DPHIL THESIS

VISUAL ANALYSIS OF ARTICULATED MOTION

PHILIP A. TRESADERN

October 12, 2006

ROBOTICS RESEARCH GROUP
DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY OF OXFORD

This thesis is submitted to the Department of Engineering Science, University of Oxford, for the degree of Doctor of Philosophy. This thesis is entirely my own work and, except where otherwise indicated, describes my own research.

For Mum and Dad

Philip A. Tresadern
Exeter College

Doctor of Philosophy
October 12, 2006

VISUAL ANALYSIS OF ARTICULATED MOTION

Abstract

The ability of machines to recognise and interpret human action and gesture from standard video footage has wide-ranging applications for control, analysis and security. However, in many scenarios the use of commercial motion capture systems is undesirable or infeasible (*e.g.* intelligent surveillance). In particular, commercial systems are restricted by their dependence on markers and the use of multiple cameras that must be synchronized and calibrated by hand. It is the aim of this thesis to develop methods that relax these constraints in order to bring inexpensive, off-the-shelf motion capture several steps closer to a reality.

In doing so, we demonstrate that image projections of important anatomical landmarks on the body (specifically, joint centre projections) can be recovered automatically from image data. One approach exploits geometric methods developed in the field of Structure From Motion (SFM), whereby point features on the surface of an articulated body impose constraints on the hidden joint locations, even for a single view. An alternative approach explores Machine Learning to employ context-specific knowledge about the problem in the form of a corpus of training data. In this case, joint locations are recovered from similar exemplars in the training set via searching, sampling or regression.

Having recovered such points of interest in an image sequence, we demonstrate that they can be used to synchronize and calibrate a pair of cameras, rather than employing complex engineering solutions. We present a robust algorithm for synchronizing two sequences, of unknown and different frame rates, to sub-frame accuracy. Following synchronization, we recover affine structure using standard methods. The recovered affine structure is then upgraded to a Euclidean co-ordinate frame via a novel self-calibration procedure that is shown to be several times more efficient than existing methods without sacrificing accuracy.

Throughout the thesis, methods are quantitatively evaluated on synthetic data for a ground truth comparison and qualitatively demonstrated on real examples.

Acknowledgements

Many thanks go first to my supervisor, Dr. Ian Reid, for his enthusiastic support during the good times and endless patience during the bad. Papers always sounded better after his comments and suggestions, ideas came thick and fast, and he was always there to steer me away from the more torturous paths ahead.

Thanks also go to all members of the Active Vision and Visual Geometry groups at Oxford. They are a source of inspiration, enthusiasm and assistance whenever required. Joint thanks must also go to the staff of the Royal Oak, Woodstock Rd, for their good service during the weekly post-reading-group lab banter.

My time in Oxford would have been a much less pleasant experience had it not been for the good people I socialized with during my stay. In particular, thanks to Adrian and Nick for the *numerous* hours spent down the pub patiently listening to my griping about the PhD, only to return the favour and reminding me I wasn't alone in my frustration. Thanks also to absent friends Emily and Diane - we miss you.

Special thanks must go to Joanne for being such a loving companion during an otherwise difficult year.

Thanks also go to friends from outside of the dreaming spires – Ste, Andy, Matt, Tim, Chris, Rebecca, Melissa, Charlie, Gill etc. etc. Whenever Oxford felt a little too small for comfort, they were there to remind me that there is another world outside, too.

Finally, of course, ∞ thanks go to my parents for their love and support, both emotional and financial. Their appreciation of the education system that their country has to offer and the encouragement of their children to make the most of it got myself, Nick and Simon where we are today. Thanks, folks – I'm dead proud.

And to anyone I've forgotten to mention - thanks and apologies. I'm sure I'll remember you later and feel sorry that I ever forgot in the first place.

Contents

Chapter 1

Introduction

*The ability to interpret actions and "body language" is arguably the ability that has enabled humans to form complex social structures and become the dominant species on the planet. This thesis focuses on a computational solution to this problem, known as Human Motion Capture (HMC), where we wish to recover the human body pose in each frame of an image sequence. In this first chapter, we introduce HMC in the wider context of Machine Vision before outlining its applications, commercial (*i.e. *markered) solutions and limitations. We then discuss markerless systems that exist in research environments, the problems they overcome and the problems yet to be solved.*

1.1 Background

Human beings absorb much of their information regarding the real world via visual input. This visual input is essential for day-to-day tasks such as searching for food, detecting and avoiding hazards, and navigating within our environment. The aim of Machine Vision is to replicate this faculty using cameras and computers, rather than the eyes and brain, to receive and process the data, thus bestowing the same abilities on mobile robots and intelligent computer systems of the future.

Since the mapping from the 3D world to a 2D image incurs significant information loss (*i.e*. depth), we impose constraints, typically encoded as assumptions or rules learned from experience, to rule out spurious or inconsistent interpretations of complex scenes. Indeed, these assumptions are sufficiently strong that they may induce

Figure 1.1: Two twins in an Ames room.

an *incorrect* interpretation of the scene geometry, as demonstrated by optical illusions such as the Ames room (Figure 1.1).

This thesis focusses on constraints that apply to images of articulated objects. We define an articulated object as any structure that is *piecewise* rigid but deforms according to a finite number of degrees of freedom. Since a rigid body has 6 degrees of freedom (corresponding to translation and orientation in 3D), a collection of N rigid bodies will in general have $6N$ degrees of freedom. However, articulation between objects reduces the number of degrees of freedom such that the structure can be completely determined by $< 6N$ parameters.

Articulated objects are of considerable interest to us since they are abundant in our environment, ranging from furniture fittings and mechanical linkages to biological organisms, including the human body itself. It is our highly developed ability to interpret images of such dynamic structures that have enabled humans to interact and communi-

cate with each other, arguably resulting in our complex social structure and becoming the dominant species on the planet.

This ability was vividly demonstrated some years ago by Johansson [59] who introduced the famous "Moving Light Displays". In these experiments, human subjects, dressed entirely in black, walked in front of a black background such that bright lights placed close to anatomical joints (*e.g*. shoulders, knees) provided the only visual stimulus. Surprisingly, it was noted that 'all [observer]s, without any hesitation, reported seeing a walking human being' after being exposed to just one second of footage. It appears that our brains are so well tuned to recognizing human motion that we are able to form a correct interpretation of even the most limited visual input.

It is the aim of this thesis to develop a similar ability for machines. Specifically, given an image (or image sequence) of a human in motion, we would like to recover the pose (position and orientation of the body, plus angles at joints) at every instant in time. Sequences of poses define gestures that may then be analysed for higher level interpretation. We refer to this process as *Human Motion Capture*.

1.2 Applications

The applications of human motion capture are highly diverse but can be separated approximately into three principal areas: control, analysis and surveillance.

1.2.1 Control

In many applications, the recovered pose is used as input to *control* a system. A particularly prominent end-user in this category is the entertainment industry, where human motion capture is used to drive a computer generated character (avatar) in movies (*e.g*. Gollum from 'The Lord of the Rings', Figure 1.2) and video games (*e.g*. Lara

Figure 1.2: (left) An actor, wearing markers during motion capture. (right) The captured pose applied to the virtual character, Gollum.

Croft from 'Tomb Raider'). For accurate reproduction of movement, commercial systems are employed in an off-line process (see Section 1.3).

If only approximate movement is required, simple image processing can be used to control the system in real-time as demonstrated in systems such as the Sony i-Toy. This device provides a novel interface for video games whereby gross movements of the user are translated directly into actions on the screen, resulting in a more interactive experience.

Alternatively, rather than mimicking the observed actions it may be desirable to *react to* the human motion. This is particularly the case in humanoid robotics where a natural human-machine interface is required for the robots to become more socially

acceptable.

1.2.2 Analysis

Motion capture systems are also commonly used as an *analysis* tool. In medicine, for example, commercial systems are used to analyse motion data for biomechanical modelling, diagnosis of pathology and post-injury rehabilitation. Until recently, the most common medical application was in gait analysis where kinematic motion data would be augmented with kinetic data acquired using force plates. However, motion capture is now being employed for the analysis of upper-body movements. For example, motion capture data of the arm during reaching and grasping is being used to develop algorithms to trigger Functional Electrical Stimulation (FES) of the muscles at the correct time for patients that have suffered a stroke or spinal cord injury [109].

1.2.3 Surveillance

In contrast, *surveillance* applications cannot be implemented using commercial systems since the subjects are (by definition) unaware that they are under observation and therefore do not willingly participate in the motion capture process. In most cases, however, the level of required accuracy is much lower than in other applications – often we need only to *detect* suspicious behaviour. This is a rapidly growing application area (especially given the current security climate) and is closely linked to *biometrics* where gait could be used for identification [89] when the subject is too far away to make conventional measurements (*e.g.* iris pattern, fingerprints, speech, face recognition).

Figure 1.3: A typical motion capture studio employing ten cameras. A minimum of three cameras are required although for the system to be robust to tracking error and self-occlusion of markers, many more are usually employed.

1.3 Commercial Motion Capture

There are a number of commercial motion capture systems on the market (*e.g.* Vicon [119]). In this system, infra-red cameras observe a workspace under the illumination of infra-red strobe lamps located close to the cameras. Retro-reflective markers, attached to tight fitting clothing worn by the actor, reflect the incoming rays from the lamps directly back to the cameras such that the markers appear as bright dots in the image. The use of infra-red cameras (rather than the visible spectrum) ensures a high contrast between the markers and background in the image.

Knowing the locations of these dots in the images together with the positions of the cameras in the workspace gives the 3D position of each marker at every instant in time. From these 3D marker locations, joint centre locations are inferred (by treating each

limb as a rigid body) in order to compute the pose of the underlying skeleton.

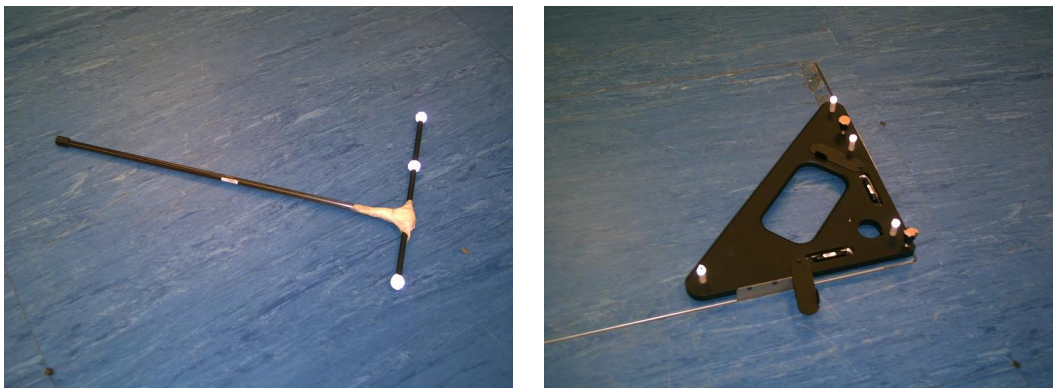
1.3.1 Limitations

Figure 1.3 shows a typical motion capture studio with ten cameras. The system is necessarily complex to overcome the various number of limitations of this approach:

- **Joint centre occlusion:** Since the joint centre is hidden under skin and muscle, is it inferred from the relative motion of markers on the *surface* of adjacent body segments via a calibration procedure where the actor performs an artificial movement. However, the markers may restrict the movement of the actor and are easily brushed off during vigorous movement. Furthermore, the movement of the skin over underlying tissue violates the assumption that a limb is a rigid body, increasing uncertainty in the estimate of the joint centre location.

- **Synchronization:** In order to triangulate the 3D positions of the markers from their 2D projections in multiple views, it is necessary to ensure that the image projections all correspond to the exact same instant in time (*i.e.* the cameras must be synchronized). This problem is addressed by generating a clock pulse from a common source to open all camera shutters at the same instant.

- **Calibration:** To triangulate the position of the markers, all cameras must be accurately calibrated with respect to a global co-ordinate frame. This is achieved via an off-line calibration process where the user waves a markered "wand" (Figure 1.4a) of accurately known geometry around the workspace. Each image in the sequence then contains a set of points corresponding to markers that are a known and fixed distance apart in the scene. Since the cameras are stationary, all images captured by a given camera can then be treated as a single image. From



(a) (b)

Figure 1.4: (a) Wand and (b) axes used during camera calibration.

the known geometry of the wand, the cameras are then calibrated with respect to each other. All cameras are then calibrated to a common co-ordinate frame using a markered structure representing the global X and Y axes (Figure 1.4b) located at the desired origin.

- **Spatial correspondence:** Although, in theory, only two views are required to triangulate 3D position from 2D images, it is necessary to ensure that we use the image of the *same* marker in each view to compute its 3D position. It can be shown that the image of a marker in one view constrains the location of the corresponding image in a second view to lie on a *line* (the epipolar line) such that an infinite number of correspondences are possible. In stereo applications, this ambiguity is typically resolved by minimizing an error metric based on the rich image information (*e.g.* normalized cross-correlation). However, in the absence of rich image information (as in this case) a third camera is required to recover a consistent set of matched image features.

- **Marker occlusion:** Since markers are attached to the surface of the body, each marker is typically visible from only half of the workspace at any one time (Fig-

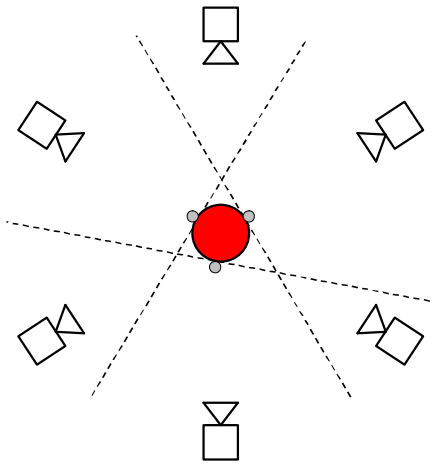


Figure 1.5: Marker occlusion. A marker on the surface of an opaque object is typically invisible to any camera on the opposite side of the tangent plane. Therefore, in order to reconstruct all markers at any given frame, it is necessary to use at least six cameras that are evenly spaced around the workspace.

ure 1.5). Therefore, with cameras distributed evenly around the workspace at least six cameras are required for robust tracking. In practice, since the human body is highly non-convex markers are obscured more often (*e.g*. markers on the torso are occluded as the arm passes in front of the body). As a result, motion capture systems typically employ at least seven cameras and even then, complex post-processing is usually required to fill in small periods of marker occlusion.

From these limitations, we see that markers provide the greatest strength but also the Achilles' Heel of commercial motion capture systems. Not only are markers cumbersome and unsuitable for surveillance applications but they reduce the rich data contained in an image (due to colour, texture, edges *etc*.) to a number of point features. Engineering solutions to the limitations described above only add to the technical complexity and cost of commercial systems.

1.4 Markerless Motion Capture

We now consider systems that recover pose by employing the rich data available in standard image sequences. In such cases, problems such as marker self-occlusion are avoided since the entire surface of the limb is employed rather than a finite set of points from it. Furthermore, the rich data available provides additional cues (*e.g.* edges, perspective, texture variation) that may permit a solution using a single camera such that synchronization and calibration become unnecessary. Other problems, such as joint centre occlusion, are intrinsic to the problem and therefore present in both markerless and markered motion capture systems.

1.4.1 Limitations

In spite of these promises, body parts can still be occluded by each other and multiple cameras are still desirable to increase accuracy so these problems are not entirely solved. We therefore focus on other problems introduced in such systems.

- **High dimensionality:** Since markers are no longer available, it is very difficult to track individual body parts independently whilst satisfying constraints imposed by articulated motion. As a result, it is commonly the case that the whole body is tracked in one go. However, due to the large number of degrees of freedom possessed by the human body the number of possible poses increases exponentially and tracking becomes computationally infeasible.

- **Appearance variation:** In markered motion capture, markers have a known appearance (*i.e.* high-contrast dots) in the image. However, due to lighting, orientation, clothing, build *etc.*, images of limbs captured using visible light cameras have a highly varied appearance that must be accounted for. This may be

achieved in part by discarding certain parts of the data (*e.g.* by using only the silhouette) but is largely an unsolved problem at this time.

1.5 Thesis Contributions

In this thesis, we investigate articulated motion with a bias toward human motion analysis. During the course of this investigation, we present methods that may prove beneficial in both markered and markerless tracking of the human body.[1]

We begin in Chapter 2 with a review of previous work, particularly in Human Motion Capture and Structure From Motion. Following this, we present contributions in four areas:

- Chapter 3 describes a geometric approach to recovering joint locations from a monocular image sequence alone. This is based upon the Structure from Motion paradigm, incorporating articulation constraints into the "factorization" method of Tomasi and Kanade [111].

- In contrast, Chapter 4 compares several different approaches that uses Machine Learning to estimate the joint locations from low-level image cues using a stored dataset of poses.

- Chapter 5 demonstrates how projected joint locations in the image are used to synchronize image sequences of the same motion. Joint locations from corresponding frames are then used to compute the pose of the subject in an affine coordinate frame using the factorization method.

- Chapter 6 details the self-calibration of the cameras, "upgrading" the recovered

[1] Parts of this thesis were previously published as [114, 115, 116].

affine structure to a metric co-ordinate frame where we are able to measure joint angles.

Chapter 7 concludes the thesis, outlines unfinished investigation and discusses the future direction of this work. Appendix A presents an empirical comparison of a number of shape representations for markerless motion capture including the recently proposed Histogram of Shape Contexts that has shown promise in this application area.

Chapter 2

Related work

The study of visual processes using computational methods was popularized by the seminal text of David Marr [69], a pioneer in the field now known as computational neuroscience. In this chapter, we present a brief review of selected papers from the two fields most relevant to this thesis: Human Motion Capture (HMC) and Structure From Motion (SFM).

2.1 Human Motion Capture

Due to the volume of literature regarding human motion tracking, we will not attempt to present a comprehensive review in this section (see [40, 6, 71] for more thorough surveys). Instead, we focus on the two seemingly opposite paradigms of model-based ("top down") and data-driven ("bottom up") tracking. In particular, we note the 'paradigm shift' from model-based to data-driven approaches during the 1990s and also how the two methodologies complement each other through *importance sampling*.

2.1.1 Tracking people from the top down

Top-down (or model-based) tracking refers to the process whereby an observation model, specifying how measurements are generated as a function of the state (pose), is combined (typically via Bayes' rule) with a predictive prior model that specifies our certainty of state before any measurements are made.

With a few exceptions (*e.g.* [12]), most model-based approaches to human motion

tracking are based upon the hierarchical kinematic model proposed by Marr and Nishihara [70]. This 3D model consists of a wireframe skeleton surrounded by volumetric primitives such as cylinders [70, 86, 93], spheres [78], truncated cones [41, 28, 122, 29], superquadrics [38, 21, 99] or complex polygonal meshes [61]. From a hand initialization in the first frame, the pose of this model is predicted at the next time step using a dynamical motion model. It is then reprojected in the predicted pose, compared with observations and a "best" estimate selected as some combination of the two.

Alternatively, using a 2D model requires fewer parameters to describe pose and does not suffer from kinematic singularities during monocular tracking [76]. However, perspective must be accounted for explicitly [60, 76] and only 2D pose is recovered, although by imposing constraints (*e.g.* anatomical joint limits) over the sequence it is possible to rule out implausible 3D poses [32].

Following the earliest examples of human motion analysis [78, 50, 86, 41], model-based tracking remained popular for many years since it is simple to implement, allows the recovery of joint angles in a 3D coordinate frame, and provides a framework for handling occlusion and self-intersection. However, there are also a number of difficult problems associated with human motion tracking. Bregler and Malik [21] tackle the issue of motion non-linearity using a first order approximation, employing a 'twist' notation to represent orientation. To address the issue of several possible solutions from a single view, many approaches use multiple cameras [38, 28, 61].

Density propagation

This approach to tracking is also known as a *generative* model approach and typically employs Bayes' rule to assimilate predictions with observations. Specifically, denoting the state at time t by x_t and the image data at time t by D_t, Bayes' rule states that:

$$p(x_t|D_t, D_{t-1}, \ldots) = \frac{p(D_t|x_t, D_{t-1}, \ldots)p(x_t|D_{t-1}, \ldots)}{p(D_t|D_{t-1}, \ldots)} \tag{2.1}$$

$$\propto p(D_t|x_t) \int p(x_t, x_{t-1}|D_{t-1}, \ldots)\, dx_{t-1} \tag{2.2}$$

$$= p(D_t|x_t) \int p(x_t|x_{t-1}, D_{t-1}, \ldots)p(x_{t-1}|D_{t-1}, \ldots)\, dx_{t-1} \tag{2.3}$$

$$= p(D_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|D_{t-1}, \ldots)\, dx_{t-1} \tag{2.4}$$

where sensible independence assumptions have been made.

In this form, $p(x_t|D_t, D_{t-1}, \ldots)$ is the *posterior* probability density that takes into account predictions and observations. The *likelihood*, $p(D_t|x_t)$, reflects how well a predicted state matches the current measurements via an observation model. Similarly, the *prior*, $p(x_t|x_{t-1})$, specifies how the state is expected to evolve from one time instant to the next via a predictive motion model. The posterior from the previous time instant, $p(x_{t-1}|D_{t-1}, \ldots)$, is therefore propagated through time via (2.4).

Multiple hypothesis tracking and the CONDENSATION algorithm

In order to combine the prediction and observations in an optimal way, many systems employed the Kalman Filter (KF) or Extended Kalman Filter (EKF). These have the desirable property that the posterior can be propagated analytically in a computationally optimal way (see Figure 2.1), *as long as the noise distribution is Gaussian* (and hence unimodal).

However, in practice the observation likelihood is seldom expressible in an analytical form as a result of the many local maxima (due to clutter, kinematic ambiguities, self-occlusion *etc*.) and tracking is easily lost. Nonetheless, it is generally possible to *evaluate* the likelihood at a given value of x_t. This property was exploited by methods that could support multiple hypotheses such that ambiguities could be resolved using

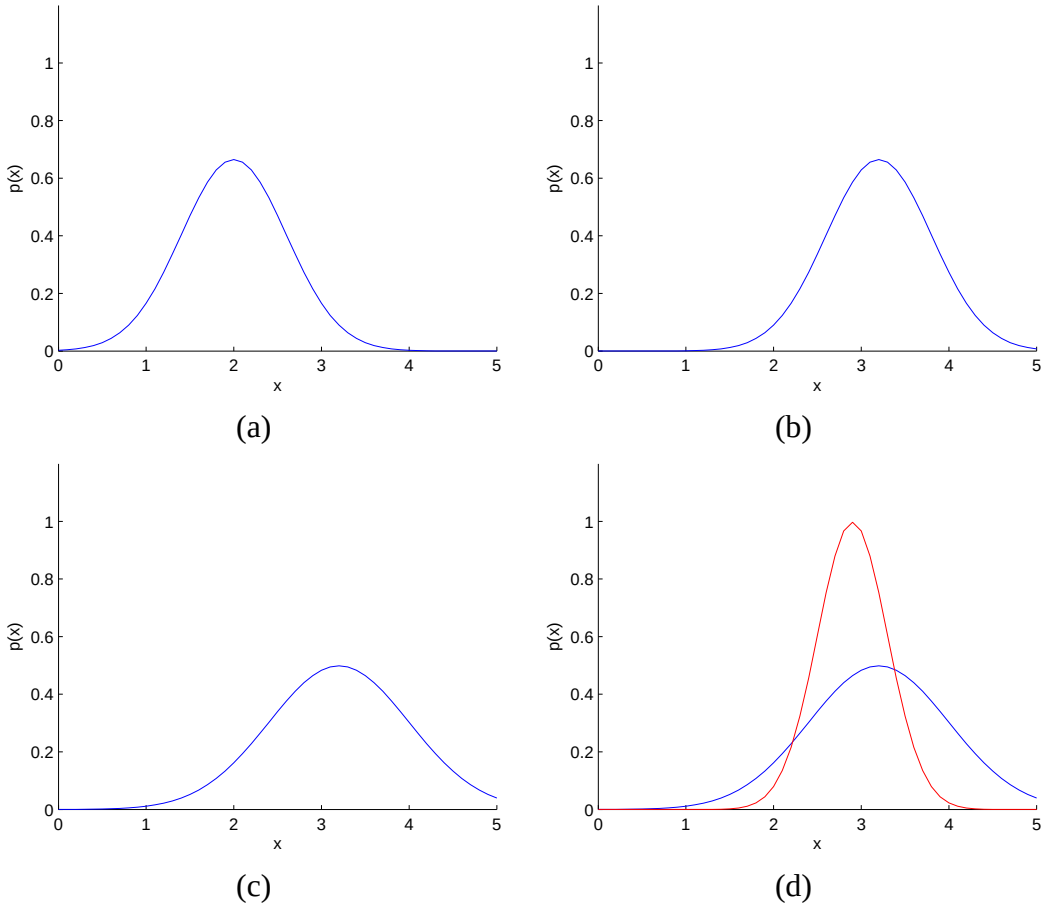


Figure 2.1: Kalman filtering: (a) Estimated posterior at time $t-1$; (b) Predicted distribution at time t; (c) Diffused predictive distribution; (d) Diffused predictive distribution with likelihood distribution shown in red. Assimilation of the predition with current observations via the Kalman gain matrix gives the posterior at time t in preparation for the next iteration.

future observations. Although some approaches dealt with this explicitly [25], by far the most popular was the generic CONDENSATION algorithm of Isard and Blake [57] (introduced earlier for radar systems by Gordon as the "particle filter" [42]).

Originally developed for contour tracking, CONDENSATION (a form of sequential Monte Carlo sampling [33]) represents a non-parametric probability distribution with a set of "particles", each representing a state estimate and weighted with respect to the likelihood. At each step, the weighted particle set (a sum of delta functions) is prop-

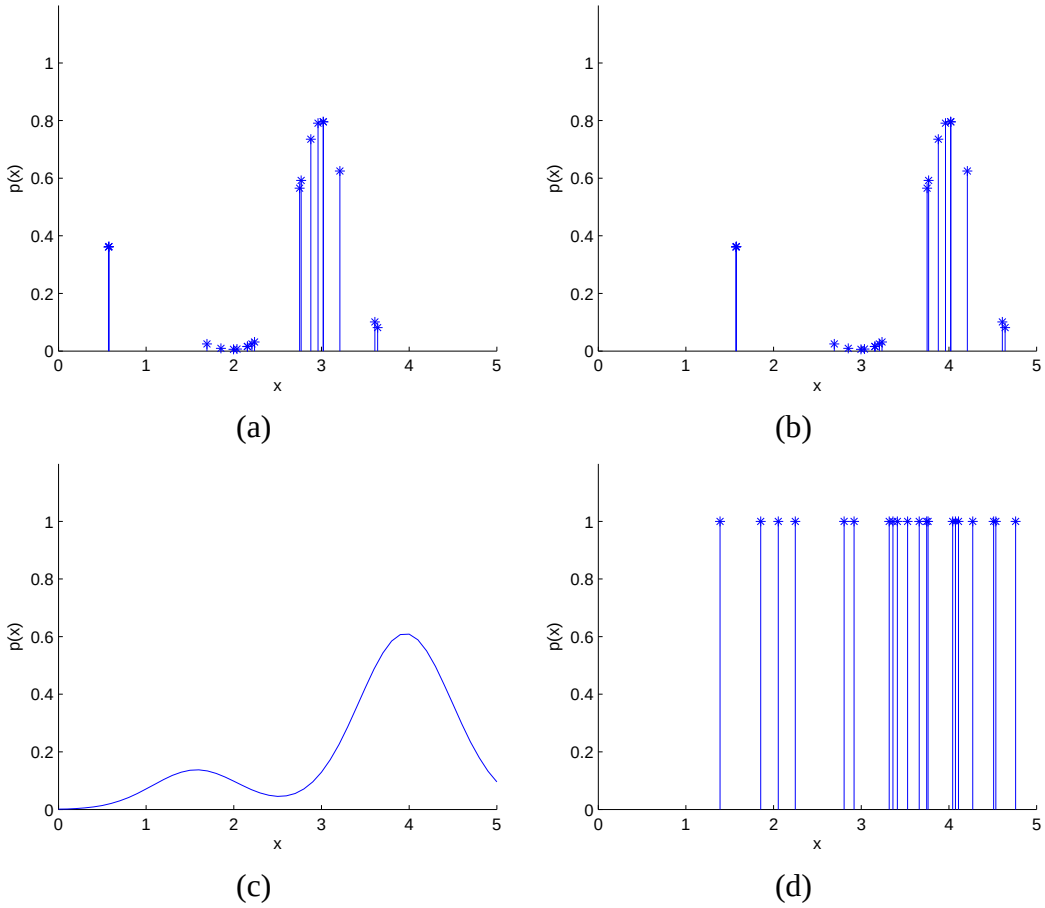


Figure 2.2: Particle filtering: (a) Weighted samples representing the posterior at time $t-1$; (b) Particles following propagation via the motion model; (c) Diffused particles giving a continuous distribution from which we can sample; (d) Samples drawn from mixture of Gaussians. The resulting particles are then weighted to give a particle set representing the posterior at time t in preparation for the next iteration. Note that particles are shown un-normalized for illustrative purposes only.

agated to the next time instant via the deterministic component of the state evolution model, $p(x_t|x_{t-1})$. The propagated particles are then diffused with stochastic noise to give a continuous density estimate (typically a mixture of Gaussians) that is resampled to generate new (unweighted) predictions. These predictions are then weighted via the likelihood, $p(D_t|x_t)$, with respect to the new observations to form a new weighted particle set. Iteration of this process propagates the multimodal posterior through time (see Figure 2.2).

Deutscher *et al*. [31] demonstrated the advantages of CONDENSATION for human motion by tracking an arm through singularities and discontinuities where the Kalman filter suffered from terminal failure. However, CONDENSATION was originally developed for relatively low ($\sim$6) dimensional state spaces whereas full body pose commonly lies within state spaces of high ($\sim$30) dimension. Due to the exponential explosion in the required number of particles with increasing dimension (known as the "curse of dimensionality") methods were developed to concentrate particles in small regions of high probability, reducing the total number needed for effective tracking.

An approach specific to kinematic trees known as *partitioned sampling* [68] (or *state space decomposition* [38]) exploited the conditional independence of different branches of the tree by working from the root (*i.e*. torso) outwards, thus constraining the locations of the leaves independently. In practice, however, it proved very difficult to localize the human torso independently of the limbs. An implicit form of partitioning was later demonstrated using the 'crossover' operator from genetic algorithms [30].

Sidenbladh *et al*. [93] used a learned walking model to enforce a strong dynamic prior and capture correlations between pose parameters. Deutscher *et al*. [29] implemented *annealing* in order to smooth the likelihood function and introduce sharp maxima gradually, thus avoiding premature "trapping" of particles. Other approaches used deterministic optimization techniques to recover distinct modes in the cost surface such that it could be represented in a parametric form [25, 99].

In particular, Sminchisescu and Triggs [99] introduced *covariance-scaled sampling* whereby samples are diffused in the directions of highest covariance to deal with kinematic singularities. To explore local maxima close to the current estimate, they employed sampling and optimization methods developed for computational chem-

istry [100, 101]. They later investigated local maxima far from the current estimate due to monocular ambiguities ("kinematic flips") that could be determined from straightforward geometry [102]. These studies of the cost surface clearly demonstrated how abundant local maxima are in monocular body tracking.

Despite these developments, however, accurate model-based tracking of general human motion remained elusive. Furthermore, hand initialization is required and designing a smooth observation model takes considerable effort. As a result, model-based tracking for human motion capture suffered a decline in favour of more data-driven approaches as described in Section 2.1.2.

Observation (likelihood) and motion (prior) models

We digress for a moment to discuss the observation (likelihood) and predictive motion (prior) distributions. Their product gives the posterior distribution representing our 'best' estimate of the state based on what we see (observations) and what we expected to see (prior). Effectively, the motion prior imposes smoothness on the state over time, maintaining a delicate balance between "truth" and "beauty".[1]

With respect to the observation model, various image features are available (see Figure 2.3) such as the occluding contour (silhouette) [28, 29], optic flow [21, 60, 122, 93, 99] and edges, as derived from rapid changes in intensity [29, 122, 38, 99] or texture [90]. Having projected the model into the image, observations are compared with what we expected. To define more clearly "what we expect to see", Sidenbladh and Black learn spatial statistics of edges and ridges in images of humans [95], rather than assume a known distribution. Note that it is common to combine different visual cues to overcome characteristic failings of particular features such as edges (sparse but

[1] A rather bohemian exposition provided by Dr. Andrew Fitzgibbon.

(a) (b) (c)

Figure 2.3: (a) Example frame from a starjumps sequence; (b) Occluding contour (silhouette); (c) Distance transform of the masked edge map.

well localized) and optic flow (dense but ill-defined in regions of uniform texture and prone to drift).

The predictive motion model, $p(x_t|x_{t-1})$ simply tells us, given a pose at time $t-1$, what we expect it to be at time t and with what certainty. The most common model for general motion is the constant velocity model whereby the velocity at time $t-1$ is used to predict the pose at time t. This common model is easily incorporated into the Kalman filter, EKF and particle filter for human body tracking [60, 61, 29, 99, 122, 93] although higher order models (*e.g.* constant acceleration [38]) have also been employed.

Although the constant velocity/position/acceleration model is simple to implement, it is seldom accurate enough to allow tracking over long sequences. One way to address this problem is to use more specialized (possibly non-linear) motion models learned from training data. As an extreme example, Rohr [86] reduces the state space to a single dimension representing the phase of a walk cycle. Sidenbladh *et al*. [93] compute a statistical model (via Principal Component Analysis) of various walk cycles to

account for variation in gait, whilst maintaining a low dimensional (5D) state space. Alternatively, the predicted pose can be obtained from stored pose sequences by simple database look-up [51] or probabilistic sampling [94]. One problem with such specific approaches is that they rarely generalize well to novel motions.

Another alternative is to use several motion models and switch between them depending on the current estimated action [124, 79, 3]. Since each model has different parameters, they are more specialized and can predict the future pose with greater accuracy. However, the task of determining the most appropriate model is not trivial and is often implemented by a Hidden Markov Model (HMM), with transitions between models learned from training data.

