MIXED SCALE MOTION RECOVERY

James Edward Davis, Ph.D.
Stanford University, 2002.

Advisor: Pat Hanrahan

Motion frequently occurs at mixed scales. Subtle localized motion often takes place together with long range movements. For instance, the pose of a soccer player is contained in a much smaller volume than the entire playing area, and the expression on an actor's face is at a much smaller scale than the room in which the actor moves. Recovering these motions requires high resolution imagery suitable for resolving small details. It simultaneously requires increasing the working volume in which recovery is possible. Traditional motion capture has concentrated on single scale domains. This dissertation presents a framework for discussing mixed-scale motion recovery systems, and proposes a specific system design.

Mixed scale motion can be recovered with a foveated tracking system. Such a system has multiple stages. A wide area tracking subsystem makes use of many cameras to robustly localize motion. A small scale tracking subsystem makes use of narrowly focused cameras to recover motion with high accuracy. These latter cameras are mounted on pan-tilt heads and can be dynamically oriented in the correct direction to provide high resolution coverage of the active region. This system is demonstrated on a specific application of practical interest to the entertainment industry, capturing simultaneous body and facial motion.

Approved for publication:

By: _____

For Department of Computer Science

MIXED SCALE MOTION RECOVERY

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

James Edward Davis

June 2002

© Copyright by James Edward Davis 2002
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Pat Hanrahan – Principal Advisor

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Christoph Bregler

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Gene Alexander

Approved for the University Committee on Graduate Studies

Abstract

Motion frequently occurs at mixed scales. Subtle localized motion often takes place together with long range movements. For instance, the pose of a soccer player is contained in a much smaller volume than the entire playing area, and the expression on an actor's face is at a much smaller scale than the room in which the actor moves. Recovering these motions requires high resolution imagery suitable for resolving small details. It simultaneously requires increasing the working volume in which recovery is possible. Traditional motion capture has concentrated on single scale domains. This dissertation presents a framework for discussing mixed-scale motion recovery systems, and proposes a specific system design.

Mixed scale motion can be recovered with a foveated tracking system. Such a system has multiple stages. A wide area tracking subsystem makes use of many cameras to robustly localize motion. A small scale tracking subsystem makes use of narrowly focused cameras to recover motion with high accuracy. These latter cameras are mounted on pan-tilt heads and can be dynamically oriented in the correct direction to provide high resolution coverage of the active region. This system is demonstrated on a specific application of practical interest to the entertainment industry, capturing simultaneous body and facial motion.

Acknowledgements

The list of people who have supported, encouraged, or influenced this work is just too large to give in detail. However there are three people that I want to single out as 'most influential' regarding the fact that this document exists. — Pat Hanrahan, my advisor, has been an exceedingly bright beacon of what high quality research should be. His genuine scientific curiosity and ability to quickly determine core intellectual challenges has been a constant source of inspiration. It has perhaps been more natural for me to achieve 'curiosity' than anything else, and I want to express my gratitude for the considerable freedom Pat has given me to explore seemingly unrelated but none-the-less interesting ideas. While all graduate students take meandering paths to enlightenment, support for *my* path amounts to genuine kindness. — Xing Chen, my research partner, collaborated on all of the research contained in this document. None of it would have been possible without her contributions. In addition to countless research discussions, she provided the primary inducement to focus on one topic long enough to complete it. Lastly, she has become a close friend, and her support has been responsible for my sanity through many personal joys and crises. — Venkat Krishnamurthy played the role of friend and advisor at just the right time to keep me from abandoning the graduate program. Although it was with regard to a research topic unrelated to this dissertation, he provided the needed encouragement, support, and ideas that gave me hope during a critical period.

Pat Hanrahan, Brian Wandell, Chris Bregler, Gene Alexander, Marc Levoy, Cindy Chen, Ada Glucksman, Heather Gentner, Erika Chuang, Homan Igehy, Venkat Krishnamurthy, Tamara Munzner, François Guimbretière, Szymon Rusinkiewicz, Maneesh Agrawala, Lucas Pereira, Kari Pulli, Shorty, Sean Anderson, Reid Gershbein, Philipp Slusallek, Milton Chen, Mathew Eldridge, Natasha Gelfand, Olaf Hall-Holt, Humper, Brad Johanson, Sergey Brin, Dave Koller, John Owens, Kekoa Proudfoot, Kathy Pullen, Bill Mark, Dan Russel, Larry Page, Li-Yi Wei, Gordon Stoll, Julien Basch, Andrew Beers, Hector Garcia-Molina, Brian Freyburger, Mark Horowitz, Chase Garfinkle, John Gerth, Xie Feng, Craig Kolb, Toli, Mom, Dad, Holly Crawford, Chris, Crystal, Lara, Grace Gamoso, Matt Hamre, Nancy Schaal, Aaron Jones, Bandit, Xiaoyuan Tu, Abigail, Shefali, Liza, Phil, Deborah, Brianna, Poncho, Cecile, Chayla, Alejandra, Ms Dungan, Gabe, Sedona, Sharon, Gonzalo, and many children whose names I can no longer remember

Table of Contents

Abstract iv

Acknowledgements v

1 Introduction 1
- 1.1 Existing technology 2
- 1.2 Applications of motion recovery 3
- 1.3 Mixed scale domains 4
- 1.4 Overview of system 5
- 1.5 Desirable system properties 7
- 1.6 Related work 9
- 1.7 Contributions 19
  - Framework for mixed scale motion recovery. 19
  - Specific system design. 19
  - Application to simultaneous face-body recovery. 19

2 Framework 20
- 2.1 Problem domain characterization 20
- 2.2 Hierarchical paradigm 22
- 2.3 Interface requirements 25
- 2.4 Data flow analysis 27
- 2.5 Model based solutions 30

3 Large scale recovery 32
- 3.1 Physical setup 34
- 3.2 Calibration 37
- 3.3 Feature detection 39
- 3.4 Integrating observations 47
- 3.5 Time and communication 51
- 3.6 Adding and removing subsystems 54
- 3.7 Evaluation 59

4 Sub-region selection 68
- 4.1 Physical setup 68
- 4.2 Calibration 71
- 4.3 Directing pan-tilt cameras 76
- 4.4 Latency 77
- 4.5 Evaluation 80

5 Small scale recovery **84**
5.1 Feature detection 85
5.2 Integrating observations 87
5.3 Occluded features 89
5.4 Face model 90
5.5 Evaluation 95

6 Conclusions **100**
6.1 Applications 100
6.2 Contributions 104
6.3 Future Work 105

References **107**

List of Illustrations

Note that some illustrations are marked as *Video figure*. These figures have associated video content which can be viewed on the supplementary CD-ROM or online at:

`http://graphics.stanford.edu/papers/MixedScaleMotionRecovey`

Figure 1 Example mixed scale environment. The motion of players in a soccer stadium occurs in a very large working volume. However the motion itself is extremely detailed relative to the size of the space. 1

Figure 2 Example of existing body motion capture system. A common use of existing motion capture systems is the recovery of an actor's body motion. 2

Figure 3 Example of existing facial motion capture system. Facial expressions can be captured by placing a set of markers on an actors face, and observing the motion of these markers with a set of cameras. The motion can later be used to drive the expression of an animated character. 3

Figure 4 Overview of system design. A multi-camera large scale motion recovery system identifies targets, and directs a set of pan-tilt cameras which obtain high resolution video of the desired target. 6

Figure 5 *Video figure* Motivational example from demonstration system. As a person moves through a room sized environment, body and facial motion are simultaneously recovered. (Bottom left) The recovered large scale body motion model. (Bottom right) The recovered facial motion model. 7

Figure 6 Areas of research related to the broad goals of this thesis. None of these areas addresses all of the desired design goals, although each does introduce technologies from which this work borrows. 9

Figure 7 Typical equipment used by marker based motion capture systems. Retroreflective markers and a camera used to track them. The ring of infrared LEDs surrounding the camera lens insure that a bright light source exists which can be reflected directly back into the camera. 12

Figure 8 Example of markerless motion capture technology. Systems such as this can recover movement without augmenting the scene. Unfortunately they tend to be specialized and slow, making them unsuitable for use in a real-time application. (Left) from Bregler and Malik (1998). (Right) from Cham and Rehg (1999). 14

Figure 9 Typical equipment used with magnetic motion capture systems. Left. Sensors used for magnetic motion capture. Right. The tangle of wires often associated with this technology. 15

Figure 10 Example exoskeleton based motion capture technology. These systems use a mechanical structure fitted onto the actor's body. 17

Figure 11 Motion of players in a stadium occurs at multiple scales. The position on the field, the players body pose, and the players facial expression are three possible scales we might wish to consider.....21

