Winter 2-2017

Measuring Human Motion in Social Interactions

Tomas Simon
Carnegie Mellon University

Measuring Human Motion in Social Interactions

Tomas Simon
tsimon@cs.cmu.edu

CMU-RI-TR-17-03

Submitted in partial fulfillment
of the requirements of the degree of
Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

February, 2017

Doctoral Committee:

Yaser Sheikh, *Carnegie Mellon University (CMU)* (Chair)	yaser@cs.cmu.edu
Iain Matthews, *CMU* (Chair)	iainm@cs.cmu.edu
Fernando De la Torre, *CMU*	ftorre@cs.cmu.edu
Brooke Feeney, *CMU*	bfeeney@andrew.cmu.edu
David Fleet, *University of Toronto*	fleet@cs.toronto.edu

Abstract

This thesis develops methods for social signal reconstruction—in particular, we *measure human motion during social interactions*. Compared to other work in this space, we aim to measure the entire body, from the overall body pose to subtle hand gestures and facial expressions. The key to achieving this without placing markers, instrumentation, or other restrictions on participants is the *Panoptic Studio*, a massively multi-view capture system which allows us to obtain 3D reconstructions of room-sized scenes.

To measure the position of joints and other landmarks on the human body, we combine the output of 2D keypoint detectors across multiple views and triangulate them in 3D. We develop a semi-supervised training procedure, *multi-view bootstrapping*, which uses 3D triangulation to generate training data for keypoint detectors. We use this technique to train fine-grained 2D keypoint detectors for landmarks on the hands and face, allowing us to measure these two important sources of social signals.

To model human motion data, we present the *Kronecker Markov Random Field* (KMRF) model for keypoint representations of the face and body. We show that most of the covariance in natural body motions corresponds to a specific set of spatiotemporal dependencies which result in a Kronecker or matrix normal distribution over spatiotemporal data, and we derive associated inference procedures that do not require training sequences. This statistical model can be used to infer complete sequences from partial observations and unifies linear shape and trajectory models of prior art into a probabilistic shape-trajectory distribution that has the individual models as its marginals.

Finally, we demonstrate full-body motion reconstructions by using the KMRF model to combine the various measurements obtained from the Panoptic Studio. We capture a dataset of groups of people engaged in social games and fit mesh models of the body, face, and hands—a representation that encodes many of the social signals that characterize an interaction and can be used for analysis, modeling, and animation.

Acknowledgements

The Panoptic Studio is the brainchild of Yaser Sheikh and was materialized mainly by Hanbyul Joo, Lei Tan, Shohei Nobuhara, Tim Godisart, Sean Banerjee, Lin Gui, Hao Liu, Hyun Soo Park, Minh Vo, Gaku Nakano, Taiki Shimba, Xulong Li, Fanyi Xiao, Shih-En Wei, Yair Movshovitz-Attias, Varun Ramakrishna, Natasha Kholgade, Yiying Li, and Iain Matthews. Without their work, the captures shown in this document would not have been possible.

In addition, I also thank Paulo Gotardo, Ijaz Akhter, Jack Valmadre, Zhe Cao, Fernando de la Torre, Jeffrey Cohn, Feng Zhou, Javier Hernandez, Liz Carter, Alberto Rodriguez, Nuria Jane, Sohaib Khan, Aayush Bansal, Moshe Mahler, Alvaro Collet, Rafael Tena, Jiuguang Wang, Patrick Lucey, Sarah Taylor, Simon Baker, Kit Ham, and my family for ideas, discussions, help, and support.

Contents

Chapter 1.

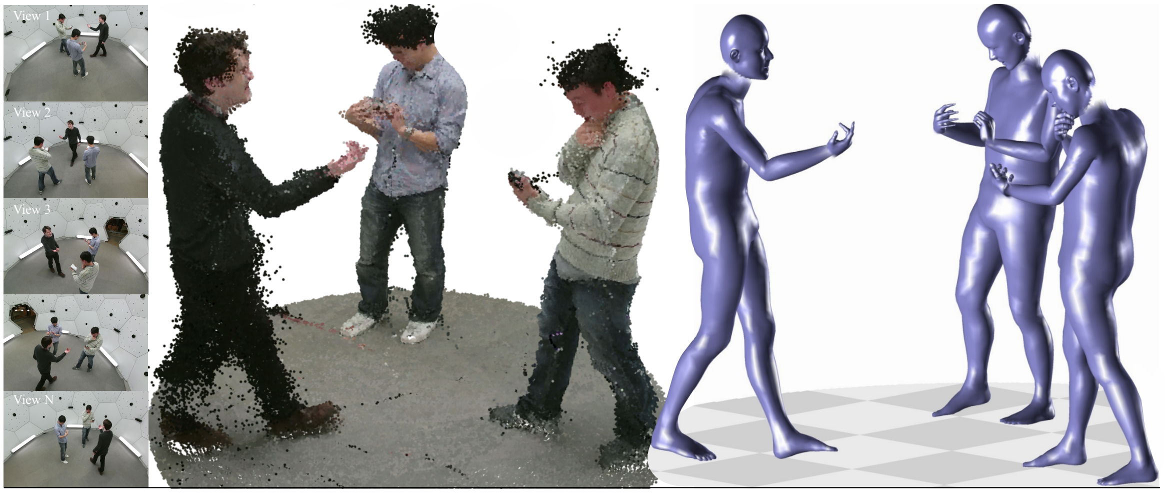
Introduction

Our social signals—the protocols that mediate our daily interactions—are readily observable and yet poorly understood. Sapir [1928] eloquently described it as an "*elaborate and secret code that is written nowhere, known by none, and understood by all.*" However, to understand this code, we first need to be able to measure the raw underlying signals involved. To this end, this thesis develops methods to measure the movements of interacting people: we present a capture system for social signal reconstruction, and statistical models of spatiotemporal data that are useful for the computational analysis of human motion.

The study of nonverbal communication has a long history. In the academic literature, it dates back at least to 1872, with the publishing of Darwin's treatise on the expression of emotion in animals [Darwin 1872]. Even in the popular press, so-called self-improvement books on the subject—with sensationalist titles such as "How to Read Others' Thoughts by Their Gestures" [Pease 1981]—have been on best-seller lists since the early 80s. And yet, to date, we lack the ability to conduct statistical analyses on how people move their bodies to communicate. Our understanding of nonverbal communication is in its infancy, especially when compared to natural language processing or speech processing. A key component to major advances in these subfields has been the availability of large amounts of training data. We believe that when large quantities of motion data of human interactions become available, similar strides will be made towards a computational understanding of human behavior.

A major hindrance to obtaining this kind of data has been the inherent difficulty in recording, measuring, and quantifying social signals. While we can record video of a person's motions as easily as we can record audio of a person speaking[1], the viewpoint from which a video is taken can completely change the recorded signal. Conversely, the placement of a microphone has little effect on recorded audio. So, while many of the confounds are similar when processing speech compared to human motion signals, the variability in recorded audio is far, far lower. More importantly, speech can be processed as a language; it has strict grammatical rules, unambiguous meaning, and a set of well-understood elements for which we have computational and statistical models [Huang et al. 2001]—for example, in order of increasing granularity, one might single out cepstral coefficients, phonemes, n-gram models, and grammar models.

[1] *About* as easily: the compressed data rate is around two orders of magnitude higher (e.g., *YouTube* recommends high-quality audio at 512 kbps but video at $50,000$ kbps).



(a) Multiview Input (b) 3D Point Clouds from RGB-D Sensors (c) Parameterizations of the body, face, and hands.

Figure 1.1: The Panoptic studio can (a) record video and RGB-D images from multiple vantage points, (b) reconstruct room-sized scenes in 3D, and (c) determine body pose, hand pose, and facial expression for socially interacting people.

The elements of nonverbal communication are comparatively far less understood. In 1952, Birdwhistell [1952] coined the term "kinemes"—analogously to phonemes—to denote these elements, and the corresponding term "kinesics"—analogously to phonetics—to denote their study. Setting aside for a moment the question of whether this analogy is apt, it is clear that a computational model for kinesics (as automatic speech recognition is to phonetics) would be desirable and has yet to be developed. However, obtaining data for the development of such computational models is not straightforward. While large amounts of video data of interacting people exist, extracting measurements of nonverbal signals from this data is difficult for two reasons: First, people move their bodies for *many* purposes besides nonverbal communication, masking the signal of interest. Second, measuring 3D body movements from video is difficult. This problem, known as pose detection or estimation, implies determining the 3D spatial position and orientation of individual limbs from 2D camera views [Forsyth et al. 2005]. In the particular case of capturing group interactions, the problem is greatly exacerbated by inter-occlusions between interacting people, which unavoidably render the view from any single camera's vantage point incomplete.

To tackle this problem, we design a capture environment, the "Panoptic Studio", a massively multiview system equipped with multiple RGB and RGB-D sensors (capturing color and depth) to minimize data loss due to occlusions. Fig. 1.1 shows a scene within the capture room, a dome-like structure with cameras and RGB-D sensors mounted across its exterior. In Chapter 3, we present methods to calibrate this system, reconstruct 3D scenes, and recover the 3D pose of interacting people within the capture space. Crucially, our system avoids the need for instrumenting the participants with specialized suits or markers, as would be required by other alternatives such as motion capture systems. While motion capture systems yield very accurate 3D body motion, the suit and markers hinder natural interactions and movements, and are limited to capturing only the motion of things on which we can place a marker (which is particularly cumbersome when capturing facial motion and hand movements). We believe that the naturalness and spontaneity of the interactions that we get by having people simply walk into the capture room—as they would into any other room, with no setup or preparation—greatly outweighs the added computational difficulty of recovering body pose. Additionally, to ensure that the data that we capture predominantly contains nonverbal signaling rather than other types of body motion, we design a study protocol wherein participants are asked to play *social games*, such as the "Ultimatum" game. These are games that feature a significant social component and are commonly used in behavioral studies. For our purposes, however, we use these games simply as a way of eliciting lively discussions between participants, where we expect nonverbal behavior to be readily observable—and more importantly, in which we expect motions with a functional objective to be minimized.

A key difficulty in measuring the social signals of multiple interacting people is that it requires capturing subtle details across a large space—a space large enough to allow people to move freely and naturally. However, social signals are expressed at a variety of spatial and temporal scales, including small and subtle motions such as facial expression and hand gestures [Kendon 1994]. None of the markerless motion capture systems we reviewed for this work were able to simultaneously reconstruct hand pose and facial expressions for multiple people who are unconstrained in their motions. We address this in Chapter 4, where we design 2D keypoint detectors for the joints of the hand, as well as facial keypoint detectors that work across the large variation in viewing angle that occurs in the Studio. In particular, we present an algorithm that bootstraps the generation of training data for a keypoint detector using multiview 3D triangulation as supervision.

However, due to occlusions our keypoint detections and other measurements (e.g., depth) are often contaminated by either noise or missing data. We therefore need a prior model to be able to infer a consistent and temporally smooth estimate of the social signals of participants. To this end, in Chapter 5 we present a probabilistic model that captures the spatiotemporal correlations in motion data, and allows inference of complete reconstructions from partial observations. The model subsumes linear shape and trajectory models of prior art into a single, probabilistic representation that has these individual distributions as its marginals, using a Kronecker-Markov correlation structure.

Finally, in Chapter 6, we present a method that combines the measurements from Panoptic Studio capture system (Chapter 3) with fine-grained keypoint detections across multiple views (Chapter 4) and the Kronecker-Markov prior (Chapter 5) to reconstruct the social signals of interacting people. We parameterize the combined body pose, facial expression, and hand pose in terms of a mesh model that is fit to all available measurements (see Fig. 1.1c). This parameterization is common across different subjects, and factors variations in the raw measurements—the geometric position of joints and other landmarks on the body—into subject-specific shape or identity parameters, and a set of time-varying *expression* or pose parameters. Such a factored parameterization is crucial to finding commonalities across the social signals evinced by different people.

Importantly, we currently lack a formal framework for understanding how we coordinate movements across different body parts. Partly, this difficulty arises because social signals often involve subtle changes in the relative position of several body parts simultaneously (e.g., changes in posture, facial expression, and hand pose). These spatiotemporal relations are difficult to describe, let alone transcribe. *Kinetography*, or the transcription of movement (as is used for example by choreographers), requires complex notational systems such as *Laban Movement Analysis* or *Labanotation* [Laban 1926]. These notational systems must simultaneously record the timing and the configuration of multiple body parts as well as subjective qualities of the movement, such as “weight” and “effort”. The space of possible nonverbal signals is clearly complex, with large variability in the execution of movements, and, importantly, also in the *interpretation* of these movements. Contrary to language, in which signs or symbols have distinct meaning and are combined using specific grammatical rules, the vocabulary of bodily expression is ambiguous, highly dependent on context, and may even be a function of timing or synchrony rather than a particular motion [Ramseyer and Tschacher 2014]. This aspect of social interaction makes it a distinct problem from that of learning a general dictionary of atomic motions, such as the *movemes* of Bregler [1997] or their non-periodic extensions [Vecchio et al. 2002]. In contrast to these approaches, rather than representing goal-driven units of action (such as “grab”, “kick”, or “pull”), nonverbal behavior communicates information: a social signal to be interpreted by others.

In this manner, clusters of motions that have a predictable, repeatable, and observable effect should be identified as nonverbal signals that carry meaning. We call these motions with observable effects “sociemes”, i.e., a unit of social signaling. We believe that identifying a set of computationally recognizable sociemes could represent a significant step towards a computational understanding of behavior. For example, our understanding of facial expressions—merely one aspect of nonverbal communication—was advanced substantially by the development of FACS, the Facial Action Coding System [Ekman and Friesen 1978]. FACS is a taxonomy of the observable (i.e., humanly distinguishable) components of facial expressions—essentially an illustrated manual describing the appearance of certain facial actions. Originally developed as a tool for studies in behavioral psychology, FACS has since found many applications in affective computing: in the animation of interactive virtual agents, it is used in the synthesis of facial expressions; in computer vision, FACS has shown promise in the automatic analysis of pain, deceit, and depression severity, among others (see e.g., [Cohn 2010]).

A practical, computationally tractable taxonomy does not currently exist for the behavioral elements of social interaction—including gestures, head and body motion, gaze, posture, or relative position and orientation—nor do we understand the rules that mediate their communication. In contrast to the "human expert" methodology that was used in developing FACS, our overarching goal is to use statistical analysis to discover these fundamental units of social interaction. This thesis represents a first step toward obtaining the data that can make this possible by reconstructing the social signals of groups of naturally interacting people.

1.1 Thesis Summary

This thesis aims to develop statistical methods and representations to *measure human motion during natural interactions*. The core challenge is reconstructing *subtle motions across a large space*—a necessity when capturing several interacting people without restricting their movements or instrumenting them with markers or specialized suits. This difficulty has created a deficiency in the current research landscape, namely, the lack of databases of 3D human motion data of social interactions reconstructing the hands, the face, and the body simultaneously.

We address this by collecting a dataset of group interactions, which required developing systems and methods to infer the body pose of multiple people simultaneously (Chapter 3). We measure subtle motion by adding fine-grained joint localization for the hands and face, which required developing 2D keypoint detectors and producing the necessary training data (Chapter 4). The application of these single-image detectors to the Panoptic Studio data produces partial observations of a small number of detected landmarks. To infer complete and temporally consistent representations from partial observations, we present a spatiotemporal prior for time-varying 3D data based on the Kronecker structure of the covariance of natural motions (Chapter 5). Finally, we present a method that uses this spatiotemporal prior to integrate the detections and RGB-D data from the Studio into a consistent representation of the body, hands, and face (Chapter 6).

1.1.1 Overview

This document is structured into four parts:

1. Social Interaction Capture. (Chapter 3)
   This chapter discusses the design and capture of a 3D dataset of social interactions with the Panoptic Studio. This work represents several years of collaboration with Hanbyul Joo, most of which was published as

   "Panoptic Studio: A Massively Multiview System for Social Interaction Capture," *Hanbyul Joo, Tomas Simon, Xulong, Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh*, arXiv:1612.03153, 2016 (under review)

   This thesis builds on the markerless body motion capture system described above and we include the most relevant sections as background information. Additionally, we expand on other aspects that were not included in the original publication, particularly on details about the RGB-D subsystem and information about the experimental protocol.

   - Video: (link)
   - Browsable dataset: `http://domedb.perception.cs.cmu.edu`

2. Fine-Grained Joint Detection Using Multiview Bootstrapping. (Chapter 4)
   This chapter adds detailed hand pose and facial keypoint estimation to our system by developing a fine-grained landmark detector for RGB images. To produce the necessary training data, we present *multiview bootstrapping*, a procedure that generates training data using a multiview camera setup and an initial, noisy keypoint detector. We also demonstrate usage of the procedure to train facial landmark detectors that work across a very large range of viewing angles. An earlier version of this work is currently under review (anonymously),

   "Hand Keypoint Detection in Single Images Using Multiview Bootstrapping,"
   Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh, 2016 (under review)

   - Video: (link)

3. Kronecker Markov Random Fields for Human Motion Data (Chapter 5)
   This chapter presents the *Kronecker Markov Random Field* (KMRF) model for keypoint representations of time-varying 3D data. We show that most of the covariance in natural body motions corresponds to a specific set of spatiotemporal dependencies which result in a Kronecker or matrix normal distribution over spatiotemporal data, and we derive associated inference procedures that do not require training sequences. We originally formulated a non-probabilistic version of this model, and subsequently discovered the probabilistic formulation that is presented in this chapter. These developments were published across three different papers, chronologically,

"Bilinear Spatiotemporal Basis Models,"
Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, Yaser Sheikh, ACM Transactions on Graphics (TOG), 2012
"Separable Spatiotemporal Priors for Convex Reconstruction of Time-Varying 3D Point Clouds,"
Tomas Simon, Jack Valmadre, Iain Matthews, Yaser Sheikh, European Conference on Computer Vision (ECCV), 2014
"Kronecker-Markov Prior for Dynamic 3D Reconstruction,"
Tomas Simon, Jack Valmadre, Iain Matthews, Yaser Sheikh, Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2016

- Video: (link)

4. Full-Body Markerless Motion Capture. (Chapter 6)
This chapter demonstrates full-body motion reconstructions by using the KMRF model to combine the various measurements and detections obtained from the Panoptic Studio. We capture a dataset of groups of people engaged in social games and fit mesh models of the body, face, and hands—a representation that encodes many of the social signals that characterize an interaction and can be used for analysis, modeling, and animation. This is unpublished work.

1.2 Notation

This thesis deals with spatiotemporal data that often varies across more than 2 dimensions—typically, several points sampled across several video frames, and along 3 spatial dimensions. Indexing this kind of spatiotemporal data requires more than 2 indices: (1) an index into the set of points from 1 to P, (2) an index into the set of frames from 1 to F, and (3) an index into the set of coordinates describing each point, for example x, y, and z. However, for mathematical convenience, it will often be useful to arrange our data into 2-dimensional matrices and vectors. Different 2D arrangements of the data are possible, and some operations are more intuitively expressed in each of the possible arrangements. I've tried to use notation that will clarify the concepts wherever possible by following a set of conventions:

a	A scalar.
A	A scalar typically assumed to be constant (e.g., P, the number of points).
$\mathbf{v}$	An array or vector (column, unless noted otherwise).
$\mathbf{M}$	A matrix, e.g., $\mathbf{M} \in \mathbb{R}^{M,N}$.
$[\mathbf{A}]_{i,j} = A_{i,j}$	The scalar i, j entry of $\mathbf{A}$.
$\mathbf{X}$	An unknown during an optimization, e.g., $\min_{\mathbf{X}} f(\mathbf{X})$.

$\mathbf{A} \otimes \mathbf{B}$ The Kronecker product of $\mathbf{A}$ and $\mathbf{B}$, i.e.,

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} A_{1,1}\mathbf{B} & \cdots & A_{1,n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ A_{m,1}\mathbf{B} & \cdots & A_{mn}\mathbf{B} \end{bmatrix},$$

I will refer to multi-arrays ("matrices" with more than 2 dimensions) with the same capital boldface notation, so that:

$\mathbf{M}$	A multi-array, e.g., $\mathbf{M} \in \mathrm{R}^{M,N,O}$.
$\mathbf{A}^t$ or $\mathbf{A}_t$	A sub-matrix $\mathbf{A}^t \in \mathbb{R}^{M,N}$ in the multi-array $\mathbf{A} \in \mathbb{R}^{M,N,T}$.
$\mathbf{A}^t_{:,j}$	The j-th column of the matrix $\mathbf{A}^t$.
$\mathbf{A}^\mathsf{T}$	The transpose of matrix $\mathbf{A}$.

Further notation will be defined as it becomes relevant.

Chapter 2.

Related Work

This section reviews related work in capturing and analyzing social signals related to body movements, as well as in statistically modeling and inferring 3D motion of people. While the terms “nonverbal behavior” and “social signal” are typically understood to include any kind of nonverbal cue—for example, audible cues such as intonation or prosody—in this thesis, we will use these terms to refer only to those signals that are expressed through motion.

2.1 Measuring Social Signals

In the following subsections, we briefly review work on measuring or modeling the dynamics of social signals during group interactions. We omit work focused on static understanding of nonverbal cues, and also work on facial expression recognition in isolation. The latter is unquestionably the most researched aspect of nonverbal behavior, see [Sandbach et al. 2012] for a review. Instead, we restrict this review to work modeling motion and gestures during interactions, rather than those pertaining to the recognition and detection of particular indicators.

2.1.1 Overview

The most widely used tool for measuring nonverbal behaviors is probably the camera. Famously, Ekman and Friesen spent countless hours studying video of facial expressions to develop their Facial Action Coding System (FACS) [Ekman and Friesen 1976], but even Darwin used photographs in his studies of the expression of emotion [Darwin 1872], and Birdwhistell [1952] developed his work on kinemes by filming people. Since its invention, the video camera has been an integral tool in behavioral research.

However, extracting behavioral measurements from video typically has to rely on human observer annotations (for example, to count all occurrences of a specific action or gesture). The facial action taxonomy proposed FACS is such an observer-based annotation system, and the de-facto standard for facial measurement in behavioral research [Cohn and Ekman 2005]. The problem with observer-based annotation systems is that they are time consuming to learn and use, requiring trained annotators to spend hours painstakingly analyzing video data. More importantly, these annotations are also difficult to standardize across laboratories [De la Torre et al. 2011] due to subjective interpretation of annotation guidelines. For the case of full body motion, a universally accepted annotation guideline such as FACS does not even exist. In the case of hand pose, only a few signs are readily recognized by name, and most hand pose taxonomies are either associated with sign language or descriptive of manipulation tasks [Feix et al. 2009].

Signals which are easier to record and analyze (e.g., EMG, electrodermal skin response, heart rate, or audio) have also received some attention, and have been used to measure physiological correlates of social signals (e.g., during digital game play [Ekman et al. 2011] or during stressful events [Lundberg et al. 1994]). EMG in particular has been used as a way to precisely measure muscle activation during facial motion [Cohn and Ekman 2005], but requires the placement of surface electrodes on the skin, making it intrusive, impractical, and cumbersome.

2.1.2 Measuring Group Interactions

The dynamics of nonverbal signals have been computationally analyzed predominantly either on individuals in isolation, or during their occurrence in two-person interactions. Analysis of group interactions has been more limited due to difficulties in recording from sufficient vantage points. Regardless of the size of the group, the main lines of research have attempted to (1) detect or recognize specific events, (2) infer attributes, or (3) synthesize virtual animations (see e.g., Vinciarelli et al. [2012] and Rudovic et al. [2014], for extensive reviews on the state of the art).

Studies of body and face motion in isolated individuals in particular are focused on detecting the occurrence of certain events or synthesizing animations. For example, high quality recordings of body motion are extensively used to improve the realism of computer animations, but are available mostly for movements with a functional objective (e.g., the Carnegie Mellon University Motion Capture Database [Carnegie Mellon University]). A variety of video datasets of human motions in individuals have been captured as well (e.g., KTH, Weizmann, UCF-Sports, CMU-MMAC, MSR Action I, II, and 3D [Schuldt et al. 2004; Gorelick et al. 2007; Rodriguez et al. 2008; De la Torre et al. 2009; Yuan et al. 2009; Yu et al. 2015]), and are typically used as a testbed for action detection algorithms. Analysis of the facial dynamics of isolated individuals has also been used to infer attributes about the individuals, such as the occurrence of pain, the preferences of viewers who are shown advertisements [McDuff et al. 2014], or the level of engagement during a learning task [Whitehill et al. 2014].

Dynamical analysis of social signaling during two-person interactions is an area that has garnered substantial interest, especially with the focus of inferring attributes about the interacting participants. For example, a number of machine learning techniques are aimed at inferring measures of affect, such as “valence” and “arousal” (see e.g., Calvo and D’Mello [2010] for a review of affect detection techniques). Similarly, determining social roles and interactions from single images (e.g., [Chakraborty et al. 2013; Ramanathan et al. 2013]), inferring the level of attention (e.g., engagement, synchrony), detecting cues of agreement (e.g., Bousmalis et al. [2013]), measuring depression severity [Cohn et al. 2009], pain [Littlewort et al. 2007], and detecting signs of deception have been major goals of the research agenda as well [Pavlidis et al. 2002; Ekman 2003].

Studies of mimicry come closest to analyzing the exchange as a dynamic process, but again have focused on detecting mimicry with the goal of inferring affect or agreement (e.g., Sun et al. [2011] and Bilakhia et al. [2013]). None of these studies has used a 3D representation of the body, severely limiting the generalizability of what is learned; in particular, the features are necessarily computed from the specific vantage point from which the dataset’s cameras recorded the interaction.

Studies using 3D motion measurements for analysis of interactions (for example, using motion capture) have shown promise in helping understand how nonverbal signals affect our communication. For example, Boker et al. [2011] showed that motion has a greater influence on the occurrence of head nods than, for example, the apparent gender of the speakers. In this study, the capture setup was limited to measuring head motion—our goal is to extend this type of dynamical analysis to the full range of body motion. Full body skeletal motion of individuals has been measured directly using 3D motion capture technology, by far the most reliable method to date to obtain accurate measurements of position and velocity of the human body. However, it is impractical for natural social interactions due to the relatively difficult setup and obtrusiveness of the markers. Setting up a capture session requires a specialized suit (or attaching motion capture markers directly to the face), and additionally requires capturing a calibration sequence with the range of motion of the interacting participants. For these reasons, it has been used in specialized studies, for example, in measuring how social context affects the movements of musicians [Glowinski et al. 2013], or recording performers acting out a social interaction. Similar studies have attempted to measure motion using mobile devices as a hand-held inertial measurement device [Varni et al. 2010]; however, the motion data that can be recovered in this way is limited to the motion of the device itself. Because of this, the focus of this thesis is in capturing human motion using a multiview camera system. With current technology, this seems to us the only viable, non-intrusive option, but sensor miniaturization and other sensing modalities may become more attractive in the near future (e.g., [Adib et al. 2015]).

Indeed, there has been increasing interest over the last decade in automatically analyzing multiple people's social interaction using multiple camera sensors. Video analysis of group interactions has been successful mostly in structured rather than social interactions, such as group formations in sports games (e.g., Liu et al. [2013a]) or the motion of crowds. For social interactions, the AMI Meeting Corpus is perhaps the most extensive video dataset containing group interactions of (contrived) meetings. Studies analyzing the dynamics of non-verbal signals have again been used to infer or measure attributes about the interaction, such as performance or the dominance of participants. Unfortunately, the number of cameras used is insufficient to perform a detailed 3D reconstruction of the body configurations.

Several datasets recording unstructured social scenes have been presented, where multiple people (from 2 to 14 subjects) naturally communicate without restriction in their behavior [Zen et al. 2010; Cristani et al. 2011; Rehg et al. 2013; Alameda-Pineda et al. 2015]. In contrast to the scenes captured in structured environments (such as round-table meetings [Lepri et al. 2012] or constrained dyadic capture [McKeown et al. 2010]), the subjects in unstructured environments show richer social signals in their body motion, locations, and orientations. However, due to the unconstrained nature of the capture, it is much more challenging to measure their body motion, particularly from the relatively small (4–6) number of cameras used. Thus, work in this area usually aims to obtain coarse measurements (e.g., quantized body/head orientation), and they rather focus on higher level descriptors, such as F-formation detection [Lepri et al. 2012] or personality predictions [Alameda-Pineda et al. 2015; Zen et al. 2010]. None of these databases and associated methods addresses reconstructing full body skeletal motion of individuals during unconstrained interactions—the main goal of this thesis.

2.1.3 Markerless Motion Capture Using Multiple View Systems

In computer vision, there has been a large number of approaches to measure 3D structure and motion of dynamically moving people using multiple camera sensors. Kanade et al. [1997] pioneered the use of multi-view sensing systems to "virtualize" reality, using 51 cameras mounted on a geodesic dome of 5 meters in diameter. A number of systems were subsequently proposed to produce realtime virtualizations [Matusik et al. 2000; Matsuyama and Takai 2002; Gross et al. 2003; Petit et al. 2009]. Vlasic et al. [2009] recovered detail by applying multi-view photometric stereo constraints using a system with 1200 lights on a dome and eight cameras. More recently, a multimodal multi-view stereo system fusing 53 RGB cameras and 53 infrared cameras has been proposed to reconstruct high quality 3D virtual characters [Collet et al. 2015]. A smaller, realtime version of a similar system was presented by Dou et al. [2016].

Other methods explicitly tackle markerless motion capture by producing 3D skeletal structures over time, similarly to their marker-based counterparts [Gall et al. 2009; Gavrila and Davis 1996; Cheung et al. 2005; Plankers and Fua 2003; Bregler et al. 2004; Kehl and Gool 2006; Corazza et al. 2010; Vlasic et al. 2008; Brox et al. 2010; Stoll et al. 2011; de Aguiar et al. 2008a; Furukawa and Ponce 2008]. The methods deform pre-defined articulated templates of fixed topology to recover the details that were subsampled or occluded in the set of views at a time instant. These methods require an offline method to generate a rigged 3D model for each individual, and the quality of the template is important to achieve high accuracy. The template models need to be aligned at the initial frame to be tracked, and usually a predefined pose (such as a T-pose) are assumed and performed by all individuals. The methods in this area fundamentally suffer from topological changes restricted by the template model, and, similar to other tracking methods, error accumulation is a critical issue in tracking for long durations. Although the 3D template-based method shows good performance—and has become a standard in markerless motion capture approaches—the requirement of a high quality 3D template for each individual limits the practicality of the method, especially in our scenario where dozens of individuals are involved, as the method does not scale well to multiple people. Previous work is demonstrated on a single actor with few exceptions [Ye et al. 2012; Liu et al. 2013b]. For example, it is required to segment image cues per subject to track them independently as in [Liu et al. 2013b], which becomes more complicated if a large number of people are involved, as in our scenes. It should be noted that none of the previous markerless motion capture approaches focus on capturing non-verbal social behaviors of naturally interacting multiple people, where robustness to inter-occlusions is paramount.

Rather than relying on tracking, Over the last few years, single view 2D pose estimation methods have shown great advances using large scale human pose datasets [Tompson et al. 2014b; Ramakrishna et al. 2014; Andriluka et al. 2014; Pishchulin et al. 2015; Wei et al. 2016]. State-of-the-art methods [Wei et al. 2016; Newell et al. 2016; Cao et al. 2016; Insafutdinov et al. 2016] show an excellent performance in various environments with varying subject's shape, appearance, and scales. Recently, a few methods use pose detection in multiple views to reconstruct 3D body poses [Burenius et al. 2013; Amin et al. 2013; Belagiannis et al. 2014; Elhayek et al. 2015, 2016; Rhodin et al. 2016]. To infer 3D skeletal parameters from 2D pose detection cues, unary and pairwise terms are defined based on the pre-training data of joint length, relative joint angles, and body colors. The methods are performed at each time independently in fewer camera settings, and thus they typically suffer from motion jitter. Although the results show potential in general environment settings (e.g., outdoors), the methods in this category do not yet reach similar quality compared to the 3D template-based approaches. Our approach alleviates this firstly by using a very large number of cameras (480) across which to integrate the signal provided by 2D pose detection. The large number of views makes this robust to inter-occlusions, and reduces the variance of the measurement by achieving consensus across such large a number of views. Additionally, we use multiple

RGB-D sensors simultaneously to improve the final estimation of body pose parameters. More importantly though, we extend traditional body pose detectors by adding fine-grained detection of joints on the hand and keypoints on the face. These detectors allow us to measure motion details that are not considered by any of the previous approaches.

2.1.4 Fine-grained Keypoint Detection

As mentioned in the previous section, the past decade has seen great advances in body pose estimation, both for depth sensor data [Shotton et al. 2013; Baak et al. 2011] as well as for RGB images [Tompson et al. 2014b; Wei et al. 2016; Newell et al. 2016; Insafutdinov et al. 2016]. Similarly, facial keypoint detection has seen such a rapid development (e.g., [Xiong and De La Torre 2013; Cao et al. 2014]) that it is now a standard component in commodity handheld devices. Combined with a 3D model of the face, even realtime 3D facial performance capture is possible (e.g., [Weise et al. 2011; Ichim et al. 2015; Thies et al. 2016]).

However, while depth-based hand pose estimation methods have advanced remarkably in recent years—with consumer hardware already available—there is comparatively very little work on estimating hand joints in RGB images. The major obstacle is the lack of large datasets of annotated images (for body pose estimation such datasets exist, e.g., [Andriluka et al. 2014; Lin et al. 2014; Ionescu et al. 2014]). While recent papers have investigated creating such datasets for depth data (e.g., [Oberweger et al. 2016; Tang et al. 2013; Supancic et al. 2015; Tompson et al. 2014a]), there is no equivalent for RGB images.

Early work in hand pose estimation originally focused on RGB data, with Rehg and Kanade [Rehg and Kanade 1994; Rosales et al. 2001; Athitsos and Sclaroff 2003] exploring vision-based Human-Computer Interaction (HCI) applications. Most popular were methods based on fitting complex 3D models with strong priors, including e.g., physics or dynamics [Lu et al. 2003], particle filters considering multiple hypotheses [Stenger et al. 2006], and analysis-by-synthesis [de La Gorce et al. 2011]. These methods use cues such as silhouettes, edges, skin color, and shading, and were demonstrated in very controlled environments, with restricted poses and slow, simple motions. The method of Wang and Popović [Wang and Popović 2009] lifted some of these restrictions, but required a specialized colored glove. A number of other capture setups exist that make use of specialized gloves or motion capture markers [Jörg et al. 2012], but these require instrumenting the participants. In contrast, our target application is unconstrained "in-the-wild" images of people in everyday activities.

Multiview RGB methods are often similarly based on fitting sophisticated mesh models (e.g., [Ballan et al. 2012; Sridhar et al. 2013]) and show excellent accuracy, but under very controlled conditions. These methods are based on tracking and rely on temporal coherence, often requiring an initial alignment and struggling to overcome cases where the model prior doesn't hold, such as different hand sizes, appearance, and interactions with unknown objects. In contrast, our approach to using multiview images is entirely mesh-free and model-free, and is performed on each frame independently.

With the introduction of commodity depth sensors, single-view depth-based hand pose estimation became the major focus of research. A large number of depth-based methods exist, broadly classifiable into generative methods [Oikonomidis et al. 2012], discriminative methods [Tang et al. 2014; Tompson et al. 2014a; Keskin et al. 2012; Xu and Cheng 2013; Sun et al. 2015; Wan et al. 2016], or hybrid methods [Sridhar et al. 2013; Sharp et al. 2015; Sridhar et al. 2015; Tzionas et al. 2016; Ye et al. 2016]. Generative methods typically rely on rendering a depth-map to be compared with the measured depth, and are sensitive to initialization. Recently, the hybrid method of Sharp et al. [2015] demonstrated practical performance across a larger range, but there are still difficult cases such as hand-hand interactions and hand-object interactions. Sridhar et al. [2016] and Tzionas et al. [2016] have made progress in coping with hand-object and hand-hand interactions with hybrid methods as well, but still within a tracking framework. Most depth-based methods struggle to capture room-sized scenes (and therefore, do not work for multiple people) because the resolution of depth sensors is typically much lower than camera resolutions. Therefore, our work focuses on single-image detection in RGB images, and does not rely on tracking for robustness.

Discriminative methods predict hand pose by directly mapping observed sensor data to hand configuration space [Tang et al. 2014; Tompson et al. 2014a; Keskin et al. 2012; Xu and Cheng 2013; Sun et al. 2015; Wan et al. 2016]. To train the mapping, a large scale dataset is required, and, due to the fact that annotation on real images is costly, generating the labeled data is one of the major challenges.

Recent approaches based on depth sensors rely heavily on synthetic data. Oberwerger et al.'s use of feedback loops to generate synthetic training data for hand pose estimation [Oberweger et al. 2015] is motivated by the same principles as our work, but focuses on generating depth images. In contrast, our work does not use a renderer as part of the training loop—we instead label real images. The semi-automatic data annotation scheme presented in [Oberweger et al. 2016] is also similar in motivation, however, our approach uses multiview geometry and keypoint detection to provide automated supervision.

Our keypoint detector is similar to that of Tompson et al. [2014a], who propose using confidence maps to represent keypoint locations, and also use a Convolutional Neural Network (CNN) architecture to estimate the confidence maps. However, our method takes as input RGB images rather than depth, and we use a network architecture based on Wei et al.'s Convolutional Pose Machines [Wei et al. 2016], originally proposed for body joint detection.

The combination of this RGB-based 2D keypoint detector with our multiview camera system allows us to recover 3D hand pose for multiple interacting people, and even for complex hand-object interactions.

Finally, it is worth underlining that recovering hand pose in a large capture area remains very difficult even with traditional, marker-based motion capture. Not only do the markers interfere with hand motions, but due to the severe self-occlusions of the hand, the motion capture data has to undergo very extensive cleanup. Because of this, several methods that aim to generate plausible hand motion (given e.g., only body motion) have been developed [Majkowska et al. 2006; Jörg et al. 2012]. We believe our system is one of the first that allows detailed hand-pose capture of multiple subjects in a room-sized scene while being robust to hand-object interactions, such as playing instruments or handling tools. Similarly, we are the first to demonstrate combined body, face, and hand capture for multiple interacting people in a markerless, un-instrumented scenario.

2.2 Modeling Time-Varying 3D Point Data

After obtaining measurements of sparse joint locations for the human body, this thesis relies on statistical models to infer complete body descriptions, and more importantly, temporally consistent motion. In this section, we review work in the context of modeling 3D motion data for the task of inferring or reconstructing a complete set of 3D point trajectories from partial or incomplete observations. Representing time-varying 3D point data, particularly for motion of the body and face, is a well-studied area in computer graphics and vision. An extensively researched subfield is the literature on tracking body motion (especially in computer vision) and synthesizing body motion (especially in computer graphics); a review of these two subjects can be found in [Forsyth et al. 2005]. An overview of representations and analysis techniques for more general deforming 3D shapes can be found in [Bronstein et al. 2008]. The focus of the review presented here is particularly on linear models and spatio-temporal extensions that combine temporal models with shape models.

Partial observations occur in problems with missing data, in the 3D reconstruction from 2D images, and when inferring a complex model from a limited or noisy set of measurements. These applications typically result in under-constrained, ambiguous, or ill-posed problems with more than one possible solution. There are largely two approaches: physically-based approaches, where ill-posed systems are conditioned according to a physically-grounded model, and statistically-based methods, where expected statistical properties of the data are used to regularize the ill-posed system without explicitly appealing to any physical grounding. For our work, because we are interested in modeling social and nonverbal behaviors which do not follow strict physical rules, the most relevant methods are those that capture statistical or probabilistic relations rather than fixed rules.

Regardless of whether the model or prior for the data is physically-based or statistically-based, the underlying representation for the motion data is typically of one of two kinds: (1) the Euclidean coordinates of a set of 3D points, or (2) the configuration of joint angles of an articulated representation of the human body. In either case, the statistical methods used to model the data are largely the same. In both approaches, a set of spatial configuration dimensions is recorded at every time instant, and it is this set of spatial configurations that is modeled—either the XYZ coordinates of points, or the angles for a set of joints. Generically, we can refer to this set of modeled dimensions as the configuration space of the object.

2.2.1 Physically Based Methods

In the context of reconstructing 3D deforming objects, the earliest physically-based representation was by Terzopoulos et al. [1988]; subsequent work [Metaxas and Terzopoulos 1993] presented a physically-based approach using nonlinear filtering over a superquadratic representation. Concurrently, Pentland and Horowitz [1993] presented an approach where a finite element model described deformations in terms of a small number of free vibration modes, equivalent to a Kalman filter accounting for dynamics. Taylor et al. [2010] revisited the idea of using rigidity but at a local scale using a minimal configuration orthographic reconstruction. Salzmann and Urtasun [2011] described a number of physically-based constraints on trajectories of points that could be applied via convex priors.

In graphics, physical spatiotemporal models have also been used for the tasks of motion editing and motion adaptation. Here, the partial observations correspond to a few user inputs or user constraints, from which a complete motion sequence must be inferred. Methods related to spacetime constraints [Witkin and Kass 1988] aim to globally modify a character's motion to meet certain requirements; these methods commonly aim to produce physically-realistic motions by minimizing an energy function. Most approaches focus on carefully formulating the optimization function to enforce the characteristics of spatiotemporal data, and not on the representation itself. Typically, the representation is based on keyframe interpolation of joint angles (for articulated characters) or rig parameters (for facial animation). For body motion in particular, a large number of approaches use a combination of per-frame inverse kinematics and temporal filtering; see for example the methods of Gleicher [2001],[Gleicher 1997], who offers an extensive review of this method and related techniques.

2.2.2 Statistically Based Methods

Because 3D motion data is high-dimensional, the most prevalent modeling approach is often the application of a dimensionality reduction technique. The specifics of the method may change depending on which dimensions of the data are modeled, but the core procedure is fairly standard: Principal Component Analysis (PCA) or a similar technique is applied to a training set to find the most significant modes of deformation, and the data samples are then described as a linear combination of these modes in terms of a set of coefficients. The reduced set of coefficients is then used as the representation to be analyzed or modeled.

This general approach has subsequently been extended to model nonlinear relationships between the 3D points by using kernel methods, for example, using Kernel PCA [Schölkopf et al. 1997] and Gaussian Process Latent Variable Models (GPLVMs) [Lawrence 2004]. Nonlinear methods often model the data more accurately, but they carry a significant cost in memory and computation that in many situations makes the linear model preferable because there are simple and efficient algorithms available for model fitting, reconstruction, and model estimation.

2.2.2.1 Spatial Models

For spatial data, the linear model is commonly called a point distribution model and was established through the work of Mardia and Dryden [1989], Le and Kendall [1993], and Cootes and colleagues [1995a]. Investigation into factoring the spatial component for inference of partial observations began with Tomasi and Kanade's rank 3 theorem [Tomasi and Kanade 1992], which established that image measurements of a rigidly rotating 3D object lay in a three dimensional subspace. The associated factorization algorithm was extended by Bregler et al. [2000] for nonrigid objects, positing that a shape space spanned the set of possible shapes. Unlike the rigid case, where the bilinear form could be solved using singular value decomposition (SVD), this formulation had a trilinear form. Bregler et al. proposed a nested SVD routine, which proved to be sensitive to initialization and missing data. A series of subsequent papers investigated various constraints to better constrain the solution or relax the optimization (a sample of major work includes [Brand 2005; Yan and Pollefeys 2005; Vidal and Abretske 2006; Fayad et al. 2009; Russell et al. 2011]). Recently, Dai et al. [2012] presented a method that uses a trace-norm minimization to enforce a low rank shape space, and Garg et al. [2013] showed that the method can be applied to recover dense, non-rigid structure. Lee et al. [2013a] formulated a normal distribution over shapes in a Procrustes aligned space.

2.2.2.2 Temporal Models

For temporal data, dimensionality reduction has also been applied to learn a compact linear basis of trajectories [Sidenbladh et al. 2000b; Torresani and Bregler 2002; Akhter et al. 2008a]. For reconstructing 3D dynamic point clouds from partial 2D observations, trajectory space representations were proposed by Sidenbladh et al. [2000b], which they referred to as *eigenmotions*, and Zelnik-Manor explored factorizations in the temporal domain [Zelnik-Manor and Irani 2004]. Akhter et al. [2008b] noted that, in trajectory space, a predefined basis could be used, which reduced the trilinear form to a bilinear form and allowed the use of SVD once again to recover the nonrigid structure. Unfortunately, the solution was shown to be sensitive to missing data and cases where the camera motion is smooth [Park et al. 2010]. Park et al. [2010] used static background structure to estimate camera motion, reducing the optimization into a linear system, and were able to handle missing data. Valmadre and Lucey [2012] presented various priors on trajectories in terms of convex priors on 3D point differentials, showing better noise performance than truncation of the trajectory basis.

2.2.2.3 Joint Spatial and Temporal Models

A number of approaches have combined spatial and temporal constraints for inference from partial observations [Pentland and Horowitz 1993; Metaxas and Terzopoulos 1993; Olsen and A. Bartoli 2008; Torresani et al. 2008; Gotardo and Martinez 2011]. Linear models that jointly span both space and time have been used to track shapes deforming over time and to describe their principal modes of spatiotemporal variation [Hamarneh and Gustavsson 2004], for registration in both space and time [Perperidis et al. 2004], for spatiotemporal segmentation [Mitchell et al. 2002], for action recognition [Zelnik-Manor and Irani 2006], for motion synthesis [Urtasun et al. 2004; Min et al. 2009], and for denoising [Lou and Chai 2010]. Typically, joint spatiotemporal models are a direct application of linear dimensionality reduction where each spatiotemporal sequence is vectorized and represents one sample. Torresani et al. [2008] presented a probabilistic model using probabilistic PCA combined within a linear dynamical system. Subsequently, the shape space and trajectory space approach were combined by Gotardo and Martinez [2011], and Lee et al. [2014] embedded the Procrustean normal distribution within a temporal Markov process.

These models are often specific to the particular sequence length that was chosen during training, and very specific correlations between points at different space-time locations are consequently learned. These correlations are most prominent in periodic motions. To generalize beyond specific spatiotemporal correlations, joint linear spatiotemporal models therefore typically require a large training set. In contrast, the statistical model we present jointly models space and time as a linear combination but relies on a bilinear factorization of the spatial and temporal variations that substantially reduces the amount of required training data and simplifies computations.

2.2.2.4 Dynamical Models

Time-varying spatial data has also been modeled as a dynamical system, where a fixed rule describes transitions across time [Thrun et al. 2006]. Compared to basis representations, dynamical systems model the evolution of a process as transitions between time-steps, making them especially attractive for online processing. Conversely, because it is a model of the process rather than a direct model of the data, operations affecting the entire sequence are usually more costly. Autoregressive models, which assume that motion of a point over time will be highly correlated, have been used for compression of motion capture data, as demonstrated by Arikan [2006]. Li and colleagues [2009] model marker trajectories as a Linear Dynamical System (LDS) to infer missing markers. Nonlinear dynamical systems have also been successfully applied to motion data, most notably Gaussian Process Dynamical Models (GPDMs) [Wang et al. 2008], which have been shown to be an excellent model for synthesis and inference. The main drawback of Gaussian process models is significant computational and memory cost, making them impractical for very large datasets. Inference is usually iterative in the case of missing data. Model estimation is costly as well, and typically accomplished using a nonlinear optimization (expectation-maximization) that requires adequate initialization. To alleviate these restrictions, recent work has focused on black-box, learning-based methods. This approach has been particularly successful when using deep architectures to model complex dynamics [Taylor et al. 2007; Fragkiadaki et al. 2015; Holden et al. 2015, 2016].

2.3 Overview of Contributions

In contrast to previous markerless motion capture methods, the system we present in Chapter 3 is explicitly designed to reconstruct the social signals exhibited during group interactions, combining a multiview configuration of cameras with multiple RGB-D sensors. We present an approach to temporally align the data from the unsynchronized RGB-D sensors, and a calibration approach that explicitly models the temporal offsets between sensors. The system allows a fully 3D dynamic reconstruction of the entire capture room, ensuring coverage even in areas that are occluded in any one view.

In Chapter 4, our core contribution is using 3D triangulation as supervision to generate additional training data for 2D detectors. We use this method to train keypoint detectors for the joints of the hand, which allows us to reconstruct hand pose in the Panoptic Studio. Because work to date in hand pose estimation has focused mainly on using depth data, there is currently no dataset for hand joint detection in general purpose images. We also introduce such a dataset and associated performance metrics, evaluating our hand keypoint detections on web images. Additionally, we demonstrate markerless motion capture of 3D hands in unprecedented scenarios, including tool manipulation, musical performances, and social interactions between multiple people.

The statistical model for spatiotemporal data that we present in Chapter 5 is a joint spatial and temporal model. A similar spatiotemporal factorization approach for non-rigid 3D data first appeared in [Gotardo and Martinez 2011; Akhter et al. 2012], which proposed an estimation procedure with similar (but truncated) bilinear spatial and temporal factors. In contrast, we develop a fully probabilistic data model and show the dependency assumptions that this model implies. Further, we show that the Kronecker covariance pattern predicted by the model can be found empirically in real data, and that the marginal distributions of the model correspond to spatial and temporal models of prior art. We additionally present a novel inference procedure for reconstructing partially observed 3D data with the Kronecker MRF model. We show that, when the spatial covariance is unknown, we can estimate both the unknown spatial covariance and the 3D data convexly in terms of a trace-norm optimization. We also relate this optimization to prior work using the trace-norm in non-rigid structure from motion [Dai et al. 2012].

In Chapter 6 we present a markerless motion capture method that simultaneously reconstructs parameterizations of the body, face, and hands for multiple interacting people. Our method combines the output of keypoint detectors with depth measurements from the Panoptic Studio, using the Kronecker MRF model as a spatiotemporal prior. This allows us to measure, for the first time, the relationship between the motions of each of these parts.

Chapter 3.

Social Interaction Capture

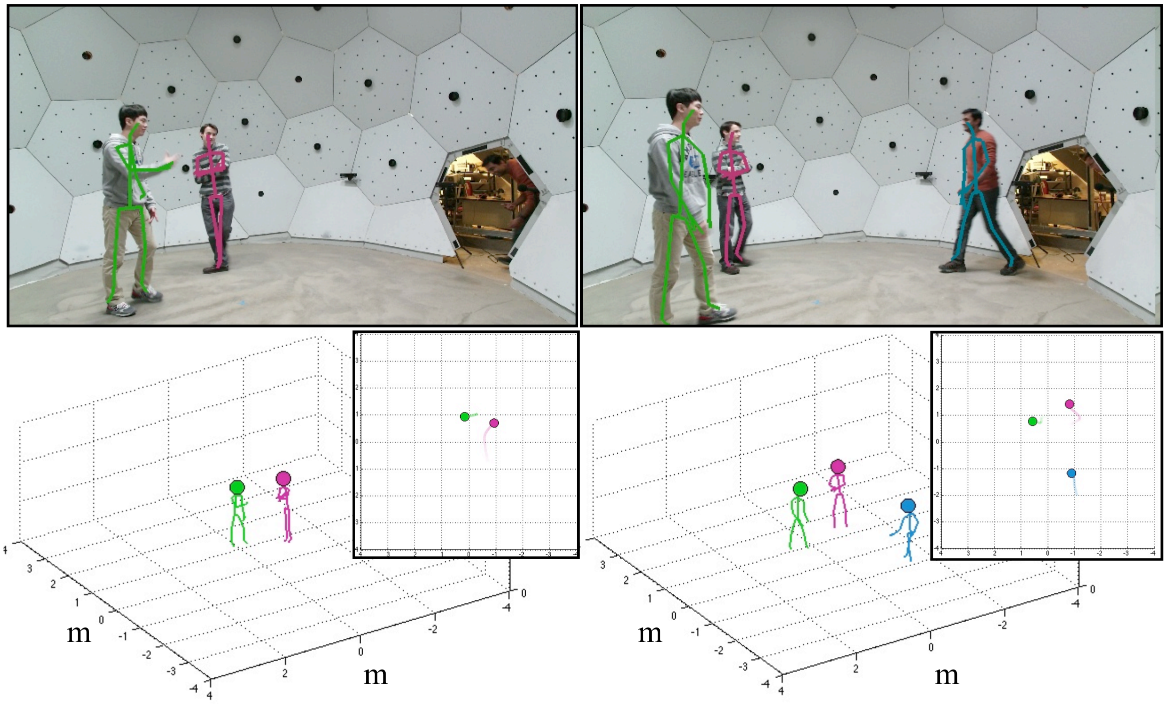


Figure 3.1: We collect group interactions with multiple RGB-D sensors to obtain room-sized 3D reconstructions. From these, we infer body pose, gaze direction, and other markers of social signaling. Top, two sensor views with overlaid pose detections. Bottom, pose reconstruction in three-quarters and overhead views.

3.1 Introduction

There currently exist no databases of social group interactions that are annotated with full 3D body pose. In this section, we propose an experimental protocol for the capture of such a dataset and outline the technical requirements. We aim to build computational models of human body motion in social situations, such as that of the scene depicted in Figure 3.1. To this end, we plan to record a dataset of social interactions using sufficient cameras (including depth sensors) to allow full 3D body and face motion reconstruction. Because the breadth of human social interactions is vast, we will focus on a restricted set of scenarios designed to naturally elicit nonverbal social signals from the participants in a repeatable fashion.

3.1.1 Motivation

We are interested in analyzing 3D movements corresponding to nonverbal signals that occur during natural interactions between groups of people. Databases currently in existence (see Sect. 2.1.2) do not satisfy two key requirements: (1) we want to accurately recover full 3D body motion, and (2), we want group interactions where nonverbal signals and their corresponding motions occur naturally.

Marker-based motion capture is the method of choice when it comes to recording 3D body motion accurately, but it requires a lengthy setup and a specialized suit that encumbers movement. Additionally, wearing the suit places people in an atypical circumstance that makes them especially self-conscious about their body and movements, making it more difficult to capture natural interactions.

While a large number of less intrusive, video-based databases of group interactions exist, most are activity centric (e.g., sports, action recognition datasets, see Sect. 2.1.2 for a review) and do not feature the more natural movements of daily life. Many video datasets do capture nonverbal behavior—including TV series, movies, and a large number of behavioral studies involving dyadic and triadic interactions—-but we cannot accurately measure 3D body movements from these with current machine vision algorithms.

We argue that a multi-camera setup can yield motion to sufficient accuracy while calling far less attention to the capture system itself, and without encumbering the motion of the participants. An example of such a capture is shown in Figure. 3.1, and the system design is discussed in Sect. 3.3. This system yields sufficient information to capture body motion during social group interactions—crucially, multiple views to avoid problems of occlusion between participants—and we expect that it will allow answers with statistical certainty to questions of nonverbal behavior. In the remainder, we also refer to the capture room and system simply as “the Studio”.

3.1.2 Research Goal

It is important to underline that it is not our intention to analyze the overt content of the interaction but rather the communication of social signals that occur during the interaction. In creating this dataset, our main goal is to enable the analysis human social behaviors using measured social signals to facilitate social behavior understanding in a data-driven manner. We are especially interested in finding motions that occur commonly across people—and have a common effect when others observe them. For this reason, our data collection stresses capturing as many different subjects as possible, rather than repeating the capture many times with the same individual.

3.2 Collecting a Dataset of Social Interactions

To evoke natural interactions, we involved participants in various games: “Ultimatum”, “Haggling”, “Mafia”, and “007-Bang”. The first two games are used in experimental economics and psychology to study conflict and cooperation, and the latter two games also induce a variety of rich non-verbal signals in participants. We describe the structure of the first two games in detail.

3.2.1 Operationalizing the Capture of Natural Social Interactions

We are limited by technical as well as practical issues in the amount of data we can capture and process. Because of this, we have subjects playing short games which tend to engage participants in lively discussions that naturally feature nonverbal behavior prominently. In fact, we would like to minimize the occurrence of motions with a functional objective, to make the study of nonverbal motions more straightforward. We focus on capturing at least 3-way interactions to be able to capture social signals beyond the dyadic:

1. **Ultimatum.** This game is played in two teams. One team (team “A”) starts out with a certain amount of money. Team A decides on how to split the money between teams A and B, and then makes an offer of this amount to team B. If team B accepts the offer, each team receives the amount specified by the split. If team B rejects the offer (typically because it is unfair), then neither team receives any money. Team B must come to a decision within 1 minute and the teams are allowed to discuss until then. It is the nonverbal behavior during this discussion that we aim to analyze.

