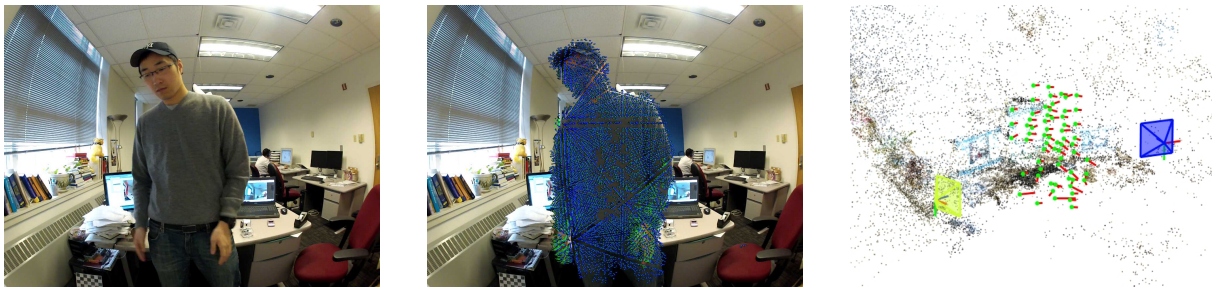
change of the descriptor forms a descriptor trajectory and this can be a discriminative classifier to detect the corresponding trajectories from other videos. The descriptor trajectory and the motion constraints will provide criteria to robustly find trajectory correspondences in the trajectory domain.

Eulerian Approach

In contrast to the Lagrangian approach, the Eulerian approach such as 3D scene flow [117, 118] does not require point correspondences between images. It estimates motion at a certain 3D point by measuring flow of the projected point onto each camera plane as shown in Figure 6.1(b). Let $\mathbf{x}_1$ and $\mathbf{x}_2 \in \mathbb{R}^2$ be projected 2D points of a 3D point, $\mathbf{X} \in \mathbb{R}^3$, onto a camera plane at time instant 1 and 2, respectively as shown in Figure 6.2(a). The projected displacement, $\Delta\mathbf{x}_2$, can be written as

$$\Delta\mathbf{x}_2 = S(\mathbf{x}_1) - \mathbf{x}_2, \tag{6.2}$$

where $S(\mathbf{x}_1) \in \mathbb{R}^2$ is the optical flow displacement of the 2D point, $\mathbf{x}_1$. The projected displacement, $\Delta\mathbf{x}_2$, accounts for how much $\mathbf{X}$ is displaced during time instant 1 and 2 in image space and thus, it is the projection of actual displacement of $\Delta\mathbf{X}$. As shown in Figure 6.2(b), there are an infinite number of possible $\Delta\mathbf{X}$ given $\Delta\mathbf{x}_2$, i.e., any point on the line that connects $S(\mathbf{x}_1)$ and the camera center can be $\Delta\mathbf{X}$. This implies that the 3D scene flow cannot be determined by a single camera because of motion ambiguity. If there are more than two cameras, the motion ambiguity can be resolved. This approach enables us to estimate a motion vector (velocity) at a given point in 3D without a point correspondence as shown in Figure 6.3.



(a) Input (b) Tracking (c) Scene flow

Figure 6.3: We reconstruct motion in 3D using the Eulerian approach, i.e., 3D scene flow. We track motion using the Kanade-Lucas-Tomasi algorithm and triangulate the optical flow in 3D.

(a) Trajectory clustering

(b) Limb position estimation

Figure 6.4: (a) 2D unstructured trajectories are clustered into three sets of trajectories. (b) Based on the clustered trajectories, we apply a rigidity constraint to estimate limb positions.

6.2.2 3D Human Motion Reconstruction

With a few exceptions, socially salient structures are associated with humans. People are interested in other people and pay attention to their motion during social interactions. The human body can be modeled by a tree structure where each limb is articulated by the parent joint and its motion is induced by the motion of the corresponding joint. We will investigate two methods to estimate human motion from videos using an articulation constraint of the human body: (1) We will detect human body or motion in videos via 2D trajectories or a part-based model [134] and then, reconstruct 3D human body pose that is consistent through all videos; (2) we will find 3D reconstructed trajectories obtained from Section 6.2.1, which satisfy the articulation constraint. We will study these two methods.