Finally, the predictive model may incorporate hard constraints to rule out unlikely poses. The most common of these are anatomical joint limits (usually enforced as limits on Euler angles [29, 99]) but may also be learned from training data in order to model dependencies between degrees of freedom [49]. Further constraints can be enforced to prevent the self-intersection of limbs [99].

2.1.2 Tracking people from the bottom up

Whereas model-based tracking approaches fit a parametric model to observations using a likelihood function, data-driven methods attempt to recover pose parameters directly from the observations. Methods that estimate $p(x_t|D_t, D_{t-1}, \ldots)$ directly from training data, also known as *discriminative* model approaches, vary much more than model-based tracking and are often more applicable to monocular tracking.

Early approaches [65, 46, 131] heuristically assigned sections of the occluding contour to various body parts before estimating joint locations and pose. Later methods used shape context matching [73], geometric hashing [105] and optic flow [36] of the

input image to find its nearest neighbour in a large database of stored examples. The stored joint locations were then transferred by warping the corresponding examplar to the presented input. Due to the exponentially high number of examples required for general motion, efficient searching methods have also been developed for nearest neighbour retrieval [91, 43].

Another popular approach is to detect parts independently and assemble them into a human body. Early approaches classified coloured "blobs" as head, hands, legs *etc*. to interpret gross movements [19, 125]. More recently, body parts located with primitive classifiers (*e.g*. "ribbon" detectors) have been assembled using dynamic programming [37], sampling [54] and spatiotemporal constraint propagation [83]. Two-stage methods have also been employed where body parts are detected with one classifier and assembled with another, such as a Support Vector Machine (SVM), in a "combination of classifiers" framework [72, 87].

For the multi-view 3D case, similar methods have recently been applied by Sigal *et al*. [96] using Belief Propagation (BP) to assemble body parts in time and space. Grauman *et al*. [45] use a mixture of probabilistic principal component analysers to learn the joint manifold of observations and pose parameters such that projection of the input silhouettes onto the manifold recovers the estimated 3D pose. With multiple cameras, volumetric methods such as voxel occupancy [103] and visual hull reconstruction [26, 44] are also possible. However, the number of cameras required to accurately recover structure (and pose) is high.

Other approaches ignore the fact that they are tracking a kinematic model and directly model a functional relationship[2] between inputs (observations) and outputs (pose

[2]Strictly speaking, the relationship is a many-to-many *mapping* rather than a function

parameters) using a corpus of training data. Once the mapping has been learned, the training data can be discarded for efficient on-line processing. Brand [16] uses entropy minimization to learn the most parsimonious explanation of a silhouette sequence while Agarwal and Triggs [2] use a Relevance Vector Machine (RVM) to obtain 3D pose directly from a single silhouette. Rosales and Sclaroff [88] cluster examples in pose space and learn a different function for each cluster using neural networks. Their "Specialized Mappings Architecture" (SMA) recovers a different solution for each cluster to accommodate the ambiguities inherent in monocular pose recovery, albeit in a less principled manner than the more recent "mixtures of regressors" [4, 98].

2.1.3 Importance sampling

So far we have discussed two seemingly opposite paradigms – model-based tracking and data-driven approaches – each with their own strengths and weaknesses. In particular, model-based tracking requires hand initialization and does not take the most recent measurements into account until *after* future state estimates have been predicted. The effect of this latter point is that we risk wasting particles in regions of low probability density if we have a poor motion model. However, it is more difficult to incorporate prior knowledge (*e.g.* motion models, kinematic constraints) into data-driven approaches.

Importance sampling combines the strengths of both paradigms and is easily incorporated into the particle filter framework [58]. It is employed when the posterior (that can be evaluated at a given point but not sampled from) can be approximated by a *proposal distribution*, $q(x_t|D_t)$, that is cheap to compute from the most recent observations and *can* be both evaluated point-wise and sampled. Rather than sampling from the prior, samples are drawn from the proposal distribution and multiplied by a

reweighting factor, w, where:

$$w = \frac{p(x_t|D_{t-1}, D_{t-2}, \ldots)}{q(x_t|D_t)} \tag{2.5}$$

such that the samples are correctly weighted with respect to the motion model before reweighting again with respect to the likelihood. However, these samples are now concentrated in regions of high *posterior* (rather than prior) probability mass and should therefore be more robust to 'unpredictable' motions that are incorrectly modelled by the dynamical motion model. Note that, if $q(x_t|D_t) = p(x_t|D_{t-1}, D_{t-2}, \ldots)$ then all weights are equal, resulting in the standard particle filter.

Since the proposal distribution is generated from current observations, it is used both for initialization and guided sampling such that particles are selected based on the *most recent* observations and then takes into account the predicted state using the motion model. In the original hand-tracking application [58], skin-colour detection was used to generate a proposal distribution before evaluating the more computationally expensive likelihood, resulting in a significant speed-up during execution.

Importance sampling was later applied to single-frame human pose estimation in [64, 106] by locating image positions of the head and hands using a face detector [121] and skin colour classification, respectively. From this, they were able to produce 2D proposal distributions for the image locations of intermediate joints. An initial hypotheses was drawn from these distributions and inverse kinematics applied to give a plausible 3D pose. The space of 3D poses could then be explored using Markov Chain Monte Carlo (MCMC) sampling techniques [64] to give plausible estimates of human pose that were then compared with measurements using an observation model.

2.2 Structure From Motion

This thesis also draws strongly upon the field of Structure From Motion (SFM), following early studies by Ullman [117] to investigate human perception of 3D objects. Ullman demonstrated that the relative motion between 2D point features in an image gives the perception of a three dimensional object, as exemplified using features from the surfaces of two co-axial cylinders rotating in different directions.

2.2.1 Rank constraints and the Factorization Method

Although Structure from Motion was an active research field in the 1980s and early 1990s, approaches typically employed perspective cameras [67] (possibly undergoing a known motion [15]) and recovered structure or motion from optical flow [1, 10] or minimal 'n-point' solutions [53].

In contrast, other approaches [53, 62] employed affine projection models. This culminated in the ground-breaking paper of Tomasi and Kanade [111], resulting in a paradigm shift within the field. Specifically, they noted that under an *affine* camera model (a sensible approximation in many cases) the projection of features that are moving with respect to the camera is *linear*. As a result, all features and all frames can be considered simultaneously by defining a matrix of feature tracks (trajectories):

$$\mathbf{W} = \begin{bmatrix} \mathbf{x}_1^1 & \cdots & \mathbf{x}_N^1 \\ \vdots & & \vdots \\ \mathbf{x}_1^V & \cdots & \mathbf{x}_N^V \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ \vdots & \vdots \\ \mathbf{R}_V & \mathbf{t}_V \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_N \\ 1 & \cdots & 1 \end{bmatrix} = \mathbf{P}_{(2V\times 4)}\mathbf{X}_{(4\times N)} \qquad (2.6)$$

where $\mathbf{x}_n^v$ is the 2×1 position vector of feature n in view v, $\mathbf{R}_v$ is the first two rows of the vth camera orientation matrix, $\mathbf{t}_v = \frac{1}{N}\sum_n \mathbf{x}_n^v$ is the projected centroid of the features in frame v and $\mathbf{X}_n$ is the 3×1 position vector of feature n with respect

to the objects local co-ordinate frame. This critical observation demonstrated that $rank(\mathbf{W}) \leq 4$ such that $\mathbf{W}$ can be factorized into $\mathbf{P}$ and $\mathbf{X}$ using the Singular Value Decomposition (SVD) to retain only the data associated with the four largest singular values. Normalizing the data with respect to the centroid results in the $rank(\widetilde{\mathbf{W}}) \leq 3$ system:

$$\widetilde{\mathbf{W}} = \begin{bmatrix} \mathbf{x}_1^1 - \mathbf{t}_1 & \cdots & \mathbf{x}_N^1 - \mathbf{t}_1 \\ \vdots & & \vdots \\ \mathbf{x}_1^V - \mathbf{t}_V & \cdots & \mathbf{x}_N^V - \mathbf{t}_V \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_V \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_N \end{bmatrix} = \mathbf{P}_{(2V \times 3)} \mathbf{X}_{(3 \times N)} \quad (2.7)$$

where the structure's centroid is now located at the global origin.

Since these two factors can be interpreted as structure and motion in an affine co-ordinate frame, it is necessary to "upgrade" them to a Euclidean co-ordinate frame before meaningful lengths and angles can be recovered. This can be seen by the fact that post-multiplication (pre-multiplication) of the motion (structure) by a matrix $\mathbf{B}$ ($\mathbf{B}^{-1}$) leaves the resulting $\mathbf{W}$ unaltered (known as a *gauge freedom*):

$$\mathbf{PX} = \mathbf{PBB}^{-1}\mathbf{X}. \quad (2.8)$$

It can be shown that the 3×3 calibrating transformation, $\mathbf{B}$, can be expressed in upper-triangular form:

$$\mathbf{B} = \begin{bmatrix} a & b & c \\ & d & e \\ & & 1 \end{bmatrix} \quad (2.9)$$

whose lower-right element is fixed at unity to avoid any depth-scale ambiguity.

The value of $\mathbf{B}$ is computed by making sensible assumptions (*e.g.* zero skew, unit aspect ratio) about the camera to impose constraints on the rows of $\mathbf{PB}$. Specifically,

every $\mathbf{R}_v\mathbf{B}$ block corresponding to a given frame should be close to the first two rows of a scaled rotation matrix [82]. Defining $\mathbf{R}_v$ as:

$$\mathbf{R}_v = \begin{bmatrix} \mathbf{i}^T \\ \mathbf{j}^T \end{bmatrix}, \tag{2.10}$$

the constraints of unit aspect ratio and zero skew are expressed algebraically as:

$$\mathbf{i}^T\mathbf{B}\mathbf{B}^T\mathbf{i} - \mathbf{j}^T\mathbf{B}\mathbf{B}^T\mathbf{j} = 0, \tag{2.11}$$

$$\mathbf{i}^T\mathbf{B}\mathbf{B}^T\mathbf{j} = 0. \tag{2.12}$$

These constraints are linear in the elements of the matrix $\mathbf{\Omega} = \mathbf{B}\mathbf{B}^T$, that is recovered by linear least squares. Cholesky decomposition of $\mathbf{\Omega}$ should then give the required value of $\mathbf{B}$ as required.

2.2.2 Extensions to the Factorization Method

The Factorization Method's simplicity and robustness to noise (it recovers the Maximum Likelihood solution in the presence of isotropic Gaussian noise [84]) has ensured that it remains popular to this day. Extensions to the method incorporated new camera models [80], used multiple bodies [27], recast the batch process as a sequential update [74], and generalized for other measurements such as lines and planes [75]. Further developments used the spatial statistics of the image features to account for non-isotropic noise [75, 56] while similar principles were also shown to hold for optical flow estimation [55].

Statistical shape models were later developed to deal with deformable objects, treating the structure at each instant as a sample drawn from a Gaussian distribution in

shape space [20, 113, 17, 18]. In this way, non-rigid shapes such as faces can be captured and reconstructed.

In the context of human pose estimation, the factorization method has seen little use due to the lack of salient features on the human body. One approach uses joint locations in a pair of sequences and the factorization method applied independently at each time instant [66]. With only two views at each time instant, projection constraints alone are insufficient to recover metric structure and motion so prior knowledge of the structure (in this case, the human body) is employed to further constrain the solution. This calibration method is discussed in greater detail in Chapter 6.

In related work [107, 11] the affine camera assumption is employed in single view pose reconstruction (although factorization is not used). In these cases, it is assumed that the *ratios* of body segments are known in order to place a lower bound on the scale factor in the projection.

To begin the thesis, we return to the multibody factorization case with particular focus on articulated objects.

Chapter 3

Recovering 3D Joint Locations I : Structure From Motion

In this chapter, we present a method for recovering centres and axes of rotation between a pair of objects that are articulated. The method is an extension of the popular Factorization method for Structure From Motion and therefore is applicable to sequences of unknown structure from a single camera. In particular, we show that articulated objects have dependent motions such that their motion subspaces have a known intersection that results in a tighter lower bound on $rank(\mathbf{W})$*. We consider pairs of objects coupled by prismatic, universal and hinge joints, focussing on the latter two since they are present in the human body. Furthermore, we discuss the self-calibration of articulated objects and present results for synthetic and real sequences.*

3.1 Introduction

In this chapter we develop Tomasi and Kanade's Factorization Method [111], originally applied to static scenes, for dynamic scenes containing a pair of objects moving relative to each other in a constrained way. In this case, we say that their motions are *dependent*. In contrast, objects that move relative to each other in an unconstrained way are said to have *independent* motions.[1]

As in the original formulation, we assume that perspective effects are small and employ an affine projection model. Under this assumption, we recover structure and motion directly using the Singular Value Decomposition (SVD) of a matrix, $\mathbf{W}$, of

[1] Portions of this chapter were published in [116]

image features over the sequence. Specifically, with affine projection it was shown that $rank(\mathbf{W}) \leq 4$ for a static scene. Intuitively, $rank(\mathbf{W}) \leq 4k$ with k objects in the scene. However, we demonstrate that if the objects' motions are *dependent* then the reduced degrees of freedom result in a tighter upper bound such that $rank(\mathbf{W}) < 4k$.

In particular, we investigate exactly how dependent motions impose this tighter bound and how underlying parameters of the system can be recovered from image measurements. We investigate three cases of interest:

- **Universal joint**: Two objects coupled by a two or three degree of freedom joint such that there is a single *centre of rotation* (CoR).
- **Hinge joint**: Two objects coupled by a one degree of freedom joint such that there is an *axis of rotation* (AoR). The system state at any time is parameterized by the angle of rotation about this axis of one object with respect to the other.
- **Prismatic joint**: Two objects coupled by a one degree of freedom "slide" such that there is an *axis of translation*. The system state at any time is parameterized by the displacement along this axis from a reference point.

Of these three cases, we investigate universal joints and hinges more closely since they are found in the human body whereas prismatic joints are included for completeness. These cases of interest are selected from a large number of potential dependencies as discussed in Section 3.2.

3.1.1 Related work

Costeira and Kanade [27] extended The Factorization Method for dynamic scenes as a motion segmentation algorithm. However, the method assumed that the motions were

independent. It was later shown that when the relative motion of the objects is *dependent*, the motion subspaces have a non-trivial intersection [128]. As a result, algorithms assuming that the motion subspaces are orthogonal suffered terminal failure.

In other work, factorization was used to recover structure and motion of deformable objects represented as a linear combination of "basis shapes" [17, 20, 113]. This is a reasonable assumption for *small* changes in shape (*e.g*. muscular deformation) although more pronounced deformations (*e.g*. large articulations at a joint) violate this assumption.

Aside from human motion tracking (see Section 2.1) and model-based tracking systems [34], articulated objects have been largely neglected in the tracking literature. At the time of this research taking place, the only directly related work was that of Sinclair *et al*. [97] who recovered articulated structure and motion using perspective cameras. However, they assumed that articulation was about a hinge and that the axis of rotation was approximately vertical in the image. Furthermore, non-linear minimization was used to find points on the axis and they assumed that some planar structure was visible.

In contrast, we exploit an affine projection model since the two objects are coupled such that their relative depth is small compared to their distance from the camera. As a result, our method is much simpler since (for the most part) we use computationally cheap linear methods rather than expensive search and iterative optimization techniques. Furthermore, we do not assume to know how the objects are coupled, nor do we require the axis of rotation to be visible in the image, nor any structure (visible or otherwise) to be planar. In fact, we show that the nature of the dependency between the objects is readily available from the image information itself. Although we use a fixed camera in this work, this is not a requirement and the method is equally applicable to

a camera moving within the scene.

We note that Yan and Pollefeys [126] published an almost identical method developed independently of this work. As a result, our works can be considered complementary since we verify each other's (repeatable) results. However, we also consider calibration of the cameras and how this process is affected by the additional constraints that should be imposed.

We also note that this method is in contrast to other methods that deal with articulated structure [66, 107, 115] where only one point (typically a joint centre) per segment is included in the data. In such cases, there is no redundancy to be exploited in the point feature data (since four points per segment are required to define a coordinate frame in 3D) and rank constraints over the whole sequence do not apply.

3.1.2 Contributions

The contributions of this chapter can be summarised as follows:

- We demonstrate that dependent motions impose stronger rank constraints on a matrix of image features. Furthermore, we show that the nature of the dependency can be recovered from the measurements themselves in order to select appropriate constraints for future operations.

- We impose the selected constraints *during* factorization and self-calibration (rather than as a post-processing step) in order to recover metric structure and motion that is consistent with the underlying scene. We also show that under some circumstances, self-calibration becomes a *non-linear* problem that requires more complex computation.

- We present results on both real and synthetic data for a qualitative and quantitative analysis. Our results show that, despite its simplicity, the method is accurate and captures the scene structure correctly.

3.2 Multibody Factorization

Relative motion between two objects can be dependent in either translation or rotation (or both), as summarized in Table 3.1.

		DOF$_{rot}$		
		0	1	2/3
DOF$_{trans}$	0	Same object	**Hinge joint**	**Universal joint**
	1	**Linear track**	Cylinder on a plane	Sphere in tube?
	2	Draftsman's board	Computer mouse	Ball on a plane
	3	Cartesian robot	SCARA end effector	Independent objects

Table 3.1: Possible motion dependencies between two objects.

For two bodies moving independently, the 'motion space' scales accordingly such that $rank(\mathbf{W}) = 8$. However, when the motions are *dependent* there is a further decrease in $rank(\mathbf{W})$ that we use both to detect articulated motion and to estimate the parameters of the joint. For the remainder of this chapter, quantities associated with the second object are primed (*e.g.* $\mathbf{R}'$, $\mathbf{t}'$, etc).

3.2.1 Universal joint: DOF$_{rot} = 2, 3$

When two objects are coupled by a universal[2] joint, the bodies cannot translate with respect to each other but their relative orientation is unconstrained. Universal joints are commonly found in the form of ball-and-socket joints (*e.g.* on a camera tripod, shoulders, hips).

[2]In this definition, we include joints with two degrees of freedom as well as those with three.

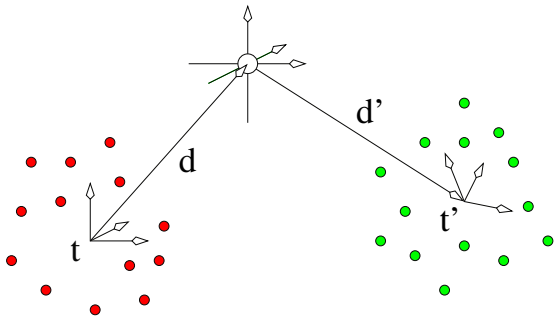


Figure 3.1: Schematic of a universal joint.

The universal joint is illustrated schematically in Figure 3.1, where $\mathbf{t}$ and $\mathbf{t}'$ represent the centroids of the objects. The position of the CoR in the co-ordinate frame of each object is denoted by $\mathbf{d} = [u, v, w]^T$ and $-\mathbf{d}' = [u', v', w']^T$, respectively. For accurate structure and motion recovery, the location of the CoR must be consistent (in a global sense) in the co-ordinate frames of the two objects such that:

$$\mathbf{t} + \mathbf{R}\mathbf{d} = \mathbf{t}' - \mathbf{R}'\mathbf{d}'. \tag{3.1}$$

Alternatively, we can say that $\mathbf{t}'$ is completely determined once $\mathbf{d}$ and $\mathbf{d}'$ are known since:

$$\mathbf{t}' = \mathbf{t} + \mathbf{R}\mathbf{d} + \mathbf{R}'\mathbf{d}'. \tag{3.2}$$

Rearranging (3.1) or (3.2) gives:

$$\mathbf{R}\mathbf{d} + \mathbf{R}'\mathbf{d}' - (\mathbf{t}' - \mathbf{t}) = \mathbf{0}, \tag{3.3}$$

showing that $[\mathbf{d}^T, \mathbf{d}'^T, -1]^T$ lies in the right (column) nullspace of $[\mathbf{R}, \mathbf{R}', \mathbf{t}' - \mathbf{t}]$. Not only does this show that $rank(\mathbf{W}) \leq 7$ but also that $\mathbf{d}$ and $\mathbf{d}'$ can be recovered once $\mathbf{R}$, $\mathbf{R}'$, $\mathbf{t}$ and $\mathbf{t}'$ are known. Since $\mathbf{t}$ and $\mathbf{t}'$ are the 2D centroids of the two point clouds, they are simply the row means of the matrix of feature tracks for the first and second

object, respectively. Following [111] we translate each object to the origin, giving the 'normalized' $rank = 6$ system:

$$\widetilde{\mathbf{W}} = \begin{bmatrix} \mathbf{R} & \mathbf{R}' \end{bmatrix} \begin{bmatrix} \mathbf{S} & \\ & \mathbf{S}' \end{bmatrix}. \tag{3.4}$$

This is effectively "full rank" since the rotations are independent and have been decoupled from the translations (where the dependency resides). From (3.4), we can recover $\mathbf{R}$ and $\mathbf{R}'$ by factorization using the SVD. In practice, however, taking the SVD of $\widetilde{\mathbf{W}}$ recovers a full structure matrix, $[\mathbf{V}, \mathbf{V}']$, rather than the block diagonal form seen in (3.4). We therefore separate the objects by premultiplying $[\mathbf{V}, \mathbf{V}']$ with a matrix, $\mathbf{A}_U$:

$$\mathbf{A}_U[\mathbf{V}, \mathbf{V}'] = \begin{bmatrix} N_L(\mathbf{V}') \\ N_L(\mathbf{V}) \end{bmatrix} [\mathbf{V}, \mathbf{V}'] \tag{3.5}$$

$$= \begin{bmatrix} N_L(\mathbf{V}')\mathbf{V} & N_L(\mathbf{V}')\mathbf{V}' \\ N_L(\mathbf{V})\mathbf{V} & N_L(\mathbf{V})\mathbf{V}' \end{bmatrix} \tag{3.6}$$

$$= \begin{bmatrix} N_L(\mathbf{V}')\mathbf{V} & \mathbf{0} \\ \mathbf{0} & N_L(\mathbf{V})\mathbf{V}' \end{bmatrix} \tag{3.7}$$

where $N_L(\cdot)$ is an operator that returns the left (row) nullspace of its matrix argument. Finally, we transform the recovered motion matrix, $[\mathbf{U}, \mathbf{U}']$, accordingly: $[\mathbf{U}, \mathbf{U}']\mathbf{A}_U^{-1} \rightarrow [\mathbf{R}, \mathbf{R}']$. Having recovered $\mathbf{R}$, $\mathbf{R}'$, $\mathbf{t}$ and $\mathbf{t}'$ we can now compute $\mathbf{d}$ and $\mathbf{d}'$. The reprojected joint centre is then simply $\mathbf{t} + \mathbf{R}\mathbf{d}$ (or $\mathbf{t}' - \mathbf{R}'\mathbf{d}'$).

Although in this case we could recover $\mathbf{R}$ and $\mathbf{R}'$ by factorization of each object independently, here we use a method that deals with both objects simultaneously for consistency with the hinge case where independent factorization is not so straightforward.

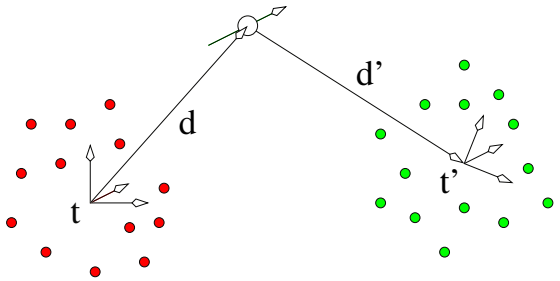


Figure 3.2: Schematic of a hinge joint.

3.2.2 Hinge joint: DOF$_{rot} = 1$

We now investigate two bodies coupled by a hinge joint. As with the universal joint, translation is not permitted between the two objects. However, unlike the universal joint a hinge permits rotation about an axis that is fixed in the co-ordinate frame of each object (see Figure 3.2). Like the universal joint, hinges are also found in the human body (*e.g.* knees, elbows) and are also common in man-made environments (*e.g.* doors, wheels).

In this case, *all* points on the rotation axis satisfy both motions such that the subspaces have a 2D intersection and $rank(\mathbf{W}) \leq 6$. Aligning the rotation axis with the x-axis by chosing an appropriate global co-ordinate frame, we denote the motion matrices by $\mathbf{R} = [\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3]$ and $\mathbf{R}' = [\mathbf{c}_1, \mathbf{c}'_2, \mathbf{c}'_3]$ to give the 'normalized' system:

$$\widetilde{\mathbf{W}} = [\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{c}_3\ \mathbf{c}'_2\ \mathbf{c}'_3] \begin{bmatrix} X_1 \cdots X_{n_1} & X'_1 \cdots X'_{n_2} \\ Y_1 \cdots Y_{n_1} & \\ Z_1 \cdots Z_{n_1} & \\ & Y'_1 \cdots Y'_{n_2} \\ & Z'_1 \cdots Z'_{n_2} \end{bmatrix}. \tag{3.8}$$

Due to the dependency in rotation, factorizing the objects independently requires constraints to be applied *after* factorization and is not straightforward. In contrast, using the form in (3.8) ensures that both objects have the same x-axis and respect the

"common axis" constraint such that rotations are *not* independent. To zero out entries of the recovered $[\mathbf{V}, \mathbf{V}']$ we premultiply with a matrix, $\mathbf{A}_H$:

$$\mathbf{A}_H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ & N_L(\mathbf{V}') & & & \\ & N_L(\mathbf{V}) & & & \end{bmatrix} \tag{3.9}$$

and transform $[\mathbf{U}, \mathbf{U}']$ accordingly.

Note that the 'joint centre' may lie anywhere on the axis of rotation, provided that $u + u' = k$ where k is the distance between object centroids parallel to the rotation axis. As a result, we can show that $[u + u', v, w, v'w', -1]^T$ lies in the nullspace of $[\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}'_2, \mathbf{c}'_3, \mathbf{t}' - \mathbf{t}]$ and can be recovered with ease. The reprojected axis of rotation is then given by the line:

$$l(\alpha) = \mathbf{t} + [\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3][\alpha, v, w]^T \tag{3.10}$$

where α is any real number.

3.2.3 Prismatic joint: DOF$_{rot} = 0$

Since we are less concerned with prismatic joints (they are of little relevance to human motion tracking), we only provide a brief note about their factorization. In fact, normalization of the sets of feature tracks effectively removes any relative translation between the two objects such that they become indistinguishable from a single, normalized object. As a result, $rank(\widetilde{\mathbf{W}}) \leq 3$, detection of a prismatic joint is relatively straightforward and the two objects can be recovered simultaneously using the original Factorization method.

3.3 Multibody calibration

Although we have shown how to recover *affine* structure and motion that is consistent with the underlying scene structure, we are primarily interested in recovering meaningful distances and angles. This requires the 'upgrading' to a Euclidean co-ordinate frame via self-calibration (see 2.2.1). In this section, we investigate how constraints imposed by articulated structures affect the self-calibration process and how we may exploit this fact to recover metric structure and motion that is consistent with the underlying scene.

3.3.1 Universal joint

For two objects coupled by a universal joint, a gauge freedom exists since:

$$\widetilde{\mathbf{W}} = \begin{bmatrix}\mathbf{R} & \mathbf{R}'\end{bmatrix} \cdot (\mathbf{B}\mathbf{B}^{-1}) \cdot \begin{bmatrix}\mathbf{S} & \\ & \mathbf{S}'\end{bmatrix} \tag{3.11}$$

where the calibrating matrix, $\mathbf{B}$, takes the form of a 6×6 upper triangular matrix:

$$\mathbf{B} = \begin{bmatrix} a & b & c & & & \\ & d & e & & & \\ & & f & & & \\ & & & a' & b' & c' \\ & & & & d' & e' \\ & & & & & 1 \end{bmatrix}. \tag{3.12}$$

The upper-right 3×3 block must be zero in order to prevent mixing of $\mathbf{R}$ with $\mathbf{R}'$ (or $\mathbf{S}$ with $\mathbf{S}'$). Including f in the parameters to be determined allows us to constrain the scaling induced by the projections $\mathbf{R}$ and $\mathbf{R}'$ to be equal at any given time. This is a sensible restriction since the two bodies are attached to each other and therefore at approximately the same depth with respect to the camera at all times (such that any scaling induced by perspective affects both objects equally).

In contrast, two objects that are independent may have different depths with respect to the camera at different times (*e.g.* when one moves towards the camera and the other away from it). In such cases, the scaling over time that is induced by perspective cannot be assumed to be equal for both $\mathbf{R}$ and $\mathbf{R}'$. As a result, unless projection is known to be truly orthographic, f must be constrained to unity and the method becomes equivalent to calibrating both objects independently.

As in the single object case, the constraints are linear in the elements of $\mathbf{B}\mathbf{B}^{-1}$ such that a solution for $\mathbf{B}$ can be found using the SVD followed by Cholesky decomposition.

3.3.2 Hinge joint

For two objects joined by a hinge, the gauge freedom can be expressed as:

$$\widetilde{\mathbf{W}} = [\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{c}_3\ \mathbf{c}'_2\ \mathbf{c}'_3] \cdot (\mathbf{B}\mathbf{B}^{-1}) \cdot \begin{bmatrix} X_1 \cdots X_{n_1} & X'_1 \cdots X'_{n_2} \\ Y_1 \cdots Y_{n_1} & \\ Z_1 \cdots Z_{n_1} & \\ & Y'_1 \cdots Y'_{n_2} \\ & Z'_1 \cdots Z'_{n_2} \end{bmatrix} \tag{3.13}$$

where the motions share a common axis such that $\mathbf{B}$ takes the form:

$$\mathbf{B} = \begin{bmatrix} a & b & c & b' & c' \\ & d & e & & \\ & & f & & \\ & & & d' & e' \\ & & & & 1 \end{bmatrix}. \tag{3.14}$$

In contrast to the single object and universal joint cases, it can be shown that the constraints are no longer linear in the elements of $\mathbf{B}\mathbf{B}^{-1}$. Therefore, as a first approximation, we perform self-calibration on the motion matrix $[\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_1, \mathbf{c}'_2, \mathbf{c}'_3]$ using a calibration matrix of the form given in (3.12). We then rescale the upper-left 3×3 submatrix such that $a = a'$. and rearrange the elements to give the form shown in (3.14). Since this is only an approximate calibration, we use this as an initial value

in a non-linear optimization to compute a locally optimal solution.

3.3.3 Prismatic joint

Since the rotation matrices are equal for both objects, the single-body calibration method is applicable in this case.

3.4 Estimating system parameters

We now briefly outline how the system parameters of interest (*i.e*. lengths and angles) are recovered from the structure and motion that we have computed.

3.4.1 Lengths

Recovering lengths is particularly simple in this framework. For a universal joint, premultiplying $[\mathbf{d}^T, \mathbf{d'}^T]^T$ by the 6×6 calibration matrix, $\mathbf{B}^{-1}$ gives the equivalent link vectors in a Euclidean space. Similarly for a hinge joint, premultiplying $[\alpha, v, w, v'w']^T$ by the corresponding 5×5 calibration matrix gives the location of a point (parameterized by α) on the axis in Euclidean space. Note, however, that the definition of 'link length' for a hinge joint is somewhat arbitrary.

3.4.2 Angles

For two bodies joined at a hinge, we choose the x-axis as the axis of rotation such that (with a slight abuse of notation) at a given frame, f:

$$\begin{bmatrix}\mathbf{c}_2' & \mathbf{c}_3'\end{bmatrix}_{2\times2} = \begin{bmatrix}\mathbf{c}_2 & \mathbf{c}_3\end{bmatrix}_{2\times2} \begin{bmatrix}\cos\theta(f) & -\sin\theta(f) \\ \sin\theta(f) & \cos\theta(f)\end{bmatrix}. \tag{3.15}$$

QR decomposition of $[\mathbf{c}_2\ \mathbf{c}_3]^{-1}[\mathbf{c}_2'\ \mathbf{c}_3']$ then gives a rotation matrix from which the angle at the joint, $\theta(f)$, can be recovered.

3.5 Robust segmentation

Before multibody factorization can proceed, it is first necessary to segment the objects in order to group feature tracks according to the object that generated them. However, many existing methods are prone to failure in the presence of dependent motions [27] and gross outliers [120]. We therefore implement a RanSaC strategy for motion segmentation and outlier rejection [112].

Since four points in general position are sufficient to define an object's motion, we use samples of four tracks to find consensus among the rest. We employ a greedy algorithm that assigns the largest number of points with the same motion to the first object. We then remove all of these features and repeat for the second. All remaining feature tracks are discarded since the factorization method uses the SVD (a linear least squares operation) and gross outliers severely degrade performance.

Having segmented the motions, we group the columns of $\mathbf{W}$ accordingly and project each object's features onto its closest $rank = 4$ matrix to reduce noise. We are then in a position to compute the SVD again – this time on the combined matrix of *both* sets of tracks – in order to estimate the parameters of the coupling between them.

3.6 Results

We begin by presenting results for a synthetic sequence of a kinematic chain consisting of three boxes with nine uniformly spaced features on each face (Figure 3.3). Zero-mean Gaussian noise of $\sigma_n \approx 3$ pixels (typical noise levels were measured as $\sigma_n \approx 1$ pixel for real sequences of a similar image size) was then added for a quantitative analysis of the error induced in the recovered joint angle and segment lengths.