2. **Haggling.** This game is played by at least three people. Two sellers try to sell an item of equal price but different characteristics (e.g., an apple vs. a piece of chocolate) to a buyer, who is provided with sufficient money to buy 1 of the items. Within 1 minute, the sellers try to convince the buyer to purchase their item. The reward is straightforward: the seller who “wins” gets the money, the buyer gets the item they purchased.

We outline the following logistical details of the study:

- **Participant Requirements**: The participant must be over 18 years old. We did not place any other restrictions on participants, but to date, all participants in our social games have been healthy university students under 35 years with no motor impairments.

The distribution of men to women reflects the local student population, with men outnumbering women at a ratio of about 2.5 to 1.

- **Expected Duration**: The expected total duration of each capture is 1 minutes. Each participant may play the game more than once.
- **Compensation and Cost**: We compensate participants depending on game outcomes (more details below).
- **Privacy and use of data**: The recorded data will be shared with the research community when consent is given by the participants. We will not gather or store any information about subjects besides the audio and video information recorded during the capture and the game outcomes. We will keep a confidential record of who has participated and what compensation they received for bookkeeping.
- **Risks**: Minimal risk. The risks and discomfort associated with participation in the study are no greater than those ordinarily encountered in daily life.
- **Number of sequences**: We aim to capture more than 60 sequences. To date we have captured 20 "Haggling" sequences and 25 "Ultimatum" sequences.

3.2.2 Capture Protocol

The protocol to be followed by the experimenters when running subjects is detailed below for the two types of social games. As a first step, we record each participant's name and take a picture for record-keeping. We additionally assign each participant a subject number.

Ultimatum (10 minutes)

1. There is one experimenter who directs participants.
2. The participants are first provided with the rules of the game as a group, and presented with the roles of the acceptor and proposer.
3. They are individually tested to see if they understand the game and are corrected by the experimenter.
4. The three participants are grouped into two teams (on Proposer and one Acceptor) by random assignment. No team combination that is exactly the same will be repeated.
5. The teams enter the Studio and are allowed 1 minute to negotiate.
6. When 1 minutes elapse, the Proposer makes a final proposal.
7. Team Acceptor accepts or rejects the proposal.

8. Both teams exit the Studio.

Haggling (10 minutes)

1. There is one experimenter who directs participants.
2. The three participants are first provided with the rules of the game as a group, and presented with the roles of the acceptor and proposer.
3. The three participants will be grouped into a buyer and two sellers by random assignment, and given random objects to sell.
4. The Buyer enters the studio. The Sellers enter the Studio.
5. Free form discussion is allowed for 1 minute inside the Studio.
6. When 1 minutes elapses, the Buyer is forced to decide.
7. All participants exit the Studio.

For each game, we record the outcome immediately after the participants exit the Studio. We compensate players of the Ultimatum game with the split amount that was decided (the participants always split $10 dollars), or nothing if no agreement was reached. During the Haggling game, the team with a successful sale receives 2 dollars compensation. Additionally, we reimburse all participants with a minimum of $10 if they did not make at least that much while playing the games. On average, participants earned about $15 dollars.

3.2.2.1 Compliance

The data is captured as part of the study "*HS14-532: Visual Capture of Polyadic Human Interactions in the Panoptic Studio*", with IRB approval under the direction of the P.I. Yaser Sheikh and with co-investigators Hanbyul Joo and Minh Vo.

3.2.3 Discussion

Example frames from captured sequences can be seen in Fig. 3.2. More complete examples can be downloaded from the database website, http://domedb.perception.cs.cmu.edu.

We made a few important observations when reviewing the data. The foremost (but unsurprising) conclusion is that lab members make for poor test subjects, resulting, in some cases, in the capture of contrived motions. Familiarity with the capture system led them to either consciously or unconsciously modify their motions (presumably to make them easier to capture). For this reason, we explicitly flag sequences with lab members in our website.

Figure 3.2: Frames from several sequences acquired using lab members. The sixth vignette (bottom right in each panel) shows the 3D reconstruction. Four sequences are "Ultimatum" sequences, and only one sequence below is a "Haggling" sequence. Can you tell which one?

We observed that playing the game repeatedly produced observable differences in the actions of the people, with two main effects. First, after having played each game once, there was usually a slight increase in the overall engagement of the participant and the liveliness of his actions and dialogue. After a few games, however, the effect was reversed and the participants became less lively—particularly during the Ultimatum game (which, after all, when played within a group of 5 people, does not offer much variation).

Second, the most noticeable factor affecting the motions of the participants during the interactions was that of personality. There were large differences between how each subject used nonverbal communication—the way they move, the liveliness of their discussions—and these differences also had an effect on how the group behaved. In particular, each group of people that took part in the study showed very different characteristic behaviors—e.g., some groups had a notable preference for equal splits in the "Ultimatum" game, while others took bigger risks (and did not reach an agreement). It is clear that the data we have captured to date is not sufficient to draw conclusions about social signals across the general population; however, the data collected has allowed us to advance and improve our capture system.

3.3 Capture System: The Panoptic Studio

In this section, we detail the design of the capture system and we describe the methods by which we obtain per-frame 3D pose and other 3D geometry for the specific purpose of capturing group interactions. Given the experimental design described in the previous section, our capture system must fulfill specific requirements:

1. Recover accurate 3D body pose and positions.
2. Be unobtrusive: people should be able to interact naturally.
3. Accommodate groups of people (from 3 to 6).

We do not use retro-reflective markers or specialized motion capture suits because they hinder natural motions—for example, arm crossing is impeded by the markers on the arms, and certain motions tend to cause dropout of other markers on the body. Instrumenting the participants in this way also makes it difficult to achieve *natural* interactions: the suit makes people self-conscious about their movements and continuously calls attention to itself—both on a participant wearing the suit, as well as on someone interacting with someone wearing the suit.

To make the capture as unobtrusive as possible, we therefore prefer to instrument the room rather than the participants. The obvious disadvantage is that recovering 3D pose becomes a much more difficult task: we will have to infer it from video where participants can be wearing arbitrary clothing. This is sometimes referred to as markerless motion capture or human pose estimation—essentially, a problem of corresponding image pixels to body limbs. Recent advances in human pose estimation from images [Wei et al. 2016; Cao et al. 2016; Insafutdinov et al. 2016] have made this problem significantly simpler than it was just a few years ago: essentially, triangulating 2D pose detections across multiple views yields a very good estimate of 3D human pose. The main difficulty are inter-occlusions between interacting people, a regime that remains problematic for 2D pose estimation methods and complicates reconstruction considerably.

Our solution is to capture the scene from as many vantage points as possible. For our purposes, this turns out to be an absolute necessity: people habitually occlude each other and can face sideways or away from any particular camera. Because inter-occlusions are a natural characteristic of group interaction—i.e., people naturally face each other to talk—we do not attempt to limit inter-occlusions by asking people to situate themselves in any particular way. We instead reconstruct the entire room in 3D by merging all camera views, so that we have scene coverage almost everywhere—even in the presence of strong occlusions from the view point of any single camera view.

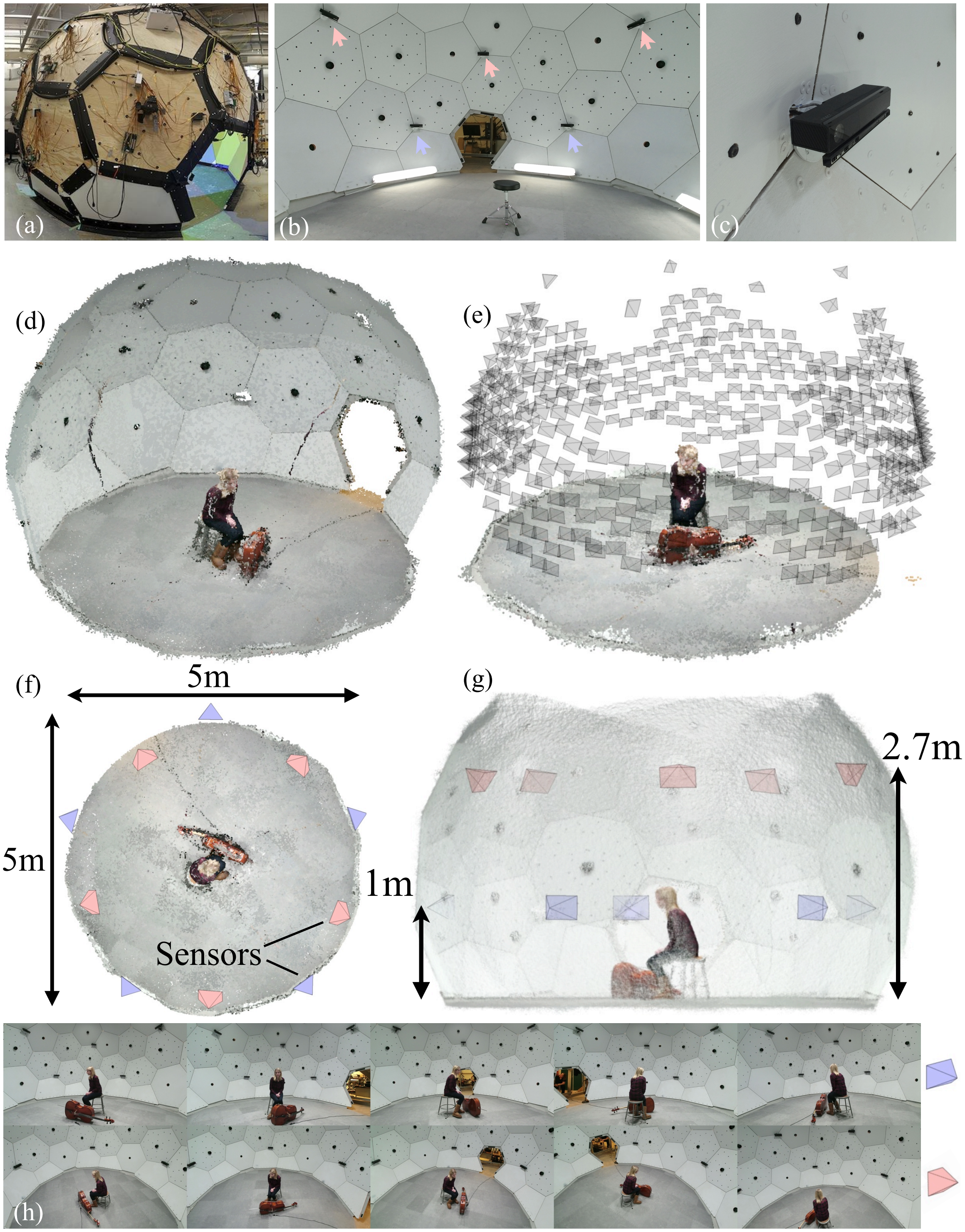
It is worth mentioning that an attractive alternative to a massively multiview system is to use egocentric video. This provides us with the most important vantage points for the interaction: namely, what each participant sees [Fathi et al. 2012; Park et al. 2013; Alletto et al. 2014; Park and Shi 2015]. However, this option requires instrumenting participants with a camera. These are still bulky enough to be obtrusive, though this will become a more and more attractive option as the technology progresses (particularly, for measuring social interactions in the wild).

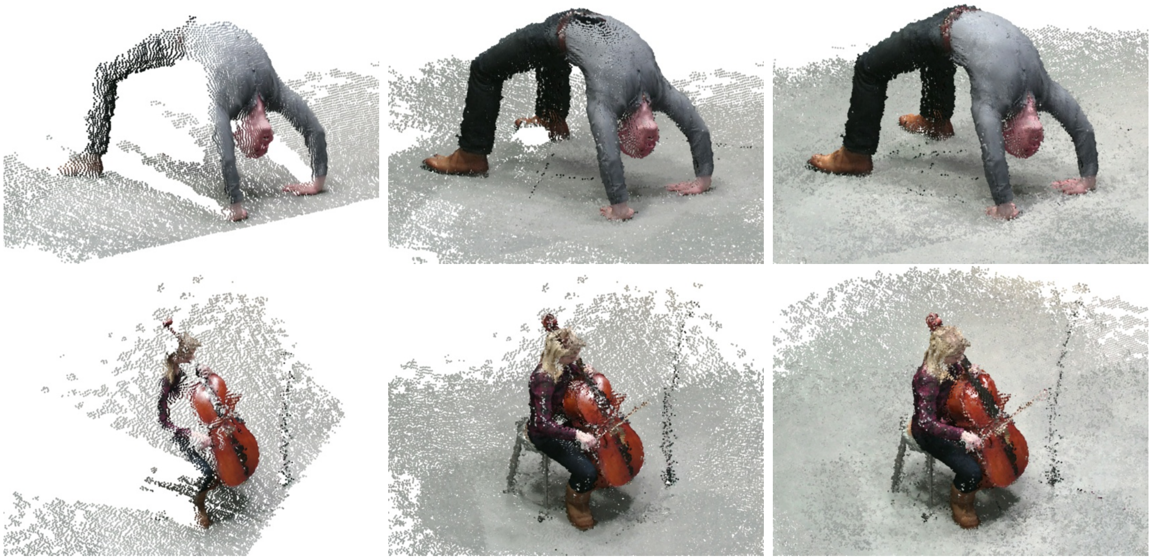
3.3.1 The “Kinoptic” Studio

The Panoptic Studio [Joo et al. 2014, 2015, 2016] is a 3D capture room designed for studies on the multi-view geometry of dynamic scenes. It was originally built with 480 ED VGA and 20 HD video cameras, which are housed in a dome-like structure that can be seen in Figure 3.3. For the methods developed as part of this thesis, we added 10 RGB-D Kinect v2 sensors to facilitate 3D reconstruction and body pose estimation. We call this sub-system the “Kinoptic” Studio to differentiate it from the RGB sub-system.

Combining scans from multiple color and depth (RGB-D) sensors can yield 3D reconstructions of volumes far larger than the working area of a single sensor. This was shown by Levoy et al. [2000] using movable laser rangefinders, and, more recently, in realtime on a

Figure 3.3: Ten RGB-D Kinect v2 sensors are arranged around a room that is 5 m (16 ft) in diameter, within a spherical dome. An external view is shown in (a), and with one hemisphere removed for visualization in (d). The dome structure also houses 480 VGA and 20 HD video cameras, (e). The RGB-D cameras, shown as a closeup in (c), are arranged around the inner walls at two levels, at roughly 1m and 2.7m height (b, f, and g). The color cameras capture 1920×1080 video at 30Hz (h).





(a) One sensor. (b) Five sensors. (c) Ten sensors.

Figure 3.4: (Top) A person attempting a handstand. (Bottom) A cellist. From left to right, point clouds of the scenes captured with (a) one RGB-D sensor, (b) five RGB-D sensors, and (c) ten RGB-D sensors.

GPU with a handheld Kinect by Newcombe et al. [2011]. However, more importantly for our purposes, it allows us to observe the scene from several vantage points simultaneously.

In contrast to [Newcombe et al. 2011], where a moving Kinect is used to scan static scenes in detail, we are interested in capturing dynamic scenes. To this end, we array several RGB-D cameras at fixed locations surrounding the capture area, mounted concentrically around the dome (see Figure 3.3, middle row). This gives us full coverage of the entire room simultaneously. Figure 3.4 shows the increased coverage and resolution (in terms of measurements per unit volume) when using one, five, and ten RGB-D sensors respectively. For our purposes, however, the additional advantage of using multiple sensors is that this makes us more robust to inter-occlusions between people during group interactions.

Merging the information from multiple RGB-D sensors presents its own set of peculiar challenges, the most important of which are (1) data management, and (2) calibration, including synchronization or time alignment, and geometric registration between the sensors. We discuss each component only briefly here.

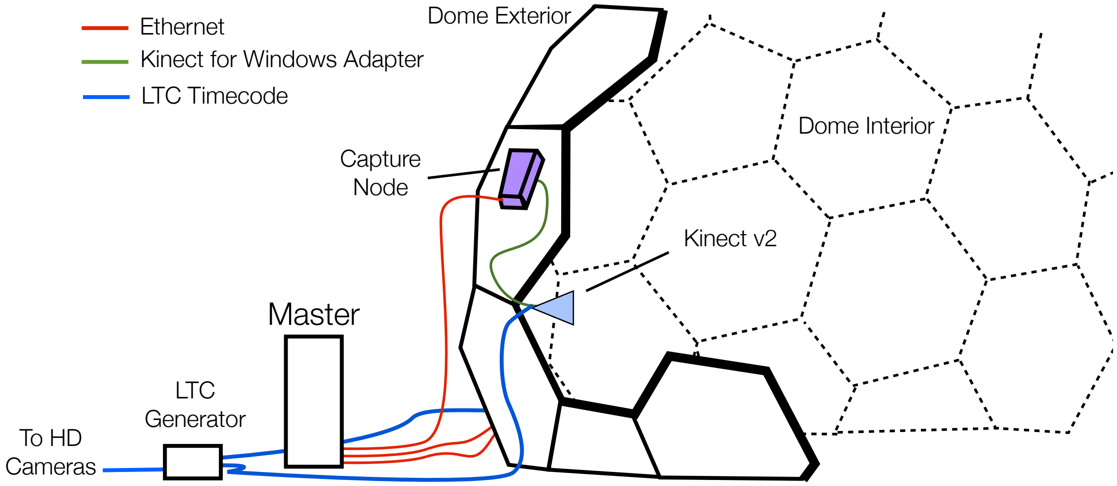


Figure 3.5: Schematic of the RGB-D subsystem of the Panoptic Studio. Each Kinect v2 sensor is mounted inside the dome and is connected to a "capture node", the computer responsible for recording all data from this sensor. The sensor is mounted on the structure's interior and the capture node is mounted on the exterior. Each capture node is networked with a central master that controls it. Additionally, an LTC timecode signal is fed into the audio channel of each sensor to provide a global clock reference. The same timecode is also used as a clock for the HD cameras.

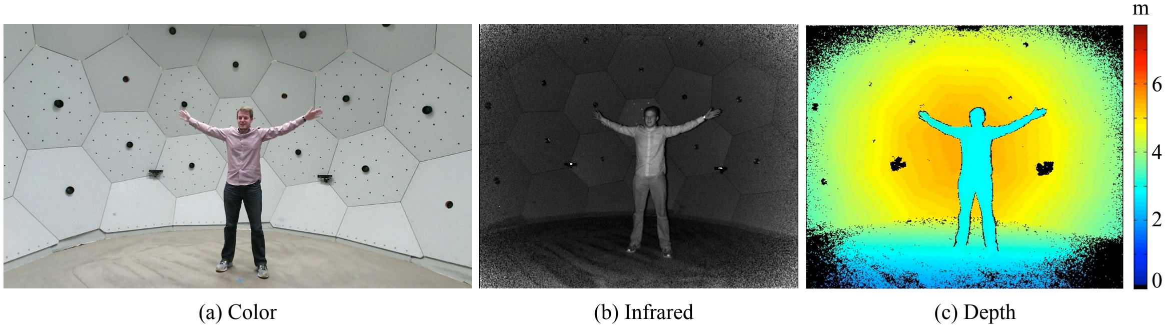
3.3.1.1 Data Management

Each Kinect v2 sensor is equipped with a 1920×1080 resolution color camera and a 512×424 resolution time-of-flight camera in the IR (infrared) spectrum, which uses a strong source of IR illumination. The time-of-flight camera produces a 16-bit per-pixel depth map and a 16-bit IR image simultaneously. All of the image streams capture at a maximum rate of 30 Hz. In addition, there is an audio stream and several inferred data streams (annotated body poses, per-pixel body segmentations, and annotated facial landmarks). Focusing on the measurements, we have[1]:

Stream	Resolution	Rate	Bandwidth
Color	1920×1080	30 Hz	120 MiB/s
Depth	512×424	30 Hz	13 MiB/s
IR	512×424	30 Hz	13 MiB/s
Audio	32-bit Mono	16 kHz	63 KiB/s

The raw color stream is the main bottleneck; frames are encoded in a YUYV422 pixel format, with 2 pixels encoded every 4 bytes. While the depth and IR streams are both 16-bit per pixel, the spatial resolution is substantially lower. We dedicate a computer (a "capture node") to each Kinect v2 sensor; each capture nodes stores its Kinect data streams onto a

[1] A mebibyte, or MiB, is $2^2$0 bytes, or 1024^2, and roughly equal to 10^6.



(a) Color

(b) Infrared

(c) Depth

Figure 3.6: (a) Color, (b) IR, and (c) depth images for a particular frame. In the depth image, black denotes no measurement; the colormap axis is in meters.

solid-state drive. These capture nodes are controlled by a networked master that triggers the capturing process and manages the devices. Figure 3.5 shows a schematic of how this is setup in the Panoptic Studio. Figure 3.6 shows an image of each of the data streams for a particular frame. Note that the infrared emitter from other sensors that are visible in the frame creates a saturated spot in the IR image, as well a spot of missing data in the depth image. Otherwise, sensor interference does not present a major problem with the chosen arrangement of the sensors.

3.3.2 Temporal and Geometric Calibration

Calibrating the multiple RGB-D Kinect sensors in the Panoptic Studio presents a unique set of challenges. The RGB cameras in the Studio are calibrated using traditional structure-from-motion algorithms, using a set of random black-and-white patterns that are projected inside the dome using 5 projectors. These patterns are then used to generate correspondences across the many camera views (see [Joo et al. 2016] for details). However, this calibration method does not work for our RGB-D sensors.

Each Kinect combines two sensors, a traditional camera (which can in fact be calibrated using the visible light projectors), and a depth sensor, which is essentially an infrared (IR) camera. Because the IR sensor is only sensitive to wavelengths in the infrared range, the visible-light patterns generated by our projectors are not visible in the IR images. This prevents us from using the same system to calibrate the depth sensors. Additionally, because the IR sensor's resolution is limited (512×424), we had difficulties creating correspondences that were visible and accurately localized in both the IR sensors and the traditional cameras (e.g., by placing objects inside the dome). When using spheres, for example, there is always a trade-off in the size of the sphere—larger spheres allow better detection and localization of the sphere center in the depth image, but make precise localization of the sphere center in the color images impossible.

In fact, our most successful tests generated these correspondences by using a traditional checkerboard pattern, printed on a large poster board with ink that is visible in both the IR and visible light spectra. Therefore, a traditional checkerboard calibration (where we place the pattern at fixed locations within the dome and take images) can be used to calibrate these sensors. However, this is logistically not practical. Due to the number of sensors, the limited overlap between RGB-D views, and the large area to calibrate, taking the required number of static shots of the checkerboard to achieve good coverage is prohibitively time consuming. Additionally, because we need to cover the entire working area, it is difficult to keep the checkerboard static in certain parts of the dome (particularly elevated positions).

We instead opt for a dynamic calibration, where we walk and wave the checkerboard in the interior of the dome and capture video, rather than rely on static shots. However, while covering the space in this manner takes only a couple of minutes, this procedure presents a different problem: because the sensors are not synchronized, they do not capture simultaneous images of the checkerboard. Because of this, while correspondences across two different sensors do represent the same location on the checkerboard itself, they do not triangulate to a unique position in 3D space, as the checkerboard experienced motion between each sensor's acquisition. We must therefore compensate for this mismatch in our calibration algorithm.

3.3.2.1 Temporal Alignment

Each of the Kinect v2 sensors runs on an independent clock. Neither the shutter nor the clocks across different sensors are synchronized, nor is there a common time reference across sensors. Since synchronizing the shutters is not possible without major changes to the sensor's hardware, we instead simply temporally align the streams, i.e., we reference each stream to a common global clock. This means that there is potentially a time offset of up to 15 ms between the times when two different sensors capture the scene. During fast motion, this becomes noticeable as ghosting, or a "doubled" version of the dynamic object. This can be seen in Figure 3.7.

This does not necessarily have to be detrimental to the reconstruction: if we know precisely when each of the sensors acquired a frame (i.e., the time offset between them), we still have valid measurements of the scene—simply at different time instants.

However, to be able to use this information, we require sub-frame resolution in the relative capture time between different sensors. To obtain such sub-frame accuracy in the temporal alignment, we embed an LTC timecode into the audio track of the sensors. Unfortunately, the Kinect v2 does not have an audio input jack—instead, we directly splice the LTC audio

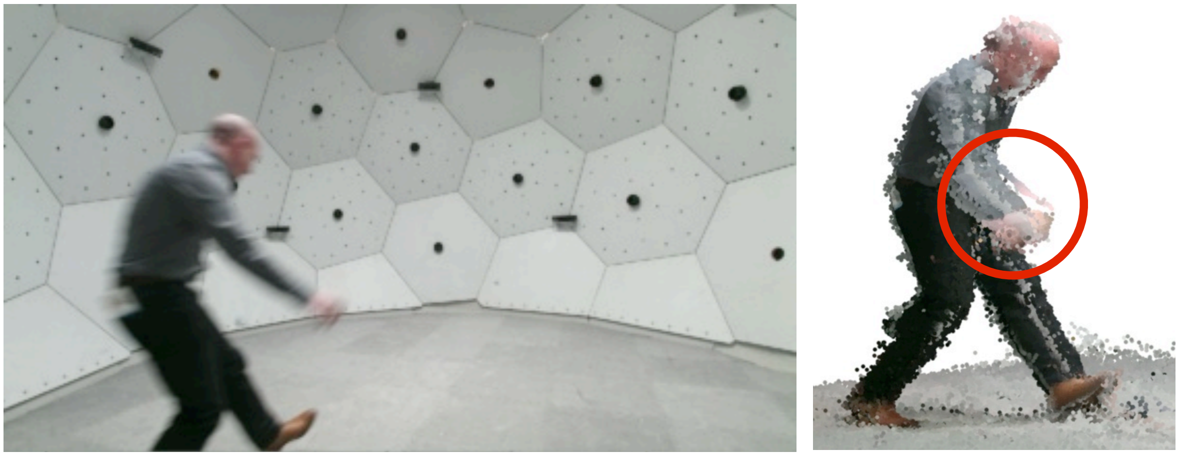


Figure 3.7: Fast motion can produce noticeable artifacts. This can be due to motion blur in the color images, but also, even in the absence of motion blur, because the shutters across different depth sensors are not synchronized. The obtained point clouds are then captured at slightly different time instants, producing noticeable motion artifacts in the reconstruction.

signal into one of the microphone signal lines. This requires a small hardware modification to the sensor itself, but yields an LTC timecode synchronized with the sensor's internal capture clock. We can decode the LTC timecode at a resolution of 16kHz, allowing for a theoretically maximum temporal resolution in the alignment of 125μs.

To complicate matters further, the color camera on the Kinect v2 is a rolling shutter camera. The implication of this for calibration with a moving checkerboard is that different scanlines *even in the same sensor's image frame* are captured at different times, with the top of the image frame being captured significantly earlier than the bottom of the image. Interestingly, these offsets in the observed checkerboard position provide a way to calibrate the temporal offset between sensors: by modeling the motion of the checkerboard as a function of time we can measure discrepancies between the observed position of the checkerboard (sampled at the combined frame rate of our full set of sensors) with our dynamical model of the checkerboard's position. We refer to this as a spatiotemporal calibration, as we must determine both geometric parameters as well as temporal alignment between sensors.

3.3.2.2 Spatiotemporal Calibration for Unsynchronized RGB-D Sensors

The parameters to be determined by calibration are the extrinsic and intrinsic camera parameters of each sensor (including RGB and depth sensors), as well as a temporal offset relating each unsynchronized sensor's measurements to a global clock. Additionally, the color camera's rolling shutter characteristics will be modeled by a single scalar measuring the time taken to a scan an entire frame. Similarly, the depth sensor has two additional parameters that determine the scaling from depth measurements to metric units, which we model as a linear function:

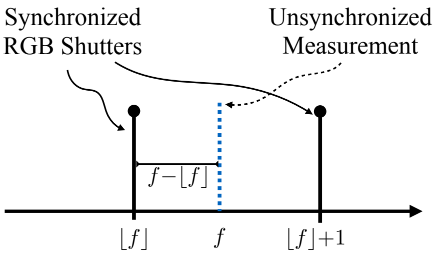
Symbol	Meaning
$\boldsymbol{\theta} \in \mathbb{R}^3$	Sensor rotation, as axis-angle
$\mathbf{t} \in \mathbb{R}^3$	Sensor translation
f_x, f_y	Focal length
c_x, c_y	Principal point, in pixels
k_1, k_2, k_3	Radial distortion parameters
p_1, p_2	Tangential distortion parameters
t_o	Temporal offset
λ	Rolling shutter duration (color only)
a	Depth scale factor (depth only)
b	Depth offset (depth only)

Each of these variables is specific to a particular sensor k (one of K sensors in total), and we omitted the superscript of k here for readability. Additionally, we must determine the time-varying angle-axis rotation and translation of the checkerboard, $\boldsymbol{\omega}(t) \in \mathbb{R}^3$ and $\mathbf{b}(t) \in \mathbb{R}^3$ respectively. In the following, we describe how each of these variables is related to the image formation process.

To model time-dependent variables—such as the dynamic position of the checkerboard—we sample and approximate each scalar value as a piecewise-linear function of time. That is, each temporally-dependent variable is discretized into a finite set of samples which occur at a constant frame rate R, with R representing the sampling rate in samples per second. We index these samples (also referred to as "frames") using a variable f, where f=0 is the first sample, f=1 the second, and so on. Defining time 0 as the time corresponding to the first sample, then $R^{-1}f$ is the continuous time associated with sample f. Because our sensors are not synchronized with a global clock, the shutters may acquire an image at an arbitrary time t, which will in general not coincide with our discrete samples but rather fall *between* two of our samples, associated with a continuous "frame" value of

$$f = Rt, \tag{3.1}$$

where $f \in \mathbb{R}$ is a real number. This way, $\lfloor f \rfloor$ is the sample immediately before time instant t, and $\lfloor f \rfloor$+1 is the sample immediately after time instant t. We approximate the value of any temporally sampled variable as a linear interpolation between these two samples. For instance, if the variable $\mathbf{x}_i \in \mathbb{R}^3$ represents the (discretely sampled) position of a certain point $\mathbf{x}$ at a sample index i, then we approximate the position of $\mathbf{x}$ at time t as,



$$\mathbf{x}(t) = \mathbf{x}_{\lfloor f \rfloor} + (f - \lfloor f \rfloor)\left(\mathbf{x}_{\lfloor f \rfloor + 1} - \mathbf{x}_{\lfloor f \rfloor}\right), \quad (3.2)$$

where $f = Rt$ (i.e., linear interpolation between the closest samples). In general, because of the direct relationship between them, we will refer to a specific time instant by either variable, t or f.

Let us denote the set of points on the checkerboard as $\mathbf{p}_j \in \mathbb{R}^3$ with $j \in \{1, \ldots, P\}$ with P the number of internal corners in the pattern. The 3D coordinates $\mathbf{p}_j$ are given in the local reference frame of the checkerboard and therefore not dependent on time. However, the position in world coordinates of the moving checkerboard does change, and we denote this position for point j at a time instant t as $\mathbf{x}_j(t)$. Expressing the motion of the checkerboard as a rigid transform,

$$\mathbf{x}_j(t) = \mathbf{T}(t)\begin{pmatrix}\mathbf{p}_j \\ 1\end{pmatrix}, \quad (3.3)$$

where $\mathbf{T}(t) \in \mathbb{R}^{3\times4}$ is the rigid transform as a function of time. We parameterize this transform as an angle-axis rotation $\boldsymbol{\omega}(t) \in \mathbb{R}^3$ and translation $\mathbf{b}(t) \in \mathbb{R}^3$, and interpolate the transform matrices directly[2] (rather than interpolating the angle-axis rotation parameters, e.g., [Grassia 1998]) such that

$$\mathbf{T}(t) = \mathbf{T}_{\lfloor f \rfloor} + (f - \lfloor f \rfloor)\left(\mathbf{T}_{\lfloor f \rfloor + 1} - \mathbf{T}_{\lfloor f \rfloor}\right), \quad (3.4)$$

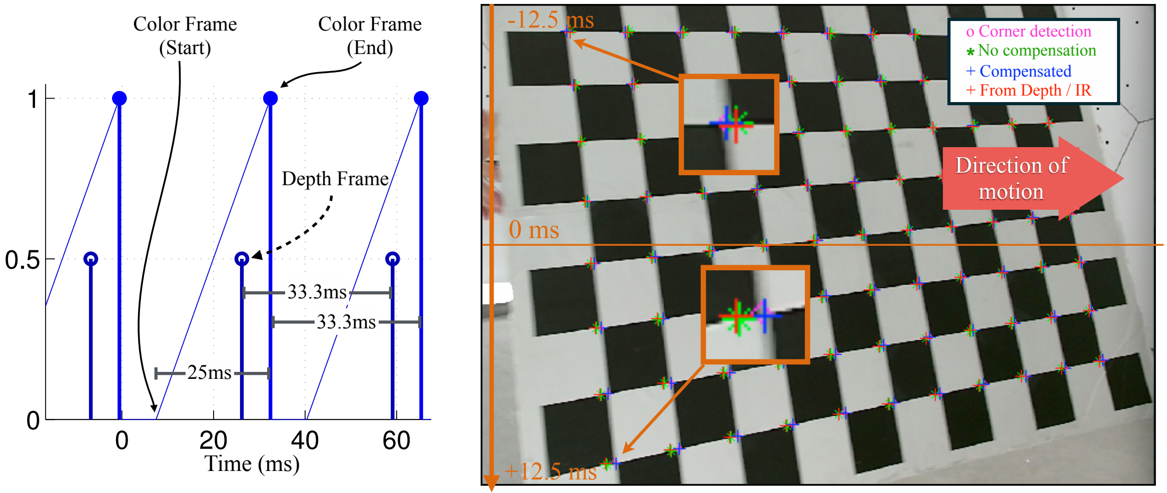
with $\mathbf{T}_i = [\exp(\boldsymbol{\omega}_i)|\mathbf{b}_i]$ for a particular sample i, and $\exp(\cdot) \in \mathbb{R}^{3\times3}$ is the matrix form of Rodrigues' rotation formula[3] [Belongie 2014]. We cannot observe the checkerboard's 3D position directly, but we do observe images of the checkerboard at a set of discrete time instants captured by each sensor. We will denote by

$$\mathbf{y}_j^{k,t'} = \mathcal{P}_k\left(\mathbf{x}_j(t' + t_o^k)\right) + \mathbf{e}, \quad (3.5)$$

the projection of point $\mathbf{x}_j$ into sensor k's image frame, with $\mathbf{y}_j^{k,t'} \in \mathbb{R}^2$ the observed pixel position and $\mathbf{e} \in \mathbb{R}^2$ a measurement error. Here, t' denotes the time the measurement was captured relative to the sensor's clock, which is related to the global clock by an unknown

[2] This is equivalent to linearly interpolating each point's position.

[3] $\exp(\boldsymbol{\omega}) = \mathbf{I}_3 + \sin\|\boldsymbol{\omega}\|\left[\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}\right]_\times + (1 - \cos\|\boldsymbol{\omega}\|)\left[\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}\right]_\times^2$, with $[\cdot]_\times$ the skew-symmetric (or cross-product) matrix constructed from its vector input.



(a) Color vs. depth acquisition measured timings

(b) Corner displacement due to rolling shutter

Figure 3.8: Rolling shutter effects. (a) Color vs. depth acquisition times on the Kinect v2 sensor. (b) Effect of rolling shutter on a moving checkerboard. Pink circles denote detected corners in the image plane. Green stars denote the estimated corner locations according to Eq. (3.5), which doesn't consider rolling shutter effects. Blue markers show the reprojections using rolling shutter compensation of Eq. (3.9). Red markers denote the position of the checkerboard corners as measured by the IR camera.

offset t_o, such that $t = t' + t_o^k$. Camera projection into sensor k's image is represented as $\mathcal{P}_k(\mathbf{x})$, expressed as pixels. Specifically, the projection function for each sensor is specified by its camera transform $[\exp(\boldsymbol{\theta}^k)|\mathbf{t}^k]$ (where $\boldsymbol{\theta}^k \in \mathbb{R}^3$ is the camera rotation as angle-axis and $\mathbf{t}^k \in \mathbb{R}^3$ is the camera's translation), its intrinsic parameters, and distortion parameters (see e.g., [Hartley and Zisserman 2003]). Specifically, the final 2D pixel coordinates of a projected point $\mathcal{P}_k(\mathbf{x})$ are given by first transforming the point into sensor k's camera frame, $\mathbf{X} = \exp(\boldsymbol{\theta}^k)\mathbf{x} + \mathbf{t}^k$, projection,

$$x_p = \frac{[\mathbf{X}]_1}{[\mathbf{X}]_3}, \qquad y_p = \frac{[\mathbf{X}]_2}{[\mathbf{X}]_3},$$

radial and tangential distortion,

$$x_d = x_p(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x_p y_p + p_2(r^2 + 2x_p^2) \tag{3.6}$$

$$y_d = y_p(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_2 x_p y_p + p_1(r^2 + 2y_p^2) \tag{3.7}$$

(with $r = x_p^2 + y_p^2$) and pixel sampling in the image coordinate system,

$$\mathcal{P}_k(\mathbf{x}) = \begin{pmatrix} f_x x_d + c_x \\ f_y y_d + c_y \end{pmatrix}. \tag{3.8}$$

Color sensor. The color camera's rolling shutter requires an additional calibration parameter to describe the time offset between different scanlines of the sensor. We express this in terms of the rolling shutter capture duration λ (in milliseconds), which is related to the rolling shutter speed. If we define the middle scanline of a frame as the reference, the first scanline will be acquired 0.5λms earlier, and the bottom scanline 0.5λms after the middle scanline. This effect can be seen in Fig. 3.8b, where the observed location of the checkerboard corners does not match the location measured in the depth sensor's image, unless the rolling shutter effect is compensated for. To compensate for this effect for a particular pixel at location $\mathbf{y}_j^{k,t'}$, we can determine it's acquisition time relative to the middle of the frame as $\lambda\delta$, with $\delta = (vH^{-1} - 0.5)$ the relative position of the measurement in the frame, with v the y coordinate of the measurement and H is the image height. We modify our projection equation as

$$\mathbf{y}_j^{k,t'} = \mathcal{P}_k\left(\mathbf{x}_j(t' + \lambda\delta + t_o^k)\right) + \mathbf{e}. \tag{3.9}$$

Depth sensor. For the depth sensor, we assume that $\lambda = 0$ (all scanlines are acquired simultaneously). The depth sensor additionally provides a measurement of the depth of each observed checkerboard corner. Let us denote this depth measurement by $d_j^{k,t'}$. Then, $\mathbf{X}_j(t) = \exp(\boldsymbol{\theta}^k)\mathbf{x}_j(t) + \mathbf{t}^k$ are the coordinates of the point transformed into the sensor's reference coordinate system and

$$d_j^{k,t'} = a[\mathbf{X}_j(t' + t_o^k)]_3 + b + \epsilon,$$

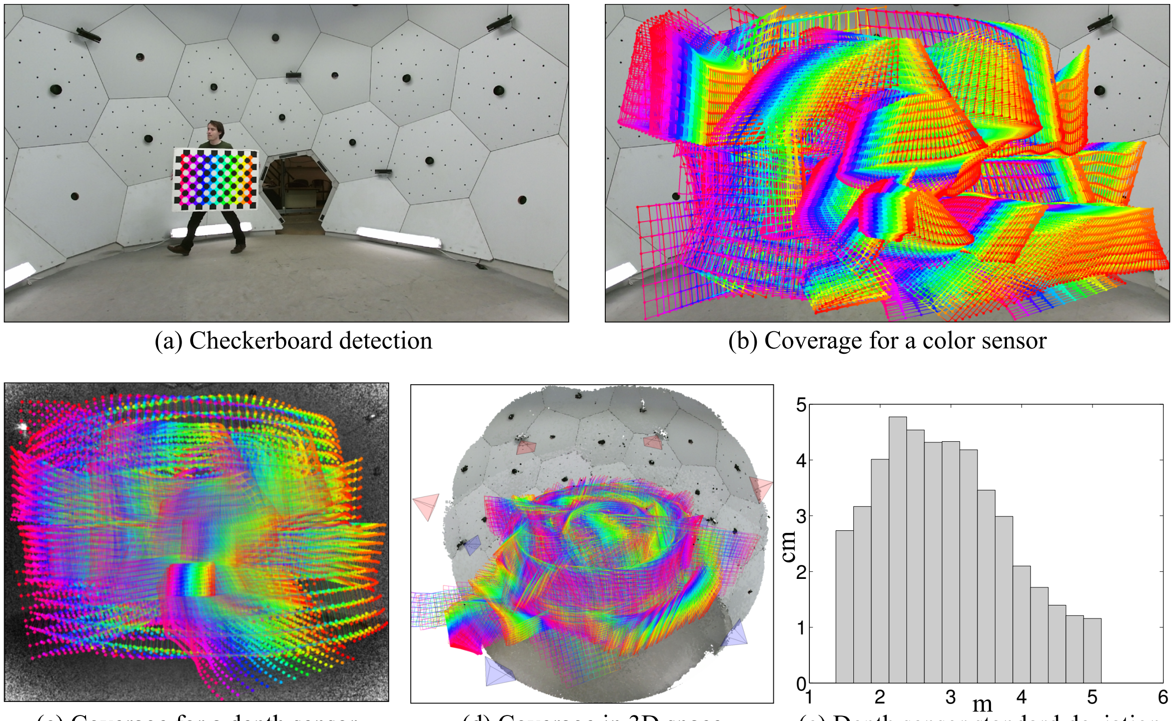
where a is a scaling factor, b is a bias term, and ϵ is random noise in the measurement. By factory default, a=1000 (for depth $[\mathbf{X}_j(t)]_3$ in meters) and b=0. We were not able to achieve a significant improvement in the calibration by allowing these values to vary.

To summarize, the data term related to image and depth measurements is

$$\sum_{j,k} \left\| \mathbf{y}_j^{k,t'} - \mathcal{P}_k\left(\mathbf{T}(t' + \lambda^k\delta + t_o^k)\begin{pmatrix}\mathbf{p}_j\\ 1\end{pmatrix}\right)\right\|^2 + \sum_{j,k}\left(d_j^{k,t'} - a^k\left[\mathbf{T}(t' + t_o^k)\begin{pmatrix}\mathbf{p}_j\\ 1\end{pmatrix}\right]_3 - b^k\right)^2, \tag{3.10}$$

where the sums are over all available measurements, where applicable. Additionally, we add a highly weighted term

$$\sum_{j:k\in\mathcal{N}(j)} \left(l - \|\mathbf{p}_j - \mathbf{p}_k\|\right)^2,$$



(a) Checkerboard detection

(b) Coverage for a color sensor

(c) Coverage for a depth sensor

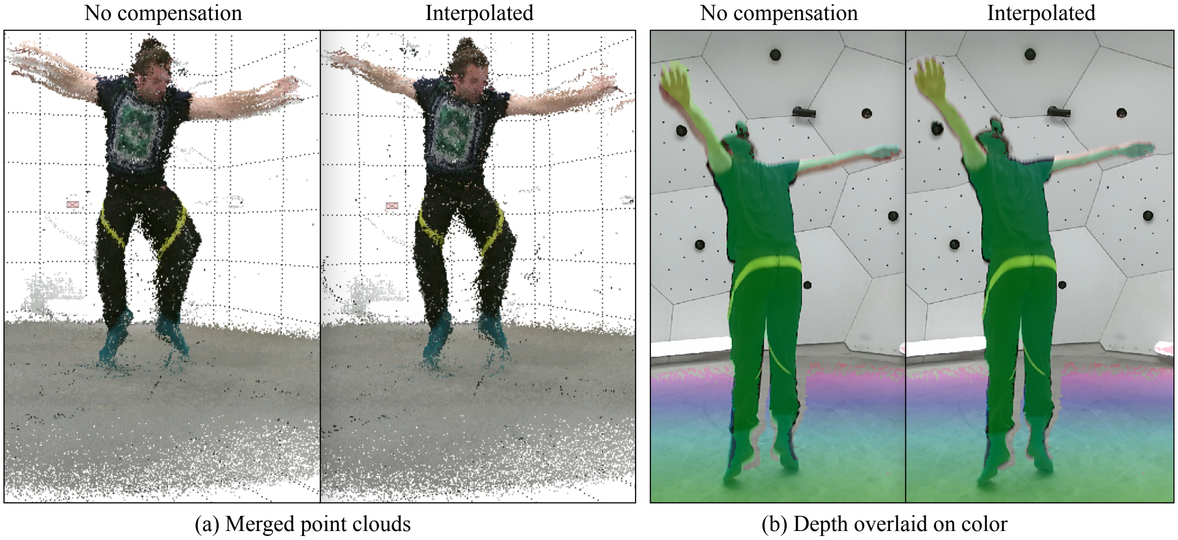
(d) Coverage in 3D space

(e) Depth sensor standard deviation

Figure 3.9: Calibration of the Kinect v2 sensors using a moving checkerboard. (a) Detection in a color image. (b) Detections after 3 minutes of capture, for a particular color camera. (c) Detections in the IR camera. (d) Coverage from all sensors in 3D space. (e) Depth measurement standard deviation as a function of distance.

where l is the known length (in metric units) of each of the corners of the checkerboard pattern, and $\mathcal{N}(j)$ denotes the set of neighbors of corner point j. We let $\boldsymbol{\theta}^1 = \mathbf{0}$ and $\mathbf{t}^1 = \mathbf{0}$ (i.e., the global frame coincides with the frame of the first sensor). We use each of the sensor's factory-calibrated internal parameters to initialize each sensor's intrinsics, and initialize the extrinsic transforms using Procrustes alignment of all depth cameras to the reference camera. The checkerboard corners are initialized to lie on a plane at their nominal locations (in metric units). We use the Ceres solver [Agarwal et al. 2009] solver to minimize Eq. (3.10).

Results We show results for a 3 minute calibration sequence using 10 Kinect v2 (K=20, 10 color cameras and 10 depth cameras). After checkerboard detection in each camera, the color sensors obtained on average 2200 detections, while the depth sensors obtained around 1500. The lower number is due to the depth sensor's image frame covering a smaller field of view, and the limited resolution which makes detection at large distances more difficult. Fig. 3.9b shows the reconstructed position of each checkerboard measurement in 3D, of all combined sensors. Fig. 3.9c and d show all the overlaid detections for two particular sensors. For this sequence, we reconstructed about 37000 checkerboards, or approximately 3.2 million points, of which 130 to 190 thousand points informed each sensor's calibration parameters. The



(a) Merged point clouds

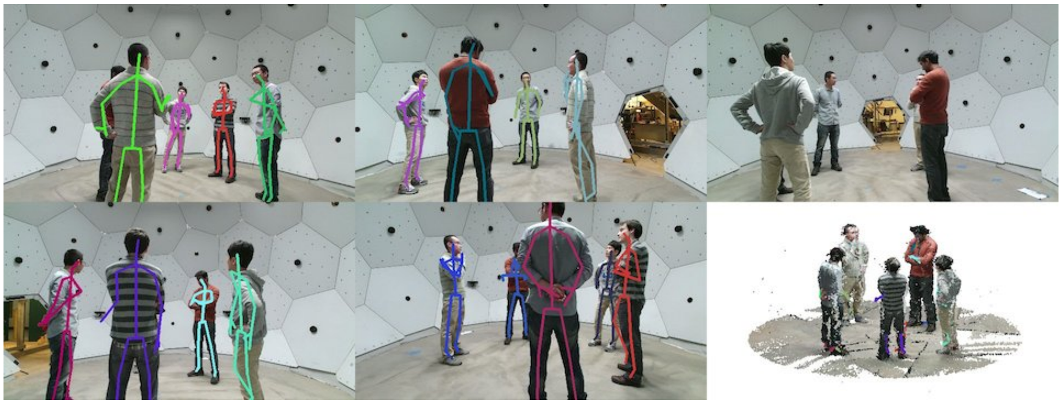
(b) Depth overlaid on color

Figure 3.10: Temporal effects on acquisition. (a) Merging the point clouds from 10 RGB-D sensors with only frame-level synchronization results in artifacts during fast motion (left). This can be alleviated by interpolating the measurements if the precise acquisition time of each sensor is known (right). (b) During fast motion, the color rolling shutter introduces alignment artifacts between the depth and color measurements within a single RGB-D sensor (left). Here, depth is shown as an overlay (green to purple) on the color. Using the calibrated rolling shutter duration, this effect can be compensated by warping the pixel locations (right). (c) Standard deviation in measured depths.

average reprojection error was 0.22px after calibration, and the average depth measurement error was 7.53mm. The estimated rolling shutter duration was approximately 25ms, with an IR image being capture approximately at the middle of a color frame acquisition. These timings are shown in Fig. 3.8a. Additionally, we consistently find that the optimized temporal offsets t_o^k are under one millisecond in magnitude, indicating that our hardware alignment (Sect. 3.3.2.1) is sufficiently accurate to disregard these offsets in practice.

While these reprojection errors are low on average, we caution that the depth measurements in particular show some systematic errors, and their standard deviation is larger than the average error (and depth-dependent, as shown in Fig. 3.9e). We believe that at least some these errors are due to interferences between the different sensor's time-of-flight cameras, as we were unsuccessful in compensating for them consistently. Additional sources of errors in practice are the temporal offset between the sensors and the rolling shutter effect, particularly during fast motion. Fig. 3.10a (left) shows the effect of overlaying the point clouds of our 10 sensors captured within less than 33.3ms of each other (i.e., assuming they are synchronized to the frame level). We can compensate for this effect by interpolating the point clouds to their appropriate locations (right). However, this requires establishing correspondences between neighboring frames and is computationally expensive. Similarly, Fig. 3.10b shows artifacts in the synchronization between the color image and the acquired

Figure 3.11: The Kinect's pose detection algorithm run on each sensor independently fails in the presence of occlusions and back- or side-facing people. The sixth vignette shows a 3D reconstruction.



depth due to the rolling shutter (left). The acquired depth is overlaid on the color image, for which there is a 25ms delay between the top of the image and the bottom. Again, this can be compensated by interpolation (right), but establishing correspondences between neighboring frames is expensive.

3.4 Markerless Motion Capture in the Panoptic Studio (Background Information)

The main goal of the Kinoptic Studio is to study the body movements of interacting people, and we want to be able to recover even *subtle* 3D motion of individual body joints and other points on the body, such as the face. Obtaining these trajectories is not trivial for groups of people—this requires measuring small changes in position (with a resolution of centimeters) over a large, room-sized capture area. For example, using the built-in pose detection algorithm of the Kinect produces several undesirable artifacts. The method works well for typical gaming scenarios, where the people are facing a screen on which the Kinect sensor is mounted, but it often fails when capturing people that are talking amongst themselves. Most importantly, the built-in pose detection algorithm was designed for frontal facing people, and the estimated pose is usually severely wrong (e.g., completely flipped) for people facing sideways or away from the camera.

To recover accurate poses for *groups* of people within 360° scenes, a single viewpoint does not suffice. The depth information in a single view is often too crowded, occluded, ambiguous, or otherwise unfavorable for pose recovery—merging information from several sensors is therefore necessary. Figure 3.11 shows how the single-view pose estimation algorithm of a Kinect v2 fails to recover accurate poses. As discussed above, the main problems are oc-

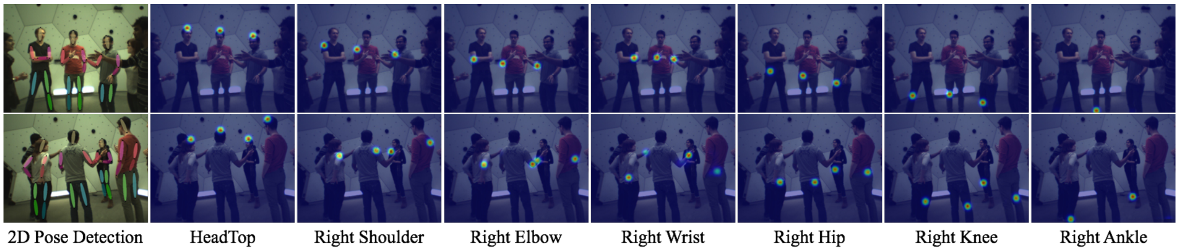


Figure 3.12: 2D pose detections and score maps generated by the method of [Wei et al. 2016]. (Column 1) Example views out of 480 views with proposals by the pose detector (Column 2-7) Heat maps for each node on each view. Note that the body pose detector distinguishes left-right limbs.

clusions between people and non-frontal people. Note, however, that this pose estimation algorithm operates only on the data from a single viewpoint.

Instead, our algorithm takes, as input, images from multiple views at a time instance (calibrated and synchronized), and produces 3D body skeletal proposals for multiple human subjects by integrating 2D pose detections across the many views of our massively multiview system, fusing simple 2D cues to estimate 3D skeletal poses at each time instance. While detections in any single view may be incomplete or inaccurate—typically due to occlusions—we find that aggregating these cues across many views yields very stable results. Our method is simple, yet robust thanks to the large number of views. In contrast, prior marker-less motion capture methods are typically "model-dependent", requiring a 3D template model to constrain shape deformations, a motion model to constrain temporal deformations, and a relatively complex energy function minimization that trades off each of these priors (e.g., [Gall et al. 2009; Furukawa and Ponce 2008; Elhayek et al. 2016]). Our method in this stage is essentially based on triangulating detections at a single time instance, and, thus, does not suffer from error accumulation or drift. It does not require a 3D template model, prior assumptions about the subject or the motion, or an initial alignment for tracking. In this section, we describe how the proposals are generated and built up from 2D cues.

3.4.1 Overview

A 2D pose detector [Wei et al. 2016] is computed independently on all 480 VGA views at each time instant t, generating detection score maps for each body joint (see Fig. 3.12). The 2D score maps for each body joint $j \in \{1, \cdots, J\}$ are combined into a 3D score map $H_j(\mathbf{Z})$ by projecting a grid of voxels $\mathbf{Z} \in \mathbb{R}^3$ onto the 2D score maps and computing an average 3D score at each voxel (subsection 3.4.2).

Our approach then generates several levels of proposals. A set of *node proposals* $\mathbf{N}_j$ for each joint j is generated by non-maxima suppression of the 3D score map $H_j(\mathbf{Z})$, where the

k-th node proposal $\mathbf{N}_j^k \in \mathbb{R}^3$ is a putative 3D position of that anatomical landmark. Similarly, the set of *part proposals* is denoted by $\mathbf{P}_{uv}$, where u and v are joints and $(u, v) \in \mathbf{B}$ is the set of body parts or *bones* composing a skeleton hierarchy. The k-th part proposal, $\mathbf{P}_{uv}^k = (\mathbf{N}_u^{k_u}, \mathbf{N}_v^{k_v}) \in \mathbb{R}^6$, is a putative body part connecting two node proposals, $\mathbf{N}_u^{k_u}$ and $\mathbf{N}_v^{k_v}$, where the index k enumerates all possible combinations of k_u and k_v. As the output of the first stage, our algorithm produces *skeletal proposals*; we refer to the k-th proposal as $\mathbf{S}^k = \{\mathbf{P}_{uv}^k\}_{uv \in \mathbf{B}}$. A skeletal proposal is generated by finding an optimal combination of part proposals using a dynamic programming method under the score function defined in subsection 3.4.4. Here, we abuse the notation to have $\mathbf{P}_{uv}^k$ refer to the optimally assigned part u, v of skeleton k (the superscript k is understood to be the optimal mapping, from context). After reconstructing skeletal proposals at each time t independently, we associate skeletons from the same identities across time and generate *skeletal trajectory proposals* $\tilde{\mathbf{S}}^k(t) = \{\tilde{\mathbf{P}}_{uv}^k(t)\}_{uv \in \mathbf{B}}$, where $\tilde{\mathbf{P}}_{uv}^k(t)$ is a *part trajectory proposal*, a moving part across time, with k similarly overloaded to denote the optimal associations determined in each frame t.