Human Motion Estimation from 2D trajectories

2D trajectories on the same rigid body satisfy the following epipolar constraint between frames,

$$\mathbf{x}_i^\mathsf{T}\mathbf{F}_{ij}\mathbf{x}_j = 0, \tag{6.3}$$

where $\mathbf{F}_{ij} \in \mathbb{R}^{3\times3}$ is a fundamental matrix and $\mathbf{x}_i$ and $\mathbf{x}_j$ are a point in the ith image and the corresponding point in the jth image, respectively. We use the affine camera model and thus, four points are required to define $\mathbf{F}_{ij}$. We estimate the fundamental matrices between consecutive frames from 2D trajectories and the array of the fundamental matrices, $\mathcal{F} = \{\mathbf{F}_{12}, \cdots, \mathbf{F}_{(F-1)F}\}$, defines the motion of each limb. We can cluster trajectories based on the array of the fundamental matrices. Figure 6.4(a) shows trajectory clustering and Figure 6.4(b) shows corresponding limb positions in 2D.

We will estimate a trajectory of joint locations from two arrays of the fundamental matrices of the adjacent limbs. Since a joint location is an unique point that two rigidity constraints from two limbs are satisfied simultaneously, the following epipolar constraints must hold between images:

$$\mathbf{x}_i^\mathsf{T}\mathbf{F}_{ij}^1\mathbf{x}_j = 0 \ , \text{ and } \ \mathbf{x}_i^\mathsf{T}\mathbf{F}_{ij}^2\mathbf{x}_j = 0, \tag{6.4}$$

where $\mathbf{F}_{ij}^1 \in \mathcal{F}_1$ and $\mathbf{F}_{ij}^2 \in \mathcal{F}_2$ are fundamental matrices for two different limbs and $\mathbf{x}_i$ and $\mathbf{x}_j$ are the corresponding joint locations in ith and jth images, respectively. Since we use the affine camera model, the fundamental matrix has a special form as,

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & 1 \end{bmatrix}. \tag{6.5}$$

By combining Equation (6.4) and Equation (6.5), we can formulate a linear system of equations for a 2D trajectory of joint locations,

$$\begin{bmatrix} \mathbf{A}_{12}^1 & & & \\ \mathbf{A}_{12}^2 & & & \\ & \ddots & & \\ & & \mathbf{A}_{(F-1)F}^1 & \\ & & \mathbf{A}_{(F-1)F}^2 & \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_F \\ y_F \end{bmatrix} = \mathcal{A}\mathbf{X} = -\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{6.6}$$

where $\mathbf{A} = \begin{bmatrix} a & b & c & d \end{bmatrix}$. Equation (6.6) is an underconstrained system because $\mathcal{A} \in \mathbb{R}^{2(F-1)\times2F}$. To solve the system, we enforce that the 2D trajectory of joint locations lies on a smooth subspace spanned by the DCT trajectory basis, $\mathbf{\Theta}$, i.e., $\mathbf{X} = \mathbf{\Theta}\boldsymbol{\beta}$ where $\boldsymbol{\beta} \in \mathbb{R}^{2K}$ is the parameters of the joint trajectory and $K < F$. Then, Equation (6.6) becomes,

$$\mathcal{A}\mathbf{\Theta}\boldsymbol{\beta} = -\mathbf{1}. \tag{6.7}$$

The trajectory parameters, $\boldsymbol{\beta}$, can be solved linearly from Equation (6.7) and the resulting trajectory of the joint locations, $\mathbf{X}$, is a smooth trajectory. Figure 6.5 shows the joint location (i.e., elbow position) estimation from two limbs.

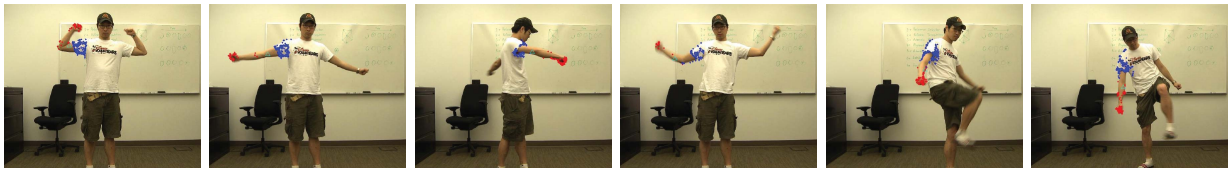


Figure 6.5: Joint location estimation from two clustered trajectories. Blue and red points are from different rigid bodies and triangles are joint locations obtained by Equation (6.7)

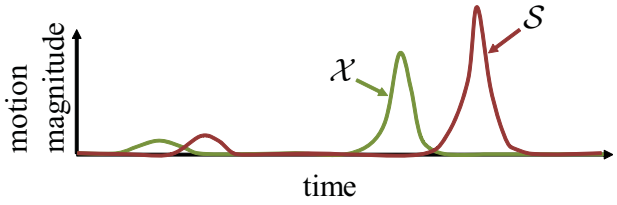


Figure 6.6: A conceptual graph of social motion and social saliency. If the motion is socially salient motion, the trajectory of social saliency will follow the motion.