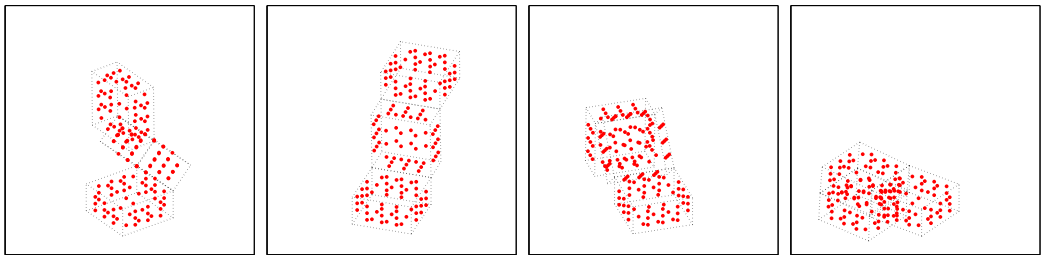
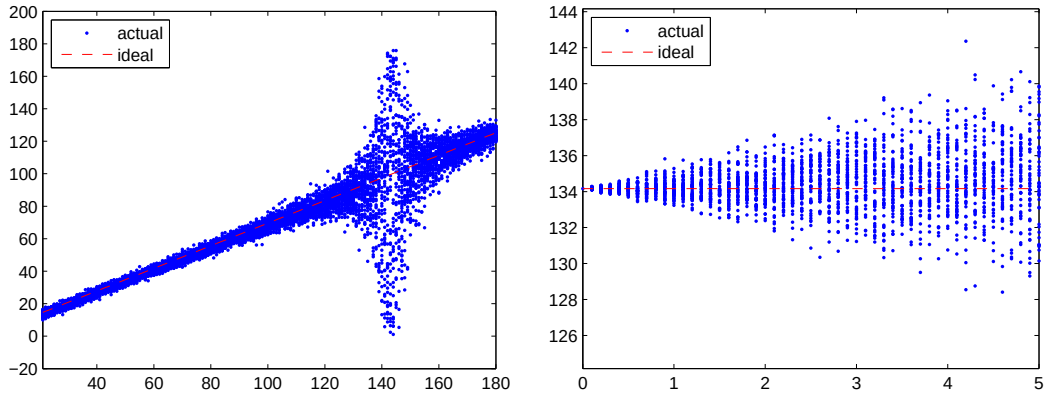


Figure 3.3: Schematic of the 'boxes' sequence displaying three boxes coupled by hinge joints at the edges. Red points indicate features used as inputs to the algorithm.



(a) (b)

Figure 3.4: (a) Recovered joint angle, over 50 trials, for noise level of standard deviation $\sigma_n = 3$ pixels. Note the large increase in error close to frame 143 where the axes of rotation are approximately parallel to the image plane. (b) Distribution of link length error with added Gaussian noise of increasing standard deviation, σ_n pixels, over 50 trials.

3.6.1 Joint angle recovery with respect to noise

Figure 3.4a illustrates the distribution of error in the joint angle at this noise level where we see that error is typically small, increasing dramatically around frame 143. At this point, the axes of rotation in the object are approximately parallel to the image plane such that both $[\mathbf{c}_2\ \mathbf{c}_3]$ and $[\mathbf{c}'_2\ \mathbf{c}'_3]$ are close to singular and the angle derived from $[\mathbf{c}_2\ \mathbf{c}_3]^{-1}[\mathbf{c}'_2\ \mathbf{c}'_3]$ is poorly estimated.

3.6.2 Link length recovery with respect to noise

Using the same sequence, we applied a modified version of the method for longer kinematic chains with parallel axes of rotation to recover the length of the middle link (defined as the distance between the two recovered axes). Since affine projection means that structure and motion can only be recovered up to a global scale, we assume orthographic projection to compare the recovered length with its ground truth value of 134.2 units.

Figure 3.4b shows the error distribution (over 100 trials) for varying levels of image noise. We see that average recovered length is close to the correct value, although the variance of the estimate increases with the level of noise added.

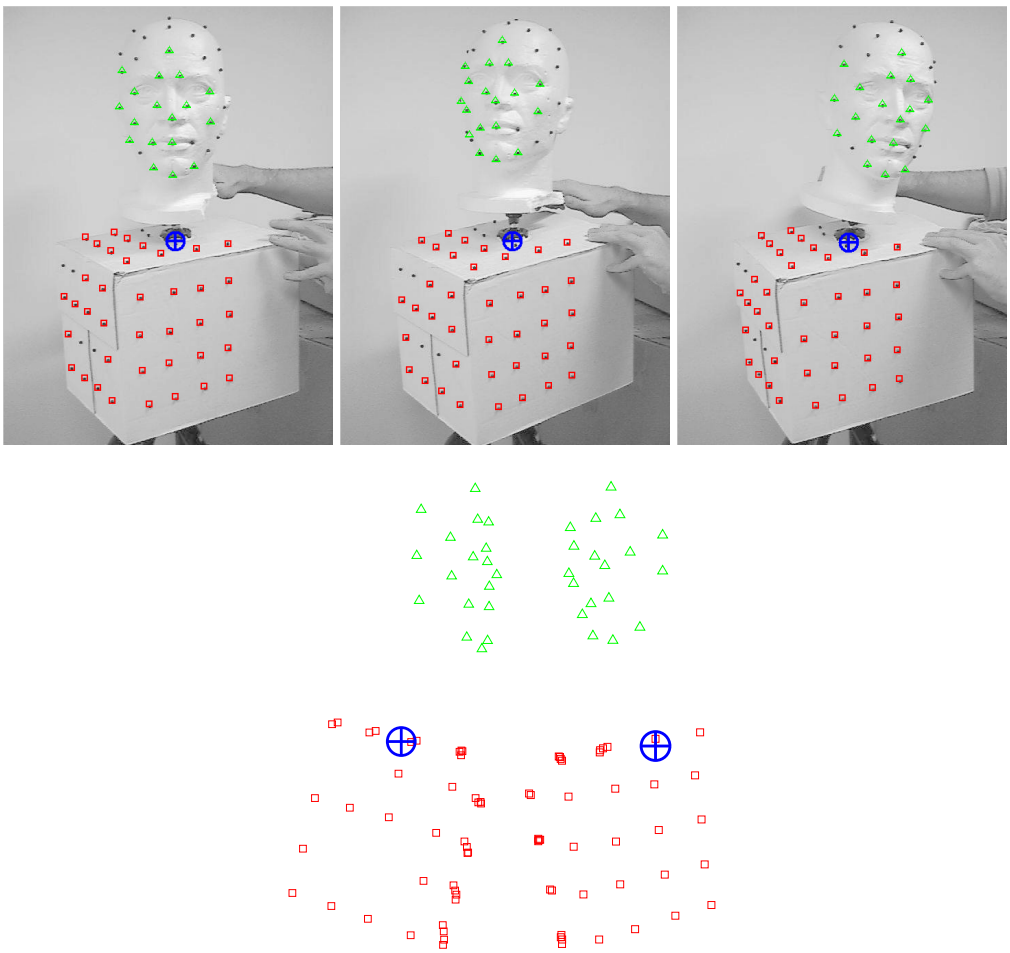


Figure 3.5: (top) Frames from ‘head’ sequence with reprojected features and joint centre. (bottom) Recovered 3D structure and joint centre.

3.7 Real examples

3.7.1 Universal joint

Figure 3.5 shows frames from the ‘Head’ sequence where a model head was coupled to a box by a ball and socket joint. Both the box and the head were rotated about the joint centre to recover structure and motion. By inspection, we see that the reprojected CoR lies within a few pixels of its true location. Visual examination of the recovered 3D structure suggests that the location of the CoR is indeed accurate.

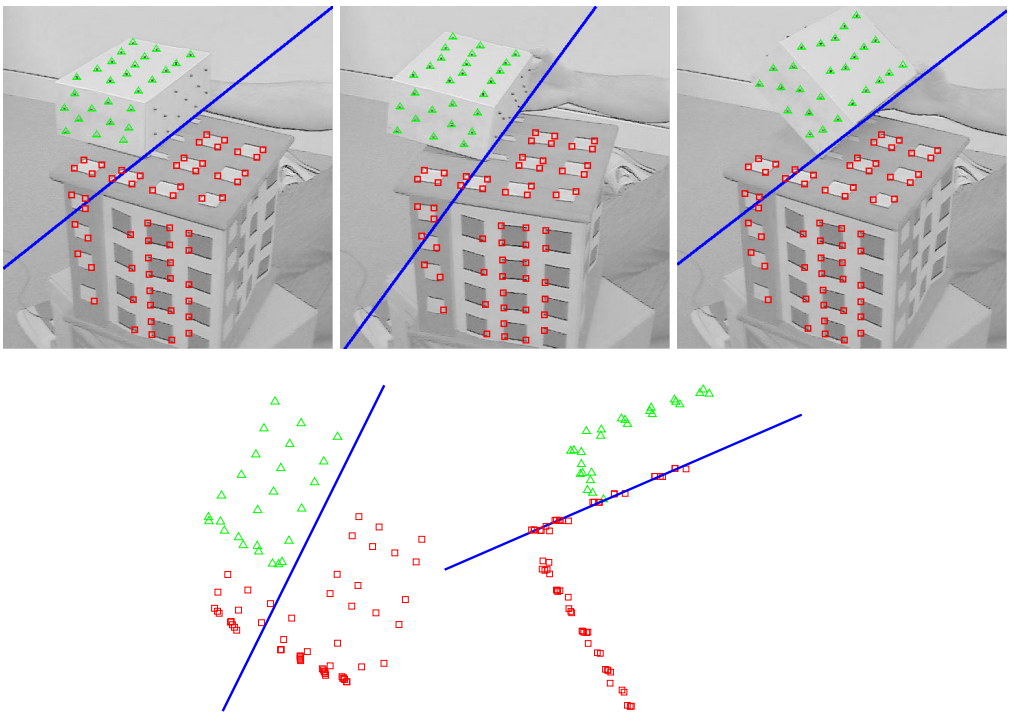
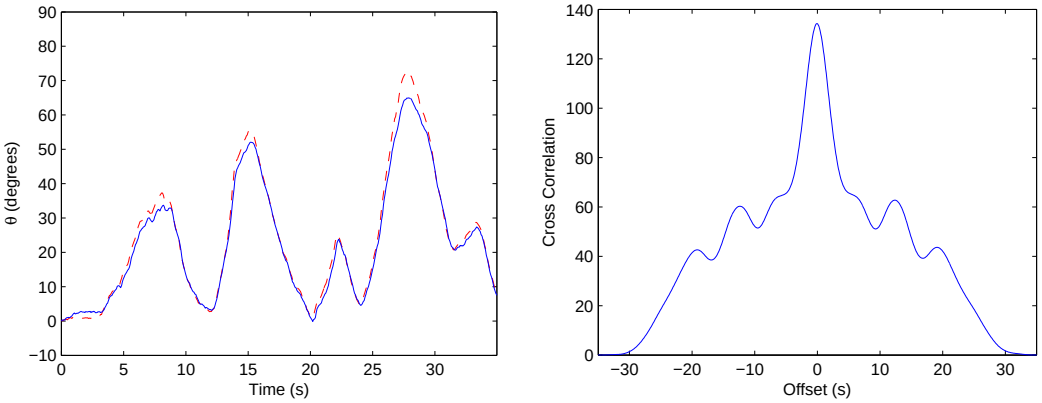


Figure 3.6: (top) Frames from the ‘hinge’ sequence with tracked features and recovered rotation axis. (bottom) Recovered 3D structure and axis of rotation.

3.7.2 Hinge joint

Similarly, Figure 3.6 shows frames from the ‘Hinge’ sequence where two boxes were coupled by a hinge joint. Inspection of the recovered 3D structure shows that the recovered axis lies close to the intersection of the two planes. However, we stress that neither do we use edge information nor do we compute homographies between planes in the scene for our method.

We also demonstrate the recovery of the joint angle for the ‘hinge’ sequence, computing the angle independently for two synchronized views of the same motion and comparing the values recovered from each view (Figure 3.7). We see that there is a error in angle of up to 10° as a result of poorly constrained self-calibration due to limited motion of the base object. This can also be observed as a slight skew in the faces of



(a) (b)

Figure 3.7: (a) Recovered joint trajectories for two sequences showing a good correlation. (b) Cross correlation between recovered trajectories.

Table 3.2: Comparison of singular values for different motions

	$\sigma \times 10^3$				
Dependency	σ_6	σ_7	σ_8	σ_6/σ_7	σ_7/σ_8
None	4.9	4.4	3.0	1.11	1.46
Universal joint	6.1	**4.4**	**0.7**	1.39	**6.28**
Hinge	**4.5**	**0.4**	0.3	**11.25**	1.33

the boxes in the recovered structure (Figure 3.6).

As an aside, we note that the signals in Figure 3.7a could potentially be used to synchronize two image sequences of the same motion by inspecting the cross correlation of the two signals (Figure 3.7b). However, specific synchronization methods exist that may be more appropriate [22, 114, 123].

3.7.3 Detecting dependent motions

Since articulated motion results in a drop in $rank(\mathbf{W})$, the singular values indicate the nature of any dependency. We used real image sequences of two bodies undergoing (i) independent motion, (ii) articulated motion at a universal joint and (iii) articulated motion at a hinge to compose $\mathbf{W}$ and recover its singular values. Table 3.2 shows σ_6, σ_7 and σ_8 (scaled such that $\sum \sigma = 1$) and their ratios where we see that the type of

articulation can be readily observed as a sharp drop in “effective rank”.

3.8 Summary

This chapter has developed the Factorization method [111] for dynamic scenes containing two objects whose motions are *dependent* due to a mechanical coupling such as a hinge. We have shown that in such cases, the rank constraints on the normalized matrix of feature tracks has a tighter lower bound than in the unconstrained case such that specific cases of articulated motion can be detected. Furthermore, we have demonstrated how to recover system parameters such as segment lengths and joint angles using simple linear methods. A quantitative analysis of algorithm performance was presented using synthetic data and the method was also demonstrated on a number of real examples.

3.8.1 Future work

Comparison with Statistical Shape Models

A popular approach towards recovering non-rigid structure from motion has been the use of statistical shape models (SSMs) that describe structure by its mean value plus deviation along some ‘modes of deformation’. This has been successfully demonstrated on faces and tennis shoes where the deformation is small. However, in the case of articulated bodies, where deformations are typically large, the SSM approach is expected to break down. It would be interesting to compare the SSM approach with our proposed method to determine at what level of deformation one approach becomes more suitable than the other.

Longer kinematic chains

Although we demonstrate this method for two links, it is equally applicable to longer kinematic chains since each additional link increases $rank(\mathbf{W})$ by $4-m$ where $m=1$ for a universal joint and $m=2$ for a hinge. However, although the rank constraints extend easily, the recovery of system parameters and self-calibration are not straightforward. This is especially the case for systems where the axes are not parallel or, worse still, where they are not orthogonal. Although each pair of segments in the chain can be treated individually, this would not satisfy all constraints at the same time and is a sub-optimal solution.

Closed chain kinematics

Although closed chains are less common in real-life, it would be interesting to examine how the constraints imposed by pairs of bodies could be applied. An example of closed chain kinematics was previously studied, although in a slightly different context, by Taylor [107] for affine reconstruction from a single view.

Chapter 4

Recovering 3D Joint Locations II : Machine Learning

*In contrast to the previous chapter, we now consider a number of Machine Learning approaches to recover joint locations in an image sequence based on a corpus of training data. In doing so, we exploit our prior knowledge of structure (*i.e. *the human body) to generate exemplars of observations. Given a novel observation, we infer joint centre locations by searching, sampling from or regressing over the stored exemplars. Putative estimates of the joint centre locations are then refined over the sequence by employing a particle filter to exploit the available rich image data and impose smoothness constraints.*

4.1 Introduction

In Chapter 3, we demonstrated a geometric method of recovering centres and axes of rotation for objects with an unknown structure. However, in the case of human motion tracking we can exploit the fact that the structure of the human body is known to generate a database of synthetic observations (*e.g.* silhouettes generated using graphical software such as Poser [35]) with their corresponding 3D poses. Given a novel observation, we may then search for nearest neighbours in the training data (using an observation-based error metric) and use their corresponding stored poses as putative estimates of the query pose.

More precisely, our goal is to estimate or sample from the distribution, $p(x_t|z_t)$,

over pose, x_t, given some observations, z_t, generated from the original image data, D_t. Due to the articulated nature of the human body, it can be shown that a given silhouette may result from one of several different underlying poses due to 'kinematic flips' [102] that occur by reversing the relative depth between two joint centres. This exponential number of 'flips' results in a highly multimodal $p(x_t|z_t)$.

Many solutions can be eliminated via joint limit constraints, enforced implicitly by including only valid poses in the training data. An alternative approach (used in this work) is to track only the 2D projections of the joint locations, thus reducing the state space since all possible 3D 'flips' project to the same solution in 2D image space. However, some ambiguity is unavoidable for a single view and cannot be resolved. For example, consider standing behind someone looking in a mirror: the reflection shows a laterally inverted image facing in the opposite direction yet the occluding contours are seen to be identical for both the person and their reflection.

In the case of human motion *tracking*, temporal constraints may also be enforced to ensure that the recovered motion is smooth. This is typically implemented using tools such as the Kalman Filter or Particle Filter. Furthermore, enforcing priors over the pose at a given instant in time, based on pose at previous instants, provides an additional mechanism for resolving ambiguity and a likelihood function allows initial estimates to be refined using rich image data. However, such trackers still require hand-initialization to resolve ambiguity at the first frame.

In this work each state vector, x, consisted of the 2D joint centre projections in the image. For synthetic data, these were computed from the original 3D pose parameters and a kinematic model whereas for real data the joint centre projections were labelled manually. Each observation, z, was represented by a 100D feature vector containing

Discrete Cosine Transform (DCT) coefficients of a 128×128 silhouette generated using a volumetric model (for synthetic data) or background subtraction (for real data). DCT coefficients were selected as an appropriate representation since they offer an excellent compromise between accuracy and efficiency. This selection is justified in the more thorough investigation presented in Appendix A. All silhouettes were normalized before computing feature vectors in order to provide some invariance to translation and scaling.

4.1.1 Related Work

As discussed in Section 2.1.1, traditional top-down (model-based) approaches to human body tracking fit a kinematic model of the body to a sequence of image observations. The state at the current instant is predicted using state estimates at the previous instant and a dynamical motion model. Predicted states are then weighted based on agreement with current observations via a likelihood model.

In contrast, bottom-up methods driven by training data, rather than a predictive dynamical model, have become highly popular in recent years (see Section 2.1.2). Training data commonly consist of synthetic images of a 3D human model generated using graphics software (*e.g.* Poser) and their corresponding state (pose). The data may then be searched to find k nearest neighbours (based on an image-based distance metric), thus recovering k state estimates. The search may be made efficient using coarse-to-fine searching [8], tree structures [39, 104] or hashing [43, 91].

Due to the heavy demands on storage and computation that are imposed by searching a database, alternative approaches directly model the mapping between image observations and pose using Machine Learning techniques such as Artificial Neural Networks [88], Relevance Vector Machines [5], Probabilistic PCA [45] and Hidden

Markov Models [16]. Once the mapping has been learned, the training data can be discarded to reduce storage requirements and increase efficiency. However, extra measures are required to recover a multimodal p.d.f. over pose, such as employing mixture models [98, 4].

It is notable that current data-driven methods for monocular tracking typically discard most of the image information, often computing a silhouette that is reduced further to a relatively low-dimensional feature vector from which pose is estimated. However, almost no published work exploits the fact that rich image information remains available to refine putative estimates or resolve ambiguities via a likelihood function.

4.1.2 Contributions

In this chapter, we combine the strengths of data-driven and top-down tracking by incorporating single-frame pose estimation techniques into a particle filtering framework. Specifically, we generate particles from a "hybrid" prior distribution over pose using both training data and a predictive motion model.

- Sections 4.2 and 4.3 outline a number of proposed methods for single-frame pose estimation: linear search, tree searching/sampling, linear/kernel regression, Relevance Vector Machines and Neural Networks. These methods are compared in terms of accuracy and efficiency on a synthetic 'exercise' sequence in Section 4.5 for training datasets of increasing size to investigate each method's ability to scale.

- The integration of pose estimation methods with top-down filtering is detailed in Section 4.4. Using a hybrid prior that exploits the most up-to-date observations and predictions based on previous estimates results in a tracker that is robust to

periods of occlusion and unpredictable motions. Furthermore, the particle filter provides a principled way of refining estimates based on silhouettes alone by exploiting the additional image data that is available (*e.g.* edges, colour, texture) via a likelihood function. We note that many data-driven methods [5, 98, 43] discard this valuable information rather than take advantage of it.

4.2 Searching and Sampling

We begin by outlining searching and sampling techniques to recover estimates of pose from the database. Such approaches have the advantage that recovered samples are constrained to be valid configurations. However, this also limits the resolution of recovered pose since the output is defined only for a number of discrete values (the training exemplars) such that the transition between recovered poses may appear discontinuous. The error induced by such discretization may be reduced by interpolation between recovered exemplars although care must be taken to ensure that the resulting configuration is valid (*e.g.* by interpolating joint *angle* rather than position).

4.2.1 Linear Search

The simplest method of exemplar-based pose estimation is simply to search the database linearly for the exact nearest neighbour of the query feature vector. Alternatively, by searching for the k nearest neighbours, we recover multiple solutions reflecting pose ambiguity in monocular tracking. It can be shown that this method is highly inefficient, scaling as $O(N)$ in both storage and computation, since every exemplar feature vector must be stored with its corresponding state and a distance computation must be performed for every exemplar in the dataset. For the purposes of the following comparison, however, this method serves as a simple baseline.

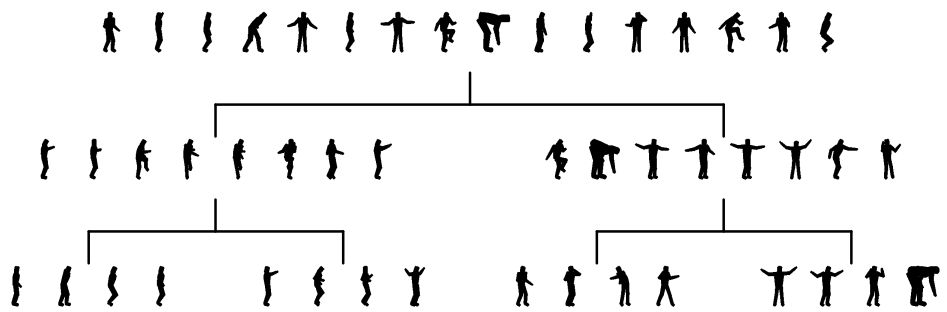


Figure 4.1: Silhouettes sampled uniformly from the tree structure. Note how the variation in the samples decreases from the root to the leaves. The leaf corresponding to a query example can be computed rapidly and only that leaf (or nearby leaves) searched or sampled to generate possible matches. Discarding a large proportion of the tree in this manner greatly increases efficiency.

4.2.2 Tree Search

Since a linear search of the database is highly inefficient, various methods have been developed to reduce the number of distance computations required. An obvious approach is to construct a tree, partitioning the input space into a number of 'leaves' (Figure 4.1). Given a novel feature vector, only those exemplars that fall into the same leaf as the query are searched while the rest are ignored. As a result, searching becomes considerably more efficient in terms of run-time processing, scaling as $O(\log N)$. In terms of storage requirements, however, tree searching still requires all feature vectors to be stored and scales as $O(N)$ although it does offer the possibility of storing each leaf of the tree at a different location in secondary storage, transferring data to primary storage only as required.

A common problem with tree searching arises when a query vector falls close to a boundary since the correct nearest neighbour may be in an adjacent leaf and therefore will not be recovered. Furthermore, in high dimensions the exponential explosion in the number of leaves results in many leaves empty. We address this second problem by

descending the tree until the number of exemplars below that node falls below some threshold (we use 100 exemplars).

4.2.3 Tree Sampling

An alternative approach assumes that all exemplars below a given depth in the tree are sufficiently similar that they can simply be sampled rather than searched. This eliminates the need to store *any* feature vectors since they can be discarded once the tree has been constructed. In theory, by discretizing pose space in an appropriate way, storage requirements may be reduced to fewer than 50 bytes/exemplar such that the entire database could easily be held in main memory for million of exemplars, rather than thousands.

In practice, the tree must be traversed to a greater depth than in searching if the assumption of sufficient similarity is to be accurate. We therefore continue to descend the tree for as long as all leaves below the current node are non-empty. This typically results in sampling from a small selection of similar examples.

4.3 Regression

We now look at regression approaches that differ from searching and sampling by assuming a continuous relationship between inputs (observations) and outputs (pose):

$$\mathbf{x} = f(\mathbf{z}) \tag{4.1}$$

such that recovered poses are no longer restricted to the training exemplars alone.

4.3.1 Linear Regression

In linear regression, it is assumed that $f(\mathbf{z})$ is a linear function such that:

$$\mathbf{x} = \mathbf{A}\mathbf{z}. \tag{4.2}$$

The matrix $\mathbf{A}$ is estimated by defining $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ such that:

$$\mathbf{A}\mathbf{Z} = \mathbf{X} \tag{4.3}$$

$$\mathbf{A}\mathbf{Z}\mathbf{Z}^T = \mathbf{X}\mathbf{Z}^T \tag{4.4}$$

$$\mathbf{A} = \mathbf{X}\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}. \tag{4.5}$$

In practice, *ridge regression* is often employed in order to impose a regularization penalty that avoids overfitting and improves generalization:

$$\mathbf{A} = \mathbf{X}\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T + \lambda\mathbf{I})^{-1}. \tag{4.6}$$

In terms of computation, 'training' of this system requires the inversion of the $d_z \times d_z$ matrix $\mathbf{Z}\mathbf{Z}^T + \lambda\mathbf{I}$ where d_z is the dimensionality of the feature space. Once $\mathbf{A}$ has been computed, the training feature vectors are no longer required and can be discarded. As a result, linear regression is efficient in both storage and run-time computation, consisting of a simple matrix multiplication to recover the state for a novel query. However, the assumption of a linear relationship between the input feature vector and corresponding state is seldom accurate, typically resulting in large errors.

Sparse RVM Regression

A recent extension to linear regression was provided by the principle of automatic relevance determination, popularized as the Relevance Vector Machine [110] and em-

ployed in [5]. This modification sparsifies $\mathbf{A}$ by applying independent priors over its columns. Specifically, (4.6) is modified such that:

$$\mathbf{A} = \mathbf{X}\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T + diag(\lambda))^{-1} \tag{4.7}$$

where $\lambda = [\lambda_1, \ldots, \lambda_{d_z}]$ is the vector of regularizing coefficients. Since columns with smaller norms contribute less to the output state vector, they are deemed less 'relevant'. Increasing the damping to these columns by making λ_i inversely proportional to the norm of column i drives them towards zero. Meanwhile, an opposing 'force' is applied by the data to provide a compromise between sparsity and accuracy. When a column norm falls below a specified threshold, the column is 'irrelevant' and is therefore removed from $\mathbf{A}$. Similarly, the corresponding rows of the feature vector are removed, thus implementing a form of feature selection.

Although this method is more computationally expensive in the training (due to multiple iterations of the least-squares operation), it saves on run-time computation since fewer features (and hence fewer multiplications) are utilized. However, since $\mathbf{A}$ is of a fixed size and is typically small, the computational benefits of linear RVM regression are limited although experimental results suggest that sparsifying $\mathbf{A}$ has other advantages such as improving robustness to noise (see Figure 4.3).

4.3.2 Kernel Regression

An alternative approach that does not assume a linear relationship between inputs and outputs is that of kernel regression. The feature vector is 'lifted' to N-dimensional space prior to linear regression taking place such that:

$$\mathbf{x} = \mathbf{A}\mathbf{k}(\mathbf{z}) \tag{4.8}$$

where:

$$\mathbf{k}(\mathbf{z}) = \begin{bmatrix} K(\mathbf{z}, \mathbf{z}_1) \\ \vdots \\ K(\mathbf{z}, \mathbf{z}_N) \end{bmatrix} \tag{4.9}$$

and $K(\mathbf{z}, \mathbf{z}_i)$ is a kernel function reflecting similarity between the given feature vector, $\mathbf{z}$ and an exemplar, $\mathbf{z}_i$. A common choice is the radially symmetric gaussian kernel:

$$K(\mathbf{z}, \mathbf{z}_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left\{ -\frac{1}{2}(\mathbf{z} - \mathbf{z}_i)^T \Sigma^{-1} (\mathbf{z} - \mathbf{z}_i) \right\}. \tag{4.10}$$

where Σ is estimated from the covariance of the data[1]. By collecting the kernelized feature vectors into a matrix:

$$\mathbf{K} = \begin{bmatrix} K(\mathbf{z}_1, \mathbf{z}_1) & \dots & K(\mathbf{z}_N, \mathbf{z}_1) \\ \vdots & \ddots & \vdots \\ K(\mathbf{z}_1, \mathbf{z}_N) & \dots & K(\mathbf{z}_N, \mathbf{z}_N) \end{bmatrix}, \tag{4.11}$$

training proceeds in the same way as linear regression such that:

$$\mathbf{A} = \mathbf{X}\mathbf{K}^T (\mathbf{K}\mathbf{K}^T)^{-1}. \tag{4.12}$$

Kernel regression typically demonstrates improved performance over linear regression, particularly when the relationship between input and output is non-linear, since it effectively interpolates between exemplars in the training set.

However, there are considerable drawbacks as a result of the increase in feature vector dimension from d_z to N since estimating $\mathbf{A}$ now requires the computation and inversion of an $N \times N$ matrix. For large N, this rapidly becomes intractable since the computation of $\mathbf{K}$ has complexity $O(N^2)$ whilst its inversion requires $O(N^3)$ computation. Furthermore, the storage of this matrix increases as $O(N^2)$ such that 15000

[1] Σ need not be the actual covariance matrix of the data

exemplars would require 1.8Gb of storage at double precision. These constraints impose severe limits to the number of exemplars that can be employed in any practical system.

Sparse RVM Regression

There is a corresponding RVM version of the kernel regressor that eliminates 'irrelevant' columns of the matrix $\mathbf{A}$, thus eliminating irrelevant *exemplars* (rather than image *features*) such that the training dataset is pruned for efficiency. However, the first iteration of this method still requires the inversion of an $N \times N$ matrix at high computational cost. Methods have been proposed to address this issue [118] by introducing exemplars sequentially.

For the purposes of this study, however, we employ a simple "one in, one out" strategy whereby we initialize with n exemplars at random and begin iterating. Whenever an exemplar is eliminated, we replace it with one of the remaining exemplars and continue until all exemplars have been presented to the algorithm. Empirically, this has proved to be a viable alternative to 'batch' RVM regression. A sensible value of n is selected heuristically, typically on the order of $n = 1000$.

4.3.3 Neural Networks

An alternative regression method is that of the Artifical Neural Network [14]. In this model, the non-linear relationhip between inputs and outputs is defined by the network structure and parameters (*i.e*. layer weights and biases). Parameters are optimized using non-linear minimization techniques (*e.g*. gradient descent, conjugate gradients) using derivatives obtained via backpropagation for efficiency. A common structure, as employed in this work, has two layers of weights with a linear transfer function at the

outputs and a tangential sigmoid transfer function at the hidden layer. As a result, the mapping from observations can be written as:

$$\mathbf{x} = \mathbf{W}_2 \cdot tansig(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2 \tag{4.13}$$

where

$$tansig(a) = \frac{2}{1 + \exp(-2a)} - 1. \tag{4.14}$$

In this work, we implement the network using the Neural Network Toolbox for Matlab although alternatives are available (*e.g.* NetLab).

4.3.4 Mixture Models

Since all of the regression methods described so far are one-to-one, they cannot model the one-to-many relationship that exists between silhouette and pose. In order to address this problem, mixture models can be employed such that several regressors trained on a particular region of feature space output the different possible solutions. This also avoids the problem of averaging that commonly occurs in methods such as kernel regression whereby a query with two neighbours, close together in feature space but far apart in pose space, is assigned the average pose that corresponds to neither of the exemplars.

However, mixture models typically require clustering of the data in pose space before training the regressors – a difficult task in itself for many exemplars in high-dimensional space. Since we do not attempt a comprehensive comparison of regression techniques in this chapter, mixture models are not pursued any further in this study.

4.4 Particle Filtering

4.4.1 Hybrid prior

In order to impose smoothness over a *sequence* of poses and exploit additional image information (*e.g.* edges), we incorporate a discriminative method into a particle filtering framework [57]. This is achieved by defining a *hybrid* prior that draws samples from both a predictive distribution, $p(x_t|D_{t-1})$, and the data-driven distribution, $p(x_t|z_t)$, that uses coarse but up-to-date observations. We combine the two distributions via a simple weighting:

$$p(x_t) = (1-\alpha)p(x_t|D_{t-1}) + \alpha p(x_t|z_t). \tag{4.15}$$

This formulation allows a simple (*e.g.* constant velocity) motion model to handle small periods of observation error while the data-driven samples provide robustness to motions that are not predicted by the motion model.

We note that α need not be fixed throughout the sequence. For example, at the beginning of the sequence, $\alpha = 1$ ensures that the predictive model is not employed (since no previous estimates are available from which to predict). Conversely, if the current observations are well outside of the training set (*e.g.* during moments of heavy occlusion or frame drop-out), $\alpha = 0$ ensures that predictions only are propagated since the discriminative model is likely to be unreliable.

4.4.2 Likelihood

Through the use of a likelihood model, we 'close the loop' with the available rich image information (*e.g.* internal edges, colour, texture) – a valuable source of information that is largely neglected in other silhouette-based methods [5, 98, 88]. For the purposes

of this chapter, we employ the silhouette and a masked edge map (see Figure 2.3).

Since we track only the 2D projections of the joint centres, an estimate is required for the scale of the body such that a volumetric model can be projected in the correct proportion. Therefore, we assume that one of the limbs lies in a plane parallel to the image plane [107] to determine the projected widths of the limbs. These predicted observations are then compared with the silhouette generated by background subtraction and the edge map derived from Canny edge detection on the original image to weight each estimate *i.e.* to evaluate $p(D_t|x_t)$. Note that this crude likelihood model would benefit from more discriminative features (*e.g.* colour, texture, optic flow) to reduce the effective spread (posterior covariance) of particles drawn from $p(x_t)$. However, we defer this for future work.

4.5 Results

We begin by presenting results on synthetic sequences for the data-driven pose estimation, comparing the described methods. A method is then selected and integrated into a particle filtering framework to serve as the proposal distribution, $p(x_t|z_t)$.

4.5.1 Data-Driven Pose Estimation

We begin by comparing the various methods outlined in Sections 4.2 and 4.3 using synthetic training sets of $\sim$1000, $\sim$5000 and $\sim$15000 exemplars. These datasets were selected to evaluate each method's scalability as most have been demonstrated on relatively small datasets of only two or three thousand exemplars.