3.4.2 3D Node Score Map and Node Proposals

A single-view 2D pose detector is computed on all VGA views at each time instant, and is used to generate 2D pose detections and per-joint score maps in each image. Because the first stage of our method is performed at each time independently, we will consider a fixed time instant t, and drop the time variable for clarity. We use the detector of Wei et al. [2016] without additional training. The method of Wei et al. requires bounding box proposals for each human body as initialization, thus, we first apply a person detector similar to R-CNN [Girshick et al. 2014], and run the pose detector on the detected person proposals represented as bounding boxes. Each 2D skeleton detection i in a camera view c is denoted by $\mathbf{s}_i^c \in \mathbb{R}^{2\times 15}$, and is composed of 15 anatomical landmarks or *nodes* (3 for the head/torso and 12 for the limbs), also referred to as joints[4]. The position of the j-th node of the i-th person detection is denoted by $\mathbf{s}_{ij}^c \in \mathbb{R}^2$. The method of Wei et al. also provides a score map representing the per-pixel detection confidence for each node s_{ij}^c, which we denote as $h_{ij}^c(\mathbf{z}) \in [0, 1]$, where $\mathbf{z} \in \mathbb{R}^2$ indexes 2D image space. We also compute a merged score map by taking the maximum across all person detections at each pixel, $h_j^c(\mathbf{z}) = \max_i h_{ij}^c(\mathbf{z})$. Merged score maps of example views are shown in Figure 3.12.

To combine 2D node score maps from multiple views into 3D, we generate a 3D score map for each node using a spatial voting method. We first index the 3D working space into a voxel grid (4cm in our implementation), and compute the *node-likelihood* score of each voxel

[4] We modify the skeleton hierarchy of [Wei et al. 2016] to have an explicit torso bone, by taking the center of the two hip nodes as a body center node.

by projecting the center of the voxel to all views and taking the average of the 2D scores at the projected locations. The 3D score map $H_j(\mathbf{Z})$ for a node j at the 3D position $\mathbf{Z} \in \mathbb{R}^3$ is defined as

$$H_j(\mathbf{Z}) = \frac{1}{|V(\mathbf{Z})|} \sum_{c \in V(\mathbf{Z})} h_j^c\left(\mathcal{P}_c(\mathbf{Z})\right), \tag{3.11}$$

where $\mathcal{P}_c(\cdot) \in \mathbb{R}^2$ denotes projection into camera c, $V(\mathbf{Z})$ is the set of cameras where the 3D location $\mathbf{Z}$ is visible, and $|V(\mathbf{Z})|$ is the cardinality of the set. Note that the 3D score map for each node is computed separately, producing fifteen 3D score maps at each time instant.

From the 3D score map for each node at each time instance, we perform Non-Maxima Suppression (NMS), and keep all the candidates above a fixed threshold τ (we use τ=0.05). The results are shown in Fig. 3.13, color-coded by the node score. Each node proposal, denoted as $\mathbf{N}_j^k$ for the k-th proposal for node j, is a putative candidate for the j-th anatomical landmark of a participant.

3.4.3 Part Proposals

Given the generated node proposals, we infer part proposals by estimating connectivity between each pair of nodes that make up a possible body part. The 2D detector [Wei et al. 2016] uses appearance information during the inference, and, thus, the result tends to preserve connectivity information (e.g., left knee is connected to the left foot of the same person). Our approach fuses them by voting 2D connectivity into 3D. More specifically, we define a connectivity score between a pair of node proposals by projecting them onto all views, and checking in how many views they are actually connected, i.e., both nodes belong to the same person detection. Formally, the connectivity score of a part $\mathbf{P}_{uv}^k$ between two node proposals $(\mathbf{N}_u^{k_u}, \mathbf{N}_v^{k_v})$, where $(u, v) \in \mathbf{B}$, is defined as

$$\Phi(\mathbf{P}_{uv}^k) = \frac{1}{|V(\mathbf{P}_{uv}^k)|} \sum_{c \in V(\mathbf{P}_{uv}^k)} \max_i \phi_{iuv}^c\left(\mathcal{P}_c(\mathbf{N}_u^{k_u}), \mathcal{P}_c(\mathbf{N}_v^{k_v})\right),$$

$$\phi_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v) = w_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v)\delta_{iuv}^c\left(\mathbf{z}_u, \mathbf{z}_v\right)$$

where

$$w_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v) = \frac{1}{2}\left(h_{iu}^c(\mathbf{z}_u) + h_{iv}^c(\mathbf{z}_v)\right), \text{and}$$

$$\delta_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v) = \begin{cases} 1 & \text{if } h_{iu}^c(\mathbf{z}_u) > \tau \text{ and } h_{iv}^c(\mathbf{z}_v) > \tau \\ 0 & \text{otherwise.} \end{cases}$$

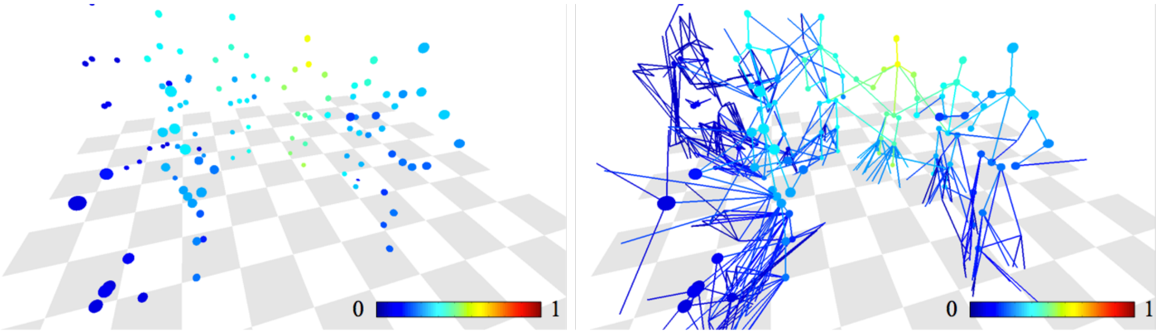


Figure 3.13: Computed scores for node proposals and part proposals. The color encodes scores.

Here, $\mathcal{P}_c(\mathbf{N}_u^{k_u})$ and $\mathcal{P}_c(\mathbf{N}_v^{k_v})$ are the projections of the two nodes of $\mathbf{P}_{uv}^k$ in view c, and $V(\mathbf{P}_{uv}^k)$ is the set of cameras where the 3D part is visible. Intuitively, the part score Φ represents the average connectivity score across all views from all potentially corresponding 2D person detections. Because we do not know the correspondence between 3D parts to 2D person detections, we take the maximum score across all possible detections i in each view. Assuming that the projected part corresponds to the i-th person detection in camera c, the part connectivity score ϕ_{iuv}^c is defined as the average score of the projected nodes, denoted by $w_{iuv}^c(\mathbf{z}_u, \mathbf{z}_v)$. The delta function δ_{iuv}^c additionally ensures that ϕ_{iuv}^c is nonzero only if both projected node locations have a sufficiently high score for the same detection i (i.e., both nodes are detected as part of a single person). An example of computed part scores is shown in Figure 3.13.

3.4.4 Generating Skeletal Proposals by Dynamic Programming

Our method generates skeleton proposals by piecing together the part proposals. Since each skeleton is a tree structure, this can be computed efficiently using Dynamic Programming (DP)—but only for a single person. Therefore, we use DP to greedily find 3D skeletons $\mathbf{S}^k$ which maximize the sum of part scores,

$$\Theta(\mathbf{S}^k) = \max_{(k_1,\cdots,k_J)} \sum_{(u,v)\in\mathbf{B}} \Phi\left(\mathbf{P}_{uv}^k\right).$$

A skeleton $\mathbf{S}^k$ is given by the mapping $k \mapsto (k_1, \cdots, k_J)$, where the J-tuple $(k_1, \cdots, k_J)$ determines the assignment of node proposals $\mathbf{N}_j^{k_j}$ for each joint j in the body. After picking the highest scoring skeleton $\Theta(\mathbf{S}^k)$, the assigned nodes $(k_1, \cdots, k_J)$ are removed from the pool of available node proposals and we run DP again to find the next highest scoring skeleton, and so on until all possible skeletons are found.

One option here would be to threshold the skeleton scores $\Theta(\mathbf{S}^k)$ at some minimum value to determine valid detections. However, we can do better: each 3D skeleton should be supported by 2D detections, and each 2D detection can correspond to only a single 3D skeleton. This observation is important because the voting used to generate 3D node proposals assigns equal score to *all* voxels along the line of sight of each 2D detection (Sect. 3.4.2), and, similarly, the max over detections in the part score $\Phi(\cdot)$ makes $\Theta(\mathbf{S}^k)$ an overestimate.

To avoid this form of double counting, our method places each 3D node $\mathbf{N}_j^k$ in skeleton $\mathbf{S}^k$ in correspondence with the closest 2D joint detection in each view. For each 3D node $\mathbf{N}_j^k$, we create a set of correspondences $\mathcal{C}_j^k$ with elements (c, i) such that the distance $\|\mathcal{P}_c(\mathbf{N}_j^k) - \mathbf{s}_{ij}^c\|_2$ is the minimum across all detections i in view c and smaller than δ=10px. Once a 2D correspondence is established, we remove it from the set of available 2D detections, and, as above, this is performed greedily in order of decreasing skeleton score $\Theta(\mathbf{S}^k)$. Skeletons where the head node has fewer than two correspondences are discarded, i.e., if $|\mathcal{C}_j^k| < 2$ for j the head.

We additionally use the set of correspondences $\mathcal{C}_j^k$ to refine the 3D node locations by minimizing reprojection error. This overcomes the discretization error introduced by the voxel grid resolution. The final 3D node location $\hat{\mathbf{N}}_j^k$ is then

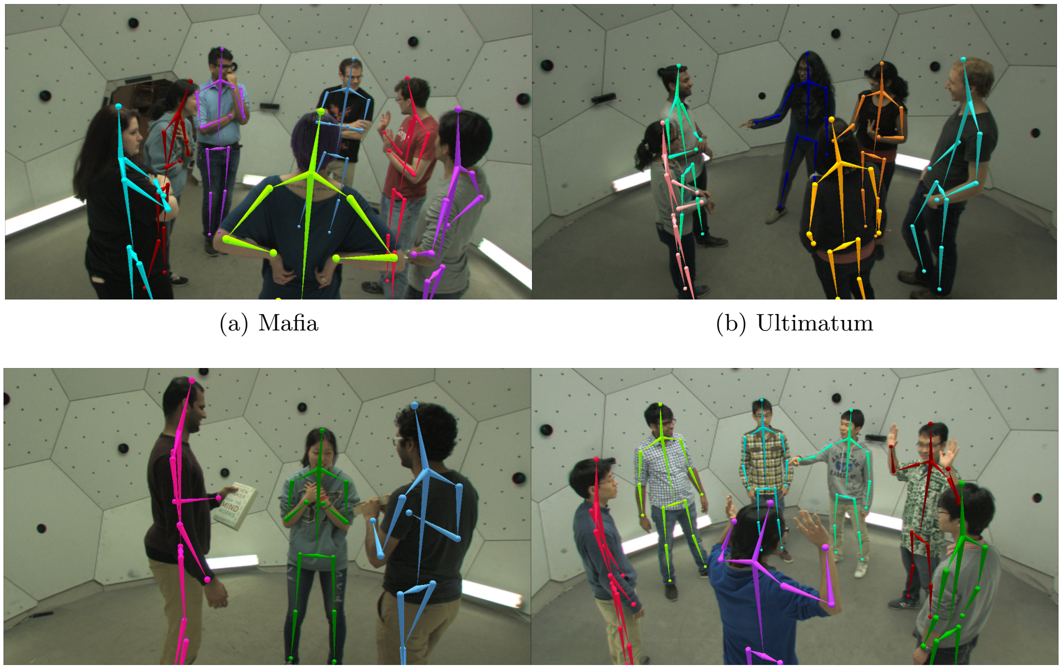
$$\hat{\mathbf{N}}_j^k = \arg\min_{\mathbf{Z}} \sum_{(c,i)\in\mathcal{C}_j^k} \left\|\mathcal{P}_c(\mathbf{Z}) - \mathbf{s}_{ij}^c\right\|_2 .$$

The output of the algorithm described in this section is 3D skeletal proposals reconstructed independently at each time instance. After performing this process on all frames, our method associates skeletons from the same identity across time by simply considering spatial distance of the head node. That is, for a $\mathbf{S}_t^{k_1}$ reconstructed at time t, we find a corresponding skeleton at $\mathbf{S}_{t+1}^{k_1}$ with the closest head node location from $\mathbf{S}_t^{k_1}$ within a threshold. To be somewhat robust to missing skeleton detections, our method associates across a window of time. If there is no corresponding skeleton at time $t+1$, we also consider the next time $t+2$ and find a corresponding skeleton.

3.4.5 Dataset and Capture Procedures

We captured a group of people engaged in social interactions using the Panoptic Studio[5]. To evoke natural interactions, we involved participants in various games: *Ultimatum*, *Mafia*, *Haggling*, and *007-Bang Game*. The first two games are used in experimental economics and psychology to study conflict and cooperation, and the latter two games also induce a variety of rich non-verbal signals in participants. Example scenes of each game are shown in Figure 3.14.

[5] Some sequences were captured with fewer than the full set of cameras due to hardware failures.



(a) Mafia

(b) Ultimatum

(c) Haggling

(d) 007-Bang

Figure 3.14: Example scenes of social game sequences. The reconstructed 3D skeletons from the 480 VGA views are projected on novel HD views.

In our captures, subjects were informed of the rules of the game but were otherwise not instructed about how to behave, nor was their clothing or appearance controlled. They were also not initially aware of our research goals to avoid potential biases in their gestures[6]. The scenes in our dataset contain various natural motions which may commonly occur in the interactions of daily life.

To additionally demonstrate the performance of our system and methods, we capture other challenging sequences, including a group of eight seated people participating in a discussion (*meeting* sequence), a mother and a toddler at play (*toddler* sequence), musical performances with severe occlusions due to the instruments (*drummer* and *cellist*), and a sequence featuring various fast motions and challenging postures (*dancer*).

In aggregate, the dataset contains about 198 minutes (∼297K frames) of videos, for a total of about 154 million images. The main distinguishing features of this collection compared to previous markerless motion capture datasets are: (1) natural interactions in the scenes showing rich and subtle non-verbal cues, (2) social groups of up to 8 interacting people, and

[6] The majority of the sequences are captured with people randomly recruited from a university campus; some sequences were captured for testing purposes and feature researchers with knowledge of the project. Those sequences are marked in our dataset website.

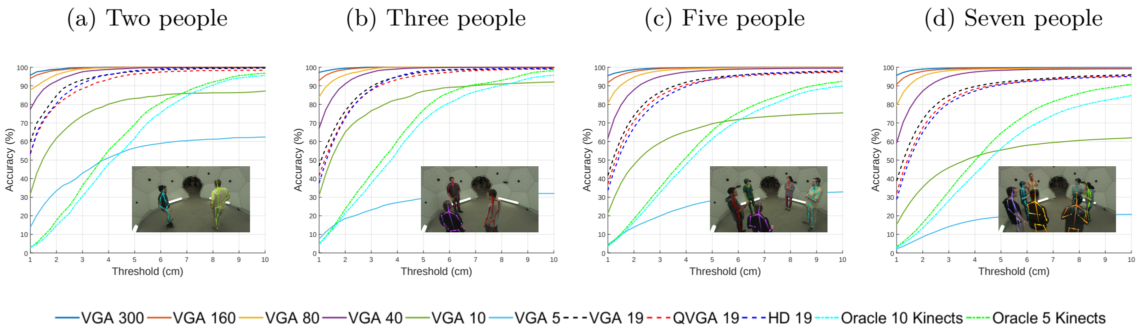


Figure 3.15: Performance evaluation using Probability of Correct Keypoint (PCK) metric for varying number and type of cameras on *160422 ultimatum1*. We use the result of 480 VGA cameras after manually excluding outliers as ground truth. The X-axis of each graph represents thresholds, and the Y-axis represents accuracy by the thresholds. Each graph is generated for scenes with a different number of people. The results demonstrate that more views (rather than higher resolution) are beneficial to improve accuracy, and the distinction is more noticeable if the scene contains more people.

Table 3.1: Processing time for one minute of data.

	Procedure	Time
	(3.4.2) 2D pose detection (1 GPU)	40 h
	(3.4.2-3.4.3) Node and part proposal recon. (1 GPU)	4 h
	(3.4.4) Skeletal proposal reconstruction by DP	3 m
	(3.4.4) Skeletal proposal optimization	11 m

(3) coverage by a large number of views (up to 521). We make all the data available on our website, including all synchronized camera feeds, calibration, 3D pose reconstruction results, and 3D trajectory streams: `https://domedb.perception.cs.cmu.edu`.

3.4.6 Processing Time

The time to process one minute of data (1500 frames) of 480 VGA views is summarized in Table 3.1. We use different computing devices for procedures. A machine with Intel i7 3.4GHz processor and 32GB RAM is used for general processing, a GTX Titan X is used for GPU computation, and a cluster server with 400 CPU cores (2.2GHz per processor) is used for trajectory stream reconstruction.

In the first stage, most of the time is spent in running the 2D body pose detector. The detector runs at about 5 frames per second on a single GPU, but due to the large number of views (720K images per minute), processing a minute of video takes about 40 hours. In practice, we use multiple GPUs to process multiple images in parallel. In the second stage, the main computational bottleneck is the trajectory stream generation. Although they are tracked in parallel, the running time is long due to the large number of patches at each time.

Table 3.2: Quantitative evaluation of the accuracy of our method on the *160422 ultimatum1* sequence.

Skel. #	Node #	Outlier Node #	Node Acc.	Skel. Acc.
81,829	1,227,435	8700	99.29%	93.55%

In our experiments, on average 15K patches are tracked per person.

3.4.7 Performance Analysis

We quantitatively evaluate the performance of our method for the *160226 ultimatum1* sequence by varying the number and type of cameras. We choose the ultimatum sequence because it captures varying number of people (from two to seven people) in each time period, which is suitable to study the relation between scene complexity and the number of cameras needed to reach a desired performance. In this experiment, we only evaluate the first stage of our method.

Performance using all VGA cameras: We first quantify the performance of our system when all 480 VGA cameras are used. Due to the absence of ground truth data, we manually annotate the correctness of the reconstructed 3D skeletons by verifying their projections in multiple 2D views. We labeled a 3D joint node as an outlier if the node is projected outside of the corresponding limb or far from the presumable target joint in multiple 2D views. We exclude the period where people come in and out of the system, since at that moment body parts lie on the edge of our system's working volume. The result of the quantitative evaluation for the 15 minutes of sequence is summarized in Table 3.2. There are 12 sessions in the sequence, and 61 temporally associated skeletal structures are reconstructed. Among about 1.2 million body joints, about 8.7K nodes are determined as outliers or missed (rejected by thresholds of our system), showing 99.29% accuracy in node reconstruction. And, 93.55% out of about 82K 3D skeletons are correctly reconstructed without any incorrect joints. The majority of the failures are caused by insufficient visibility of the target part. An example is the pose holding hands behind one's back near the wall of the system as shown in Figure 3.16 (left). Although the hands are visible from few cameras, they are too close to be detected by the pose detector. Interestingly, our method still reconstructs the hands using the "guessed" 2D locations from 2D pose detector in frontal views, although the accuracy is limited.

Comparison with varying number of cameras: To evaluate the impact of the number of views, we perform our method using varying number of cameras. The cameras are uniformly sampled (except the 19 VGA camera case explained later); i.e., we sample the next camera as the one furthest from all the already sampled cameras, and, thus, the selected

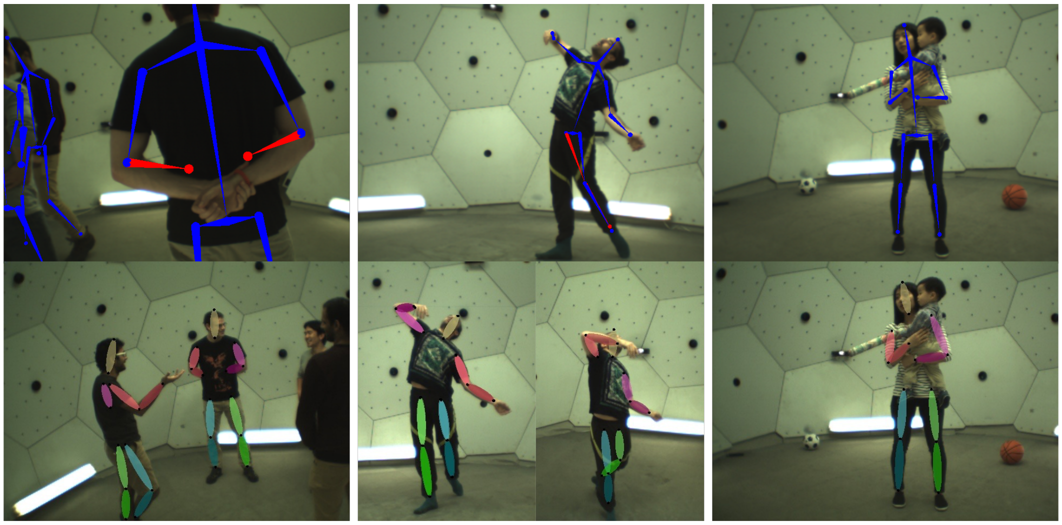


Figure 3.16: Example failure cases. For each column, the first row shows the projection of reconstructed 3D skeletons on a view where the red colored parts are manually annotated outliers. The second row shows the 2D pose detection results. (Left) The hands are severely occluded and only visible from few cameras where they are too close to be detected by 2D pose detector. (Center) The left/right legs are confused in performing 2D pose detection, which causes failures in our 3D inference. (Right) The toddler is not detected by the pose detector, since he is severely occluded.

cameras are always a subset of the set of the larger number of cameras. To quantify the results, we treat the result with 480 VGA cameras as ground truth after excluding the manually annotated outliers. For evaluation, we only use every tenth frame to reduce computation time. As an evaluation metric, we use the PCK (Probability of Correct Keypoint) metric, which is commonly used to evaluate 2D pose detectors [Yang and Ramanan 2013]. Here, we use 3D distance in physical scale (cm) obtained from calibration data for the threshold of PCK, in contrast to the 2D ratio of torso/head as in 2D pose detection cases [Andriluka et al. 2014]. Figure 3.15 shows the PCK accuracy by varying the camera number on the scenes with different number of people. In all the results, we find that using a larger number of views is beneficial. If the scene is simpler (e.g., the case with two or three people), we observe that the results with a smaller number of cameras, e.g., 160 cameras, show a similar performance with 480 cameras. However, if the scene becomes more complicated, e.g., seven people, we see clearer gaps according to the camera numbers. This results can be meaningfully used to design a multiple camera system to determine the required number of cameras given a desired group size. For example, assuming that the target scenes have about five people, we forecast that a system with 80 cameras can reach about 94% of accuracy with a 2cm threshold.

Comparison with varying resolutions: As an additional evaluation, we perform a similar experiment for different camera resolutions using the multiple HD cameras installed in our system. Among 31 HD cameras, we use 19 HD cameras installed on the same panels with VGA cameras[7]. To compose similar viewpoints, we choose the closest VGA cameras from the selected 19 HDs. Additionally, we generate 19 QVGA inputs (320 × 240 resolution) by resizing the selected VGA videos. Because the HD cameras are not perfectly synchronized with VGAs, we interpolate the result from HDs into the VGA time domain using the hardware sync data. The performance of a same number of HDs, VGAs, and QVGAs is shown as dashed lines in Figure 3.15. The result shows that the performance differences among them are marginal, although HD views have about 7 times more pixels than VGA views and about 27 times more pixels than QVGAs. The result demonstrates that the pose reconstruction performance of our method is marginally affected by the resolution changes compared to the changes by the number of views. Note that the integral of all pixels in the 19 HD views are equivalent to about 128 VGA views, and the result clearly shows that it is more advantageous to have more unique camera views rather than having higher resolutions, given a fixed pixel budget. The main reason underlying this finding is that dealing with occlusions is more crucial in interaction capture scenarios, and, in particular, higher resolution is not beneficial in our method since 2D joint localization accuracy is still limited by the 2D pose detector.

Comparison to multiple Kinects: We also compare our results with the result of multiple Kinects. Since Kinect with its accompanying SDK is one of the most commonly used sensors for markerless motion capture in various communities, using multiple Kinects can be considered as an option to handle severe occlusions for interaction capture. However, how to fuse multiple Kinect cues is not straightforward, and, thus, we naively fuse them as follows. We first generate 3D skeletal proposals from all individual Kinects, and simply find the best candidate closest to our ground truth data in Euclidean space, assuming that an Oracle chooses the best one given the GT data. This can be considered an upper bound of a naive multiple Kinects method. Since the keypoint locations of the Kinects are not identical to the skeletons of our method, for a fair comparison, we adjust the Kinect skeletons by finding an offset vector from each Kinect node toward our node of GT's skeleton in a person-centric coordinate system. As shown in Figure 3.15, the results of the *Oracle* Kinects is limited, showing less than 80% accuracy at a 5cm threshold.

3.5 Conclusions

We present the Panoptic Studio and an interaction capture method that leverages a large number of views. To demonstrate the performance of our method, we collect a large scale social interaction dataset, and produce compelling motion capture results on it. In particular, we empirically find that having a larger number of views is more beneficial than having

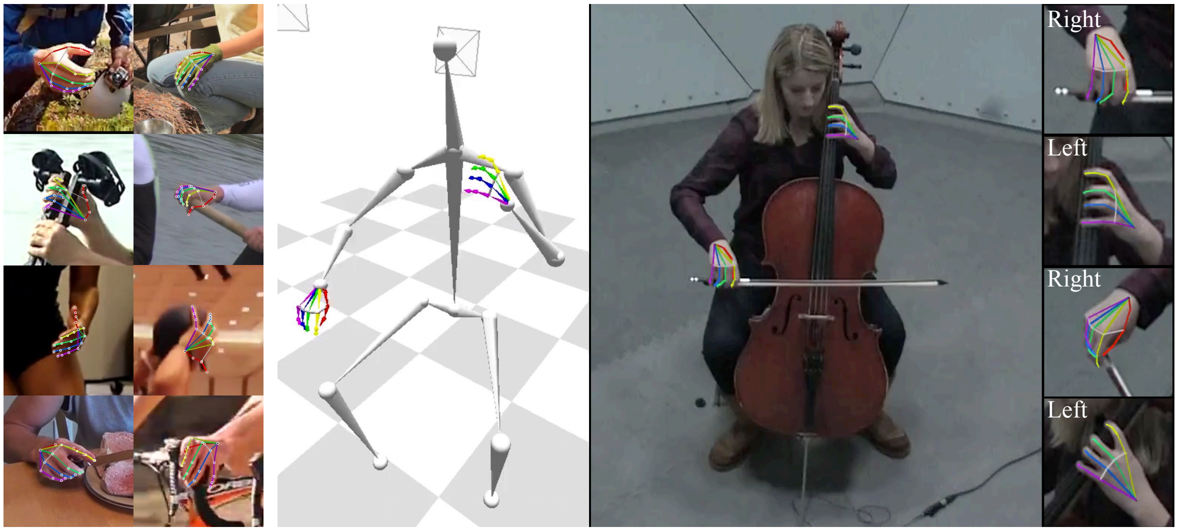
[7] We have 20 HD cameras installed on the same panels with VGAs, but we lost 1 HD camera due to the hardware failure during the capture.

higher resolution views for social interaction capture. Our quantitative comparison of various number and type of cameras can be used as a meaningful resource to design follow-up multiview systems to estimate the required number of views to achieve a desired accuracy. Our method shows its advantages in the social interaction capture scenario by reconstructing subjects of diverse appearance, body sizes, and body topology for a long term without error accumulation issues.

This first stage of our method is performed without considering any temporal cues. The advantages of this are that the method can easily handle a varying number of people, there is no need to impose priors on the motion or skeletons, and the bulk of the computation is easily parallelized across frames. In Chapters 5 and 6 we will see how these per-frame measurements are integrated across time to produce a smooth and consistent estimate of the motion. In the following chapter, we describe how we augment the estimated skeletons with fine-grained hand pose and facial details.

Chapter 4.

Fine-grained Keypoint Detection using Multiview Bootstrapping



(a) 2D Detections (b) Multiview 3D Triangulations (c) Reprojections

Figure 4.1: We estimate fine-grained hand motion by training 2D keypoint detectors for the joints of the hand. (a) 2D hand keypoint detection results on web images. (b) Multiview 3D triangulation of 2D keypoint detections in the Panoptic Studio. (c) Reprojections of the 3D points.

4.1 Introduction

The markerless motion capture system presented in the previous chapter provides only a coarse measurement of a participant's pose and movement. In fact, none of the markerless motion capture methods we reviewed included degrees of freedom for joints other than those of the major limbs [Shotton et al. 2011; Baak et al. 2011; Liu et al. 2013b; Amin et al. 2013; Burenius et al. 2013; Belagiannis et al. 2014; Elhayek et al. 2016; Rhodin et al. 2016]). While many approaches to facial keypoint localization exist, there are no markerless hand motion capture solutions that work across multiple people. However, the precise pose of the hand is often the main discriminative feature between certain actions, ranging from goal-driven manipulation tasks (e.g., playing an instrument, handling tools) to conversational gestures. Interestingly, the extent to which gestures contribute to communication is disputed [Krauss 1998], but it is undeniable that when people talk, they gesture [Goldin-Meadow and Wagner Alibali 2013]. Precisely because of our lack of understanding of the role that gestures play in communication, we contend that they need to be measured as part of a social interaction capture system.

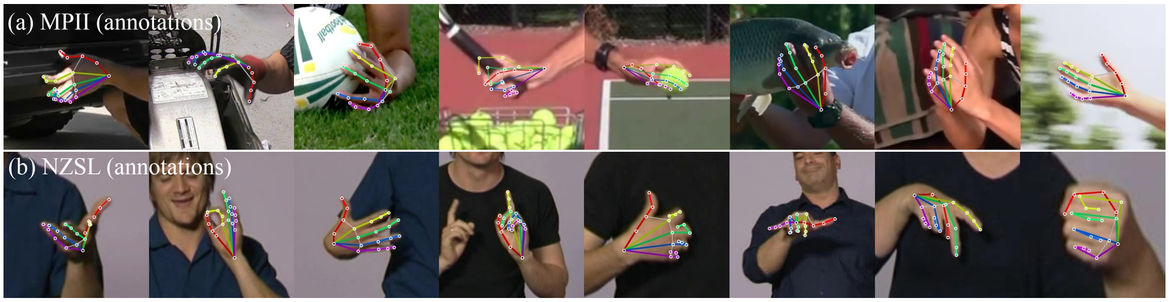


Figure 4.2: We manually annotated 21 hand keypoints (1 on the wrist and 4 per finger, see Fig. 4.7) on images from (a) MPII [Andriluka et al. 2014], and (b) NZSL [McKee et al.]. Hand joints are often occluded, making it difficult to estimate their location from a single viewpoint.

In this chapter, we develop the methods that make this possible within the Panoptic Studio. While we use the same method for both hand pose and facial keypoint detection, we focus on hand pose estimation as the more novel aspect of the work. Our approach to hand pose estimation and 3D facial keypoint estimation follows the same general principle as the method presented in the previous chapter: we triangulate 2D detections across many camera views. This approach requires 2D keypoint detectors capable of localizing the joints of the hand in RGB images, and similarly for facial keypoints.

Progress in keypoint detectors is driven by the availability of large, labeled datasets, for example, for facial landmark localization [Sagonas et al. 2016], pose estimation [Andriluka et al. 2014], or body part detection [Mathias et al. 2014]. However, large datasets of RGB images labeled with the joints of the hand do not exist, likely due in large part to the difficulty of annotating the joints of the hand. Besides being costly, manual keypoint annotations are difficult to get right: even with quality control, clear annotation protocols, and review sessions, accuracy varies greatly between annotators. The fundamental problem is that for keypoints that are occluded—and also for joints that are internal to the body, like the hips—the location is not more than an educated guess.

Our approach to building a large dataset of annotated training examples for hand keypoint detection is motivated by our own experiences manually annotating images: hand joints are very often occluded or self-occluded, and the position of certain joints is difficult to pinpoint (for example—look at the palm of your hand and try to guess where the knuckles are[1]). When we engaged annotators to build a labeled dataset, we encountered all of the problems described above (see Fig. 4.2).

To address these problems, we began by manually labeling multiview images. Using multiple views, we can accurately annotate occluded points as long as they can be triangulated

[1] Most people guess they are closer to the fingers than they actually are.

in other views. For internal joints, we can ensure that the annotations at least correspond to a consistent 3D location. Our bootstrapping approach is robust enough that we can also substitute manual annotations for an inaccurate initial detector—even one trained with only rendered data—and bootstrap the entire process creating additional labeled images with little to no manual intervention.

Because work to date in hand pose estimation has focused mainly on using depth data, there is currently no dataset for hand joint detection in general purpose RGB images. In this chapter, we introduce such a dataset and associated performance metrics. Additionally, we release a large dataset of automatically annotated 3D hand joint positions in a variety of scenarios, including playing musical instruments, using tools, and interacting socially.

4.2 Multiview Bootstrapped Training

A keypoint detector $d(\cdot)$ maps from a cropped input image patch $\mathbf{I} \in \mathbb{R}^{w \times h \times 3}$ to P keypoint locations $\mathbf{x}_p \in \mathbb{R}^2$, each with an associated detection confidence c_p:

$$d(\mathbf{I}) \mapsto \{(\mathbf{x}_p, c_p) \text{ for } p \in [1 \dots P]\}. \tag{4.1}$$

Each point p corresponds to a different landmark (e.g., the tip of the thumb, the tip of the index finger, see Fig. 4.7a), and we assume that only a single instance of the object is visible in $\mathbf{I}$. The detector is trained on images with corresponding keypoint annotations, $\left(\mathbf{I}^f, \{\mathbf{y}_p^f\}\right)$, where f denotes a particular image frame, and the set $\{\mathbf{y}_p^f \in \mathbb{R}^2\}$ includes all labeled keypoints for the image $\mathbf{I}^f$. An initial training set $\mathcal{T}_0$ having N_0 training pairs (e.g., Fig. 4.2),

$$\mathcal{T}_0 := \left\{\left(\mathbf{I}^f, \{\mathbf{y}_p^f\}\right) \text{ for } f \in [1 \dots N_0]\right\}, \tag{4.2}$$

is used to train a detector d_0 with, e.g., stochastic gradient descent,

$$d_0 \leftarrow \text{train}(\mathcal{T}_0). \tag{4.3}$$

Given an initial, single-image keypoint detector d_0 and a dataset of multiview unlabeled images, our objective is to use the detector to generate a new set of labeled images, $\mathcal{T}_1$, which can be used to train an *improved* detector, d_1, with the union of all available data:

$$d_1 \leftarrow \text{train}(\mathcal{T}_0 \cup \mathcal{T}_1). \tag{4.4}$$

Obviously, to improve upon the detector d_0, we need an external source of supervision to ensure $\mathcal{T}_1$ contains information not already present in $\mathcal{T}_0$. We propose to use multiview ge-

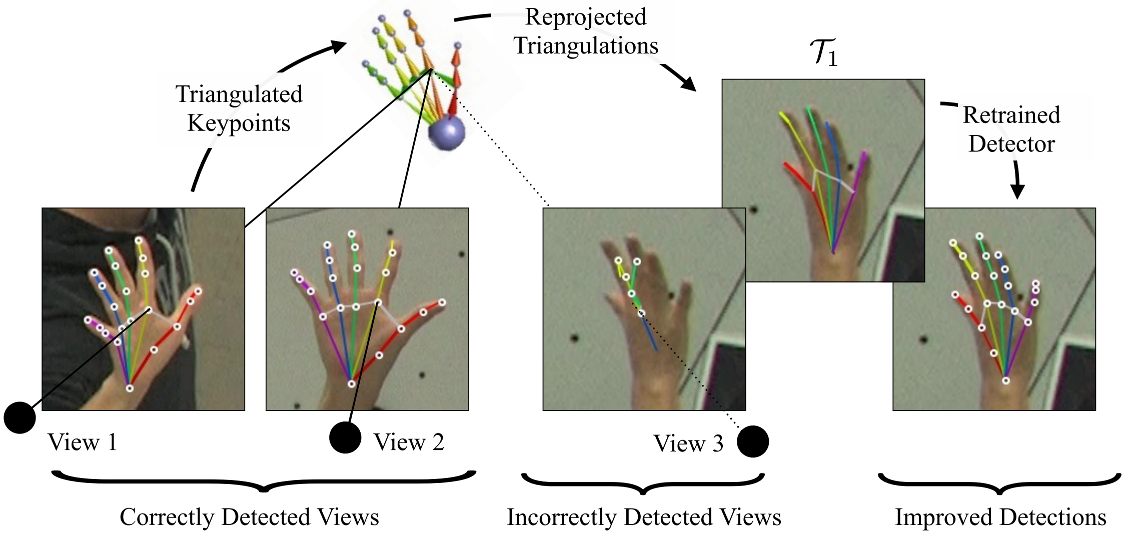


Figure 4.3: A multiview system provides several views of the hand where detection of the hand joints is easy. These are used to reconstruct the 3D position of these keypoints, which can be reprojected onto viewpoints where the detector failed, serving as additional training data for an improved detector that now works even on these difficult views.

ometry as this source. The key here is that detection is easier in some views than others: if a point is successfully localized in at least two views, the triangulated 3D position can be reprojected onto other images, providing a new 2D annotation for the views on which detection failed. This is shown in Fig. 4.3, where the detector succeeds on easy views (left) but fails on more challenging views. However, by triangulating the correctly detected viewpoints, we can generate training data particularly for those views in which the detector is currently failing.

The overall procedure for multiview bootstrapping is described in Algorithm 1, where we denote by $\{\mathbf{I}_v^f : v \in [1 \dots V], f \in, [1 \dots F]\}$ the input set of unlabeled multiview image frames, with v iterating over the V camera views, and f iterating over F distinct frames (i.e., time instants). There are three main parts to the process: (1) For every frame, the algorithm first runs the current detector on every camera view independently (Fig. 4.9a) and attempts to triangulate the point detections. (2) The set of frames is then sorted according to a score to select only correctly triangulated examples, and (3) the N-best are used to train a new detector by reprojecting the correctly triangulated points onto all views (Fig. 4.9b), producing approximately V training images for each of the N selected frames. The entire process can then be iterated with the newly trained detector (Fig. 4.9c). In the following section, we detail how we robustly triangulate noisy 2D detections into 3D and outline conditions under which triangulation can be used as supervision for learning. Then, we describe how to score and sort frames to ensure that incorrectly labeled frames are not selected for training.

Algorithm 1 Multiview Bootstrapping

```
**Inputs:**
•Unlabeled images: $\{\mathbf{I}_v^f \text{ for } v \in \text{views}, f \in \text{frames}\}$
•Keypoint detector: $d_0(\mathbf{I}) \mapsto \{(\mathbf{x}_p, c_p) \text{ for } p \in \text{points}\}$
•Labeled training data: $\mathcal{T}_0$
for iteration $i$ in 0 to $K$:
  1. Run current detector & triangulate:
    for every frame $f$:
      $(a)$ Run detector on all views:
        $\mathcal{D} \leftarrow \{d_i(\mathbf{I}_v^f) \text{ for } v \in \text{views}\}$
      $(b)$ Triangulate keypoints:
        $\{\mathbf{X}_p^f\} \leftarrow \text{RANSAC}(\mathcal{D}) \text{ for } p \in \text{points}$
  2. Score and sort frames:
    $[s_1 \dots s_F] \leftarrow \arg\text{sort}\left(\left[\text{score}(\{\mathbf{X}_p^f\}) \text{ for } f \in \text{frames}\right]\right)$ (see Eq. (4.15))
  3. Retrain with reprojections on $N$-best frames:
    $\mathcal{T}_{i+1} \leftarrow \{(\mathbf{I}_v^{s_n}, \mathcal{P}_v(\mathbf{X}^{s_n})) \text{ for } v \in \text{views}, n \in [1 \dots N]\}$ (see Eq. (4.16))
    $d_{i+1} \leftarrow \text{train}(\mathcal{T}_0 \cup \mathcal{T}_{i+1})$
**Outputs:** Improved detector $d_K(\cdot)$ and training set $\mathcal{T}_K$
```

4.2.1 3D Triangulation as Supervision

Given V views of an object in a particular frame f, we run the current detector d_i (trained on set $\mathcal{T}_i$) on each image $\mathbf{I}_v^f$, yielding a set $\mathcal{D}$ of 2D location candidates:

$$\mathcal{D} \leftarrow \{d(\mathbf{I}_v^f) \text{ for } v \in [1 \dots V]\}. \tag{4.5}$$

For each keypoint p, we have V detections $(\mathbf{x}_p^v, c_p^v)$, where $\mathbf{x}_p^v$ is the detected location of point p in view v and $c_p^v \in [0, 1]$ is a confidence measure (we omit the frame index for clarity). To robustly triangulate each point p into a 3D location, we use RANSAC [Fischler and Bolles 1981] on points with confidence above a detection threshold λ. Empirically, we find that λ=0.2 works well for our base detectors. Additionally, we use a σ=4 pixel reprojection error to accept inliers for RANSAC. Once we determine the set of inlier views for point p, denoted by $\mathcal{I}_p^f$, we minimize the reprojection error [Agarwal et al. 2009] to obtain the final triangulated position, i.e.,

$$\mathbf{X}_p^f = \arg\min_{\mathbf{X}} \sum_{v \in \mathcal{I}_p^f} ||\mathcal{P}_v(\mathbf{X}) - \mathbf{x}_p^v||_2^2, \tag{4.6}$$

with $\mathbf{X}_p^f \in \mathbb{R}^3$ the 3D triangulated keypoint p in frame f, and $\mathcal{P}_v(\mathbf{X}) \in \mathbb{R}^2$ denotes projection of 3D point $\mathbf{X}$ into view v. Given calibrated cameras, this 3D point can be reprojected into any view (e.g., those in which the detector failed) and serve as a new training label.

4.2.1.1 When does multiview supervision work?

In this section we derive two results that allow us to characterize the performance of multiview bootstrapping in terms of the number of cameras, the quality of a keypoint detector, and the thresholds used for triangulation:

Result 1. For RANSAC triangulation with n minimum inliers across V views, the probability of a false triangulation occurring among random points is approximated in Eq. (4.10).

Result 2. For multiview supervision with n minimum inliers across V views, the true and false positive rates of a given keypoint detector are approximated in Eqs. (4.11) and (4.12).

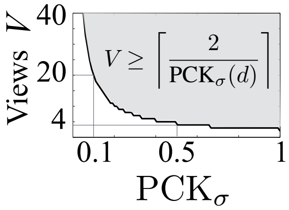
Let us first define the quality of a detector d_0 as its *Probability of Correct Keypoint* or PCK[2]: the probability that a predicted keypoint is within a distance threshold σ of its true location. For a particular keypoint p, denote this probability by $\mathrm{PCK}_\sigma^p(d_0)$, which we approximate empirically on a testing set $\mathcal{T}_t$ as

$$\mathrm{PCK}_\sigma^p(d_0) := \frac{1}{|\mathcal{T}_t|} \sum_{\mathbf{y}_p^f \in \mathcal{T}_t} I\left(||\mathbf{x}_p^f - \mathbf{y}_p^f||_2 < \sigma\right) \tag{4.7}$$

for $\mathbf{x}_p^f \in d_0(\mathbf{I}^f)$ the p-th keypoint prediction on image $\mathbf{I}^f$ and $\mathbf{y}_p^f$ its true location, with $I(\cdot)$ the indicator function.

Intuitively, if we consider *average* performance, we can say that given V views, we require a detector d with $\mathrm{PCK}_\sigma(d_0) \geq \frac{2}{V}$ for at least 2 views out of V to be correct, or equivalently,

$$V \geq \left\lceil \frac{2}{\mathrm{PCK}_\sigma(d)} \right\rceil,$$



for 2-view triangulation to succeed, where $\lceil\cdot\rceil$ is the ceiling function. This area is shown in the inset and provides a rule of thumb as to whether multiview supervision will work on average. In practice, this relationship between V and a detector's PCK is coarse because it relies on average performance, and that each view is uncorrelated. However, for many viewpoints, even if we have at least 2 inliers, the chances of also having 2 or more erroneous correspondences (among misdetections) *also* increases with V.

To quantify this in more detail, consider the probability that we triangulate incorrect detections by chance—i.e., that our supervision method accepts an incorrect triangulation as valid. For simplicity, we assume that the position of misdetections follows a uniform

[2] Alternatively *percentage* of correct keypoints.

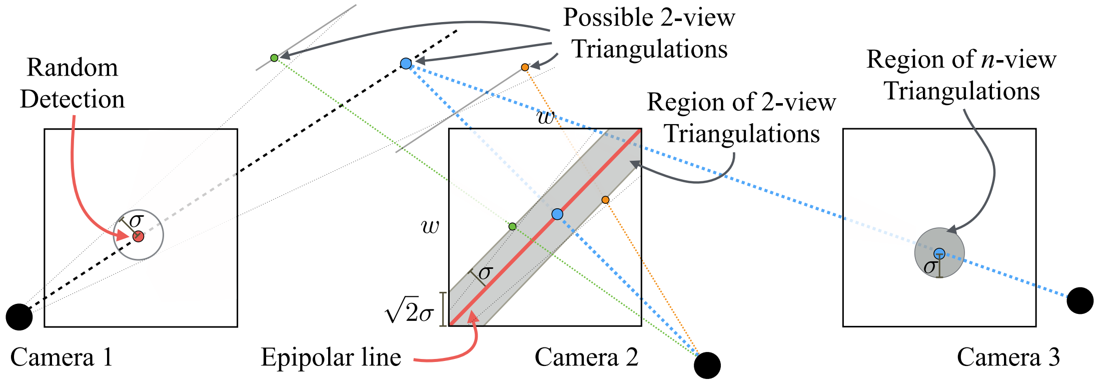


Figure 4.4: Approximate area within which a random detection will be successfully triangulated for 2 and 3+ views with an inlier threshold of σ pixels.

random distribution across the image. Then, for each RANSAC trial, we select 2 views for triangulation (the *hypothesis* set), triangulate the point, and determine inliers as those whose reprojection falls within a distance threshold of σ pixels (the *consensus* set). In the first selected view, a random detection may fall anywhere on the image plane. For the second view, the points that, when triangulated, fall within the σ threshold of the random detection in the first view are the set of pixels within σ of the corresponding epipolar line (with known extrinsic camera calibration). We assume an image size of $w \times w$, well-centered and tightly cropped around the 3D region of interest, and distant enough for a weak-perspective approximation[3], as illustrated in Fig. 4.4. We first consider the worst case as an upper bound, corresponding to the largest area of chance triangulation (with the epipolar line along the diagonal). The probability that a random point on the second view falls within the shaded area is the ratio of areas,

$$q_2 = \frac{w^2 - (w - \sqrt{2}\sigma)^2}{w^2} = 1 - \left(1 - \sqrt{2}\frac{\sigma}{w}\right)^2 \underbrace{\approxeq 2\sqrt{2}\frac{\sigma}{w}}_{\text{if } \frac{\sigma}{w} < 0.1}, \tag{4.8}$$

which we can identify as the probability of generating a false hypothesis for a normalized inlier threshold of σ/w. For two cameras, using $\sigma = 9$ with $w = 368$, the normalized distance threshold is approximately $\sigma/w = 0.025$. Then, $q_2 \approxeq 0.07$, or around 7% triangulations for a completely random detector. For 1 minute of video at 30 frames per second (and for a single keypoint) this would result in about 126 frames out of 1800 containing chance triangulations. This worst-case epipolar line will not be observed on most occasions, but approximating the area more precisely requires knowledge of the specific camera configuration. A relevant case is that of a horizontal scanline in a rectified stereo pair, in which case $q_2 = 2\frac{\sigma}{w}$. For random[4]

[3] This is the case for the Panoptic Studio when imaging cropped bodies, hands, or faces. Generally, this is also true for setups using cameras sharing similar intrinsics and imaging a small enough region.

[4] The average length of random lines bounded by a unit square is approximately 0.869 [Weisstein 2016].

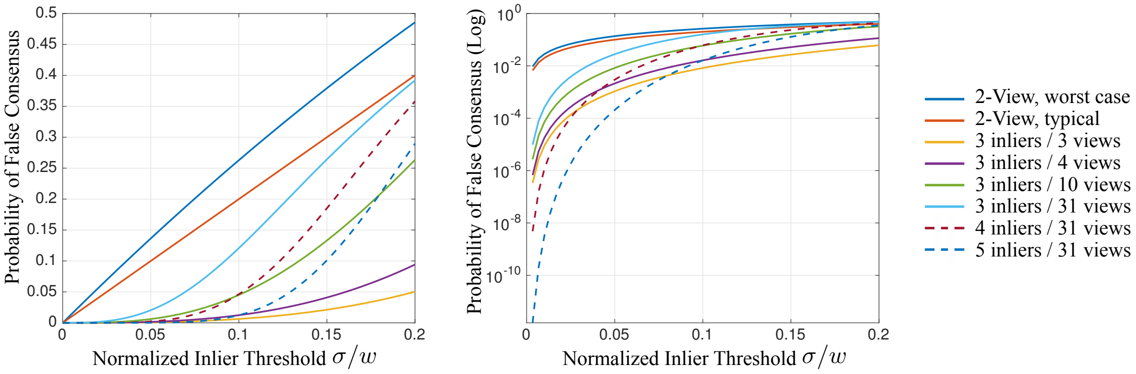


Figure 4.5: Probability (left) and log-probability (right) of a false RANSAC consensus acceptance for a completely random detector. Different multiview configurations are shown, varying the total number of views V and the minimum number of inliers n for a detection to be accepted.

epipolar lines, the probability is even lower, on average $q_2 \approxeq 1.74\frac{\sigma}{w}$.

More generally, we can require that the RANSAC consensus set contain at least n inliers to accept a triangulation for a multiview setup with V cameras. These additional $n-2$ inliers (we already have 2 from the first pair of views) can occur among any of the remaining $V-2$ views. For these additional viewpoints, the probability that a random detection *also* passes the inlier test (and is therefore included in the consensus set) requires that the detection fall within a distance σ of the reprojection of the triangulated 3D point from the original 2 views (see Fig. 4.4). Using the area ratio, this corresponds to a probability $q_c = \pi\left(\frac{\sigma}{w}\right)^2$ or approximately 4×10^{-4} for our example. Therefore, using a third view for verification provides a *much* more powerful test than merely respecting the epipolar constraint. For a typical 3-view configuration, the probability that a random RANSAC trial will find 3 inliers among random detections is then approximately $q_2 q_c = 2\pi\left(\frac{\sigma}{w}\right)^3 \approxeq 1{\times}10^{-4}$, compared to $2\frac{\sigma}{w} \approxeq 5{\times}10^{-2}$ for 2-view triangulation. This is shown in Fig. 4.6 for different values of $\frac{\sigma}{w}$.

If we assume independence among the set of $V-2$ remaining viewpoints, the probability c is the same for all views. Let X be a random variable representing the number of inliers among the $V-2$ additional views, then, X follows a binomial distribution, $X \sim \mathrm{B}(V{-}2, q_c)$, with parameters $V-2$ and c. The probability that there are at least $n{-}2$ inliers (in addition to the 2 inliers from the first two views) is $\Pr(X \geq n{-}2) = 1 - \mathrm{F}(n{-}3; V{-}2, q_c)$, with for $V \geq 3$ and $n \geq 3$, where $\mathrm{F}(\cdot)$ is the Cumulative Distribution Function (CDF) of the binomial distribution[5] with parameters $V{-}2$ and c. Finally, we can write that for any RANSAC trial (i.e., we pick 2 views at random, triangulate the point, and count the number of inliers among

[5] For $X \sim \mathrm{B}(N, p)$, then $\mathrm{F}(k; N, p) = \Pr(X \leq k) = \sum_{i=0}^{\lfloor k \rfloor} \binom{N}{i} p^i (1-p)^{N-i}$.

all views), the probability of finding n random inliers among V views is

$$q_n = q_2 \cdot (1 - \mathrm{F}(n{-}3; V{-}2, c)), \tag{4.9}$$

with $n \geq 3$. We call this the probability of false consensus, which is shown in Fig. 4.5 for a variety of multiview setups and minimum number of inliers. For our 31 camera setup with at least n=4 inliers and $\sigma/w = 0.025$, then $q_n \approxeq 2.5{\times}10^{-6}$. Note that this rate corresponds to a single RANSAC trial. For the 31 camera views, there are (V choose 2) or 465 distinct camera pairs. Assuming we exhaustively test each possibility (and that each of these trials is independent), the probability that at least one False Triangulation (FT) occurs among the V views is then

$$\mathrm{FT}_n \approxeq 1 - \mathrm{F}\left(0; \cdot \binom{V}{2}, q_n\right). \tag{4.10}$$

For a purely random detector $\mathrm{FT}_4 \approxeq 1{\times}10^{-3}$ for our example, or about 1 out of every 1000 frames will triangulate a false detection for one keypoint.

For a detector d that is not purely random but has a certain $\mathrm{PCK}^p_\sigma(d)$, we will assume that if there are at least n correct detections among the V views (i.e., at least the minimum number of inliers), then the true detection will be identified correctly. In all other cases, we assume the detector behaves as a purely random detector with false positive rate FT_n. We can now approximate the probability of getting a True Positive (TP) or False Positive (FP) on any given frame, for the detector d when using V views, inlier threshold σ, and n minimum inliers for a single keypoint p:

$$\mathrm{TP}_p(d) = 1 - \mathrm{F}(n-1; V, \mathrm{PCK}^p_\sigma(d)) \tag{4.11}$$

$$\mathrm{FP}_p(d) = (1 - \mathrm{TP}_p(d)) \cdot \mathrm{FT}_n. \tag{4.12}$$

For a complex object with a total of P keypoints, if we require that all keypoints $p \in [1 \ldots P]$ be correct to accept a frame and assume that $\mathrm{PCK}^p_\sigma(d)$ is the same for all keypoints, then

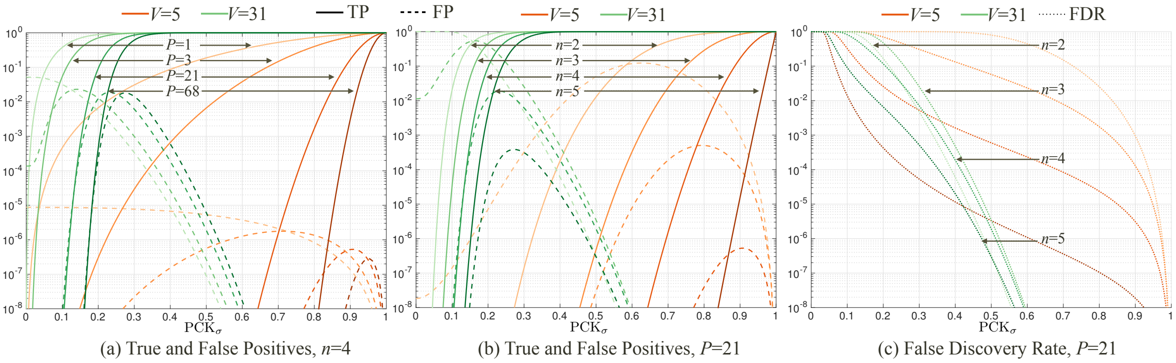
$$\mathrm{TP}(d) = (\mathrm{TP}_p(d))^P \tag{4.13}$$

$$\mathrm{FP}(d) = \sum_{i=1}^{P} \mathrm{f}\left(P - i; P, \mathrm{TP}_p(d)\right) \cdot (\mathrm{FT}_n)^i, \tag{4.14}$$

where $\mathrm{f}(\cdot)$ is the Probability Density Function (PDF) of the binomial distribution[6].

These quantities are shown in Fig. 4.6a and b, again for $\sigma/w = 0.025$. In practice, it

[6] For $X \sim \mathrm{B}(N, p)$, then $\mathrm{f}(k; N, p) = \Pr(X{=}k) = \binom{N}{k} p^k (1-p)^{N-k}$.