Human Motion Estimation from 3D trajectories

We can also reconstruct human motion from 3D unstructured trajectories that are reconstructed in Section 6.2.1. Some 3D trajectories reconstructed by 2D trajectory correspondences belong to human body trajectories and thus, these trajectories can be represented by human motion. We will find 3D trajectories that satisfy the articulation constraint and reconstruct the corresponding human motion in 3D.

3D points that move together likely belong to the same human body. Thus, trajectories belong to the same person have similar patterns. We will cluster 3D trajectories that exhibit similar patterns via a spectral clustering method. This method does not require the number of cluster, explicitly. This will enable us to isolate trajectories for one person. From these clustered trajectories, we will identify the extremities from trajectories and fit a human model composed of the articulated tree structure. We will study how 3D trajectories are constrained and develop an algorithm to robustly fit 3D human body model onto the unlabeled 3D trajectories.

6.2.3 Inference of Relationship Between Motion and Saliency

Social saliency is driven by social motion, such as visible social signals. From reconstructed human motion in conjunction with gaze concurrences, we will present a method to infer the relationship between social motion and social saliency and identify socially salient motion. Inferring the relationship requires computational representations of social motion and social saliency. We will propose spatio-temporal representations and find the correlation between them.

Let $\mathcal{X}$ be a time series of a 3D point on human body and $\mathcal{S}$ be a time series of a gaze concurrence, i.e., $\mathcal{X} = \{\mathbf{X}_i\}_{i=1,\cdots,F}$ and $\mathcal{S} = \{\mathbf{S}_i\}_{i=1,\cdots,F}$ where $\mathbf{X}_i \in \mathbb{R}^3$ and $\mathbf{S}_i \in \mathbb{R}^3$ are a point on human body and a gaze concurrence at ith time instant, respectively and F is the number of frames. For a socially salient structure, there is $\mathcal{X}$ for each limb, e.g., $\mathcal{X}_{head}$. Since social saliency follows socially salient motion, we expect that $\mathcal{S}$ will be highly correlated with $\mathcal{X}$

with some time lag as shown in Figure 6.6. The peaks of $\mathcal{S}$ will be observed after the peaks of $\mathcal{X}$ are observed. We will use cross-correlation as a measure of the relationship. Cross-correlation is widely used to define a correlation between time-lag signals in econometrics [23, 42, 121]. Our $\mathcal{X}$ and $\mathcal{S}$ will be time-lag correlated signals if $\mathcal{X}$ is socially salient motion. Our cross-correlation ρ will be defined as,

$$\rho(\tau) = \frac{\mathbf{E}\left[\left(\mathcal{X}_t - \mu_{\mathcal{X}}\right)\left(\mathcal{S}_{t+\tau} - \mu_{\mathcal{S}}\right)\right]}{\sigma_{\mathcal{X}}\sigma_{\mathcal{S}}} \tag{6.8}$$

where $\mathcal{X}_t$ and $\mathcal{S}_t$ are a random variable for the time series $\mathcal{X}$ and $\mathcal{S}$, respectively and $\mu_{\mathcal{X}}$ and $\mu_{\mathcal{X}}$ are the means of $\mathcal{X}_t$ and $\mathcal{S}_t$, respectively. $\sigma_{\mathcal{X}}$ and $\sigma_{\mathcal{S}}$ are the standard deviations. By evaluating the maximum of $\rho(\tau)$, we will be able to find the causal relationship between social motion and social saliency.

6.3 Evaluation

In this chapter, we propose a method to recognize socially salient motion by inferring the relationship between social motion and social saliency. The main focus of evaluations is how well our inference of social saliency predicts real motion of people's attention. Our inference model must be able to learn the causal relationship from a social scene dataset, prioritize/reason about contributions of social motion, and anticipate group behaviors. We will measure accuracy and robustness of social saliency prediction via quantitative and qualitative evaluations. The baseline of our evaluations will be a linear predictor where social saliency is fired whenever there is motion of a socially salient person. By comparing with the baseline predictor, we will show that all motion from the socially salient person is not always socially salient motion. As a quantitative evaluation, we will design an experiment where a lecturer performs a pre-defined motion and audiences respond to the motion. We will measure how frequently the pre-defined lecturer's motion drives motion of gaze concurrences, classify types of socially salient motion and study how magnitude, speed, or abnormality of motion affects the responses. We will also evaluate error between our prediction of social saliency and ground truth data obtained by gaze estimation and compare our prediction with the baseline prediction (the linear predictor). These quantitative evaluations will allow us to understand how effectively social motion links to socially saliency and which factor of social motion is important. As a qualitative evaluation, we will apply our algorithm to real social scenes where socially salient motion governs group behavior. We are planning to collect data from a lecture, a sporting event, a party, and a theatrical event. Based on 3D motion reconstruction, we will identify motion that drives social saliency and interpret physical meaning of the relationship. Also, we will predict motion of social saliency and see how it deviates from reality.