Figure 12 A hierarchical paradigm can be used to express mixed scale motion recovery. At each scale, recovered motion drives sub-region selection. The selected sub-region in turn defines the next scale. 22

Figure 13 The specific system described in this dissertation is one example of using a hierarchical paradigm. This system has two stages of hierarchy. Large scale motion recovery is used by the sub-region selection mechanism to direct pan-tilt cameras towards the target. The video from these pan-tilt cameras becomes the basis for small scale motion recovery. 23

Figure 14 Expanded system overview. Cameras observe LED point features in the environment. These point features are detected and merged to recover the large scale pose of the target. This large scale pose is used to guide pan-tilt cameras to point at the target. These cameras in turn observe features which are detected and merged to recover the small scale pose of the target. 25

Figure 15 System overview when viewed as a sequence of data flow. Video streams are transformed into streams of image feature points. These data points are converted into a stream of target poses. The pose data is used to derive a sequence of camera parameters, which in turn generate a new set of video sequences. The video from each pan-tilt camera is transformed into a new sequence of feature points, and then into small scale object pose. ..27

Figure 16 System challenges as instances of poor quality or noisy data. Each challenge is a place where the data needs to be cleaned up before it is suitable as an interface to the next component in the system pipeline.....29

Figure 17 Models can be used to address challenges that arise due to inadequacies in the data flowing between interfaces in the system pipeline. These models constrain, filter or regularize the data so that it is suitable for the next processing component. 30

Figure 18 Cameras mounted on the ceiling of a laboratory are used to capture large scale motion. 33

Figure 19 An LED light source is a useful feature to track, since detecting it against a wide variety of backgrounds is easy and robust. 33

Figure 20 Diagram of cameras observing a single target. This is a view from the side, ceiling mounted cameras appear at the top of the figure. Each camera generates a ray in space. These rays can be triangulated to determine the target's 3D location. 34

Figure 21 The arrangement of cameras used for large scale motion recovery, shown from above. 35

Figure 22 (Left) Cameras are connected to machines through a patch panel in order to provide configuration flexibility. (Right) The stacks of Indy's shown here, as well as Wintel PCs, are used to digitize video and locate features. 35

Figure 23 Two independent arrangements of cameras were combined to produce the current configuration. On the left is an arrangement designed for head tracking in front of a display wall. On the right an arrangement used for experiments on occlusion probability. 36

Figure 24 Path of a feature used to create a virtual calibration object. Note that this path covers the entire working volume in a way that would be difficult using a physical calibration object. 38

Figure 25 Convergence of proposed calibration method. Iterations vs. projection error. It is clear that this iterative method of camera calibration converges to a good estimate of camera calibration. 39

Figure 26 Example target locations. Features are attached to a person at positions that will allow recovery of the desired motion. 40

Figure 27 Feature observations from several views. Each camera reports the point features it observes. Since the camera viewpoints vary, the number of observed points and their locations also vary. 40

Figure 28 Thresholding to detect target features. Image pixels brighter than some threshold value are marked as belonging to a feature to be tracked. Left. Observed image frame with enlarged image in the region of target features. Right. Thresholded image indicating feature locations. 42

Figure 29 A screenshot from an interactive visualization of current threshold parameters and camera observed intensities. In this case a single feature in the camera view exceeds threshold parameters. 43

Figure 30 Features reported by a particular cameras. This camera is reporting many false features in the region of a monitor. By selecting this region as invalid, false features can be eliminated. 45

Figure 31 Marked invalid regions. Regions of the camera view frustum which correspond to sources of noise have been marked as invalid, and false features are no longer present. 45

Figure 32 Visualization of which pixels have been searched for target features. Every pixel is checked within a window around the predicted feature location. Sparse subsampling of the entire frame is used to identify new features. 47

Figure 33 Intersecting rays can be used to determine the three dimensional location of targets. (a) A 3D view of a set of camera observation rays for just one target. Colored dots are cameras mounted on the ceiling, and each line represents a camera observation. (b) The set of observing rays for multiple targets. (c) The target locations extracted by intersecting these rays. 48

Figure 34 Rather than determining target location separately at each time instant, a motion model can be used to maintain an estimate of target location. One of the benefits of using a motion model is that consistent target IDs can be maintained. Here we see target locations with their associated IDs. 50

Figure 35 Camera starvation. This sequence illustrates how only a small clock discrepancy can create a situation where all remaining cameras are effectively starved. Camera A observes the target at time 2ms, but because of a 4ms clock discrepancy tags the observation as occurring at time 6ms. This observation is then integrated into the Kalman filter. Camera B observes the feature 1ms later at time 3ms, and correctly time stamps the observation. An attempt is made to integrate this observation into the state of the Kalman filter, but because it is tagged as older than the current state, it is discarded. A similar situation arises with Camera C. Since all cameras will continue to make observations at precisely 60Hz, the situation will repeat and cameras B and C will effective starve, due to a relatively small clock discrepancy on the part of camera A. 53

Figure 36 Observed path of an LED from nine camera viewpoints as it was moved along a controlled linear path. 59

Figure 37 Noticeable noise inflicts the observed path of an LED moved along a linear path. These plots are in the space of the XY image plane. 60

Figure 38 Plot of observed LED position. X-axis and Y-axis motion have been separated into separate time series. It is clear that a great deal of the noise is due to integer quantization of pixel position. 61

Figure 39 Pixel error of observed target locations relative to ideal linear motion. Observations from the camera on the left clearly have more irregular noise characteristics than the camera on the right. This is due to natural variation in viewing angle and distance....62

Figure 40 Time series plot of target position reconstructed by the large scale capture system. The enlarged section of the graph shows jitter in the z-axis direction. 62

Figure 41 Target position reconstructed by the large scale system was fit to ideal linear motion. This plot shows deviation from the ideal line in millimeters. 63

Figure 42 Target position in millimeters, calculated with two different settings of Kalman filter parameters. By changing filter parameters it is possible to provide a range of estimated motion, from noisy but responsive, to smooth but over-damped. 64

Figure 43 Estimated path of a target moved in a circular motion. (Left) The correctly estimated path obtained with good settings of Kalman filter parameters. (Right) The target path is lost when Kalman filter parameter settings are configured with too strong a belief in a constant velocity motion model. The target is repeatedly lost, deleted, and re-instantiated. 65

Figure 44 Observed path of a target along with target locations predicted by the Kalman filter motion model. Blue dots are observations, and black dots with connecting lines are associtated predicted target location. (Top) Low strength motion model. The predicted path is noisy, but positions stay close to actual observations. (Middle) Medium strength motion prior. The predicted path is somewhat smoothed, but predicted positions more strongly follow prior velocity and lie further outside the observed motion. (Bottom) High strength

motion model. Predicted target positions too strongly follow prior velocity and tracking is lost, with targets continuing under constant velocity. 66

Figure 45 Schematic of the control setup for the pan-tilt cameras that perform sub-region selection. 69

Figure 46 Pan-tilt cameras mounted in our laboratory motion recovery area. 70

Figure 47 Top down view of pan-tilt camera placement. Cameras can be directed such that their view frustrums intersect at the desired target feature location. 70

Figure 48 Simple pan-tilt camera motion model. Pan and tilt are modeled as axis aligned rotations around the origin. 72

Figure 49 Improved model of camera motion. Pan and tilt motions are modeled as rotations around arbitrary axes in space. 73

Figure 50 *Video figure* This sequence shows the video stream from a pan-tilt camera as it is directed to follow a moving target. In the first column, the camera is directed without prediction. Note that the target often falls out of the video frame. In the second column prediction is used. In this case the target stays approximately centered in the video frame at all times. 79

Figure 51 The image plane projection error of measured data diminishes as the complexity of camera calibration increases. The pan-tilt calibration model proposed in this work has the smallest error. (Left) Error when calibrating using only observations and world space coordinates collected while the camera was in its home position. (Middle) Projection error using all data in conjunction with the traditional simple model of pan-tilt calibration. (Right) The error when using the model proposed in this work. 81

Figure 52 Histograms of projection error. The use of independent rotation axes, rather than axes orthogonally aligned with the image plane improves the fit between modeled projection and observation. (Left) Calibration of a single camera position, extrapolated to pan-tilt motion. (Middle) Calibration of camera while requiring aligned orthogonal axes. (Right) Calibration allowing camera and axes to move independently. 82

Figure 53 Deviation of a target from the center of the pan-tilt image frame as a function of time. Peaks in target deviation correspond to periods of high target acceleration. 83

Figure 54 The view from each of the four pan-tilt cameras. The cameras all look at the same point, but provide images from different angles, allowing triangulation of features. 84

Figure 55 *Video figure* The small scale recovery system uses painted marks as features for motion recovery. A close-up of features on the face is shown here. 85