A 291-frame synthetic sequence of an exercise routine (Figure 4.2) was generated from the same motion capture database and used as the test sequence. This sequence was not included in the training data although it was generally well represented by

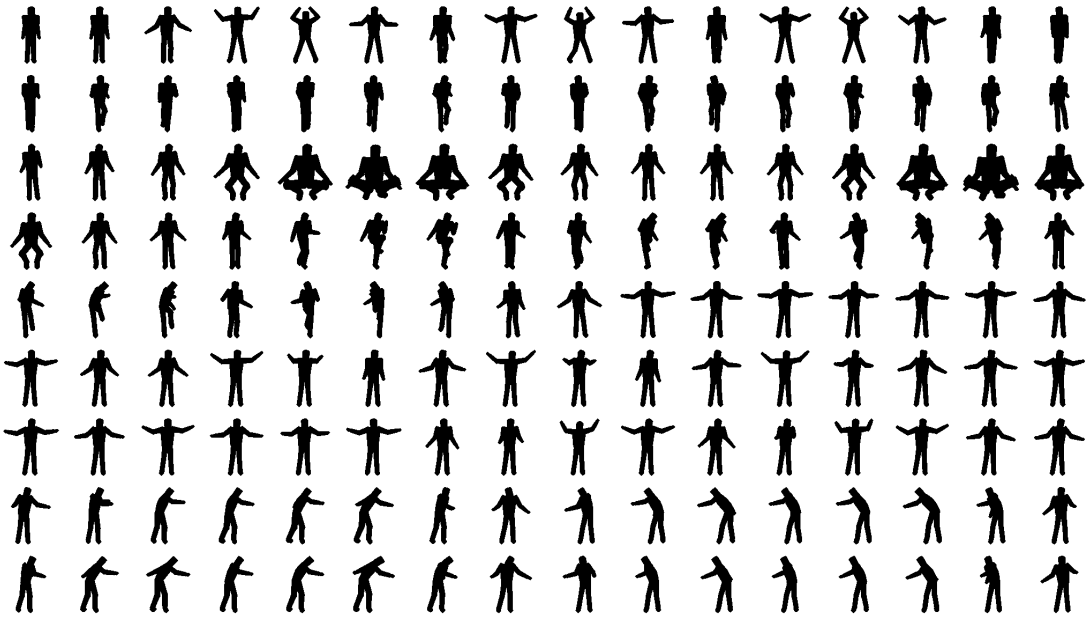


Figure 4.2: Synthetic 'exercise' sequence used to evaluate joint centre recovery methods. Each silhouette is annotated with the corresponding 2D projections of joint centres in the image, computed from the 3D pose parameters.

other exemplars in the database.

We evaluated each method in terms of accuracy with respect to known ground truth values. Specifically, we computed the mean RMS error between 2D joint centre projections for every frame of the 291-frame sequence. This was repeated for each method applied to all three training data sets, and the results are shown in Table 4.1. We also recorded the time for each method to execute, both in terms of off-line training and run-time execution, as shown in Table 4.2.

We make a number of observations from these results:

- In general, searching methods are most accurate since they are constrained to return exemplars from the training set. In contrast, sampling neglects distance in feature space and regression methods have greater freedom to deviate from the training examples as well as interpolate.

	1K	5K	15K
Linear Search	2.840	3.198	3.236
Tree Search	2.885	3.364	4.686
Tree Sampling	6.849	7.933	8.329
Linear Regression	4.479	7.839	9.433
Linear RVM	6.273	9.274	10.78
Kernel Regression	2.132	3.480	–
Kernel RVM	3.220	6.599	8.657
Neural Network	3.448	6.596	7.057

Table 4.1: Accuracy of various methods for single-frame pose estimation using training sets of increasing size. Values given are the mean RMS error between projected joint locations over the synthetic exercise sequence.

	Off-line			Run-time		
	1K	5K	15K	1K	5K	15K
Linear Search	0.000	0.000	0.000	2.217	13.11	39.29
Tree Search	0.023	0.244	0.745	0.721	2.058	5.777
Tree Sampling	0.027	0.245	0.730	0.332	1.483	5.352
Linear Regression	0.043	0.113	0.299	0.019	0.019	0.015
Linear RVM	0.071	0.127	0.326	0.016	0.016	0.019
Kernel Regression	0.875	1012	–	2.389	21.49	–
Kernel RVM	1.342	25.00	93.90	0.310	0.713	1.640
Neural Network	33.27	131.3	584.7	2.201	2.204	2.198

Table 4.2: Times to compute for various methods of single-frame pose estimation using training sets of increasing size. Times are indicated in seconds.

- Non-linear regression is accurate for small datasets but degrades in performance as datasets increase in size. Linear regression is generally poor in comparison to other methods.

- Searching and sampling are inefficient when compared with most regression approaches (with the exception of kernel regression). Sampling can be made more efficient by sampling from only those exemplars assigned to the same leaf as the query although this does not guarantee that any matches exist.

- Dense kernel regression cannot handle datasets of more than a few thousand.

- Non-linear regression methods are more efficient at run-time, albeit at the expense of high computational cost during off-line training (for only ~ 5000 exemplars, kernel regression required almost 17 minutes to compute the regression matrix, $\mathbf{A}$).

With respect to the RVM regression methods, a trade-off is made between sparseness (and hence efficiency) and accuracy via a design parameter. For the purposes of these experiments, the linear RVM was designed to retain $\sim 25\%$ of the input features whereas the kernel RVM retained 78, 284 and 528 relevant examples from the 1K, 5K and 15K datasets, respectively.

Finally, the 100D vector of DCT coefficients corresponding to a real silhouette from the starjumps sequence was computed. From this seed, Gaussian noise of standard deviation equal to 10% that of the training set was added to generate 100 noisy feature vectors. Each method was then applied to these feature vectors, effectively sampling from the distribution $p(x|z)$ for an uncertain z. The recovered 2D joint centre projections are shown in Figure 4.3.

From these samples, we can see some additional properties of the methods:

- Linear search is largely unaffected by an increase in the size of the training dataset. Tree searching also appears to be relatively robust, although baseline accuracy is lower than other methods for small datasets.

- Linear RVM regression appears to be considerably more robust to noise than standard linear regression.

- The effects on non-linear regression methods of increasing the training set size is visible as averaging takes place over exemplars that are close in feature space

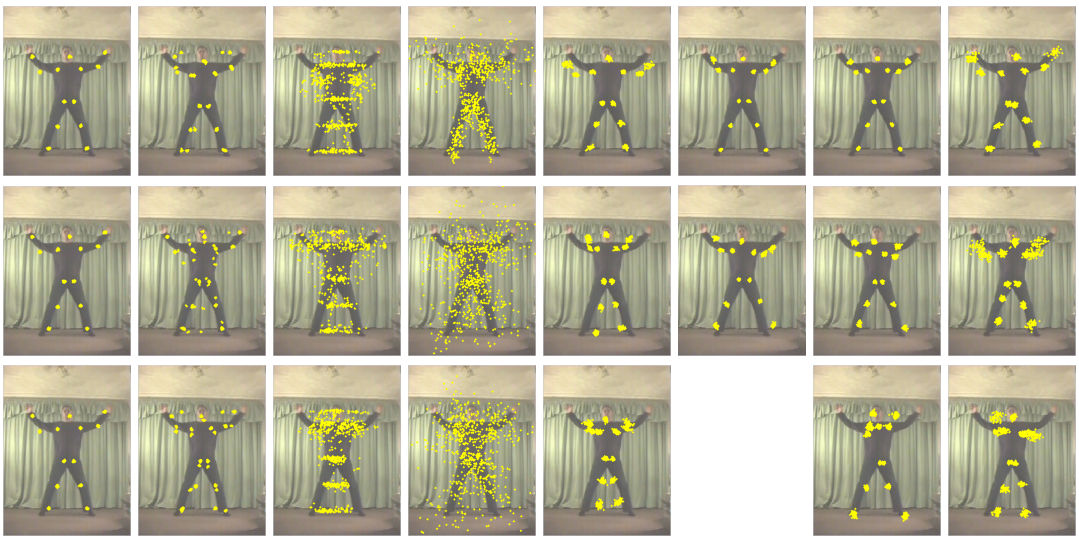


Figure 4.3: Samples drawn using (from left) linear search, tree search, tree sampling, linear regression, linear RVM, kernel regression, kernel RVM, neural network for (from top) 1K, 5K and 15K exemplars (note that kernel regression was not possible for the 15K dataset).

but distant in pose space.

4.5.2 Particle filtering

Finally, we track the exercise sequence by incorporating the tree searching method into the hybrid prior for stability. Figure 4.4 shows the results of the tracking using a weak predictive motion model (constant velocity of the 3D joint locations, learned from the dynamics of the training set)

4.6 Real Examples

We now present results on a number of real sequences, using background subtraction and morphological operators to extract the silhouette. Such operations are typically restricted to environments with controlled lighting and static backgrounds. Departure from such an environment results in corruption of the silhouette due to dynamic backgrounds, shadows and highlights. Such corruption is likely to degrade performance

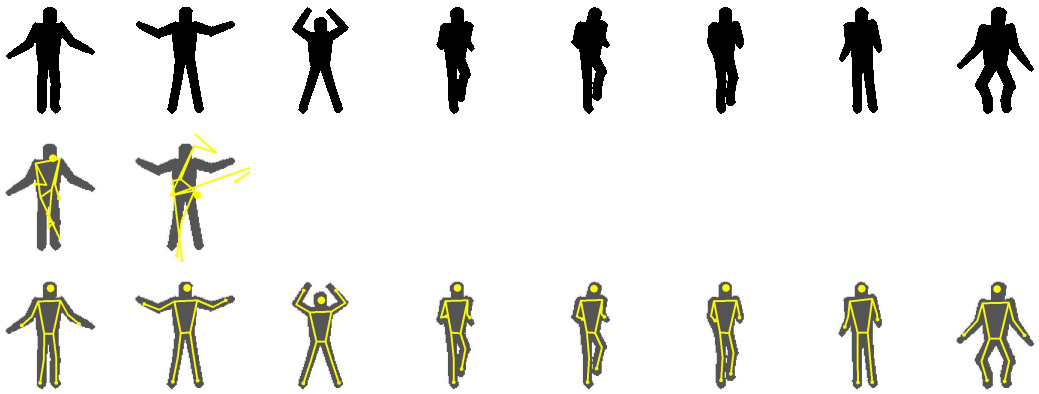


Figure 4.4: (top) Original exercise sequence (frames $5, 15, \ldots$); (centre) Sequence tracked using a weak predictive motion model (tracking was lost after 23 frames); (bottom) Sequence tracked using both predictive motion model and exemplars (whole sequence tracked).

of the Machine Learning algorithms implicitly by corrupting the feature vector that is presented to the algorithm and is therefore dependent on the choice of shape descriptor. However, to maintain the flow of the thesis this issue is investigated in in Appendix A.

4.6.1 Starjumps sequence

We applied the particle filtering algorithm to a real 157-frame sequence of the author performing starjumps in an environment with a static background (Figure 4.5). As a result, the silhouette was generated via background subtraction followed by morphological operators to remove spurious regions.

From Figure 4.5 it is evident how easily tracking is lost when using a weak predictive model. Although this may be improved by tracking in 3D using joint angles as a state vector such that projected joint locations are more constrained, this introduces problems with kinematic 'flips' [102]. In contrast, the estimates generated by the hybrid prior are tightly constrained around the correct solution.

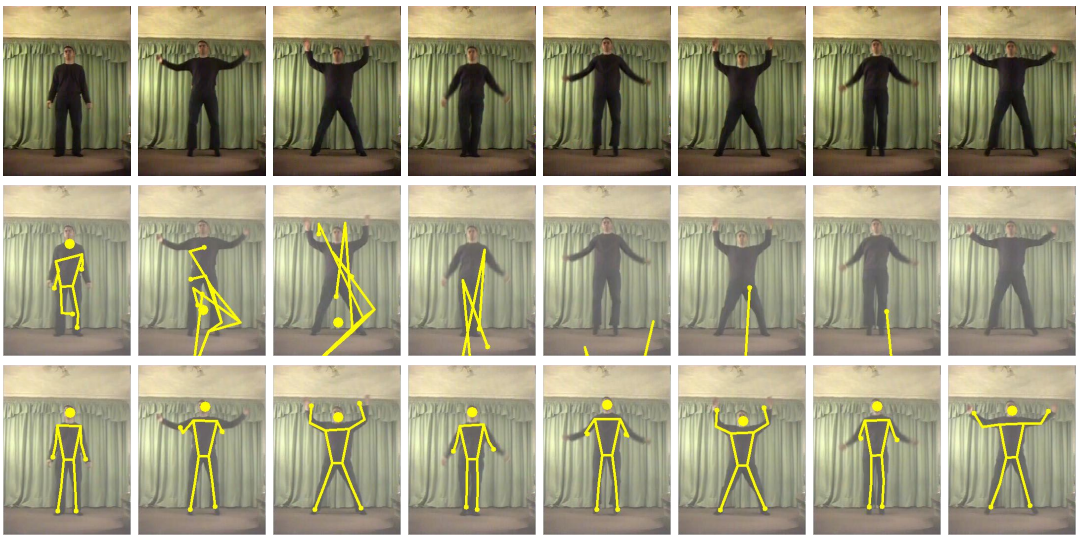


Figure 4.5: (top) Original starjumps sequence (frames $5, 15, \ldots$); (centre) Sequence tracked using a weak predictive motion model (tracking was lost after 100 frames); (bottom) Sequence tracked using both predictive motion model and exemplars (whole sequence tracked).

4.6.2 Squats sequence

Finally, we apply the method to a 284-frame squatting sequence resulting in similar tracking success (Figure 4.6). However, observe that in some frames the squatting stance is slightly different between the image and the pose estimate (especially the arms) as a result of a bias towards the training data.

4.7 Summary

This chapter has presented a comparison of several discriminative methods for estimating pose from a query silhouette and a large training corpus of synthetic exemplars. In particular, we compared several state-of-the-art methods using training datasets of increasing size to assess their scalability. Our results demonstrate that some methods, although accurate, are impractical for large datasets. In particular, regression methods are observed to degrade more rapidly than searching approaches as the dataset

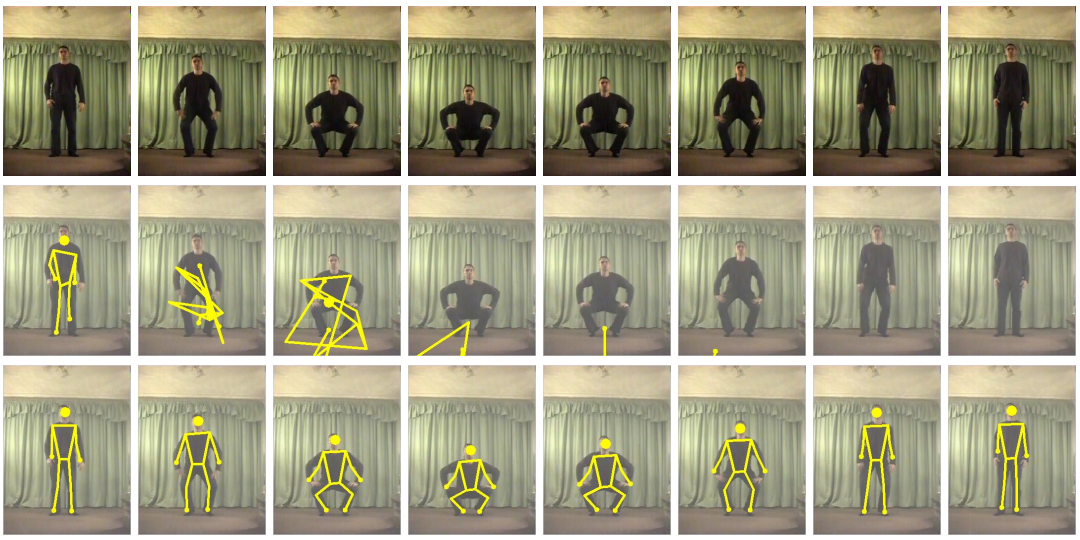


Figure 4.6: (top) Original squats sequence (frames $5, 15, \ldots$); (centre) Sequence tracked using a weak predictive motion model (tracking was lost after 87 frames); (bottom) Sequence tracked using both predictive motion model and exemplars (whole sequence tracked).

increases in size.

We also demonstrated that discriminative methods can be incorporated into a particle filtering framework in order to impose some smoothness over the sequence and exploit the available rich image data. Using a weak predictive model (as is common for highly varied training data), tracking is shown to fail after only a few tens of frames. In contrast, the closed-loop tracking provided by resampling from the proposal distribution at each frame ensures that tracking is maintained and recovery from tracking failure is possible.

4.7.1 Future work

Mixture models

As noted in Section 4.3, mixture models provide a way of generating alternative solutions for a given silhouette. Furthermore, they can modify the regression model as a function of the silhouette such that more complex mappings can be learned. This is

essential for large datasets since, as demonstrated by our results, a single regression function is rarely adequate to model such complex mappings.

Advanced tree searching and sampling

The results suggest that searching and sampling approaches scale well with respect to the training data. It may be constructive to pursue these methods further to improve efficiency further without sacrificing significant levels of accuracy.

Chapter 5

Video Synchronization

This chapter addresses the problem of automatically synchronizing two sequences using projected joint centres. We define a metric that assigns a low cost to frames that are structurally consistent and a high cost to those that are not. The metric is derived for homography, perspective and affine projection models. In the affine case, we see that the familiar rank constraints follow naturally from this general metric. Having estimated corresponding frames, we present an algorithm that estimates the alignment parameters to sub-frame *accuracy even for sequences of* different *frame rates. The performance of the algorithm is evaluated using synthetic sequences and demonstrated on several real examples.*

5.1 Introduction

So far, we have discussed how to recover the locations of joints in articulated structures from image sequences, using both geometric (Chapter 3) and Machine Learning (Chapter 4) methods. In the following two chapters, we discuss how they are used to recover the pose of the subject at each instant in time from a pair of sequences.[1]

Two sequences must first be aligned in time (*i.e.* synchronized) since the 3D position of scene features can only be triangulated from stereo images that were captured *at the same time*. Commercial motion capture systems (*e.g.* Vicon) do this using hardware – a costly and technically complex engineering solution. In contrast, we show that recovered joint locations can be used to align the sequences in time if we know

[1] Portions of this chapter were published in [114, 115]

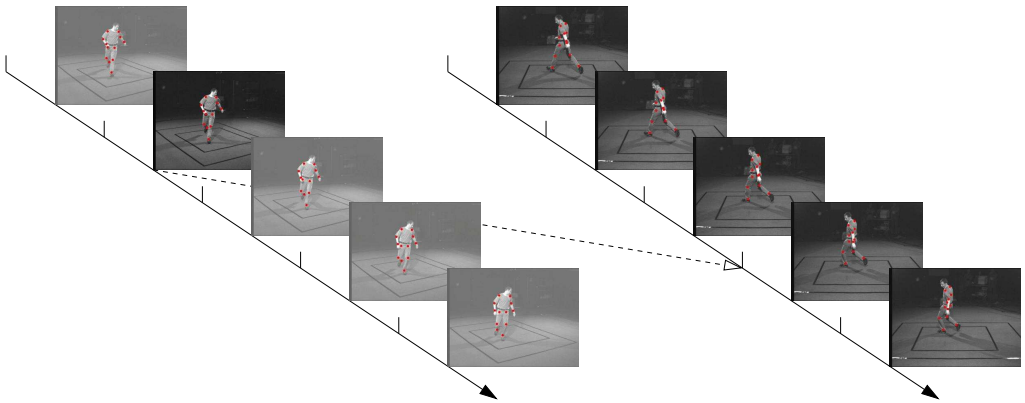


Figure 5.1: Timelines depicting a wide baseline stereo sequence with synchronization of the cameras indicated by the arrow. This shows an example where no corresponding frame exists in the second sequence due to the sub-frame offset of the cameras.

corresponding points between the sequences.

The spatial correspondence problem is solved in commercial systems using synchronized, calibrated cameras with markers – again, a complex engineering solution. In our case, we track the human body and therefore have an intuitive labelling of the joint locations such that correspondence between the sequences is provided.

Consider the case where we are presented with two sequences captured from unsynchronized cameras, possibly with different frame rates (Figure 5.1). Given a frame in the left-hand sequence (the darker frame), we recover the corresponding instant in the right-hand sequence, indicated by the arrow (in this case, exactly halfway between frames).

We see that a frame may not physically exist at the corresponding instant due to the cameras being unsynchronized. If frames f and f' (from sequences 1 and 2, respectively) correspond to the same instant in time then they are related linearly by:

$$f' = \alpha f + \delta f \tag{5.1}$$

where α is the ratio of the frame rates and δf is the offset between the 0th frame in

each sequence. In all cases we seek to recover δf to sub-frame accuracy and in some cases we also seek to recover α. In the case of non-rigid motions, we pose this search for temporal alignment as a search for consistent structure between the two sequences.

5.1.1 Related work

Our synchronization method is inspired by the work of Wolf and Zomet [123] who used rank constraints of a matrix of image measurements, as introduced by Tomasi and Kanade [111], to define its 'energy' above an expected rank bound. This energy is minimized when structure is most consistent (*i.e*. at corresponding frames), such that synchronization is recovered to the nearest single frame. We develop this method to recover synchronization to *sub-frame* accuracy for sequences of *unknown and differing frame rates*.

The spatiotemporal alignment of image sequences has also been notably studied by Caspi and Irani [22, 23, 24]. In earlier work, they use optical flow to recover the synchronization under the assumption that temporally corresponding frames are related by a homography [22] or that the cameras have approximately coincident centres of projection [23]. However, our work is more closely related to their feature-based methods for recovering synchronization [24] where they consider wide baseline stereo with temporal correspondence only. Forming putative matches between feature *tracks* and utilizing a voting scheme (RanSaC), they compute both the temporal and spatial relationship (the fundamental matrix) between the sequences, iteratively optimizing over spatiotemporal transformation parameters using the geometric distance between points and their associated epipolar lines in the manner suggested by Reid and Zisserman [85].

Pooley et al [81] also use a perspective projection model but assume that a sufficient

number of matched background features are visible in each frame pair to compute the epipolar geometry of the two cameras. Potential frame correspondences are identified using the same error metric as [24, 85] for each frame pair (using known spatial correspondences) and synchronization parameters are estimated using the Hough transform.

Zhou and Tao [132] assume that features exhibit a linear trajectory over small periods of time. The epipolar geometry of the cameras is then used to transfer features from one view to the other for two consecutive frames and the cross ratio of the four points used to estimate the temporal offset (the frame rates of the cameras are assumed to be approximately equal). Having computed the offset, stereo algorithms are applied for depth recovery of the scene at each frame. However, the authors note that the feature locations must be estimated to sub-pixel accuracy, suggesting their algorithm is highly sensitive to noise.

5.1.2 Contributions

This chapter presents work that advances the state-of-the-art in two ways:

- Rank constraints are developed for the homography and perspective projection model, using an *algebraic* rather than geometric distance measure. Moreover, we demonstrate that in the affine case this general solution reduces to the linear formulation presented by Tomasi and Kanade [111].

- The rank constraints are employed in an algorithm that recovers sychronization of sequences of *different frame rates* to *sub-frame* accuracy. This is evaluated on synthetic sequences and demonstrated on real examples.

5.2 Generalized rank constraints

The basic idea underpinning our approach is simply stated – if the motion being observed is non-rigid, a metric that measures the rigidity of the scene using both cameras will assign a low cost to frames that are temporally aligned and a high cost to those that are not. We investigate such a metric for pairs of frames related by a homography, the fundamental matrix and the affine fundamental matrix. For further details on multiple view geometry, we direct the reader to [48].

5.2.1 Homography model

The case of recovering synchronization for sequences related by a homography was notably studied by Caspi and Irani using optical flow methods [22, 23]. In contrast, we consider the case where two cameras observe *point* features moving independently in a plane. Under this model, corresponding homogeneous image features, $\mathbf{x}$ and $\mathbf{x}'$, are related by a homography, $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \tag{5.2}$$

such that:

$$\mathbf{H}\mathbf{x} = \mathbf{x}' \tag{5.3}$$

$$\Rightarrow [\mathbf{x}'_\times]\mathbf{H}\mathbf{x} = [\mathbf{x}'_\times]\mathbf{x}' = \mathbf{0} \tag{5.4}$$

where $[\mathbf{x}'_\times]$ is the matrix form of the cross product such that $[\mathbf{x}'_\times]\mathbf{y} = \mathbf{x} \times \mathbf{y}$.

The constraints imposed by all points define a linear system such that under ideal conditions:

$$\mathbf{M}_H\mathbf{h} = \mathbf{0} \tag{5.5}$$

where $\mathbf{M}_H$ is a $2N\times 9$ matrix of constraints defined by the image feature locations and $\mathbf{h} = (h_1, ..., h_8, 1)^T$ is the vector of elements of $\mathbf{H}$. It can be shown that, for a given $\mathbf{H}$ (or $\mathbf{h}$), the sum of squared *algebraic* distances, $d_{alg}(\cdot,\cdot)$, between features $\mathbf{x}'_i$ measured in a frame from sequence 2 and those transferred, $\mathbf{H}\mathbf{x}_i$, from a frame in sequence 1 are related to $\mathbf{M}_H$ by:

$$\sum_i d_{alg}(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2 = \|\mathbf{M}_H\mathbf{h}\|^2. \tag{5.6}$$

Therefore, linear least squares methods can be employed to minimize d_{alg} for a given pair of frames. For $N \leq 4$ points, any $\mathbf{h}$ in the right nullspace of $\mathbf{M}_H$ satisfies (5.5) exactly. For $N > 4$ points, d_{alg} is minimized by setting $\mathbf{h}$ to the right singular vector corresponding to the ninth singular value, σ_9, of $\mathbf{M}_H$ and rescaling appropriately. In this case, it can be shown that:

$$\sum_i d_{alg}(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2 = \sigma_9^2. \tag{5.7}$$

This suggests a 'rank constraint' framework for synchronizing sequences whereby a small value of σ_9^2 indicates the correct alignment of a pair of frames.

5.2.2 Perspective model

In the perspective projection case, studied by Caspi *et al.* [24] for feature-based methods with a geometric distance measure, we again propose using the *algebraic* distance measure in a rank-constraint framework as a computationally cheap alternative. Cor-

responding homogeneous image features, $\mathbf{x}$ and $\mathbf{x}'$, are related by the perspective fundamental matrix, $\mathbf{F}$:

$$\mathbf{F} = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & 1 \end{bmatrix} \tag{5.8}$$

such that:

$$\mathbf{x}^T \mathbf{F} \mathbf{x}' = 0. \tag{5.9}$$

Similar to the homography case, point correspondences define a linear system such that:

$$\mathbf{M}_F \mathbf{f} = \mathbf{0} \tag{5.10}$$

where $\mathbf{M}_F$ is a $N \times 9$ matrix of constraints defined by the image feature locations and $\mathbf{f} = (f_1, ..., f_8, 1)^T$ is the vector of elements of $\mathbf{F}$. It can also be shown that, for a given $\mathbf{F}$ (or $\mathbf{f}$), the sum of squared *algebraic* distances, $d_{alg}(\cdot, \cdot)$, between features $\mathbf{x}'_i$ from a frame in sequence 2 and their epipolar lines, $\mathbf{F}\mathbf{x}_i$, as computed from the corresponding features in sequence 1 are related to $\mathbf{M}_F$ by:

$$\sum_i d_{alg}(\mathbf{x}'_i, \mathbf{F}\mathbf{x}_i)^2 = \|\mathbf{M}_F \mathbf{f}\|^2. \tag{5.11}$$

Again, linear least squares methods can be employed to minimize d_{alg} for a given pair of frames. For $N \leq 8$ points, any $\mathbf{f}$ in the right nullspace of $\mathbf{M}_F$ satisfies (5.10) exactly. For $N > 8$ points, d_{alg} is minimized by setting $\mathbf{f}$ to the right singular vector corresponding to the ninth singular value, σ_9, of $\mathbf{M}_F$ and rescaling appropriately (the familiar 'eight point algorithm' [47]).

Similarly, it can be shown that:

$$\sum_i d_{alg}(\mathbf{x}'_i, \mathbf{F}\mathbf{x}_i)^2 = \sigma_9^2, \tag{5.12}$$

again suggesting that a rank constraint framework may be applicable albeit at a cost of requiring twice as many points as the homography model.

5.2.3 Affine model

We now turn to the simpler case of affine projection, a commonly used projection model in human motion analysis applications since the human body has limited depth and perspective effects are typically small. In the affine case, the fundamental matrix takes the simpler form:

$$\mathbf{F}_A = \begin{bmatrix} 0 & 0 & a_1 \\ 0 & 0 & a_2 \\ a_3 & a_4 & 1 \end{bmatrix} \tag{5.13}$$

and again:

$$\mathbf{M}_A\mathbf{a} = \mathbf{0} \tag{5.14}$$

where $\mathbf{a} = (a_1, ..., a_4, 1)^T$. However, in this case the $N \times 5$ constraint matrix, $\mathbf{M}_A$, takes the particularly simple form:

$$\mathbf{M}_A = \begin{bmatrix} x_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N & y_N & x'_N & y'_N & 1 \end{bmatrix} \tag{5.15}$$

where $(x_n, y_n)^T$ and $(x'_n, y'_n)^T$ denote the nth feature in the first and second view, respectively. As in the other projection models, linear least squares are employed such that $N = 4$ provides an exact solution whereas for $N > 4$ points, setting $\mathbf{a}$ equal to the

right singular vector corresponding to σ_5 minimizes the algebraic distance between the point sets.

However, it can be shown that *normalizing* $\mathbf{M}_A$ with respect to its row mean gives a new matrix, $\widetilde{\mathbf{M}}_A$ with a tighter lower bound on $rank(\widetilde{\mathbf{M}}_A)$. Such normalization can be interpreted as a translation of the points such that their centroid lies at the origin of the image. Under these conditions:

$$\widetilde{\mathbf{M}}_A\tilde{\mathbf{a}} = \begin{bmatrix} x_1-\bar{x} & y_1-\bar{y} & x'_1-\bar{x}' & y'_1-\bar{y}' \\ \vdots & \vdots & \vdots & \vdots \\ x_N-\bar{x} & y_N-\bar{y} & x'_N-\bar{x}' & y'_N-\bar{y}' \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \mathbf{0} \tag{5.16}$$

such that $rank(\widetilde{\mathbf{M}}_A) \leq 3$.

5.2.4 Factorization approach

In proposing the Factorization method [111], Tomasi and Kanade arrived at the same conclusion by different reasoning. For two affine views, their observation shows that the normalized $4\times N$ 'measurement matrix' of image coordinates, $\mathbf{W}$, can be written as a product:

$$\mathbf{W} = \begin{bmatrix} x_1-\bar{x} & \cdots & x_N-\bar{x} \\ y_1-\bar{y} & \cdots & y_N-\bar{y} \\ x'_1-\bar{x}' & \cdots & x'_N-\bar{x}' \\ y'_1-\bar{y}' & \cdots & y'_N-\bar{y}' \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_N \end{bmatrix} = \mathbf{PX} \tag{5.17}$$

where $\mathbf{P}_i$ is the 2×3 projection matrix of the ith view and $\mathbf{X}_n$ is the 3×1 vector of inhomogeneous 3D coordinates of the nth feature. Specifically, (5.17) shows that the rank of $\mathbf{W}$ is bounded above by 3 since it is a product of the 4×3 projection matrix $\mathbf{P}$ and $3\times N$ structure matrix, $\mathbf{X}$. Note that for the two view case $\mathbf{W} = \widetilde{\mathbf{M}}_A^T$, thus confirming the rank constraints derived in the previous section.

However, in contrast to using the affine fundamental matrix, the factorization method

naturally extends to any number of views. Tomasi and Kanade exploited this fact to propose the factorization of $\mathbf{W}$ into affine motion and structure using the Singular Value Decomposition (SVD), thus recovering all $\mathbf{P}_i$ and $\mathbf{X}_n$ up to an affine transformation. Reid and Murray [84] later demonstrated that the Factorization method recovers the 'optimal' structure and motion in terms of minimizing reprojection error and can therefore be interpreted as a Maximum Likelihood estimate, assuming isotropic Gaussian noise.

It can also be shown that the sum of squared *geometric* reprojection error, E, following factorization is directly related to the singular values of the rank r matrix, $\mathbf{W}$, by:

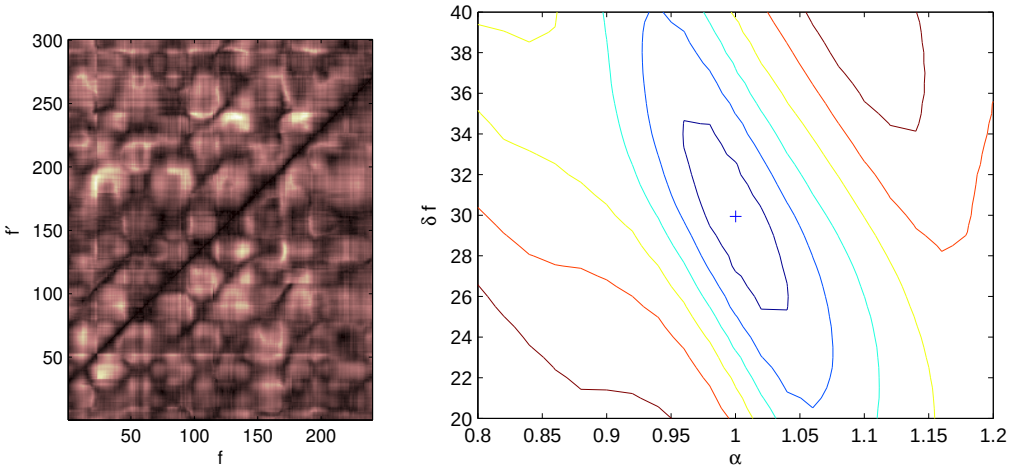
$$E = \|\mathbf{W} - \mathbf{PX}\|_F^2 = \sum_{i=4}^{r} \sigma_i^2 \tag{5.18}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. In the two view case, this reduces to $E = \sigma_4^2 = d_{alg}$ which agrees with the known property that geometric and algebraic distances are identical for the affine projection model.

5.3 Rank-based synchronization

Intuitively this measure would seem to be an appropriate metric for determining synchrony since when the frames are temporally aligned, the image correspondences are consistent with an underlying interpretation of three-dimensional structure (the pose of the person at that instant) and reprojection error is small. However, when the sequences are not aligned the images are of *different* points in space and therefore not subject to any rank constraint.