Chapter 7

Discussion

Humans are social animals and, as vision is our primary perceptual sensations, we interact with others using visible social signals. Social scene understanding is the ability to interpret a scene in terms of this social context. While social scene understanding is key to enabling an artificial agent to collaborate with humans, robotics and computer vision research has largely focused on structural scene understanding. In contrast to structural scene understanding, social scene understanding has two fundamental challenges. First, social scenes usually contain time-varying structures, such as humans. This implies that there is only one opportunity to measure the scene at each time instant. Second, interpreting the social scene may be subjective as it depends on the particular group of people and their particular relationships. These challenges make the problem of social scene understanding difficult to represent and underdetermined in terms of classic (structural) scene understanding. In this thesis, we establish a computational basis for these challenges towards understanding the relationship between social motion and social saliency.

In the first part of our work, we present a method to reconstruct motion in 3D from first person cameras. 3D reconstruction of a moving point from a series of 2D projections is an ill-posed problem. Since the point moves between image captures, triangulation methods become inapplicable. We compactly represent a trajectory with a linear combination of basis trajectories. Linear trajectory representation in conjunction with image projection constraints results in a least squares system for the trajectory parameters. We also study 3D reconstruction of an articulated trajectory. The human body is an articulated structure and trajectories of adjacent joints are spatially constrained, i.e., the distance between two adjacent joints remains constant across time instances. We apply spatial and temporal constraints simultaneously. We show that this problem is a binary quadratic programming problem and solve it using a branch-and-bound method. From 3D reconstruction of motion in 3D, we resolve the challenge involved with time-varying structure.

In the second part of this work, we present an algorithm to reconstruct social saliency from first person cameras. A gaze concurrence is a 3D point where gaze directions from multiple people converge. That point is socially salient because many people cognitively attend to it. We model gaze as a cone-shaped distribution emitted from the center of eyes. The cone-shaped gaze model captures the variation of eye-in-head motion with respect to the primary gaze ray. The gaze model is calibrated by exploiting the fixed relationship between the first person camera and the gaze model. By super-imposing all gaze models in 3D, a social saliency field is constructed.

Gaze concurrences are the modes of the social saliency field and we localize the modes using a provably convergent mean-shift algorithm. The number and 3D locations of gaze concurrences are automatically estimated. By using the concurrences of the gaze of multiple subjects, we show how subjective interpretations approach objectivity through consensus.

In the last part, we propose a method to reconstruct socially salient motion in 3D by inferring the relationship between social motion and social saliency. 3D reconstruction of motion from unstructured first person cameras includes the correspondence problem of wide baseline images. We address two approaches to resolve the problem: the Lagrangian and Eulerian approaches. For the Lagrangian approach, we will exploit the motion information of a 3D point to find trajectory correspondences. We will study an epipolar constraint for corresponding trajectories and develop a descriptor that is consistent for all 2D projected trajectories. For the Eulerian approach, we will apply 3D scene flow that does not require point correspondences. This approach will enable us to find locations where motion takes place. To apply physical constraints on motion, we will show how to estimate a joint trajectory of human body in 2D and how to cluster 3D trajectories to fit into human articulated body motion. From 3D reconstruction of motion in conjunction with gaze concurrences, we will present an algorithm to infer the relationship between social motion and social saliency. We will represent motion and saliency in a spatio-temporal domain. Inspired by the fact that social saliency is driven by social motion, we will find correlation between time-lag signals via a measure of cross-correlation. This will allow us to computationally understand their relationship and identify socially salient motion.

Motion that is very unpredictable or complex is often socially salient. People likely focus on what they have not seen. For example, an obnoxious person at a restaurant draws significant attention from guests at the restaurant. First person cameras would capture this socially salient motion. From a 3D reconstruction point of view, unpredictable or complex motion is hard to estimate because motion priors learned from a database or corpus are not applicable. However, from a sampling point of view, that motion will be well sampled by many different views. This is a key benefit of first person cameras for social scene understanding. Human intelligence is naturally encoded in the sampling procedure, which enables us to secure many views and reconstruct the scene accurately and robustly. Thus, the more unpredictable and complex a motion is, the more attention the activity would draw, and therefore it would enable accurate and robust reconstruction from first person views.