Figure 56 A screen shot of a tracking session. Automated tracking sometimes fails, so a user interface is provided to guide the tracking process in these cases. Above are the controls and an overview of the current tracking state. Below is an enlarged view so that the tracking progress of a particular feature can be monitored. 86

Figure 57 Small scale features can be reconstructed by triangulating observations from two viewpoints. 88

Figure 58 *Video figure* Recovered 3D face geometry shown from several viewpoints. 89

Figure 59 Reconstructed face. Since half of the features are missing from one of the viewpoints, these features can not be reconstructed directly. 89

Figure 60 A partial set of features can be used to fit a facial model. This model can be used to reconstruct the location of missing features. 91

Figure 61 A linear combination of valid basis faces can be used to model the range of all possible feature motion. 92

Figure 62 *Video figure* Still frames from a sequence of recovered facial geometry. 96

Figure 63 Facial motion recovered by the small scale recovery system, during one capture sequence. The global head motion is on the order of two meters. 96

Figure 64 Distance between facial features. The distance between the green pair of features on the lips varies over time, while the distance between the red pair of features on the forehead remains relatively constant. The plots on the right show actual distance deviation during one motion recovery session. In addition to the lower right plot, the median accuracy of the small scale recovery system was quantified to be 0.9 mm by evaluating the distance deviation between all pairs of features marked in blue. 97

Figure 65 The quality of reconstructed features can be evaluated by synthetically removing features to simulate occlusion. After reconstruction, these features are compared to ground truth data and found to vary by less than 1.5 mm from their original location. 98

Figure 66 A plot of error, measured as the distance between reconstructed missing features and the ground truth location of those features. The error is shown projected onto the frontal, XY, plane of the face, in mm. 99

Figure 67 *Video figure* A sequence of frames showing the quality of reconstruction as successively more feature points are occluded. Even with many occluded features, the reconstructed face remains believable. Right. Data acquired via triangulation, many features are occluded. Left. Reconstructed complete facial model. 99

Figure 68 *Video figure*. Live action of simultaneous body and facial motion. The subjects large scale motion causes the sub-volume within which small scale facial motion occurs to traverse most of the global working volume. 101

Figure 69 *Video figure*. Recovered large scale body motion. The three dimensional motion of each target has been recovered, and is used here to visualize the overall motion of the subject. 101

Figure 70 *Video figure*. Recovered small scale body motion. The motion of each facial feature was recovered, and the entire data set fit to a facial model. This model was used to reconstruct the motion of any occluded features. 101

Figure 71 Still frames from the recovered motion of knee deformation, while a subject walks across a room. 102

Figure 72 Mixed scale structured light system. A static projector and a dynamically oriented camera are used to capture geometry via triangulation of known projected stripe patterns. 103

Figure 73 Using the large scale motion recovery sub-system to track a users head position ensures that the correct projection of a virtual object can be rendered on a wall size display. 103

Figure 74 Users interact with a wall size display using a scalable pointer based input technology. The architecture of this input system draws directly from the large scale tracking presented in this work. 104

1 Introduction

Motion occurs all around us, continuously as part of daily life. By watching a mechanical object's motion we can learn how it works, and by watching a dancer's motion, we can see beauty and be inspired. Yet computer models of the world are often static, representing the shape, locations, or qualities of objects. Motion recovery is the process of capturing not just the static shape of an object, but its motion over time. These models of motion allow us to study the complexity and beauty of the natural world.

This dissertation addresses motion recovery in mixed scale environments, environments in which we desire very large working volumes and extremely detailed motion. For example, consider the motion of players in a soccer stadium as seen in Figure 1.

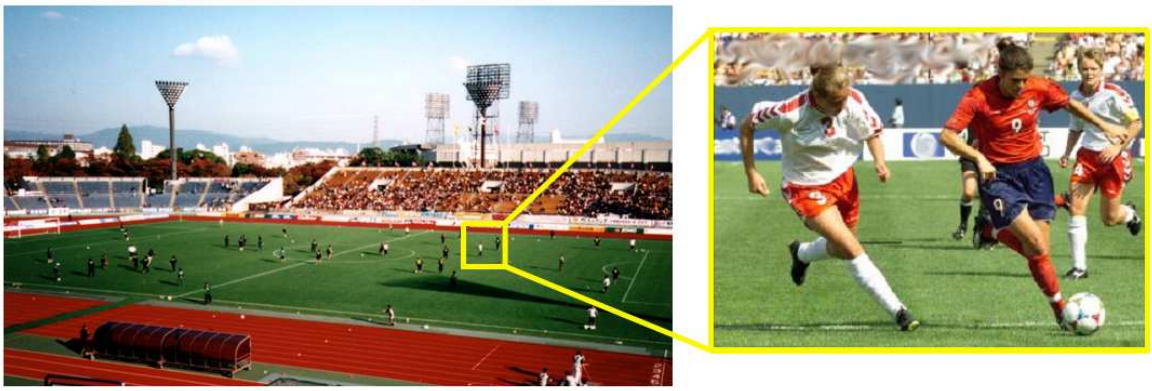


Figure 1 Example mixed scale environment. The motion of players in a soccer stadium occurs in a very large working volume. However the motion itself is extremely detailed relative to the size of the space.

Players run up and down the entire field so that a very large working volume is required in order to observe the players everywhere. However, from the view of the entire field, players appear as little specks, in which it is hard to imagine recovering anything more complex than the players position on the field. Yet, the players are more complex. Each player has some

body pose which changes over time, and at an even more detailed level, the fingers on each hand are in some configuration we might wish to capture. Existing motion recovery systems can not deal with extremely detailed motions such as these that occur in very large spaces.

1.1 Existing technology

Current technology is quite good at capturing specific classes of motion. For example, by placing markers on an actor, and using an array of cameras, the body motion of that actor can be captured in a room size environment. This motion is represented as the three dimensional trace over time of each individual marker. For the purpose of visualization, line segments are often drawn connecting these markers. A computer representation of an actor's actual pose can be seen in Figure 2. Similarly, the facial expression of an actor can be recovered by placing markers on the face of the actor. Again cameras are arranged in order to see the markers, and the three dimensional path of each marker is recovered. The motion of these markers can be used in a variety of applications, including animation. For example, in Figure 3, the position of markers on a face are being used to create a similar expression on an animated character.

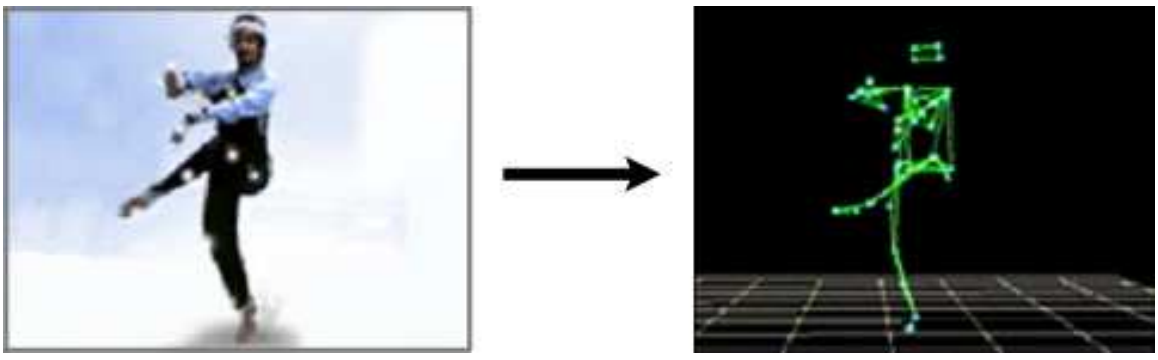


Figure 2 Example of existing body motion capture system. A common use of existing motion capture systems is the recovery of an actor's body motion.

Existing technology does not however adequately address large dynamic ranges of motion. In the examples above, we saw that both relatively large scale body motion and relatively detailed facial expression can be recovered, however simultaneously capturing both large scale and detailed motion is beyond the reach of current systems. A fundamental tradeoff

exists between range and resolution. The cameras used to recover motion in these examples can be fitted with wide angle lenses that allow a large range to be covered, or the camera can be zoomed in to cover a small working volume with much higher accuracy. Capturing a large area at high resolution is prevented by this tradeoff between range and resolution. Although the body and face examples above vary in scale, the ratio of resolution to range within each example remains constant.