Using the results derived so far, we propose two cost functions in order to recover



(a) (b)

Figure 5.2: (a) The cost surface $C_1(f, f')$ for the real running example, shown in plan view and normalized such that values range from 0 (dark) to 1 (light). Note the visible 'channel' close to the principal diagonal where the true correspondence lies. (b) Contour plot of $C_2(\alpha, \delta f)$, also indicating the solution recovered via non-linear optimization. From the elliptic shape of the basin of attraction, we see that errors in α may be compensated by a complementary error in δf.

the synchronization between two sequences. The first match cost, $C_1(f, f')$, reflects the residual reprojection error resulting from the pairing of two frames, f and f':

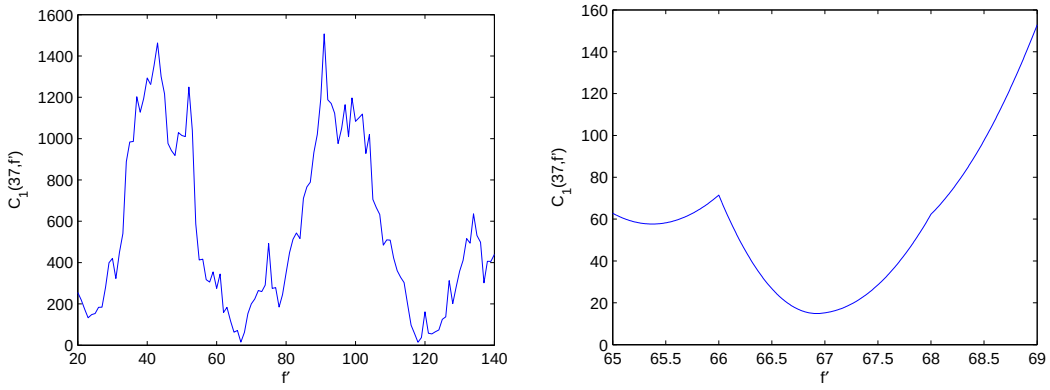
$$C_1(f, f') = \sum_{i=4}^{r} \sigma_i^2 = \sigma_4^2 \tag{5.19}$$

where σ_4 is the fourth singular value of $\mathbf{W}(f, f')$, defined as:

$$\mathbf{W}(f, f') = \begin{bmatrix} \mathbf{x}_1^f & \cdots & \mathbf{x}_N^f \\ \mathbf{x}_1^{f'} & \cdots & \mathbf{x}_N^{f'} \end{bmatrix} \tag{5.20}$$

and $\mathbf{x}_n^f$ and $\mathbf{x}_n^{f'}$ are the normalized image co-ordinates of the nth feature in frame f and f' of sequences 1 and 2, respectively. Pairs of frames with a low value of $C_1(f, f')$ are a good match whereas those with a high value of $C_1(f, f')$ are structurally inconsistent. This is apparent in Figure 5.2a showing a plan view of the cost function, $C_1(f, f')$.

Having defined a match cost between frames from two different sequences, we then



(a) (b)

Figure 5.3: (a) Plot of $C_1(37, f')$ for the running sequence. Note that in addition to the correct minimum (frame 67, in this case) another minimum is evident (frame 118) due to periodic motion (also noted by [123]). (b) $C_1(37, f')$ evaluated using interpolated feature locations in the interval [65,69]. The computed minimum is observed close to the correct minimum ($f' = 67$).

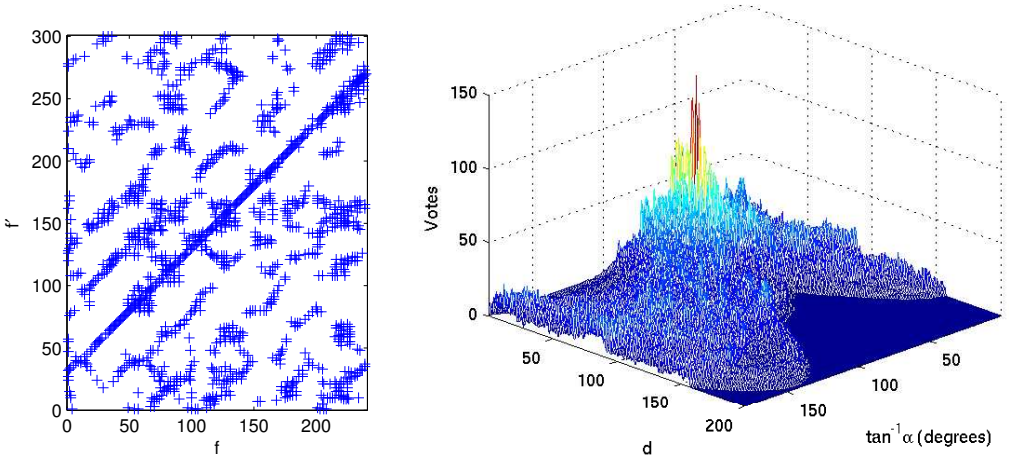
define a cost function for the synchronization parameters, α and δf. The most intuitive is simply the sum of reprojection errors over the entire sequence such that:

$$C_2(\alpha, \delta f) = \sum_f C_1(f, \alpha f + \delta f). \tag{5.21}$$

This defines a cost surface (shown in Figure 5.2b) upon which we find a local minimum via non-linear optimization based on a sensible initial estimate, as described in the following section.

5.4 Method

For every frame, f, in sequence 1 we compute the match cost for every potentially corresponding frame, f', in sequence 2. Figure 5.3a shows $C_1(37, f')$, a 'slice' through the cost function at frame 37 of sequence 1. In this example, we see that multiple minima are present due to periodic motion in the action being performed (a running motion in this case).



(a) (b)

Figure 5.4: (a) Local minima corresponding to potential frame correspondences, recovered using non-minimum suppression and thresholding of the cost surface shown in Figure 5.2a. Note the high number of good matches along the diagonal where the true correspondence lies. (b) 3D surface of the accumulator array with a visible peak at $(\alpha, \delta f) = (1, 32.76)$.

Exhaustively computing $C_1(f, f')$ for all pairings of f and f' generates a coarse 2D cost surface as shown in Figure 5.2a. Although this requires $F \times F'$ evaluations of C_1 for sequences of F and F' frames, the method is relatively efficient due to the simple form of the cost function.

From C_1, we select putative frame correspondences (Figure 5.4a) via thresholding and non-minimum suppression across f and f'. These potential frame correspondences cast votes in a Hough accumulator [9], a popular tool for line detection, from which it is straightforward to extract peaks corresponding to potential synchronization parameters. Since we expect there to be multiple peaks in ambiguous cases, we retain all peaks with a score greater than 90% of the maximum.

Since the recovered correspondences are between whole frames, the Hough transform returns estimates of potential alignment whose resolution is limited by the bin size. Moreover, if α is known to be unity then the accuracy of δf is theoretically

limited to the nearest whole frame. We therefore optimize the cost function $C_2(\alpha, \delta f)$ directly in order to recover α and δf to *sub-frame* accuracy. Since this requires the evaluation of $C_1(f, f')$ for real (*i.e.* non-integer) values of f', we use linear interpolation to determine approximate feature locations between frames (see Figure 5.3b). Linear interpolation was found to reduce reprojection errors compared with single frame accuracy although higher order interpolations (*e.g.* quadratic, cubic) may yield superior estimates.

Using the interpolated feature locations, we evaluate $C_2(\alpha, \delta f)$ for each of the selected $(\alpha, \delta f)$ pairs. Using the pair with the smallest error as an initial estimate, we then employ standard optimization methods (the Nelder-Mead Simplex algorithm, implemented as `fminsearch` in Matlab) to recover a locally optimal solution. Figure 5.2b illustrates the cost surface $C_2(\alpha, \delta f)$ together with the recovered minimum. We note that the amount of estimated overlap between the sequences may vary with α and δf such that not all frames in view 1 have a corresponding frame in view 2. Since this introduces first order discontinuities in the cost surface we assume complete overlap between the sequences, deferring the design of a more robust cost function for future work.

This process can be seen as a hierarchical search for the globally optimal solution, using computationally cheap methods (the Hough transform) to reject a high number of poor estimates early on so that relatively expensive processes, such as computing the cost $C_2(\alpha, \delta f)$, are performed for only a small number of hypotheses.

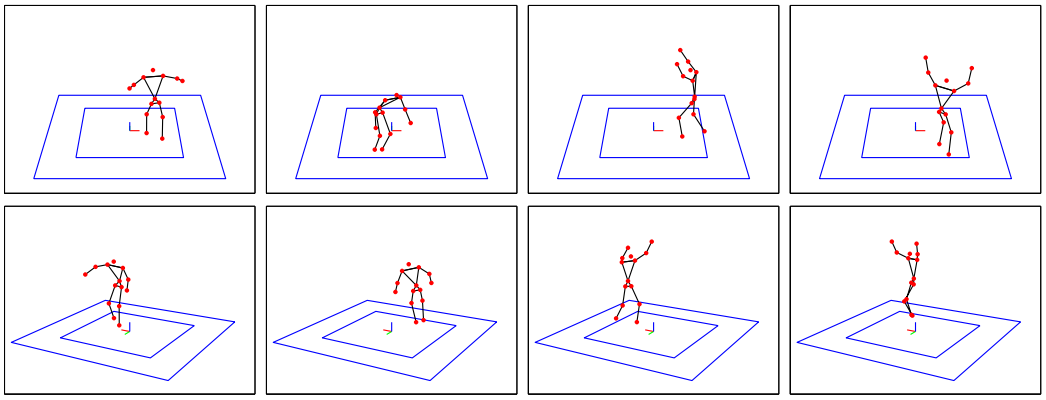


Figure 5.5: Synthetic 'monkey' sequence as seen from two wide baseline viewpoints. The red circles indicate point features used as inputs to the synchronization algorithm.

5.5 Results

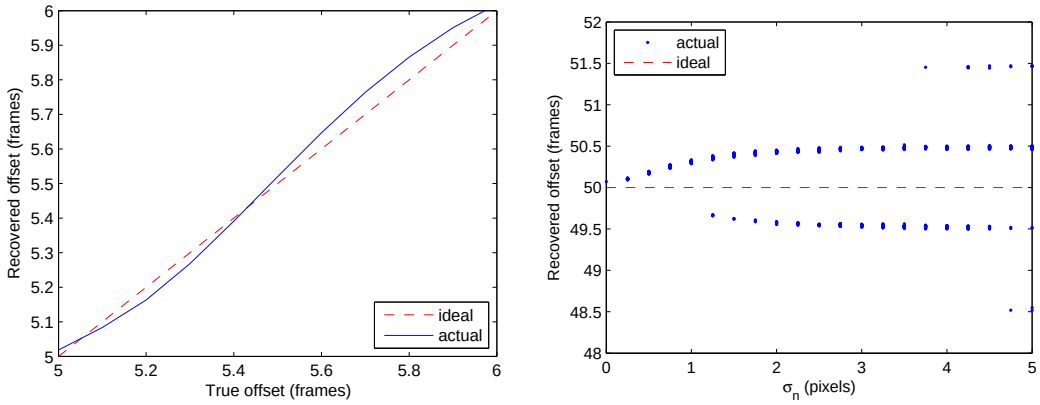
5.5.1 Monkey sequence

We demonstrate the algorithm on a synthetic sequence pair of a human impersonating a monkey (Figure 5.5). The views, synchronized by design, each contain 480 frames of 14 points features located at anatomical landmarks on the body (shoulders, elbows, wrists, hips, knees, ankles, midriff and head) imaged under perspective projection.

We then deleted 50 frames from the beginning and end of the first view to give a sequence pair with synchronization parameters $\delta f = 50$ and $\alpha = 1$. Unless specified, only the offset is recovered such that α could be fixed at unity, making the problem a one-dimensional search.

Performance over varying temporal offset

To demonstrate the accuracy of the algorithm for sub-frame offsets, unsynchronized sequences were synthesized by taking interleaved frames from the available synchronized sequences to generate pairs of sequences offset by $5, 5.1, \ldots, 5.9$ frames. Figure 5.6a compares the recovered offsets with ground truth where it can be seen that the



(a) (b)

Figure 5.6: (a) Recovered values for simulated offsets of $5, 5.1, \ldots, 5.9$ frames. We see the recovered offset is typically accurate to within hundredths of a frame. (b) Recovered offsets over 50 trials at each level of added zero-mean Gaussian noise of standard deviation, σ_n.

recovered offsets are typically accurate to within a few hundredths of a frame despite: (i) the assumption of linear motion between frames degrades for the low frame rates at which we are operating; (ii) lowering the effective frame rate reduces the number of frames available for estimation of the synchronization parameters.

Sensitivity to noise

The original image feature locations were perturbed by zero mean Gaussian noise of increasing standard deviation, σ_n pixels, for 50 tests at each level of noise. The scatter plot in Figure 5.6b shows the recovered offsets as a function of the level of noise. Interestingly, we see a tendency for the algorithm to recover offsets halfway between frames. This may indicate a preference to average out noise between consecutive frames.

Recovery of both α and δf

In the previous experiments, α was fixed at unity such that only δf was the only remaining parameter to be recovered. Under these constraints, the algorithm recovers an offset of $\delta f = 50.07$ frames – an excellent match for the ground truth offset of 50 frames. With α allowed to vary, the algorithm accurately recovered synchronization parameters of $\alpha = 1.0001$ and $\delta f = 50.05$. This suggests that small errors in offset may be compensated by a corresponding change in α. However, we remind the reader that the experiments were conducted on noiseless data such that the only error is as a result of perspective effects. In Section 5.6.3, we demonstrate the synchronization of sequences of different frame rates using NTSC and PAL cameras.

Reprojection errors

We now demonstrate how recovering sub-frame accurate synchrony reduces the error between the measured feature locations and computed feature locations. To quantify reprojection errors, we used odd frames from the first view and even frames from the other, resulting in parameters $\alpha = 1$ and $\delta f = 25.5$. With α constrained at unity, the Hough transform recovered an initial estimate of $\delta f = 26$. This was refined further using interpolation to an estimate of $\delta f = 25.53$.

For each frame, we computed four sets of feature locations for the second camera: features taken directly from the nearest frame of sub-sampled data ('Nearest'); interpolated features using recovered synchronization parameters ('Recovered'); interpolated features using known synchronization parameters ('Known'); features taken directly from *original* image data ('Original'). Since these feature locations are typically of full rank (*i.e.* not subject to the rank constraint), we also compute a reduced-rank ver-

	Full rank	Reduced rank
Nearest	$\mathbf{9.6204}$	$\mathbf{9.6878}$
Recovered	$\mathbf{1.7728}$	$\mathbf{2.9072}$
Known	1.6334	2.8266
Original	0	2.2990

Table 5.1: Reprojection errors demonstrating a considerable reduction using sub-frame accurate alignment.

sion that satisfies the rank constraints by projecting onto the appropriate subspace. For each set of computed features at every frame, $\mathbf{W}_{est}$, we then compute the sum of squared reprojection errors with respect to the original image data, $\mathbf{W}$:

$$E_{est} = \|\mathbf{W} - \mathbf{W}_{est}\|_F^2 \tag{5.22}$$

Table 5.1 shows the mean E_{est} over all frames, showing that sub-frame accuracy offers a considerable reduction in reprojection error compared with using the nearest frame.

We note that the benefit of interpolating feature locations is dependent on the speed of the motion with respect to the camera frame rate. For a slow movement (or high frame rate), the motion between consecutive frames is small such that there will be little benefit in interpolation and the nearest frame will suffice. However, for fast movements (or low frame rates) the motion between frames is higher such that interpolation is beneficial, although in such cases motion blur may introduce additional uncertainty in the projected joint locations.

One application where interpolating feature locations is particularly beneficial arises for sequences of different frame rates. In this case, generating synchronized sequences from uninterpolated data results in the nearest frame results in frames being skipped (in the slower sequence) or duplicated (in the faster sequence). For example, when

synchronizing PAL and NTSC sequences (25Hz and 30Hz, respectively) using the nearest frame duplicates every fifth frame of the PAL sequence in order to maintain temporal consistency, resulting in 'jerky' motion of the feature locations. In contrast, interpolating feature locations smoothes out these discontinuities resulting in a more agreeable motion.

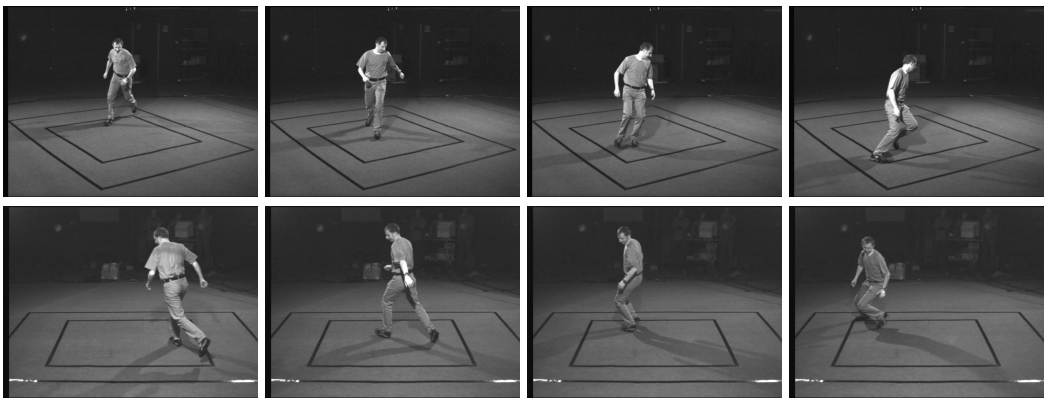


Figure 5.7: Running sequence as seen from two wide baseline viewpoints.

5.6 Real examples

5.6.1 Running sequence

We continue with a real running sequence (Figure 5.7) captured using two calibrated cameras, hardware-synchronized at 60Hz, for a quantitative ground truth comparison of recovered synchronization parameters. The sequences were then offset by 30 frames, giving ground truth values of $\alpha = 1$ and $\delta f = 30$. The locations of 13 joints (shoulders, elbows, wrists, hips, knees, ankles and midriff) were hand-labelled in each frame of the sequences.

With α constrained at its known value of 1, an offset of $\delta f = 29.96$ was recovered by the algorithm, compared with its true value $\delta f = 30$. Allowing α to vary recovered values of $\alpha = 1.0002$ and $\delta f = 29.94$. Plots related to this sequence are shown in Figures 5.2, 5.3 and 5.4.

5.6.2 Handstand sequence

The algorithm relies upon the motion of the subject being non-rigid, otherwise *all* frames are consistent throughout the sequence and the method is not valid. Rigid motion of the body may manifest itself for certain actions where the body assumes an

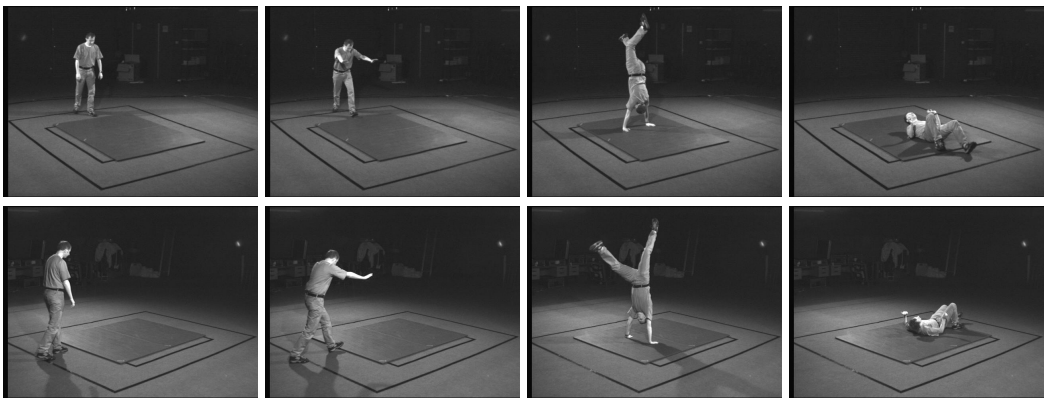
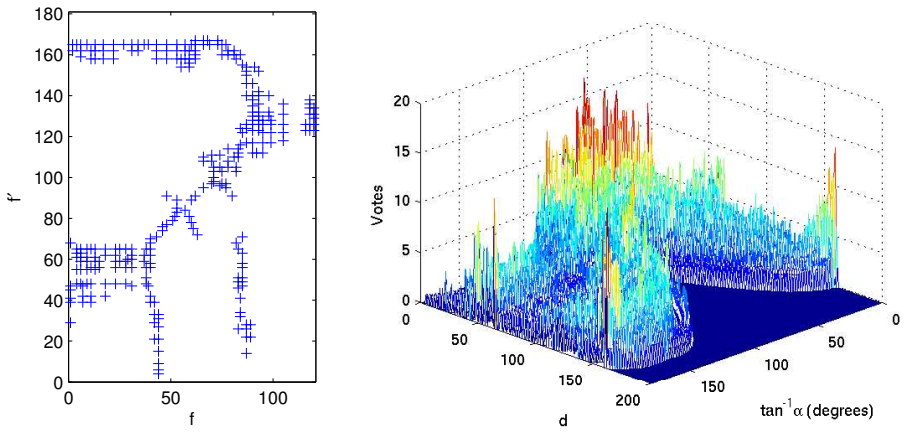


Figure 5.8: Handstand sequence as seen from two wide baseline viewpoints.



(a) (b)

Figure 5.9: (a) Recovered correspondences for the handstand sequence and (b) the corresponding Hough accumulator. Compared with Figure 5.4, we see no single dominant peak and considerable support for outlying alignment estimates.

approximately fixed pose for extended periods of time. We show this to be the case for a handstand sequence of 180 frames (Figure 5.8), also captured using synchronized cameras and manually offset by 30 frames.

Figure 5.9a shows the putative frame correspondences recovered by the algorithm where the underlying linear relationship is apparent only for a short period during the middle of the sequence (when the legs undergo a 'scissors' motion). We also observe blocks of corresponding frames suggesting that structure was consistent for extended

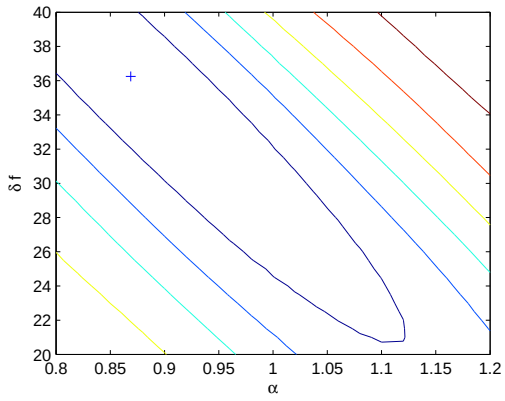


Figure 5.10: Contour plot of $C_2(\alpha, \delta f)$ for the handstand sequence. It can be seen that the cost surface is relatively flat compared with Figure 5.2b for the running sequence. Furthermore, the minimum of the cost surface appears to be some distance from the true value ($\alpha = 1$, $\delta f = 30$)

intervals of time (*i.e.* rigid). Figure 5.9b shows the corresponding Hough accumulator where we observe a cluster of peaks around the correct parameters and many outlying peaks corresponding to spurious estimates.

Despite this, after evaluating $C_2(\alpha, \delta f)$ for selected peaks, initial estimates of $\alpha = 1$ and $\delta f = 29.22$ were selected. However, blind refinement of the parameters using non-linear optimization led to a divergence of the estimate from the correct solution, instead converging to $\alpha = 0.8671$ and $\delta f = 36.29$.

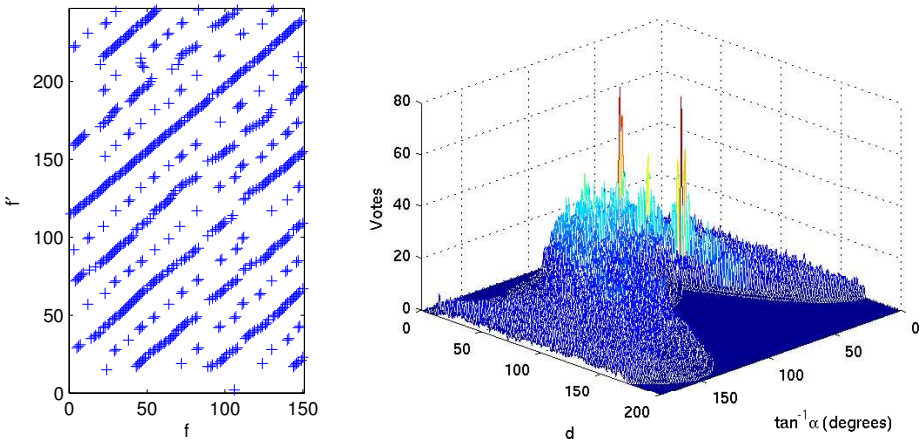
Figure 5.10 shows the cost surface, $C(\alpha, \delta f)$, where the local minimum is located some distance from the correct solution. The cost surface is relatively flat, compared with Figure 5.2b for the running sequence of identical synchronization parameters, due to the areas of low cost at the extremes of the sequence where the body was almost rigid. As a result, an increase of the cost due to variation in α is compensated by varying δf.

Figure 5.11: Juggling sequence as seen from two wide baseline viewpoints.

5.6.3 Juggling sequence

For our final sequence using the affine camera model, we demonstrate the method on a juggling sequence (Figure 5.11) captured using two wide baseline cameras that were neither synchronized nor calibrated. In particular, one sequence was captured using an NTSC digital camera and consisted of 150 colour frames at 30Hz with a resolution of 320×240 pixels. The other sequence, captured with a PAL analogue camera, contained 250 greyscale frames at 25Hz with a resolution of 720×576 pixels. Corresponding feature locations on the upper body, head and juggling balls were again marked manually.

Figure 5.12a shows the recovered frame correspondences where we observe several distinct parallel bands due to the periodicity of the juggling motion. These are observed as multiple peaks in the Hough accumulator shown in Figure 5.12b. From the known frame rates, we computed $\alpha = 25/30 \approx 0.833$ and estimated that $\delta f \approx 115$ by inspection. The recovered values of $\alpha = 0.8371$ and $\delta f = 113.60$ are close agreement with these estimates.



(a) (b)

Figure 5.12: (a) Recovered correspondences for the juggling sequences and (b) the corresponding Hough accumulator. Note the presence of multiple peaks in the accumulator array due to the periodicity of the juggling motion.

5.6.4 'Pins' sequence

To finish, we briefly demonstrate the homography model approach using a sequence pair of point features moving independently in a plane, captured using two cameras at approximately 12.5Hz and 8Hz. The sequences, shown in Figure 5.13, capture map pins moving on a flat surface under the influence of a desk fan. A crude feature tracker was then implemented to recover feature tracks automatically. Although many tracks were corrupted by noise and tracking error, thirteen clean tracks were matched by hand.

The recovered frame correspondences and corresponding Hough accumulator are shown in Figure 5.14 where it can be seen that there are very few spurious minima. The cluster of minima in the lower left corner of Figure 5.14a correspond to the beginning of the sequence, where the pins were static, such that structure was inherently 'consistent'. The true synchronization parameter values were estimated, from the known frame rates and by inspection, as $\alpha \approx 0.64$ and $\delta f \approx 16$. These values correspond closely to the recovered values of $\alpha = 0.6118$ and $\delta f = 13.50$, demonstrating the

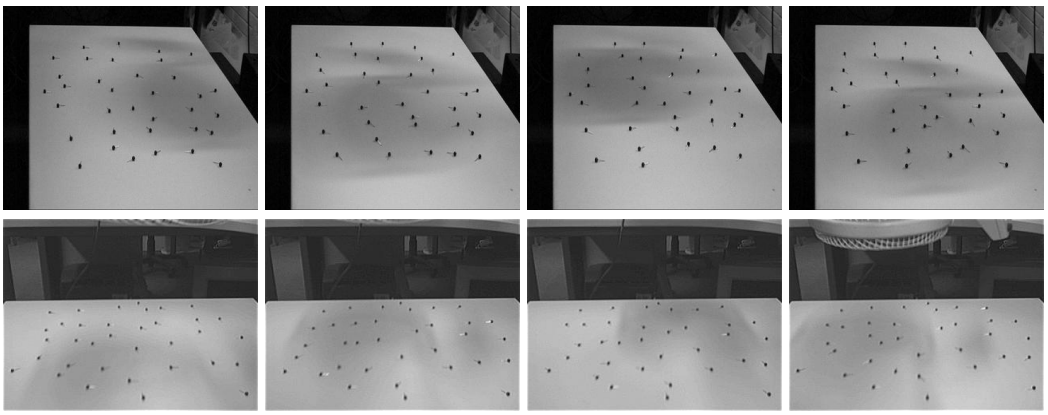


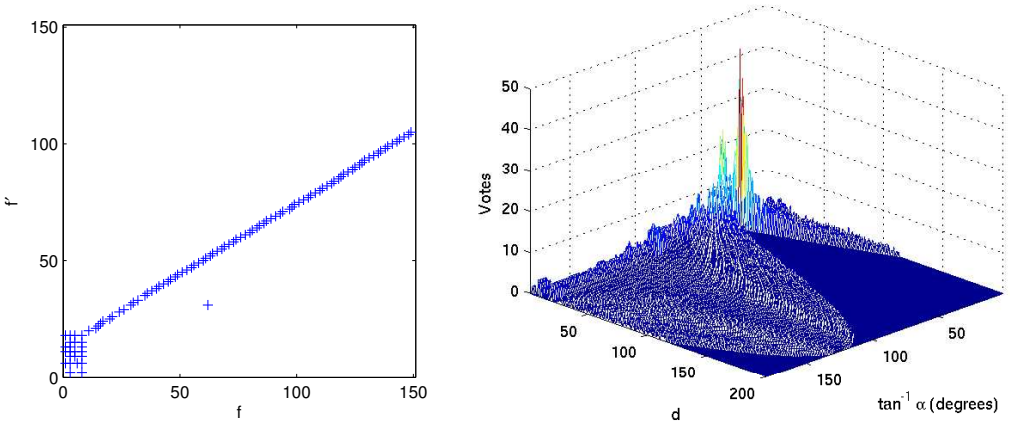
Figure 5.13: Pins sequence as seen from two wide baseline viewpoints.

effectiveness of the method.

There are several explanations for the high performance on this sequence. Firstly, we note that the uncertainty is much smaller for the pins since they are surface features and can be tracked with high accuracy, in contrast to human joint locations that are hidden beneath muscle tissue. Secondly, the pins were known to move in a plane such that our assumption of corresponding frames being related by a homography was correct, unlike the affine case where perspective effects introduced error into the system. Finally, each point feature provides two constraints for cameras related by a homography compared with one constraint each for affine and perspective projection models.

5.7 Summary

This chapter has presented a method of synchronizing two sequences from the projected locations of anatomical landmarks on the human body. Error metrics to indicate synchrony were derived for the homography, perspective and affine camera models. Furthermore, it was shown that the rank constraints employed by Tomasi and Kanade in the Factorization method form a natural extension of those derived from the affine fundamental matrix. These error metrics were used to synchronize sequences of un-



(a) (b)

Figure 5.14: (a) Recovered frame correspondences and (b) corresponding Hough accumulator with dominant peak. Note that the correspondences and resulting Hough accumulator are considerably more 'clean' than in other cases.

known and different frame rates, as demonstrated on synthetic and real sequences.

5.7.1 Future work

Synchronizing multiple sequences

Since the Factorization method extends rank constraints to more than two views, it is straightforward to extend the synchronization to multiple sequences. Due to the (albeit linear) increase in dimensionality of the parameter space, it would be sensible to synchronize all other sequences idependently with respect to a reference sequence in order to recover an initial estimate of synchronization parameters. This estimate could then be refined via non-linear optimization as in the two view case.

Multiple hypotheses

Most robust human trackers output multiple hypotheses (or a p.d.f. over pose) rather than a single point estimate at each frame. Therefore, any synchronization algorithm should accommodate this feature. A possible solution is to compute the *distribution* of

error at each frame and assign a score according to the sharpness of the peak at zero.

Using line and plane features

The Factorization algorithm has been extended to exploit other image features such as lines and planes [75]. Such features may be exploited to improve the estimation of synchronization parameters. However, more than two views are required to exploit line features due to the limited constraints they provide and planar structure is rare in human motion sequences.

Chapter 6

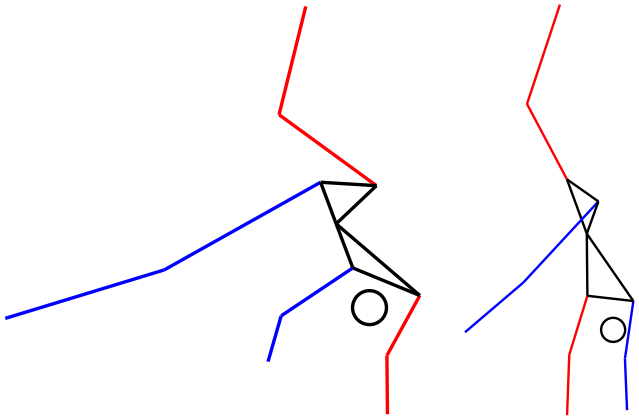
Self-Calibrated Stereo from Human Motion

In this chapter, we develop a method for the self-calibration of human motion observed by two cameras. Since only two views are available of the (time-varying) structure at each instant, constraints on the projection matrices alone are insufficient. We therefore impose symmetry and piecewise-rigidity constraints on the known structure (the human body) to recover the calibration of the two cameras. In particular, we present a novel parameterization of the system that admits a closed form initialization for optimization of the cost surface. Due to the three-fold reduction in the number of parameters, optimization is better behaved and considerably more efficient without sacrificing accuracy. We then perform bundle adjustment over the free parameters to recover the maximum likelihood solution for structure and motion. The method is demonstrated on motion captured data (for quantitative analysis) and real examples.

6.1 Introduction

So far, we have recovered joint locations of an articulated structure in an image sequence (Chapters 3 and 4) and shown that joint locations can be used to align a stereo sequence pair in time (Chapter 5). At this time, we are able to recover the 'skeleton' of the subject by factorization but in an affine co-ordinate frame. However, since lengths and joint angles can only be measured in a Euclidean co-ordinate frame, we must calibrate the cameras accordingly.[1] In this way, structure and motion are 'upgraded' to a

[1] Portions of this chapter were published in [115]



(a) (b)

Figure 6.1: Schematic of self-calibration: (a) Affine structure; (b) Euclidean structure. Note that in the Euclidean frame, the body is in the correct proportion in contrast to that in the affine frame.