An interesting question of social scene imaging is "given socially salient motion, which first person video can represent the scene best?". There are multiple first person cameras that look at the same salient structure but some may be not taken from a good view point. How can we evaluate the videos? A good video must clearly capture the socially salient motion while it fulfill aesthetic requirements. This evaluation will be important because people often want to organize huge video data acquired by first person cameras. Building a real-time system is another future direction of social scene understanding. An artificial agent needs real-time social information to respond to it. Many parts in our thesis are processed in a batch or are computationally very expensive. Thus, acceleration of computation by either software or hardware is highly relevant research direction. Beyond vision sensors, an audio sensor that is often embedded in a video camera is a promising sensor that will provide critical information about a scene through voice, tone, and speech content. Integrating vision and audio will generate significant synergy for social scene understanding.

Throughout this thesis, we assess that 3D reconstruction of socially salient motion can provide computational representations of social scene understanding. This thesis is a first step towards answering our original question, "what does it mean to understand a social scene?". We think that there are a number of directions to go forward to answering the question via visual perception, action recognition, social networks, and social psychology. We pursue embedding these research in scene understanding to create *social intelligence* for a machine.

The proposed schedule is as follows:

	2012						2013					
	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Journal for ECCV 10 paper												
Journal for NIPS 12 paper												
Socially salient motion study												
Journal preparation												
Job search												
Thesis writing												
Potential targeting conferences					CVPR				ICCV			

Table 7.1: Proposed schedule

Bibliography

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006. 2.2.3

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *Proceedings of the International Conference on Computer Vision*, 2009. 2.1

[3] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *Advances in Neural Information Processing Systems*, 2008. 2.2.2, 3.2.1, 3.4, 3.4.1, 3.4.1, 3.8

[4] I. Akhter, Y. Sheikh, and S. Khan. In defense of orthonormality constraints for nonrigid structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2.2.1

[5] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 4.2, 4

[6] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 2.2.2, 3.4

[7] K. Allemand, K. Fukuda, T. M. Liebling, and E. Steiner. A polynomial case of unconstrained zero-one quadratic optimization. *Mathematical Programming*, 2001. 4.3.2

[8] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 4.5.1

[9] D. E. Angelaki and B. J. M. Hess. Control of eye orientation: where does the brain's role end and the muscle's begin? *European Journal of Neuroscience*, 2004. 2, 5.2.1

[10] T. Asano, D. Z. Chen, N. Katoh, and T. Tokuyama. Polynomial-time solutions to image segmentation. In *Proceedings of SIAM-ACM Conference on Discrete Algorithms*, 1996. 1

[11] S. Avidan and A. Shashua. Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000. 2.2.2

[12] P. Ballard and G. C. Stockman. Controlling a computer via facial aspect. *IEEE Transactions on Systems, Man and Cybernetics*, 1995. 2.3.2

[13] F. Barahona. A solvable case of quadratic 0-1 programming. *Discrete Applied Mathemat-*

ics, 1986. 4.3.2

[14] C. Barron and I. A. Kakadiaris. Estimating anthropometry and pose from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000. 2.2.3

[15] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. I. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2.2.1

[16] L. Bazzani, D. Tosato, M. Cristani, M. Farenzena, G. Pagetti, G. Menegaz, and V. Murino. Social interactions by visual focus of attention in a three-dimensional environment. *Expert Systems*, 2011. 2.3.2

[17] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr. A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, 2010. 4.5.1

[18] N. Bilton. Behind the google goggles, virtual reality. *The New York Times*, Febrary 2012. 5.1

[19] M. Brand. Morphable 3D models from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001. 2.2.1

[20] M. Brand. A direct method for 3D factorization of nonrigid motion observed in 2D. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 2.2.1

[21] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999. 2.2.1

[22] M. A. Brubaker and D. J. Fleet. The kneed walker for human pose tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2.2.3, 4.1

[23] J. Y. Campbell, A. W. Lo, and C. MacKinlay. *The Econometrics of Financial Markets*. Princeton University Press, 1997. 6.2.3

[24] K. Choo and D. J. Fleet. People tracking using hybrid monte carlo filtering. In *Proceedings of the International Conference on Computer Vision*, 2005. 2.2.3, 4.1

[25] J. P. Costeira and T. Kanade. A multibody factorization method from independently moving objects. *International Journal of Computer Vision*, 1998. 2.2.3

[26] M. Cristani, L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. D. Bue, G. Menegaz, and V. Murino. Social interaction discovery by statistical analysis of F-formations. In *British Machine Vision Conference*, 2011. 2.3.2

[27] E. de Aguiar, L. Sigal, A. Treuille, and J. K. Hodgins. Stable spaces for real-time clothing. *ACM Transactions on Graphics (SIGGRAPH)*, 2010. 2.2.1

[28] A. Del Bue. A factorization approach to structure from motion with shape priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2.2.1

[29] A. Del Bue, X. Llad, and L. Agapito. Non-rigid metric shape and motion recovery from

uncalibrated images using priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 2.2.1

[30] N. Emery. The eyes have it: the neuroethology, function and evolution of social gaze. *Neuroscience and Biobehavioral Reviews*, 2000. 2.3.2