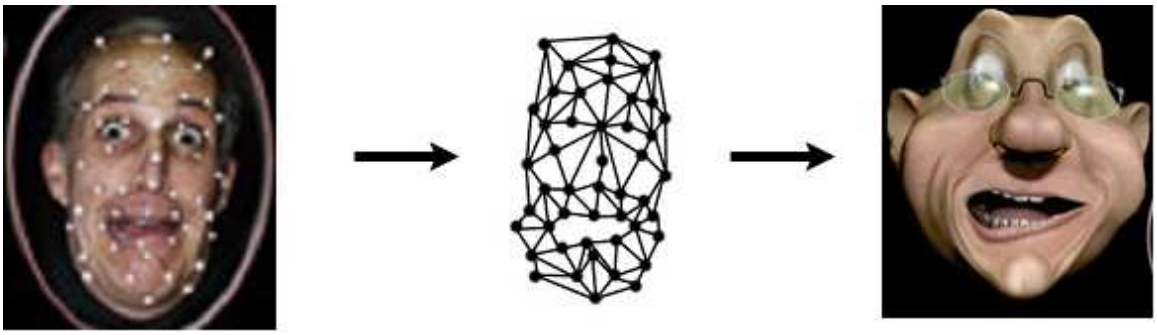


Figure 3 Example of existing facial motion capture system. Facial expressions can be captured by placing a set of markers on an actors face, and observing the motion of these markers with a set of cameras. The motion can later be used to drive the expression of an animated character.

Animation special effects companies typically must capture body and face motion separately. When capturing facial motion, the head is usually rigidly attached to a small box, precisely so that the working volume will be very limited, with a corresponding increase in resolution [51]. An animator is typically hired for the labor intensive task of joining and synchronizing the two separate motions as a post processing operation. Clearly, a method of recovering motion simultaneously in both scales would be advantageous.

1.2 Applications of motion recovery

Motion recovery is required and used in a large number of domains. We have already seen that it can be used by the animation and computer graphics industries. The goal in this case is efficiency. Animators can be hired to create completely original complex motions that will be used in an animation or special effect. However this process requires great skill and is very time consuming. If the motion of live actors is captured and used in conjunction with a computer model, then both the required skill and time of the animator are reduced. Motion

capture makes possible many of the special effects we are accustom to in modern cinematography [49, 50].

Augmented reality makes heavy use of motion recovery. These systems typically work by presenting the user with graphical images overlaid on the real world. In order to generate the correct graphical imagery it is necessary to measure the users current head position [1, 31, 89]. Interaction with the virtual environment also requires motion recovery, since, for example, it is necessary to know when the user has made a grasping motion in order to pick up an object [32, 64, 111].

Athletic analysis and biomechanics benefit from motion recovery as well [3, 5, 24, 90]. By measuring the swing of a professional golfer or tennis player, the motion can be analyzed and compared with both amateurs and other professionals to understand what is special about individual performers. More importantly, by measuring motion before and after injuries and then analyzing the changes, more effective medical treatments can be derived. Even among healthy individuals biomechanics researchers seek to understand the mechanical structures and forces that make up the human body. By measuring motion, a better understanding can be obtained.

1.3 Mixed scale domains

Although motion recovery is already important and widely used for many tasks, there are many domains which lie outside the scope of current technology. This dissertation is concerned with mixed scale motion recovery, the class of domains in which detailed motion occurs concurrently with large scale motion in a wide encompassing volume. We have already seen two examples of this kind of motion, actors for the entertainment industry, and players in a stadium.

In the case of movie special effects and animation it is desirable to capture an actor's body motion and facial expression simultaneously. The motion of the body is on the scale of a room sized environment, while the scale of the much smaller motions involved in facial expression is confined to the size of a human head. Since these two motions occur at radically different scales it is impossible to recover both with existing technology.

The task of recovering players on a soccer field, also lies within this domain. As players run to all corners of the field, their motion or position on the field is at the scale of the field itself. The configuration or pose of each player's body occurs within a space of about two meters. As with the previous example, these two scales are of radically different size, and recovering this sort of motion is beyond the reach of current technology. Nevertheless, this motion would be valuable if it was available, both for analysis of sport team performance, and for special effects production. Just as instant replays are common in today's sporting events, in the future we should expect to see reconstructed graphical representations of players. In this scenario the camera can be placed at any position, so that viewers can see for example, the game from the eyes of their favorite player. Capturing the live performance of players is the primary challenge to constructing such a graphical representation.

There are many other applications that require mixed scale motion recovery, however only one more particular example is given here. Biomechanics researchers study the mechanics, the forces, positions, and connections, of the human body. In order to understand the complex human system, they frequently measure the precise motions involved in a wide range of human behavior. For example, to study the knee joint, they may place markers on a subject, and then ask them to walk, run or jump. Since the goal is to understand the exact forces exerted on the joint, very precise measurements are required. Unfortunately a running subject requires a large working volume, and as with the other examples, existing technology does not permit a solution. Researchers currently work around this limitation by placing the subject on a treadmill in order to limit the required capture volume. However, people run differently on a treadmill than they do on a field and it would be desirable to capture this motion in a more natural environment [42, 77, 112].

1.4 Overview of system

Existing motion recovery systems make a tradeoff between resolution and working volume. In order to overcome this challenge, the design presented in this thesis uses a two part system. A multi-camera large scale recovery subsystem directs a set of pan-tilt cameras that obtain high resolution imagery, as seen in Figure 4. The subsystem responsible for large scale motion recovery can be optimized for a large working volume, while the small scale pan-tilt subsystem is optimized for resolution. Since the two goals have been split into separate

components there is no longer a need to make a tradeoff between these goals. The resulting multi-component system has both higher accuracy and a wider working volume than would be possible if a tradeoff between these goals was required from a single collection of cameras.

The large scale recovery system is responsible for locating targets of interest. In order to cover a large working volume, and to robustly locate targets despite possible occlusions, this system uses many cameras, arranged so that they cover a large space. After locating a target, pan-tilt cameras can be directed to point in the correct direction. Since these cameras are extremely zoomed in, they provide much better resolution than was available from the large scale system.

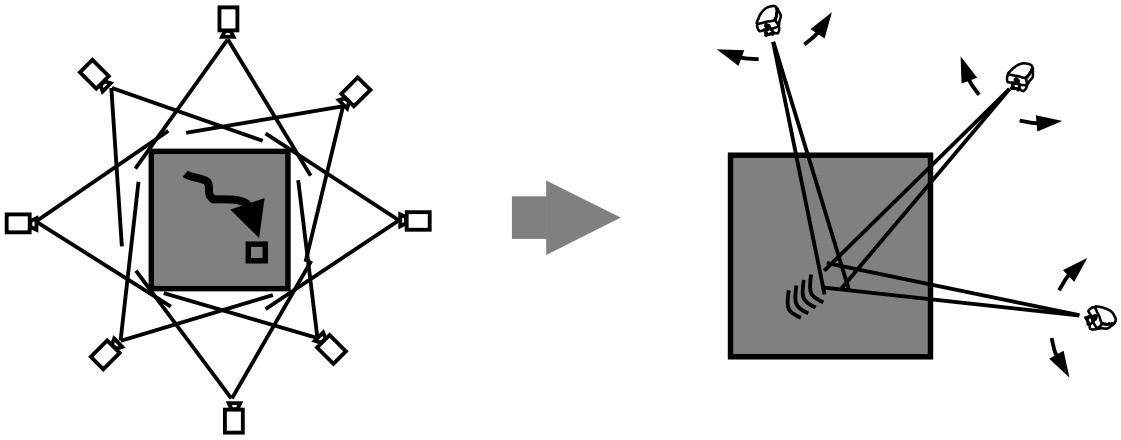


Figure 4 Overview of system design. A multi-camera large scale motion recovery system identifies targets, and directs a set of pan-tilt cameras which obtain high resolution video of the desired target.

While the methods described here are applicable to many tasks, a concrete application is desirable and simultaneous face and body capture will serve as a running example throughout this dissertation. Figure 5 shows one instance from the demonstration system built as part of this research. A person's body motion can be captured anywhere in a room sized environment. In addition, the pan-tilt subsystem recovers markers on the persons face so that facial expression can be captured as well, despite the fact that this motion is at a much smaller scale than the body motion.