(a) True and False Positives, n=4

(b) True and False Positives, P=21

(c) False Discovery Rate, P=21

Figure 4.6: (a) TP and FP probabilities (log-scale) for different values of PCK_σ, and different numbers of keypoints P, for $n = 4$ inliers. In orange, a setup with 5 cameras ($V = 5$) and in green, a setup as the one we use, with $V = 31$. With $P = 21$, (b) TP and FP for different values of n, and (c) False discovery rate, $\frac{\mathrm{FP}}{\mathrm{TP}+\mathrm{FP}}$.

is not performance on average that we require: We assume that we have a (large) set of F frames from which we will select N training examples, with $N \ll F$. Therefore, we only need to succeed on N out of F frames to obtain the required training examples. From the analysis, we want to ensure that $\mathrm{TP}(d) \cdot F > N$. Simultaneously, we must ensure that the number of random triangulations remains low. This corresponds to the False Discovery Rate (FDR), $\mathrm{FDR}(d) = \frac{\mathrm{FP}(d)}{\mathrm{TP}(d)+\mathrm{FP}(d)}$, i.e., the proportion of frames we triangulate that are in fact random triangulations. This is shown in Fig. 4.6c for the same conditions in Fig. 4.6c. It is therefore useful to think of multiview supervision in two steps: multiview triangulation (at the end of RANSAC) will yield an expected value of $N_1 = (\mathrm{TP}(d) + \mathrm{FP}(d)) \cdot F$ frames, which we will then filter for outliers to obtain on average $N \approxeq N_1 \cdot (1 - \mathrm{FDR}(d))$ correctly triangulated training examples. This selection process will be described in the following section.

Note that requiring that all P keypoints to be correct (e.g., P=21 for the hand joints) is a very stringent condition—for PCK values around 0.1 to 0.3, requiring all keypoints to be correct can spell the difference between succeeding on 1/10 frames or virtually never. To some extent, this can be controlled by decreasing the number of inliers, but the cost in the number of false discoveries for setups with a large number of cameras (e.g., $V = 31$) can be prohibitive. Therefore, for low PCK_σ, we allow partial detections—this way, the inlier threshold per-point can be high, while still obtaining enough partially labeled frames to train a detector.

Hand-specific considerations. For the particular case of hand joints, two small modifications improve robustness with respect to the analysis above. First, our detector provides a confidence value which can be used to prune out some percentage of random detections. Second, errors in finger detections are correlated: e.g., if the knuckle is incorrectly localized, then dependent joints in the kinematic chain—the inter-phalangeal joints and the tip of the finger—are unlikely to be correct. It is therefore more robust to reconstruct entire fingers simultaneously. More specifically, we triangulate all landmarks of each finger (4 points) at a time, and use the average reprojection error of all 4 points to determine RANSAC inliers. While this reduces the number of correctly triangulated points (because the entire finger needs to be correct in any given view), it also greatly decreases incorrect triangulations due to false positive detections. This is a desireable trade-off: it is much more important to ensure that there are no false positives (so that we do not train with incorrect labels) than to use all frames (because collecting unlabeled data is cheap).

4.2.2 Training Sample Selection

It is crucial that we do not include erroneously labeled frames as training data, especially if we iterate the procedure, as subsequent iterations will fail *consistently* across views—a failure which cannot be detected using geometry.

From Fig. 4.6c, even when using n=5 minimum inliers we cannot guarantee that there will be no false triangulations. For example, for n=5 with a PCK_σ=0.2 (a reasonable value for $\sigma/w \approxeq 0.025$), we can expect around 1 out of every 100 triangulated frames to contain at least 1 incorrect keypoint (out of the full 21). We take a conservative approach and try to automatically remove as many questionable triangulations as possible. We filter out entire frames using a number of heuristics: (1) average number of inliers, (2) average detection confidence, (3) difference of per-point velocity with median velocity between two video frames, (4) anthropomorphic limits on joint lengths[7], and (5) complete occlusion, as determined by camera ray intersection with potential occluders. Additionally, we require at least 3 inliers for any point to be considered valid.

Because our input is video and consecutive frames are highly correlated, we subsample the frames temporally. Instead of uniform sampling, we pick the "best" frame for every window of W frames (e.g., W=15 or W=30), defining the "best" as that frame with maximum sum of detection confidences for the inliers, i.e.,

$$\text{score}(\{\mathbf{X}_p^f\}) = \sum_{p\in[1\ldots P]} \sum_{v\in\mathcal{I}_p^f} c_p^v. \tag{4.15}$$

We sort all the remaining frames in descending order according to their score, to obtain an ordered sequence of frames, $[s_1, s_2, \ldots s_M]$, where M is the number of frames that passed all filtering criteria and s_i is the ordered frame index. We use the N-best frames according

[7] We use thresholds larger than the maximum bone lengths given in the survey of Greiner [1991], specifically 15 cm for the metacarpals, 9 cm for the proximal phalanges, and 5 cm for the remaining bones.

to this order to define a new set of training image-keypoint pairs for the next iteration $i{+}1$ detector,

$$\mathcal{T}_{i+1} = \left\{ \left(\mathbf{I}_v^{s_n}, \{ \mathcal{P}_v(\mathbf{X}_p^{s_n}) : v{\in}\mathcal{I}_p^{s_n},\, p{\in}[1 \ldots P] \} \right) \text{ for } n{\in}[1 \ldots N] \right\}, \tag{4.16}$$

where $\mathcal{P}_v(\mathbf{X}_p^{s_n})$ denotes projection of point p for frame index s_n into view v, and we aim for $\frac{N}{M}$ to be $\frac{1}{10}$ to $\frac{1}{2}$ in our experiments, yielding about 100 frames for every 3 minutes of video. Note that 100 frames yields roughly $100 \cdot \frac{V}{2} \approxeq 1500$ training samples, one for each unoccluded viewpoint.

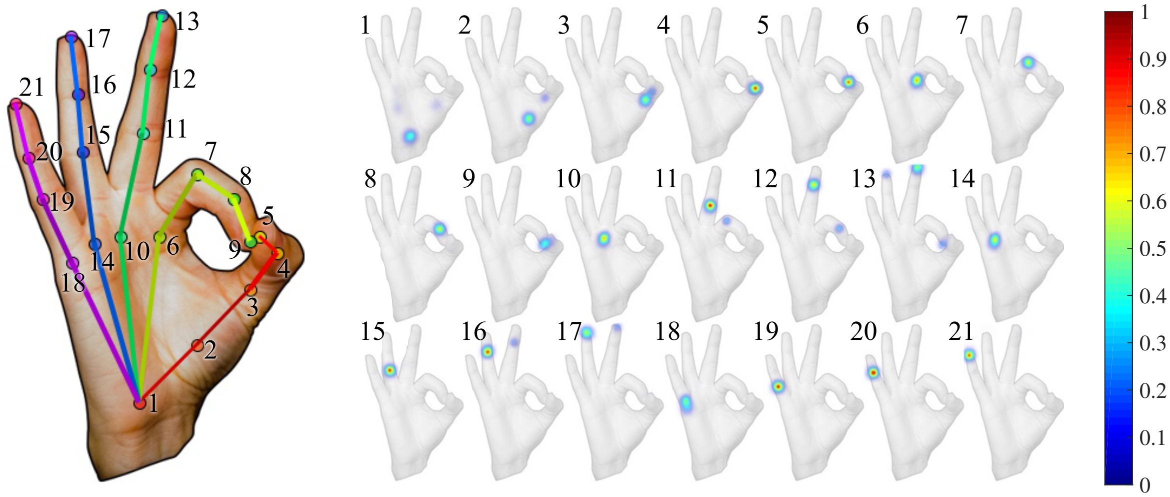
Finally, we manually verify that there are no obvious errors in the frames to be used as training data, and train a new detector $d_{i+1} {\leftarrow} \text{train}(\mathcal{T}_0 \cup \mathcal{T}_{i+1})$. While visual inspection of the training set may seem onerous, in our experience this is the least time consuming part of the training process. It typically takes us one or two minutes to inspect the top 100 frames. Crowdsourcing such a verification step for continuous label generation is an interesting future direction, as verification is easier than annotation.

4.3 Hand Keypoint Detection in Single Images

We follow the architecture of Convolutional Pose Machines (CPMs), proposed for body pose estimation by Wei et al. [2016], with some modifications. The idea behind CPMs (as with [Ramakrishna et al. 2014; Tompson et al. 2014b,a]) is to predict a confidence map for each keypoint, representing the keypoint's location as a Gaussian centered at the true position. The predicted confidence map corresponds to the size of the input image patch, and the final position for each keypoint is obtained by finding the maximum peak in the confidence map (see Fig. 4.7b). A sequential prediction framework improves the confidence maps over several stages using intermediate supervision.

4.3.1 Keypoint Detection via Confidence Maps

The convolutional prediction architecture we use is shown in Fig. 4.8. In contrast to Wei et al., we use the convolutional stages of a pre-initialized VGG-19 network [Simonyan and Zisserman 2014] up to conv4_4 as a feature extractor, with two additional convolutions producing 128-channel features $\mathbf{F}$. For an input image patch of size $w \times h$, the resulting size of the feature map $\mathbf{F}$ is $w' \times h'$ by 128 channels, with $w' = \frac{w}{8}$ and $h' = \frac{h}{8}$. There are no additional pooling or downsampling stages, so the final stride of the network is also 8. This is followed by a prediction stage that produces confidence or *score* maps, symbolized by $\mathbf{S}^1$, the set of P confidence maps $\mathbf{S}_p^1 \in \mathbb{R}^{w' \times h'}$ for each keypoint p, i.e., $\mathbf{S}^1 = \{\mathbf{S}_1^1 \ldots \mathbf{S}_P^1\}$. Each stage



(a) 21 Detected Keypoints

(b) Confidence maps of 21 Keypoints

Figure 4.7: (a) Input image with 21 detected keypoints. (b) Confidence maps produced by our detector, visualized as a "jet" colormap overlaid on the desaturated input.

after the first takes as input the score maps from the previous stage, $\mathbf{S}^{t-1}$, concatenated with the image features $\mathbf{F}$, and produces P new score maps $\mathbf{S}^t$, one for each keypoint, as detailed in Fig. 4.8. We use 6 sequential prediction stages, taking the output at the final stage, $\mathbf{S}^6$. We resize these maps to the original patch size $(w \times h)$ using bicubic resampling, and extract each keypoint location as the pixel with maximum confidence in its respective map.

We find that this modified architecture is easier and faster to train and provides similar performance once converged. We also modify the loss function f to be a weighted L2 loss to be able to handle missing data, where the weights are set to zero if annotations for a keypoint are missing (e.g., if triangulation for that point fails). Specifically,

$$f(\mathbf{S}^t) = \sum_{p=1}^{P} w_p ||\mathbf{S}_p^t - \mathbf{S}_p^*||_F^2, \tag{4.17}$$

where $\mathbf{S}_p^*$ is the ideal confidence map for the keypoint p (see [Wei et al. 2016]) and w_p is 0 if the keypoint is missing and 1 otherwise. This loss function is applied to each of the $\mathbf{S}^t$ intermediate confidence maps to reduce the effect of vanishing gradients.

4.3.2 Hand Bounding Box Detection

Our keypoint detector assumes that the input image patch $\mathbf{I} \in \mathbb{R}^{w \times h \times 3}$ is a crop around the right hand. This is an important detail: to use the keypoint detector in any practical situation, we need a way to generate this bounding box. We directly use the body pose

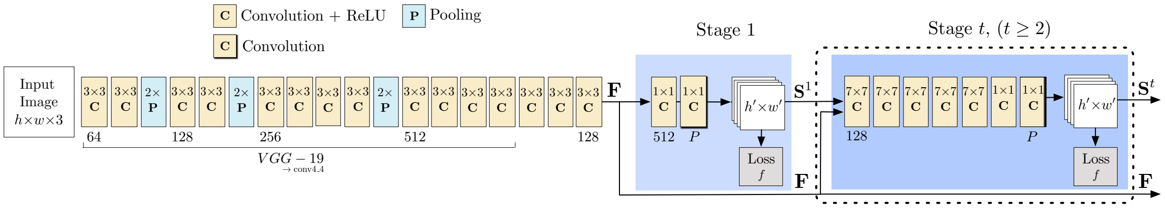


Figure 4.8: Network architecture. The kernel size is shown inside the layer, and the number of channels below, but only when they differ from those of the previous layer.

estimation model from Wei et al. [2016], and use the wrist and elbow position to approximate the hand location, assuming the hand extends 0.15 times the length of the forearm in the same direction.

During training, we crop a square patch of size $2.2B$, where B is the maximum dimension of the tightest bounding box enclosing all hand joints (see Fig. 4.2 for example crops of this size). At test time, we approximate $B = 0.7H$ where H is the length of the head "joint" (head-top to mid-point between shoulders). This square patch is resized to $w = 368$ and $h = 368$, which serves as input to the network. To process left hands, we flip the image left-to-right and treat it as a right hand.

4.4 Evaluation

In this section, we evaluate the improvement in 2D keypoint detection performance across iterations of our bootstrapping approach. We show results on images from the Panoptic Studio (Fig. 4.9) as well as web images from a dataset we collected (Sect. 4.4.1). We quantitatively evaluate performance using PCK for general 2D images as well as 3D triangulation for sequences in the Studio (Sect. 4.4.2), and compare accuracy with respect to a recent depth-based method (Sect. 4.4.3). Finally, we show qualitative results of multiview hand pose reconstruction (Sect. 4.4.4), as well as qualitative results applying the same method to facial keypoint detection. The supplementary video (link) shows many additional qualitative detection and 3D hand pose reconstruction results.

4.4.1 Datasets for RGB Hand Keypoint Detection

None of the available hand pose estimation datasets we reviewed suited our target use case: general, in-the-wild images containing everyday hand gestures and activities. We therefore chose to manually annotate two publicly available image sets: (1) The MPII Human Pose dataset [Andriluka et al. 2014], which contains images extracted from YouTube videos ex-

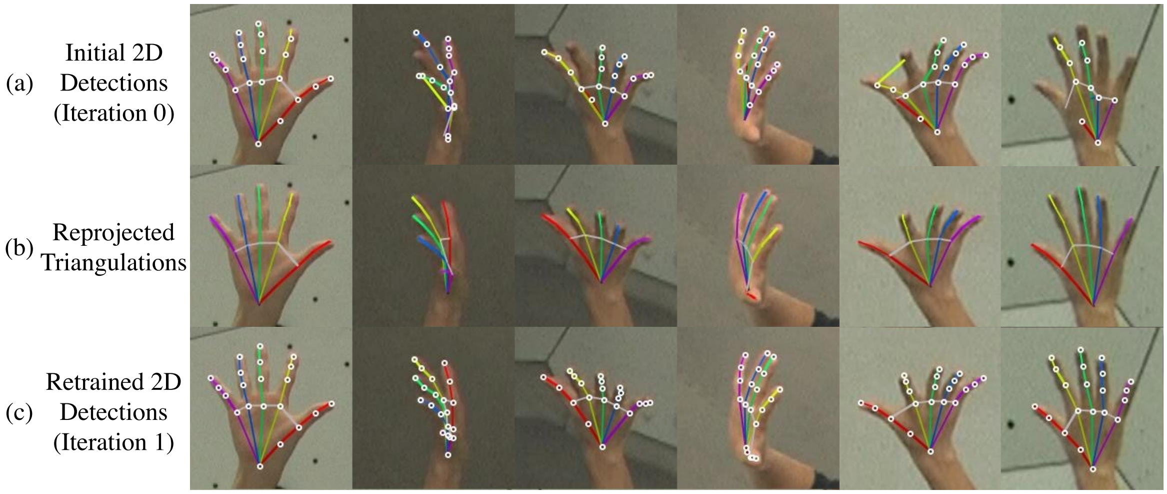

(a) Initial 2D Detections (Iteration 0)
(b) Reprojected Triangulations
(c) Retrained 2D Detections (Iteration 1)

Figure 4.9: Improvement of detections across iterations. (a) Noisy 2D keypoint detections. (b) Reprojected 3D triangulations are used as labeled data to train an improved 2D detector (c) Improved 2D keypoint detections.

plicitly collected to reflect every-day human activities, and (2) Images from the New Zealand Sign Language (NZSL) Exercises of the Victoria University of Wellington [McKee et al.], in which several people use NZSL to tell stories. We chose the latter because it contains a variety of hand poses as might be found in conversation (less common in MPII). Figure 4.2 shows a selection of images with manual annotations from both sets. To date, we have collected annotations for 1300 hands on the MPII set and 1500 on NZSL, which we split 70/30 into training (2000 hands) and testing (800) sets.

4.4.2 Improvement with Multiview Bootstrapping

We evaluate multiview bootstrapping by applying Algorithm 1 on three initial detectors. All three detectors follow the architecture described in Sect. 4.3, but are trained on 3 different sets of initial training data $\mathcal{T}_0$: (1) "Render": a set of synthetically rendered[8] images of hands, totalling around 11000 examples, (2) "Manual": the manual annotations in the MPII and NZSL training sets described above, and (3) "Mix": the combination of rendered data and manual annotations. For multiview bootstrapping, we use images from the Panoptic Studio dataset [Joo et al. 2015]. In particular, we use a subset of 31 HD camera views, and 4 sequences which contain hand motions, and we use the provided 3D body pose [Joo et al. 2015] to estimate occlusions and bounding boxes for hand detection. When performing bootstrapping iterations, we discard frames with an average number of inliers < 5 or an average reprojection error > 5. Figure 4.9 shows a frame during the first two iterations

[8] We use two renderers, UnrealEngine 4 and a simple raytracer. Characters in UnrealEngine are posed by Mixamo; for the raytracer, we randomly sample poses.

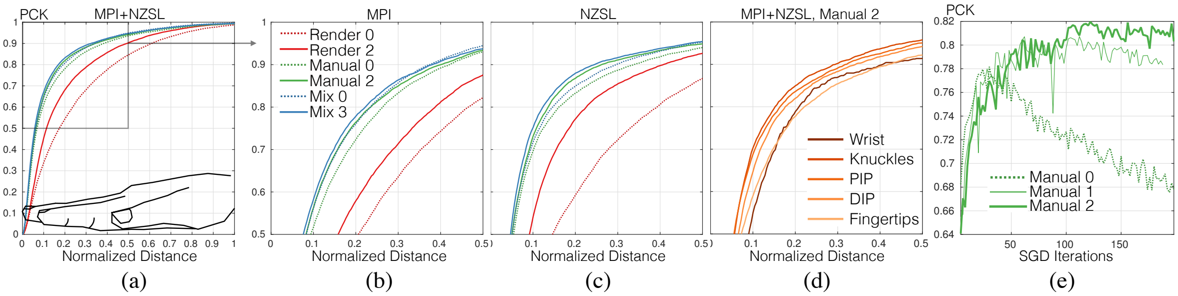


Figure 4.10: Improvement in PCK curves across multiview bootstrapping iterations for different initial training sets, "Render", "Manual" and "Mix". The iteration number is appended to the name. (a-c) PCK curves on MPII+NZSL testing images, MPII only, and NZSL only, for two different bootstrapping iterations of each model. (d) PCK for different types of hand joints. (e) Evolution of testing set PCK with SGD training iterations, for 3 bootstrapping iterations.

for the "Render" detector. For no iteration did we have to manually discard more than 15 incorrectly labeled frames.

PCK. We measure performance as PCK curves averaged over all keypoints, and we evaluate on the testing set of combined MPII and NZSL images (800 hands). This is shown in Figure 4.10a, where we append the bootstrapping iteration to the name of the model, e.g., "Manual 1" is the model trained on $\mathcal{T}_0 \cup \mathcal{T}_1$. The PCK curves are plotted by varying the accuracy threshold σ in Eq. (4.7); this parameter is shown on the x-axis. We measure σ as a normalized distance, where pixel distances in each example are normalized by the maximum length of the bounding box enclosing all annotated points. Unsurprisingly, the model trained exclusively on rendered data performs worst, but has the most to gain from bootstrapping with real training data. In Fig. 4.10b and c, we can see that the datasets reflect two levels of difficulty: MPII images vary widely in quality, resolution, and hand appearance, containing many types of occluders, hand-object interactions (e.g., sports, gardening), self-touching (e.g., resting head on hands), as well as hand-hand interactions, as shown in Fig. 4.12a. The NZSL set by contrast is fairly homogeneous, containing the upper-body of people looking directly at the camera and explicitly making visible gestures to communicate (Fig. 4.12b). Additionally, we study performance for different types of joints in Fig. 4.10d. These are ordered from closest to the wrist to farthest[9], an order which also corresponds to their difficulty. Finally, we show how multiview bootstrapping can help prevent overfitting in Fig. 4.10e, especially for small initial sets $\mathcal{T}_0$.

Robustness to View Angle. We quantify improvement in the detector's robustness to different viewing angles by measuring the percentage of outliers during 3D reconstruction. Because we have no manual annotations, we visually inspect our best 3D reconstruction result

[9] PIP and DIP are the Proximal and Distal Inter-phalangeal joints, respectively.

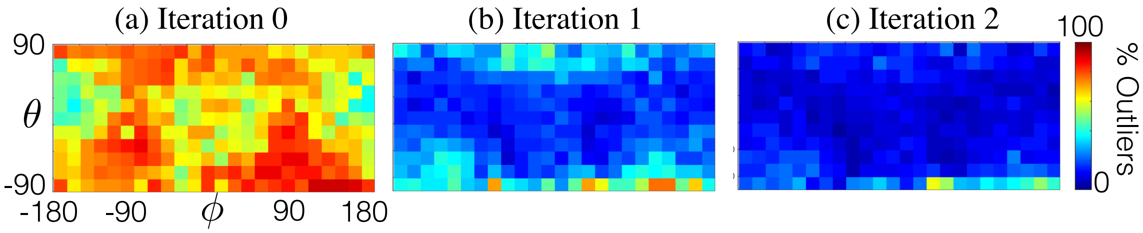


Figure 4.11: Robustness to view angle. We show percentage of outliers as a heatmap for each viewing angle, where azimuth ϕ is along the X axis and elevation θ along the Y axis.

to select correctly reconstructed frames as ground truth. We define view angle as azimuth ϕ and elevation θ w.r.t a fixed hand at the origin, with X pointing away from the palm at a right angle with the knuckles. Intuitively, angles with $\phi=\{-180, 0, 180\}$ (viewing the palm or backhand face-on) are easier because there is less self-occlusion. At $\phi=\{-90, 90\}$, we are viewing the hand from the side, from thumb to pinky or vice-versa, resulting in more occlusion. Similarly, at $\theta=\{90, -90\}$ the viewing angle is from fingertips to wrist, and vice-versa; these are the most difficult viewpoints. We compare the first two iterations of the "Mix" detector, which quickly becomes robust to view diversity. We plot this as a heatmap, where we bin hand detections on a grid using each example's azimuth and elevation. All examples falling into a particular bin are used to compute the percentage of outliers.

4.4.3 Comparison to Depth-based Methods

We quantify the performance of our method on a publicly available dataset from Tzionas et al. [2016]. Although there exist several datasets often used to evaluate depth-based methods, many of them do not have corresponding RGB images, or their annotations are only valid for depth images. Datasets with RGB images and manual annotations that can be accurately localized are rare; the dataset from [Tzionas et al. 2016] is the best match to quantify our method[10]. We run the 2D keypoint detector "Mix 3" on the RGB images of the dataset. The sequences include single-hand motion, hand-hand interaction, and hand-object interaction. To allow direct comparison with [Tzionas et al. 2016], we use average pixel errors in the location of the keypoints provided, as shown in Table 4.1. Note that the method of [Tzionas et al. 2016] is based on a complex 3D hand template and uses depth data and tracking, taking several seconds per frame. Our result shows comparable performance for single-hand and hand-object scenarios, using only per-frame detection on RGB with a runtime of about $0.7s$ per frame. Performance degrades for hand-hand interaction: When one of the hands is very occluded, our detector tends to fire on the occluding hand. Detecting joints on both

[10] Some other datasets also have manual keypoint annotations with RGB images [Sridhar et al. 2016; Tompson et al. 2014a], but the calibration parameters in [Sridhar et al. 2016] were not accurate enough, and the images in [Tompson et al. 2014a] are warped to match depth.

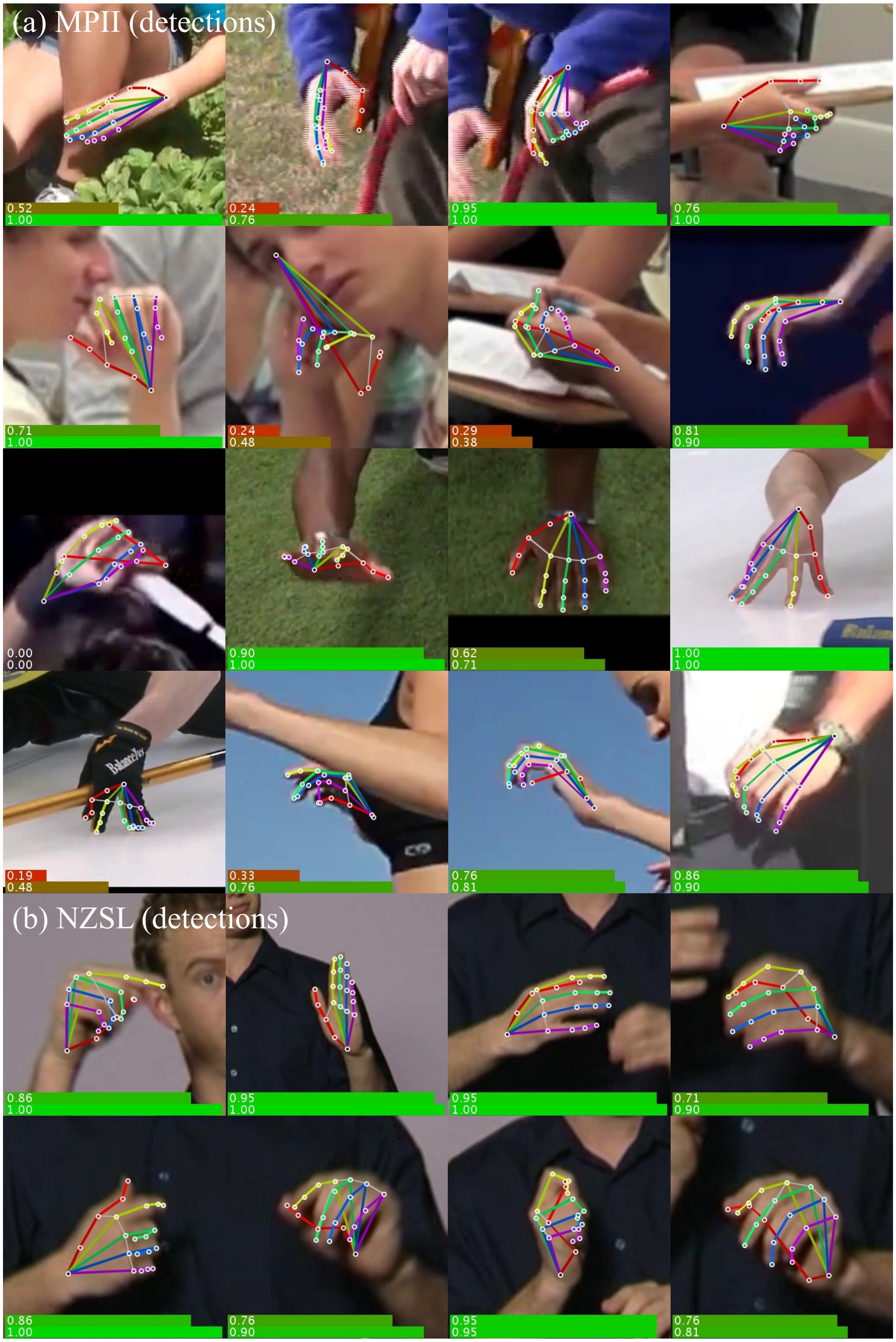


Figure 4.12: Detections using model "Mix 3" on test images. To show a representative sample, we pick (a) the first 16 images from the MPII test set, and (b) the first 8 from NZSL. Each image's PCK at $\sigma = 0.1$ and $\sigma = 0.2$ (across the 21 keypoints) is shown as a pair of bars at the bottom of the image, colored from red (0) to green (1).

hands simultaneously would be advantageous in these cases, rather than treating each hand independently as our current approach does.

Table 4.1: Average 2D error in pixels on the dataset of Tzionas et al. [2016] (a method that uses depth data).

	Single Hand				Hand-Object							Hand-Hand						
Sequence	Grasp	Flying	Rock Gesture	Bunny Gesture	Ball One Hand	Ball Two Hands	Bend Pipe	Bend Rope	Ball Occlu.	Move Cube	Moving Occlu.	Walk	Cross	Cross Twist	Tip Touch	Dancing	Tip Blend	Hugging
[Tzionas et al. 2016]	**4.37**	**5.11**	4.44	**4.50**	6.10	**7.15**	6.09	5.65	**8.03**	**4.68**	5.55	**5.99**	**4.53**	**4.76**	**3.65**	**6.49**	**4.87**	**5.22**
Ours	5.49	5.67	**4.15**	4.81	**5.75**	9.79	**5.47**	**4.35**	9.66	6.38	**5.40**	9.10	6.95	10.09	5.31	6.55	6.09	10.35

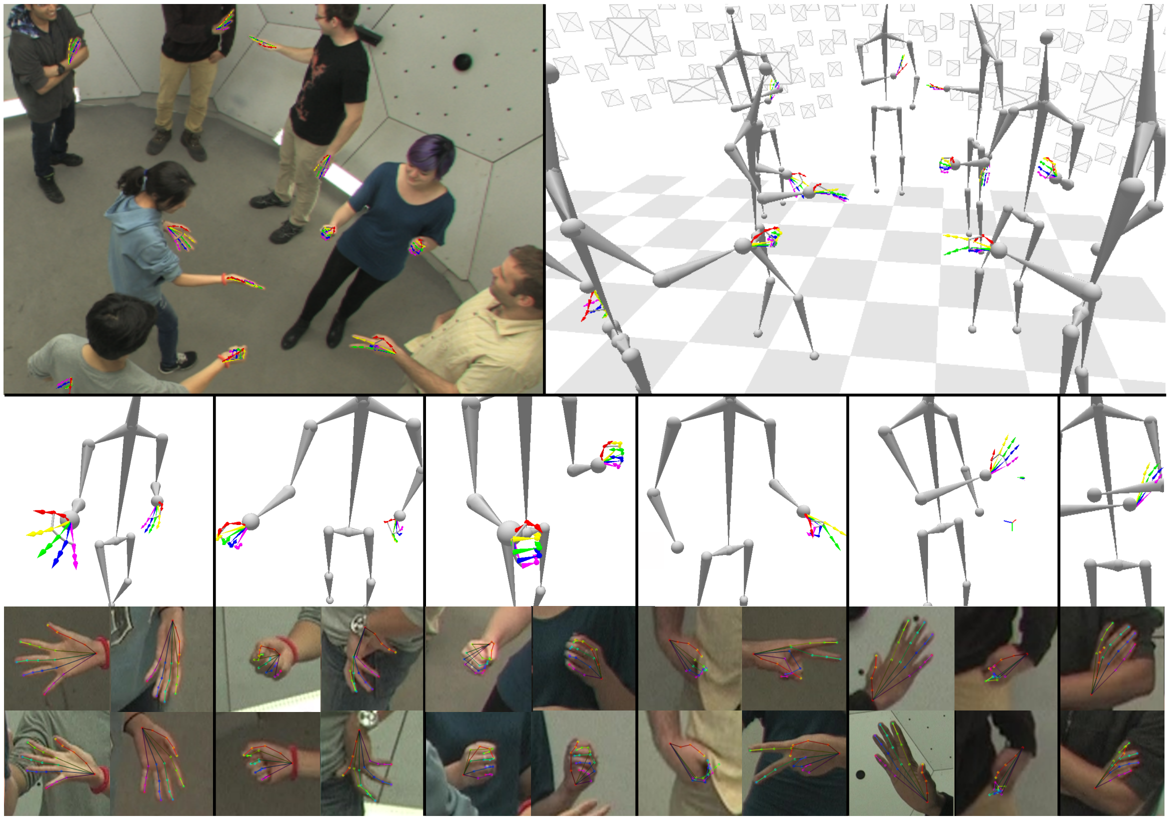


Figure 4.13: A keypoint detector that works at typical camera resolutions combined with a multiview system allows capturing the hand motions of entire groups of interacting people, a result that was not possible with any prior approaches.

4.4.4 Markerless Hand Motion Capture

The trained keypoint detectors allow us to reconstruct 3D hand motions in various challenging scenarios. We use the “Mix 3” detector on 31 HD camera views on Panoptic Studio data [Joo et al. 2015], and generate 3D hands by triangulation as we do for multiview bootstrapping. Working at normal camera resolutions allows us to reconstruct scenes with multiple interacting people including social games, as shown in Fig. 4.13, but also shelf building and a band performance. Figure 4.14 shows additional results in test scenes also include various

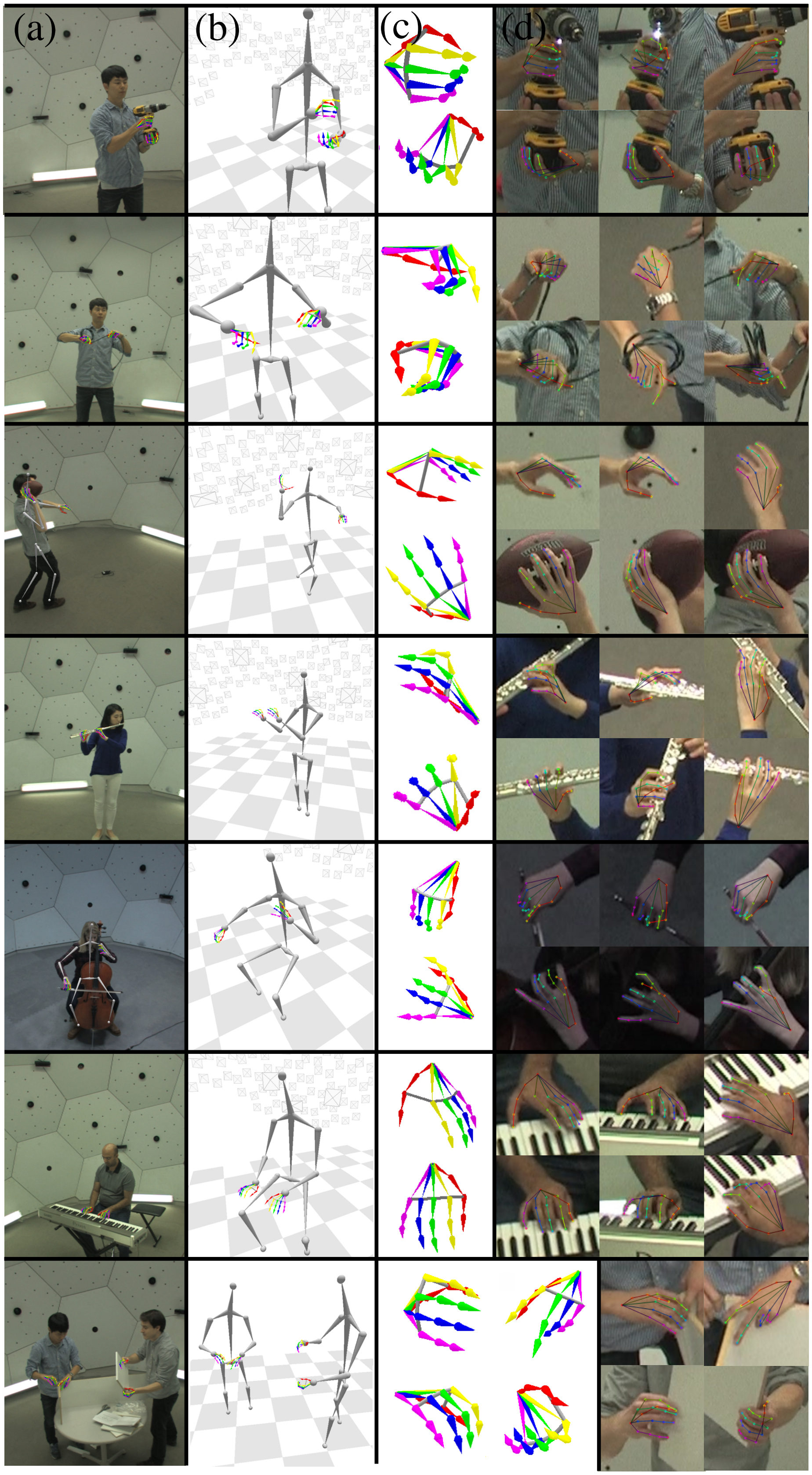


Figure 4.14: Qualitative multiview results on sequences not used during bootstrapping. (a) Reprojected triangulation. (b) 3D hands in context. (c) Metric reconstruction. (d) 2D detections from "Mix 3" on selected views.

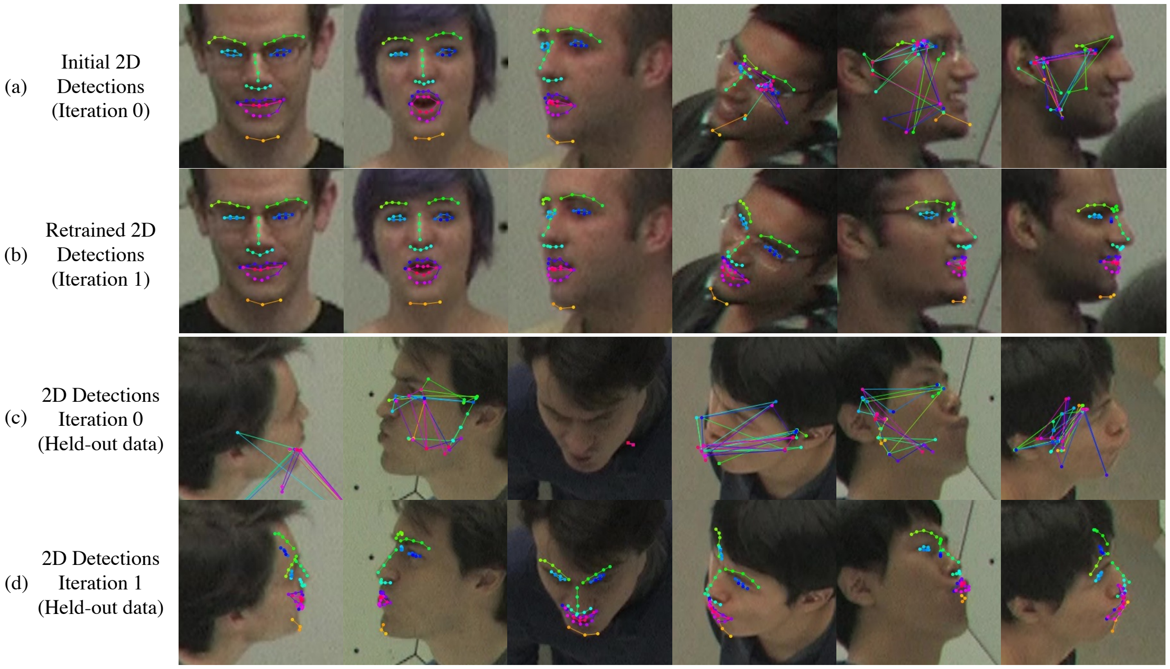

(a) Initial 2D Detections (Iteration 0)
(b) Retrained 2D Detections (Iteration 1)
(c) 2D Detections Iteration 0 (Held-out data)
(d) 2D Detections Iteration 1 (Held-out data)

Figure 4.15: Qualitative 2D detection results on the face. (a) Initial 2D detections from d_0. (b) 2D detections from d_1, on data included in the iteration 1 training set. (c) 2D detections from d_0. (d) 2D detections from d_1, on data not included in the iteration 1 training set.

practical hand motions, such as manipulating diverse tools (e.g., drill, scissors, ruler), sports motions (throwing a ball, bat swing), and playing musical instruments (piano, cello, flute, and guitars). Note that most depth-sensor based methods are not applicable in these scenarios, due to short sensor ranges and difficulties handling hand-object interactions. Refer to the supplementary video for additional reconstruction results.

4.5 Wide Viewing-Angle Face Landmark Detection

We apply the same method to the task of facial landmark detection. In this case, we detect $P = 54$ keypoints on the face, which correspond to a subset of the 68 landmarks used in the MultiPIE [Gross et al. 2008] dataset (face outline points that do not correspond to a unique 3D location are excluded). For iteration 0, the training set $\mathcal{T}_0$ consists of images from the i-Bug relabeling [Sagonas et al. 2016] of the LFPW [Belhumeur et al. 2011], AFW [Zhu and Ramanan 2012], HELEN [Le et al. 2012], IBUG [Sagonas et al. 2013] and FRGC-V2 [Phillips et al. 2005] datasets, in addition to MultiPIE. While these datasets contain a large sample of faces, they are restricted to near-frontal faces, with a range of approximately $-30°$ to $+30°$ on the horizontal plane, and an even narrower range in vertical angles.

Fig. 4.15 shows preliminary qualitative results for 2D detection after a single iteration of multiview bootstrapping. While the initial detector is reliable for frontal faces (e.g., the leftmost three faces in Fig. 4.15a) there is a breakdown point (due to viewing angle) after which the detector fails entirely (e.g., rightmost three images of Fig. 4.15a, all images of Fig. 4.15c). There is simply no training data in $\mathcal{T}_0$ similar enough to these extreme viewpoints. However, after even a single iteration of multiview bootstrapping the 2D detector is resilient to a much wider range of camera viewpoints, as shown in Fig. 4.15b and d.

4.6 Conclusions

We find that multiview bootstrapping is particularly useful to increase robustness to viewpoint and hand appearance. However, if a particular pose is difficult and detection fails across all views, reconstruction is impossible—no matter how many additional camera views we add. This is similarly true for occlusions that cannot be resolved from alternative views, e.g., hand in pocket. In this case, manual annotation (or using rendered data) is necessary. We would still recommend annotating on multiview images if possible, as it increases labeling consistency and can provide V labeled views for every pair or triplet of images annotated.

The biggest limiting factor we see is that large multiview camera systems are typically available only in laboratory conditions rather than in the wild. This limits the variety of acquired data to the particular lighting and cameras used, and the activities or actions performed, and is not reflective of any real-world distribution[11]. An exciting possibility that can bring such a system out of the laboratory are the multi-camera systems that are formed in-the-wild when several people record the same event using their mobile devices, also known as "social cameras" [Arev et al. 2014]. Although these setups present a complicated spatio-temporal calibration problem (mainly because the devices are not temporally-synchronized [Vo et al. 2016]), the potential for collecting and labeling real-world data is vast.

Despite these limitations, the hand pose estimation results obtained by our method in the multiview setting of the Panoptic Studio show much greater robustness and accuracy in the presence of hand-hand interactions and hand-object interactions than previous approaches, particularly over approaches based on model-fitting to depth data. This allows us to use our method in unconstrained scenarios where even marker-based motion capture tends to fail, including playing music, handling tools, and social interactions.

[11] However, it should be noted that this can also be an advantage when tailoring a detector for a particular sequence, application, or use-case.

Chapter 5.

Kronecker Markov Random Fields for Human Motion Data

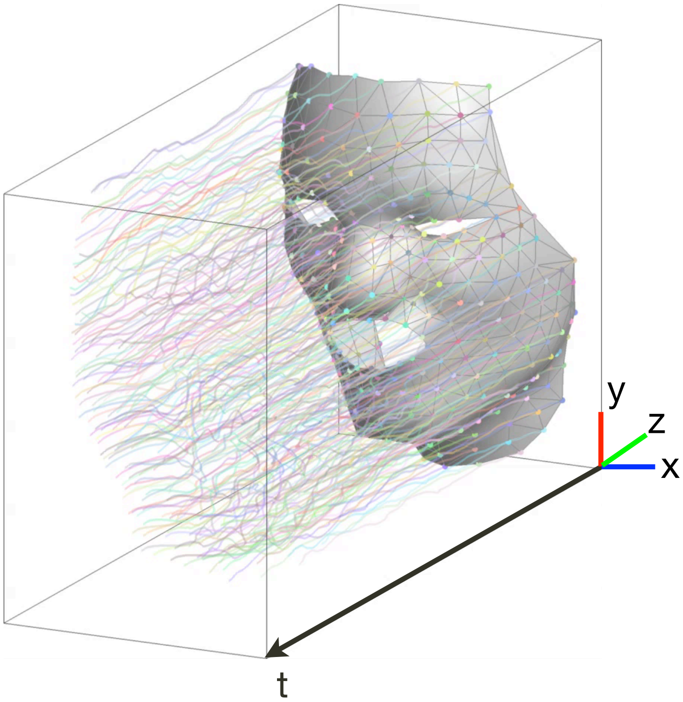


Figure 5.1: A time-varying point cloud of 313 points across 500 frames—roughly half a million degrees of freedom. Trajectories for each point are shown in translucent color. The data is very high-dimensional, driving the need for more compact and meaningful representations for modeling, inference, and prediction.

5.1 Introduction

Dynamic 3D reconstruction is the problem of recovering the time-varying 3D configuration of points from incomplete observations. The theoretical and practical challenges in this problem center on the issue of missing data. Even with a massively multiview system such as the Panoptic Studio, we cannot expect to measure the positions of every feature on the human body in every frame. A major cause of missing data is due to occlusions, particularly inter-occlusions between people but also self-occlusions and occlusions with objects (e.g., arms crossed, or hands in pockets). At other times, the keypoint detectors we use may fail if the observed data is not similar enough to any of the training samples or may be inaccurate, which produces undesirable artifacts such as jittering between frames. Thus, the question at the core of dynamic 3D reconstruction is what internal model a system should refer to when there is insufficient or noisy information.

This section introduces a model of statistical dependencies between the spatial and temporal dimensions of 3D motion data that we call a Kronecker Markov Random Field (KMRF). The model captures correlations across the spatial and temporal dimensions as a Markov random field with a particular Kronecker structure, and we show that this structure arises in the statistics of natural nonrigid objects such as the face and body. The Kronecker structure can be seen in the data's covariance matrix when the data is arranged as a vector, and corresponds equivalently to a Matrix Normal Distribution (MND) over a matrix arrangement of the data. This spatiotemporal distribution for dynamic structures unifies the shape, trajectory, and shape-trajectory models of prior art as specific instances. We demonstrate the models use in reconstructing 3D point clouds in the presence of missing data.

5.2 Time-Varying 3D Point Clouds

The time-varying trajectory of a single 3D point in $\mathbb{R}^{3\times1}$ is a sequence of 3D coordinates sampled at F discrete time instants or frames $\{1, 2, \cdots, F\}$, which can be laid out as,

$$\begin{bmatrix} X_1 & \cdots & X_f & \cdots & X_F \\ Y_1 & \cdots & Y_f & \cdots & Y_F \\ Z_1 & \cdots & Z_f & \cdots & Z_F \end{bmatrix} \tag{5.1}$$

or concatenated into a matrix of size $3 \times F$,

$$\underbrace{[\mathbf{x}_1 \;\; \mathbf{x}_2 \;\; \cdots \;\; \mathbf{x}_F]}_{\text{point trajectory}} \tag{5.2}$$

For multiple points, the time-varying structure of the entire configuration of P 3D points across F frames can be similarly be represented as a matrix $\mathbf{X} \in \mathbb{R}^{3P\times F}$, which is a stack of P point trajectories,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^1 & \mathbf{x}_2^1 & \cdots & \mathbf{x}_F^1 \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_1^P & \mathbf{x}_2^P & \cdots & \mathbf{x}_F^P \end{bmatrix}. \tag{5.3}$$

Here, I have used colors to highlight columns of the matrix, where the column f corresponds to the 3D shape in frame f, and is formed by vertically stacking the 3D points $\mathbf{x}_f^p \in \mathbb{R}^{3\times1}$ (denoting the p-th 3D point at frame f). In a slight abuse of notation we will interchangeably use lowercase bold letters to denote vectorized matrices, i.e., $\mathbf{x} = \text{vec}(\mathbf{X})$, where $\text{vec}(\mathbf{X})$ is used to denote the column-major vectorization of the matrix $\mathbf{X}$,

$$\operatorname{vec}(\mathbf{X}) = \begin{bmatrix} \mathbf{x}_1^1 \\ \vdots \\ \mathbf{x}_1^P \\ \mathbf{x}_2^1 \\ \vdots \\ \mathbf{x}_2^P \\ \vdots \\ \mathbf{x}_F^1 \\ \vdots \\ \mathbf{x}_F^P \end{bmatrix}. \tag{5.4}$$

Here, $\mathbf{x}$ is a tall vector with the columns of $\mathbf{X}$ stacked on top of each other.

5.2.1 Inference from Partial Observations

In practice, we often don't observe the 3D points directly. Among other factors, missing data and camera projection can result in only a reduced set of measurements of $\mathbf{X}$ being observed. Assuming a linear observation model, we can formalize this as

$$\mathbf{y} = \mathbf{O} \operatorname{vec}(\mathbf{X}) + \epsilon, \tag{5.5}$$

where $\mathbf{y}$ is a vector of observations of size n_{obs} (the number of observations), $\mathbf{O} \in \mathbb{R}^{n_{\text{obs}} \times 3PF}$ is the observation matrix, and ϵ is noise i.i.d. sampled from a normal distribution. In the simplest case of fully observed data, $\mathbf{O}$ is an identity matrix of size $3PF \times 3PF$. For entries x, y, or z that are missing, we would remove the corresponding rows of the identity matrix, yielding a matrix $\mathbf{O}_{\text{miss}}$ containing only a subset of the rows.

Many problems can be formulated as estimating the most likely spatiotemporal structure $\hat{\mathbf{X}}$ given some observations $\mathbf{y}$. Note, however, that typically $n_{\text{obs}} < 3PF$, and the problem $\min_{\mathbf{X}} \sigma^{-2} \|\mathbf{y} - \mathbf{O} \operatorname{vec}(\mathbf{X})\|_2^2$ is therefore often severely underconstrained. We take a Bayesian approach to the estimation problem and maximize the posterior,

$$\hat{\mathbf{X}} = \operatorname*{argmax}_{\mathbf{X}} p(\mathbf{X}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{X}) p(\mathbf{X}), \tag{5.6}$$

where $p(\mathbf{y}|\mathbf{X}) = \mathcal{N}\left(\mathbf{O} \operatorname{vec}(\mathbf{X}), \sigma^2 \mathbf{I}\right)$ from Eq. (5.5). Two missing data problems of particular relevance to this thesis are forecasting or prediction and dynamic 3D reconstruction cameras.

Forecasting. Similarly, predicting future time instants from past observations can be posed as a missing data problem where the matrix $\mathbf{X}$ is only observed up until a particular frame f, that is,

$$\underbrace{\begin{bmatrix} \mathbf{x}_1^1 & \cdots & \mathbf{x}_f^1 \\ \vdots & \cdots & \vdots \\ \mathbf{x}_1^P & \cdots & \mathbf{x}_f^P \end{bmatrix.}_{\text{observed history}} \underbrace{\begin{matrix} \mathbf{x}_{f+1}^1 & \cdots & \mathbf{x}_F^1 \\ \vdots & \cdots & \vdots \\ \mathbf{x}_{f+1}^P & \cdots & \mathbf{x}_F^P \end{matrix}\Bigg]}_{\text{future to predict}}. \tag{5.7}$$

This can be modeled as a missing data problem, where $\mathbf{O}_{\text{miss}} \in \mathbb{R}^{3Pf \times 3PF}$ is a truncated identity matrix, where the last $3P(F-f)$ rows have been removed, or equivalently,

$$\mathbf{y} = \begin{bmatrix} \mathbf{x}_1^1 \\ \vdots \\ \vdots \\ \mathbf{x}_f^P \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3Pf} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^1 \\ \vdots \\ \vdots \\ \mathbf{x}_f^P \\ \mathbf{x}_{f+1}^1 \\ \vdots \\ \vdots \\ \mathbf{x}_F^P \end{bmatrix}, \tag{5.8}$$

where $\mathbf{I}_N$ is the identity of size $N \times N$ and $\mathbf{0}$ is a matrix of zeros of the appropriate size.

Dynamic 3D reconstruction. The action of camera projection can also be modeled using a matrix $\mathbf{O}$. The effect of orthographic projection from a single camera can be expressed as a matrix $\mathbf{O}_{\text{ortho}}$ such that

$$\mathbf{y} = \underbrace{\begin{pmatrix} \mathbf{I}_P \otimes \mathbf{R}_1 & & \\ & \ddots & \\ & & \mathbf{I}_P \otimes \mathbf{R}_F \end{pmatrix}}_{\mathbf{O}_{\text{ortho}}} \text{vec}(\mathbf{X}) + \epsilon, \tag{5.9}$$

i.e., each of the P points is transformed by the camera matrix of its corresponding frame f, where the camera matrices $\mathbf{R}_f \in \mathbb{R}^{2\times 3}$ are typically truncated rotation matrices, and $\otimes$ denotes the Kronecker product. The case of a single camera observing the scene with unknown rotations $\mathbf{R}_f$ is the problem of Nonrigid Structure from Motion (NRSfM). For multiview reconstruction, several $\mathbf{O}_{\text{ortho}}$ matrices can be stacked, one for each camera observing the scene. If some of the projected points are missing, we can concatenate the effect of the matrices: $\mathbf{O} = \mathbf{O}_{\text{miss}}\mathbf{O}_{\text{ortho}}$. For this exposition, we assume that the observation matrix $\mathbf{O}$ is known (e.g., by construction, camera calibration, or rigid SfM [Del Bue et al. 2005]).

The goal for the remainder of this chapter is to design a prior $p(\mathbf{X})$ that models the data well while remaining amenable to global optimization. Ideally, a good model should capture all available correlations in the data—spatial, temporal, and spatiotemporal—as these correlations allow us to reason about the information that is missing. Because dynamic structure is high dimensional (e.g., 100 points over 120 frames is 36,000 degrees of freedom), the number of possible correlations is very large (i.e., ∼648 million parameters), and learning these correlations therefore requires a large quantity of samples, where each sample is a full spatiotemporal sequence. For most applications, such large numbers of sequences are not accessible. In the following, we present a probabilistic model of 3D data that captures most salient correlations but can still be estimated from a few or even one sequence.

The correlations present in spatiotemporal sequences are primarily a result of separable correlations across time and correlations across structure or shape [Gotardo and Martinez 2011; Akhter et al. 2012]. Our model represents these correlations as a matrix normal distribution (MND) over dynamic structure, which translates into a Kronecker pattern in the spatiotemporal covariance matrix. We show that this pattern is observed empirically. Additionally, we show that the marginal distributions over shape (independent of time) and over trajectories (independent of shape) predicted by the model correspond to the commonly used point distribution and trajectory models respectively. Similarly, bilinear shape-trajectory basis models can be derived from the MND model.

5.3 Statistics of Sampled Deforming Objects

To begin characterizing the spatiotemporal distribution $p(\mathbf{X})$, consider the lifetime of a non-rigid object, for example, a human face. Let us denote the position of P points on the face by $\mathbf{y}(t) \in \mathbb{R}^{3P}$, where t is time. At a given time instant t (which a priori could be any moment with equal probability), you decide to hit "record" on your favorite motion capture system or video camera. The device takes observations at discrete intervals (say, 60Hz), and for some finite duration (say, from a few seconds to a few minutes). Let us denote the set of recorded time instants by $\{t+1, t+2, \cdots, t+F\}$. The object's lifetime (for example, the facial expressions of a person throughout his life) is typically much larger than the duration F that we can record, and we can assume time extends infinitely before and after the recording event, yielding a vector-valued time-series:

$$\mathbf{y}_{t-\infty} \quad \cdots \quad \mathbf{y}_{t-1} \quad \mathbf{y}_t \quad \underbrace{\mathbf{y}_{t+1} \quad \mathbf{y}_{t+2} \quad \cdots \quad \mathbf{y}_{t+F}}_{\text{captured sequence}} \quad \mathbf{y}_{t+F+1} \quad \cdots \quad \mathbf{y}_{t+\infty} \tag{5.10}$$

We would like to determine an a priori spatiotemporal distribution for the recorded sequence. For simplicity, let's consider a device that captures noiseless data (i.e., the observation matrix $\mathbf{O}$ is the identity), so that the problem is equivalent to determining $p(\mathbf{X}_t)$, where $\mathbf{X}_t \in \mathbb{R}^{3P\times F}$ is the recorded sequence starting at time t, $\mathbf{X}_t = \begin{pmatrix}\mathbf{y}_{t+1} & \mathbf{y}_{t+2} & \cdots & \mathbf{y}_{t+F}\end{pmatrix}$. If we had access to the entire lifetime of the object, one way to proceed would be to record a large set of such possible sequences and directly compute statistics on them. In fact, let us assume that we can record an infinity of such possible training sequences of length F, and let us reference all time with respect to some arbitrary origin so that t=0. Our training set can then be denoted as the set $\{\mathbf{X}_0, \mathbf{X}_2, \cdots, \mathbf{X}_\infty\}$. We will study the first and second moments of this process for various cases of interest.