[31] S. Eubank, H. Guclu, V. S. A. Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkal, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 2004. 2.3.1

[32] A. Fathi, J. K. Hodgins, and J. M. Rehg. Social interactions: A first-person perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2.3.2

[33] O. Faugeras, Q. Luong, and T. Papadopoulou. *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press, 2001. 2.1

[34] J. Fayad, L. Agapito, and A. Del Bue. Piecewise quadratic reconstruction of non-rigid surface from monocular sequences. In *Proceedings of the European Conference on Computer Vision*, 2010. 2.2.1

[35] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003. 1, 1.1, 5.1

[36] J.-A. Ferrez, K. Fukuda, and T. M. Liebling. Solving the fixed rank convex quadratic maximization in binary variables by a parallel zonotope construction algorithm. *European Journal of Operations Research*, 2004. 4.3.2

[37] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 5.2, 5.2.2

[38] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 1975. 5.2.3

[39] M. R. Garey and D. S. Johnson. *Computer and Interactability: A guide to the theory of NP-Completeness*. Freeman, 1979. 4.1

[40] A. H. Gee and R. Cipolla. Determining the gaze of faces in images. *Image and Vision Computing*, 1994. 2.3.2

[41] E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *Proceedings of the 27th international conference on Human factors in computing systems*, 2009. 2.3.1

[42] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 1969. 6.2.3

[43] S. Gu. Polynomial time solvable algorithms to binary quadratic programming problems with q being a tri-diagonal or five-diagonal matrix. In *Proceedings of the International Conference on Wireless Communications and Signal Processing*, 2010. 4.3.2

[44] E. D. Guestrin and M. Eizenman. General theory of remote gaze estimation using the pupil center and corneal reflection. *IEEE Transactions on Biomedical Engineering*, 2006. 2.3.2

[45] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From scene geometry to human workspace. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1.1, 5.1

[46] M. Hamidi and J. Pearl. Comparison of the cosine and fourier transforms of markov-i signal. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1976. 3.4, 4

[47] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000. 2.2.2

[48] R. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997. 3.4

[49] R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *Proceedings of the European Conference on Computer Vision*, 2008. 2.2.4, 1

[50] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 2.1, 2.2.4, 3.1, 3.2.1, 3.2.3, 3.4, 5.2

[51] M. J. B. Hedvig Sidenbladh and D. J. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *Proceedings of the European Conference on Computer Vision*, 2000. 2.2.2, 2.2.3

[52] C. Hennessey and P. Lawrence. 3D point-of-gaze estimation on a volumetric display. In *Symposium on Eye tracking research & applications*, 2008. 2.3.2

[53] N. R. Howe, M. E. Leventon, and W. T. Freeman. Bayesian reconstruction of 3D human motion from single-camera video. In *Advances in Neural Information Processing Systems*, 1999. 2.2.3, 4.1

[54] Z. L. Husz, A. M. Wallace, and P. R. Green. Evaluation of a hierarchical partitioned particle filter with action primitives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2007. 4.5.1

[55] R. S. Jampel and D. X. Shi. The primary position of the eyes, the resetting saccade, and the transverse visual head plane. head movements around the cervical joints. *Investigative Ophthalmology and Vision Science*, 1992. 3

[56] G. Jansson, S. S. Bergstorm, and W. Epstein. *Perceiving Events and Objects*. Lawrence Erlbaum, 1994. 4.1

[57] R. E. Kalman. Mathematical description of linear dynamical systems. *Jounal of SIAM on Control*, 1963. 3.3.3

[58] J. Y. Kaminski and M. Teicher. A general framework for trajectory triangulation. *Journal of Mathematical Imaging and Vision*, 2004. 2.2.2

[59] L. B. Kara and K. Shimada. Sketch-based 3d-shape creation for industrial styling design. *IEEE Computer Graphics and Applications*, 2007. 2.2.1

[60] L. B. Kara, C. M. D'Eramo, and K. Shimada. Pen-based styling design of 3d geometry using concept sketches and template models. In *Proceedings of the ACM symposium on Solid and physical modeling*, 2006. 2.2.1

[61] E. M. Klier, H. Wang, A. G. Constantin, and J. D. Crawford. Midbrain control of three-dimensional head orientation. *Science*, 2002. 2, 5.2.1

[62] H. W. Lauw, E.-P. Lim, H. Pang, and T.-T. Tan. Social netwrok discovery by mining spatio-temporal events. *Computational and Mathematical Organization Theory*, 2005. 2.3.1

[63] H.-J. Lee and Z. Chen. Determination of 3D human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 1985. 4.1