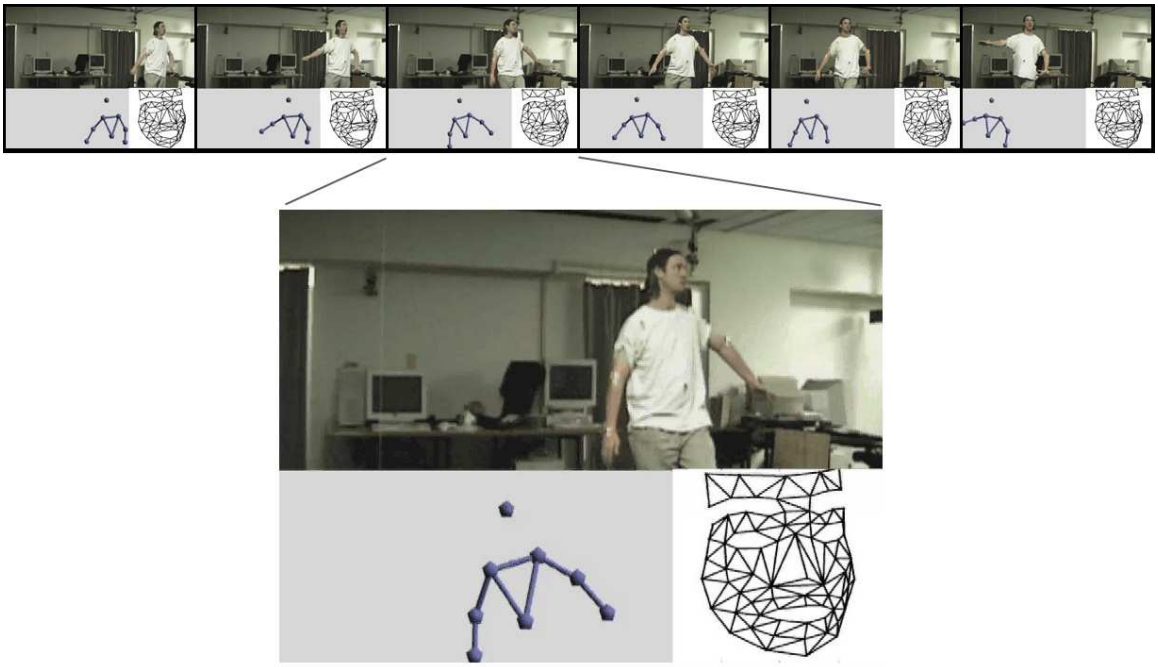


Figure 5 *Video figure* Motivational example from demonstration system. As a person moves through a room sized environment, body and facial motion are simultaneously recovered. (Bottom left) The recovered large scale body motion model. (Bottom right) The recovered facial motion model.

1.5 Desirable system properties

Ideally, a system for capturing motion will robustly obtain results of very high accuracy, in any conceivable space. Of course, not every system does everything well, and a set of criteria for comparing different approaches is desirable.

High range/resolution ratio

A very large range or working volume within which we can recover motion is clearly desirable. Similarly, extremely accurate or high resolution data should be recovered. However these two goals present a fundamental tradeoff when designing a system. For example, consider the image from a single digital camera. We can choose to use a telephoto lens that provides extremely detailed imagery of a small area, or a wide angle lens that provides much lower resolution imagery of a large area. It is possible to obtain either high resolution, or large working volume, but not both. Since the number of pixels in the imager is

fixed, the ratio of range/resolution is fixed. While large range, and high resolution are the underlying goals, it often more useful to evaluate the dynamic range, the ratio of range to resolution, obtainable by a system.

Scalability

Scalability is an important aspect when building a system, and often applies to a number of design parameters. In the context of this thesis, scalability follows the definition of Azuma, a scalable system is one that can be expanded to cover any desired range, simply by adding more modular components to the system [7]. That is, if a working system exists that uses N sensors and captures motion in a X meter^2 area, a scalable system will make it relatively easy to add an additional N sensors and cover an area twice as large, 2X meters^2. In some system designs with many cameras, it is a trivial matter to add more cameras. However in other systems adding additional sensors may be impossible. For example, it may be difficult to add components to a system that relies on actively modifying the environment by projecting light into the scene, since additional projected light may interfere with the original system. In addition, systems with only a few sensors are often not designed with scalability in mind. Scalability requires careful system design, so that data from multiple sensors can be integrated into a single estimate.

Occlusion robustness

Ideally, complex motion can be recovered completely, with no missing segments. For systems designed using cameras, also known as optical motion capture systems, this can be complicated by occlusion. Occlusion occurs when the desired object or action takes place behind some other object, out of view of the camera. If only a single camera is used, then no information will be available and motion data will be missing during this period of time. To combat this problem, many systems use multiple cameras to provide some robustness against occlusion. By using multiple cameras, the probability is increased that at least one of the cameras can observe the target at any given time. By aggregating information from all cameras, continuous motion can be recovered. In practice, suitably complex motions will always produce some occlusion. However a robust system addresses the need to minimize this difficulty.

Runtime automation

Some methods of recovering motion have required a human operator. This operator may point a camera, provide target tracking assistance, or merely identify and initialize new targets. Ideally a fully automated system is desired. While automation during all stages of recovery is desirable, particular attention is required at run time. Since many motions occur very quickly it is impossible to rely on a human operator for guidance while the motion itself is occurring.

1.6 Related work

Capturing motion in both very large and very small areas has been attempted by many researchers. While no existing systems adequately address all of the objectives of this thesis, many technologies have been developed that are useful and related. This section outlines several categories of previous work related to the broad high-level goals of this thesis, and discusses each in the context of these goals. Research related to detailed components of the system will be discussed throughout the text within the subsections to which they are applicable.

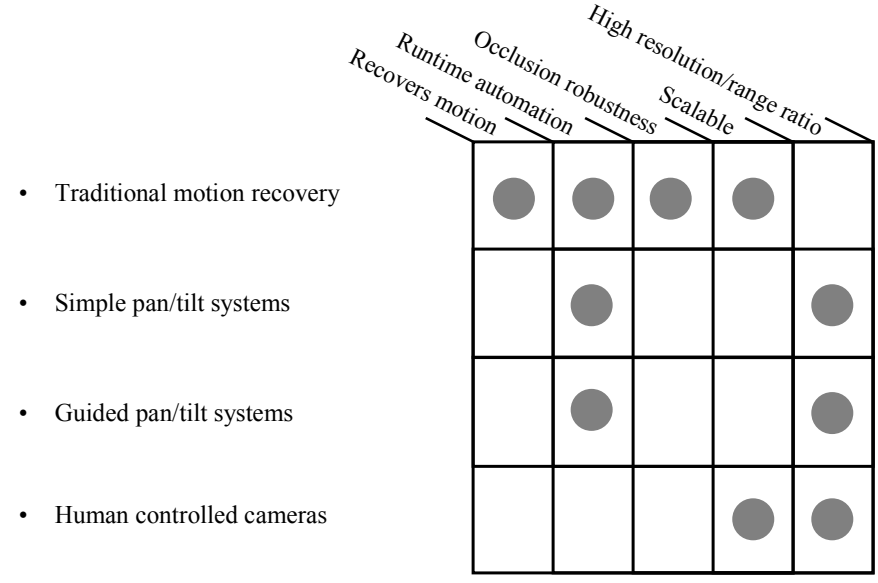


Figure 6 Areas of research related to the broad goals of this thesis. None of these areas addresses all of the desired design goals, although each does introduce technologies from which this work borrows.

Figure 6 shows a table of research areas, together with the design goals that are addressed by each. We can see that none of these categories of previous research address all of our goals, instead each has focused on some subset. The remainder of this section will describe and discuss each of these areas.

Traditional motion recovery

Motion recovery has been broadly investigated in both the research community and commercial domain [4, 6, 14, 17, 20, 22, 23, 24, 33, 35, 37, 46, 47, 48, 51, 58, 59, 63, 75, 79, 83, 84, 85, 88, 92, 106, 108, 114]. A variety of competing systems are available, ranging from optical vision based technologies, to mechanical exoskeletons worn by an actor. These technologies, particularly marker based optical motion capture, are the foundation upon which this work is built. Current commercial systems are very good, but have not adequately addressed the need to capture motion in very large environments. While the commercial interests are aware of this, the current trend is to design cameras of ever higher resolution. While this is certainly desirable, I believe the challenge of achieving truly high range to resolution ratios, such as those needed for the applications I am interested in, will not be possible by pursuing this approach. This thesis proposes a new system that builds upon the current state of the art in existing motion capture systems, primarily in order to address this challenge. Since the work in this thesis is so strongly inspired by existing motion capture systems, the following subsections discuss a number of competing technologies: marker based optical motion capture, markerless optical motion capture, magnetic systems, and exoskeletons.

Marker based optical motion capture

Marker based optical motion capture arranges visible markers of some sort in the environment. A set of cameras observes the motion of these markers, and data from all cameras is used to estimate the 3D motion of each marker over time. This technology was probably developed for military applications in the 1970s. By the early 1980s, several commercial optical systems were available, such as those from Op-Eye and SelSpot [99]. At about this time researchers at the MIT Architecture Machine Group and the New York Institute of Technology Computer Graphics Lab began experimenting with tracking the

human body. Of particular note, in 1983 Ginsberg and Maxwell presented the Graphical Marionette, a system for scripting animations by enacting the desired motions [46, 71]. By 1989 the technology was mature enough for Kleiser and Walczak to produced Dozo, a computer animation of a woman dancing in front of a microphone [99]. Since that time, commercial vendors of optical systems and animation production houses, rather than academic labs, have remained the primary developers of this technology and much of the knowledge surrounding these systems is poorly documented [47]. Today, marker based optical motion capture is the most common commercially used technology.