5.3.1 Individual Frames: Shapes

Let us first consider the case where the capture device acquires only a single frame—i.e., we capture still frames or snapshots instead of video sequences. This corresponds to the case where $F = 1$, so that we sample a single frame $\mathbf{X}_t = \begin{pmatrix}\mathbf{y}_{t+1}\end{pmatrix}$. In this case, $\mathbf{X}_t \in \mathbb{R}^{3P}$ describes the instantaneous position of the P points at that particular time instant, which we refer to as the *shape* or configuration of the object. Here $p(\mathbf{X})$ models the prior distribution of captured 3D points independently of time. In fact, we will see that the Point Distribution Model (PDM) or linear shape basis model [Mardia and Dryden 1989; Cootes et al. 1995a] can be derived from the first and second moments of this distribution:

- What is the mean shape that we capture, $E[\text{vec}(\mathbf{X})]$?

$$E[\text{vec}(\mathbf{X})] = \lim_{N\to\infty} \frac{1}{N}\sum_{t=0}^{N} \text{vec}(\mathbf{X}_t) = \lim_{N\to\infty} \frac{1}{N}\sum_{t=0}^{N} \begin{pmatrix}\mathbf{y}_{t+1}\end{pmatrix} = E[\mathbf{y}]. \tag{5.11}$$

Which (unsurprisingly) says that, averaged out over a large enough set of captured data, we can expect the mean of the captured data to be the same as the mean of the object throughout its lifetime.

- What is the covariance, $\text{cov}(\text{vec}(\mathbf{X}))$? For simplicity, let us assume that our data is zero mean, $E[\mathbf{X}] = \mathbf{0}$. Then,

$$\begin{aligned}\text{cov}(\text{vec}(\mathbf{X})) &= \lim_{N\to\infty} \frac{1}{N-1}\sum_{t=0}^{N} \text{vec}(\mathbf{X}_t)\,\text{vec}(\mathbf{X}_t)^\mathsf{T} \\ &= \lim_{N\to\infty} \frac{1}{N-1}\sum_{t=0}^{N} \begin{pmatrix}\mathbf{y}_{t+1}\mathbf{y}_{t+1}^\mathsf{T}\end{pmatrix} = \boldsymbol{\Delta},\end{aligned} \tag{5.12}$$

which again says that, averaged out over a large enough set of captured time instants, we

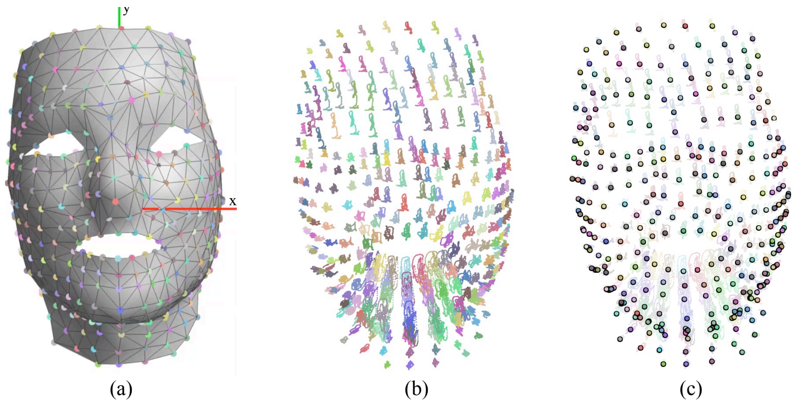


Figure 5.2: (a) A single frame from the training set. (b) The entire training set of frames, overlaid on top each other. Each dot corresponds to the position of a point in a particular frame of the training set. (c) The mean shape.

can expect the covariance of the captured data to tend to be the same as the covariance of the object's shape throughout its lifetime, which can be written as a matrix $\mathbf{\Delta} \in \mathbb{R}^{3P \times 3P}$.

To illustrate these quantities visually, we can simulate the sampling process using a finite set of training data. For this example, we take dense motion capture sequences of the face during speech. An example frame from the sequence can be seen in Fig 5.2 (a), where there are 313 points on the face (each given a distinct color), and we show a triangulated mesh for visualization. The entire set of training frames (about a 1000 in this case) is plotted overlaid on top of each other in the middle vignette, where it is apparent that most of the movement in this sequence is due to speech. The mean shape is shown in Fig 5.2 (c), and is equivalent to the mean position of each point individually.

Visualizing the covariance is more difficult due to its high dimensionality, but conceptually, it simply measures the linear correlation between all pairs of variables. We have a total of $3P$ variables (the x,y, and z coordinates of P points), and the covariance is a $3P \times 3P$ matrix $\mathbf{\Delta}$ where each entry $\Delta_{i,j}$ measures the covariance between variables i and j. The matrix is symmetric and positive semi-definite. Fig. 5.3 (a) visualizes the absolute magnitude of the entries in grayscale, where darker means more correlated. We can display projections of this high dimensional covariance to give us intuitions about the data. For example, Fig. 5.3 (b) visualizes the 3×3 block diagonal entries of the covariance matrix, i.e, the covariance between the x, y, and z coordinates of each point[1], This is the same as modeling each point independently, and we can visualize the spatial distribution over position as Gaussian ellipsoids.

The study of this covariance gives rise to the Point Distribution Model (PDM) of shape

[1] This corresponds to an ordering of variables such that the XYZ coordinates are laid out contiguously, $\mathbf{X} = [x_1\ y_1\ z_1\ x_2\ y_2\ z_2\ \cdots\ x_P\ y_P\ z_P]^\top$, so that the block-diagonal entries correspond to the spatial distribution.

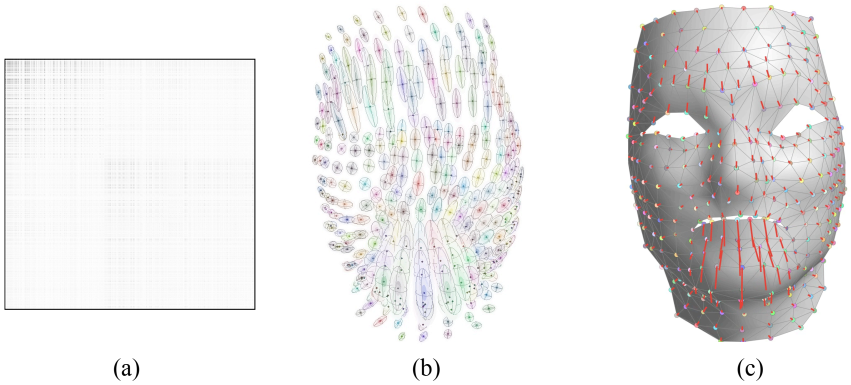


Figure 5.3: (a) The covariance matrix, shown as grayscale absolute magnitude. (b) The 3×3 block diagonals of the covariance matrix correspond to Gaussian ellipsoids indicating the spatial distribution of individual 3D points. (c) The principal component of the shape covariance matrix, shown as a deformed shape. In red, we show the displacement of each point along this particular mode of deformation.

deformation. The basic approach is to perform an eigen-decomposition of the matrix, such that $\mathbf{\Delta} = \tilde{\mathbf{B}}\mathbf{W}_b^2\tilde{\mathbf{B}}^\mathsf{T}$, where $\tilde{\mathbf{B}} \in \mathbb{R}^{3P\times 3P}$ is an orthonormal matrix and $\mathbf{W}_b^2$ is a diagonal matrix of eigenvalues. It can be shown that columns of the matrix $\tilde{\mathbf{B}}$ correspond to the principal modes of variation of the data, known variously as the *principal components*, *shape basis vectors*, *deformation modes*, *eigen-shapes*, or *eigen-modes* [Mardia and Dryden 1989; Le and Kendall 1993; Cootes et al. 1995a; Bregler et al. 2000]. The eigenvalues along the diagonal matrix $\mathbf{W}_b$ correspond to the "energy" associated with each component. This is the fraction of the total variance that is captured or explained by this component. The sum of all components makes up the total variance within the training set:

$$\mathbf{\Delta} = \sum_{i=1}^{3P} \lambda_i \tilde{\mathbf{b}}_i \tilde{\mathbf{b}}_i^\mathsf{T}, \qquad \lambda_i = [\mathbf{W}_b^2]_{i,i}, \tag{5.13}$$

where λ_i is the eigenvalue or total variance captured by eigenvector $\tilde{\mathbf{b}}_i$. A characteristic of most naturally deforming objects[2] is that the eigenvalues λ_i decay very quickly, with most of the variance concentrated in the first few principal components. This is the underlying reason for the effectiveness of *truncated* shape basis models, which approximate a shape $\mathbf{x}_t$ as the sum of a linear combination of k_s shape modes[3], with $k_s \ll 3P$,

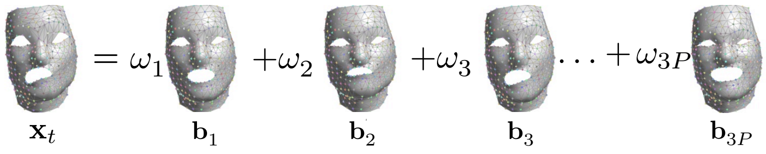
$$\mathbf{x}_t = \sum_{i=1}^{k_s} \omega_i^t \mathbf{b}_i, \tag{5.14}$$

where we let $\mathbf{b}_i = \sqrt{\lambda_i}\tilde{\mathbf{b}}_i$ be a (scaled) shape mode, and ω_i^t is the coefficient or relative

[2] Or at least in a spatially localized neighborhood thereof.

[3] These expressions can all be written as versions that include a mean, e.g., $\mathbf{x}_t = \mu + \sum_{i=1}^{k_s} \omega_i^t \mathbf{b}_i$.

contribution of that shape mode. Visually,



any shape is a sum of a set of weighted basis shapes, which we can write as $\mathbf{x}_t = \mathbf{B}\boldsymbol{\omega}$, where $\mathbf{B} = \tilde{\mathbf{B}}\mathbf{W}_b$. Generally, these models represent a change of basis, where the basis $\mathbf{B}$ is chosen such that the coefficients $\boldsymbol{\omega} \in \mathbb{R}^{3P}$ are uncorrelated or *whitened*. It is easy to see that if $\text{cov}(\boldsymbol{\omega}) = \mathbf{I}_{3P}$, (i.e., the covariance of ω is the identity), and we perform the change of basis $\mathbf{x}_t = \mathbf{B}\boldsymbol{\omega}$, then $\text{cov}(\mathbf{x}_t) = E\left[\mathbf{B}\boldsymbol{\omega}\boldsymbol{\omega}^\mathsf{T}\mathbf{B}^\mathsf{T}\right] = \boldsymbol{\Delta}$. If we additionally assume that $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{3P})$, then $\mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Delta})$, which is equivalent to the probabilistic PCA model of Torresani et al. [2008].

5.3.2 Individual Coordinates: Trajectories

We can consider another edge case in which the sequences that we are recording consist of a single point. In fact, to make it simpler, we will consider the case where we record a single scalar value—for example the time-varying y coordinate of a single point. In this case, the data we are capturing is a scalar-valued time-series,

$$y_{t-\infty} \quad \cdots \quad y_{t-1} \quad y_t \quad \underbrace{y_{t+1} \quad y_{t+2} \quad \cdots \quad y_{t+F}}_{\text{captured sequence}} \quad y_{t+F+1} \quad \cdots \quad y_{t+\infty} \tag{5.15}$$

We repeat the same analysis we did before but now for the timeseries $\mathbf{X}_t = \begin{pmatrix} y_{t+1} & y_{t+2} & \cdots & y_{t+F} \end{pmatrix}$.

- What is the mean sequence, $E[\text{vec}(\mathbf{X})]$?

$$E[\text{vec}(\mathbf{X})] = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N} \text{vec}(\mathbf{X}_t) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N} \begin{pmatrix} y_{t+1} \\ y_{t+2} \\ \vdots \\ y_{t+F} \end{pmatrix} = \begin{pmatrix} E[y] \\ E[y] \\ \vdots \\ E[y] \end{pmatrix}. \tag{5.16}$$

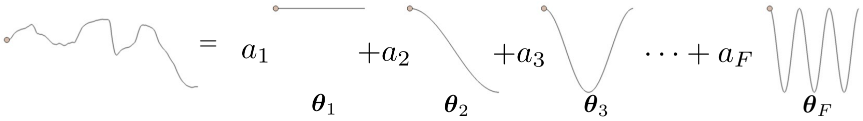
For a large set of training sequences, the mean tends to stationary and all the recorded time instants in $\mathbf{X}$ are equal, so the mean sequence becomes,

$$E[\mathbf{X}] = \begin{bmatrix} E[y] & E[y] & \cdots & E[y] \end{bmatrix}. \tag{5.17}$$

- What is the temporal covariance, $\text{cov}(\text{vec}(\mathbf{X}))$? For simplicity, let us assume that our data is zero mean, $E[\mathbf{X}] = \mathbf{0}$. Then,

$$\begin{aligned}\operatorname{cov}(\operatorname{vec}(\mathbf{X})) &= \lim_{N\to\infty} \frac{1}{N-1} \sum_{t=0}^{N} \operatorname{vec}(\mathbf{X}_t) \operatorname{vec}(\mathbf{X}_t)^\mathsf{T} \\ &= \lim_{N\to\infty} \frac{1}{N-1} \sum_{t=0}^{N} \begin{pmatrix} y_{t+1}y_{t+1}^\mathsf{T} & y_{t+1}y_{t+2}^\mathsf{T} & y_{t+1}y_{t+3}^\mathsf{T} & \cdots & y_{t+1}y_{t+F}^\mathsf{T} \\ y_{t+2}y_{t+1}^\mathsf{T} & y_{t+2}y_{t+2}^\mathsf{T} & y_{t+2}y_{t+3}^\mathsf{T} & \cdots & y_{t+2}y_{t+F}^\mathsf{T} \\ \vdots & & & & \vdots \\ y_{t+F}y_{t+1}^\mathsf{T} & y_{t+F}y_{t+2}^\mathsf{T} & y_{t+F}y_{t+3}^\mathsf{T} & \cdots & y_{t+F}y_{t+F}^\mathsf{T} \end{pmatrix} = \mathbf{\Sigma}.\end{aligned} \tag{5.18}$$

As before, we can analyze this covariance in terms of the eigenvectors—in this case, the eigen-trajectories [Sidenbladh et al. 2000b], or principal modes of temporal variation. We can write the trajectory covariance analogously as before, $\mathbf{\Sigma} = \tilde{\mathbf{\Theta}}\mathbf{W}_t^2\mathbf{\Theta}^\mathsf{T}$, where $\tilde{\mathbf{\Theta}}$ is the orthonormal basis of principal components, and the transform $\mathbf{X}_t = \mathbf{a}_t^\mathsf{T}\mathbf{\Theta}^\mathsf{T}$ corresponds to a representation of the data in which the coefficients, $\mathbf{a} \in \mathbb{R}^F$, are uncorrelated and have an autocovariance of 1. Visually, if time increases along the x axis in the plots below,



we are decomposing the temporal variation into a set of temporal basis vectors. This corresponds to a probabilistic PCA model analogous to the shape model of Torresani et al. [2008] but in the space of trajectories, similar to the general trajectory prior of Valmadre and Lucey [2012].

5.3.3 Sequences of Shapes: Shape-Trajectories

We return to the original problem of modeling spatiotemporal data, where the recorded data is now a vector-valued sequence of F frames or shapes $\mathbf{X}_t \in \mathbb{R}^{3P\times F}$,

$$\mathbf{X}_t = \begin{pmatrix} \mathbf{y}_{t+1} & \mathbf{y}_{t+2} & \cdots & \mathbf{y}_{t+F} \end{pmatrix}. \tag{5.19}$$

- What is the mean spatiotemporal sequence, $E[\operatorname{vec}(\mathbf{X})]$?

$$E[\operatorname{vec}(\mathbf{X})] = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N} \operatorname{vec}(\mathbf{X}_t) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N} \begin{pmatrix} \mathbf{y}_{t+1} \\ \mathbf{y}_{t+2} \\ \vdots \\ \mathbf{y}_{t+F} \end{pmatrix} = \begin{pmatrix} E[\mathbf{y}] \\ E[\mathbf{y}] \\ \vdots \\ E[\mathbf{y}] \end{pmatrix}. \tag{5.20}$$

For a large set of training sequences, the mean is a stationary vector across all columns of $\mathbf{X}$, i.e.,

$$E[\mathbf{X}] = \begin{bmatrix} E[\mathbf{y}] & E[\mathbf{y}] & \cdots & E[\mathbf{y}] \end{bmatrix}. \tag{5.21}$$

• What is the spatiotemporal covariance, $\mathrm{cov}(\mathrm{vec}(\mathbf{X}))$? For simplicity, let us assume that our data is zero mean, $E[\mathbf{X}] = \mathbf{0}$. Then,

$$\begin{aligned} \mathrm{cov}(\mathrm{vec}(\mathbf{X})) &= \lim_{N\to\infty} \frac{1}{N-1} \sum_{t=0}^{N} \mathrm{vec}(\mathbf{X}_t)\,\mathrm{vec}(\mathbf{X}_t)^{\mathsf{T}} \\ &= \lim_{N\to\infty} \frac{1}{N-1} \sum_{t=0}^{N} \begin{pmatrix} \mathbf{y}_{t+1}\mathbf{y}_{t+1}^{\mathsf{T}} & \mathbf{y}_{t+1}\mathbf{y}_{t+2}^{\mathsf{T}} & \mathbf{y}_{t+1}\mathbf{y}_{t+3}^{\mathsf{T}} & \cdots & \mathbf{y}_{t+1}\mathbf{y}_{t+F}^{\mathsf{T}} \\ \mathbf{y}_{t+2}\mathbf{y}_{t+1}^{\mathsf{T}} & \mathbf{y}_{t+2}\mathbf{y}_{t+2}^{\mathsf{T}} & \mathbf{y}_{t+2}\mathbf{y}_{t+3}^{\mathsf{T}} & \cdots & \mathbf{y}_{t+2}\mathbf{y}_{t+F}^{\mathsf{T}} \\ \vdots & & & & \vdots \\ \mathbf{y}_{t+F}\mathbf{y}_{t+1}^{\mathsf{T}} & \mathbf{y}_{t+F}\mathbf{y}_{t+2}^{\mathsf{T}} & \mathbf{y}_{t+F}\mathbf{y}_{t+3}^{\mathsf{T}} & \cdots & \mathbf{y}_{t+F}\mathbf{y}_{t+F}^{\mathsf{T}} \end{pmatrix}. \end{aligned} \tag{5.22}$$

It is apparent that the resulting covariance will be symmetric and have a clear block structure,

$$\mathrm{cov}(\mathrm{vec}(\mathbf{X})) = \begin{pmatrix} \mathbf{\Delta}_{1,1} & \mathbf{\Delta}_{1,2} & \mathbf{\Delta}_{1,3} & \cdots & \mathbf{\Delta}_{1,F} \\ \mathbf{\Delta}_{2,1} & \mathbf{\Delta}_{2,2} & \mathbf{\Delta}_{2,3} & \cdots & \mathbf{\Delta}_{2,F} \\ \mathbf{\Delta}_{3,1} & \mathbf{\Delta}_{3,2} & \mathbf{\Delta}_{3,3} & \cdots & \mathbf{\Delta}_{3,F} \\ \vdots & & & & \vdots \\ \mathbf{\Delta}_{F-1,1} & \mathbf{\Delta}_{F-1,2} & \mathbf{\Delta}_{F-1,3} & \cdots & \mathbf{\Delta}_{F-1,F} \\ \mathbf{\Delta}_{F,1} & \mathbf{\Delta}_{F,2} & \mathbf{\Delta}_{F,3} & \cdots & \mathbf{\Delta}_{F,F} \end{pmatrix}, \tag{5.23}$$

where each matrix $\mathbf{\Delta}_{f_1,f_2} = \lim_{N\to\infty} \frac{1}{N-1} \sum_{t=0}^{N} \mathbf{y}_{t+f_1}\mathbf{y}_{t+f_2}^{\mathsf{T}}$. In fact, as $N \to \infty$, this autocovariance matrix[4] will tend to be block Toeplitz, i.e., $\mathbf{\Delta}_{f_1,f_1+\delta} = \mathbf{\Delta}_{f_2,f_2+\delta}$. Equivalently, the covariance between two point coordinates p_1, p_2 and two time instants f_1, f_2 can be expressed as:

$$\Delta_{f_1,f_2}^{p_1,p_2} = c_{p_1,p_2}[f_2 - f_1], \tag{5.24}$$

where $\mathrm{c}_{p_1,p_2}[f]$ is an appropriate covariance function (i.e., $\mathrm{c}_{p_1,p_2}[f] = \mathrm{c}_{p_1,p_2}[-f]$ and $\mathrm{c}_{p_1,p_2}[0] \geq \mathrm{c}_{p_1,p_2}[f]$, and $\mathrm{c}_{p_1,p_2}[0] > 0$). Note that this means that the sampling of this spatiotemporal process is weak-sense stationary—the second moment only depends on the time difference or lag $f_2 - f_1$, not on the particular frames. In this case, the matrix is fully specified by its first

[4] Or autocorrelation matrix, where $\mathbf{C}(\mathrm{vec}(X)) = \mathbf{R}(\mathrm{vec}(\mathbf{X})) - \mathbf{m}_x\mathbf{m}_x^{\mathsf{T}}$ with $\mathbf{m}_x = E[\mathrm{vec}(\mathbf{X})]$, where $\mathbf{R}$ is autocorrelation and $\mathbf{C}$ autocovariance.

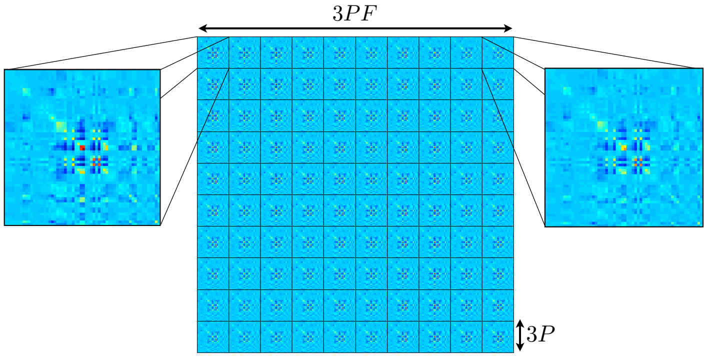


Figure 5.4: Empirical cov(vec($\mathbf{X}$)) computed from a large set of facial motion capture sequences. The $\mathbf{\Delta}_{f_1,f_2}$ blocks of size $3P \times 3P$ have been separated by lines for visualization, and there are F of these per row. The sampling process was simulated by taking subsequences of size $F=10$ frames, and choosing $P=14$ points on the face. $N=4000$ training windows were used to produce this matrix.

block row, $\mathbf{\Delta}_{1,:}$, and we can simply write $\mathbf{\Delta}_f$ corresponding to lag $f-1$, i.e.,

$$\mathrm{cov}(\mathrm{vec}(\mathbf{X})) = \begin{pmatrix} \mathbf{\Delta}_1 & \mathbf{\Delta}_2 & \mathbf{\Delta}_3 & \cdots & \mathbf{\Delta}_F \\ \mathbf{\Delta}_2 & \mathbf{\Delta}_1 & \mathbf{\Delta}_2 & \cdots & \mathbf{\Delta}_{F-1} \\ \mathbf{\Delta}_3 & \mathbf{\Delta}_2 & \mathbf{\Delta}_1 & \cdots & \mathbf{\Delta}_{F-2} \\ \vdots & & & & \vdots \\ \mathbf{\Delta}_{F-1} & \mathbf{\Delta}_{F-2} & \mathbf{\Delta}_{F-3} & \cdots & \mathbf{\Delta}_2 \\ \mathbf{\Delta}_F & \mathbf{\Delta}_{F-1} & \mathbf{\Delta}_{F-2} & \cdots & \mathbf{\Delta}_1 \end{pmatrix}, \tag{5.25}$$

Figure 5.4 visualizes this covariance for the dataset of facial motion capture sequences (an example sequence can be seen in Figure 5.1). Not only is the matrix approximately block Toeplitz (as predicted), but there is a lot of redundancy—it is apparent that there are correlations between the entries. This pattern leads us to propose the following approximation of the spatiotemporal covariance,

$$\mathrm{cov}(\mathrm{vec}(\mathbf{X})) = \mathbf{\Sigma} \otimes \mathbf{\Delta} = \begin{pmatrix} \Sigma_{1,1}\mathbf{\Delta} & \Sigma_{1,2}\mathbf{\Delta} & \cdots & \Sigma_{1,F}\mathbf{\Delta} \\ \Sigma_{2,1}\mathbf{\Delta} & \Sigma_{2,2}\mathbf{\Delta} & \cdots & \Sigma_{2,F}\mathbf{\Delta} \\ \vdots & & & \vdots \\ \Sigma_{F,1}\mathbf{\Delta} & \Sigma_{F,2}\mathbf{\Delta} & \cdots & \Sigma_{F,F}\mathbf{\Delta} \end{pmatrix}, \tag{5.26}$$

where $\otimes$ is the Kronecker product and $\mathbf{\Delta} \in \mathbb{R}^{3P\times 3P}$ is a shape covariance matrix (as in Sect. 5.3.1), and $\mathbf{\Sigma} \in \mathbb{R}^{F\times F}$ is a trajectory covariance matrix (as in Sect. 5.3.2). We have now derived expressions for the first and second moments of our spatiotemporal distribution.

5.4 The Matrix Normal Distribution

The Matrix Normal Distribution (MND) [Dutilleul 1999] is the maximum entropy distribution for which the first and second moments match the specific expressions given in the previous section[5], which we denote by $\mathcal{MN}$ and write as

$$\mathbf{Z} \sim \mathcal{MN}(\mathbf{U}, \mathbf{\Delta}, \mathbf{\Sigma}), \tag{5.27}$$

where $\mathbf{U} \in \mathbb{R}^{F\times 3P}$, $\mathbf{\Delta} \in \mathbb{R}^{3P\times 3P}$ is the row covariance and $\mathbf{\Sigma} \in \mathbb{R}^{F\times F}$ is the column covariance, can equivalently be expressed as a multivariate Gaussian over the vectorized elements of the matrix, i.e. $\mathbf{z} = \mathrm{vec}(\mathbf{Z})$, where $\mathbf{z} \in \mathbb{R}^{3PF}$ and $\mathbf{Z} \in \mathbb{R}^{3P\times F}$. Here, $\mathrm{vec}(\cdot)$ is the column-major unfolding, which results in the following distribution over the entries:

$$\mathbf{z} \sim \mathcal{N}(\mu, \mathbf{\Sigma} \otimes \mathbf{\Delta}), \tag{5.28}$$

The covariance of this multivariate Gaussian distribution is given by the Kronecker product, $\mathbf{\Sigma} \otimes \mathbf{\Delta}$. Let us denote the corresponding precision matrix as $\mathbf{Q} = (\mathbf{\Sigma} \otimes \mathbf{\Delta})^{-1} = \mathbf{\Sigma}^{-1} \otimes \mathbf{\Delta}^{-1}$ where $\mathbf{Q} \in \mathbb{R}^{3PF\times 3PF}$, which exists when both $\mathbf{\Delta}$ and $\mathbf{\Sigma}$ are positive definite; the density function is

$$p(\mathbf{z}) = (2\pi)^{-\frac{3PF}{2}} |\mathbf{Q}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{z}-\mu)^\top \mathbf{Q}(\mathbf{z}-\mu)\right), \tag{5.29}$$

or equivalently,

$$p(\mathbf{Z}) = (2\pi)^{-\frac{3PF}{2}} |\mathbf{\Sigma}|^{-\frac{3P}{2}} |\mathbf{\Delta}|^{-\frac{F}{2}} \exp\left(-\frac{1}{2}\,\mathrm{tr}\left[\mathbf{\Sigma}^{-1}(\mathbf{Z}-\mathbf{U})^\top \mathbf{\Delta}^{-1}(\mathbf{Z}-\mathbf{U})\right]\right). \tag{5.30}$$

A completely equivalent multivariate distribution can be formed by considering the row-major unfolding, $\mathbf{z}^r = \mathrm{vec}(\mathbf{Z}^\top)$, which is simply a permutation of the entries of $\mathbf{z}$. This ordering will result in a covariance matrix given by $\mathbf{\Delta} \otimes \mathbf{\Sigma}$. These two possibilities are illustrated in Fig. 5.5. Although visually quite distinct, it should be noted that they are equivalent representations of the same underlying distribution.

To gain further insight into the structure of these covariance and precision matrices, it is useful to think of the dependencies between the variables by modeling the set of variables in the matrix $\mathbf{X}$ as a Markov random field. A Gaussian Markov Random Field (GMRF) is simply a random vector that follows a multivariate Gaussian distribution [Rue and Held 2005], as does $\mathbf{z}$ in our case. The precision matrix of the multivariate Gaussian dictates the graph structure of the associated Markov random field. The field can be thought of as a lattice of nodes where each node corresponds to an entry i into the random vector $\mathbf{z}$, as

[5] The Gaussian distribution is the maximum entropy distribution for fixed first and second moments.

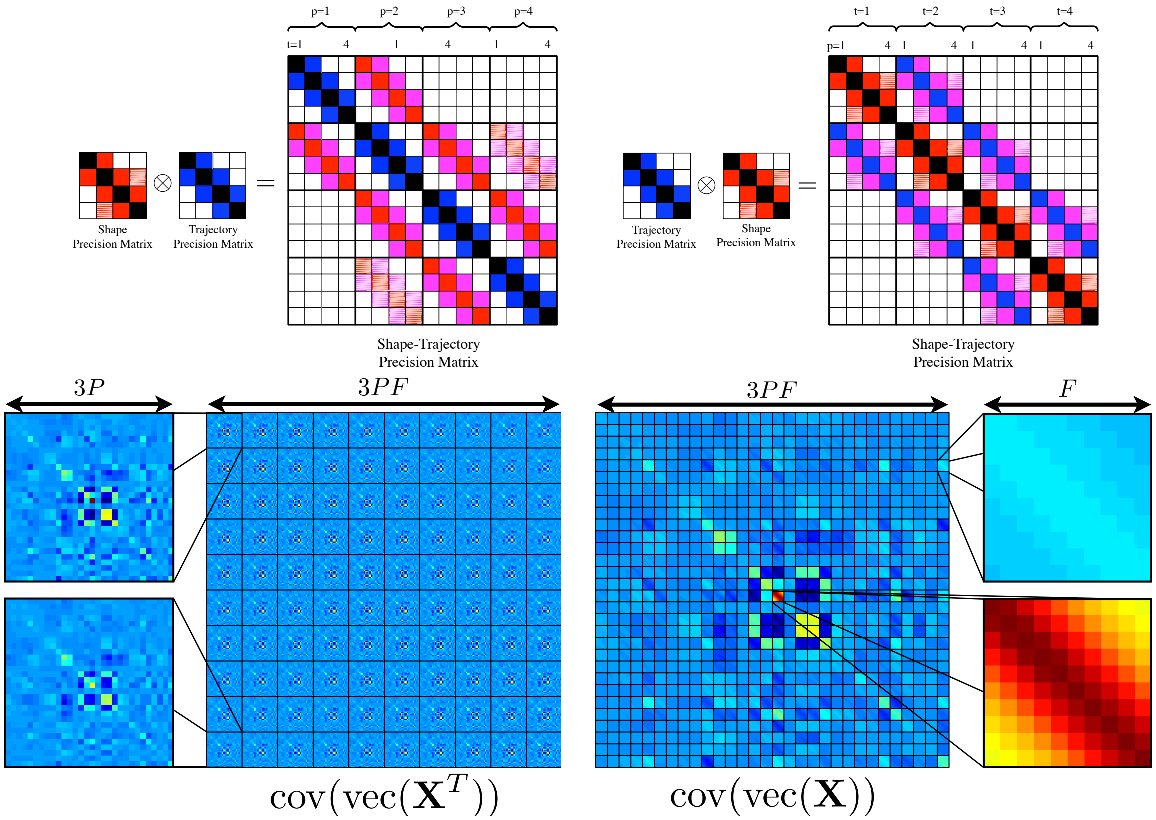


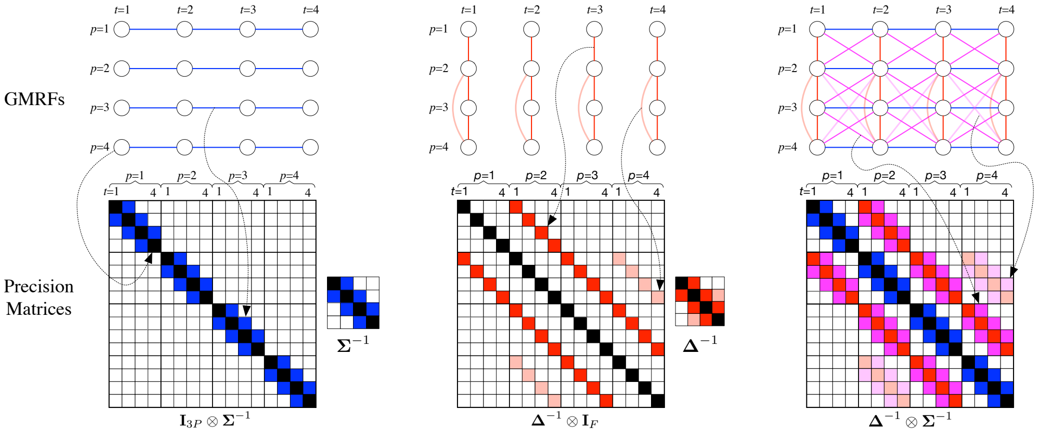
Figure 5.5: Top: Time major (left, row-major) and coordinate major (right, column-major) unfoldings. Blank entries represent zeros, and magenta entries represent the multiplication of blue and red entries. Bottom: Empirical spatiotemporal covariance matrix for a subset of face motion capture data (P=10, F=10), shown for two possible vectorizations of the matrix $\mathbf{X}$. (Left) The row-major arrangement shows blocks that are approximately scaled versions of the spatial or row covariance. (Right) The column-major arrangement shows more clearly the trajectory or column covariance.

shown in Fig. 5.6. Links between the nodes in the field indicate dependence relationships, where each link is associated with a value that characterizes one variable's influence on the other. The GMRF interpretation elucidates the conditional independence relations between the entries of $\mathbf{z}$ implied by the MND distribution.

In particular, two entries, z_i and z_j, are conditionally independent given all other z_k : $k \neq i$ and $k \neq j$ if and only if $Q_{i,j} = 0$, i.e., there is a zero in the corresponding entry of the precision matrix. In the corresponding graph structure, edges or links will only be present for those entries for which $Q_{i,j} \neq 0$.

5.4.1 Kronecker Markov Random Fields

For more convenient indexing, we can similarly define a Kronecker Markov random field as the direct extension of GMRFs to the case of Kronecker structured covariance matrices (and


GMRFs
Precision Matrices
t=1
t=2
t=3
t=4
p=1
p=2
p=3
p=4
Σ^{-1}
$\mathbf{I}_{3P} \otimes \Sigma^{-1}$
Δ^{-1}
$\Delta^{-1} \otimes \mathbf{I}_F$
$\Delta^{-1} \otimes \Sigma^{-1}$

(a) AR-1 Trajectory Model (b) Point Distribution Shape Model (c) Kronecker Shape-Trajectory Model

Figure 5.6: Markov random field and precision matrices for shape, trajectory, and shape-trajectory models.

equivalently, the matrix normal distribution).

In particular, a point's coordinate at time t_i and shape dimension s_i will be conditionally independent from a coordinate at time t_j and s_j (given all other variables) iff $\mathbf{\Delta}^{-1}_{s_i,s_j} = 0$ or $\mathbf{\Sigma}^{-1}_{t_i,t_j} = 0$ (or both). The MND model therefore imposes spatiotemporal conditional independence for all coordinates or time instants that are conditionally independent in either the shape or trajectory precision matrices. In fact, in the following, we show that by relaxing the independence assumptions of the shape and trajectory models, the natural combination of these priors results in an approximately *Kronecker* spatiotemporal covariance matrix. This result follows from simultaneously applying the shape and temporal priors:

1. At any time instant t, the configuration of points follows a zero-mean PDM:

$$\mathbf{z}^t = \mathbf{B}\mathbf{c}^t \quad \text{with} \quad \mathbf{c}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{5.31}$$

   where $\mathbf{z}^t$ is the t^{th} row of matrix $\mathbf{Z}$ arranged as a column vector, $\mathbf{\Delta} = \mathbf{B}\mathbf{B}^T \in \mathbb{R}^{D\times D}$ is the shape covariance and $\mathbf{c}_t$ a vector of coefficients.

2. The dynamic evolution of the system follows a vector-valued $AR(1)$ process:

$$\mathbf{z}^t = \boldsymbol{\phi}\mathbf{z}^{t-1} + \mathbf{v}^t \quad \text{with} \quad \mathbf{v}^t \sim \mathcal{N}(\mathbf{0}, \mathbf{K}) \tag{5.32}$$

   where $\boldsymbol{\phi} \in \mathbb{R}^{D\times D}$ describes the temporal dynamics, and $\mathbf{K} \in \mathbb{R}^{D\times D}$ describes i.i.d. Gaussian deviations from the model. Later, we will demonstrate that the matrix $\mathbf{K}$ is a scaled version of the spatial covariance matrix of the corresponding PDM.

From Eq. (5.32), using the Markov property and the chain rule we can write the joint distribution over a set of frames $[1, \ldots, F]$ as,

$$\begin{aligned} p(\mathbf{Z}) = p(\cdots, \mathbf{z}^{t-1}, \mathbf{z}^{t}, \mathbf{z}^{t+1}, \mathbf{z}^{t+2}, \cdots) = \\ \cdots p(\mathbf{z}^{t+1}|\mathbf{z}^{t}) p(\mathbf{z}^{t}|\mathbf{z}^{t-1}) p(\mathbf{z}^{t-1}|\mathbf{z}^{t-2}) \cdots, \end{aligned}$$

where each conditional distribution is independently Gaussian and is given by

$$p(\mathbf{z}^t|\mathbf{z}^{t-1}) = \frac{1}{C} \exp\left(-\frac{1}{2}(\mathbf{z}^t - \boldsymbol{\phi}\mathbf{z}^{t-1})^T \mathbf{K}^{-1} (\mathbf{z}^t - \boldsymbol{\phi}\mathbf{z}^{t-1})\right). \tag{5.33}$$

Taking the negative log-likelihood of the joint model, the general form is

$$\begin{aligned} -\log(p(\mathbf{Z})) = \cdots &+ (\mathbf{z}^{t+1} - \boldsymbol{\phi}\mathbf{z}^{t})^T \mathbf{K}^{-1} (\mathbf{z}_{t+1} - \boldsymbol{\phi}\mathbf{z}^{t}) \\ &+ (\mathbf{z}^{t} - \boldsymbol{\phi}\mathbf{z}^{t-1})^T \mathbf{K}^{-1} (\mathbf{z}^{t} - \boldsymbol{\phi}\mathbf{z}^{t-1}) \\ &+ \cdots - \log(C), \end{aligned}$$

for some normalizing constant C. Letting $\mathbf{J} = (\mathbf{K}^{-1} + \boldsymbol{\phi}^T\mathbf{K}^{-1}\boldsymbol{\phi})$, and $\mathbf{H} = -\boldsymbol{\phi}^T\mathbf{K}^{-1}$, the negative log-likelihood of the set of frames can be written as a block tri-diagonal quadratic form,

$$\begin{pmatrix} \vdots \\ \mathbf{z}^{t+1} \\ \mathbf{z}^{t} \\ \mathbf{z}^{t-1} \\ \vdots \end{pmatrix}^T \begin{pmatrix} \ddots & & & & \\ \mathbf{H}^T & \mathbf{J} & \mathbf{H} & & \\ & \mathbf{H}^T & \mathbf{J} & \mathbf{H} & \\ & & \mathbf{H}^T & \mathbf{J} & \mathbf{H} \\ & & & & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ \mathbf{z}^{t+1} \\ \mathbf{z}^{t} \\ \mathbf{z}^{t-1} \\ \vdots \end{pmatrix}. \tag{5.34}$$

When the transition matrix tends to $\boldsymbol{\phi} \to \mathbf{I}$, then $\mathbf{J} \approx 2\mathbf{K}^{-1}$, and $\mathbf{H} \approx -\mathbf{K}^{-1}$, and, in the limit, we can re-write the negative log-likelihood as $\text{vec}(\mathbf{Z}^T)^T \Psi^{-1} \text{vec}(\mathbf{Z}^T)$ with

$$\Psi^{-1} \approx \alpha \left[\underbrace{\begin{pmatrix} \star & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \vdots \\ \vdots & & & & \\ 0 & 0 & 0 & -1 & \star \end{pmatrix}}_{\boldsymbol{\Sigma}^{-1}} \otimes \mathbf{K}^{-1} \right], \tag{5.35}$$

where the entries marked $\star$ depend on the boundary conditions chosen for the first and last time instants (see [Strang 1999]), and α is a normalizing constant. This is a generalization of the result of Ahmed, Natarajan, and Rao (see Sect. **??**), but for multivariate $AR(1)$ processes:

the optimal (in the L2 sense) spatiotemporal basis for a vector-valued $AR(1)$ process with constant noise covariance will be given by the eigenvectors of Eq. (5.35) when $\boldsymbol{\phi} \to \mathbf{I}$.

In general, we will say that a vector-valued $AR(1)$ process $\mathbf{z}^t$ with $\boldsymbol{\phi} \approx \mathbf{I}$ has a spatiotemporal covariance Ψ that is approximately Kronecker-Markov, with covariance $\Psi \approx \boldsymbol{\Sigma} \otimes \mathbf{K}$, for some matrix $\mathbf{K}$, or equivalently $\boldsymbol{\Phi} \approx \mathbf{K} \otimes \boldsymbol{\Sigma}$ for the corresponding column-major arrangement.

The relationship between $\mathbf{K}$ and the PDM shape covariance $\boldsymbol{\Delta}$ can be derived by taking the marginal probability $p(\mathbf{z}^t)$, i.e., the probability of observing a particular shape at time t after marginalizing out all other time instants. In this case, we can see that the marginal shape distribution is

$$\mathbf{z}^t \sim \mathcal{N}(\mathbf{0}, \Sigma_{t,t}\mathbf{K}). \tag{5.36}$$

From the assumption that individual frames follow a PDM distribution (Eq. (5.31)), we can conclude that $\mathbf{K} = \Sigma_{t,t}^{-1}\boldsymbol{\Delta}$, i.e., $\mathbf{K}$ is equal to the shape covariance up to a constant scale factor[6]. Note, however, that $\boldsymbol{\Sigma}$ and $\boldsymbol{\Delta}$ are not uniquely identifiable since $\boldsymbol{\Sigma} \otimes \boldsymbol{\Delta} = \frac{1}{\alpha}\boldsymbol{\Sigma} \otimes \alpha\boldsymbol{\Delta}$ for any non-zero scalar; we will therefore assume that we can find a scale factor such that $\Sigma_{t,t} = 1$.

We define the Kronecker-Markov prior as a GMRF with a precision matrix $\boldsymbol{\Phi}^{-1} = \boldsymbol{\Delta}^{-1} \otimes \boldsymbol{\Sigma}^{-1}$, where $\boldsymbol{\Sigma}^{-1}$ is the DCT-2 matrix. Its connectivity diagram is shown in Fig. 5.6 (c), where we link a point p_1 at time instant t_1 and point p_2 at time instant t_2 iff there exists a link between p_1 and p_2 in the shape MRF, *and* a link between t_1 and t_2 in the temporal MRF. This can be generalized to higher order Markov chain models by allowing arbitrary temporal precision matrices $\boldsymbol{\Sigma}^{-1}$.

This is illustrated in Fig. 5.6 for the particular case of an AR(1) trajectory covariance matrix and an arbitrary shape precision matrix. The GMRFs for the shape-only and trajectory-only models can be visualized alongside the MRF of the MND. It is easy to see how spatiotemporal links in the MRF for the spatiotemporal distribution can only be present when links already exist in *both* the respective trajectory and shape MRFs. Further, the "weight" associated with each link is a product of the two corresponding precision matrix entries.

5.4.2 Factoring Spatial and Temporal Correlations

Which correlations have we lost by assuming the Kronecker MRF structure on the spatiotemporal dependencies? Let us return to the block Toeplitz matrix that we determined in (5.25), where we found that we can express the spatiotemporal covariance between points p_1 and

[6] In practice, the first and last time instants can be scaled differently depending on the boundary conditions, see Eq. (5.35).

p_2 as a function $\mathrm{c}_{p_1 p_2}[f]$ of the lag f. The canonical representation of these functions is as a covariance matrix that explicitly holds the value for each pair of points at each lag f. In functional form,

$$\mathrm{c}_{p_1 p_2}[f] = \sum_{k=1}^{F} \Delta_f^{p_1,p_2} \delta_k[f], \tag{5.37}$$

where $\delta_k[x]$ is a Kronecker delta centered at k (1 when $x = k$, and 0 elsewhere). We can think of this as a basis function decomposition, where the basis functions δ_k are multiplied by coefficients that are fully specified by the multi-array $\boldsymbol{\Delta} \in \mathbb{R}^{F \times 3P \times 3P}$. This multi-array can be seen in Figure 5.4, and corresponds to the first row of $3P \times 3P$ block matrices, where each block corresponds to a different lag. Clearly, the coefficients for this functional form of the covariance in the canonical basis are highly correlated, motivating us to find a more suitable basis to represent the covariance. We will use principal component analysis on the set of covariance functions to find an uncorrelated basis $\gamma_k[f]$ for the autocovariance functions, and the corresponding set of transformed coefficients $\tilde{\boldsymbol{\Delta}}$,

$$\mathrm{c}_{p_1 p_2}[f] = \sum_{k=1}^{F} \tilde{\Delta}_f^{p_1,p_2} \gamma_k[f]. \tag{5.38}$$

Dimensionality reduction in this space corresponds to a decomposition of the 3-way array $\boldsymbol{\Delta}$, but we can approach the problem more simply by rearranging the data into a matrix. We will write $\mathbf{A} \in \mathbb{R}^{(3P)^2 \times F}$, where each column contains a vectorized form of the matrices $\boldsymbol{\Delta}_{1,f}$, so that $\mathbf{A} = \Big(\mathrm{vec}(\boldsymbol{\Delta}_{1,1}) \cdots \mathrm{vec}(\boldsymbol{\Delta}_{1,F})\Big)$. The optimal (in the Frobenious sense) rank-K (with $K \leq F$) approximation of this matrix can then be written using a singular value decomposition as

$$\mathbf{A} \approx \sum_{k=1}^{K} d_k \mathbf{u}_k \mathbf{v}_k^{\mathsf{T}}, \tag{5.39}$$

with $\mathbf{u}_k \in \mathbb{R}^{(3P)^2}$ and $\mathbf{v}_k \in \mathbb{R}^F$, and $d_1 > d_2 > \cdots > d_K$ a set of positive scalars. We can see by inspection that a rank one approximation of the multi-array corresponds to a factorization such that

$$\mathrm{c}_{p_1 p_2}[f] \approxeq \tilde{\Delta}_{p_1,p_2} \gamma[f], \tag{5.40}$$

where $\tilde{\boldsymbol{\Delta}} = \mathrm{unvec}(\mathbf{u}_1)$ and $\gamma[f] = v_1^f$ is the f entry into vector $\mathbf{v}_1$. The key observation is that this form factors spatiotemporal correlations into a shape covariance, $\tilde{\boldsymbol{\Delta}}$, independent of time, and a temporal autocovariance function, $\gamma[f]$, independent of shape. Figure 5.7 shows the percent cumulative variance captured by a rank-K approximation of Eq. (5.39) on the face sequence dataset. The rank-1 approximation already captures 80% of the variance. We will later see that this holds for other types of spatiotemporal point clouds as well, including full body motion.

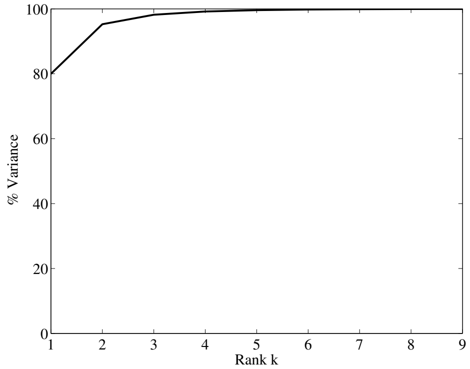


Figure 5.7: Percent variance captured by the rank-K approximation of Eq. (5.39).

Visually, the approximation can be seen in Figure 5.8, which shows the components $\tilde{\mathbf{\Delta}}$ and $\gamma[f]$ for the face sequence data, as well as the corresponding rank-1 reconstruction of the first row of the spatiotemporal covariance matrix.

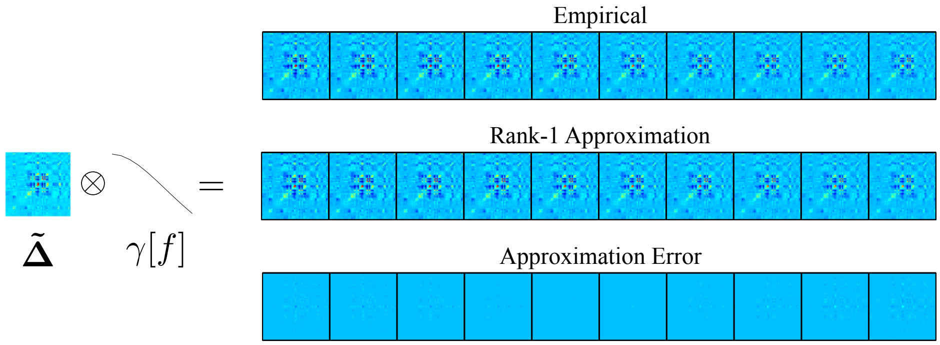


Figure 5.8: Reconstruction of the first row of the covariance in Figure 5.4 using a rank-1 approximation. $\tilde{\mathbf{\Delta}}$ captures shape correlations, and $\gamma[f]$ captures temporal correlations.

The functional form of the autocovariance $\gamma[f]$ merits further investigation. Figure 5.9 shows $\gamma[f]$ for a longer window size, using $F = 30$. It decays with the lag f, meaning that samples separated further in time are less correlated. Visually, we can see that this decay is approximately geometrical and therefore similar to the autocovariance function of an autoregressive order-1 or AR(1) process:

This suggests an analytical expression for the temporal component of the spatiotemporal covariance. Mathematically, a one-dimensional AR(1) random process [Rue and Held 2005] is characterized by the recurrence relation

$$y_t = \phi y_{t-1} + \epsilon, \tag{5.41}$$

where ϵ is Gaussian noise of variance σ^2 and ϕ controls the degree of correlation between

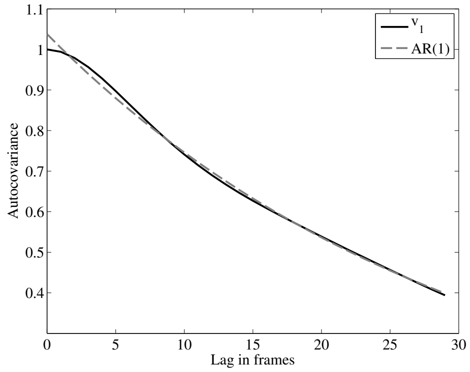


Figure 5.9: The temporal autocovariance of the rank-1 approximation, $\gamma[f] = v_1^f$, plotted as a function of the lag f. The dashed line is the autocovariance function of the best approximating AR(1) process.

neighboring samples. Such a process has the following autocovariance function,

$$\eta[f] = \frac{\phi^f}{(1-\phi^2)}\sigma^2. \tag{5.42}$$

For example, the fit in Figure 5.9 was produced with parameters $\phi = 0.97$ and $\sigma = 0.5$.

To briefly recap:

1. The prior distribution of a randomly sampled spatiotemporal sequence tends to weak-sense stationary if the captured sequence is short compared to the lifetime of the object.
2. Most of the spatiotemporal correlations can be factored into a spatial component, $\boldsymbol{\Delta}$, and a temporal autocovariance, $\gamma[f]$. This statement can be made precise using the rank-K approximation of Eq. (5.39).
3. The temporal autocovariance function $\gamma[f]$, may be approximated analytically by an AR(1) process for stationary distributions.

We will now extend the analysis to a particular kind of non-stationary distributions by considering the spatial and temporal factorization in the original dimensionality of the problem.

5.4.3 Tensoring the Spatiotemporal Factorization

Recall that the approximation of Eq. 5.39 can be seen as a decomposition of the multi-dimensional array or tensor $\boldsymbol{\Delta} \in \mathbb{R}^{F \times 3P \times 3P}$, where we vectorized the spatiotemporal data for algebraic manipulation. However, we can carry out an equivalent statistical treatment of the objects in their native dimensionality. The spatiotemporal data we are studying was originally presented as a matrix $\mathbf{X} \in \mathbb{R}^{3P \times F}$, with $3P$ shape dimensions (3D points) and F

frames. Assuming zero-mean, the covariance of $\mathbf{X}$ is the expectation of its outer product:

$$E[\mathbf{X} \otimes \mathbf{X}^{\mathsf{T}}], \tag{5.43}$$

which can be thought of as a $3P \times 3P \times F \times F$ tensor. This might seem daunting at first, but simply expresses the fact that we are computing a covariance between all pairs of variables in our original data, of which we have $(3PF)$—yielding a total of $(3PF)^2$ entries in the covariance: all pairs of point coordinates at every frame. [7] In fact, Figure 5.4 is more intuitively understood in terms of this tensor: each $3P \times 3P$ block represents correlations between pairs of point coordinates, and there are $F \times F$ of these (one for each pair of frames).

The factorization into spatial and temporal components of Eq. (5.39) can now be seen in a more general light as the rank-1 decomposition of an order-4 tensor into the outer product of two order-2 tensors (matrices). This decomposition can be formalized using the Kronecker Product (KP) SVD [Van Loan 2000],

$$E[\mathbf{X} \otimes \mathbf{X}^{\mathsf{T}}] = \sum_{k=1}^{r_{KP}} \sigma_k^2 \mathbf{U}_k \otimes \mathbf{V}_k, \tag{5.44}$$

where it can be shown that there exist $\mathbf{U}_1, \cdots, \mathbf{U}_{r_{KP}} \in \mathbb{R}^{F \times F}$ and $\mathbf{V}_1, \cdots, \mathbf{V}_{r_{KP}} \in \mathbb{R}^{3P \times 3P}$, with scalars $\sigma_1 > \sigma_2 > \cdots \sigma_{r_{KP}} > 0$, where r_{KP} is the so-called KP-rank of the tensor. The rank-1 decomposition is called the nearest KP-matrix, and for our original spatiotemporal covariance matrix can be written as,

$$\operatorname{cov}(\operatorname{vec}(\mathbf{X})) \approxeq \mathbf{\Sigma} \otimes \mathbf{\Delta}, \tag{5.45}$$

where we have factored the spatiotemporal covariance $\operatorname{cov}(\mathbf{X})$ (i.e., an order-4 tensor measuring correlations between point coordinates at different frame pairs) into two order-2 tensors:

1. $\mathbf{\Sigma} \in \mathbb{R}^{F \times F}$, which captures correlations across pairs of frames (independently of the spatial dimensions), and
2. $\mathbf{\Delta} \in \mathbb{R}^{3P \times 3P}$, which captures correlations across pairs of points (independently of the temporal dimensions).

Note that this decomposition does not require the process to be stationary. However, if the process is indeed stationary, the relationship between $\mathbf{\Sigma}$ and the stationary autocovariance

[7] Note that if $\mathbf{X} \otimes \mathbf{X}^{\mathsf{T}}$ is seen as a Kronecker product matrix with size $3PF \times 3PF$, the entries have a different ordering from the entries in $\operatorname{cov}(\operatorname{vec}(\mathbf{X}))$ seen previously.

function $\gamma[f]$ is

$$\mathbf{\Sigma} = \text{toeplitz}\left(\left[\gamma[0], \cdots, \gamma[F-1]\right]\right), \tag{5.46}$$

where the operator toeplitz$(\cdot)$ creates a Toeplitz matrix using the vector argument—in this case, the function $\gamma[f]$ evaluated at $0, \cdots, F-1$.

5.4.4 Parameterization of the Spatiotemporal Mean and Covariance

We began by computing the first- and second-order moments of our spatiotemporal samples and derived a series of constraints specific to spatiotemporal sequences that followed logically or empirically, namely:

5.4.4.1 Mean

We saw that we may model the mean as stationary. The full mean requires $3P \times F$ free variables, whereas a stationary mean requires only $3P$ parameters (see Eq.(5.20)).