[64] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *IJCV*, 2009. 5.2

[65] D. Li, J. Babcock, and D. J. Parkhurst. openEyes: a low-cost head-mounted eye-tracking solution. In *Symposium on Eye-Tracking Research and Application*, 2006. 2.3.2

[66] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981. 2.1, 2.2, 6.2.1

[67] M. I. A. Lourakis and A. A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 2009. 3.4.2

[68] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004. 2.1, 3.4.2, 6.2.1

[69] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003. 2.1

[70] S. Marks, B. Wünsche, and J. Windsor. Enhancing virtual environment-based surgical teamwork training with non-verbal communication. In *International Conference on Computer Graphics Theory and Applications*, 2009. 5.1

[71] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Phenomenology and the Cognitive Sciences, 1982. 5.1

[72] P. Merz and B. Freisleben. Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, 2002. 4.1

[73] S. Milgram. The small world problem. *Psychology Today*, 1967. 2.3.1

[74] H. Misslisch, D. Tweed, and T. Vilis. Neural constraints on eye motion in human eye-head saccades. *Journal of Neurophysiology*, 1998. 2, 5.2.1

[75] C. Moore and P. Dunham. *Joint Attention: Its Origins and Role in Development*. Lawrence Erlbaum Associates, 1995. 1

[76] S. M. Munn and J. B. Pelz. 3D point-of-regard, position and head orientation from a portable monocular video-based eye tracker. In *Symposium on Eye-Tracking Research and Application*, 2008. 2.3.2

[77] R. R. Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man and Cybernetics*, 2004. 5.1

[78] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009. 2.3.2

[79] B. Mutlu, J. K. Hodgins, and J. Forlizzi. A storytelling robot: Modeling and evaluation of human-like gaze behavior. In *IEEE-RAS International Conference on Humanoid Robots*, 2006. 5.1

[80] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 2002. 2.3.1

[81] B. Noris, K. Benmachiche, and A. G. Billard. Calibration-free eye gaze direction detection with gaussian processes. In *International Conference on Computer Vision Theory and Applications*, 2006. 2.3.2

[82] S. Olsen and A. Bartoli. Using priors for improving generalization in non-rigid structure-from-motion. In *Proceedings of British Machine Vision Conference*, 2007. 2.2.1

[83] C. Olsson, A. P. Eriksson, and F. Kahl. Solving large scale binary quadratic problems: Spectral methods vs. semidefinite programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 4.1

[84] K. E. Ozden, K. Cornelis, L. V. Eycken, and L. V. Gool. Reconstructing 3D trajectories of independently moving objects using generic constraints. *Computer Vision and Image Understanding*, 2004. 4.1

[85] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito. Factorization for non-rigid and articulated structure using metric projections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2.2.1, 3.4.1, 3.4.1, 3.8

[86] V. Parameswaran and R. Chellappa. View independent human body pose estimation from a single perspective image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 2.2.3

[87] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3D reconstruction of a moving point from a series of 2D projections. In *Proceedings of the European Conference on Computer Vision*, 2010. 3.2.2, 3.5, 4.1, 4.4, 6.2.1

[88] P. Peursum, S. Venkatesh, and G. West. A study on smoothing for particle-filtered 3D human body tracking. *International Journal of Computer Vision*, 2010. 4.5.1

[89] J. C. Picard and P. M. Ratliff. Minimal cost cut equivalent networks. *Management Science*, 1973. 4.3.2

[90] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, 1997. 2.2.1

[91] F. Pirri, M. Pizzoli, and A. Rudi. A general method for the point of regard estimation in 3d space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2.3.2

[92] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for (0,1)-quadratic prgramming. *Journal of Global Optimization*, 1995. 4.1

[93] I. D. Pool and M. Kochen. Contacts and influence. *Social Networks*, 1978. 2.3.1

[94] R. Rae and H. J. Ritter. Recognition of human head orientation based on artificial neural

networks. *IEEE Transactions on Neural Networks*, 1998. 2.3.2

[95] N. Robertson and I. Reid. Estimating gaze direction from low-resolution faces in video. In *Proceedings of the European Conference on Computer Vision*, 2006. 2.3.2

[96] M. Salzmann and P. Fua. *Deformable Surface 3D Reconstruction from a Single Viewpoint*. Morgan-Claypool, 2010. 2.2.1

[97] J. Sánchez-Riera, J. Öslund, P. Fua, and F. Moreno-Noguer. Simultaneous pose, correspondence and non-rigid shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2.2.1

[98] A. Shaji, A. Varol, L. Torresani, and P. Fua. Simulataneous point matching and 3D deformable surface reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2.2.1

[99] A. Shashua and L. Wolf. Homography tensors: On algebraic entities that represent three views of static or moving planar points. In *Proceedings of the European Conference on Computer Vision*, 2000. 2.2.2, 1