Euclidean co-ordinate frame, as shown schematically in Figure 6.1.

The standard approach to self-calibration is to apply constraints to the projection matrices, such as fixed lens parameters or the slightly weaker requirement that the imaging system has zero skew and/or unit aspect ratio. However, such methods rely on there being multiple cameras (or a single moving camera providing multiple views) so that the system is overconstrained. Such constraints were proposed by Tomasi and Kanade [111] for orthographic projection (where at least three views are required) and later generalized to all parallel projection models by Quan [82].

In this chapter, we use binocular sequences of human motion to recover instantaneous affine structure and motion by factorizing independently at each time instant. Although we have many more than three images in each sequence, structure differs at each time instant and self-calibration is underconstrained using constraints on the projection matrices alone. Furthermore, simple engineering solutions (*e.g.* using a third view) are not always applicable, such as when reconstructing from sporting or surveillance footage.

6.1.1 Related work

To address this problem, we exploit the fact that additional *structural* constraints are available in human motion analysis. Taylor [107] showed that knowing the *ratios* of lengths was sufficient to recover scene structure (up to some depth ambiguities) for a single image although our work is more directly inspired by the method of Liebowitz and Carlsson [66] who enforce the symmetry and piecewise rigidity of the human body. They recover affine structure up to a rectifying transformation at each frame and optimize over the free parameters under *weak* motion and structural constraints. Although projection and symmetry constraints alone are sufficient for self-calibration at each instant, rigidity constraints (that apply at different instants) account for scale changes, due to perspective, over time.

6.1.2 Contributions

Our method employs the same principles as [66] but overcomes a number of practical difficulties. Specifically:

- We propose a reduced parameterization of the system that implicitly enforces the required conditions for self-calibration. We show that our parameterization is considerably more efficient and better behaved during optimization, for which we are guaranteed an intuitive initialization in closed form.

- Having recovered an initial estimate, we refine this further by applying a bundle adjustment over all parameters that correctly minimizes a geometric reprojection error in the image, thus recovering the maximum likelihood solution.

6.2 Self-Calibration

We begin by reviewing the camera calibration, described in Section 2.2.1, in greater detail. Given two sequences of image features, we can recover structure and motion by factorization [111] independently at each time instant, i. With some abuse of notation, we define $\mathbf{P}_i$ as the 4×3 normalized (with respect to translation) projection matrix at time i and $\mathbf{X}_i$ as the structure matrix at time i. It can be shown that, at each instant i, structure is known only up to an unknown affine transformation, $\mathbf{G}_i$:

$$\mathbf{W}_i = \mathbf{P}_i\mathbf{X}_i = \mathbf{P}_i\mathbf{G}_i^{-1}\mathbf{G}_i\mathbf{X}_i \tag{6.1}$$

where each $\mathbf{G}_i$ is an invertible, homogeneous 3×3 matrix that can be factorized by QR-decomposition ($\mathbf{G}_i \rightarrow \mathbf{Q}_i\mathbf{B}_i$) into a 3D rotation, $\mathbf{Q}_i$, and an upper-triangular matrix, $\mathbf{B}_i$. Since $\mathbf{Q}_i$ effects a change of Euclidean coordinate frame *after calibration* it can be discarded without loss of generality. Consequently, as each $\mathbf{B}_i$ has six independent, non-zero elements a sequence of F frames has $6F-1$ degrees of freedom, up to a global scale factor.

We define $\mathbf{\Omega}_i = \mathbf{B}_i^T\mathbf{B}_i$ such that $\mathbf{B}_i$ is recovered from $\mathbf{\Omega}_i$ by Cholesky factorization *if and only if* $\mathbf{\Omega}_i$ *is positive definite*. Eigen-decomposition of $\mathbf{\Omega}_i = \mathbf{V}_i\mathbf{D}_i\mathbf{V}_i^T$ such that $\mathbf{B}_i = \mathbf{D}_i^{1/2}\mathbf{V}_i^T$ explains the action of $\mathbf{B}_i$ geometrically as a rotation into a new coordinate frame, followed by an anisotropic scaling.

6.2.1 Motion constraints

To recover the required set of all $\mathbf{B}_i$ that transforms each affine reconstruction into Euclidean space, constraints are applied to all projection matrices, $\mathbf{P}_i$, in a form of self-calibration [111, 82]. Specifically, for a given $\mathbf{B}$ the axes, $\mathbf{i}^T$ and $\mathbf{j}^T$, of an affine

projection matrix transform to $\mathbf{i}^T\mathbf{B}^{-1}$ and $\mathbf{j}^T\mathbf{B}^{-1}$ where the skew, r_{skw}, and difference in length, r_{asp}, are given by:

$$\begin{aligned} r_{skw} &= \mathbf{i}^T\mathbf{B}^{-1}\mathbf{B}^{-T}\mathbf{j} \\ &= \mathbf{i}^T\boldsymbol{\Omega}^{-1}\mathbf{j} \qquad (6.2) \\ r_{asp} &= \mathbf{i}^T\mathbf{B}^{-1}\mathbf{B}^{-T}\mathbf{i} - \mathbf{j}^T\mathbf{B}^{-1}\mathbf{B}^{-T}\mathbf{j} \\ &= \mathbf{i}^T\boldsymbol{\Omega}^{-1}\mathbf{i} - \mathbf{j}^T\boldsymbol{\Omega}^{-1}\mathbf{j}. \qquad (6.3) \end{aligned}$$

Under most circumstances, it is sensible to impose constraints that the vectors $\mathbf{i}^T\mathbf{B}^{-1}$ and $\mathbf{j}^T\mathbf{B}^{-1}$ be orthogonal and have unit aspect ratio (*i.e.* $r_{skw} = r_{asp} = 0$). As a result, at a given instant, i, three or more views of the subject provide at least six linear constraints on $\mathbf{B}_i^{-1}\mathbf{B}_i^{-T} = \boldsymbol{\Omega}_i^{-1}$ and a linear least squares solution for $\boldsymbol{\Omega}_i^{-1}$ minimizes r_{skw} and r_{asp}. However, for only two views there are insufficient constraints on $\boldsymbol{\Omega}_i^{-1}$ and an infinite number of solutions exist.

6.2.2 Structural constraints

It has been shown [66, 107] that using knowledge of the human body imposes further constraints on reconconstructions. Figure 6.2 shows the four *symmetry* constraints (solid arrows) between the arms and legs and nine *rigidity* constraints (dashed arrows) on the left/right upper arm, forearm, thigh and foreleg, and hips, as suggested by Liebowitz and Carlsson [66].

More formally, two 3D vectors, $\mathbf{X}_{i,p}$ and $\mathbf{X}_{i,q}$, representing *different* links in the *same* affine reconstruction, i, transform to $\mathbf{B}_i\mathbf{X}_{i,p}$ and $\mathbf{B}_i\mathbf{X}_{i,q}$ in Euclidean space. Likewise, the vectors $\mathbf{X}_{i,p}$ and $\mathbf{X}_{j,p}$ representing the *same* link in *different* affine reconstructions, i and j, constrain both $\boldsymbol{\Omega}_i$ and $\boldsymbol{\Omega}_j$. The residual errors, r_{sym} and r_{rig},

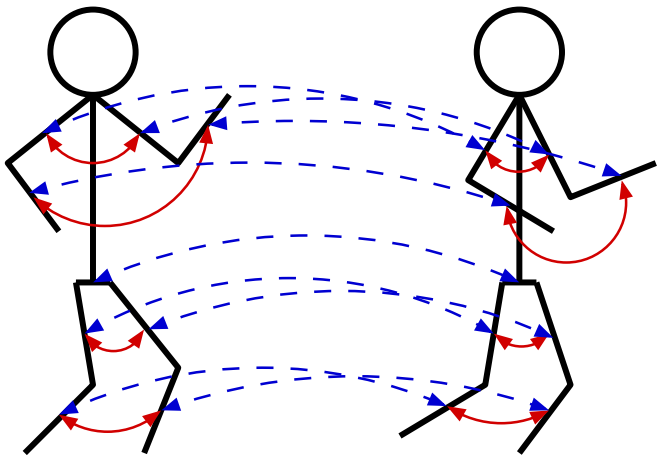


Figure 6.2: Symmetry (solid) and rigidity (dashed) constraints between a pair of reconstructions.

are given by:

$$\begin{aligned} r_{sym} &= \mathbf{X}_{i,p}^T \mathbf{B}_i^T \mathbf{B}_i \mathbf{X}_{i,p} - \mathbf{X}_{i,q}^T \mathbf{B}_i^T \mathbf{B}_i \mathbf{X}_{i,q} \\ &= \mathbf{X}_{i,p}^T \mathbf{\Omega}_i \mathbf{X}_{i,p} - \mathbf{X}_{i,q}^T \mathbf{\Omega}_i \mathbf{X}_{i,q} \qquad (6.4) \\ r_{rig} &= \mathbf{X}_{i,p}^T \mathbf{B}_i^T \mathbf{B}_i \mathbf{X}_{i,p} - \mathbf{X}_{j,p}^T \mathbf{B}_j^T \mathbf{B}_j \mathbf{X}_{j,p} \\ &= \mathbf{X}_{i,p}^T \mathbf{\Omega}_i \mathbf{X}_{i,p} - \mathbf{X}_{j,p}^T \mathbf{\Omega}_j \mathbf{X}_{j,p}. \qquad (6.5) \end{aligned}$$

Since rigidity constraints apply between pairs of reconstructions there is a combinatorial number of them, not all independent (*e.g.* $\mathbf{X}_{i,p} = \mathbf{X}_{j,p}$ and $\mathbf{X}_{i,p} = \mathbf{X}_{k,p}$ imply $\mathbf{X}_{j,p} = \mathbf{X}_{k,p}$). Although they may be applied between consecutive instants ($\{0,1\}$, $\{1,2\}$ *etc*.) as in [66], this allows the scale to drift over the sequence so we apply them with respect to the *same* reconstruction ($\{0,1\}$, $\{0,2\}$ *etc*.).

6.3 Baseline method

We begin by presenting the 'baseline' method proposed by Liebowitz and Carlsson [66]. It is against this method that we base our comparisons in Section 6.7.

6.3.1 Recovery of local structure

To recover the rectifying transformations (and hence Euclidean structure and motion), all residuals must be minimized. However, this cannot be achieved using linear methods since motion and structure constrain $\mathbf{\Omega}^{-1}$ and $\mathbf{\Omega}$, respectively. Liebowitz and Carlsson optimize directly over the $6F - 1$ elements of all $\mathbf{B}_i$ (up to scale) using the cost function:

$$C = w_{cam} \cdot c_{cam} + c_{str} \tag{6.6}$$

where

$$c_{cam} = \sum r_{skw}^2 + \sum r_{asp}^2 \tag{6.7}$$

$$c_{str} = \sum r_{sym}^2 + \sum r_{rig}^2 \tag{6.8}$$

and w_{cam} weights the costs according to the relative confidence in the motion and structural constraints. Having recovered all $\mathbf{B}_i$, they compute Euclidean structure and motion at each frame: $\widetilde{\mathbf{X}}_i \leftarrow \mathbf{B}_i\mathbf{X}_i$ and $\widetilde{\mathbf{P}}_i \leftarrow \mathbf{P}_i\mathbf{B}_i^{-1}$, respectively. We refer to this as *local* since the choice of coordinate frame is arbitrary at each time instant and rigid transformations between frames are not recovered.

6.3.2 Recovery of global structure

From the enforcement of rigidity over the sequence, any scaling due to perspective over time can be recovered from the computed projection matrices. Therefore, perspective effects over time can be removed by rescaling the image measurements as if viewed orthographically. All F normalized images of N features can then be treated as a *single* image of FN features in a static scene with a common co-ordinate frame. To

normalize the data, each Euclidean projection matrix $\widetilde{\mathbf{P}}_i$ is decomposed into its internal and external parameters:

$$\widetilde{\mathbf{P}}_i = \begin{bmatrix} \mathbf{K}_{i,1} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{i,2} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{P}}_{i,1} \\ \widehat{\mathbf{P}}_{i,2} \end{bmatrix} \tag{6.9}$$

where $\widehat{\mathbf{P}}_{i,n}$ is an orthographic projection matrix (*i.e.* $\hat{\mathbf{i}}^T\hat{\mathbf{j}} = 0$ and $\hat{\mathbf{i}}^T\hat{\mathbf{i}} = \hat{\mathbf{j}}^T\hat{\mathbf{j}} = 1$) and $\mathbf{K}_{i,n}$ is the corresponding affine calibration matrix of the form:

$$\mathbf{K} = \begin{bmatrix} s & \beta \\ 0 & \kappa s \end{bmatrix} \tag{6.10}$$

where s is the scale, κ is the aspect ratio and β the skew (subscripts are omitted for clarity). The image measurements are normalized to the same size using the scale factors, s, and a single $\mathbf{\Omega}$ is recovered for the entire sequence, yielding *global* structure where rotation and relative translation of the body between frames is also recovered. This global structure is then approximated by an articulated body of median segment lengths.

6.4 Proposed method

Although theoretically sound, the method presented in [66] has a number of practical limitations: it is inefficient since optimization is performed over $6F - 1$ variables; it has no intuitive initialization since linear solutions for $\mathbf{\Omega}_i$ are seldom positive definite such that the $\mathbf{B}_i$ cannot be recovered by Cholesky decomposition; there is considerable ambiguity when implementing the method since each $\mathbf{B}_i$ can be parameterized in several different ways (our experience suggests this can significantly affect performance); the value of w_{cam} must be chosen empirically.

6.4.1 Minimal parameterization

To address these shortcomings, we propose an improved method that exploits a minimal parameterization of $\mathbf{\Omega}_i$ based upon reasonable assumptions regarding camera calibration. Specifically, we *strictly* enforce motion constraints, resulting in reconstructions that are constrained to lie in a Euclidean coordinate frame. This has an unambiguous implementation, reduces computational complexity and provides an intuitive starting point for non-linear optimization.

By strictly enforcing motion constraints, we eliminate four degrees of freedom in $\mathbf{\Omega}_i^{-1}$. The four motion constraints defined by (6.2) and (6.3) yield a linear system with a two dimensional null-space that is spanned by two possible values for $\mathbf{\Omega}_i^{-1}$ (denoted by $\mathbf{\Omega}_{i,1}^{-1}$ and $\mathbf{\Omega}_{i,2}^{-1}$). Any linear combination of $\mathbf{\Omega}_{i,1}^{-1}$ and $\mathbf{\Omega}_{i,2}^{-1}$ satisfies all motion constraints *exactly*. We parameterize all such $\mathbf{\Omega}_i^{-1}$ using polar coordinates:

$$\mathbf{\Omega}_i^{-1}(r,\theta) = r(\cos(\theta)\cdot\mathbf{\Omega}_{i,1}^{-1} + \sin(\theta)\cdot\mathbf{\Omega}_{i,2}^{-1}) \tag{6.11}$$

$$= r\cos(\theta)(\mathbf{\Omega}_{i,1}^{-1} + \tan(\theta)\cdot\mathbf{\Omega}_{i,2}^{-1}) \tag{6.12}$$

such that for any given θ, the eigenvalues of $\mathbf{\Omega}_i^{-1}$ are equal up to scale for all positive r. Using this parameterization, only $2F-1$ parameters are required to describe the calibration of the entire sequence (in contrast to the $6F-1$ non-zero elements of $\mathbf{B}_i$ employed in the original method [66]). However, additional measures are required in order to enforce the constraint that $\mathbf{\Omega}_i^{-1}$ be positive-definite.

6.4.2 Optimization

In an early version of this method, we proposed a simple solution to this problem. From the polar parameterization of $\mathbf{\Omega}_i^{-1}$, it can be shown that $|\mathbf{\Omega}_i^{-1}|$ is expressible in

closed form as a cubic polynomial in $\tan(\theta)$ for any given r. As a result, we can compute the six values of θ for which $|\mathbf{\Omega}_i^{-1}| = 0$ as eigenvalues pass through zero. The range $[0, 2\pi)$ is therefore divided into six intervals, only one of which corresponds to a positive-definite $\mathbf{\Omega}_i^{-1}$ for all positive r. This interval, $(\theta_{min}, \theta_{max})$, is recovered by evaluating the eigenvalues of $\mathbf{\Omega}_i^{-1}$ at the midpoints of the six intervals. The midpoint of $(\theta_{min}, \theta_{max})$ then provides a simple initial value for θ, whilst r is initialized to unity.

Further investigation of the problem reveals that r_{sym} is also expressible as a polynomial in $\tan(\theta)$. As a result, we minimize r_{sym} in closed form for every time instant in the sequence to provide an improved initial value of θ. However, preliminary investigations suggest there is no closed form solution for the complete system.

We then minimize c_{str} only ($c_{cam} = 0$ by design such that w_{cam} is no longer required) over all $r > 0$ and $\theta \in (\theta_{min}, \theta_{max})$ such that the resulting $\mathbf{\Omega}_i^{-1}$ are guaranteed to be positive definite and all $\mathbf{B}$ can be recovered by Cholesky factorization. Note that since $\mathbf{\Omega}_i^{-1}$ is singular at θ_{min} and θ_{max} the cost at these values increases to infinity. As a result, the minimization is effectively 'self-constraining' and unconstrained methods are successfully employed in all but a few cases.

6.5 Bundle adjustment

Having recovered local and global structure using the minimal parameterization, we approximate the recovered structure with an articulated model of median segment lengths and estimated pose, as in [66]. However, we then optimize these parameters further using a final bundle adjustment (Levenberg-Marquardt, implemented as `lsqnonlin` in Matlab). At this point we no longer enforce symmetry constraints since they are the most uncertain of our assumptions.

Minimization of the geometric reprojection error is achieved by optimizing over the v views of i frames for all camera parameters – image scales $\{s_{i,v}\}$, camera rotations $\{\mathbf{R}_v\}$ and translations, $\{\mathbf{t}_v\}$ – and structural parameters – segment lengths, $\mathbf{L}$, and pose parameters, $\{\boldsymbol{\phi}_i\}$. We retain the assumption that the cameras have unit aspect ratio and zero skew.

Defining $\boldsymbol{\epsilon}$ as the vector of reprojection errors over all measurements, we seek to minimize the sum of squared reprojection errors, $\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}$, over all frames:

$$\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} = \sum_v \sum_i \sum_n \|s_{i,v}\mathbf{R}_v\mathbf{X}_{i,n}(\mathbf{L},\boldsymbol{\phi}_i) + \mathbf{t}_v - \mathbf{x}_{i,v,n}\|_F^2 \tag{6.13}$$

where $\mathbf{X}_{i,n}(\mathbf{L},\boldsymbol{\phi}_i)$ is the 3D location of the nth feature in the ith frame given the link lengths, $\mathbf{L}$, and pose parameters, $\boldsymbol{\phi}_i$ and $\mathbf{x}_{i,v,n}$ is the corresponding image measurement. This minimization is achieved by iteratively solving:

$$\boldsymbol{\Delta}\mathbf{p} = -(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})^{-1}\mathbf{J}^T\boldsymbol{\epsilon}\mathbf{p} \tag{6.14}$$

for $\boldsymbol{\Delta}\mathbf{p}$ where $\mathbf{p}$ is the vector of all parameters and $\mathbf{J}$ is the Jacobian (matrix of derivatives) of all measurements with respect to the parameters. λ is a regularization parameter to ensure that the step size remains within the trust region where the linearization, upon which Levenberg-Marquardt is based, remains valid. Since scale and pose parameters are frame dependent, $\mathbf{J}$ is sparse and minimization is computationally efficient. The end result is an articulated model of fixed link lengths, fitted to the anthropomorphic dimensions of the subject (up to scale) and capturing the pose at every frame such that *all* constraints are strictly enforced.

6.6 Practicalities

There are three sources of error in the presented method: incorrect spatial correspondence; incorrect joint labelling; gross outliers as a result of tracking failure. Since Chapter 4 outlines several methods for automatically recovering joint locations in an image (complete with labelling and spatial correspondence), we do not discuss these matters further here.

There remains, however, a question of robustness to tracking failure resulting in gross outliers in joint locations. We note, however that simple measures can be taken to eliminate many gross outliers using random sampling methods [112] to estimate the (affine or projective) fundamental matrix. In the case of affine projection, it has been shown that computationally cheap subspace-based methods can be employed to verify spatial matching [128].

We take a different approach based on full perspective projection: the cameras in our application are fixed, and therefore all image pairs in an entire sequence must share the same epipolar geometry. Although at each time instant it is possible to use an affine approximation (since a person's relief is typically much smaller than the viewing distance), over the entire sequence motion towards and away from a camera induces perspective effects that we can use to our advantage. Each putative feature match in an entire sequence constrains the epipolar geometry and we use this large feature set to estimate the fundamental matrix robustly using RanSaC. The results from these experiments are given in Section 6.7.1.

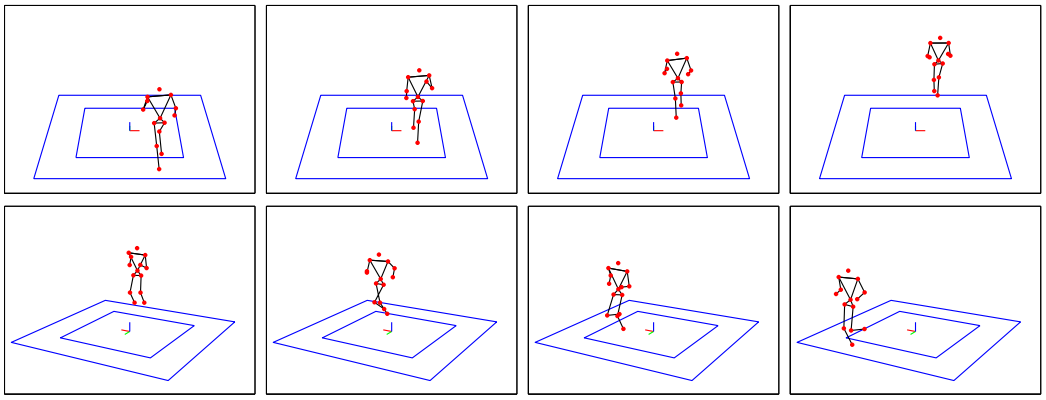


Figure 6.3: Synthetic 'running' sequence as seen from two wide baseline viewpoints. The red circles indicate point features used as inputs to the synchronization algorithm.

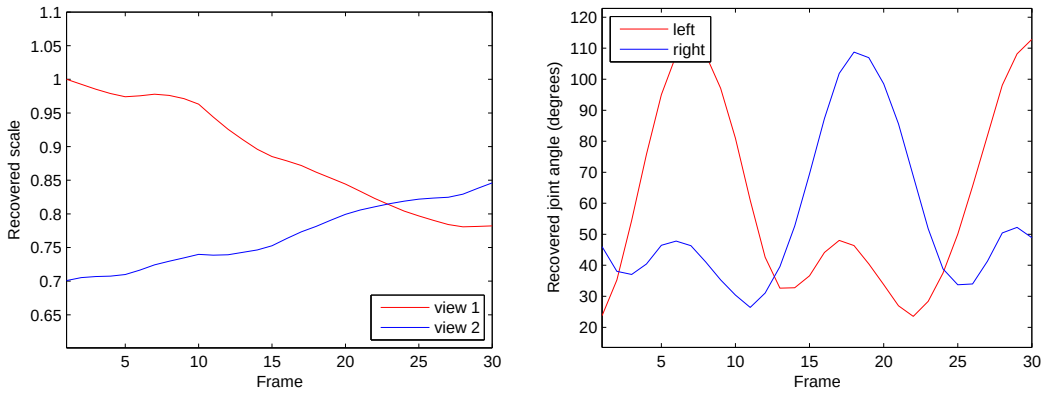
6.7 Results

We now present results using synthetic data to demonstrate the benefits of the proposed method over the original implementation [66].

6.7.1 Running sequence

Two views of a short running motion (consisting of 30 frames) were synthesized using motion capture data from a commercial system (Figure 6.3). An articulated model of known segment lengths was imaged under perspective projection and the projected image features used to recover affine structure by factorization. Metric structure and motion was then recovered using four methods: (i) rectification using a local implementation of Liebowitz and Carlsson's method ('L&C'); minimal parameter rectification with (ii) no bundle adjustment ('Minimal'); (iii) affine bundle adjustment ('A.B.A.'); (iv) perspective bundle adjustment ('P.B.A.', a 'gold standard' for comparison). This particular sequence was selected since the translation of the subject induced scaling over time due to perspective.

Figure 6.4a shows the recovered scales as a results of perspective – the subject runs



(a) (b)

Figure 6.4: (a) Recovered scaling as a result of perspective effects. (b) Recovered trajectories of the knees during running sequence. The expected periodicity and phase difference is clearly evident.

toward one camera and away from the other. The recovered angles at the knees are shown in Figure 6.4b where the periodicity and phase difference of the running motion is clearly observable.

Comparison of algorithm efficiency

Table 6.1 compares the described methods using noiseless data, based upon (i) number of iterations required for convergence, (ii) time taken (using a 2.4GHz Pentium 4 desktop computer) for convergence, (iii) total time taken (including fixed overhead costs) and (iv) final RMS reprojection error. We show separate measurements for the recovery of local structure (A), recovery of global structure (B) and bundle adjustment (C).

Minimal parameterization clearly outperforms the previously proposed method [66] in efficiency with little penalty in accuracy while bundle adjustment increases accuracy further, although at some additional computational cost. As expected, perspective bundle adjustment converges to an (almost) exact solution with noiseless data. In the remaining experiments, we show how the accuracy of the method degrades with added zero-mean Gaussian noise of standard deviation σ_n.

		L&C	Minimal	A.B.A.	P.B.A.
A	# iterations	16	10	10	10
	Time (sec)	2.08	0.68	0.62	0.62
B	# iterations	382	6	6	6
	Time (sec)	2.84	0.058	0.053	0.053
C	# iterations	-	-	12	108
	Time (sec)	-	-	16.25	163.67
Total time (sec)		6.96	2.81	23.00	233.4
RMS error (pixels)		1.41	1.44	0.785	2.9×10^{-4}

Table 6.1: Performance comparison of four methods where it is clear that the minimal parameterization heavily outperforms the original parameterization. Bundle adjustment reduces the errors further at some computational cost.

	σ (pixels)	L&C	Minimal	A.B.A.	P.B.A.
ψ	0	0.101	0.102	0.076	1.79×10^{-5}
	1	0.094	0.095	0.077	0.003
	2	0.071	0.076	0.076	0.004
	4	0.055	0.045	0.071	0.010
ω_{err}	0	0.087	0.086	0.048	3.3×10^{-5}
	1	0.138	0.134	0.034	0.013
	2	0.317	0.285	0.038	0.006
	4	0.394	0.470	1.8×10^{-4}	0.045

Table 6.2: Recovered ψ (rad) and ω_{err} (rad) with increasing image noise.

Recovery of camera parameters

To compare the recovered rotation between the cameras we recover external parameters from the computed projection matrices. Using the axis-angle notation, a rotation is represented by a unit vector, $\mathbf{a}$, parallel to the axis of rotation and the angle of rotation, ω, about this axis. We denote ground truth values by $\mathbf{a}_{gt}$ and ω_{gt}, respectively, quantifying error using the angle between the vectors $\mathbf{a}$ and $\mathbf{a}_{gt}$, $\psi = \cos^{-1}(\mathbf{a}_{gt}^T\mathbf{a})$, and the difference in angle of rotation, $\omega_{err} = |\omega_{gt} - \omega|$. Table 6.2 shows increased accuracy of the methods following bundle adjustment plus some degradation with image noise.

σ (pixels)	L&C	Minimal	A.B.A.	P.B.A.
0	0.871	0.905	0.724	0.001
1	3.541	3.576	1.428	1.078
2	7.830	6.195	2.561	2.415
4	10.10	10.60	8.666	8.256

Table 6.3: Mean percentage error in recovered limb lengths with increasing image noise.

Recovery of segment lengths

To compare segment lengths, we recover metric 3D structure over the entire sequence and compute the median length for each body segment. These median values are then normalized such that the hips have unit length before comparing them with ground truth values. Table 6.3 shows mean percentage errors in recovered body segment length using the four methods for a single test. We see a sharp increase in error with image noise since even a small amount of noise may result in a large *percentage* error in projected length for frames where the limb is almost normal to the image plane. Since our minimal parameterization strictly enforces motion constraints we might expect a deterioration in the recovered structure (which 'absorbs' all of the measurement errors). However, our results suggest that this effect is very slight.

Recovery of joint trajectories

We now show how image noise affects RMS error in joint angle, using the elbow and knee joints that are invariant to global coordinate frame. Table 6.4 shows error increase sharply since even a small error in projected length is interpreted as a large error in joint angle. The converse problem is encountered in model-based tracking where rotations out of the image plane are almost unobservable since they result in small image motion.

σ (pixels)	L&C	Minimal	A.B.A.	P.B.A.
0	0.0521	0.0511	0.0328	3.8×10^{-5}
1	0.1276	0.1263	0.0716	0.0597
2	0.2851	0.2776	0.1712	0.1644
4	0.3390	0.3435	0.3255	0.3220

Table 6.4: Mean RMS error in joint angle (rad) over the knee and elbow joints.

Sensitivity to gross outliers

Finally, we investigate the sensitivity of the algorithm to gross outliers as a result of tracking error. Such errors have two deleterious effects: (i) increased RMS projection errors and consequent increased errors in recovered structure; (ii) more seriously, they often result in failure of the algorithm to converge to a sensible solution. We show that such problems are significantly reduced using robust matching techniques.

Using a different synthetic sequence of 38 frames, we added Gaussian noise ($\sigma = 2$ pixels) and performed self-calibration ('No outliers'). We then deliberately corrupted approximately 10% of the correspondences (selected randomly) with Gaussian noise of standard deviation 40 pixels to simulate gross error and performed self-calibration three more times: (i) after removing all known outliers ('Known'); (ii) after removing outliers detected using robust matching ('RanSaC'); (iii) after removing none of the outliers ('Naive'). Since this experiment concerns only the early stages of the algorithm, no bundle adjustment was used.

Table 6.5 shows the convergence frequency over 100 tests, and the RMS reprojection and structure errors averaged over the tests that did converge (only points labelled as inliers were used to compute these values). Methods 'Naive' and 'Known' respectively show that performance is poor with outliers present but improves dramatically when they are all removed. The 'RanSaC' method shows that robust matching methods [112] provide some defence against such outliers. In particular the percentage of trials that

Method	Convergence	Reproj. error RMS (pixels)	Limb error Mean (%)	Limb error Max. (%)
No outliers	100%	1.78	2.52	6.31
Known	100%	1.81	2.71	6.55
RanSaC	81%	2.23	4.36	9.56
Naive	31%	7.30	5.11	12.10

Table 6.5: Convergence frequency, RMS reprojection error and limb lengths error with outliers

converge is dramatically increased, as well as an expected decrease in structural error.

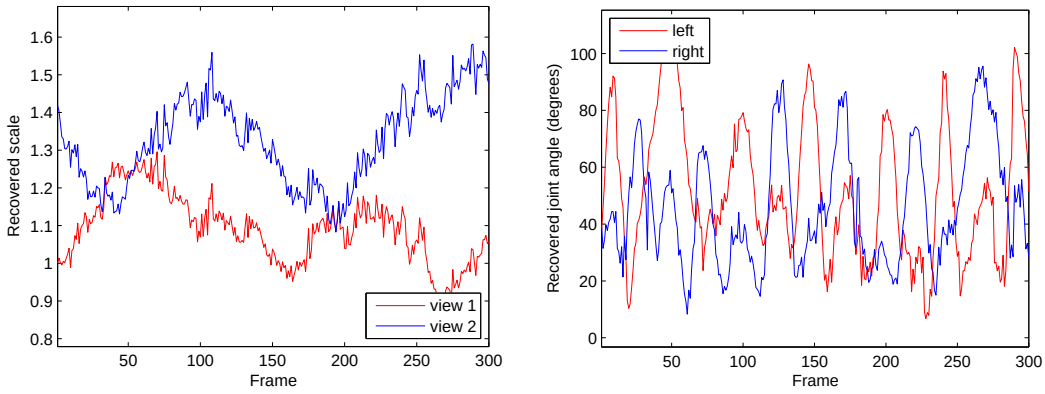
However, one weakness of binocular outlier rejection schemes is that only those outliers lying far from their estimated epipolar line are detected. Large noise components parallel to the epipolar line remain undetected and continue to influence the recovered structure and motion adversely. Further mitigation against these effects could be obtained using, for example, smooth motion priors to detect remaining outliers.

6.8 Real examples

6.8.1 Running sequence

Applying the algorithm to the 'running' sequence (Figure 5.7), the affine reconstructions were calibrated using the minimal parameterization in 37 iterations, taking approximately 4.3 seconds. In contrast, Liebowitz's method took 38 seconds to compute local structure and did not converge on global structure within 10^4 iterations. Affine bundle adjustment was then applied to the recovered structure reducing RMS reprojection error from 5.44 pixels to 2.76 pixels. For comparison, perspective bundle adjustment reduced RMS reprojection error to 2.24 pixels.

Figure 6.5a shows the recovered scaling of the body as a result of perspective whilst Figure 6.5b shows the joint angle trajectories of the knees over 150 frames of the running sequence. The anticipated periodicity and phase difference in the running



(a) (b)

Figure 6.5: (a) Recovered scaling as a result of perspective effects. (b) Recovered trajectories of the knees during running sequence. The expected periodicity and phase difference is clearly evident.