5.4.4.2 Covariance

We derived a set of constraints and approximations for the covariance statistics of spatiotemporal data. The number of parameters for each of the covariance models discussed is summarized below:

Covariance type	Number of parameters	Equation
Full	$\frac{(3PF)^2-3PF}{2}$	(5.23)
Stationary process	$F \cdot \frac{(3P)^2-3P}{2}$	(5.24)
Rank-1 Stationary	$\frac{(3P)^2-3P}{2} + F$	(5.40)
Rank-1 Kronecker	$\frac{(3P)^2-3P}{2} + \frac{(F)^2-F}{2}$	(5.45)

For the remainder, we will focus on the rank-1 Kronecker covariance model of Eq. (5.45) because of its elegant interpretation as a matrix normal distribution. The important takeaway here is that, for typical values of F and P, the number of parameters for this model is vastly smaller than that of a full covariance matrix. This will be a useful property when estimating model parameters from observations with missing or corrupted data, and will be discussed in Chapter 5.4.4.2.

5.5 3D Reconstruction of Dynamic Scenes

In theory, dynamic 3D reconstruction is often an ill-posed problem because of *projection loss* due to the imaging of 3D information to 2D. In practice, a number of additional sources of missing data arise. First, occlusions, self-occlusions, and imaging artifacts (such as motion blur) can cause *detection loss* where points of interest are simply not detected in particular frames. Second, if points are not re-associated to their earlier detection, the system may break one trajectory into two separate trajectories, causing *correspondence loss.* While missing data issues are present in static 3D reconstruction, they are of greater significance in dynamic 3D reconstruction, as the observation system has only one opportunity to directly measure information about the structure at a particular time instant.

In this section, we apply the KMRF model to recover reconstructions of dynamic scenes in the presence of missing data or camera projection. The model as presented here applies to any dynamic 3D reconstruction problem, including nonrigid structure from motion, stereo, and multi-view dynamic 3D reconstruction.

5.5.1 Modeling Translating Nonrigid Objects

We model the time-varying structure $\mathbf{X} \in \mathbb{R}^{F\times 3P}$ as the sum of a mean component $\mathbf{M}$ and a residual non-rigid component $\mathbf{Z}$,

$$\mathbf{X} = \mathbf{M} + \mathbf{Z}. \tag{5.47}$$

While this does not reduce the number of variables to estimate, this decomposition will allow us to set sensible priors over the individual components.

5.5.1.1 Modeling the Non-rigid Component Z

From Eq. (5.35), the dynamic 3D structure will have an approximately Kronecker structured covariance matrix, and

$$\mathrm{vec}(\mathbf{Z}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Delta} \otimes \boldsymbol{\Sigma}). \tag{5.48}$$

Equivalently, this model corresponds to a matrix normal distribution [Dutilleul 1999; Allen and Tibshirani 2010] over the non-rigid component $\mathbf{Z}$, which we can write as

$$\mathbf{Z} \sim \mathcal{MN}(\mathbf{0}, \boldsymbol{\Sigma}, \boldsymbol{\Delta}), \tag{5.49}$$

where $\mathcal{MN}$ denotes an MND with row covariance $\boldsymbol{\Delta}{=}\mathbf{B}\mathbf{B}^T$ (describing shape correlations) and column covariance $\boldsymbol{\Sigma}{=}\boldsymbol{\Theta}\boldsymbol{\Theta}^T$ (describing trajectory correlations). This formulation exposes the relationship to bilinear spatiotemporal basis models [Akhter et al. 2012], with

$$\mathbf{Z} = \mathbf{\Theta}\mathbf{C}\mathbf{B}^T, \tag{5.50}$$

where $\mathbf{C} \in \mathbb{R}^{F\times 3P}$ is a matrix of mixing coefficients, $\mathbf{B} \in \mathbb{R}^{3P\times 3P}$ is a complete shape basis and $\mathbf{\Theta} \in \mathbb{R}^{F\times F}$ a complete trajectory basis such that $\mathbf{B} = \tilde{\mathbf{B}}\mathbf{W}_b$ and $\mathbf{\Theta} = \tilde{\mathbf{\Theta}}\mathbf{W}_t$ where $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{\Theta}}$ are orthonormal and $\mathbf{W}_b, \mathbf{W}_t$ are diagonal weighting matrices. When the distribution over coefficients $\mathbf{C}$ is $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then the distribution over $\mathbf{Z}$ is matrix normal as in Eq. (5.49).

This probabilistic formulation subsumes the bilinear basis models of [Gotardo and Martinez 2011] and [Akhter et al. 2012], where truncated versions of the orthonormal basis matrices $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{\Theta}}$ are used. The limit of Eq. (5.35) explains the effectiveness of using a truncated DCT trajectory basis in a bilinear formulation. In fact, as $\boldsymbol{\phi} \to 1$ (e.g., if we sample at increasing rates) the optimal L2 spatiotemporal basis vectors will be the eigenvectors of $\mathbf{\Delta} \otimes \mathbf{\Sigma}$, which are[8] $\{\tilde{\mathbf{B}}_p \otimes \tilde{\mathbf{\Theta}}_t\}$ for $p \in \{1, \ldots, 3P\}$ and $t \in \{1, \ldots, F\}$. This set corresponds directly to a bilinear basis model [Akhter et al. 2012], with $\mathbf{\Theta}$ the DCT basis.

5.5.1.2 Modeling the Mean Component M

In addition to the non-rigid component $\mathbf{Z}$ described above, we model the rigid shape of the object and its translational motion as a mean component $\mathbf{M}$. This component is

$$\mathbf{M} = \mathbf{1}_F\mathbf{m}_{\text{shape}} + \mathbf{M}_{\text{trans}}\mathbf{P}_{\text{trans}}, \tag{5.51}$$

where the zero-centered mean 3D shape is $\mathbf{m}_{\text{shape}} \in \mathbb{R}^{1\times 3P}$, and the mean 3D trajectory is $\mathbf{M}_{\text{trans}} \in \mathbb{R}^{F\times 3}$ (containing the per-frame translation of the object), where $\mathbf{P}_{\text{trans}} = \text{blkdiag}(\mathbf{1}_P^T; \mathbf{1}_P^T; \mathbf{1}_P^T) \in \mathbb{R}^{3\times 3P}$, with $\mathbf{1}_P$ denoting a column vector of ones of size P, and blkdiag produces a block diagonal matrix.

We do not have a preferred shape of objects, and so we do not set a prior over the mean shape $\mathbf{m}_{\text{shape}}$. However, the translational motion $\mathbf{M}_{\text{trans}}$ is necessarily smooth, and we will therefore favor smooth trajectories of the object using the trajectory prior[9]:

$$\mathbf{M}_{\text{trans}} \sim \mathcal{MN}(\mathbf{0}, \mathbf{\Sigma}, \mathbf{I}_3), \tag{5.52}$$

where the row covariance $\mathbf{I}_3$ reflects that there are no a priori correlations between the x, y, and z components of motion.

[8] See [Laub 2004] for properties of the Kronecker product.

[9] We use the same trajectory covariance $\mathbf{\Sigma}$ as for the non-rigid component, but, more generally, a different covariance matrix could be used.

5.5.2 Relationship to Previous Work

The model over dynamic 3D structure described in this paper can be related to shape, trajectory, and shape-trajectory representations used in prior art [Gotardo and Martinez 2011; Bregler et al. 2000; Cootes et al. 1995b; Sidenbladh et al. 2000a; Xiao et al. 2004; Torresani et al. 2008; Akhter et al. 2008b; Valmadre and Lucey 2012] (see Table 5.1).

In the following, consider the MND prior over point cloud data $\mathbf{X} \sim \mathcal{MN}(\mathbf{M}, \boldsymbol{\Delta}, \boldsymbol{\Sigma})$ with known distribution parameters $\mathbf{M}$, $\boldsymbol{\Delta}$, and $\boldsymbol{\Sigma}$.

Trajectory Methods. The MND describes a joint shape-trajectory distribution, but the marginal distribution it induces for a particular trajectory $\mathbf{x}_p$ (a column p of $\mathbf{X}$) independent of all other points corresponds to a basis representation over trajectories, as described by Sidenbladh et al. [Sidenbladh et al. 2000a]. The marginal distribution is $\mathbf{x}_p \sim \mathcal{N}(\mathbf{M}_p, \Delta_{p,p}\boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}{=}\boldsymbol{\Theta}\boldsymbol{\Theta}^T$ is the trajectory covariance matrix, and $\Delta_{p,p}$ loosely corresponds to the mass of each point. This expression is equivalent to the *filtering* solution proposed by Valmadre and Lucey [Valmadre and Lucey 2012], who observe that a combination of first and second-order differences fit natural motions well. See also [Salzmann and Urtasun 2011] for a physically-based formulation of the same model.

Shape Methods. The marginal distribution of a particular shape $\mathbf{x}^t$ (a row t of $\mathbf{X}$ arranged as a column) independent of all other time instants corresponds exactly to shape-only distributions used in prior art, such as the Point Distribution Model (PDM) of Cootes et al. [Cootes et al. 1995b], and the shape basis model of Torresani et al. [Torresani et al. 2008]. It follows from the matrix normal model that $\mathbf{x}^t \sim \mathcal{N}(\mathbf{M}^t, \Sigma_{t,t}\boldsymbol{\Delta}),$, where $\boldsymbol{\Sigma}_{t,t}$ is the entry (t,t) in $\boldsymbol{\Sigma}$ and $\boldsymbol{\Delta}{=}\mathbf{B}\mathbf{B}^T$ is the shape covariance matrix. An equivalent shape covariance is estimated using PCA by Cootes et al., where $\mathbf{B}$ is a shape basis [Bregler et al. 2000; Xiao et al. 2004; Akhter et al. 2008b; Torresani et al. 2008].

Similarly, the PND model of Lee et al. [Lee et al. 2013b] is related in the same way save for two distinctions: firstly, the shape covariance in the PND model is restricted to exclude the subspace of small-angle rotations, scaling, and translation of the mean shape (i.e., adding the constraints that $\mathbf{P}_N^T\boldsymbol{\Delta} = \mathbf{0}$, where $\mathbf{P}_N$ is that subspace, $||\mathbf{m}_{\text{shape}}|| = 1$ and $\mathbf{m}_{\text{shape}}\mathbf{1} = 0$, see [Lee et al. 2013b]); secondly, the shape covariance is rotated into the coordinate system of every frame (i.e., the PND models rotated shapes, whereas we model aligned shapes). While the Procrustean constraints can be incorporated into the MND model, this would prohibit the convex solution presented in Sect. 5.6.2. Similarly, modeling rotated shapes would make the $\mathbf{H}$ and $\mathbf{J}$ matrices in Eq. 5.34 become time-dependent, and we would loose the compact Kronecker expression of the covariance.

Table 5.1: Comparison of linear methods for structure reconstruction. The symbols are explained in Section 5.5.

	Truncation	**Probabilistic**	**Low Rank**
Shape	[Bregler et al. 2000] $\mathbf{Z} = \mathbf{\Omega}\tilde{\mathbf{B}}^T$	[Torresani et al. 2008] $\mathbf{Z} \sim \mathcal{MN}(\mathbf{0}, \mathbf{I}, \mathbf{B}\mathbf{B}^T)$	[Dai et al. 2012] $\lVert\mathbf{Z}\rVert_*$
Trajectory	[Akhter et al. 2008b] $\mathbf{Z} = \tilde{\mathbf{\Theta}}\mathbf{A}$	[Valmadre and Lucey 2012] $\mathbf{Z} \sim \mathcal{MN}(\mathbf{0}, \mathbf{\Theta}\mathbf{\Theta}^T, \mathbf{I})$	
Shape-Trajectory	[Gotardo and Martinez 2011] $\mathbf{Z} = \tilde{\mathbf{\Theta}}\mathbf{C}\tilde{\mathbf{B}}^T$	(KMRF) $\mathbf{Z} \sim \mathcal{MN}(\mathbf{0}, \mathbf{\Theta}\mathbf{\Theta}^T, \mathbf{B}\mathbf{B}^T)$ or $\lVert\mathbf{\Theta}^+\mathbf{X}\mathbf{P}_\perp\rVert_*$	

Spatiotemporal Methods. The model we present is a probabilistic formulation of the shape-trajectory basis models described in [Gotardo and Martinez 2011; Akhter et al. 2012]. These models describe spatiotemporal sequences as a linear combination of the outer product of a reduced set of trajectory basis vectors and a set of shape basis vectors. They rely on truncation of the basis to achieve compaction, while the probabilistic MND model describes the relative variance of each spatiotemporal mode with the weighting matrices $\mathbf{W}_t$ and $\mathbf{W}_b$. Additionally, the MND allows us to compute a confidence bound on the imputed missing data. We visualize this distribution in Fig. 5.17 on a facial motion capture sequence from [Akhter et al. 2012].

As with the shape-only model, the PND Markov process (PMP) [Lee et al. 2014] is very closely related. With the same distinctions about the rotated coordinate space for the shape covariance discussed above, the Markov PND process is essentially the same as the Kronecker-Markov process in Sect. 5.4.1 but with a stationarity constraint $\Phi = \alpha\mathbf{I}$ rather than $\alpha \to 1$. The PMP model is therefore more general, but the trade-off is a non-convex optimization that requires careful initialization and explicitly solving for the parameter α.

5.6 Convex MAP Reconstruction for the Kronecker-Markov Prior

In this section, we derive convex estimation procedures to recover the most likely dynamic structure $\mathbf{X}$ given the measurements $\mathbf{y}$ using the Kronecker-Markov shape-trajectory prior, $p(\mathbf{y}|\mathbf{X})p(\mathbf{X}) = p(\mathbf{y}|\mathbf{Z}, \mathbf{M})p(\mathbf{Z})p(\mathbf{M})$, where the non-rigid and mean components are distributed according to Eqs. (5.49) and (5.52).

5.6.1 Known Distribution Parameters

With known covariance matrices $\mathbf{\Sigma}$ and $\mathbf{\Delta}$, the negative log-likelihood of the MND is quadratic, and inference under an MND prior is straightforward and can be posed as a least-squares problem:

$$\begin{aligned}\underset{\mathbf{M},\mathbf{Z}}{\operatorname{argmax}}\ & p(\mathbf{y}|\mathbf{Z},\mathbf{M})p(\mathbf{Z})p(\mathbf{M}) = \\ & \underset{\mathbf{M},\mathbf{Z}}{\operatorname{argmin}}\, \sigma^{-2}||\mathbf{y}-\mathbf{O}\,\mathrm{vec}(\mathbf{M}+\mathbf{Z})||_F^2 \\ & \qquad + \underbrace{\mathrm{tr}\left[\mathbf{\Delta}^{-1}\mathbf{Z}^T\mathbf{\Sigma}^{-1}\mathbf{Z}\right]}_{-\log(p(\mathbf{Z}))+c_1} \\ & \qquad + \lambda \underbrace{\mathrm{tr}\left[\mathbf{M}_{\mathrm{trans}}{}^T\mathbf{\Sigma}^{-1}\mathbf{M}_{\mathrm{trans}}\right]}_{-\log(p(\mathbf{M}))+c_2}, \end{aligned} \tag{5.53}$$

where λ is a scaling factor related to the mass of the object and the variance of its translational motion.

5.6.2 Unknown Distribution Parameters

We can approximate the trajectory distribution using a DCT−2 matrix $\mathbf{\Sigma}^{-1}$ (Sect. 5.4.1). However, the PDM shape distribution $\mathbf{\Delta}$ covariance depends on the object and is typically unknown a priori, and therefore needs to be estimated as well. The least-squares problem of Eq. (5.53) now becomes bilinear in $\mathbf{Z}$ and $\mathbf{\Delta}$, and seemingly non convex.

In the following, we show that there exists a convex solution when we set a hierarchical Wishart covariance [Rao 1973] prior over $\mathbf{\Delta}$. Using the bilinear parameterization (Eq. (5.50)), $\mathbf{Z} = \mathbf{\Theta}\mathbf{C}\mathbf{B}^T$,

$$\begin{aligned} p(\mathbf{X}|y) = p(\mathbf{C},\mathbf{B},\mathbf{M}|\mathbf{y}) \propto & \\ & p(\mathbf{y}|\mathbf{C},\mathbf{B},\mathbf{M})p(\mathbf{B}|\mathbf{C})p(\mathbf{C})p(\mathbf{M}). \end{aligned} \tag{5.54}$$

In this parameterization, $p(\mathbf{B}|\mathbf{C})$ is the only prior that remains to be specified.

To obtain a convex solution, we assume that $p(\mathbf{B}|\mathbf{C}) = p(\mathbf{B})$, i.e., the distribution over shape covariance is independent of the particular shape configurations observed. We use a normal prior over the entries of $\mathbf{B} \sim \mathcal{MN}(\mathbf{0}, \mathbf{I}_{3P}, \mathbf{I}_{3P})$, equivalent to a Wishart prior over the covariance $\mathbf{\Delta}$. Intuitively, this captures the low-rank characteristic of shape covariance matrices: that the singular values of the covariance matrix should decrease rapidly (see Sect. 5.8.1 for an illustration of this prior).

Combining these priors and writing this optimization in terms of the component negative log-likelihoods,

$$\begin{aligned}&\underset{\mathbf{X}}{\operatorname{argmax}}\ p(\mathbf{X}|\mathbf{y}) = \\ &\quad \underset{\mathbf{M},\mathbf{C},\mathbf{B}}{\operatorname{argmin}}\ \sigma^{-2}||\mathbf{y}-\mathbf{O}\,\mathrm{vec}(\mathbf{M}+\mathbf{\Theta}\mathbf{C}\mathbf{B}^T)||_F^2 \\ &\qquad\qquad +\|\mathbf{C}\|_F^2 + \|\mathbf{B}\|_F^2 + \lambda\|\mathbf{\Theta}^+\mathbf{M}_{\text{trans}}\|_F^2 \\ &\quad\ \text{s.t.}\quad \mathbf{X} = \mathbf{M}+\mathbf{\Theta}\mathbf{C}\mathbf{B}^T.\end{aligned} \tag{5.55}$$

This expression is bilinear in $\mathbf{C}$ and $\mathbf{B}$. However, we can transform this bilinear equation into a convex problem using the matrix trace-norm $\|\cdot\|_*$, where $\|\mathbf{R}\|_* = \min_{\mathbf{U},\mathbf{V}}\{\frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2\}$ subject to $\mathbf{R} = \mathbf{U}\mathbf{V}^T$ with $\mathbf{R} \in \mathbb{R}^{m\times n}$, $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{n\times r}$. Mazumder et al. [Mazumder et al. 2010] show the equivalence of these two formulations when $r \geq \mathrm{rank}(\mathbf{R})$, and we can transform Eq. (5.55) by writing[10] $\mathbf{A} = \mathbf{C}\mathbf{B}^T$ and so

$$\begin{aligned}&\underset{\mathbf{M},\mathbf{A}}{\operatorname{argmin}}\ \sigma^{-2}||\mathbf{y}-\mathbf{O}\,\mathrm{vec}(\mathbf{M}+\mathbf{\Theta}\mathbf{A})||_F^2 + \|\mathbf{A}\|_* \\ &\qquad\qquad +\lambda\|\mathbf{\Theta}^+\mathbf{M}_{\text{trans}}\|_F^2\end{aligned} \tag{5.56}$$

By definition, $\mathbf{X}$ is parameterized into a mean shape component, $\mathbf{m}_{\text{shape}} = \frac{1}{F}\mathbf{1}_F^T\mathbf{X}$, a translational component, $\mathbf{M}_{\text{trans}} = \frac{1}{P}\mathbf{X}\mathbf{P}_{\text{trans}}^T$, and the remaining non-rigid component $\mathbf{Z} = \mathbf{\Theta}\mathbf{A}$ where

$$\mathbf{Z} = \mathbf{X} - \mathbf{M} = (\mathbf{I}_F - \frac{1}{F}\mathbf{1}_F\mathbf{1}_F^T)\mathbf{X}\underbrace{(\mathbf{I}_{3P} - \frac{1}{P}\mathbf{P}_{\text{trans}}^T\mathbf{P}_{\text{trans}})}_{\mathbf{P}_\perp}. \tag{5.57}$$

Finally, note that $\mathbf{1}_F$ is in the left null-space of $\mathbf{Z}$, and right null-space of $\mathbf{\Theta}^+$, and so $\mathbf{Z} = \mathbf{\Theta}\mathbf{\Theta}^+\mathbf{Z}$. We can therefore write the change of variables $\mathbf{\Theta}^+\mathbf{X}\mathbf{P}_\perp = \mathbf{A}$ resulting in

$$\begin{aligned}&\underset{\mathbf{X}}{\operatorname{argmin}}\quad \frac{1}{2\sigma^2}\underbrace{||\mathbf{O}\,\mathrm{vec}(\mathbf{X})-\mathbf{y}||_2^2}_{\text{observations}} + \underbrace{||\mathbf{\Theta}^+\mathbf{X}\mathbf{P}_\perp||_*}_{\text{shape-trajectory prior}} \\ &\qquad\qquad + \underbrace{\lambda\frac{1}{2}||\mathbf{\Theta}^+\mathbf{X}\mathbf{P}_{\text{trans}}^T||_2^2}_{\text{translational regularizer}}.\end{aligned} \tag{5.58}$$

Relationship to Trace-norm Methods. The convex MAP minimization of Eq. (5.58). when using a normal prior over $\mathbf{B}$ can be related to the use of the trace-norm in rigid and

[10] In this case, $m = F$, $n = 3P$, and $r = 3P \geq \mathrm{rank}(\mathbf{A}) = \min(F, 3P)$.

non-rigid structure from motion [Angst et al. 2011; Dai et al. 2012]. Note that the shape-trajectory prior component of this objective function is similar to the trace-norm energy term of Dai et al. [Dai et al. 2012], if we set $\boldsymbol{\Theta}^+$ to the identity. This amounts to assuming that every frame is independent and there exist no temporal correlations. The trace-norm method of Dai et al. can then be interpreted as a prior of non-rigid shape that corresponds to $\mathbf{C}\mathbf{B}^T$ with normal priors over coefficients $\mathbf{C}$ and shape basis $\mathbf{B}$. The effect of this is most easily understood for the case of interpolation: frames (rows) for which all points are missing will be set to zero by the $\|\mathbf{X}\|_*$ penalizer. This effect can result in abrupt changes in the reconstruction, and can be seen in the spiked blue curves in Fig. 5.14 (right). Making a similar observation, Angst et al. [Angst et al. 2011] proposed the "generalized trace-norm" for rigid SfM to incorporate temporal smoothness constraints in trace-norm approaches to SfM, resulting in a similar prior term. Compared to the rigid model of Angst et al., our work draws an explicit connection between the row and column spaces of an MND distribution of dynamic 3D structure.

5.7 Optimization via ADMMs

The objective of Eq. (5.58) lends itself to optimization by the Alternating Direction Method of Multipliers (ADMM) [Boyd et al. 2011]. We discuss two cases: the convex solution, where the observation matrix $\mathbf{O}$ is fixed (and consequently, the camera rotation matrices are fixed), and a non-convex procedure that additionally optimizes the rotation matrices.

5.7.1 Fixed Camera Matrices

Let $\mathbf{F} = \mathbf{P}_{\text{trans}} \otimes \boldsymbol{\Theta}^+$ and $\mathbf{G} = \mathbf{P}_{\perp}^T \otimes \boldsymbol{\Theta}^+$. Re-writing Eq. (5.58) in ADMM form (see Boyd et al. [Boyd et al. 2011]),

$$\begin{aligned} \underset{\mathbf{x},\mathbf{z}}{\text{minimize}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{subject to} \quad & \mathbf{G}\mathbf{x} - \mathbf{z} = \mathbf{0} \\ & f(\mathbf{x}) = \frac{1}{2\sigma^2}||\mathbf{O}\mathbf{x} - \mathbf{y}||_2^2 + \frac{\lambda}{2}||\mathbf{F}\mathbf{x}||_2^2 \\ & g(\mathbf{z}) = ||\,\text{unvec}(\mathbf{z})||_* \end{aligned} \tag{5.59}$$

where $\mathbf{x} = \text{vec}(\mathbf{X})$, $f(\mathbf{x})$ and $g(\mathbf{z})$ are convex, unvec$(\cdot)$ reshapes the argument into the desired matrix[11] of size $F \times 3P$, and σ is the observation noise variance. Written in this more general

[11] The development is valid even if $\mathbf{Z}$ is not the same size as $\mathbf{X}$. In particular, the two arrangements described by Dai et al. [Dai et al. 2012], $3F \times P$ and $F \times 3P$, are options to consider.

form, we identify this as a *trace-norm regularized least squares* problem. The ADMM method iteratively updates the variables in two steps, according to the following subproblems:

$$\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \left(f(\mathbf{x}) + \frac{\rho}{2} ||\mathbf{G}\mathbf{x} - \mathbf{z}^k + \mathbf{u}^k||_2^2 \right) \tag{5.60}$$

$$\mathbf{z}^{k+1} = \underset{\mathbf{z}}{\operatorname{argmin}} \left(g(\mathbf{z}) + \frac{\rho}{2} ||\mathbf{G}\mathbf{x}^{k+1} - \mathbf{z} + \mathbf{u}^k||_2^2 \right) \tag{5.61}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \left(\mathbf{G}\mathbf{x}^{k+1} - \mathbf{z}^{k+1} \right) \tag{5.62}$$

with $\mathbf{u}$ the scaled dual variables of the augmented Lagrangian.

The $\mathbf{x}^{k+1}$ update Eq. (5.60) is a least-squares problem and is readily solvable. The $\mathbf{z}^{k+1}$ update Eq. (5.61) involves the nuclear norm and is more difficult to solve, but there exists a closed form solution [Boyd et al. 2011] for problems of the form

$$\operatorname{prox}_{\lambda}(\mathbf{W}) = \underset{\mathbf{Z} \in \mathbb{R}^{m \times n}}{\operatorname{argmin}} \frac{1}{2} ||\mathbf{Z} - \mathbf{W}||_2^2 + \lambda ||\mathbf{Z}||_*. \tag{5.63}$$

Define the *shrinkage* or *soft-thresholding* operator,

$$s_{\lambda}(x) = max(x - \lambda, 0) - max(0, -x - \lambda), \tag{5.64}$$

which we will apply entry-wise to vectors. The solution to this type of problems is then $\operatorname{prox}_{\lambda}(\mathbf{W}) = \mathbf{S}_{\lambda}(\mathbf{W})$, where the *matrix soft-thresholding operator* $\mathbf{S}_{\lambda}(\mathbf{W})$ will be

$$\mathbf{S}_{\lambda}(\mathbf{W}) = \mathbf{U}\boldsymbol{\Sigma}_{\lambda}\mathbf{V}^T, \tag{5.65}$$

where $\mathbf{W} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, and $\boldsymbol{\Sigma}_{\lambda}$ is diagonal with $(\boldsymbol{\Sigma}_{\lambda})_{ii} = s_{\lambda}(\boldsymbol{\Sigma}_{ii})$, the soft-thresholded singular values of $\mathbf{W}$ [Boyd et al. 2011]. The solution to Eq. (5.61) is then

$$\begin{aligned} \mathbf{Z}^{k+1} &\leftarrow \operatorname{prox}_{\frac{1}{\rho}} \left(\operatorname{unvec}(\mathbf{G}\mathbf{x}^{k+1} + \mathbf{u}^k) \right) \\ \mathbf{z}^{k+1} &\leftarrow \operatorname{vec}\left(\mathbf{Z}^{k+1} \right), \end{aligned}$$

5.7.2 Optimizing the Camera Matrices

The ADMM procedure described in the preceding section suggests a simple way to incorporate the estimation of camera (or object) rotation into the optimization, at the cost of making the problem non-convex. For fixed camera matrices, we wrote the observation cost in matrix form as $\frac{1}{2\sigma^2}||\mathbf{O}\mathbf{x} - \mathbf{y}||$, where the matrix $\mathbf{O}$ was assumed to be constant. This expression only appears in the $\mathbf{x}^{k+1}$ update equation, which we can now rewrite more generally as

$$\{\mathbf{x}^{k+1}, \mathbf{p}^{k+1}\} = \underset{\mathbf{x},\mathbf{p}}{\operatorname{argmin}} \left(f(\mathbf{x}, \mathbf{p}) + \frac{\rho}{2} ||\mathbf{G}\mathbf{x} - \mathbf{z}^k + \mathbf{u}^k||_2^2 \right) \tag{5.66}$$

where $\mathbf{p}^k$ is the current estimate of the camera parameters, and we redefine f as

$$f(\mathbf{x}, \mathbf{p}) = \mathcal{P}(\mathbf{x}, \mathbf{p}, \mathbf{y}) + \frac{\lambda}{2} ||\mathbf{F}\mathbf{x}||_2^2 \tag{5.67}$$

where the function $\mathcal{P}$ measures the total observation cost and can be any smooth differentiable function. Without going into the verbose particulars of how to index each observation and its corresponding camera parameters, we parameterize each camera at each time instant as an axis-angle rotation vector and a translation vector (when camera and object motion are not ambiguous). The function $\mathcal{P}$ computes the reprojection error residuals for each of the observed points. We solve the ADMM $\mathbf{x}^{k+1}$ update Eq. (5.66) for both $\{\mathbf{x}, \mathbf{p}\}$ using Levenberg-Marquadt and the ceres-solver [Agarwal et al. 2009] within the ADMM framework. As in [Boyd et al. 2011], we use the previous $\mathbf{x}^k$ and $\mathbf{p}^k$ to warm-start the optimization, and only run a few iterations (5, in our experiments) for each $\mathbf{x}^{k+1}$ update.

5.7.3 Implementation details

The choice of the augmented Lagrangian parameter ρ greatly affects the convergence. We follow the heuristic described in Boyd et al. [Boyd et al. 2011], halving or doubling ρ when the r-norm and s-norm ratios are greater than 2, and we use a maximum of 500 iterations with the stopping criteria described in Section 3.3.1 of [Boyd et al. 2011].

The algorithm relies on two crucial operations: solving a large linear system of equations, and computing an SVD. For typical matrix completion problems, it is usually assumed that computing the SVD is the most time consuming operation. For our problem, it is typically the case that solving the system of equations is more difficult: our problem size is one with $3FP$ unknowns, and the matrices $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{O}$ are not necessarily sparse. The matrices $\mathbf{G}$ and $\mathbf{F}$ do have a limited number of non-zero entries: $\boldsymbol{\Theta}^+$ is the forward differences matrix (each residual involves at most two time instants) and $\mathbf{P}_{\text{trans}}$ has bandwidth $3P$ (each row involves summing over P points at a single time instant).

To solve each linear problem, we therefore either pre-compute the Cholesky-factors for quicker per-iteration solves, or, if the factors are too large to build, we use an iterative solver relying on matrix-vector products. In particular, we use Matlab's `lsqr`: for a problem of size F=2000 frames and P=49 points, there are 294000 unknowns, and each linear solve dominates the ADMM iteration time and takes 4 to 8 seconds on an Intel Core i7 at 2.7GHz,

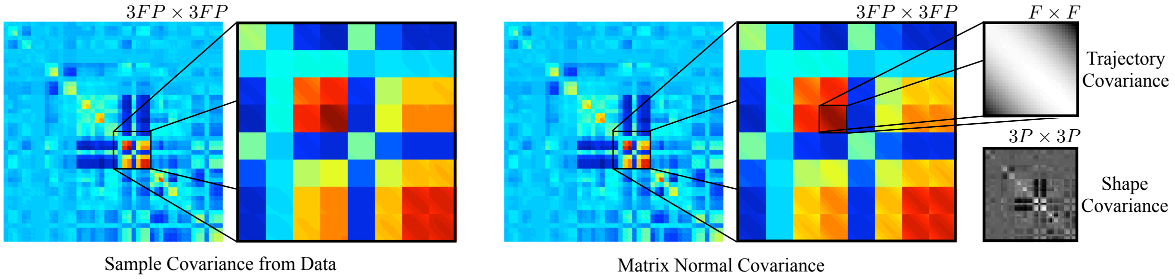


Figure 5.10: Human spatiotemporal point cloud data exhibits a Kronecker structured covariance matrix, allowing us to model the distribution over sequences as matrix normal. (Left) The spatiotemporal covariance computed from 5402 vectorized sequences shows a distinct block structure, highlighted in the inset. (Right) The corresponding covariance of the matrix normal model, where the full $(3FP) \times (3FP)$ matrix is separable into two smaller covariance matrices, the $F \times F$ trajectory (row) and $3P \times 3P$ shape (column) covariances respectively. Here, $F = 30$ frames and $P = 16$ points.

for a total runtime of 38 min. or 1.1 s per frame.

5.8 Evaluation

5.8.1 Validation on Natural Motions

We validate the proposed distribution and the four components of our model by computing statistics on a large set of natural motions. We use the CMU Motion Capture database, where we subsample the data to retain point tracks for 15 joint locations on the body, yielding $N = 5402$ 30-frame sub-sequences $\mathbf{X}_n$ which we also align using Procrustes analysis and center around the mean.

I. Kronecker-Markov Covariance Structure. (Sect. 5.5)
Fig. 5.10(left) shows the empirical sample covariance matrix $\frac{1}{N}\sum_n \operatorname{vec}(\mathbf{X}_n)\operatorname{vec}(\mathbf{X}_n)^T$ computed on the full set of sequences. On the right, we show the covariance associated with the matrix normal distribution, i.e., $\boldsymbol{\Delta} \otimes \boldsymbol{\Sigma}$, where $\boldsymbol{\Delta}$ is computed[12] as the covariance of the rows $\boldsymbol{\Delta} = \frac{1}{NF}\sum_n \mathbf{X}_n^T\mathbf{X}_n$, and $\boldsymbol{\Sigma} = \frac{1}{vN3P}\sum_n \mathbf{X}_n\mathbf{X}_n^T$, with $v = \frac{1}{3P}\operatorname{tr}(\boldsymbol{\Delta})$. Note that this separable approximation captures most of the structure and energy in the covariance using far fewer parameter than a full covariance matrix. Fig. 5.5 shows the empirical covariance matrix for a dataset of face motion capture data (an example frame can be seen in Fig. 5.17), for a subset of $P = 10$ points and $F = 10$ frames at 30Hz sampled from 158s of data.

II. Analytical Trajectory Distribution. Fig. 5.11(a) shows that the empirical precision matrix computed over trajectories (the inverse of the sample covariance, $\boldsymbol{\Sigma}^{-1}$) closely re-

[12] ML estimates of the parameters for noiseless data can be obtained using a "flip-flop" algorithm [Dutilleul 1999], but in practice we obtained better results with the described procedure.

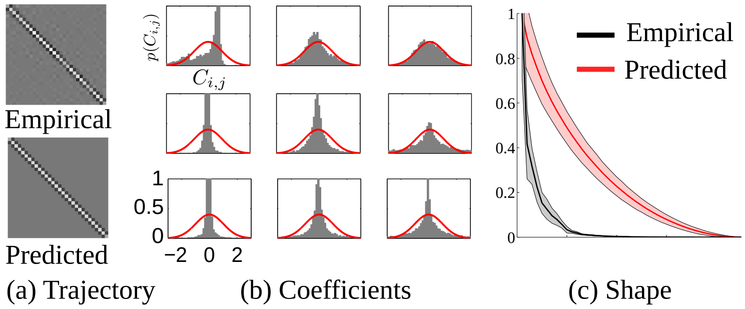


Figure 5.11: Empirical and predicted model parameter distributions. (a) Top, the empirical trajectory precision matrix. Below, the DCT−2 matrix from Sect. 5.4.1. (b) Each plot corresponds to a coefficient $C_{i,j}$ in the matrix $\mathbf{C}$. The red curve shows the predicted standard normal pdf, the histogram shows the empirical distribution. (c) Distribution of singular values for empirical shape covariances (black), compared to the predicted fall-off induced by $p(\mathbf{B})$ (red).

sembles the regularizer predicted by the DCT−2 matrix. Most correlations in the data are captured by the analytical model.

III. Distribution of Coefficients. The matrix normal model assumes a standard normal distribution over the latent coefficients, i.e., $C_{i,j}{\sim}\mathcal{N}(0,1)$. Given a large set of natural motion sequences, we can verify the accuracy of this assumption by fitting the model coefficients $\mathbf{C}_n \in \mathbb{R}^{F\times 3P}$ to each sequence $\mathbf{X}_n$, and plotting the resulting histogram of coefficient values. Fig. 5.11(b) shows that the empirical distribution can be more spiked, closer to Laplacian or Cauchy in practice.

IV. Hierarchical Prior on Shape Covariance. (Sect. 5.6.2) We sample shape covariance matrices from the prior $\mathbf{B}{\sim}\mathcal{MN}(\mathbf{0},\mathbf{I}_{3P},\mathbf{I}_{3P})$ and compute their singular values (SVs). Fig. 5.11(c) compares the energy fall-off in SVs from sampled matrices to that of empirically computed covariance matrices. The plot shows the mean SVs and ±3 standard deviations. The fall-off in energy of the singular values by the prior on $\mathbf{B}$ is not as quick as observed in data, but the choice allows convex optimization.

5.8.2 Missing Data in Motion Capture

To characterize the models' resilience to the patterns of missing data encountered in dynamic reconstruction, we simulate different patterns of occlusion, and we decouple the problem from that of projection loss and reconstructibility [Valmadre and Lucey 2012] by studying inference in 3D. The task is to infer or complete a dynamic 3D point cloud from a reduced set of 3D observations—a practical application would be filling in missing markers in motion capture data. We use the observation model $\mathbf{O}_{\text{miss}}$ as per Sect. 5.2.

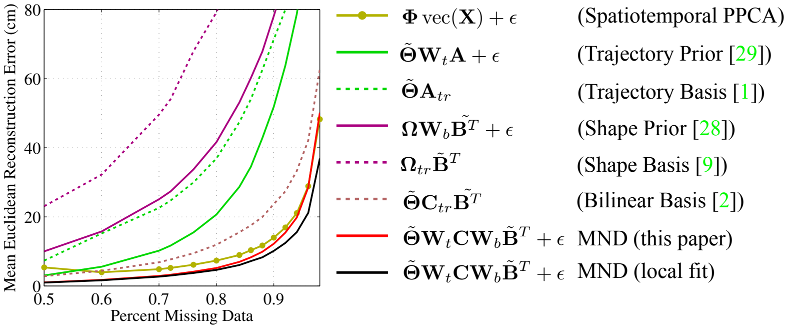


Figure 5.12: Inference of missing data with learned distribution parameters. Subscript *tr* indicates an orthonormal truncation method. See text for a description of the compared methods.

I. Known Distribution Parameters. (Sect. 5.6.1)

When 3D training data is available, we can learn the parameters for MND distribution and perform inference with Eq. (5.53). We compare with the models corresponding to *probabilistic* and *truncated* versions of shape, trajectory, and shape-trajectory distributions (summarized in Table 5.1). Additionally, we evaluate against a probabilistic PCA model trained on the vectorized spatiotemporal sequences, i.e., $\mathbf{y} = \Phi\,\mathrm{vec}(\mathbf{X}) + \epsilon$. We report mean 3D error in Fig. 5.12. As a reference, the error incurred when using the mean shape at every frame as an estimation is ∼175 cm.

For this experiment, we use data from the CMU Motion Capture database. We take 50 random sequences of 20 s in duration, sample them at 30 Hz and Procrustes align and mean center them. There are 31 markers on the body, and we subdivide each sequence into 1s chunks resulting in F=30 and P=31. We train all models on 49 of the sequences, and test on a random 1s segment of the left out sequence. We simulate random occlusion on a percentage of the points and report the average over 50 trials. For the probabilistic models, we set the noise variance to 0. For models relying on truncation of the basis, we sweep over all possible levels of truncation and pick the best number *a posteriori*. Note that the MND model with factored covariance performs equally well or better than PCA on the vectorized sequences, while requiring less training data (50 times less in this experiment). This allows us to train a *local* model only on subsequences neighboring the test data; the model is more specific and results in lower error.

II. Unknown Distribution Parameters. (Sect. 5.6.2)

When no training data is available, we perform inference with Eq. (5.58). In Figure 5.13, we compare our approach with three different priors: (1) a trajectory-only prior, (2) a trace-norm prior, and (3) an additive combination of the trace-norm and trajectory priors. We assume Gaussian observation noise with standard deviation σ=1 mm for all methods. We use dense

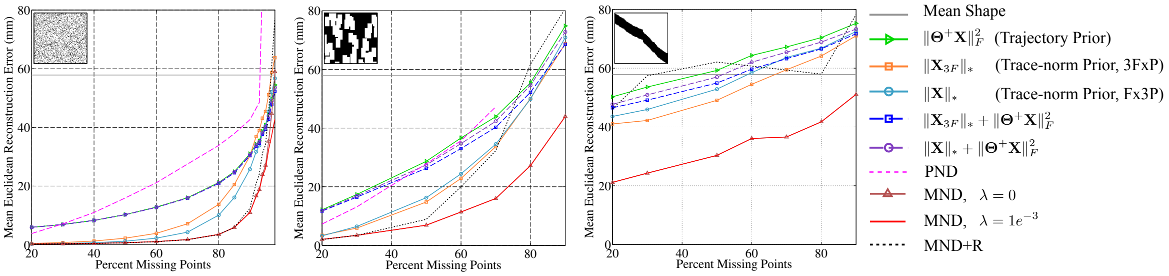


Figure 5.13: Inferring missing data under three different occlusion patterns when the shape distribution is unknown. The graphs show mean Euclidean error in the reconstruction under the occlusion models discussed in Section 5.8.2. The bottom two results correspond to the method of Sect. 5.6.2. We investigate two different arrangements for the data matrix, $3F \times P$ and $F \times 3P$, which capture different correlations of the data. For this experiment, $3F \times P$ usually offered better performance, which we report on our method. The data is from dense human motion capture originally intended to measure non-rigid skin deformation while running in place.

motion capture data from Park and Hodgins [Park and Hodgins 2008]. The sequences are captured at 120 Hz with a dense spatial sampling across the body. We downsample by four spatially and temporally, yielding a point cloud of 118 points at 30 Hz across 162 frames. We measure reconstruction error as mean Euclidean distance over all points, under three different patterns of missing data: **(a) Random:** We occlude points (x,y,z) at random until we achieve a percentage of missing data. This pattern of occlusion is not common in practical situations, but is of interest theoretically: theorems about performance of the trace norm as an approximation to the rank are based on this pattern. **(b) Detection loss:** We model detection loss by occluding spatially proximal points during 1 second durations (30 frames), simulating an occlusion. We superimpose these simulated occlusions to increase the amount of missing data. **(c) Correspondence loss:** We duplicate every point trajectory. Each of the resulting trajectories is observable during a non-overlapping duration, resulting in a pattern similar to that observed when tracking from visual features.

The resulting occlusion patterns are shown as insets in Fig. 5.13. We note that *correspondence loss* results in a much harder problem. Independently of the occlusion pattern, the proposed approach shows improved results. The performance drop when additionally optimizing rotations (Sect. 5.7.2) is explained by the nature of the data, which contains almost no rotation and very little translation. We expect the reduced performance of PND[13] in this experiment is for the same reason. Because there is no temporal smoothness constraint on the rotations, for high percentages of missing data the rotation estimation overfits the observed points.

[13] The original code was modified to compensate for translations of the object. Large amounts of missing data also proved problematic, resulting in numerical issues for some data points in the graph.

Table 5.2: Comparison on zero-noise standard NRSfM sequences using normalized mean 3D error [Dai et al. 2012; Gotardo and Martinez 2011].

Dataset	KSTA	Dai	Traj.	PND	MND	MND+R
Drink	0.0156	0.0266	0.0102	0.0868	**0.0099**	0.0898
Pick-up	0.2322	0.1731	0.1707	0.1188	0.1707	**0.0935**
Yoga	0.1476	0.1150	0.1125	**0.1040**	0.1114	0.1084
Stretch	**0.0674**	0.1034	0.0972	0.0908	0.0940	0.1213
Dance	0.2504	0.1842	0.1385	0.6394	**0.1347**	0.1598
Face2	0.0339	0.0303	0.0408	0.0306	**0.0299**	0.0333
Walking2	**0.1029**	0.1298	0.3111	0.2948	0.1615	0.705
Shark2	**0.0160**	0.2358	0.1380	0.6166	0.1297	0.0684

Table 5.3: Comparison with the PND method of Lee et al. [Lee et al. 2013b] for 0%, 30%, and 60% missing data. We show results using fixed cameras (MND), and optimized using the algorithm of Section 5.7.2 (MND+R).

0% missing data					30% missing data				60% missing data			
Name	PND	PMP	MND	MND+R	Name	PND	MND	MND+R	Name	PND	MND	MND+R
yoga	0.0140	**0.0128**	0.0137	0.0145	yoga	0.0324	0.0430	**0.0179**	yoga	0.0277	0.0519	**0.0246**
pickup	0.0372	**0.0127**	0.0154	0.0142	pickup	0.0366	**0.0141**	0.0145	pickup	0.0267	0.0675	**0.0161**
stretch	0.0156	0.0124	**0.0116**	0.0170	stretch	0.0151	**0.0138**	0.0173	stretch	0.0308	0.0447	**0.0236**
drink	0.0037	**0.0018**	0.0021	0.0022	drink	0.0055	0.0027	**0.0024**	drink	0.0169	0.0519	**0.0051**
dance	0.1834	0.1278	**0.1035**	0.1205	dance	0.1768	**0.1020**	0.1205	dance	0.1512	**0.1072**	0.1212
face	**0.0165**	0.0166	0.0177	0.0195	face	**0.0177**	0.0200	0.0251	face	**0.0208**	0.0279	0.0487
walking	0.0465	**0.0424**	0.1360	0.3756	walking	**0.0459**	0.1256	0.3567	walking	**0.0608**	0.1293	0.3564
jaws	0.0134	**0.0099**	0.0882	0.0687	jaws	**0.0154**	0.0825	0.0696	jaws	**0.0139**	0.0813	0.0713

5.8.3 Non-rigid Structure from Motion

We compare the performance of our time-varying point cloud reconstruction method using Eq. (5.58) on a standard set of structure from motion sequences, where the only data loss is from projection. In Table 5.2, we report normalized mean 3D error as computed in [Gotardo and Martinez 2011] for four methods, (1) KSTA [Gotardo and Martinez 2011], a non-linear kernelized shape-trajectory method, (2) Dai et al. [Dai et al. 2012], (3) a trajectory-only prior, (4) PND [Lee et al. 2013b], and (5) our approach. For our methods (MND and MND+R), we compute the camera matrices as in Dai et al. [Dai et al. 2012] [14], and set σ=1 and λ=0 (the sequences are translationally mean-centered). For Dai et al. and KSTA, the optimal parameter k was chosen for each test.

We also evaluated the robustness with respect to missing data compared to the Procrustean Normal Distribution (PND) of Lee et al. [Lee et al. 2013b]. We report these results in Table 5.3 using the metric used in [Lee et al. 2013b]. Note that this metric is different from

[14] For KSTA [Gotardo and Martinez 2011], the camera matrices are computed as per Akhter et al. [Akhter et al. 2008b]. Our method shows improved performance on 5 of 8 sequences, while the non-linear KSTA method can achieve better performance on some sequences. The implementation of Dai et al. and KSTA was provided by the respective authors.

that used in [Dai et al. 2012], computing normalized mean 3D error on the mean-centered trajectories including camera motion (see [Lee et al. 2013b]). When using the camera matrix optimization procedure of Sect. 5.7.2 (MND+R), we see similar or a slight improvement in performance for most sequences. However, the improvements using MND+R are much smaller than we expected; and the solutions have larger error variance. We attribute this to two factors: (1) the estimated rotations are completely unconstrained and may not be smooth, and (2) the optimization procedure of Sect. 5.7.2 is no longer being convex and the solution can stagnate at a poor local minimum.

5.8.4 Multiview Dynamic Reconstruction

We perform a qualitative evaluation of the method of Sect. 5.6.2 on a dynamic reconstruction sequence from Park et al. [Park et al. 2010]. This sequence is observed very sparsely by multiple cameras taking snapshots of the scene at a rate of around 1 per second. We aim to reconstruct the original motion at 30 Hz. Because the observations are now 2D image measurements under 3D-to-2D perspective projection, we use an observation model $\mathbf{O}_{\text{proj}}$ corresponding to a matrix re-arrangement of the observation model described in [Park et al. 2010].

Fig. 5.14 shows reconstructions on a climbing sequence, where we have simulated occlusion of the left foot. Because ground truth is not available, to obtain a reference reconstruction we first run all methods on the full data and average the resulting structure.

This result is shown in black. Fig. 5.14(left) shows a simulated occlusion of the points on the left foot during the first 6 seconds of the sequence. The trajectory-only prior $\|\mathbf{\Theta}^{+}\mathbf{X}\|_F^2$ gives a smooth solution, but the foot is not at a coherent location with respect to the body. Conversely, all trace-norm based methods are able to infer the position of the left foot (bottom row of images) fairly plausibly in the shape domain. However, when we look at the temporal domain Fig. 5.14(right), we observe that the trace-norm penalization $\|\mathbf{X}\|_*$ results in temporal artifacts—rows in the matrix with no observations are set to zero. This model is not adequate for data interpolation: as observed in the matrix completion literature, the non-uniformity of the missing entries (as happens when interpolating a sparsely observed signal at 30 Hz) negatively affects the performance of trace-norm methods. Our method is able to combine both properties and achieve a smoother interpolation while maintaining a low-rank structure.

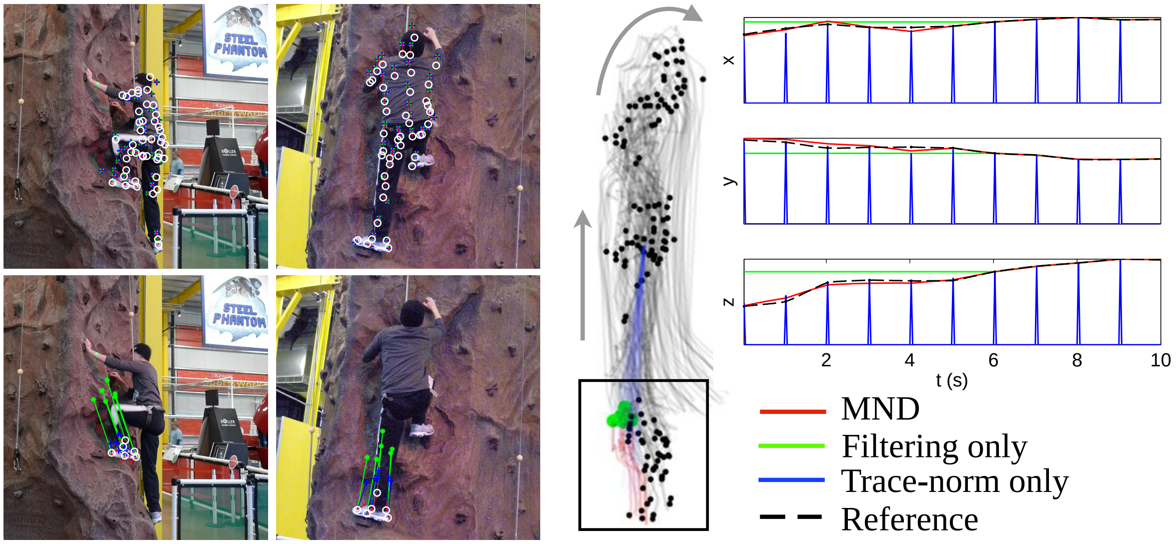


Figure 5.14: Multiview reconstruction on the “Rock Climbing” sequence from [Park et al. 2010]. Annotated labels are shown in white. (Left) Qualitative comparison. The top row shows a result on the full data (104 camera snapshots of 45 points). All methods perform similarly for fully observed frames. The bottom row shows a result on a simulated occlusion (see text). (Center) Reconstructed 3D trajectories of the points, side view of the climbing wall. The arrows denote the direction of motion of the climber. (Right) x,y,z-plot of the mean trajectories of the imputed points.

5.8.5 Monocular reconstruction

In Fig. 5.15 we show a 3D point cloud reconstruction example from a frontal view of a face using 2D landmark detections provided by IntraFace [De la Torre et al. 2015]. The original video is around ∼1500 frames long, which we reconstruct simultaneously. Only a subset of frames is shown here. We directly use the model of Sect. 5.6.2 and build an observation matrix $\mathbf{O}_{\text{ortho}}$ using the head pose estimation matrices provided by IntraFace. Our method recovers a time-varying 3D point cloud of the face, which we can project onto three other views (not used during reconstruction) to evaluate the accuracy. As a quantitative comparison, the ground truth was computed by running the face detector on all views and triangulating the position of each point. The mean 3D error after Procrustes alignment to the ground truth shape was 3.3 mm for MND (λ=1e^{-3}), compared to 3.8 mm for the trace-norm prior (choosing the best weight λ=0.025), and 5.4 mm for MND+R.

5.8.6 3D Time-varying Point Cloud Reconstruction

In Fig. 5.16 we show a reconstruction of the baseball sequence acquired by Joo et al. [2014]. The input is a set of 3D point trajectories obtained from a multi-camera system. Each trajectory is only partially observed (i.e., once a point cannot be tracked forwards or backwards,

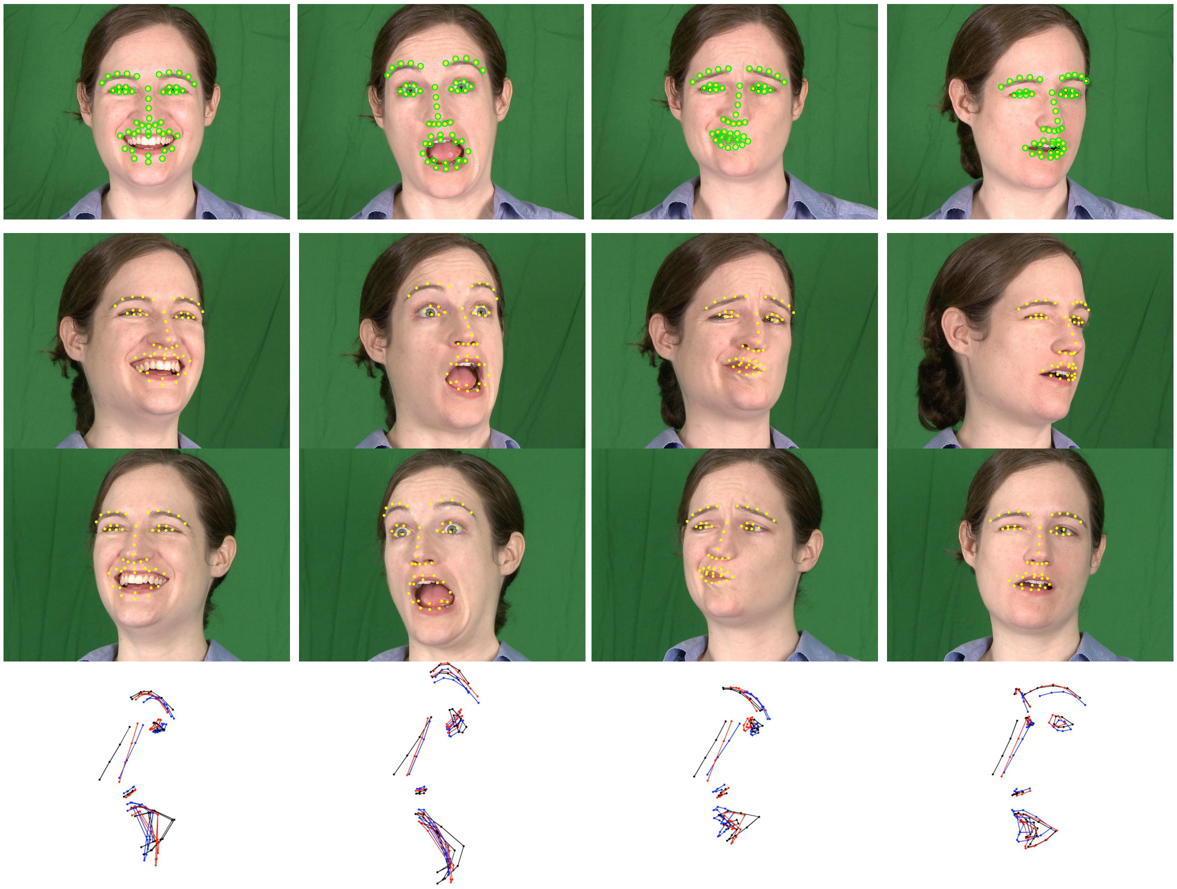


Figure 5.15: Reconstructing a dynamic face from a frontal view. The top row shows frames from a video with superimposed detected 2D landmarks (green circles). We reconstruct the face in full 3D using Eq. (5.58) and show the reprojection onto three other (held out) views for comparison (yellow). Bottom: ground truth (black), MND (red), trace-norm (blue).