[100] T. Shiratori, H. S. Park, L. Sigal, Y. Sheikh, and J. K. Hodgins. Motion capture from body-mounted cameras. *ACM Transactions on Graphics (SIGGRAPH)*, 2011. 2.1, 5.1, 5.2, 6.1

[101] K. Smith, S. Ba, J.-M. Odobez, and D. Gatica-Perez. Tracking the visual focus of attention for a varying number of wandering people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008. 2.3.2

[102] R. C. Smith and P. Cheeseman. On the represetation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 1986. 1

[103] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (SIGGRAPH)*, 2006. 1.1, 2.1, 3.4.2, 5.1, 5.2, 6.2.1

[104] R. Stiefelhagen, M. Finke, J. Yang, and A. Waibel. From gaze to focus of attention. In *International Conference on Visual Information and Information Systems*, 1999. 2.3.2

[105] K. Takemura, Y. Kohashi, T. Suenaga, J. Takamatsu, and T. Ogasawara. Estimating 3D point-of-regard and visualizing gaze trajectories under natural head movements. In *Symposium on Eye-Tracking Research and Application*, 2010. 2.3.2

[106] C. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, 2000. 2.2.3

[107] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2.2.1

[108] J. R. Tena, F. De la Torre, and I. Matthews. Interactive Region-Based Linear 3D Face Models. *ACM Transactions on Graphics (SIGGRAPH)*, 2011. 2.2.1

[109] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 1992. 2.1, 2.2.1, 6.2.1

[110] L. Torresani and C. Bregler. Space-time tracking. In *Proceedings of the European Con-*

ference on Computer Vision, 2002. 2.2.1, 2.2.2

[111] L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001. 2.2.1

[112] L. Torresani, A. Hertzmann, and C. Bregler. Learning non-rigid 3D shape from 2D motion. In *Advances in Neural Information Processing Systems*, 2003. 2.2.1

[113] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008. 2.2.1, 2.2.4, 3.4.1, 3.4.1, 3.8

[114] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, 2000. 2.1, 6.2.1

[115] R. Urtasun, D. J. Fleet, and P. Fua. Temporal motion models for monocular and multiview 3-D human body tracking. *Computer Vision and Image Understanding*, 2006. 2.2.3, 4.1

[116] J. Valmadre and S. Lucey. Deterministic 3D human pose estimation using rigid structure. In *Proceedings of the European Conference on Computer Vision*, 2010. 2.2.3, 4.1

[117] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the International Conference on Computer Vision*, 1999. 6.2.1

[118] S. Vedula, S. Baker, and T. Kanade. Image-based spatio-temporal modeling and view interpolation of dynamic events. *ACM Transactions on Graphics*, 2005. 6.2.1

[119] R. Vidal and D. Abretske. Nonrigid shape and motion from multiple perspective views. In *Proceedings of the European Conference on Computer Vision*, 2006. 2.2.4

[120] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 2009. 1, 5.1

[121] H. von Storch. *Statistical Analysis in Climate Reseach*. Cambridge University Press, 2002. 6.2.3

[122] M. Vondrak, L. Sigal, and O. C. Jenkins. Physic simulation for probabilistic motion tracking. In *Proceedings of the European Conference on Computer Vision*, 2008. 2.2.3, 4.1

[123] J.-G. Wang and E. Sung. Study on eye gaze estimation. *IEEE Transactions on Systems, Man and Cybernetics*, 2002. 2.3.2

[124] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 1998. 2.3.1

[125] X. Wei and J. Chai. Modeling 3D human poses from uncalibrated monocular images. In *Proceedings of the International Conference on Computer Vision*, 2009. 2.2.3, 4.1

[126] X. Wei and J. Chai. Videomocap: Modeling physically realistic human motion from monocular video sequences. *ACM Transactions on Graphics (SIGGRAPH)*, 2010. 2.2.3, 4.1

[127] G. Welch and E. Foxlin. Motion tracking: no silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications*, 2002. 2.3.2

[128] Y. Wexler and A. Shashua. On the synthesis of dynamic scenes from reference views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000. 2.2.2

[129] L. Wolf and A. Shashua. On projection matrices $\mathcal{P}^k \rightarrow \mathcal{P}^2, k = 3, \cdots, 6$, and their applications in computer vision. *International Journal of Computer Vision*, 2002. 2.2.2

[130] J. Xiao and T. Kanade. Non-rigid shape and motion recovery: Degenerate deformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 2.2.1

[131] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 2006. 2.2.1

[132] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 2.2.3

[133] J. Yan and M. Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 2.2.3

[134] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures of parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 6.2.2

[135] S. Zhu, L. Zhand, and B. M. Smith. Model evolution: An incremental approach to non-rigid structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 2.2.4