These systems are attractive for several reasons. They do not require expensive or bulky equipment to be placed on the object or person who's motion we would like to recover. Further, since the markers are usually passive, no wires need be connected to the target. Since known targets are identified, very precise estimates of target location are possible when high resolution cameras are used. Together, these advantages allow relatively free and unencumbered movement and provide high accuracy data, making these systems a preferred method for both the movie industry capturing actors and biomechanics researchers capturing scientific motion data.

Although optical motion capture has a number of advantages, there are a number of difficulties that prevent it from being useful in every situation. The most important is occlusion. In very complex environments, markers may be visible from only a limited number of viewpoints meaning that many cameras are required to adequately ensure that all markers are visible. Processing video from a large number of cameras is expensive computationally, and in fact, optical systems are among the most monetarily expensive commercial systems [27]. Even when large numbers of cameras are used, some cases of occlusion are unavoidable, such as when an actor lies flat on the ground rendering markers on the front side of the body invisible. These difficulties with occlusion are typically handled by capturing carefully scripted motions designed to minimize missing data. One additional difficulty is that these systems often rely on a large number of identical markers. Because of this, determining correspondence between observed markers and physical positions, such as left wrist, can not be completely automated. Consequently, an artist is typically employed to label marker correspondences, 'fill in' any remaining occlusions, and otherwise 'clean-up' the data [44, 47, 99].

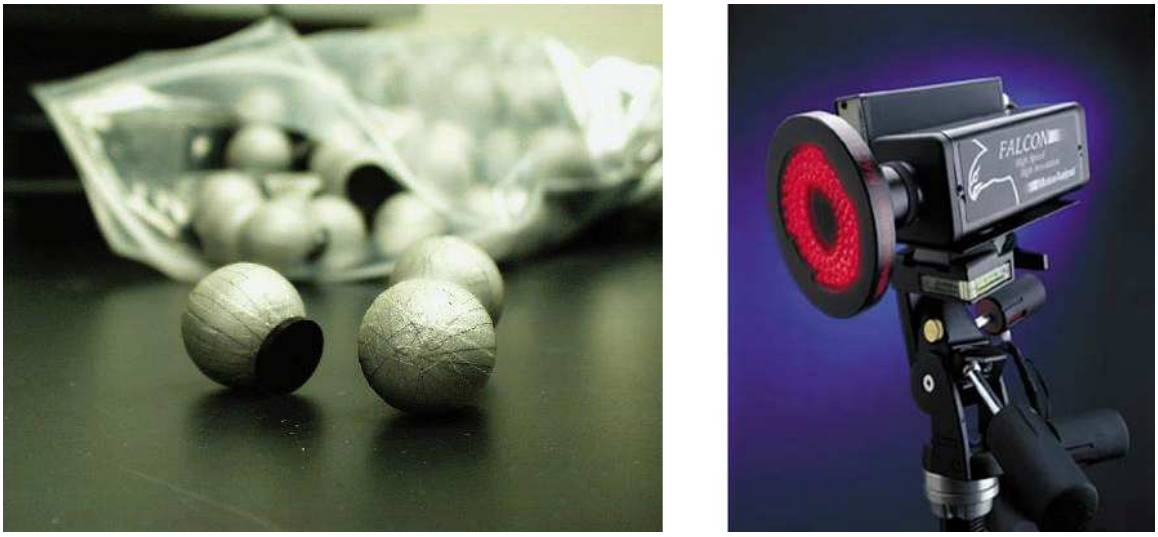


Figure 7 Typical equipment used by marker based motion capture systems. Retroreflective markers and a camera used to track them. The ring of infrared LEDs surrounding the camera lens insure that a bright light source exists which can be reflected directly back into the camera.

Although many smaller companies and custom environments exist, two commercial vendors, Vicon and Motion Analysis, dominate the market for full body motion capture systems [75, 106]. A stage area is prepared by mounting between six and thirty-two high-resolution cameras so that the desired volume is covered. As previously mentioned, using more cameras helps to combat occlusion. Actors are fitted with a set of retro-reflective markers, at various position, such as shoulder, elbow, wrist, etc. Figure 7 shows an example of the retroreflective markers and the customized cameras used to track them. Each camera is connected to a hardware device which extracts the 2D image locations of each visible retro-reflective marker. These markers appear as bright white spots in the image. These 2D locations are later merged to recover the 3D motion of each marker. Typical capture volumes range from 2x2x2 meters, to 5x10x3 meters. Estimates of the resolution of these devices range from the plainly ridiculous 50 microns reported by Motion Analysis sales staff, to 1 cm reported by an animator who makes regular use of this technology.

Recently Peak Performance Technologies and Simi Software began selling solutions to allow similar but lower resolution capture using ordinary video cameras [80, 97].

Guenter et.al. introduce a conceptually similar system for capturing faces [51]. Six cameras are pointed at the face of an actress, who places her head in an immovable box. Approximately 200 brightly colored dots are placed on her face, and the 3D motion of these

dots is recovered while she talks, smiles, or otherwise deforms her face. X-IST sells a face capture system that includes a single camera at the end of a head mounted boom [114]. In this way, the camera moves with the head, allowing a greater range of motion.

Some systems use active markers rather than passive. The advantage of this is that the markers can be identified, solving the correspondence problem mentioned earlier. For example, phoeniXtechnologies sells the Visualeyez system that uses time multiplexing of LED markers, and photo-effect diodes rather than cameras [83]. Since only one marker is active at any time, there is no ambiguity and all markers can be located and identified correctly. Ascension also sells an active marker based systems [6].

Markerless optical motion capture

In some situations it is not feasible to place markers on the object or person that we wish to capture. For instance, during theatrical performances or sports games, any modification of the actors will change the nature of the performance. During surgical medical procedures, augmentation of the patient may be impossible. Systems which attempt to capture motion by passively observing the environment in its natural state, without augmenting it in any way are called markerless motion capture systems. Since these systems also rely on a set of cameras, they share many of the advantages and disadvantages of marker based optical motion capture. They are unobtrusive by design, wireless, and perform well in both large and small environments. In addition, they theoretically have the same high accuracy as marker based systems.

The added flexibility of markerless systems does have a cost. Rather than locating easily identifiable features, these systems rely on either sub-optimal naturally occurring features, or the statistics of whole regions of motion. In either case, robustly locating and tracking these targets is difficult, and indeed the primary factor that keeps markerless motion recovery in the research lab is the lack of robustness. Impressive results have been obtained on some motion sequences, however there is almost invariable a great deal of parameter tweaking involved, making these solutions still impractical for the commercial market. Specific problems often relate to occlusion, regions of flat shading, or incorrectly identified markers. Because these systems usually make use of some higher level model to constrain the recovered motion, incorrectly estimated motion in one part of the model can cause the entire recovered motion to be corrupted, rather than the somewhat more graceful degradation of

marker based systems, in which occluded markers merely disappear, leaving a small hole in the recovered data.

Although the system described in this thesis uses markers, it is valuable to consider current markerless systems since many researchers see marker based systems as merely a stepping stone along the path to the true goal of markerless motion recovery.

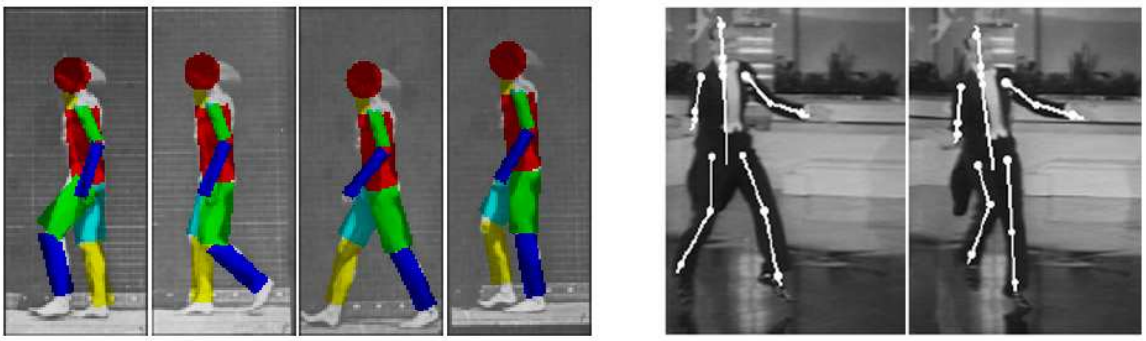


Figure 8 Example of markerless motion capture technology. Systems such as this can recover movement without augmenting the scene. Unfortunately they tend to be specialized and slow, making them unsuitable for use in a real-time application. (Left) from Bregler and Malik (1998). (Right) from Cham and Rehg (1999).