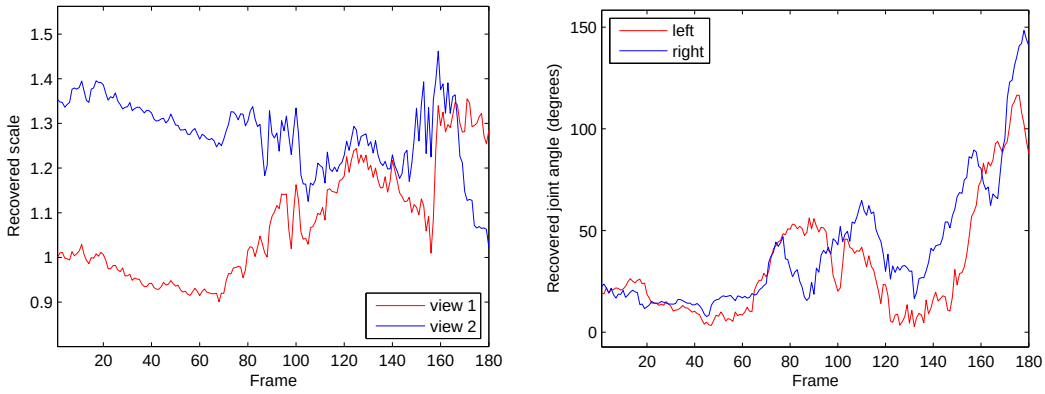
Limb	Left	Right
Upper arm	1.223	1.249
Lower arm	1.004	1.071
Upper leg	1.619	1.679
Lower leg	1.693	1.709

Table 6.6: Recovered body segment lengths (relative to the hips) for the running sequence. The recovered limbs are approximately symmetric and in proportion.

motion is clearly evident. Table 6.6 shows the recovered body segment lengths (again, normalized such that the hips have unit length). It can be seen that the recovered body model is in proportion and approximately symmetric, despite the fact we impose no constraints on the symmetry of the body during bundle adjustment.

6.8.2 Handstand sequence

For the 'handstand' sequence (Figure 5.8), our method converged in 109 iterations, taking only 9.5 seconds, with an RMS reprojection error of 6.79 pixels. Affine bundle adjustment reduced RMS reprojection error to 3.92 pixels, compared with 3.41 pixels following perspective bundle adjustment. In contrast, Liebowitz's method required 6951 iterations, taking 101 seconds, with an RMS reprojection error of 7.56 pixels.



(a) (b)

Figure 6.6: (a) Recovered scaling as a result of perspective effects. (b) Recovered trajectories of the knees during handstand sequence shoing now periodicity or particular phase difference.

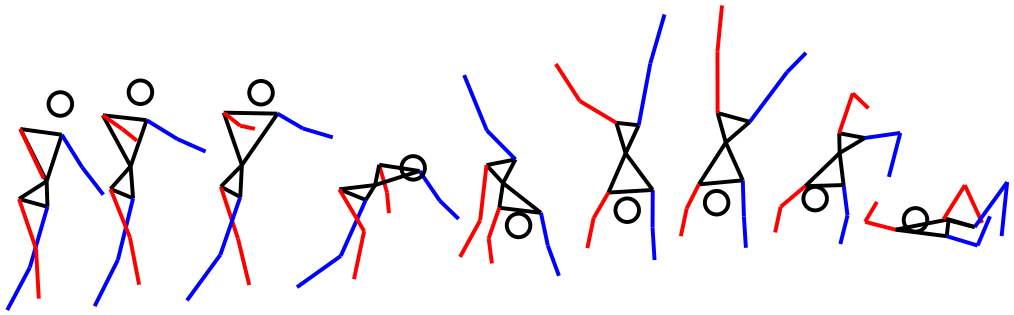


Figure 6.7: Euclidean reconstruction of a handstand sequence

Figure 6.6a shows the recovered scales due to perspective and Figure 6.6b shows the joint angle trajectories of the knees. In this case, there is no periodicity or phase change since the motion is not cyclic. Again, we see that the recovered kinematic structure (Table 6.7) is in proportion and approximately symmetric.

6.8.3 Juggling sequence

For the juggling sequence (Figure 5.11), the minimal parameterization converged in 19 iterations, taking approximately 0.8 seconds, with an RMS reprojection error of 4.13 pixels. In contrast, Liebowitz's method required 1425 iterations, taking 20.2 seconds,

Limb	Left	Right
Upper arm	1.076	1.105
Lower arm	0.856	0.968
Upper leg	1.645	1.719
Lower leg	1.458	1.584

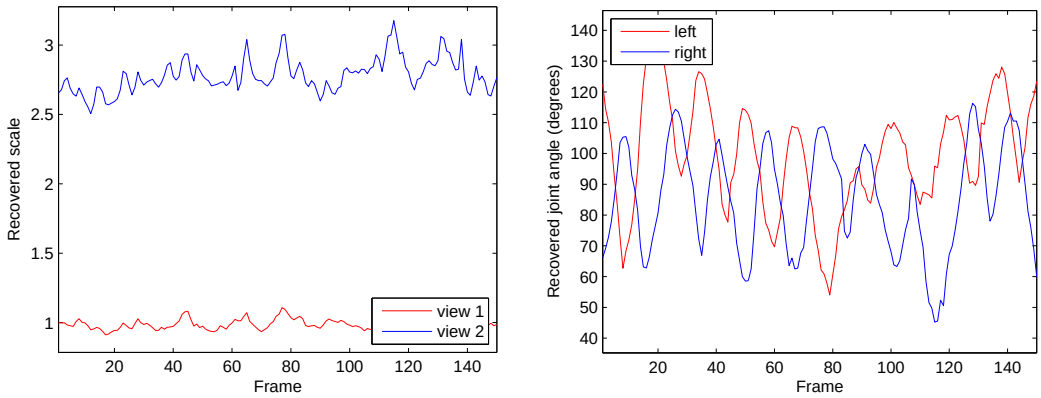
Table 6.7: Recovered body segment lengths (relative to the hips) for the handstand sequence. The recovered limbs are approximately symmetric and in proportion.

Limb	Left	Right
Upper arm	1.000	1.032
Lower arm	0.984	0.982

Table 6.8: Recovered limb lengths (relative to the left upper arm) for the juggling sequence. The recovered limbs are approximately symmetric and in proportion.

albeit with a better RMS reprojection error of 3.78 pixels. Affine bundle adjustment reduced RMS reprojection error further to 2.13 pixels, compared with 2.15 pixels following perspective bundle adjustment.

Again, Figure 6.8a shows the scales due to perspective effect that are small in this case since the subject does not move towards or away from the camera. This lack of change in depth would explain why perspective bundle adjustment performed no better than the affine bundle adjustment for this sequence. Figure 6.8b shows the recovered joint trajectories of the elbows during the motion where the periodicity of the motion is clearly apparent in addition to the phase difference. Table 6.8 shows the recovered body segment lengths where we see that the symmetry has been recovered and the segments are in proportion, despite the reduced number of structural constraints (the lengths are normalized with respect to the upper left arm). Figure 6.9 shows the reconstructed upper body in a Euclidean co-ordinate frame.



(a) (b)

Figure 6.8: (a) Recovered scales where we see little change since the subject was not moving with respect to the camera. (b) Recovered trajectories of the elbows during juggling where the out of phase periodic motion is clearly observable.

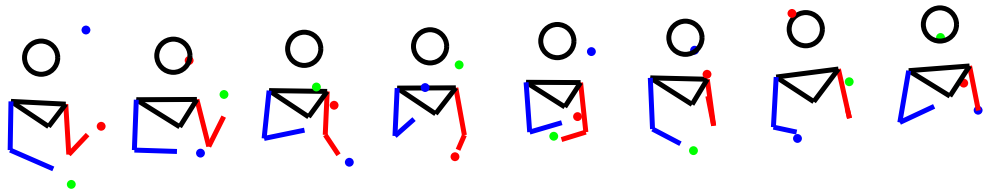


Figure 6.9: Euclidean reconstructions from juggling sequence

6.9 Summary

In this chapter, we have presented a self-calibration method for the underconstrained case where only two views are available of the motion. We extended current methods [66] that exploit the structure of the human body by proposing a minimal parameterization of the solution space. This resulted in a computationally efficient algorithm with an intuitive initialization that resolves implementation ambiguity. Bundle adjustment then recovered the maximum likelihood solution by minimizing a geometric reprojection error. We demonstrated the method on synthetic and real sequences of human motion, showing accurate recovery of joint angles and camera parameters. We also presented an analysis of sensitivity to outliers, showing that robust matching

greatly improves performance.

6.9.1 Future work

Closed-form solution

The existence of a closed-form solution for the symmetry cost offers hope for a similar solution for the entire system. Preliminary investigations suggest this may not be the case although further work is required in this exciting direction.

Sequential implementation

Since the method uses all affine reconstructions simultaneously, it is strictly a batch process. An obvious extension would be to develop a sequential process that converges to the maximum likelihood solution as more frames are added.

Regularization

The sharp increase in joint angle error with noise suggests that integration with a motion model would also be beneficial during the bundle adjustment to impose smoothness priors. This would also aid in the detection of gross outliers where the error vector is parallel to the epipolar line and undetectable by robust matching techniques (*e.g.* RanSaC).

Chapter 7

Conclusion

This thesis has presented a study of articulated motions, as viewed through the lens of a camera. We conclude by summarizing the main contributions of the thesis and reviewing directions for future research.

7.1 Contributions

The key contributions of the thesis can be summarized as follows:

- An extension of the Factorization algorithm [111] was presented in Chapter 3 for articulated objects. It was shown that for a pair of objects coupled by a universal joint or hinge, the rank of the resulting matrix of feature tracks is decremented by 1 or 2, respectively. The presented method exploits this fact to detect articulated motion from feature tracks and recover the parameters of the system such as centres/axes of rotation and joint angles.

- An empirical comparison of several methods for estimating joint centre projections in an image of a human using a training corpus of synthetic data was undertaken in Chapter 4. It was shown that some popular methods for this task do not scale well for large training datasets, placing intractable demands on com-

putation and memory storage. A simple tree searching algorithm was integrated with a particle filter to track human motion from a single view.

- A novel method of synchronizing video sequences of human motion using projected joint centres was presented. The algorithm was based on the Factorization method, although a general framework was presented for different camera models. Synchronization parameters were recovered for sequences of unknown and different frame rates using interpolation of feature locations. The method was demonstrated to be robust to noise and accurate to within fractions of a frame.

- A self-calibration method for a pair of cameras observing human motion was presented. Developing an existing method [66], the algorithm proposed a reduced parameterization of the solution space resulting in well-behaved and efficient optimization with an intuitive initialization in closed form. Bundle adjustment was then applied to reduce a geometric reprojection error for the recovery of a maximum likelihood solution.

7.2 Future work

Of the various directions for further research we have outlined in this thesis, we consider the following to be most important and interesting:

- In order for articulated structure from motion to be employed in a human motion analysis context, it must be extended to handle longer kinematic chains featuring a mixture of joint types. A unified framework that can detect and process all types of dependency is also highly desirable.

- The comparison of Machine Learning methods provided in Chapter 4 did not in-

clude mixture models nor attempt a thorough review of current techniques. This is a rapidly expanding area in the human motion tracking community that should be investigated more rigorously. In particular, the synergy between discriminative and generative methods was touched upon but has yet to be exploited to its full potential.

- Searching, sampling and regression methods based on the occluding contour (silhouette) of the subject currently rely on an accurate segmentation of the subject from the background. In this work, this was achieved via background subtraction. An interesting line of inquiry may investigate whether a training corpus could be employed for 'intelligent' segmentation of the subject from the background for improved tracking.

- Since self-calibration using symmetry constraints only can be shown to have a closed-form solution, such constraints may be incorporated into the synchronization framework. This would effectively impose priors on a pair of frames such that cost is not only dictated by reprojection error (derived indirectly from rank constraints) but also by the maximum possible symmetry of the recovered structure. Initial results in this line of inquiry have already shown promise.

- A closed form solution for self-calibration is highly desirable since independence from non-linear optimization methods would result in more robust and efficient algorithms. In particular, it remains to be established whether the solution space is convex (within parameter bounds) such that an iterative algorithm not based on gradient descent could be implemented to find the solution in an efficient manner.

Bibliography

[1] G. Adiv. Determining the three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:384–401, July 1985.

[2] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression. In *Proc. 22nd IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July*, volume 2, pages 882–888, 2004.

[3] A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In *Proc. 8th European Conf. on Computer Vision, Prague, 11–14 May*, pages 54–65. Springer LNCS 3023, 2004.

[4] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. In *Proc. IEEE Workshop on Vision for Human-Computer Interaction, San Diego, CA, 21 June*, 2005.

[5] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):1–15, January 2006.

[6] J. K. Aggarwal and Q. Cai. Human motion analysis : A review. *Computer Vision and Image Understanding*, 73(3):428–440, March 1999.

[7] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. BoostMap : A method for efficient approximate similarity rankings. In *Proc. 22nd IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July*, volume 2, pages 268–275, 2004.

[8] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *Proc. 21st IEEE Conf. on Computer Vision and Pattern Recognition, Madison, WI, USA, 16–22 June*, volume 2, pages 432–442, 2003.

[9] D. Ballard and C. Brown. *Computer Vision*. Prentice Hall, 1982. ISBN 0131653164.

[10] D. H. Ballard and O. A. Kimball. Rigid body motion from depth and optical flow. *Computer Vision, Graphics, and Image Processing*, 22(1):95–115, April 1983.

[11] C. Barron and I. Kakadiaris. Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding*, 81(3):269–284, March 2001.

[12] A. Baumberg and D. Hogg. Learning flexible models from image sequences. In *Proc. European Conf. on Computer Vision*, volume 1, pages 299–308. Springer LNCS 800, 1994.

[13] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[14] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[15] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, March 1987.

[16] M. Brand. Shadow puppetry. In *Proc. 7th Int'l Conf. on Computer Vision, Kerkyra, 20–25 September*, volume 2, pages 1237–1244, 1999.

[17] M. Brand. Morphable 3D models from video. In *Proc. 20th IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December*, volume 2, pages 456–463, 2001.

[18] M. Brand and R. Bhotika. Flexible flow for 3D nonrigid tracking and shape recovery. In *Proc. 20th IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December*, volume 1, pages 315–324, 2001.

[19] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. 16th IEEE Conf. on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 17–19 June*, pages 568–574, 1997.

[20] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proc. 19th IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, 13–15 June*, pages 690–696, 2000.

[21] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. 17th IEEE Conf. on Computer Vision and Pattern Recognition, Santa Barbara, CA, USA, 23–25 June*, pages 8–15, 1998.

[22] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *Proc. 19th IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, 13–15 June*, pages 682–689, 2000.

[23] Y. Caspi and M. Irani. Alignment of non-overlapping sequences. In *Proc. 8th Int'l Conf. on Computer Vision, Vancouver, 7–14 July*, volume 2, pages 76–83, 2001.

[24] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. In *Proc. Workshop on Vision and Modelling of Dynamic Scenes, 2 June*, 2002.

[25] T.-J. Cham and J. Rehg. A multiple hypothesis approach to figure tracking. In *Proc. 18th IEEE Conf. on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 23–25 June*, volume 2, pages 239–245, 1999.

[26] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proc. 9th Int'l Conf. on Computer Vision, Nice, 14–17 October*, volume 1, pages 77–84, 2003.

[27] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Proc. 5th Int'l Conf. on Computer Vision, Boston, 20–23 June*, pages 1071–1077, 1995.

[28] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *Proc. 7th Int'l Conf. on Computer Vision, Kerkyra, 20–25 September*, volume 2, pages 716–721, 1999.

[29] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc. 19th IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, 13–15 June*, volume 2, pages 126–133, 2000.

[30] J. Deutscher, A. Davison, and I. Reid. Automatic partitioning of high dimensional search spaces associated with articulated body motion capture. In *Proc. 20th IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December*, volume 2, pages 669–676, 2001.

[31] J. Deutscher, B. North, B. Bascle, and A. Blake. Tracking through singularities and discontinuities by random sampling. In *Proc. 7th Int'l Conf. on Computer Vision, Kerkyra, 20–25 September*, volume 2, pages 1144–1149, 1999.

[32] D. DiFranco, T.-J. Cham, and J. Rehg. Reconstruction of 3-D figure motion from 2-D correspondences. In *Proc. 20th IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December*, volume 1, pages 307–314, 2001.

[33] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[34] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.

[35] e-Frontier. Poser software website. http://www.e-frontier.com.

[36] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. 9th Int'l Conf. on Computer Vision, Nice, 14–17 October*, volume 2, pages 726–733, 2003.

[37] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Proc. 19th IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, 13–15 June*, volume 2, pages 66–73, 2000.

[38] D. Gavrila and L. Davis. 3-D model-based tracking of humans in action : A multi-view approach. In *Proc. 15th IEEE Conf. on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 18–20 June*, pages 73–80, 1996.

[39] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *Proc. 7th Int'l Conf. on Computer Vision, Kerkyra, 20–25 September*, volume 1, pages 87–93, 1999.

[40] D. M. Gavrila. The visual analysis of human movement : A survey. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999.

[41] L. Goncalves, E. Di Bernado, E. Ursella, and P. Perona. Monocular tracking of the human arm in 3D. In *Proc. 5th Int'l Conf. on Computer Vision, Boston, 20–23 June*, pages 764–770, 1995.

[42] N. Gordon, D. Salmond, and A. Smith. A novel approach to non-linear and non-gaussian Bayesian state estimation. *IEE Proceedings F*, 140:107–113, 1993.

[43] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *Proc. 22nd IEEE Conf. on Computer Vision and Pattern*

Recognition, Washington, DC, USA, 27 June–2 July, volume 1, pages 220–227, 2004.

[44] K. Grauman, G. Shakhnarovich, and T. Darrell. A Bayesian approach to image-based visual hull reconstruction. In *Proc. 21st IEEE Conf. on Computer Vision and Pattern Recognition, Madison, WI, USA, 16–22 June*, volume 1, pages 187–194, 2003.

[45] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3D structure with a statistical image-based shape model. In *Proc. 9th Int'l Conf. on Computer Vision, Nice, 14–17 October*, volume 1, pages 641–648, 2003.

[46] I. Haritaoglu, D. Harwood, and L. Davis. Ghost : A human body part labeling system using silhouettes. In *Proc. 14th Int'l Conf. on Pattern Recognition, 16–20 August*, volume 1, pages 77–82, 1998.

[47] R. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997.

[48] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. ISBN 0521540518.

[49] L. Herda, R. Urtasun, and P. Fua. Hierarchical implicit surface joint limits to constrain video-based motion capture. In *Proc. 8th European Conf. on Computer Vision, Prague, 11–14 May*, volume 2, pages 405–418. Springer LNCS 3022, 2004.

[50] D. Hogg. Model-based vision : A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, February 1983.

[51] N. Howe, M. Leventon, and W. Freeman. Bayesian reconstruction of 3D human motion from single-camera video. In *Proc. Advances in Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December*, pages 820–826, 1999.

[52] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8:179–187, February 1962.

[53] T. S. Huang and C. H. Lee. Motion and structure from orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):536–540, May 1989.

[54] S. Ioffe and D. Forsyth. Probabilistic methods for finding people. *International Journal of Computer Vision*, 43(1):45–68, June 2001.

[55] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Proc. 7th Int'l Conf. on Computer Vision, Kerkyra, 20–25 September*, volume 1, pages 626–633, 1999.

[56] M. Irani and P. Anandan. Factorization with uncertainty. In *Proc. 6th European Conf. on Computer Vision, Dublin, 26 June–1 July*, volume 1, pages 539–553. Springer LNCS 1842, 2000.

[57] M. Isard and A. Blake. ConDensAtion – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998.

[58] M. Isard and A. Blake. IConDensAtion : Unifying low-level and high-level tracking in a stochastic framework. In *Proc. 5th European Conf. on Computer*

Vision, Freiburg, 2–6 June, volume 1, pages 893–908. Springer LNCS 1406, 1998.

[59] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.

[60] S. Ju, M. Black, and Y. Yacoob. Cardboard people : A parameterized model of articulated image motion. In *Proc. 2nd Int'l Conf. on Automatic Face and Gesture Recognition, Killington, VT, USA, 14–16 October*, pages 38–44, 1996.

[61] I. Kakadiaris and D. Metaxas. Model-based estimation of 3D human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1453–1459, December 2000.

[62] J. J. Koenderink and A. J. van Doorn. Affine structure from motion. *Journal of the Optical Society of America*, 8(2):377–385, February 1991.

[63] L. J. Latecki and R. Lakamper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73(3):441–454, March 1999.

[64] M. W. Lee and I. Cohen. A model-based approach for estimating human 3D poses in static images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):905–916, June 2006.

[65] M. K. Leung and Y.-H. Yang. First sight : A human body outline labeling system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):399–377, April 1995.

[66] D. Liebowitz and S. Carlsson. Uncalibrated motion capture exploiting articulated structure constraints. *International Journal of Computer Vision*, 51(3):171–187, 2003.

[67] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[68] J. MacCormick and M. Isard. Partitioned sampling, articulated objects and interface-quality hand tracking. In *Proc. 6th European Conf. on Computer Vision, Dublin, 26 June–1 July*, volume 2, pages 3–19. Springer LNCS 1843, 2000.

[69] D. Marr. *Vision : A computational investigation into the human representation and processing of visual information*. W. H. Freeman, 1982. ISBN 0716715678.

[70] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three dimensional shapes. Technical Report AIM-416, Massachusetts Institute of Technology, 1977.

[71] T. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, March 2001.

[72] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, April 2001.

[73] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Proc. 7th European Conf. on Computer Vision, Copenhagen, 28–31 May*, volume 3, pages 663–680. Springer LNCS 2352, 2002.

[74] T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):858–867, August 1997.

[75] D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Proc. 6th Int'l Conf. on Computer Vision, Bombay, 4–7 January*, pages 696–702, 1998.

[76] D. Morris and J. Rehg. Singularity analysis for articulated object tracking. In *Proc. 17th IEEE Conf. on Computer Vision and Pattern Recognition, Santa Barbara, CA, USA, 23–25 June*, pages 289–297, 1998.

[77] R. Mukundan, S. H. Ong, and P. A. Lee. Image analysis by Tchebichef moments. *IEEE Transactions on Image Processing*, 10(9):1357–1364, September 2001.

[78] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(6):522–536, 1980.

[79] V. Pavlovic, J. Rehg, T.-J. Cham, and K. Murphy. A dynamic Bayesian network approach to figure tracking using learned dynamic models. In *Proc. 7th Int'l Conf. on Computer Vision, Kerkyra, 20–25 September*, volume 1, pages 94–101, 1999.

[80] C. Poelman and T. Kanade. A paraperspective factorization method for structure and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218, March 1997.

[81] D. Pooley, M. Brooks, A. van den Hengel, and W. Chojnacki. A voting scheme for estimating the synchrony of moving-camera videos. In *Proc. Int'l Conf. on Image Processing, Barcelona, 14–18 September*, volume 1, pages 413–416, 2003.

[82] L. Quan. Self-calibration of an affine camera from multiple views. *International Journal of Computer Vision*, 19(1):93–110, July 1996.

[83] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *Proc. 21st IEEE Conf. on Computer Vision and Pattern Recognition, Madison, WI, USA, 16–22 June*, volume 2, pages 467–474, 2003.

[84] I. Reid and D. W. Murray. Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision*, 18(1):41–60, April 1996.

[85] I. Reid and A. Zisserman. Goal-directed video metrology. In *Proc. 4th European Conf. on Computer Vision, Cambridge, 15–18 April*, volume 2, pages 647–658. Springer LNCS 1065, 1996.

[86] K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision and Image Understanding*, 59(1):94–115, January 1994.

[87] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *Proc. 7th European Conf. on Computer Vision, Copenhagen, 28–31 May*, volume 4, pages 700–714. Springer LNCS 2353, 2002.

[88] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *Proc. 19th IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, 13–15 June*, volume 2, pages 721–727, 2000.

[89] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer. The HumanID gait challenge problem : Data sets, performance, and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):162–177, February 2005.

[90] A. Shahrokni, T. Drummond, and P. Fua. Texture boundary detection for real-time tracking. In *Proc. 8th European Conf. on Computer Vision, Prague, 11–14 May*, volume 2, pages 566–575. Springer LNCS 3022, 2004.

[91] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proc. 9th Int'l Conf. on Computer Vision, Nice, 14–17 October*, volume 2, pages 750–759, 2003.

[92] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32, November 1999.

[93] H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3D human figures using 2D image motion. In *Proc. 6th European Conf. on Computer Vision, Dublin, 26 June–1 July*, volume 2, pages 702–718. Springer LNCS 1843, 2000.

[94] H. Sidenbladh, M. Black, and L. Sigal. Implicit probabilistic models of human motion for synthesis and tracking. In *Proc. 7th European Conf. on Computer Vision, Copenhagen, 28–31 May*, volume 1, pages 784–800. Springer LNCS 2350, 2002.

[95] H. Sidenbladh and M. J. Black. Learning the statistics of people in images and video. *International Journal of Computer Vision*, 54(1–3):183–209, August 2003.

[96] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking loose-limbed people. In *Proc. 22nd IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July*, volume 1, pages 421–428, 2004.

[97] D. Sinclair, L. Paletta, and A. Pinz. Euclidean structure recovery through articulated motion. In *Proc. 10th Scandinavian Conf. on Image Analysis, Lappeenranta, Finland, 9–11 June*, pages 991–998, 1997.

[98] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3D human motion estimation. In *Proc. 23nd IEEE Conf. on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June*, volume 1, pages 390–397, 2005.

[99] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3D body tracking. In *Proc. 20th IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December*, volume 1, pages 447–454, 2001.

[100] C. Sminchisescu and B. Triggs. Building roadmaps of local minima of visual models. In *Proc. 7th European Conf. on Computer Vision, Copenhagen, 28–31 May*, volume 1, pages 566–582. Springer LNCS 2350, 2002.

[101] C. Sminchisescu and B. Triggs. Hyperdynamics importance sampling. In *Proc. 7th European Conf. on Computer Vision, Copenhagen, 28–31 May*, volume 1, pages 769–783. Springer LNCS 2350, 2002.

[102] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3D human tracking. In *Proc. 21st IEEE Conf. on Computer Vision and Pattern Recognition, Madison, WI, USA, 16–22 June*, volume 1, pages 69–76, 2003.

[103] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In *Proc. 19th IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, 13–15 June*, volume 1, pages 345–352, 2000.

[104] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *Proc. 9th Int'l Conf. on Computer Vision, Nice, 14–17 October*, volume 2, pages 1063–1070, 2003.

[105] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *Proc. 7th European Conf. on Computer Vision, Copenhagen, 28–31 May*, volume 1, pages 629–644. Springer LNCS 2350, 2002.

[106] L. Taycher and T. Darrell. Bayesian articulated tracking using single frame pose sampling. In *Proc. IEEE Workshop on Statistical and Computational Theories of Vision, Nice, 13 October*, 2003.

[107] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, 80(3):349–363, December 2000.

[108] M. R. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70:920–930, August 1980.

[109] S. B. Thies, P. Tresadern, L. Kenney, D. Howard, Y. Goulermas, C. Smith, and J. Rigby. A "virtual sensor" tool to simulate accelerometer output for upper limb FES triggering. In *Proc. World Conference of Biomechanics*, July 2006.

[110] M. Tipping. The Relevance Vector Machine. In *Proc. Advances in Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December*, pages 652–658, 1999.

[111] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[112] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, September 1997.

[113] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proc. 20th IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December*, volume 1, pages 493–500, 2001.

[114] P. Tresadern and I. Reid. Synchronizing image sequences of non-rigid objects. In *Proc. 14th British Machine Vision Conf., Norwich, 9–11 September*, volume 2, pages 629–638, 2003.

[115] P. Tresadern and I. Reid. Uncalibrated and unsynchronized human motion capture : A stereo factorization approach. In *Proc. 22nd IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July*, volume 1, pages 128–134, 2004.

[116] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *Proc. 23nd IEEE Conf. on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June*, volume 2, pages 1110–1115, 2005.

[117] S. Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London, Series B*, 203(1153):405–426, January 1979.

[118] J. Vermaak, S. J. Godsill, and A. Doucet. Sequential Bayesian kernel regression. In *Proc. Advances in Neural Information Processing Systems*, 2003.

[119] Vicon Motion Capture Solutions. Online specifications. http://www.vicon.com.

[120] R. Vidal and R. Hartley. Motion segmentation with missing data using Power-Factorization and GPCA. In *Proc. 22nd IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July*, volume 2, pages 310–316, 2004.

[121] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. 20th IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December*, volume 1, pages 511–518, 2001.

[122] S. Wachter and H-H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192, June 1999.

[123] L. Wolf and A. Zomet. Correspondence-free synchronization and reconstruction in a non-rigid scene. In *Proc. Workshop on Vision and Modelling of Dynamic Scenes, 2 June*, 2002.

[124] C. Wren and A. Pentland. Dynamic models of human motion. In *Proc. 3rd Int'l Conf. on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April*, pages 22–27, 1998.

[125] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder : Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[126] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. In *Proc. 23nd IEEE Conf. on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June*, volume 2, pages 815–821, 2005.

[127] P.-T. Yap, R. Paramesran, and S.-H. Ong. Image analysis by Krawtchouk moments. *IEEE Transactions on Image Processing*, 12(11):1367–1377, November 2003.

[128] L. Zelnik-Manor and M. Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorization. In *Proc. 21st IEEE Conf. on Computer Vision and Pattern Recognition, Madison, WI, USA, 16–22 June*, volume 2, pages 287–293, 2003.

[129] D. S. Zhang and G. Lu. A comparative study of curvature scale space and Fourier descriptors. *Journal of Visual Communication and Image Representation*, 14(1):41–60, 2003.

[130] D. S. Zhang and G. Lu. Study and evaluation of different Fourier methods for image retrieval. *Image and Vision Computing*, 23(1):33–49, January 2005.

[131] L. Zhao and C. Thorpe. Recursive context reasoning for human detection and parts identification. In *Proc. IEEE Workshop on Human Modelling, Analysis and Synthesis, Hilton Head Island, SC, 16 June*, 2000.

[132] C. Zhou and H. Tao. Dynamic depth recovery from unsynchronized video streams. In *Proc. 21st IEEE Conf. on Computer Vision and Pattern Recognition, Madison, WI, USA, 16–22 June*, volume 2, pages 351–358, 2003.

Appendix A

An Empirical Comparison of Shape Descriptors

In this appendix, we present a brief comparison of a number of shape representations for searching a database of training examples. We discuss potential candidates for the task, justifying the selected representations included in the comparison. In particular, we include the recently proposed Histogram of Shape Contexts and demonstrate that it provides little, if any, benefit over alternative methods despite the considerable increase in computational cost that is required.

A.1 Introduction

Due to the rapid increase in affordable secondary storage over the last few years, it is becoming increasingly important to develop systems that retrieve data based on *content* rather than annotating the data by hand. This has led to the growth of interest in shape matching and retrieval algorithms. Application areas for such Content Based Image Retrieval (CBIR) include searching the Web (*e.g.* Google Images) and more specific fields such as the enforcement of trademarks.

In such applications, it is typically infeasible to use the raw, high-dimensional image to describe the data. Instead, features are computed that retain the most informative data in the image. This dimensionality reduction provides three major benefits:

- **Lower storage requirements** since each image is represented by a compact feature vector.
- **Increased efficiency** since the training data can be processed more rapidly.
- **Reduced sensitivity** to noise since the features should capture the most informative shape characteristics whilst ignoring irrelevant details (*e.g.* noise).

In this appendix, we investigate several selected shape representations that reduce the dimensionality of training images for the purpose of shape retrieval in applications such as human pose estimation (see Chapter 4).

A.1.1 Related Work

Due to the nature of the dataset used in this investigation (binary silhouettes of 128×128 pixels), certain shape representations are inappropriate for this task. Descriptors based on the topology of the occluding contour [63] are unsuitable since they may change dramatically with small changes in underlying pose (*e.g.* as the subject places their hands on their hips, 'holes' are created that modify the topology). Furthermore, representations based on curvature [129] typically require a continuous (or sufficiently high resolution) contour that is rarely available in this application. Similar arguments rule out Fourier decompositions [130] and shock graphs/median axis representations [92].

The remaining candidates can be divided into two classes: *global* and *local* descriptors. Global representations use every pixel to compute every feature such that a localized corruption of the input image (*e.g.* occlusion, shadow) induces an error in every feature. Such representations include moments [108, 77, 127], Lipschitz embeddings [8] and Principal Component Analysis (PCA) of the image. In contrast, local representations use only a subset of the image to compute each feature such that only

certain features are affected by a localized error in the input image. Such representations include the recently proposed Histogram of Shape Contexts (HoSC) that has successfully been employed in human pose estimation [5]. Each of these representations is described in detail in Section A.3.

A.1.2 Contributions

This chapter presents a comparison of several selected shape representations for the application of human pose estimation. In particular, the comparison includes the recently proposed Histogram of Shape Contexts (HoSC) [5] that has demonstrated seemingly successful results in the application of human pose estimation. However, to date no comparison has been undertaken between the HoSC and other shape representations. This comparison suggests that any benefit gained from the HoSC representation is small and does not justify the considerable increase in computation required.

A.2 Method

We begin by discussing the dataset used for the evaluation, which shape representations were evaluated, and how.

A.2.1 Dataset generation

We generated a training set of $N = 10000$ binary silhouettes of 128×128 pixels, as shown in Figure A.1, of a human body model using motion capture data (available at the time of printing from `http://mocap.cs.cmu.edu`). An additional 250 silhouettes were generated to test the retrieval performance of the shape descriptors. The training set included silhouettes from several different motions observed from 4 camera locations equally spaced from 0° to 90° in azimuth. For the purposes of this

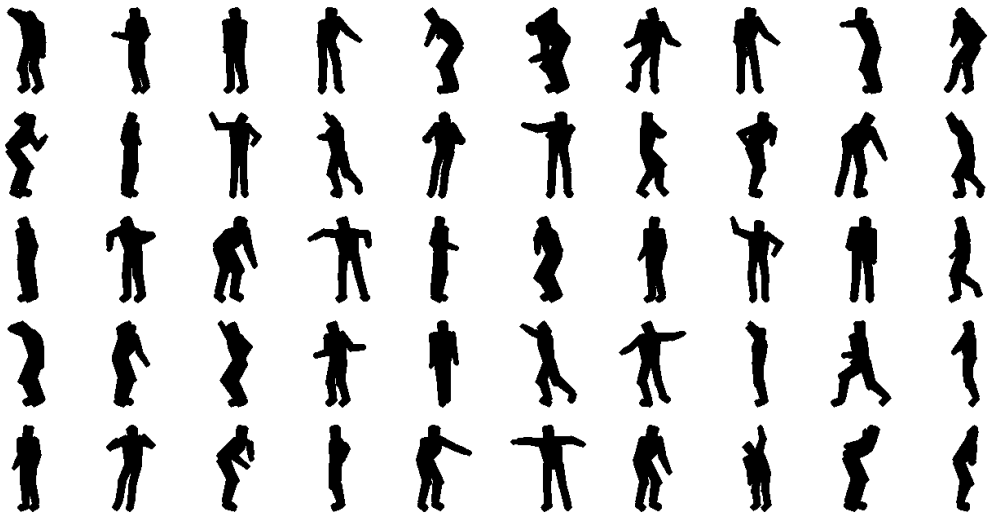


Figure A.1: Example silhouettes from the synthetic dataset.

comparison, every image was explicitly normalized by translating and scaling each silhouette such that it lay within the central 90% of the image. Furthermore, we assumed that the subject was upright in the image to avoid any need for rotation invariance – any exceptions to this rule (*e.g.* handstands, cartwheels) are explicitly modelled in the dataset.