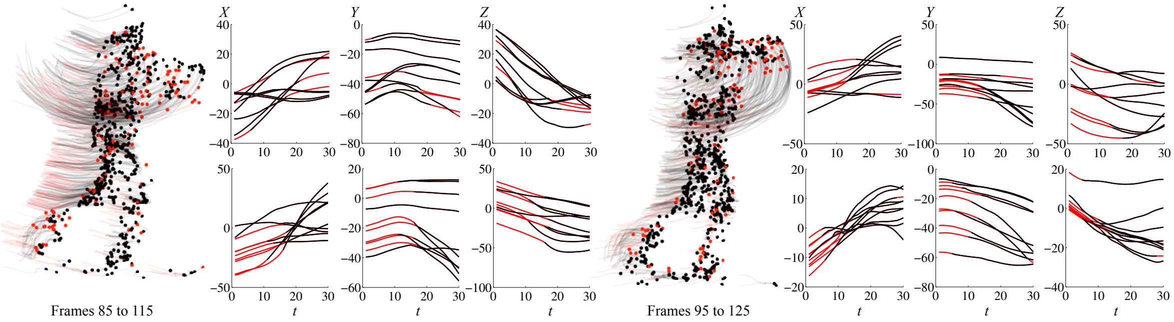


Figure 5.16: Reconstructing a baseball motion sequence. Black lines indicate observed points, red lines are inferred trajectories. Two motion trail diagrams of 30-frame overlapping parts of a baseball swing are shown. The graphs show a close up reconstruction for different subsets of the points.

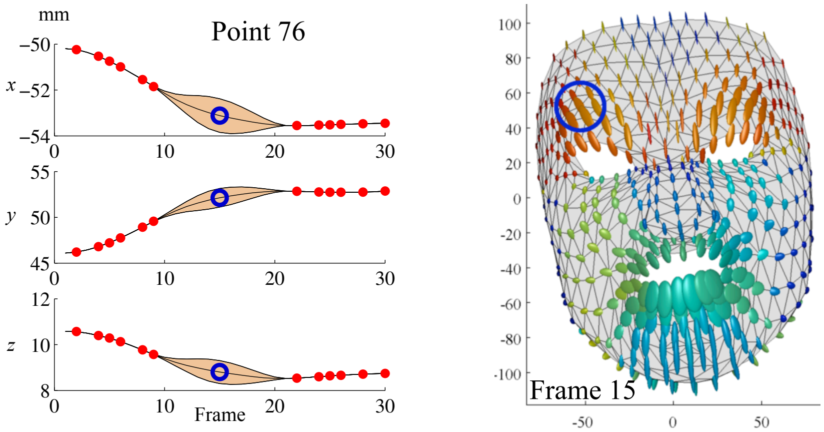


Figure 5.17: The matrix normal model allows us to compute the expected value and spatiotemporal covariance of missing data. For this 30 frame sequence, points have been removed completely from frames 10–20. Observed points are marked by red dots. We infer missing values and visualize the mean and 95% confidence bound.

its coordinates in subsequent frames are missing). These sequences are 30-frames in duration and have around ∼800 points, which where occluded on average ∼15% of the time. The goal is to obtain complete trajectories for the entire duration of the video. Here, we show two qualitative reconstructions for two overlapping 30-frame subsets of these sequences. The graphs show the trajectories for subsets of points. Note how the recovered trajectories are smooth, and motion occurs in groups because of the low-rank effect of the shape prior.

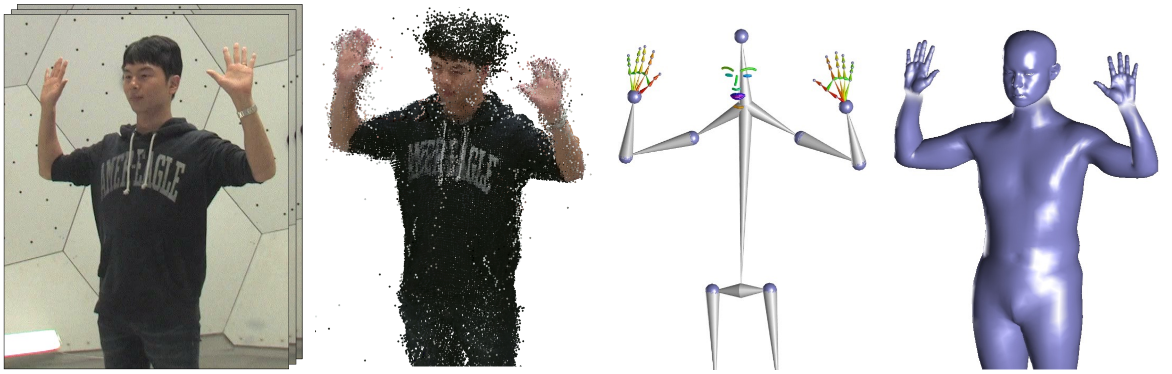
5.9 Conclusions

We have identified the Kronecker-Markov structure of the covariance of time-varying 3D point cloud data and presented a generative, probabilistic model based on the MND that explains this pattern. The model unifies a number of shape and trajectory models, both probabilistic and algebraic, used in prior art. When training data is available, the prior is easy to use in a least-squares framework and greatly outperforms using either shape or trajectory models independently.

When no training data is available, we show how a connection between the MND and the trace norm leads to a convex MAP objective for missing data reconstruction. The advantage of our convex method is that finding a good solution to the shape factorization problem is guaranteed—however, this comes at the expense of employing a prior over shape covariance that is not as concentrated as observed in practice (see Fig. 5.11(c)), and not being able to optimize rotations within the same convex framework. Determining under precisely which conditions a generalized trace norm regularization implies a Kronecker-Markov covariance structure is a possible direction of future work. Conversely, the PND-based optimization procedures, particularly the closely related Markov PND [Lee et al. 2014] prior, show very good results in practice, despite requiring a non-convex optimization and being sensitive to initialization. Ideally, a combination of the properties of both models would result in a stable, highly accurate prior for spatiotemporal data that can be estimated reliably. Along these lines Cabral et al. [Cabral et al. 2013] have shown that under certain conditions a bilinear factorization approach using a non-convex procedure can still converge to the global optimum, and that this optimization can also be simpler and faster. This seems to suggest that, at least for 3D dynamic reconstruction problems, procedures for finding good solutions with guarantees that are less restrictive than convexity might be found.

Chapter 6.

Markerless Body, Face, and Hands Capture



(a) Multiview Images (b) RGB-D Point Clouds (c) Triangulated Keypoint Detections (d) Parameterized Body, Face, and Hands

Figure 6.1: (a) Multiview input images from the Panoptic Studio. (b) 3D point clouds from the RGB-D sensors. (c) Detected and triangulated 3D keypoints. (d) Parameterizations of the body, face, and hands.

6.1 Introduction

The face is a rich source of social signals during interactions, and the hands often play a key role as well—in fact, using the face and hands to signal is an indicator of social skill[1]. However, while several performance capture approaches reconstruct each of these body parts in isolation, there is very little work in reconstructing detailed body, face, and hand motion simultaneously. In large part, this is because of the fundamental difficulty in capturing social interactions unobtrusively, stated at the beginning of this thesis: namely, the need to capture subtle motions across a large space.

Capture systems that work with scenes large enough to house social interactions (i.e., room-sized scenes) can be categorized according to the interpretability of their outputs. On one end of the spectrum, dense multiview stereo or depth-based surface geometry reconstruction algorithms can generate very accurate 3D reconstructions, but are essentially oblivious to the subject matter being captured. Highly detailed and textured 3D reconstructions of social interactions (e.g., [Collet et al. 2015]) and even realtime 3D reconstructions (e.g, [Dou et al. 2016]) have been demonstrated for Augmented Reality (AR) and Virtual Reality (VR) applications. However, while these methods can record or transmit an interaction, much like

[1] For example, the Autism Behavior Checklist [Krug et al. 1980] relies in part on observing hand motions (such as gesturing and pointing) and responses to facial expressions to measure social relating and adapting.

a video they do not capture any of the semantics involved. The geometric representation of the reconstructed surfaces provides no insight into the social signals involved in the interaction. This is the case of Fig. 6.1a and b, where arguably, the multiview video or the 3D point clouds from RGB-D sensors may represent the social signals in the scene accurately, but lack computational interpretability.

On the other end of the spectrum, performance capture methods aim to mimic traditional, marker-based motion capture systems and produce outputs with far fewer degrees of freedom—e.g., joint orientations for a skeleton representation of the body. The advantage here is that the outputs are readily interpretable and can even be used to retarget the captured motion to a different character. In this case, even methods that aim to recover accurate surface detail (e.g., [de Aguiar et al. 2008b; Vlasic et al. 2009; Wu et al. 2013]), do not attempt to reconstruct fingers or facial expressions, mainly due to limits in sensor resolution. To overcome these limits, our method relies on inference over a prior model of body, face, and hand motion, coupled with fine-grained detectors that recognize important landmarks on the body, face, and hands. The triangulated 3D detections are shown in Fig. 6.1c. By combining these sparse, triangulated observations with adequate priors and the depth sensor data, we are able to infer detailed, parameterized motion for body, face, and hands simultaneously, as shown in Fig. 6.1d.

We hypothesize that it is precisely the complex relationship between the motion of each of these different parts of the body that allows the interpretation of social interactions and characterizes realistic-looking animations. Therefore, while static modeling of geometry can afford to capture each of the parts in isolation (e.g., using specialized setups, such as face capture [Beeler et al. 2011]) and then stitch them together into a high-quality mesh, to measure social interactions we need to be able to reconstruct the motion of these body parts simultaneously.

6.2 Method

Fig. 6.2 shows an overview of our method. As input, we take a multiview video in the Panoptic Studio, and obtain 3D detections for the different joints of the body using the markerless motion capture method introduced in Sect. 3.4. The position of the wrist and head joints are then used to detect and triangulate fine-grained joints and keypoints for the hands and face, using the method presented in Sect. 4.4.4. We then fit a parameterization of the body, face, and hands to match these detected positions by minimizing a cost function. For the set of sparse detections, the cost function penalizes deviations from the estimated 3D positions. For the remaining parameters (those that are unobserved or whenever a detection

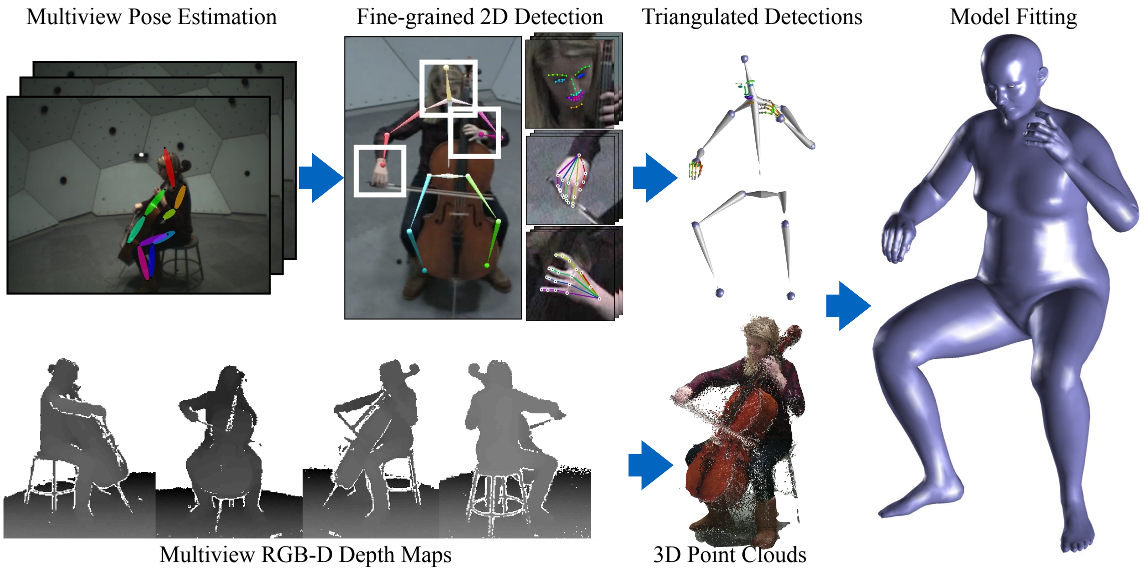

Multiview Pose Estimation
Fine-grained 2D Detection
Triangulated Detections
Model Fitting
Multiview RGB-D Depth Maps
3D Point Clouds

Figure 6.2: Overview of the method. First, multiview images from the Panoptic Studio are used to generate a pose reconstruction. Then, fine-grained 2D detections of keypoints on the face and hands are triangulated. These measurements are combined with a 3D point cloud generated using multiple RGB-D sensors to fit the final parameterization of the body, face, and hands.

is missing or has low confidence), the cost function uses the KMRF spatio-temporal prior over deformation introduced in Sect. 5.4. In addition to the sparse measurements from the detectors, the body parameterization is also fit to depth measurements from the RGB-D sensors.

We refer the reader to Sect. 3.4 and Sect. 4.4.4 for details on the detectors and how we obtain the sparse, 3D triangulated keypoints on the body, face, and hands. In the following, we focus only on fitting the parameterized model of the body to the data.

6.2.1 Modeling the Body, Face, and Hands

To construct a model that generalizes across people but can still capture the required degrees of freedom on the hands and face, we combine three different models: (1) the SMPL model [Loper et al. 2015] for the overall body shape and pose, (2) a face model built from the FaceWarehouse dataset [Cao et al. 2014], and (3) a professionally rigged hand mesh. Each of these models has its own particular parameterization, but they all result in mesh surface representations that we use to fit the observed data.

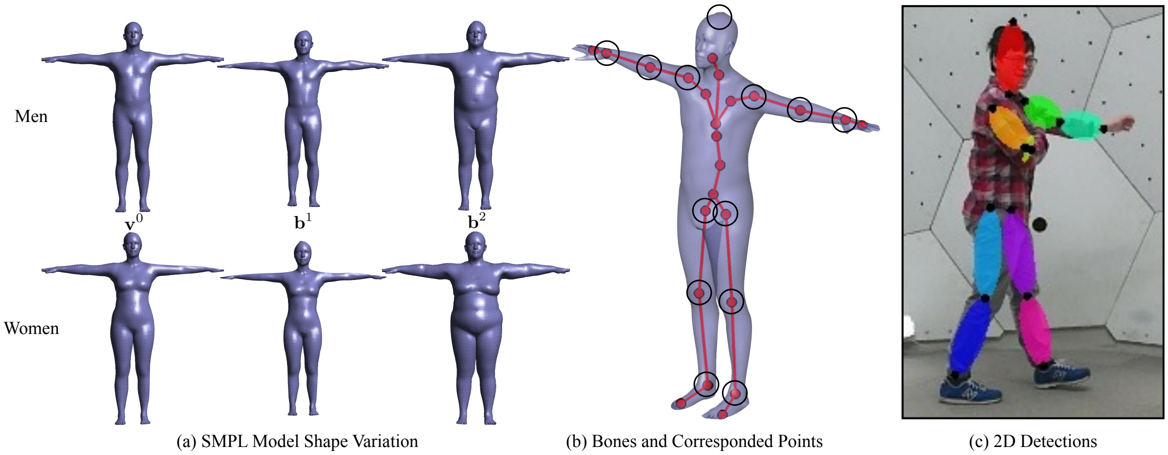


Figure 6.3: (a) SMPL model [Loper et al. 2015], template shapes (male and female) and first two basis vectors of shape variation. (b) SMPL model joint hierarchy. Joints or vertices with a direct correspondence to a detected point are circled. (c) 2D detections corresponding to the outlined joints.

6.2.1.1 Body Model

For the body, we use the SMPL model of Loper et al. [2015], a linear blend shape model of body shape that is deformed via linear blend skinning. The model is parameterized by the gender, a set of identity coefficients (10, in our case), and a set of joint angles represented as axis-angle 3-vectors. The model uses a template mesh of V=6890 vertices $\mathbf{v}_i^0 \in \mathbb{R}^3$, with i iterating over vertices. The vertices of this template mesh are first displaced by a set of blend shapes describing the *identity* or overall body shape, yielding mesh vertices $\mathbf{v}_i'$ in the rest shape (a standard T-pose),

$$\mathbf{v}_i' = \mathbf{v}_i^0 + \sum_{k=1}^{K_b} \mathbf{b}_i^k a_k, \tag{6.1}$$

where K_b is the number of identity body shape coefficients, the vector $\mathbf{a} \in \mathbb{R}^{K_b} = [a_1 \dots a_{K_b}]$ is the vector of identity coefficients, and $\mathbf{b}_i^k \in \mathbb{R}^3$ is the i-th vertex of the k-th blend shape. With respect to the original model of Loper et al., we do not use the pose blendshapes during fitting; this results in faster optimization runtimes and a sparser Jacobian matrix during the minimization. Given the vertices in the rest pose, the posed mesh vertices are obtained by linear blend skinning using transformation matrices $\mathbf{T}_j \in \mathbb{R}^{4\times4}$ for each of the J joints,

$$\mathbf{v}_i = \mathbf{I}_{3\times4} \cdot \sum_{j=1}^{J} w_{i,j} \mathbf{T}_j \begin{pmatrix} \mathbf{v}_i' \\ 1 \end{pmatrix}, \tag{6.2}$$

where $w_{i,j}$ is the weight of transform $\mathbf{T}_j$ for vertex i, with $\sum_{j=1}^{J} w_{i,j}$=1 and $\mathbf{I}_{3\times 4}$ is the 3×4 truncated identity matrix to transform from homogenous coordinates to a 3 dimensional vector. For the SMPL model, $J = 24$, with 23 joints and an additional root transform to orient the body in world-space coordinates. The transformation matrices $\mathbf{T}_j$ encode the transform for each joint j from the rest pose to the posed mesh in world coordinates, and are a function of the joint angles $\boldsymbol{\theta} \in \mathbb{R}^{3\times J}$ (including the world-space orientation of the body) and the world-space translation $\mathbf{t} \in \mathbb{R}^3$. Each column $\boldsymbol{\theta}_j$ is the angle-axis representation of the relative rotation of joint j with respect to its parent joint. A key characteristic of the SMPL model is that these transformations also depend on the shape coefficients $\mathbf{a}$; we refer the reader to [Loper et al. 2015] for details on the construction of these matrices. Let us call $\mathbf{V}_b$=$[\mathbf{v}_1 \ldots \mathbf{v}_V] \in \mathbb{R}^{3\times V}$ the full set of posed vertices in world coordinates. We can then compose the action of Eq. (6.1) and Eq. (6.2) into a function that transforms from the set of body parameters (body shape $\mathbf{a}$, body pose $\boldsymbol{\theta}$, and translation $\mathbf{t}$) to the posed vertices of the body mesh in world coordinates,

$$\mathbf{V}_b = b(\mathbf{a}, \boldsymbol{\theta}, \mathbf{t}). \tag{6.3}$$

Fig. 6.3a shows the vertices of the SMPL model in its rest pose, color coded to illustrate the region of influence of each bone. Each color corresponds to a different bone, and the smooth blending between colors reflects the changing relative weights of each of the bones influencing a single vertex. Fig. 6.3b shows the full joint hierarchy, which at 23 joints is more complex than the subset of joints detected (Fig. 6.3c) and triangulated (Fig. 6.3d) by the method of Sect. 3.4. Note that the detected points do not cover all degrees of freedom captured by the SMPL model, particularly details about body shape as well as the orientation of certain limbs, notably the forearms, head, and feet.

6.2.1.2 Face Model

Because the SMPL model lacks any detail and expressivity in the face, we discard the geometry around the head in the SMPL model and instead use the FaceWarehouse dataset [Cao et al. 2014] to build a facial expression model. We decompose the shapes in the dataset into two linear subspaces, one corresponding to identity and one corresponding to expression variations,

$$\mathbf{v}'_i = \mathbf{v}^0_i + \sum_{k=1}^{K_{fi}} \mathbf{f}^k_i c_k + \sum_{k=1}^{K_{fe}} \mathbf{e}^k_i u_k \tag{6.4}$$

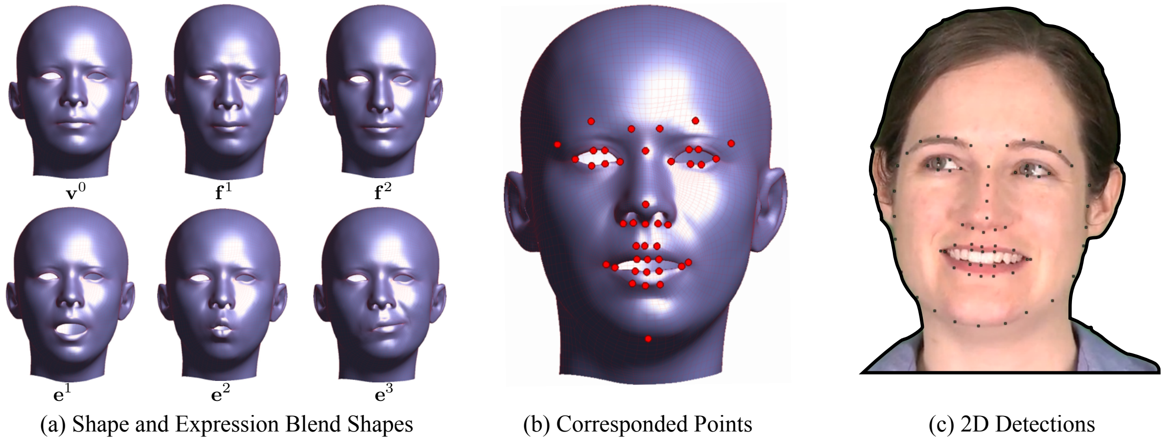


Figure 6.4: (a) FaceWarehouse [Cao et al. 2014] based face model. From left to right, top to bottom, neutral, first and second shape blend shapes, and the first three expression blend shapes. (b) Subset of points in correspondence with detected points. (c) 2D landmark detections. Only a subset of these are used for fitting.

where as before $\mathbf{v}_i^0$ denotes the overall mean shape, which is defined by a mesh of $V{=}11510$ vertices. (We use the same letter $\mathbf{v}$ to denote vertices, but it should be clear from the context which model these vertices belong to.) Here, $\mathbf{f}_i^k \in \mathbb{R}^3$ is the i-th vertex of the k-th identity blend shape, and $\mathbf{e}_i^k \in \mathbb{R}^3$ is the i-th vertex of the k-th expression blend shape. The mean shape and first few blend shapes of each subspace are visualized in Fig. 6.4a. To build these subspaces, we first compute the mean neutral face over all instances of a neutral expression in the dataset. Principal Component Analysis (PCA) is then used to build the identity shape basis using the set of neutral faces. Then, for each subject, we subtract the subject's neutral expression from each example facial expression to obtain the displacement (from a neutral face) that characterizes each expression. The expression subspace is built using PCA on these vertex displacements. Each of these subspaces are parameterized respectively by a vector $\mathbf{c} \in \mathbb{R}^{K_{fi}} = [c_1 \ldots c_{K_{fi}}]$ of face identity coefficients (of which there are K_{fi}) and a vector $\mathbf{e} \in \mathbb{R}^{K_{fe}} = [u_1 \ldots u_{K_{fe}}]$ of facial expression coefficients (of which there are K_{fe}). Finally, a transformation brings the face vertices into world coordinates,

$$\mathbf{v}_i = \mathbf{I}_{3\times 4} \cdot \mathbf{T}_f \begin{pmatrix} \mathbf{v}_i' \\ 1 \end{pmatrix}, \tag{6.5}$$

where the transform $\mathbf{T}_f$ corresponds to the rotation and translation of the face/head joint from the SMPL body model described in the previous section. We can then compose the action of Eq. (6.4) and Eq. (6.5) into a function that transforms from the set of face and body parameters to the posed vertices of the face mesh in world coordinates,

$$\mathbf{V}_f = f(\mathbf{a}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{e}, \mathbf{t}), \tag{6.6}$$

where $\mathbf{c}$ is the vector of coefficients controlling face shape due to identity and $\mathbf{e}$ is the vector of coefficients controlling facial expression.

6.2.1.3 Hand Model

The SMPL model lacks the required degrees of freedom for modeling hand pose. We therefore use a professionally rigged hand mesh with J=16 joints. Similarly to the body model, the mesh is deformed via linear blend skinning. However, because we do not have a dataset of high-quality hand scans at our disposal, we cannot build a shape space of hand size and shape variations as we did for the body and face. Instead, we increase the degrees of freedom of the joints to allow for different finger sizes by adding an X,Y, and Z scaling factor to each bone. The transform for each joint j is then parameterized by the Euler angle rotation[2] with respect to its parent, $\boldsymbol{\phi}_j \in \mathbb{R}^3$, and an additional anisotropic scaling factor along each axis, $\boldsymbol{s}_j \in \mathbb{R}^3$ (similar to Jacobson and Sorkine [2011][3]). Specifically, the transform for points in joint j's local reference frame becomes

$$\mathbf{T}'_j = \begin{bmatrix} \mathrm{eul}(\boldsymbol{\phi}_j) \cdot \mathrm{diag}(\boldsymbol{s}_j) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{6.7}$$

where $\mathrm{eul}(\boldsymbol{\phi}_j)$ converts from an Euler-angles representation $\boldsymbol{\phi}_j \in \mathbb{R}^3$ to a 3×3 rotation matrix, and $\mathrm{diag}(\boldsymbol{s}_j)$ is the 3×3 diagonal matrix with the entries of $\mathbf{s}_j$ on the diagonal, dictating the scaling factor along each of the bone's axes in its local reference frame. This way if, say, $\mathbf{s}_j = [2, 1, 1]^T$, the bone j is scaled by 2 along its principal direction (i.e., the finger becomes twice as long but does not change its girth). The final transforms for each joint are

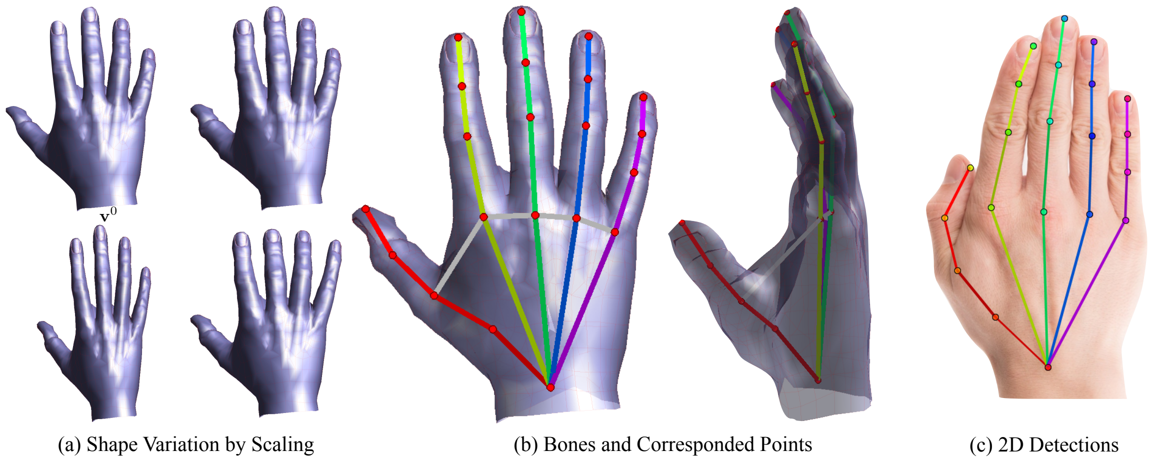
$$\mathbf{T}_j = \left(\prod_{i \in \mathcal{A}(j)} \mathbf{T}_{p(i) \leftarrow i} \cdot \mathbf{T}'_i \right) \mathbf{B}_j^{-1}, \tag{6.8}$$

where $\mathcal{A}(j)$ is the list of ancestors for joint j (ordered so that the product goes from j to the root transform), $\mathbf{T}_{p(i) \leftarrow i}$ is the transform from bone i's local frame to that of its parent $p(i)$ in the rest pose, and $\mathbf{B}_j$ transforms[4] the bone from its local reference frame to its bind pose

[2] We use Euler angles instead of axis-angle for the joints of the hand because most joints have either only one degree of freedom (e.g., the distal and proximal interphalangeal joints), or a very limited range of motion along the other axes (e.g., the metacarpophalangeal joints). These constraints are easier to express in the Euler angle parameterization, and the anthropometric limits on their values avoid many of the usual problems with this parameterization (e.g., the Gimbal lock).

[3] We did not find the endpoint weighting scheme of Jacobson and Sorkine [2011] necessary, most likely because the range of scalings is limited.

[4] For the hand mesh model that we use, $\mathbf{B}_j = \prod_{i \in \mathcal{A}(j)} \mathbf{T}_{p(i) \leftarrow i}$ so that the rest pose coincides with the bind pose.



(a) Shape Variation by Scaling

(b) Bones and Corresponded Points

(c) 2D Detections

Figure 6.5: (a) Template mesh (top left) and variations in hand shape obtained by anisotropic scaling of the joints. (b) Bones and corresponded points, including joints and the tips of the fingers. (c) 2D landmark detections.

in world coordinates (i.e., the position and orientation of the bone relative to the template mesh). The template mesh vertices in the bind or rest pose are denoted by $\mathbf{v}_i^0$ and the mesh has V=2068 vertices. This template mesh and variations obtained by the anisotropic scaling of Eq. (6.8) are shown in Fig. 6.5a. The bones and joint ancestors are shown in Fig. 6.5b. As before, the final vertices of the hand in world coordinates are given by linear blend skinning with weights $w_{i,j}$,

$$\mathbf{v}_i = \mathbf{I}_{3\times 4} \cdot \sum_{j=1}^{J} w_{i,j} \mathbf{T}_j \begin{pmatrix} \mathbf{v}_i^0 \\ 1 \end{pmatrix} . \tag{6.9}$$

We can write the full set of vertices for the hand model in world coordinates as a function of the parameters,

$$\mathbf{V}_h = h(\mathbf{a}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{s}, \mathbf{t}). \tag{6.10}$$

This is done for each hand in turn, so that there are a set of hand pose parameters for the left hand $\boldsymbol{\phi}^l$ and right hand $\boldsymbol{\phi}^r$. However, the scale parameters are shared (we assume that the hands are symmetric enough).

6.2.1.4 Full Body Model

To recap, the full body model is parameterized by the following variables, which define the final position of each of the model's vertices in world coordinates:

Symbol	Meaning	Size
$\mathbf{a}$	Body shape coefficients	10
$\boldsymbol{\theta}$	Body pose	3×24
$\mathbf{c}$	Face shape coefficients	150
$\mathbf{e}$	Facial expression coefficients	200
$\boldsymbol{\phi}^r$	Right hand pose	3×16
$\boldsymbol{\phi}^l$	Left hand pose	3×16
$\boldsymbol{s}$	Hand scale parameters	3×16
$\mathbf{t}$	Global translation	3

While the number of facial coefficients is larger than the degrees of freedom of other parts of the body, it should be noted that these parameters are heavily constrained by a prior, and, in fact, almost all of the variation in the training set is captured within approximately the first 60 coefficients.

Conceptually, the parameter set can be split into two categories. One set of parameters are static shape or *identity* parameters, including body shape, face shape, and hand scaling factors, which do not vary with time (e.g., bone lengths):

$$\mathcal{S} = \{\mathbf{a}, \mathbf{c}, \mathbf{s}\}. \tag{6.11}$$

The remaining parameters are per-frame *expression* parameters, including body pose, hand pose, facial expression, and global orientation and translation,

$$\mathcal{E}_t = \{\boldsymbol{\theta}_t, \mathbf{e}_t, \boldsymbol{\phi}^r_t, \boldsymbol{\phi}^l_t, \mathbf{t}_t\}. \tag{6.12}$$

This set of parameters is inherently dynamic and we index it with the sub-index t to indicate that they are time-varying. To simplify the exposition, we will abstract away the details of each of the models used and instead denote the concatenated set of vertices,

$$\mathbf{V}_t = [\mathbf{V}_b, \mathbf{V}_f, \mathbf{V}_{hr}, \mathbf{V}_{hl}] = B(\mathcal{S}, \mathcal{E}_t), \tag{6.13}$$

where $\mathbf{V}_t \in \mathbb{R}^{3 \times V}$ with V the total number of vertices, and the function $B(\cdot)$ maps from the combined model parameters, including static and per-frame parameters, to the full configuration of vertices for the body, face, and hands.

6.2.2 Objective Function

The parameters of the full body model $B(\cdot)$ are fit to match the available 3D measurements in the Panoptic Studio by minimization of a suitable cost function. These measurements can either be triangulated 3D detections that are in correspondence with the model (i.e., they have a semantic label attached, e.g., "left elbow") or simply represent a reconstructed 3D point (from either stereo or a depth sensor) with no additional information. These latter points could potentially correspond to an arbitrary location on the surface of the body, but it is also possible that they do not belong to the body at all (e.g., other people, objects). Therefore, the two types of measurements must be handled differently, and we refer to each of these as $E_{\text{detection}}$ and E_{icp} respectively. Additionally, the cost function will include priors over both the shape parameters, and the expression parameters capturing the pose at any given time instant. A high-level description of the optimization problem is then to find the parameters that minimize the cost function:

$$\{\mathcal{S}, \mathcal{E}_1, \ldots, \mathcal{E}_T\} = \arg \min_{\{\mathcal{S}, \mathcal{E}_1, \ldots, \mathcal{E}_T\}} E_{\text{detection}} + E_{\text{icp}} + E_{\text{shape prior}} + E_{\text{expression prior}}, \tag{6.14}$$

where T is the total number of frames. We describe each of these error terms in more detail in the following sections.

6.2.2.1 Corresponded Points

For corresponded points, we enforce the constraint that their measured 3D position should match that of the corresponding point on the mesh models. Our detectors fire both on internal points (the joints of the body and hands) as well as on points that are on the surface of the body (the facial landmarks, the top of the head, and the tips of the fingers). To treat all corresponded points in the same way, we define each of these positions on the mesh as a linear combination of a small set of vertices on the respective models. As in the previous chapter, we formulate this as the action of a linear observation matrix $\mathbf{O}$,

$$\text{vec}(\mathbf{Y}_t) = \mathbf{O}\,\text{vec}(\mathbf{V}_t) + \boldsymbol{r} \tag{6.15}$$

with $\mathbf{Y}_t \in \mathbb{R}^{3\times n_{\text{det}}}$ the set of triangulated 3D detections for frame t, n_{det} the number of detections, and $\mathbf{O} \in \mathbb{R}^{n_{\text{obs}}\times 3V}$ the observation matrix, where $n_{\text{obs}} = 3n_{\text{det}}$. Here, $\boldsymbol{r}$ is the vector of residuals, i.e., the mismatch between the model and the observed point. For example, let's consider a single triangulated 3D detection $\mathbf{y}_i \in \mathbb{R}^3$ which is in direct correspondence with a vertex j. In this case, we can write

$$\mathbf{y}_i = \underbrace{[\mathbf{0}_{3\times 3(j-1)}, \mathbf{I}_3, \mathbf{0}_{3\times 3(V-j)}]}_{\mathbf{O}_i} \operatorname{vec}(\mathbf{V}) + \boldsymbol{r}_i, \tag{6.16}$$

where the 3×3 identity matrix $\mathbf{I}_3$ selects the vertex j from the vectorized matrix of vertices $\mathbf{V}$ (i.e., there are $3(j-1)$ zeroes in front of it), and $\boldsymbol{r}_i$ is the residual error. As an example, the eye corner detector points can be placed in exact correspondence with a single vertex at the corner of the eye. However, internal body joints are (by construction of the SMPL model, see [Loper et al. 2015]) a function of several vertices. These joints can be expressed as the linear function $\sum_i w_i \mathbf{v}_i$, with $\mathbf{v}_i$ the vertices that surround a joint (e.g., a circle of vertices around the knee, for example) and w_i a set of weights that sum to one. For these points, we gather these linear terms as a sub-matrix $\mathbf{O}_i$, where the entries are positive, sum to one, and are very sparse. The joints of the hand can similarly be expressed as a linear function of the posed vertices. In this way, we can write the full vector of residuals

$$\mathbf{r} = \operatorname{vec}(\mathbf{Y}_t) - \mathbf{O} \operatorname{vec}(\mathbf{V}_t), \tag{6.17}$$

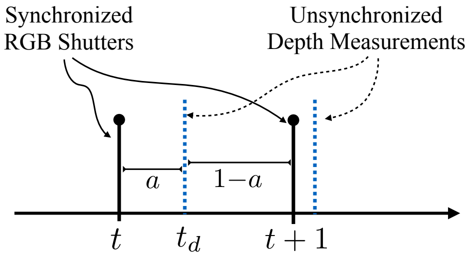
where $\mathbf{O}$ is obtained by stacking the $3 \times 3V$ sub-matrices $\mathbf{O}_i$ of each correspondence. Note that typically, n_{det} is at most the number of detectors used (82 in our case) while the number of vertices is in the tens of thousands and the degrees of freedom of the model number several hundreds. Figures 6.3, 6.4, and 6.5 show the correspondences available for each of the models. Not all the correspondences are always available, nor should all residuals be weighted equally, so we write the final cost due to detections as

$$E_{\text{detection}} = \frac{1}{2}\mathbf{r}^T \mathbf{W} \mathbf{r}, \tag{6.18}$$

with $\mathbf{W}$ a weighting matrix. Nominally, $\mathbf{W}$ corresponds to the inverse covariance of the residuals; however, in practice, we approximate this as a diagonal with weight c_i for the residual entries corresponding to detection i, where $c_i \in [0, 1]$ is the detection confidence of that detection (and 0 if the detection is missing).

6.2.2.2 Uncorresponded Points

In addition to the sparse measurements at corresponded detections, we make use of the dense depth data captured by the RGB-D sensors in the Panoptic Studio. These depth measurements, however, are not a priori in correspondence with the model meshes. We therefore establish their correspondence to the mesh using Iterative Closest Point (ICP) during each solver iteration.



However, because the exposure of the RGB-D sensors is not synchronized to that of the remaining RGB cameras on the Studio, we must first compensate for the temporal offset between each RGB-D sensor and the global time reference of the system. The global time reference is a constant frame-rate clock, coinciding with the shutters of RGB cameras in the Studio. For simplicity, we will denote instants of this clock as integer values of t, and assume that at each of these instants we define the per-frame parameters $\mathcal{E}_t$ (with t in $\{1, 2, \dots F\}$, with F the number of frames in the sequence). As we saw in Sect. 3.3.2.1, even though the shutters are not synchronized to the same clock, we do have accurate timing information that allows us to put all measurements on a common timeline. For every depth sensor measurement (i.e., a depth image from a single RGB-D sensor), we can find the closest neighboring frames t and $t+1$ in the global timeline and compute an appropriate interpolation factor[5]. That is, if the depth image arrives at time t_d with $t < t_d < t+1$, then the interpolated mesh that approximates the depth acquired by the sensor is

$$\mathbf{V}_{t_d} = \mathbf{V}_t + a(\mathbf{V}_{t+1} - \mathbf{V}_t), \tag{6.19}$$

with $a = t_d - t$. We can now find the closest 3D point from the depth sensor to each of the mesh vertices,

$$i^* = \arg\min_i ||\mathbf{x}_i - \mathbf{v}_j||^2, \tag{6.20}$$

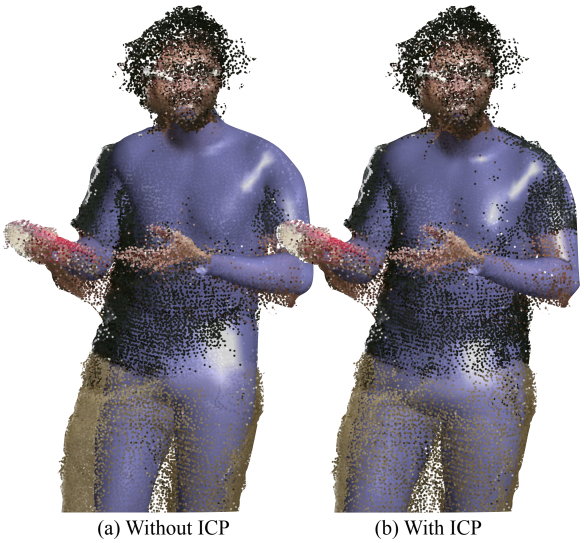
where $\mathbf{x}_{i^*}$ is the closest 3D point to vertex j, where $\mathbf{v}_j$ is a vertex[6] in $\mathbf{V}_{t_d}$. To ensure that this is a reasonably good correspondence, we require that the minimum matching distance be lower than a certain threshold, i.e., $||\mathbf{x}_{i^*} - \mathbf{v}_j|| < d$, where we use d=10cm. We additionally ensure that the normal direction at the mesh vertex is compatible with the normal direction at the depth measurement (i.e., the two surfaces locally face in similar directions—if they do not, this is likely an incorrect correspondence). As do Newcombe et al. [2011], we compute a weight using the dot product of the normals, where the weight is zero if the angle between the normals is greater than 90°, it is 1 if they are parallel, and a smooth interpolation elsewhere,

$$w_j = \max\Big(0,\ \mathbf{n}(\mathbf{x}_{i^*})^T \mathbf{n}(\mathbf{v}_j)\Big) \cdot I(||\mathbf{x}_{i^*} - \mathbf{v}_j|| < d), \tag{6.21}$$

where $\mathbf{n}(\cdot) \in \mathbb{R}^3$ approximates the (outward-facing) surface normal at its argument, and $I(\cdot)$

[5] When fitting a single frame, we use the closest sensor depth measurements if they fall within a temporal window of 15ms.

[6] We do not consider hand vertices, as depth sensor resolution is too low to improve the estimate.


(a) Without ICP

(b) With ICP

Figure 6.6: Body mesh alignment before (left) and after ICP (right).

is the indicator function. For mesh vertices, $\mathbf{n}(\mathbf{v}_j)$ is computed as the average normal of the facets which share j as a vertex. For depth measurements, $\mathbf{n}(\mathbf{x}_i)$ is approximated using depth measurements in the 3×3 pixel neighborhood of i, also as in [Newcombe et al. 2011]. Finally, for each vertex j we compute the point-to-plane residual, i.e., the distance along the normal direction,

$$r_j = \mathbf{n}(\mathbf{x}_{i^*})^T(\mathbf{x}_{i^*} - \mathbf{v}_j), \tag{6.22}$$

where we use the normal estimate of the depth measurement to simplify the Jacobian. As before, we express the full vector of residuals ($\mathbf{r}_u = [r_1 \dots r_V] \in \mathbb{R}^V$) as an observation matrix times the vectorized vertices, where the equation above corresponds to one row in the observation matrix, i.e.,

$$r_j = \mathbf{n}(\mathbf{x}_{i^*})^T\mathbf{x}_{i^*} - \underbrace{[\mathbf{0}_{1\times 3(j-1)},\ \mathbf{n}(\mathbf{x}_{i^*})^T,\ \mathbf{0}_{1\times 3(V-j)}]}_{\mathbf{O}^u_j}\operatorname{vec}(\mathbf{V}_{t_d}), \tag{6.23}$$

We also gather each of the V weights w_j along the diagonal of a matrix $\mathbf{W}_u$, such that the contribution to the final cost function is

$$E^k_{\text{icp}} = \frac{1}{2}\mathbf{r}_u^T\mathbf{W}_u\mathbf{r}_u, \tag{6.24}$$

where k indexes a particular RGB-D sensor. If there are K such sensors, then the total cost

$$E_{\text{icp}} = \sum_{k=1}^{K} E_{\text{icp}}^{k}, \tag{6.25}$$

where each of the k terms has its own acquisition time, interpolation, and set of closest point matches. The improvement in alignment when using ICP iterations is shown in Fig. 6.6.

6.2.2.3 Shape Priors

The set of identity parameters $\mathcal{S}$ is in some cases an over-parameterization, and we may lack sufficient measurements to determine them uniquely. This is the case with the hand shape parameterization, where the anisotropic scaling factors pertaining to finger girth are difficult to observe in any of the detected points. A more subtle problem that requires the use of priors here is model error. This is most noticeable for the SMPL body parameterization: the SMPL shape model is designed to fit semi-naked people, whereas we are interested in measuring people who are wearing arbitrary clothes and accessories. Similarly, hair geometry is not captured in the FaceWarehouse data. We can therefore expect incorrect correspondences and large residuals due to these limitations in the shape models, particularly when fitting depth measurements via ICP. This requires putting a prior over the model parameters or constraining them in order to avoid over-fitting to these large residuals.

We therefore set the following priors over the parameters (ideally with as low an influence as the data allows). The priors over body shape and face shape are Gaussians with an identity covariance (because the corresponding basis vectors obtained with PCA are scaled such that the variance of the coefficients is one), so that

$$E_{\text{body}} = \frac{1}{2}\mathbf{a}^T\mathbf{a}, \tag{6.26}$$

$$E_{\text{face}} = \frac{1}{2}\mathbf{c}^T\mathbf{c}. \tag{6.27}$$

The prior we set on the scaling factors of the hand has a more complex expression because we have no data to train on,

$$E_{\text{hand}} = \frac{1}{2}\left[(\mathbf{s}-1)^T(\mathbf{s}-1) + ||\mathbf{s}^1 - \mathbf{s}^2||_F^2 + ||\mathbf{s}^1 - \mathbf{s}^3||_F^2 + ||\mathbf{s}^2 - \mathbf{s}^3||_F^2\right]. \tag{6.28}$$

However, this expression should be fairly intuitive, and simply says that the scaling factors have a Gaussian distribution around the value 1 and scaling along each axis should be similar (i.e., we expect that longer fingers will also have larger girth unless there is evidence to the contrary), and we additionally constrain the scaling values to be within the range $[0.2, 1.75]$.

6.2.2.4 Expression Priors

The priors over the dynamical expression parameters (body pose, facial expression, hand pose, and global movement) are spatio-temporal in nature. As we saw in the previous chapter, we can use a KMRF prior with an analytical (DCT-2) trajectory basis, which is equivalent to enforcing Gaussian terms among frames of the form (see Eq. (5.33)),

$$E_{\text{bpose0}} = \frac{1}{2}(\boldsymbol{\theta}_t - \overline{\boldsymbol{\theta}})^T \boldsymbol{\Delta}_b^{-1}(\boldsymbol{\theta}_t - \overline{\boldsymbol{\theta}}) \tag{6.29}$$

$$E_{\text{bpose1}} = \frac{1}{2}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1})^T \boldsymbol{\Delta}_b^{-1}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}) \tag{6.30}$$

$$E_{\text{bpose2}} = \frac{1}{2}(2\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_{t-1})^T \boldsymbol{\Delta}_b^{-1}(2\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_{t-1}), \tag{6.31}$$

where $\boldsymbol{\Delta}_b$ is the covariance matrix of the body pose parameters and $\overline{\boldsymbol{\theta}}$ is the mean vector of pose parameters. Each of the terms above corresponds to different levels of autocorrelation or smoothing. If we are optimizing a single frame or there exists no frame $t{+}1$, we can only use the first expression. If we have frames at t and $t{+}1$ frames but not $t{-}1$, we use the second expression, and if we have both $t{-}1$ and $t{+}1$ around t we use the third expression. In practice, we always include E_{pose0} to prevent large deviations in the parameterization, but with a very small weight if any of the other terms are applicable. The facial expression coefficients are also penalized by a similar KMRF prior

$$E_{\text{fexp0}} = \frac{1}{2}\boldsymbol{e}_t^T \boldsymbol{\Delta}_f^{-1}\boldsymbol{e}_t \tag{6.32}$$

$$E_{\text{fexp1}} = \frac{1}{2}(\boldsymbol{e}_t - \boldsymbol{e}_{t+1})^T \boldsymbol{\Delta}_f^{-1}(\boldsymbol{e}_t - \boldsymbol{e}_{t+1}) \tag{6.33}$$

$$E_{\text{fexp2}} = \frac{1}{2}(2\boldsymbol{e}_t - \boldsymbol{e}_{t+1} - \boldsymbol{e}_{t-1})^T \boldsymbol{\Delta}_f^{-1}(2\boldsymbol{e}_t - \boldsymbol{e}_{t+1} - \boldsymbol{e}_{t-1}), \tag{6.34}$$

where $\boldsymbol{\Delta}_f$ is the covariance matrix of the facial expression parameters, and we assume the mean expression is the neutral expression, for which $\mathbf{e} = \mathbf{0}$.

We obtain the covariance matrix over pose parameters $\boldsymbol{\Delta}_b$ and the mean pose parameter vector $\bar{\boldsymbol{\theta}}$ by fitting the SMPL model to the CMU mocap dataset [Carnegie Mellon University]. The facial expression covariance $\boldsymbol{\Delta}_e$ is obtained by fitting the face model to all the example facial expressions in the FaceWarehouse data.

However, for the hand pose parameters of the left and right hands, $\boldsymbol{\phi}^r$ and $\boldsymbol{\phi}^l$, as well as for the global translation parameter $\mathbf{t}$ we do not have training data, and use identity covariance matrices. There is also no “single-frame” version of a prior over global translation, as position can be arbitrary, but we derive priors of the same form as above, E_{trans1}, E_{trans2}, and similarly $E_{\text{hposer*}}$ and $E_{\text{hposel*}}$ for the hand pose.

6.2.2.5 Full Objective Function

Optimizing the complete objective function for a set of T frames involves optimizing the parameter sets $\mathcal{S}$ and $\{\mathcal{E}_t : t \in \{1, \dots, T\}\}$, subject to the set of triangulated 3D detections, $\{\mathbf{Y}_t : t \in \{1, \dots, T\}\}$, the set of depth sensor maps, $\{\mathbf{X}^k_{t_d} : 1{\leq}t_d{\leq}T,\ k \in \{1, \dots, K\}\}$, with K the number of depth sensors ($K{=}10$ in our setup), as well as the various priors. Thus, each of the error terms described above can occur multiple times depending on the frames and available observations. The full objective can then be summarized as the sum over the following terms, where each term appears in a certain number of instances,

Term	Weight	Parameters	Instances	Equations
$E_{\text{detection}}$	1	$\mathcal{S}$, $\mathcal{E}_t$	$\{\mathbf{Y}_t \in \mathbb{R}^{3\times P} : \forall t\}$	(6.18)
E_{icp}	λ_i	$\mathcal{S}$, $\mathcal{E}_t$, $\mathcal{E}_{t+1}$	Per depth-map $t{\leq}t_d{\leq}t{+}1$, $\forall t$	(6.25)
E_{body}	$T\lambda_b$	$\boldsymbol{a}$	1	(6.26)
E_{face}	$T\lambda_f$	$\boldsymbol{c}$	1	(6.27)
E_{hand}	$T\lambda_h$	$\boldsymbol{s}$	1	(6.28)
E_{bpose}	λ_p	$\boldsymbol{\theta}_{t-1}$, $\boldsymbol{\theta}_t$, $\boldsymbol{\theta}_{t+1}$	$\forall t$	(6.29–6.31)
E_{fexp}	λ_{fe}	$\boldsymbol{e}_{t-1}$, $\boldsymbol{e}_t$, $\boldsymbol{e}_{t+1}$	$\forall t$	(6.32–6.34)
E_{hposer}	λ_{hp}	$\boldsymbol{\phi}^r_{t-1}$, $\boldsymbol{\phi}^r_t$, $\boldsymbol{\phi}^r_{t+1}$	$\forall t$	As above
E_{hposel}	λ_{hp}	$\boldsymbol{\phi}^l_{t-1}$, $\boldsymbol{\phi}^l_t$, $\boldsymbol{\phi}^l_{t+1}$	$\forall t$	As above
E_{trans}	λ_t	$\boldsymbol{t}_{t-1}$, $\boldsymbol{t}_t$, $\boldsymbol{t}_{t+1}$	$\forall t$	As above

The different weights of each term were tuned by hand, and we minimize this error function as a non-linear least-squares problem using the Ceres solver [Agarwal et al. 2009]. In particular, we use the trust-region Levenberg-Marquardt (LM) solver, calculating each of the Jacobians and ICP correspondences in every iteration, and using a sparse Cholesky solver for each LM step.

6.2.3 Fitting the Model

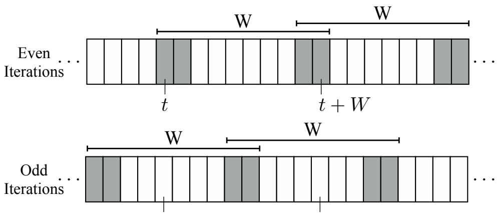
The complete model is highly non-linear and has a large number of unknowns, some of which are subject to bound constraints. Therefore, an iterative optimization scheme can fall into poor local minima if not initialized close enough to the correct solution. We therefore follow a particular optimization schedule to mitigate this problem. Additionally, because sequences can be many thousands of frames long, it is infeasible to optimize all parameters for all frames simultaneously. Therefore, we first sample 20 frames at uniform intervals from the entire sequence and fit the full set of parameters, including the identity parameters $\mathcal{S}$ and the expression parameters for each frame, $\mathcal{E}_t$ with t in the set of sampled frames. After this initial fitting, we fix the identity parameters $\mathcal{S}$ and proceed to optimize the expression parameters of all frames in the sequence. To initialize the expression parameters of each frame, we first optimize each frame independently, and only then, to obtain a temporally smooth reconstruction, we optimize the entire sequence while enforcing the spatiotemporal priors.

6.2.3.1 Per-Frame Optimization

Once the set of identity parameters $\mathcal{S}$ is fixed, we start the optimization by fitting the expression parameters of each frame independently. To initialize the expression parameters in each frame, we first determine the global orientation of the body and global position $\mathbf{t}$ using Procrustes alignment of the body model's shoulder and hip joints to the 3D detected positions of these joints. The remaining parameters are initialized to their default value (body limbs in the mean pose, hand joints in the rest pose, and a neutral facial expression). We then run the non-linear optimization described in the previous section in two stages: During the first stage, the priors are given a large weight to avoid local minima and increase the capture region. Once this minimization converges, we run a second stage of non-linear optimization where the weight of the priors is set very small—ideally, the effect of the prior should be negligible on joints for which we do have a measurement, and only affect those degrees of freedom that are unobserved. In practice, due to noise in the measurements and model error, we cannot set the weight of the priors to zero.

6.2.3.2 Block-wise Optimization

The per-frame optimization results in jittering motion between frames. While ideally we would optimize the parameters of all frames simultaneously to obtain a smooth spatiotemporal reconstruction, this is impractical—it would require keeping the depth maps for all frames in memory at once, and each non-linear least-squares solver iteration would involve hundreds of thousands of coupled unknowns (for example, for 2 minutes of video at 30 fps, there are approximately 1.3 million unknowns). Instead, we optimize blocks (windows) of only W=10 frames simultaneously. This amounts to solving the original optimization problem (i.e., optimizing all frames at once) using block-coordinate descent. To ensure that the cost function being optimized is the same as the original (non-windowed) cost function, we hold the parameters of two frames at the border of each windowed block constant (two frames is sufficient because none of the residuals involve parameters at a distance of more than one frame). This procedure is conceptually similar to a red-black optimization ordering for Gauss-Seidel solvers on a grid (see e.g., [Saad 1996]). We alternate between even and odd solver iterations, where the frames that are held constant (shown as gray frames in the inset) are switched so that, given sufficient iterations, the problem converges to the same solution as the original cost function. Note that within each even or odd iteration, each block of W frames is independent of all other blocks, meaning that these blocks of frames can be optimized in parallel.



6.3 Results

We show qualitative results of our method in Fig. 6.7 and Fig. 6.8. We find that the alignment of the hand mesh is surprisingly accurate, as can be seen at the bottom of Fig. 6.8. The largest discrepancies occur at the body near clothed areas and the hair. Note, however, that the rendered parameterization—even lacking any texture information—convey the gist of the interaction in much the same way that the original image does.