A large amount of computer vision research has focused on tracking image regions with arbitrary content, rather than specific markers. These techniques, e.g. EigenTracking, CONDENSATION, and others are often used as the basis for more elaborate motion recovery methods [14, 61, 68]. The methods layered on top of simple tracking typically build customized models that constrain the range of expected visual input. Models have been developed for nearly every conceivable human motion, for example, researchers have built models to track hands [91, 92, 98], faces [36, 37, 38, 48], and full body motion [19, 20, 23, 35, 59, 63, 79, 84, 95, 96, 107]. Surveys of techniques for facial [78], and full body [45, 73] motion recovery are also available. Figure 8 shows examples of tracking from two such systems.

Magnetic systems

Using a magnetic field established within some volume, position and orientation of a sensor can be determined by measuring the relationship of this sensor to the beacons creating the

field. This type of system has a number of advantages over the optical systems we have previously discussed. Because it is based on a magnetic field, rather than line of sight, it is not occluded by intervening human bodies. This means that motion with no missing data is obtainable. Furthermore, since explicit sensors are used there is no ambiguity, therefore the marker at the wrist, for example, can be robustly distinguished from the marker at the elbow. This robustness makes magnetic systems ideal for applications that require real time estimates of motion. Finally magnetic systems normally come at only half the expense of a comparable optical capture system.

Although magnetic systems have a number of advantages, their disadvantages prevent their use in many applications. Since actual sensors and the associated tangle of wiring must be placed on the body to be tracked, magnetic systems are more cumbersome than optical systems, as seen in Figure 9. In addition, the working volume over which the sensors can detect the magnetic field is limited, and there is no easy tradeoff of working volume and resolution available if larger spaces are desired. These systems usually support a relatively limited number of sensors, with the capture rate decreasing as additional sensors are added. Finally, the magnetic field is distorted by metallic objects, so that the space must be recalibrated each time equipment is moved in the room.

Polhemus and Ascension both offer commercial products using this technology [6, 85]. It should be noted that both companies have recently begun using wireless communication technology to relieve the difficulty of tethered motion.

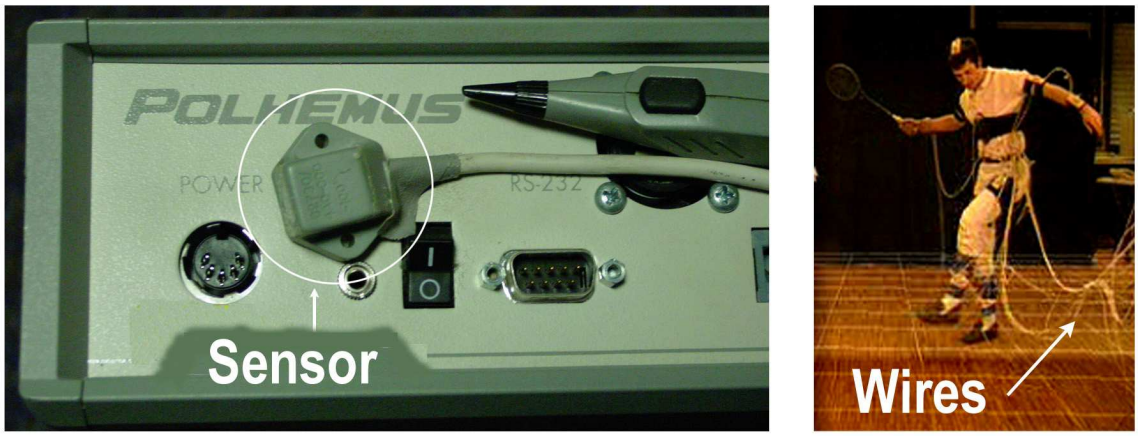


Figure 9 Typical equipment used with magnetic motion capture systems. Left. Sensors used for magnetic motion capture. Right. The tangle of wires often associated with this technology.

Exoskeletons

Exoskeleton based motion capture systems fit a mechanical skeleton onto the body of the actor. These systems were among the earliest motion capture technologies developed. Harrison began experimenting with motion driven animated figures in the early 1960s, and commercialized the technology as SCANIMATE in 1969 [100]. Biomechanics researchers were also early adopters, using the technology for clinical assessment, .e.g. [22].

The joints of the exoskeleton correspond to the joints of the actor, and rotation of each joint can be measured directly. Since these systems are attached to the body and no external equipment is required, the actor can roam freely in as large an area as desired. Furthermore, the angular measures of joint rotation are robust, fast, and accurate. These systems are also the least expensive of the technologies described here.

Unfortunately these mechanical structures are heavy and somewhat limit the motion available to actors. In particular, any sort of interaction or collision between multiple actors is impossible. In addition, only articulated joint motion can be measured. Many applications require the more subtle deformations of non-rigid objects. For example, the human back has a fairly complicated range of motion, and approximating it as a single ridged body is not always acceptable.

The Gypsy system manufactured by Analogus, and the system from X-IST are both commercial example of this technology, shown in Figure 10 [4, 114].

Simple pan-tilt systems

While the methods of traditional motion recovery discussed above attended to the goal of recovering motion, the following categories of related work primarily focus on extending the range of observation.

Cameras can be mounted on pan-tilt motorized drive mechanisms in order to address the need for a high resolution to range ratio. In this scenario, each camera is left in a relatively zoomed in state providing high resolution imagery, while simultaneously using the pan-tilt mechanism to point in a variety of directions, thus providing a large working volume. After a target is identified, the camera attempts to follow the target providing continuous high

resolution imagery. The idea of using pan-tilt cameras to simultaneously obtain high resolution and large working volume is integral to this thesis.

Figure 10 Example exoskeleton based motion capture technology. These systems use a mechanical structure fitted onto the actor's body.

Systems of this sort have been predominately used in applications requiring only capture of video, rather than target motion. For example, the Sony EVI-D30 pan-tilt camera used in my own research has simple target following built into the hardware [39]. The difficulties with this camera become apparent after only a few minutes of use. Targets must be manually located and identified to initialize the tracking. After tracking begins, any sort of occlusion or nearby similar object can cause the target tracking to become confused and either follow the wrong target, or wander aimlessly. Although some provisions exist to reacquire the correct target after occlusion, this is possible only if the target is still within the video frame. Since the camera is zoomed in to provide high resolution video of only a very small subset of the whole working volume, this is unlikely. The result is a systems that works well only for unoccluded targets that move in front of simple backgrounds, such as a lecturer at the front of a classroom, or a person seated at a desk video conferencing system. This thesis builds on the idea of using an actuated and zoomed in camera, to create a much more robust motion recovery system.

Despite their limitations, simple pan-tilt systems of this sort have been used for many practical tasks, for example determining the path of flying insects, and following human faces [13, 39, 43].

Guided pan-tilt systems

In order to address the lack of occlusion robustness in simple pan-tilt systems, a second wide area camera can be added to the configuration. This second camera maintains a view of the entire working volume so that targets of interest can be identified wherever they appear. The pan-tilt camera then provides high resolution images of the target as it moves. This greatly increases robustness, since even if the target temporarily falls out of the pan-tilt video frame, the separate wide area guidance camera can direct the pan-tilt camera to recover. This idea of a separate guidance system that has wider coverage than the high resolution imaging system is an important component of this thesis as well.

As mentioned previously these systems have been primarily concerned with capturing video, rather than capturing motion. Due to this, the wide area camera is nearly always collocated with the pan-tilt camera, creating a fundamentally 2D device. It follows that all of the difficulties with occlusion and similar nearby objects are still present. This thesis expands the idea of a separate wide area guidance system to multiple cameras, and uses the resulting, more robust, system for motion recovery.

Systems that uses a pan-tilt module in conjunction with a wide angle guidance camera have been used in a number of applications. The ALIVE system at MIT uses a guided system to track position and pose of faces [33]. Additional examples include surveillance [18, 81], and video conferencing [93].

Human controlled cameras

The 2001 Super Bowl used a system of 33 pan-tilt cameras that surround a stadium and can provide high resolution video of players on the field from any angle [52, 70]. A human operator directs one of the cameras to point at the region of interest, and the other cameras are directed by an automated means to look at the same position on the field. While systems of this sort use many pan-tilt cameras, they are only superficially related to the work

presented here. The super bowl system is manually operated and currently used only to capture video, with no further processing to recover motion.