Although silhouettes are generally restricted to scenes with a static camera and known background, and useful image data (*e.g.* internal edges) are discarded, they are readily obtained from image data by background subtraction and are relatively invariant to clothing and lighting, making them a popular choice for such applications.

A.2.2 Evaluation method

Most content-based image retrieval tasks require *classification* of the query input such that stored examples of the same class are returned. As such, recovered exemplars are classed as positive or negative such that evaluation tools such as the Receiver Operating Characteristic (ROC) curve and Precision-Recall curve may be used to compare

retrieval accuracy between different shape descriptors.

In the context of human pose estimation, however, exemplars cannot be classified into 'positives' and 'negatives' since the underlying space is continuous. Therefore, in the context of the task (recovering examplars of similar underlying pose) we use the sum of squared errors between corresponding joint centre *projections*[1] in the image to compute the distance, $d(x_i, x_q)$ in pose space between each training example, x_i, and the query, x_q.

Given a query silhouette, we rank the training data in order of similarity to the query as quantified by the chosen shape descriptor, denoting the index of the closest training example by $r(1)$ and the furthest by $r(N)$. We then generate a curve, $f(k)$:

$$f(k) = \frac{\sum_{j=1}^{k} d(x_{r(j)}, x_q)}{k}, \tag{A.1}$$

indicating the mean distance to the query of the k highest ranking training examples for $k = 1 \ldots N$. For a qualitative evaluation of the performance of two shape representations, we compare the normalized curves k/N against $f(k)/f(N)$. An example is shown in Figure A.2.

Effectively, this can be seen as a measure of correlation between distance in state space and distance in feature space – a high correlation (desirable) produces low curves whereas low correlation produces high curves. In addition, we also indicate the expected curve for a random ranking of the training data (*i.e.* unity) and the curve for the best possible ranking.

[1] Using projected joint centres rather than their full 3D position avoids many (though not all) problems associated with 'kinematic flip' ambiguities [102] where very different poses give rise to very similar projected joint centres.

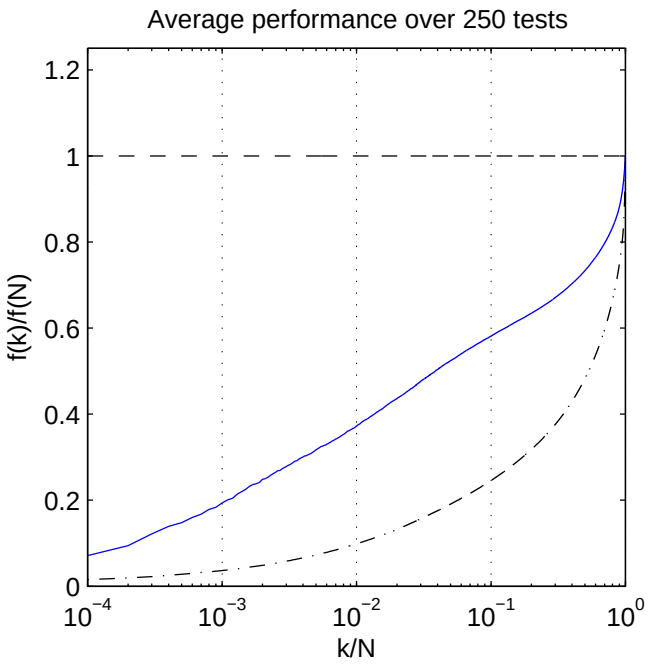


Figure A.2: Example graph of k/N against $f(k)/f(N)$. For comparison, the dashed line at unity indicates the average curve produced by random ordering whilst the dash-dot curve indicates the best possible ranking where distance in image space correlates perfectly with distance in pose space.

A.3 Shape representation

We now describe each representation and perform a number of experiments to determine the sensitivity of performance with respect to parameter values for each descriptor. For each descriptor, we aim to represent the original image by a 100D feature vector.

A.3.1 Linear transformations

We begin with linear transformations of the input image, namely geometric moments, orthogonal moments and PCA. Each feature, M_{pq}, is computed by convolving the entire image with a filter, f_{pq}, of equal size such that:

$$M_{pq} = \sum_{x} \sum_{y} I(x,y) f_{pq}(x,y). \tag{A.2}$$

Therefore, each feature is equal to the projection of the input image onto the basis 'vector' $f_{pq}(x,y)$. In PCA, the $f_{pq}(x,y)$ are the "eigenimages" corresponding to the directions of maximum variance. Moments, however, can be factored further such that:

$$f_{pq}(x,y) = f_p(x) f_q(y) \tag{A.3}$$

and

$$M_{pq}(x,y) = f_p(x) I(x,y) f_q(y). \tag{A.4}$$

For an image of size $P \times Q$, the moments employed in this study take the following functional forms:

- **Geometric moments:** $f_p(x) = x^p$
- **Krawtchouk moments:** $f_p(x) = \sum_{k=0}^{\infty} \frac{(-p)_k (-x)_k}{(-P)_k} \cdot \frac{2^k}{k!}$
- **Tchebishef moments:** $f_p(x) = (1-P)_p \sum_{k=0}^{\infty} \frac{(-p)_k (-x)_k (1+p)_k}{(1)_k (1-P)_k} \cdot \frac{1}{k!}$
- **Discrete Cosine Transform:** $f_p(x) = \sqrt{\frac{1+\min(p,1)}{2P}} \cos(\frac{2x+p\pi}{2P})$

where

$$(a)_k = a(a+1)(a+2)\dots(a+k-1) \tag{A.5}$$

is the Pockhammer symbol.

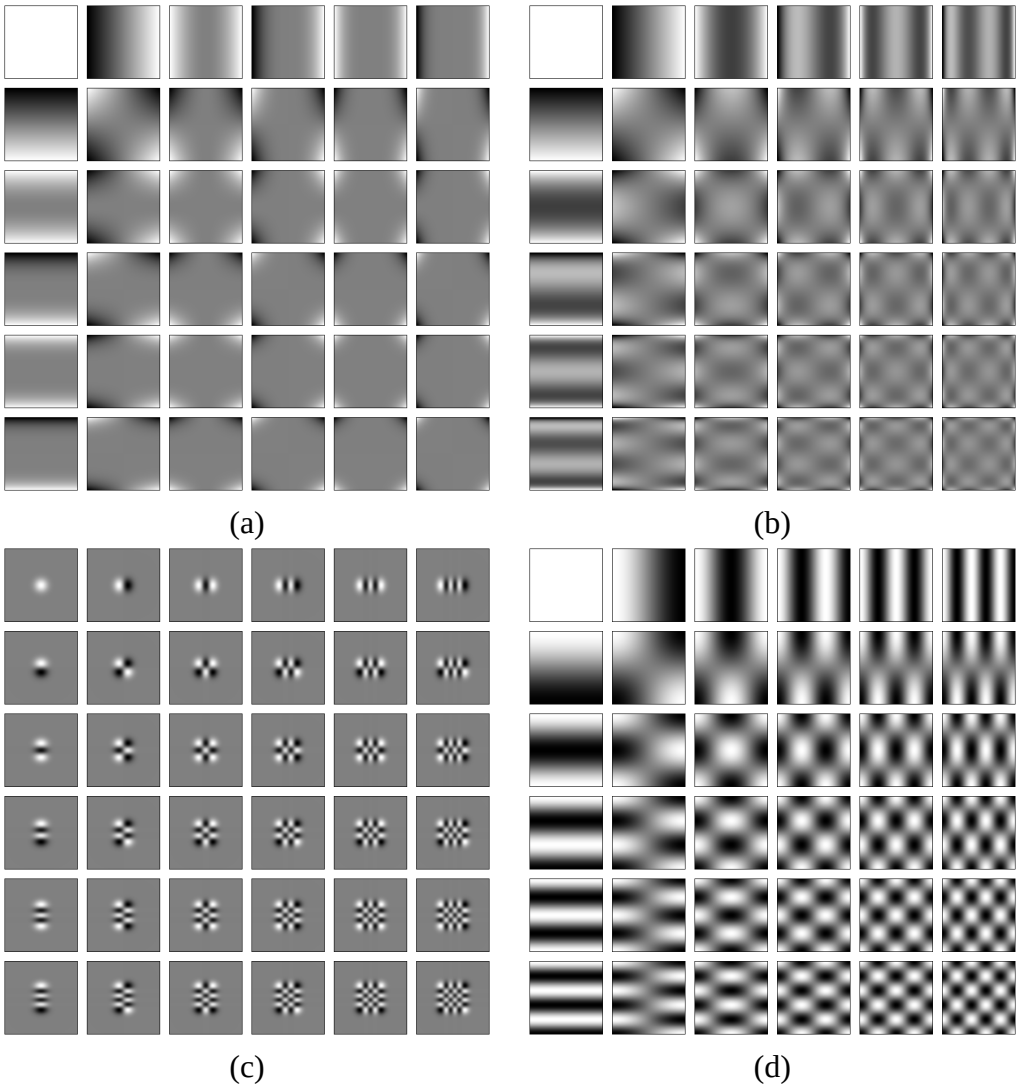


Figure A.3: Filter bank equivalents of moment generating functions up to order 5: (a) Geometric, (b) Tchebishef, (c) Krawtchouk and (d) DCT.

The latter three moments are known as *orthogonal moments* due to the following property:

$$\int f_p(x) f_q(x) dx = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases} \tag{A.6}$$

that results in low correlation between the coefficients such that fewer are required to describe the image within a given error bound.

Importantly, the orthogonal moments can be considered as a rotation of the vectorized image such that the Euclidean distance between feature vectors is equal to the sum of squared error between the original images. Orthogonal moments can also be considered as a generic basis set for the low dimensional approximation of images, as opposed to PCA that is data-dependent, computing the optimal basis for a given set of images. Figure A.3 shows the filter bank equivalent of the moment generating functions described.

Geometric moments have a mechanical interpretation in that each 'on' pixel represents a small mass in the image such that the moments correspond to total mass, centre of gravity, moments of inertia *etc*. However, since the geometric moments are not orthogonal, they do not represent a rotation of the vectorized image such that no intuitive distance metric exists between feature vectors.

Filter type

In the first test (Figure A.4a), we compared the different choices of transforms. We see that the geometric moments perform very poorly – a not unexpected result since as the order increases the moment function becomes dominated by points at the boundary of the image that typically contain little useful information (see Figure A.3). Furthermore, simple distance metrics such as the Euclidean distance (used in this example) are inappropriate for moments with a high dynamic range such as the geometric moments.

In contrast, Tchebishef moments and the Discrete Cosine Transform perform well in this test. These filter banks are qualitatively similar, representing an approximate frequency decomposition of the image. However, like the geometric moments, Tchebishef moments are weighted more heavily at the boundary of the image which may explain their inferior performance when compared with the DCT. Krawtchouk moments per-

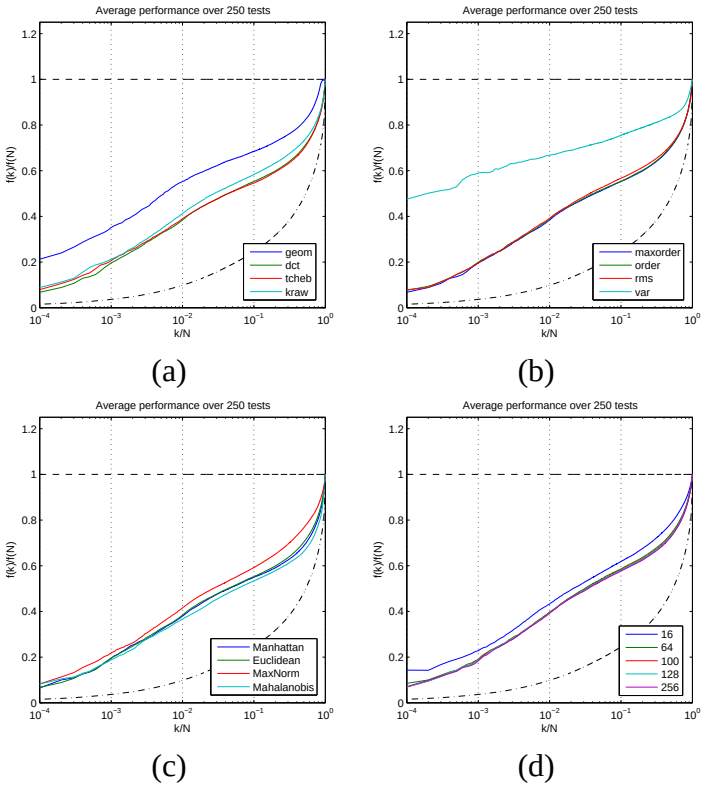


Figure A.4: Linear transform comparison. (a) Choice of moment; (b) Selection of features; (c); Distance measure; (d) Number of PCA coefficients used.

form only slightly less well, probably due to their limited spatial support over the image.

Feature selection

In the second experiment (Figure A.4b), we investigate several heuristics for feature selection using the DCT filter bank. Since there are as many features as pixels in the image, we must select a subset of the computed features in order to reduce the dimensionality of the feature vector. In general, feature selection is a highly complex task that is beyond scope of this thesis. For the purposes of these experiments, we select features based on heuristics such as maximum order (max(m, n)), order ($m+n$), RMS value and variance. We see that variance is a poor indicator of feature 'information'.

Distance metric

In the third experiment (Figure A.4c), we compare different distance metrics for ranking the database in order of similarity to the query in feature space. Although the Mahalinobis distance outperforms the other distance metrics, the improvement is small at additional computational cost. We also note that Euclidean distance is the most intuitive metric to use since the distance between feature vectors approaches the true Euclidean distance between the original images as the number of features increases. However, there is little penalty in accuracy when using the Manhattan (L_1) distance between feature vectors – a somewhat cheaper alternative.

Number of features

For the final investigation (Figure A.4d), we compared performance for varying numbers of principal components used to compute the feature vector. The graph suggests that there is little improvement above 64 features – well within the 100D limit we have imposed. We note that PCA is a computationally expensive method for feature extraction. In fact, only an approximation can be computed in this case since the full evaluation requires the inversion of a 16384×16384 matrix that is an infeasible task on current hardware. In contrast, the DCT is efficient to compute without sacrificing accuracy, as can be seen by inspection of the graphs in Figure A.4a and Figure A.4d.

A.3.2 Hu moments

Alternative descriptors for shape matching and retrieval include the Hu moments [52], popular due to their invariance to translation, scale and rotation. However, since they are based on the geometric moments they too lack an intuitive interpretation and distance metric. Furthermore, only seven moments are typically defined making them

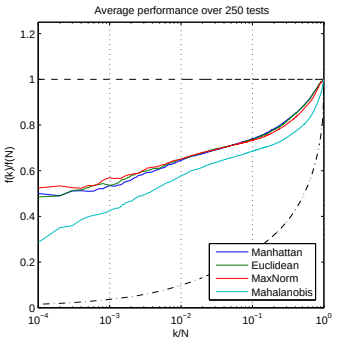


Figure A.5: Hu moments. Note that only seven moments are typically available, resulting in inferior performance compared with the other selected descriptors.

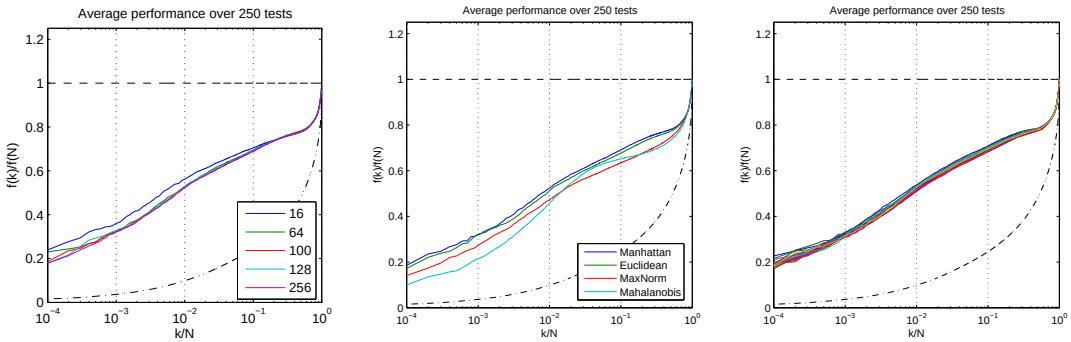
difficult to compare with richer descriptors. Although we have eliminated the need for invariance, we include the Hu moments for completeness.

Distance metric

It is clear from Figure A.5 that the performance of the Hu moments suffers badly due to the limited number of features that are available (only seven in this case). Furthermore, due to the high dynamic range of the Hu moments, Euclidean distance is a poor distance metric to use. Instead, the more computationally complex Mahalanobis distance is required for adequate performance.

A.3.3 Lipschitz embeddings

The final global representation we include is that of the Lipschitz embedding, describing an image by its distance from a number of 'pivot' examples as demonstrated for hand tracking [8]. Intuitively, images that are close together in image space have similar distances to the pivot examples and therefore will have similar feature vectors. However, it is again difficult to identify an intuitive distance metric between two feature vectors.



(a) (b) (c)

Figure A.6: Lipschitz embeddings. (a) Number of pivot exemplars; (b) Distance metric; (c) Initialization.

Number of pivot exemplars

In the first experiment (Figure A.6a), we investigate the effect of increasing the number of pivot examples *i.e.* the dimensionality of the feature space. As with other descriptors, increasing the number of features improves performance up to a point with little improvement above 100.

Distance metric

In contrast to other descriptors, Figure A.6b shows that Mahalanobis distance and the Max-Norm provide the best performance, offering a noticeable improvement over the Euclidean and Manhattan norms. Why this should be the case is unclear and may merit further investigation.

Initialization

In a separate experiment (Figure A.6c), selecting 100 different sets of 100 pivot examples and comparing the resulting curves suggested that performance is largely insensitive to initialization. Intuitively, some selections will perform better than others. For example, pivots from the same region of space will result in highly correlated (and

hence redundant) features leading to poor performance. Conversely, careful selection of pivots may improve performance beyond that shown here (although by how much is hard to say). Again, however, we note that such feature selection is beyond the scope of this work and has been tackled using Machine Learning methods such as Boosting [7].

A.3.4 Histogram of Shape Contexts

We now examine a *local* descriptor – the Histogram of Shape Contexts (HoSC) suggested by Agarwal and Triggs [5]. In this descriptor, every point along the contour of the image is assigned a Shape Context vector [13] representing the distribution of other contour points in a local neighbourhood. A number of Shape Contexts are selected at random from the whole set and used as initial centres in a clustering scheme. Having clustered the training Shape Context vectors, the updated centres are used as a vector quantization 'codebook' in order to assign every contour point on a silhouette to a cluster. The histogram over cluster assignments then forms the feature vector for a given silhouette. As a result of the complexity, this representation has a high number of parameters and is more expensive to compute, particularly during off-line clustering.

This approach is considered to be local since, if corruption of the silhouette is localized to a relatively small region of the silhouette, most of the remaining contour points will (in theory) vote into the same bins of the histogram such that the change in the feature vector is localized to only a few features. It is this property of 'locality' that is cited as a beneficial attribute of this shape representation. However, this justification of the HoSC can be questioned for a number of reasons:

- In most cases the corruption of the silhouette results in an increase or decrease in the total number of contour points (*e.g.* due to shadows or occlusion) such that

Figure A.7: Escher's 'Angels and Demons'. Since both the angel and the demon are composed of exactly the same contour segments, they have similar feature vectors using the Histogram of Shape Contexts. The feature vectors become identical as the spatial extent of the Shape Context decreases toward zero.

upon normalizing the histogram *every* bin is affected and locality is lost.

- Typical distance metrics (*e.g*. Euclidean, Manhattan, Mahalanobis distances) do not distinguish between a large change in a few bins of the histogram and a small change in every bin. As a result, it is unclear that the property of locality provides any real benefit when using such metrics.

- For every contour point the distribution of other *contour points* is computed. As a result, no explicit distinction is made between the interior and exterior of the silhouette, thus discarding yet more information. As an example, consider the 'angel' and 'demon' of the Esher tesselation shown in Figure A.7. In the limit as the spatial extent of the shape context vector approaches zero, the two shapes are indistinguishable as they are composed of the same contour segments. The method may be modified to take this matter into account albeit at additional computational expense.

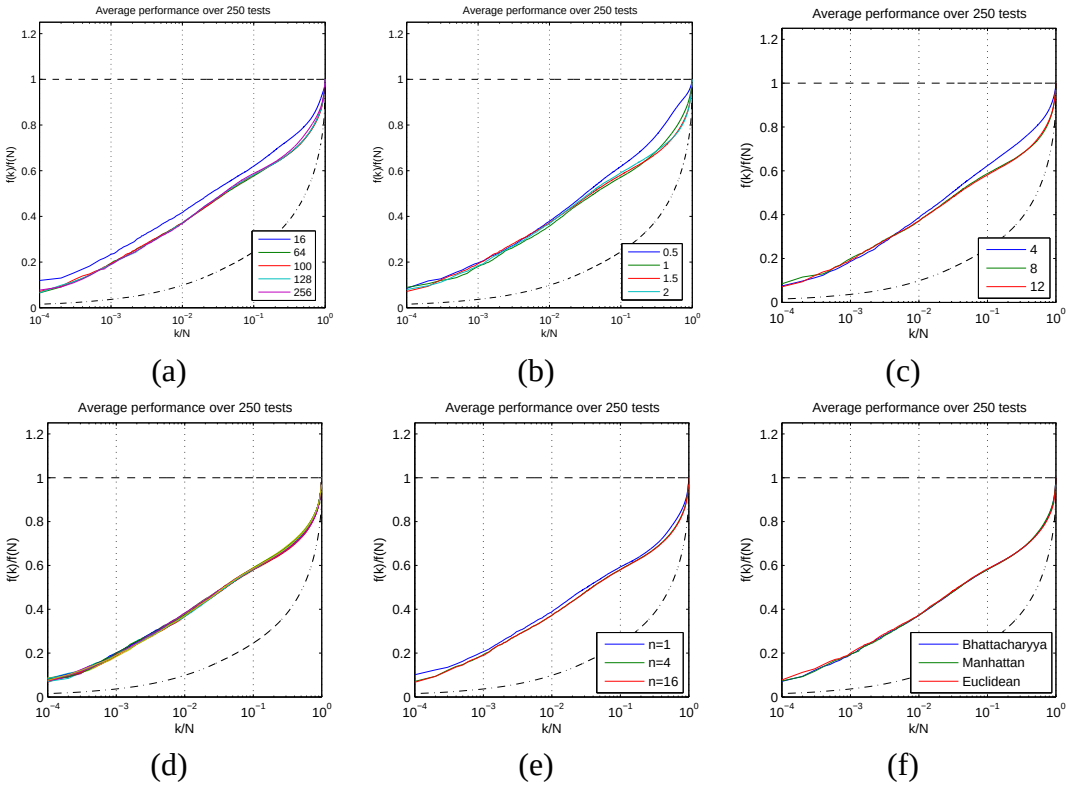


Figure A.8: Histogram of Shape Contexts. (a) Number of codebook vectors; (b) Spatial extent of Shape Context; (c) Number of angular bins of Shape Context; (d) Initialization; (e) Number of bins 'softly' voted into for histogram computation; (f) Distance metric.

Number of codebook vectors

As with the other descriptors, we evaluate how the number of features (codebook vectors in this case) affects performance (Figure A.8a). Similarly, it can be seen that above around 64 features there is little further improvement. However, using fewer codebook vectors provides other benefits such as reducing computational expense.

Shape Context spatial support

The Shape Context vector takes three parameters: spatial extent (*i.e.* radius); radial bins; angular bins. We define the spatial extent of the Shape Context by a multiple of the mean distance between all points on the contour. We see in Figure A.8b that

performance is largely invariant to this value although too small a value does degrade performance as k increases. Performance is also shown to improve with the number of angular bins (Figure A.8c) although above 8 bins the benefit is small. A similar experiment (unshown) suggests that the number of radial bins does not adversely affect performance.

Initialization

We examine the effect of selecting different Shape Context vectors from the training set to serve as centres during clustering. As with Lipschitz embeddings, Figure A.8d shows that performance is largely unaffected by the initialization. This is likely due to the large number of Shape Context vectors available such that the cluster centres converge to approximately the same values each time.

'Soft' voting

In [5], it is suggested that a 'soft' voting scheme be employed to avoid quantization effects. We examine the effect of this mechanism by voting into an increasing number of bins. Figure A.8e shows that there is merit in soft voting although benefits are diminished for more than 4 bins.

Distance metric

Finally, we examine performance under different distance metrics. Although there exist intuitive distance measures for histograms (*e.g.* Bhattacharyya distance, cross entropy), Figure A.8f shows that other metrics such as Manhattan and Euclidean distance work equally well. This may be as a result of the soft voting, as suggested in [5].

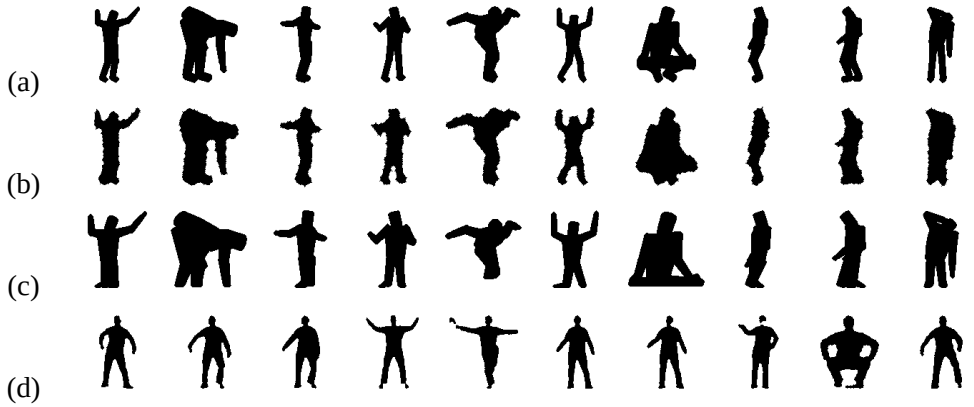


Figure A.9: Four test datasets: (a) clean silhouettes; (b) with added noise; (c) with lower quarter removed; (d) real silhouettes manifesting some segmentation error.

A.4 Final comparison

Having performed extensive tests to select parameter values for the various presented methods, we now undertake a comparison of performance for three of the four selected methods: Discrete Cosine Transform coefficients; Lipschitz embeddings; Histogram of Shape Contexts.

In order to compare the methods, curves were generated for four test datasets: perfect, clean silhouettes; noisy silhouttes; silhouettes with occlusion; real silhouettes. Each dataset is described in more detail below and examples are shown in Figure A.9.

A.4.1 Clean data

We begin by comparing the three methods for clean data (Figure A.9a) taken directly from the synthetic dataset. Figure A.10a shows that the Histogram of Shape Contexts method exhibits slightly better performance than DCT coefficients for small values of k, although for higher values of k the situation is reversed. Lipschitz embeddings are less successful in this test.

A.4.2 Noisy data

To create the noisy dataset, we corrupted the clean test silhouettes with gaussian noise along the occluded contour (Figure A.9b). Such corruption typically results from compression artefacts and segmentation errors at the boundaries. From Figure A.10b, we see that DCT coefficients outperform both other methods on average. This can be explained by the fact that lower order DCT coefficients, as used in this case, encode only the lower frequencies within the image and thus suppress noise.

A.4.3 Occluded data

In order to simulate occluded data, we removed the bottom quarter of each test silhouette and renormalized, as if the subject had been obscured from approximately the knee down (Figure A.9c). Although this is a relatively crude approach, it presents each method with data that is somewhat different from the training data and may occur in real life. Figure A.10c shows that the Histogram of Shape Contexts again performs well for small k but is outperformed for higher k by the DCT. Lipschitz embeddings perform poorly, hovering around the performance of a random ranking.

A.4.4 Real data

For the final experiment, we use real silhouettes from the starjumps sequence (Figure A.9d), obtained via background subtraction and with projected joint centres labelled by hand. Due to the limited number of test images, the curves in Figure A.10d are slightly noisier but suggest that DCT coefficients significantly outperform both Histogram of Shape Contexts and Lipschitz embeddings. On average, in fact, HoSC and Lipschitz perform worse than random for this dataset.

This is a surprising and interesting result, particularly since this is arguably the most

important test set of the four. There is a question of whether the normalization procedure employed in these experiments could favour one method over another. However, the test silhouettes show little corruption that would have a significant effect on this process. A closer inspection of the output data may provide fruitful insights into the reasons behind the poor performance of the HoSC with respect to DCT coefficients.

A.5 Summary

This appendix has presented a rudimentary comparison of selected shape representations for the specific task of human pose estimation from a corpus of training data. For each selected representation, performance was evaluated with respect to parameters before a final comparison was undertaken on synthetic and real datasets.

The results of the comparison suggest that the recently proposed Histogram of Shape Contexts [5] has little or no benefit over more 'primitive' shape representations, despite its complexity. Furthermore, the complexity of the descriptor makes it highly inefficient in terms of computation. Comparable results were achieved using coefficients of the Discrete Cosine Transform (DCT) to generate the required feature vector.

This comparison was performed principally as justification for the use of the DCT coefficients in Chapter 4 and is not intended as a comprehensive review.

A.5.1 Future work

Future research could investigate other shape representations although many are ruled out by the nature of the dataset. However, the principal line of inquiry should be focussed on the surprisingly poor performance of HoSC descriptor for real data. This was an unexpected result and should be investigated further to gain a deeper insight into the relationship between the data and the descriptor.

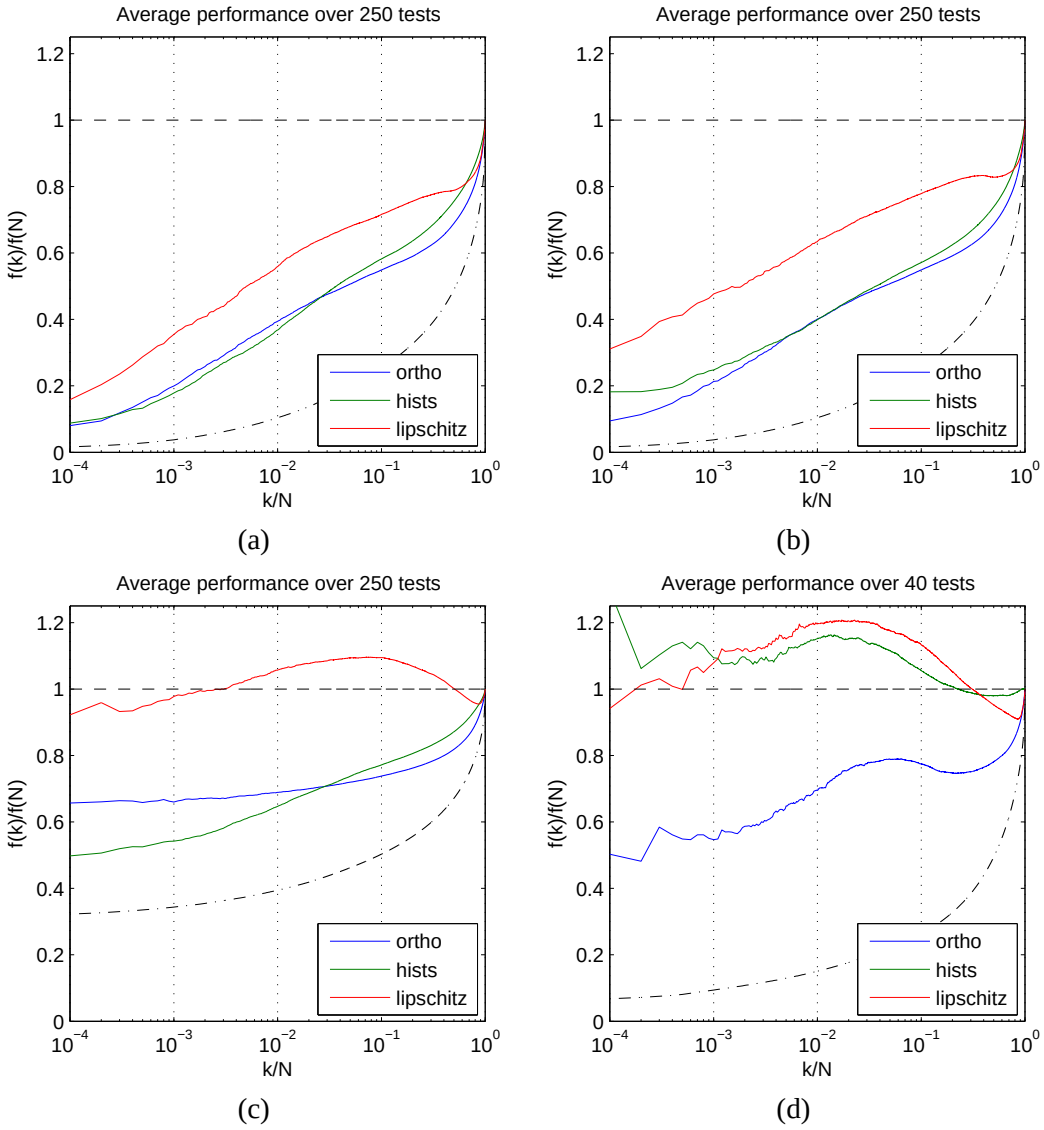


Figure A.10: Results for (a) clean data; (b) noisy data; (c) occluded data; (d) real data.