6.3.1 Limitations

We highlight three important limitations of our method. The first and most important is resolution, particularly for facial details. Even though 31 of the cameras in the Panoptic Studio capture HD frames (1920×1080 resolution), these pixels are distributed over a large capture region (around $25m^2$) and the size of an imaged face is typically 50 to 100px vertically. This makes resolving subtle facial expressions difficult. For example, accurately measuring lip motion during speech is not possible with the current resolution, nor is measuring eye gaze. The second limitation is due to the (lack of) generalizability of the models used. This is noticeable for the body, where clothing can severely interfere with estimation (particularly for very baggy clothes and accessories, such as long scarves). We believe this can be mitigated in part by allowing subject specific deformations of the mesh to better fit an entire sequence; however, this will likely still be problematic for e.g., very small children, which are not observed in the SMPL model training data. Lastly, very large occlusions that last for a long time (e.g., hand in pocket, arms folded, or arms behind the back) are very challenging for our inference procedure. In our results, we also observe failures of the keypoint detectors.

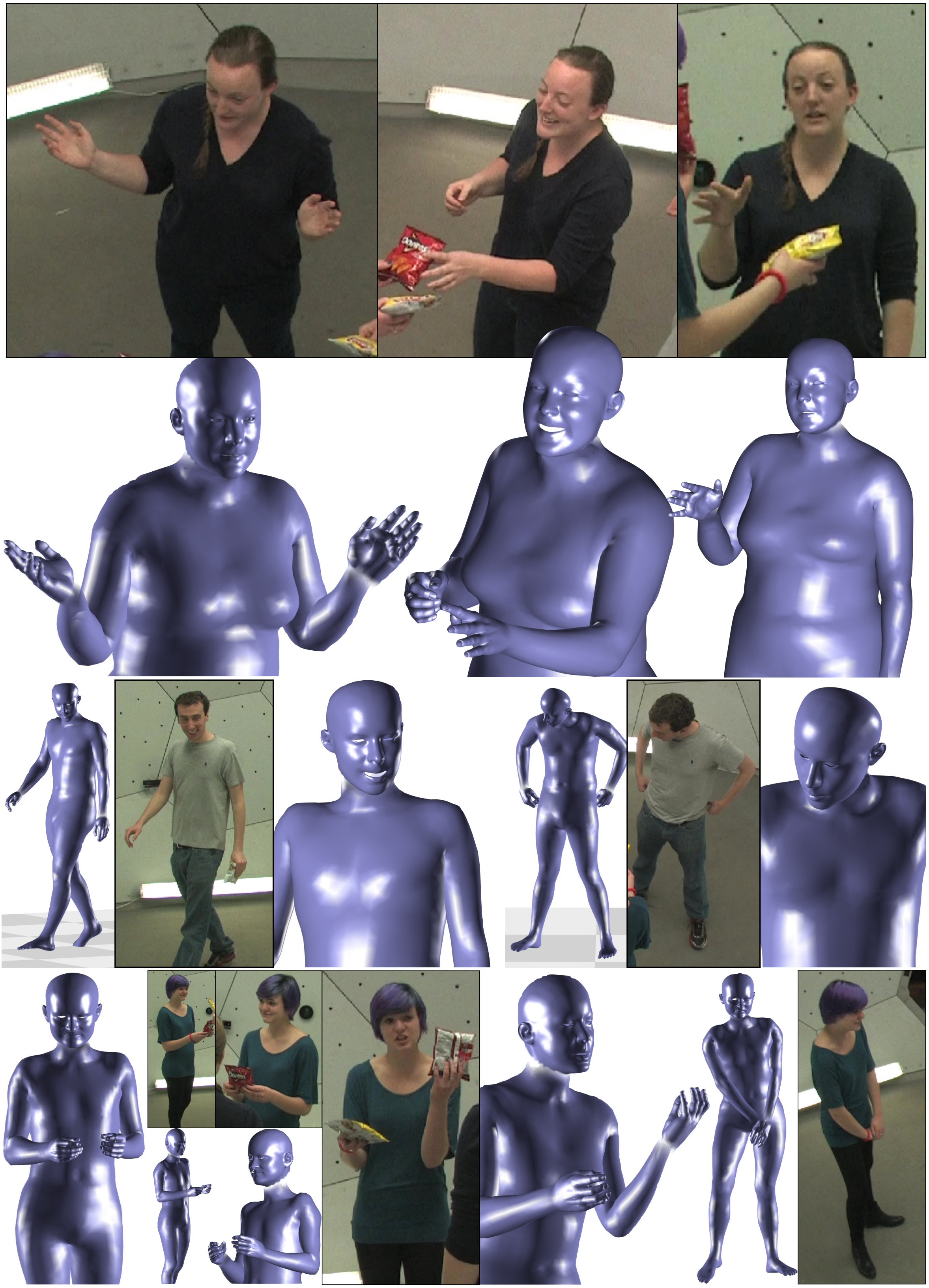


Figure 6.7: Qualitative body, face, and hands reconstruction results on one of the “haggling” sequences from the Panoptic Studio.

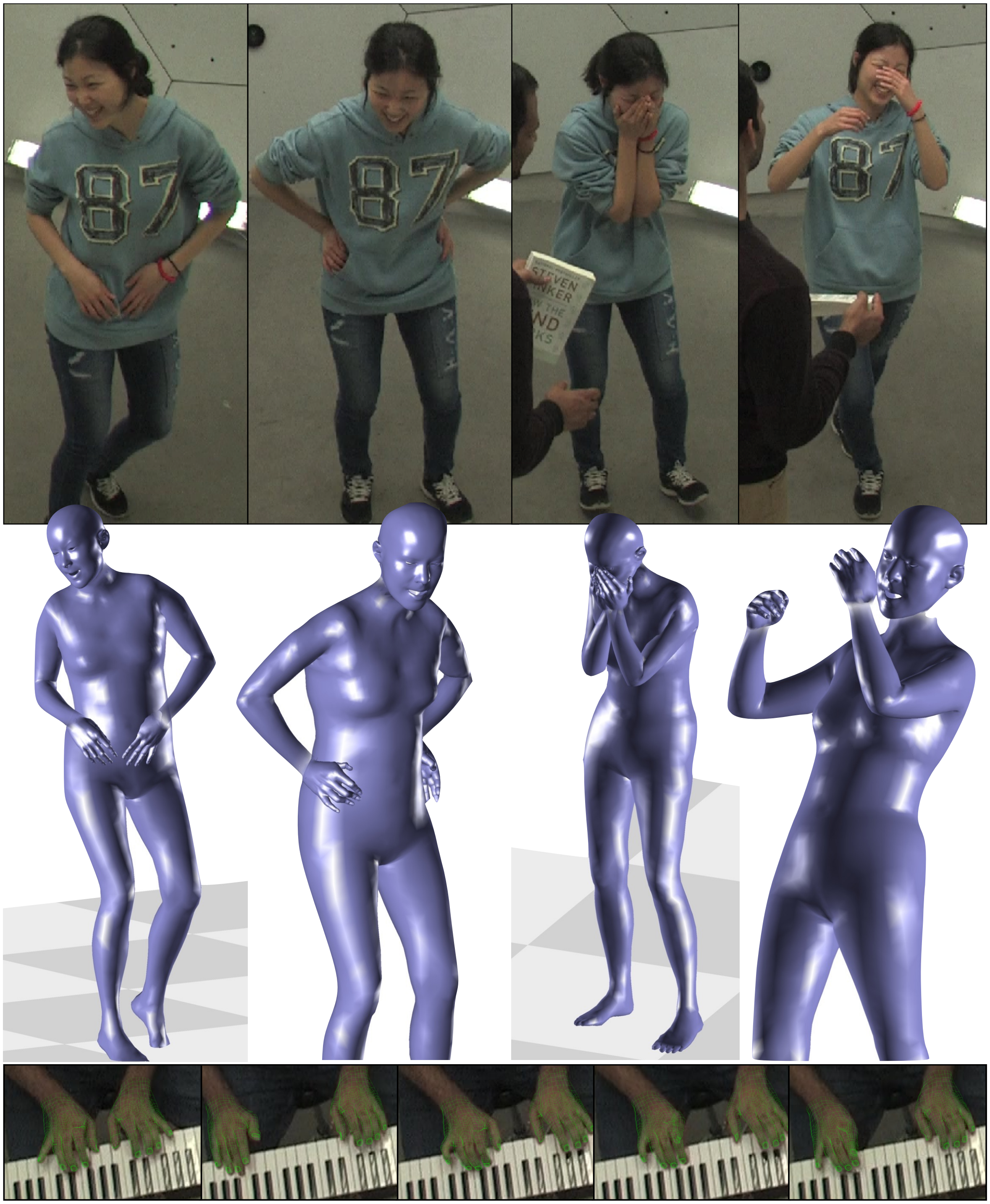


Figure 6.8: Top: Qualitative body, face, and hands reconstruction results on one of the “haggling” sequences from the Panoptic Studio. Bottom: A few frames of a “piano” sequence. The hand mesh model is shown as an overlay on the RGB image.

However, we believe that with some engineering and retraining, most of these issues can be resolved. An exception to this is the face. Besides the problem of resolution, another important failure case we have observed are inaccuracies due to insufficient frontal viewpoints of the face. This happens most often due to inter-occlusions, but also at certain points within the capture room. Having a larger number of HD camera views would alleviate this problem.

6.4 Conclusions

We have presented a markerless motion capture system that captures body, face, and hand details of multiple people simultaneously. This allows us to measure, for the first time, the interactions between the motions of each of these parts during natural conversations. Additionally, because our system works in a large capture area, we are able to reconstruct several people simultaneously, allowing us to study relationships between the motions of different people. The process is fully automatic and requires no manual intervention, allowing us to capture as much data as we can process[7].

Our approach is essentially model fitting. However, an interesting direction of future work is to tackle this as a retargeting problem instead. Indeed, we find that many of the larger problems encountered with model fitting (e.g., the model is not expressive enough for a particular action, or has more degrees of freedom than we can fit) are very similar to those that we would encounter when retargeting human motion from one character to another. We believe that with regards to future work, the biggest gains can therefore be achieved by improving the *measurement* rather than improving the model priors.

[7] We have not studied processing time in detail, but the current implementation requires about 1 second per iteration per frame, which we run for 30 to 60 iterations.

Chapter 7.

Conclusion

In this thesis, we developed methods for social signal reconstruction—in particular, we *measure human motion during social interactions*. Compared to other work in this area, the methods presented in this thesis are able to measure the configuration of the multiple interacting people at different spatial and temporal scales without using markers: from overall body pose to subtle hand gestures and facial expressions. This allows us to measure, for the first time, the interactions between the motions of each of these parts during natural conversations. Additionally, because our system works in a large capture area, we are able to reconstruct several people simultaneously, allowing us to study relationships between the motions of different people.

The key to achieving this without placing markers, instrumentation, or other restrictions on participants is the *Panoptic Studio*, a massively multi-view capture system which allows us to obtain 3D reconstructions of room-sized scenes. In particular, we empirically find that having a large number of views is necessary for social interaction capture of groups of people. To measure the position of joints and other landmarks on the human body, we combine the output of 2D keypoint detectors across multiple views and triangulate them in 3D. This approach of integrating simple detection cues over a large number of views shows advantages in reconstructing variable number of subjects of diverse appearance, body sizes, and body topology over long time periods—crucially, without the need to impose priors on the motion or skeletons or perform any initial calibration of the subjects.

The overarching goal that originally motivated this thesis was to create computational models of human motion during social interactions, particularly to develop predictive models for our responses to social signals. This is what prompted us to collect a very large dataset of natural interactions from which we could learn statistical models of behavior, which in turn required designing both hardware and algorithms that would allow us to measure human motion at reasonable levels of detail.

With the capture hardware and measuring software described in this thesis now in place, we have begun large scale captures which we hope will provide rich datasets for the analysis of social interactions. We are now faced with a very practical concern: how much data do we need, and how much data can we capture and process?

In practice, capturing about 30 minutes of three-person interactions requires around 40 terabytes of storage and around 3 to 4 weeks of processing time to obtain full-body reconstructions with our currently available computational resources. This processing time is likely to go down (even without algorithmic improvements on our part), but at the moment, computational resources are a clear limiting factor. This complexity limits the amount of data we can record, particularly when compared to datasets collected at "internet scale", like ImageNet [Deng et al. 2009] or MS-COCO [Lin et al. 2014].

Because of this, we have scoped out a well-defined subset of social interactions to analyze first: namely, one-minute triadic interactions with two competing sellers and one buyer (the "haggling" game). Even in this restricted setting, it is apparent that people with different dispositions (e.g., competitiveness, motivation, gender, age, etc.) will behave differently and elicit different behaviors, and that we will need to capture many different combinations of these traits. How many example captures are necessary to be able to say we have sufficiently sampled this restricted space of triadic interactions? While this is difficult to define abstractly, it is at least now possible (with the ability to measure the social signals of the participants) to quantify the amount of variance captured by specific models we develop.

In this sense, the approach we took is classically bottom-up: we aim to "measure everything" in as much detail as our camera's resolution allows, and we ground our reconstructions and models in geometrical measurements. However, leveraging the vast amount of single-view video already available on the web seems a promising route to achieve breadth and variety in measuring social motions. We believe the value of the Panoptic Studio in this domain is as a *training data generation machine*.

In particular, in this thesis we developed a semi-supervised training procedure, *multi-view bootstrapping*, which uses 3D triangulation to generate training data for keypoint detectors. We use this technique to train fine-grained 2D keypoint detectors for landmarks on the hands and face, allowing us to measure these two important sources of social signals. More generally however, the Panoptic Studio allows us to combine multiple detectors (for example, detectors trained specifically on face datasets and detectors trained on hand datasets) into integrated full-body representations. These representations are immediately corresponded with a multitude of 2D views of the scene, producing training data useful for single-view reconstruction. The major limiting factor for our current system is resolution: at HD resolutions and a distance of about five feet, the resolution of the face is relatively poor[1].

In terms of modeling human motion data, the *Kronecker Markov Random Field* (KMRF) model for keypoint representations of the face and body developed in this thesis is a very low level model. We show that most of the covariance in natural body motions corresponds to a specific set of spatiotemporal dependencies which result in a Kronecker or matrix normal

[1] Anecdotally, while people seem to be able to interpret facial expressions at these resolutions, it is difficult to precisely localize particular facial keypoints.

distribution over spatiotemporal data. This statistical model can be used to infer complete sequences from partial observations and unifies linear shape and trajectory models of prior art into a probabilistic shape-trajectory distribution that has the individual models as its marginals. This model only captures short-term spatiotemporal correlations between point locations rather than model human interactions at a higher level (e.g., modeling intentions or goals).

However, we still found that this KMRF model is useful to combine the various measurements obtained from the Panoptic Studio into full-body motion reconstructions, allowing us to fit mesh models of the body, face, and hands. This representation encodes many of the social signals that characterize an interaction and can be used for analysis, modeling, and animation. This approach is essentially model fitting by minimization of an energy function. However, especially compared to the discriminative approach we used to detect individual joints, this model fitting step is prone to catastrophic failure. This happens for example when the initial parameters are far from the true solution, or if the model is not expressive enough to reconstruct a particular motion. We believe there is great potential in end-to-end learning of functions to recover the final parameterization (e.g., a mesh representation factored into identity and pose parameters) directly, rather than depend on a two-step process. An intermediate approach would be to substitute the model-fitting minimization with a data-driven discriminative fitting procedure, which we believe would also alleviate the catastrophic failure modes of the current approach. Again, we believe that the Panoptic Studio here will serve as a training data generation machine, where the massively multiview input allows us to generate ground truth data while providing corresponded single-view data.

In this spirit, we have made an effort to make all the collected data and processed labels easily available on our website, http://domedb.perception.cs.cmu.edu, and hope that the collected data will prove a useful resource to other researchers. This website is a work in progress which we hope will continue to grow and improve beyond the completion of this thesis, and we encourage the reader to visit it for more up-to-date developments.

Bibliography

Adib, F., C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand
2015. Capturing the human figure through a wall. *ACM Transactions on Graphics*, 34(6):1–13. (Link). 2.1.2

Agarwal, S., K. Mierle, and Others
2009. Ceres solver. http://ceres-solver.org. 3.3.2.2, 4.2.1, 5.7.2, 6.2.2.5

Akhter, I., Y. Sheikh, S. Khan, and T. Kanade
2008a. Nonrigid structure from motion in trajectory space. In *Advances in Neural Information Processing Systems*. 2.2.2.2

Akhter, I., Y. Sheikh, S. Khans, and T. Kanade
2008b. Nonrigid structure from motion in trajectory space. *NIPS*. 2.2.2.2, 5.5.2, 5.1, 14

Akhter, I., T. Simon, I. Matthews, S. Khan, and Y. Sheikh
2012. Bilinear spatiotemporal basis models. *TOG*. 2.3, 5.2.1, 5.5.1.1, 5.5.1.1, 5.5.2

Alameda-Pineda, X., J. Staiano, R. Subramanian, L. Batrinca, E. Ricci, B. Lepri, O. Lanz, and N. Sebe
2015. Salsa: A novel dataset for multimodal group behavior analysis. *TPAMI*. 2.1.2

Allen, G. and R. Tibshirani
2010. Transposable regularized covariance models with an application to missing data imputation. *Annals of Applied Statistics*. 5.5.1.1

Alletto, S., G. Serra, S. Calderara, F. Solera, and R. Cucchiara
2014. From ego to nos-vision: Detecting social relationships in first-person views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Pp. 594–599. 3.3

Amin, S., M. Andriluka, M. Rohrbach, and B. Schiele
2013. Multi-view pictorial structures for 3d human pose estimation. *BMVC*. 2.1.3, 4.1

Andriluka, M., L. Pishchulin, P. Gehler, and B. Schiele
2014. 2d human pose estimation: New benchmark and state of the art analysis. *CVPR*. 2.1.3, 2.1.4, 3.4.7, 4.2, 4.1, 4.4.1

Angst, R., C. Zach, and M. Pollefeys
2011. The generalized trace-norm and its application to structure-from-motion problems. *ICCV*. 5.6.2

Arev, I., H. S. Park, Y. Sheikh, J. Hodgins, and A. Shamir
2014. Automatic Editing of Footage from Multiple Social Cameras. *ACM Transactions on Graphics*, 33(4):1–11. (Link). 4.6

Arikan, O.
2006. Compression of motion capture databases. In *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, volume 25, P. 890, New York, New York, USA. ACM Press. (Link). 2.2.2.4

Athitsos, V. and S. Sclaroff
2003. Estimating 3D hand pose from a cluttered image. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003 Proceedings*, 2:II–432–9. (Link). 2.1.4

Baak, A., M. M, G. Bharaj, H.-p. Seidel, and C. Theobalt
2011. A Data-Driven Approach for Real-Time Full Body Pose Reconstruction from a Depth Camera. *ICCV*. 2.1.4, 4.1

Ballan, L., A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys
2012. Motion capture of hands in action using discriminative salient points. In *ECCV*. 2.1.4

Beeler, T., F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross
2011. High-quality Passive Facial Performance Capture Using Anchor Frames. In *ACM SIGGRAPH 2011 Papers*, Pp. 75:1—-75:10. 6.1

Belagiannis, V., S. Amin, and M. Andriluka
2014. 3D pictorial structures for multiple human pose estimation. *CVPR*. 2.1.3, 4.1

Belhumeur, P., D. Jacobs, D. Kriegman, and N. Kumar
2011. Localizing parts of faces using a consensus of exemplars. In *IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*. 4.5

Belongie, S.
2014. Rodrigues' Rotation Formula. *MathWorld - A Wolfram Web Resource*. (Link). 3.3.2.2

Bilakhia, S., S. Petridis, and M. Pantic
2013. Audiovisual Detection of Behavioural Mimicry. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, Pp. 123–128. IEEE. (Link). 2.1.2

Birdwhistell, R. L.
1952. *Introduction to kinesics: an annotation system for analysis of body motion and gesture*. University of Louisville. (Link). 1, 2.1.1

Boker, S. M., J. F. Cohn, B.-J. Theobald, I. Matthews, M. Mangini, J. R. Spies, Z. Ambadar, and T. R. Brick
2011. Something in the way we move: Motion dynamics, not perceived sex, influence head movements in conversation. *Journal of Experimental Psychology*, 37(3):874–891. (Link). 2.1.2

Bousmalis, K., M. Mehu, and M. Pantic
2013. Towards the automatic detection of spontaneous agreement and disagreement based on nonverbal behaviour: A survey of related cues, databases, and tools. *Image and Vision Computing*, 31(2):203–221. (Link). 2.1.2

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein
2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Statistical Machine Learning*. 5.7, 5.7.1, 5.7.1, 5.7.1, 5.7.2, 5.7.3

Brand, M.
2005. A direct method for 3d factorization of nonrigid motion observed in 2d. *ICCV*. 2.2.2.1

Bregler, C.
1997. Learning and recognizing human dynamics in video sequences. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Pp. 568–574. IEEE Comput. Soc. (Link). 1

Bregler, C., A. Hertzmann, and H. Biermann
2000. Recovering non-rigid 3d shape from image streams. *CVPR*. 2.2.2.1, 5.3.1, 5.5.2, 5.1

Bregler, C., J. Malik, and K. Pullen
2004. Twist based acquisition and tracking of animal and human kinematics. *IJCV*. 2.1.3

Bronstein, A., M. Bronstein, and R. Kimmel
2008. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company. 2.2

Brox, T., B. Rosenhahn, J. Gall, and D. Cremers
2010. Combined region and motion-based 3D tracking of rigid and articulated objects. *TPAMI*. 2.1.3

Burenius, M., J. Sullivan, and S. Carlsson

2013. 3D pictorial structures for multiple view articulated pose estimation. *CVPR*. 2.1.3, 4.1

Cabral, R., F. D. L. Torre, J. P. Costeira, and A. Bernardino

2013. Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. *ICCV*. 5.9

Calvo, R. A. and S. D'Mello

2010. Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *IEEE Transactions on Affective Computing*, 1(1):18–37. (Link). 2.1.2

Cao, C., Y. Weng, S. Zhou, Y. Tong, and K. Zhou

2014. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425. 2.1.4, 6.2.1, 6.2.1.2, 6.4

Cao, Z., T. Simon, S. Wei, and Y. Sheikh

2016. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv:1611.08050*. (Link). 2.1.3, 3.3

Carnegie Mellon University

. CMU Graphics Lab Motion Capture Database. (Link). 2.1.2, 6.2.2.4

Chakraborty, I., H. Cheng, and O. Javed

2013. 3D Visual proxemics: Recognizing human interactions in 3D from a single image. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Pp. 3406–3413. 2.1.2

Cheung, K. M., S. Baker, and T. Kanade

2005. Shape-from-silhouette across time part i: Theory and algorithms. *IJCV*. 2.1.3

Cohn, J.

2010. Advances in Behavioral Science Using Automated Facial Image Analysis and Synthesis. *IEEE Signal Processing Magazine*, 27(6):128–133. (Link). 1

Cohn, J. F. and P. Ekman

2005. Measuring facial action. In *The new handbook of methods in nonverbal behavior research*, J. A. Harrigan, R. Rosenhal, and K. R. Scherer, eds., Pp. 9–64. Oxford University Press. 2.1.1

Cohn, J. F., T. Simon, I. Matthews, Y. Yang, M. H. Nguyen, M. T. Padilla, F. Zhou, and

F. De la Torre
2009. Detecting Depression from Facial Actions and Vocal Prosody. In *Affective Computing and Intelligent Interaction (ACII)*. 2.1.2

Collet, A., M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan
2015. High-quality streamable free-viewpoint video. *ACM Trans. Graph.*, 34(4):69:1–69:13. (Link). 2.1.3, 6.1

Cootes, T., C. Taylor, D. Cooper, and J. Graham
1995a. Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59. (Link). 2.2.2.1, 5.3.1, 5.3.1

Cootes, T. F., C. J. Taylor, D. H. Cooper, and J. Graham
1995b. Active shape models, their training and application. *CVIU*. 5.5.2

Corazza, S., L. Mündermann, E. Gambaretto, G. Ferrigno, and T. P. Andriacchi
2010. Markerless Motion Capture through Visual Hull, Articulated ICP and Subject Specific Model Generation. *IJCV*. 2.1.3

Cristani, M., L. Bazzani, G. Paggetti, A. Fossati, D. Tosato, A. Del Bue, G. Menegaz, and V. Murino
2011. Social interaction discovery by statistical analysis of f-formations. *BMVC*. 2.1.2

Dai, Y., H. Li, and M. He
2012. A simple prior-free method for non-rigid structure-from-motion factorization. *CVPR*, Pp. 2018–2025. (Link). 2.2.2.1, 2.3, 5.1, 5.6.2, 11, 5.2, 5.8.3

Darwin, C.
1872. The Expression of the Emotions in Man and Animals. *The American Journal of the Medical Sciences*, 232(4):477. (Link). 1, 2.1.1

de Aguiar, E., C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun
2008a. Performance capture from sparse multi-view video. *SIGGRAPH*. 2.1.3

de Aguiar, E., C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun
2008b. Performance capture from sparse multi-view video. *ACM Transactions on Graphics*, 27:98:1–98:10. 6.1

de La Gorce, M., D. J. Fleet, and N. Paragios
2011. Model-based 3D hand pose estimation from video. *TPAMI*. 2.1.4

De la Torre, F., W.-S. Chu, X. Xiong, F. Vicente, X. Ding, and J. Cohn
2015. Intraface. In *FG*, Pp. 1–8. 5.8.5

De la Torre, F., J. K. Hodgins, J. Montano, and S. Valcarcel
2009. Detailed Human Data Acquisition of Kitchen Activities: the CMU-Multimodal Activity Database (CMU-MMAC). In *Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research, in conjuction with CHI 2009*. 2.1.2

De la Torre, F., T. Simon, Z. Ambadar, and J. F. Cohn
2011. FAST-FACS: A Computer-Assisted System to Increase Speed and Reliability of Manual FACS Coding. In *Affective Computing and Intelligent Interaction (ACII)*. 2.1.1

Del Bue, A., X. Llad, and L. Agapito
2005. Non-rigid face modelling using shape priors. In *Analysis & Modelling of Faces & Gestures*. 5.2.1

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei
2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*. 7

Dou, M., S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi
2016. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph.*, 35(4):114:1–114:13. (Link). 2.1.3, 6.1

Dutilleul, P.
1999. The MLE algorithm for the matrix normal distribution. *Statistical Computation and Sim.* 5.4, 5.5.1.1, 12

Ekman, I., G. Chanel, S. Jarvela, J. M. Kivikangas, M. Salminen, and N. Ravaja
2011. Social Interaction in Games: Measuring Physiological Linkage and Social Presence. *Simulation & Gaming*, 43(3):321–338. (Link). 2.1.1

Ekman, P.
2003. Darwin, Deception, and Facial Expression. In *Annals of the New York Academy of Sciences*, volume 1000, Pp. 205–221. 2.1.2

Ekman, P. and W. V. Friesen
1976. Measuring facial movement. *Environmental Psychology and Nonverbal Behavior*, 1:56–75. 2.1.1

Ekman, P. and W. V. Friesen
1978. *The Facial Action Coding System*. 1

Elhayek, A., E. Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt
2015. Efficient convnet-based marker-less motion capture in general scenes with a low number of cameras. *CVPR*. 2.1.3

Elhayek, A. et al.
2016. Marconi-convnet-based marker-less motion capture in outdoor and indoor scenes. *TPAMI*. 2.1.3, 3.4, 4.1

Fathi, A., J. K. Hodgins, and J. M. Rehg
2012. Social interactions: A first-person perspective. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Pp. 1226–1233. 3.3

Fayad, J., A. Del Bue, L. Agapito, and P. Aguiar
2009. Non-rigid structure from motion using quadratic deformation models. *BMVC*. 2.2.2.1

Feix, T., R. Pawlik, H.-B. Schmiedmayer, J. Romero, and D. Kragi
2009. A comprehensive grasp taxonomy. *Robotics, Science and Systems Conference: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation*, Pp. 2–3. 2.1.1

Fischler, M. A. and R. C. Bolles
1981. Random sample consensus. *Communications of the ACM*, 24(6). 4.2.1

Forsyth, D., O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan
2005. Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis. *Foundations and Trends in Computer Graphics and Vision*, 1(2-3):77–254. (Link). 1, 2.2

Fragkiadaki, K., S. Levine, P. Felsen, and J. Malik
2015. Recurrent Network Models for Human Dynamics. *2015 IEEE International Conference on Computer Vision (ICCV)*, Pp. 4346–4354. (Link). 2.2.2.4

Furukawa, Y. and J. Ponce
2008. Dense 3d motion capture from synchronized video streams. *CVPR*. 2.1.3, 3.4

Gall, J., C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel
2009. Motion capture using joint skeleton tracking and surface estimation. *CVPR*. 2.1.3, 3.4

Garg, R., A. Roussos, and L. Agapito
2013. Dense variational reconstruction of non-rigid surfaces from monocular video. *CVPR*. 2.2.2.1

Gavrila, D. and L. Davis

1996. Tracking of humans in action: A 3-D model-based approach. *ARPA Image Understanding Workshop*. 2.1.3

Girshick, R., J. Donahue, T. Darrell, and J. Malik

2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*. 3.4.2

Gleicher, M.

1997. Motion editing with spacetime constraints. *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*, Pp. 139–ff. (Link). 2.2.1

Gleicher, M.

2001. Comparing constraint-based motion editing methods. *Graphical Models*, 63(2). 2.2.1

Glowinski, D., M. Mancini, R. Cowie, and A. Camurri

2013. How Action Adapts to Social Context: The Movements of Musicians in Solo and Ensemble Conditions. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, Pp. 294–299. IEEE. (Link). 2.1.2

Goldin-Meadow, S. and M. Wagner Alibali

2013. Gesture's role in speaking, learning, and creating language. In *Annual Review of Psychology*. 4.1

Gorelick, L., M. Blank, E. Shechtman, M. Irani, and R. Basri

2007. Actions as space-time shapes. *IEEE transactions on pattern analysis and machine intelligence*, 29(12):2247–53. (Link). 2.1.2

Gotardo, P. and A. Martinez

2011. Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. *PAMI*. 2.2.2.3, 2.3, 5.2.1, 5.5.1.1, 5.5.2, 5.1, 5.5.2, 5.2, 5.8.3, 14

Grassia, F. S.

1998. Practical Parameterization of Rotations Using the Exponential Map. *Journal of Graphics Tools*, 3(3):29–48. 3.3.2.2

Greiner, T. M.

1991. Hand anthropometry of US army personnel. Technical report, DTIC Document. 7

Gross, M., S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt
2003. Blue-c: A spatially immersive display and 3d video portal for telepresence. *SIGGRAPH*. 2.1.3

Gross, R., I. Matthews, J. Cohn, T. Kanade, and S. Baker
2008. Multi-pie. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, Pp. 1–8. 4.5

Hamarneh, G. and T. Gustavsson
2004. Deformable spatio-temporal shape models: extending active shape models to 2D+time. *Image and Vision Computing*, 22(6):461 – 470. 2.2.2.3

Hartley, R. and A. Zisserman
2003. *Multiple View Geometry in Computer Vision*, volume 2. (Link). 3.3.2.2

Holden, D., J. Saito, and T. Komura
2016. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics*, 35(4):1–11. 2.2.2.4

Holden, D., J. Saito, T. Komura, and T. Joyce
2015. Learning Motion Manifolds with Convolutional Autoencoders. *SIGGRAPH Asia 2015 Technical Briefs*, Pp. 18:1—-18:4. (Link). 2.2.2.4

Huang, X., A. Acero, and H.-W. Hon
2001. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR. (Link). 1

Ichim, A. E., S. Bouaziz, and M. Pauly
2015. Dynamic 3D avatar creation from hand-held video input. *ACM Transactions on Graphics*, 34(4):45:1–45:14. (Link). 2.1.4

Insafutdinov, E., L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele
2016. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9910 LNCS, Pp. 34–50. 2.1.3, 2.1.4, 3.3

Ionescu, C., D. Papava, V. Olaru, and C. Sminchisescu
2014. Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *TPAMI*. 2.1.4

Jacobson, A. and O. Sorkine
2011. Stretchable and twistable bones for skeletal shape deformation. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)*, 30(6):165:1–165:8. 6.2.1.3, 3

Joo, H., H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh
2015. Panoptic studio: A massively multiview system for social motion capture. *ICCV*. 3.3.1, 4.4.2, 4.4.4

Joo, H., H. S. Park, and Y. Sheikh
2014. MAP Visibility Estimation for Large-Scale Dynamic 3D Reconstruction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Pp. 1122–1129. IEEE. (Link). 3.3.1, 5.8.6

Joo, H., T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh
2016. Panoptic studio: A massively multiview system for social interaction capture. *arXiv:1612.03153*. (Link). 3.3.1, 3.3.2

Jörg, S., J. Hodgins, and A. Safonova
2012. Data-driven finger motion synthesis for gesturing characters. *ACM Trans. Graph.*, 31(6):189:1—-189:7. (Link). 2.1.4

Kanade, T., P. Rander, and P. Narayanan
1997. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*. 2.1.3

Kehl, R. and L. V. Gool
2006. Markerless tracking of complex human motions from multiple views. *CVIU*. 2.1.3

Kendon, A.
1994. Do Gestures Communicate? A Review. *Research on Language & Social Interaction*, 27(3):175–200. (Link). 1

Keskin, C., F. Kiracc, Y. E. Kara, and L. Akarun
2012. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *ECCV*. 2.1.4

Krauss, R.
1998. Why do we gesture when we speak? In *Current Directions in Psychological Science*, volume 7, Pp. 54–59. 4.1

Krug, D., J. Arick, and P. Almond

1980. Behavior checklist for identifying severely handicapped individuals with high levels of autistic behavior. *Journal of child psychology and psychiatry*, 21(3):221—229. (Link). 1

Laban, R.

1926. *Choreographie: Erstes Heft.* Jena: Eugen Diedrichs. (Link). 1

Laub, A. J.

2004. *Matrix Analysis For Scientists And Engineers.* Society for Industrial and Applied Mathematics. 8

Lawrence, N. D.

2004. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in Neural Information Processing Systems.* 2.2.2

Le, H. and D. G. Kendall

1993. The riemannian structure of euclidean shape spaces: A novel environment for statistics. *The Annals of Statistics*, 21(3):pp. 1225–1271. 2.2.2.1, 5.3.1

Le, V., J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang

2012. Interactive facial feature localization. In *European Conference on Computer Vision (ECCV).* 4.5

Lee, M., J. Cho, C.-H. Choi, and S. Oh

2013a. Procrustean normal distribution for non-rigid structure from motion. *CVPR.* 2.2.2.1

Lee, M., J. Cho, C.-H. Choi, and S. Oh

2013b. Procrustean normal distribution for non-rigid structure from motion. In *CVPR*, Pp. 1280–1287. 5.5.2, 5.3, 5.8.3

Lee, M., C.-H. Choi, and S. Oh

2014. A procrustean markov process for non-rigid structure recovery. *CVPR.* 2.2.2.3, 5.5.2, 5.9

Lepri, B., R. Subramanian, K. Kalimeri, J. Staiano, F. Pianesi, and N. Sebe

2012. Connecting meeting behavior with extraversion&# x2014; a systematic study. *IEEE Transactions on Affective Computing.* 2.1.2

Levoy, M., J. Ginsberg, J. Shade, D. Fulk, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, and J. Davis

2000. The digital Michelangelo project. In *Proceedings of the 27th annual conference on*

Computer graphics and interactive techniques - SIGGRAPH '00, Pp. 131–144, New York, New York, USA. ACM Press. (Link). 3.3.1

Li, L., J. McCann, N. S. Pollard, and C. Faloutsos
2009. Dynammo: mining and summarization of coevolving sequences with missing values. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Pp. 507–516. 2.2.2.4

Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick
2014. Microsoft coco: Common objects in context. In *ECCV*. 2.1.4, 7

Littlewort, G., M. S. Bartlett, and K. Lee
2007. Faces of pain: automated measurement of spontaneousallfacial expressions of genuine and posed pain. *of the 9th international conference on*. (Link). 2.1.2

Liu, J., P. Carr, R. T. Collins, and Y. Liu
2013a. Tracking Sports Players with Context-Conditioned Motion Models. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Pp. 1830–1837. IEEE. (Link). 2.1.2

Liu, Y., J. Gall, C. Stoll, Q. Dai, H.-P. Seidel, and C. Theobalt
2013b. Markerless motion capture of multiple characters using multiview image segmentation. *TPAMI*. 2.1.3, 4.1

Loper, M., N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black
2015. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16. 6.2.1, 6.3, 6.2.1.1, 6.2.1.1, 6.2.2.1

Lou, H. and J. Chai
2010. Example-based human motion denoising. *IEEE Transactions on Visualization and Computer Graphics*, 16:870–879. 2.2.2.3

Lu, S., D. Metaxas, D. Samaras, and J. Oliensis
2003. Using multiple cues for hand tracking and model refinement. In *CVPR*. 2.1.4

Lundberg, U., R. Kadefors, B. Melin, G. Palmerud, P. Hassmen, M. Engstrom, and I. E. Dohns
1994. Psychophysiological stress and EMG activity of the trapezius muscle. *International journal of behavioral medicine*, 1(4):354–70. (Link). 2.1.1

Majkowska, A. D., V. B. Zordan, and P. Faloutsos

2006. Automatic Splicing for Hand and Body Animations. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2006*, Pp. 309–316. (Link). 2.1.4

Mardia, K. V. and I. L. Dryden

1989. Shape distributions for landmark data. *Advances in Applied Probability*, 21(4):pp. 742–755. 2.2.2.1, 5.3.1, 5.3.1

Mathias, M., R. Benenson, M. Pedersoli, and L. Van Gool

2014. Face detection without bells and whistles. In *ECCV*. 4.1

Matsuyama, T. and T. Takai

2002. Generation, visualization, and editing of 3d video. *3DPVT*. 2.1.3

Matusik, W., C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan

2000. Image-based visual hulls. *SIGGRAPH*. 2.1.3

Mazumder, R., T. Hastie, and R. Tibshirani

2010. Spectral regularization for learning large incomplete matrices. *JMLR*. 5.6.2

McDuff, D., R. El Kaliouby, T. Senechal, D. Demirdjian, and R. Picard

2014. Automatic measurement of ad preferences from facial responses gathered over the Internet. *Image and Vision Computing*, 32(10):630–640. (Link). 2.1.2

McKee, R., D. McKee, D. Alexander, and E. Paillat

. NZ sign language exercises. *Deaf Studies Department of Victoria University of Wellington*, `http://www.victoria.ac.nz/llc/llc_resources/nzsl/`. 4.2, 4.4.1

McKeown, G., M. F. Valstar, R. Cowie, and M. Pantic

2010. The semaine corpus of emotionally coloured character interactions. In *2010 IEEE International Conference on Multimedia and Expo, ICME 2010*, Pp. 1079–1084. 2.1.2

Metaxas, D. and D. Terzopoulos

1993. Shape and nonrigid motion estimation through physics-based synthesis. *PAMI*. 2.2.1, 2.2.2.3

Min, J., Y.-L. Chen, and J. Chai

2009. Interactive generation of human animation with deformable motion models. *ACM Transactions on Graphics*, 29(1):9:1–9:12. 2.2.2.3

Mitchell, S., J. Bosch, B. Lelieveldt, R. van der Geest, J. Reiber, and M. Sonka

2002. 3-D active appearance models: segmentation of cardiac MR and ultrasound images. *IEEE Transactions Transactions on Medical Imaging*, 21(9):1167–1178. 2.2.2.3

Newcombe, R. A., A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon
2011. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, Pp. 127–136. IEEE. (Link). 3.3.1, 6.2.2.2, 6.2.2.2

Newell, A., K. Yang, and J. Deng
2016. Stacked hourglass networks for human pose estimation. *arXiv preprint arXiv:1603.06937*. 2.1.3, 2.1.4

Oberweger, M., G. Riegler, P. Wohlhart, and V. Lepetit
2016. Efficiently Creating 3D Training Data for Fine Hand Pose Estimation. In *CVPR*. 2.1.4

Oberweger, M., P. Wohlhart, and V. Lepetit
2015. Training a Feedback Loop for Hand Pose Estimation. In *ICCV*. 2.1.4

Oikonomidis, I., N. Kyriazis, and A. A. Argyros
2012. Tracking the articulated motion of two strongly interacting hands. In *CVPR*. 2.1.4

Olsen, S. and A. A. Bartoli
2008. Implicit non-rigid structure-from-motion with priors. *Journal of Mathematical Imaging and Vision*. 2.2.2.3

Park, H. S. and J. Shi
2015. Social saliency prediction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, Pp. 4777–4785. 3.3

Park, H. S., T. Shiratori, I. Matthews, and Y. Sheikh
2010. 3D reconstruction of a moving point from a series of 2D projections. *ECCV*. 2.2.2.2, 5.8.4, 5.14

Park, J., S. N. Sinha, Y. Matsushita, Y.-W. Tai, and I. S. Kweon
2013. Multiview Photometric Stereo Using Planar Mesh Parameterization. In *2013 IEEE International Conference on Computer Vision*, Pp. 1161–1168. IEEE. (Link). 3.3

Park, S. I. and J. K. Hodgins
2008. Data-driven modeling of skin and muscle deformation. *TOG*. 5.8.2

Pavlidis, I., N. L. Eberhardt, and J. A. Levine
2002. Seeing through the face of deception. Technical Report 6867, Honeywell Laboratories, 3660 Technology Drive, Minneapolis, Minnesota 55418, USA. 2.1.2

Pease, A.
1981. *Body Language: How to Read Others' Thoughts by Their Gestures.* (Link). 1

Pentland, A. and B. Horowitz
1993. Recovery of nonrigid motion & structure. *PAMI*. 2.2.1, 2.2.2.3

Perperidis, D., R. Mohiaddin, and D. Rueckert
2004. Spatio-temporal free-form registration of cardiac MR image sequences. In *Medical Image Computing and Computer-Assisted Intervention*, C. Barillot, D. R. Haynor, and P. Hellier, eds., volume 3216 of *Lecture Notes in Computer Science*, Pp. 911–919. Springer Berlin / Heidelberg. 2.2.2.3

Petit, B., J.-D. Lesage, E. Boyer, and B. Raffin
2009. Virtualization Gate. *SIGGRAPH Emerging Technologies*. 2.1.3

Phillips, P., P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek
2005. Overview of the Face Recognition Grand Challenge. In *IEEE International Conference on Computer Vision & Pattern Recognition(CVPR)*. 4.5

Pishchulin, L., E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele
2015. DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. *arXiv:1511.06645 [cs]*, Pp. 4929–4937. (Link). 2.1.3

Plankers, R. and P. Fua
2003. Articulated Soft Objects for Multi-View Shape and Motion Capture. *TPAMI*. 2.1.3

Ramakrishna, V., D. Munoz, M. Hebert, A. J. Bagnell, and Y. Sheikh
2014. Pose machines: Articulated pose estimation via inference machines. In *ECCV*. 2.1.3, 4.3

Ramanathan, V., B. Yao, and L. Fei-Fei
2013. Social role discovery in human events. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Pp. 2475–2482. 2.1.2

Ramseyer, F. and W. Tschacher
2014. Nonverbal synchrony of head- and body-movement in psychotherapy: different signals have different associations with outcome. *Frontiers in psychology*, 5:979. (Link). 1

Rao, C. R.
1973. *Linear Statistical Inference and its Applications.* Wiley-Interscience. 5.6.2

Rehg, J. M., G. D. Abowd, A. Rozga, M. Romero, M. a. Clements, S. Sclaroff, I. Essa, O. Y. Ousley, Y. Li, C. Kim, H. Rao, J. C. Kim, L. L. Presti, J. Zhang, D. Lantsman, J. Bidwell, and Z. Ye
2013. Decoding Children's Social Behavior. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Pp. 3414–3421. (Link). 2.1.2

Rehg, J. M. and T. Kanade
1994. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Motion of Non-Rigid and Articulated Objects*. 2.1.4

Rhodin, H., C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt
2016. Egocap: Egocentric marker-less motion capture with two fisheye cameras. In *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia)*. 2.1.3, 4.1

Rodriguez, M. D., J. Ahmed, and M. Shah
2008. Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Pp. 1–8. IEEE. (Link). 2.1.2

Rosales, R., V. Athitsos, L. Sigal, and S. Sclaroff
2001. 3D hand pose reconstruction using specialized mappings. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 1(2000):378–385. (Link). 2.1.4

Rudovic, O., M. A. Nicolaou, and V. Pavlovic
2014. Machine Learning Methods for Social Signal Processing. In *Social Signal Processing*. Cambridge University Press (In Press). (Link). 2.1.2

Rue, H. and L. Held
2005. *Gaussian {M}arkov Random Fields: {T}heory and Applications*, volume 104 of *Monographs on Statistics and Applied Probability*. London: Chapman & Hall. 5.4, 5.4.2

Russell, C., J. Fayad, and L. Agapito
2011. Energy based multiple model fitting for non-rigid structure from motion. *CVPR*. 2.2.2.1

Saad, Y.
1996. Iterative Methods for Sparse Linear Systems. *IEEE Computational Science and Engineering*, 3:88–88. (Link). 6.2.3.2

Sagonas, C., E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic
2016. 300 faces in-the-wild challenge. *Image and Vision Computing (IMAVIS)*, 47:3–18. 4.1, 4.5

Sagonas, C., G. Tzimiropoulos, S. Zafeiriou, and M. Pantic
2013. 300 faces in-the-wild challenge: the first facial landmark localization challenge. In *IEEE International Conference on Computer Vision Workshop on 300 Faces in-the-Wild Challenge (300-W)*. 4.5

Salzmann, M. and R. Urtasun
2011. Physically-based motion models for 3d tracking: A convex. formulation. *ICCV*. 2.2.1, 5.5.2

Sandbach, G., S. Zafeiriou, M. Pantic, and L. Yin
2012. Static and dynamic 3D facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30(10):683–697. (Link). 2.1

Sapir, E.
1928. The Unconscious Patterning of Behavior in Society. In *The Unconscious: A Symposium*, Pp. 114–142. 1

Schölkopf, B., A. J. Smola, and K.-R. Müller
1997. Kernel principal component analysis. In *Artificial Neural Networks, 7th International Conference*, Pp. 583–588. 2.2.2

Schuldt, C., I. Laptev, and B. Caputo
2004. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, Pp. 32–36 Vol.3. IEEE. (Link). 2.1.2

Sharp, T., C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al.
2015. Accurate, robust, and flexible real-time hand tracking. In *CHI*. 2.1.4

Shotton, J., A. Fitzgibboan, M. Cook, and T. Sharp
2011. Real-time human pose recognition in parts from single depth images. *CVPR*. 4.1

Shotton, J., T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore
2013. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*. 2.1.4

Sidenbladh, H., M. Black, and D. Fleet
2000a. Stochastic tracking of 3d human figures using 2d image motion. *ECCV*. 5.5.2

Sidenbladh, H., M. J. Black, and D. J. Fleet

2000b. Stochastic tracking of 3D human figures using 2D image motion. In *Proceedings of the European Conference on Computer Vision*, Pp. 702–718. 2.2.2.2, 5.3.2

Simonyan, K. and A. Zisserman

2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556. 4.3.1

Sridhar, S., F. Mueller, A. Oulasvirta, and C. Theobalt

2015. Fast and robust hand tracking using detection-guided optimization. In *CVPR*. 2.1.4

Sridhar, S., F. Mueller, M. Zollhoefer, D. Casas, A. Oulasvirta, and C. Theobalt

2016. Real-time joint tracking of a hand manipulating an object from RGB-D input. In *ECCV*. 2.1.4, 10

Sridhar, S., A. Oulasvirta, and C. Theobalt

2013. Interactive markerless articulated hand motion tracking using RGB and depth data. In *ICCV*. 2.1.4

Stenger, B., A. Thayananthan, P. H. Torr, and R. Cipolla

2006. Model-based hand tracking using a hierarchical bayesian filter. *TPAMI*. 2.1.4

Stoll, C., N. Hasler, J. Gall, H.-P. Seidel, and C. Theobalt

2011. Fast articulated motion tracking using a sums of gaussians body model. *ICCV*. 2.1.3

Strang, G.

1999. The discrete cosine transform. *SIAM review*. 5.4.1

Sun, X., A. Nijholt, K. P. Truong, and M. Pantic

2011. Automatic visual mimicry expression analysis in interpersonal interaction. In *CVPR 2011 WORKSHOPS*, Pp. 40–46. IEEE. (Link). 2.1.2

Sun, X., Y. Wei, S. Liang, X. Tang, and J. Sun

2015. Cascaded hand pose regression. In *CVPR*. 2.1.4

Supancic, J. S., G. Rogez, Y. Yang, J. Shotton, and D. Ramanan

2015. Depth-based hand pose estimation: data, methods, and challenges. In *ICCV*. 2.1.4

Tang, D., H. Jin Chang, A. Tejani, and T.-K. Kim

2014. Latent regression forest: Structured estimation of 3D articulated hand posture. In *CVPR*. 2.1.4

Tang, D., T.-H. Yu, and T.-K. Kim

2013. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*. 2.1.4

Taylor, G. W., G. E. Hinton, and S. Roweis

2007. Modeling Human Motion Using Binary Latent Variables. *Advances in Neural Information Processing Systems (NIPS)*, Pp. 1345–1352. 2.2.2.4

Taylor, J., A. Jepson, and K. Kutulakos

2010. Non-rigid structure from locally-rigid motion. *CVPR*. 2.2.1

Terzopoulos, D., A. Witkin, and M. Kass

1988. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial Intelligence*. 2.2.1

Thies, J., M. Zollhöfer, and M. Stamminger

2016. Face2face: Real-time face capture and reenactment of rgb videos. *... Vision and Pattern ...*, Pp. 2387–2395. (Link). 2.1.4

Thrun, S., W. Burgard, and D. Fox

2006. *Probabilistic Robotics*. Cambridge University Press. 2.2.2.4

Tomasi, C. and T. Kanade

1992. Shape and motion from image streams under orthography: a factorization method. *IJCV*. 2.2.2.1

Tompson, J., M. Stein, Y. Lecun, and K. Perlin

2014a. Real-time continuous pose recovery of human hands using convolutional networks. *ACM TOG*. 2.1.4, 4.3, 10

Tompson, J. J., A. Jain, Y. LeCun, and C. Bregler

2014b. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*. 2.1.3, 2.1.4, 4.3

Torresani, L. and C. Bregler

2002. Space-time tracking. In *Proceedings of the European Conference on Computer Vision*, Pp. 801–812. 2.2.2.2

Torresani, L., A. Hertzmann, and C. Bregler

2008. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*. 2.2.2.3, 5.3.1, 5.3.2, 5.5.2, 5.1

Tzionas, D., L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall
2016. Capturing hands in action using discriminative salient points and physics simulation. *IJCV*. 2.1.4, 4.4.3, 4.1

Urtasun, R., P. Glardon, R. Boulic, D. Thalmann, and P. Fua
2004. Style-based motion synthesis. *Computer Graphics Forum*, 23(4):799–812. 2.2.2.3

Valmadre, J. and S. Lucey
2012. A general trajectory prior for non-rigid reconstruction. *CVPR*. 2.2.2.2, 5.3.2, 5.5.2, 5.1, 5.8.2

Van Loan, C. F.
2000. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123(1-2):85–100. (Link). 5.4.3

Varni, G., G. Volpe, and A. Camurri
2010. A System for Real-Time Multimodal Analysis of Nonverbal Affective Social Interaction in User-Centric Media. *IEEE Transactions on Multimedia*, 12(6):576–590. (Link). 2.1.2

Vecchio, D. D., R. Murray, and P. Perona
2002. Primitives for human motion: a dynamical approach. *IFAC World Congress*. (Link). 1

Vidal, R. and D. Abretske
2006. Nonrigid shape and motion from multiple perspective views. *ECCV*. 2.2.2.1

Vinciarelli, A., M. Pantic, D. Heylen, C. Pelachaud, I. Poggi, F. D'Errico, and M. Schroeder
2012. Bridging the Gap between Social Animal and Unsocial Machine: A Survey of Social Signal Processing. *IEEE Transactions on Affective Computing*, 3(1):69–87. (Link). 2.1.2

Vlasic, D., I. Baran, W. Matusik, and J. Popović
2008. Articulated mesh animation from multi-view silhouettes. *ACM TOG*. 2.1.3

Vlasic, D., P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik
2009. Dynamic shape capture using multi-view photometric stereo. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 28(5). 2.1.3, 6.1

Vo, M., S. G. Narasimhan, and Y. Sheikh
2016. Spatiotemporal Bundle Adjustment for Dynamic 3D Reconstruction. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4.6

Wan, C., A. Yao, and L. Van Gool
2016. Direction matters: hand pose estimation from local surface normals. *arXiv preprint arXiv:1604.02657*. 2.1.4

Wang, J., D. Fleet, and H. Aaron
2008. Gaussian process dynamical models for human motion. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 30:283–298. 2.2.2.4

Wang, R. Y. and J. Popović
2009. Real-time hand-tracking with a color glove. *ACM TOG*, 28(3). 2.1.4

Wei, S.-E., V. Ramakrishna, T. Kanade, and Y. Sheikh
2016. Convolutional pose machines. In *CVPR*. 2.1.3, 2.1.4, 3.3, 3.12, 3.4.1, 3.4.2, 4, 3.4.3, 4.3, 4.3.1, 4.3.2

Weise, T., S. Bouaziz, H. Li, and M. Pauly
2011. Realtime performance-based facial animation. *ACM Transactions on Graphics*, 30(4):1. 2.1.4

Weisstein, E. W.
2016. Square line picking. From MathWorld–A Wolfram Web Resource, http://mathworld.wolfram.com/SquareLinePicking.html. 4

Whitehill, J., Z. Serpell, A. Foster, and J. R. Movellan
2014. The Faces of Engagement: Automatic Recognition of Student Engagementfrom Facial Expressions. *IEEE Transactions on Affective Computing*, 5(1):86–98. (Link). 2.1.2

Witkin, A. and M. Kass
1988. Spacetime constraints. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, Pp. 159–168. 2.2.1

Wu, C., C. Stoll, L. Valgaerts, and C. Theobalt
2013. On-set performance capture of multiple actors with a stereo camera. *SIGGRAPH*. 6.1

Xiao, J., J. Chai, and T. Kanade
2004. A closed-form solution to non-rigid shape and motion recovery. *ECCV*. 5.5.2

Xiong, X. and F. De La Torre
2013. Supervised descent method and its applications to face alignment. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Pp. 532–539. 2.1.4

Xu, C. and L. Cheng
2013. Efficient hand pose estimation from a single depth image. In *ICCV*. 2.1.4

Yan, J. and M. Pollefeys
2005. A factorization-based approach to articulated motion recovery. *CVPR*. 2.2.2.1

Yang, Y. and D. Ramanan
2013. Articulated Human Detection with Flexible Mixtures-of-Parts. *TPAMI*. 3.4.7

Ye, G., Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt
2012. Performance capture of interacting characters with handheld kinects. *ECCV*. 2.1.3

Ye, Q., S. Yuan, and T.-K. Kim
2016. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. *ECCV*. 2.1.4

Yu, G., Z. Liu, and J. Yuan
2015. Discriminative Orderlet Mining for Real-Time Recognition of Human-Object Interaction. In *Asian Conference on Computer Vision (ACCV)*, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, eds., volume 9007 of *Lecture Notes in Computer Science*, Cham. Springer International Publishing. (Link). 2.1.2

Yuan, J., Z. Liu, and Y. Wu
2009. Discriminative subvolume search for efficient action detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Pp. 2442–2449. IEEE. (Link). 2.1.2

Zelnik-Manor, L. and M. Irani
2004. Temporal Factorization vs. Spatial Factorization. In *European Conference on Computer Vision (ECCV)*, T. Pajdla and J. Matas, eds., volume 3022 of *Lecture Notes in Computer Science*, Berlin, Heidelberg. Springer Berlin Heidelberg. (Link). 2.2.2.2

Zelnik-Manor, L. and M. Irani
2006. Statistical analysis of dynamic actions. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1530–5. (Link). 2.2.2.3

Zen, G., B. Lepri, E. Ricci, and O. Lanz
2010. Space speaks: towards socially and personality aware visual surveillance. *ACM International Workshop on Multimodal Pervasive Video Analysis*. 2.1.2

Zhu, X. and D. Ramanan
2012. Face detection, pose estimation, and landmark localization in the wild. In *IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*. 4.5