1.7 Contributions

The contributions of this dissertation fall into three categories: a framework for thinking about mixed scale motion recovery, a specific system design, and the application of these ideas to simultaneous face and body motion recovery.

Framework for mixed scale motion recovery.

A method for thinking about and understanding any mixed scale motion recovery system is desirable. The particular two stage design presented here is an example of a more general hierarchical paradigm for addressing mixed scale motion recovery tasks. The motivation and details of this framework are given in chapter 2.

Specific system design.

Within a general framework, I present the details of a specific design which allows for mixed scale motion recovery. By using multiple subsystems this design provides for a high resolution to range ratio. It uses many cameras, is scalable to large environments, and robust to occlusion. This system is discussed in chapter 3.

Application to simultaneous face-body recovery.

Mixed scale motion recovery is not of merely theoretical interest. Many applications are beyond the reach of existing technology. The systems described here is applied to the task of simultaneously recovering face and body motion. This is a task of practical importance to the animation industry that currently must perform these captures separately. This application servers as a running example presented throughout the text.

2 Framework

Mixed scale motion recovery is a problem with large scope and many solutions. In order to clarify and organize the space of possible solutions, a framework is needed for thinking about the issues involved. It is of course possible to come to the solutions presented here without the use of this or any framework. However, a framework makes it easier to systematically arrange and categorize ideas, in order to understand the tradeoffs between competing solutions. Importantly, a framework also provides a method of restricting the space of ideas that must be considered so that appropriate solutions can be determined more quickly [65].

The framework presented in this chapter consists of three main ideas. The most important of which is to understand that we are not interested in recovering just any motion, rather that we are interested in *mixed scale* motions. These motions can be characterize in a particular way that leads to a hierarchical paradigm for the solution space. The next portion of the framework makes explicit the duality of system design. Systems are typically designed as a sequence of components. It is also possible and desirable to analyze systems as a sequence of data flows. The last portion of the framework expresses each instance of a system design challenge as a specific inadequacy in the data flow. By interpreting challenges consistently in this manner a common class of solutions, data models, can be investigated.

2.1 Problem domain characterization

The goal of this research can be stated as: the recovery of very high accuracy motion in a very large environment. However this characterization of the problem leads to the belief that high accuracy is required everywhere, all the time. In fact the situation is often quite different. If we follow the obvious line of reasoning we are lead to develop higher resolution cameras in order to increase the total available resolution. While this is useful, other solutions become apparent when we characterize the problem domain differently.

Motion often occurs at *mixed scales*, that is, there are multiple distinct scales in which the motion occurs. By isolating these scales and treating them separately, more efficient solutions are possible. Consider the previously described example of players in a stadium. At any given moment, most of the stadium is empty, with no motion available to recover. By concentrating resources where the action is, far fewer resources are necessary.

Rather than viewing the whole stadium as a single unit, let us instead consider the three scales of motion illustrated in Figure 11. Each player's motion can be considered at each of these scales. The largest scale is the player's motion as they run up and down the field, their current position. At this scale, players appear as little specks, and extreme resolution would be required to recover any sort of detail regarding the players body configuration. At a smaller scale is the player's body pose, the exact configuration of joints and limbs as the player moves. While this configuration could occur at any position on the field, it is clear that the relevant working volume needed to recover this pose at any given time is much smaller than the space of the entire field. For example, the middle image in this figure, shows just the relevant part of the field, with enough detail to recover body pose. At an even smaller scale is the players facial expression. At this scale, the motion is confined to a very small working volume around the players head, as shown in the third panel.

Figure 11 Motion of players in a stadium occurs at multiple scales. The position on the field, the players body pose, and the players facial expression are three possible scales we might wish to consider.

This example allows us to give a new characterization to our problem domain. Motion occurs on multiple scales. At each scale, the working volume is local. That is, the motion at a given scale is confined to a small region, rather than spread uniformly throughout the entire space. The working volume at smaller scales is moving. This shifting is directly related to the

current larger scale motion in the space. In terms of this example, the players body pose or facial expression is confined to a local region. However, this region moves as determined by the players position on the field, or current body pose. A wide variety of motion can be characterized in this way, as mixed scale.

2.2 Hierarchical paradigm

Mixed scale motion can be characterized as a set of motions, each element of which is at a different scale. This characterization can be incorporated in a hierarchical paradigm of system design by explicitly expressing the motion hierarchy within the design. The soccer example previously discussed is used again in Figure 12 to show a generic system description that expresses this hierarchical paradigm.

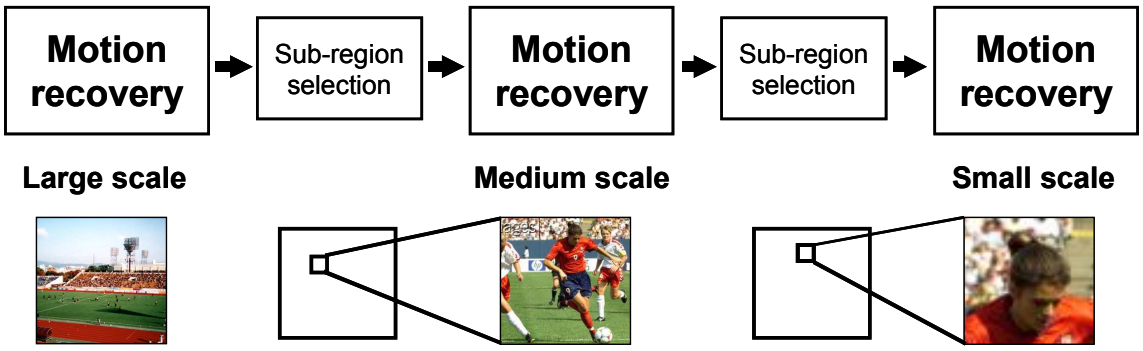


Figure 12 A hierarchical paradigm can be used to express mixed scale motion recovery. At each scale, recovered motion drives sub-region selection. The selected sub-region in turn defines the next scale.

At the largest scale, some sub-system exists to recover the motion of the player on the field. This recovered motion, the current position of the player, is used to select a sub-region from the complete working volume. Since the player occupies a much smaller working volume than the entire field at any given moment, sub-region selection can be used to define just this smaller area as the working volume for the next scale. After a sub-region has been selected, motion can be recovered at the next scale. In this case, the next scale of recovered motion is the body pose of the player within just this smaller space. This process can be repeated by

selecting an even smaller sub-region, which will define the next scale at which motion should be recovered. The important aspect of this system framework, is that at each scale, the entire working volume need not be considered, instead only a sub-region volume is used. The resolution of the motion recovered within this sub-region is proportional to the new working volume.

The specific system described in this dissertation is one example of this hierarchical paradigm. As shown in Figure 13, a multi-camera large scale recovery system determines the location of a target and recovers its large scale motion. A sub-region selection mechanism directs a set of pan-tilt cameras to point at the sub-volume surrounding the target location. Since the pan-tilt cameras converge on a small sub-region containing the target, small scale motion can be recovered within this volume. This implementation is comprised of a two part hierarchy, where the goals of the large and small scale motion recovery systems are similar except for their scale. Sub-region selection primarily involves the task of directing the pan-tilt mechanisms to accurately and robustly point at the target.

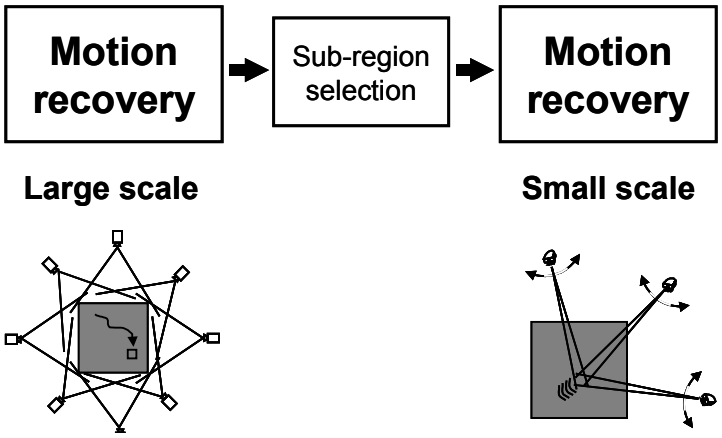


Figure 13 The specific system described in this dissertation is one example of using a hierarchical paradigm. This system has two stages of hierarchy. Large scale motion recovery is used by the sub-region selection mechanism to direct pan-tilt cameras towards the target. The video from these pan-tilt cameras becomes the basis for small scale motion recovery.

A hierarchical organization of motion recovery represents a potentially enormous efficiency gain. Rather than allocating resources evenly across an entire working volume, resource are instead allocated more heavily in the relatively small volumes in which the interesting motion is currently occurring. By concentrating resources in this way the system better matches the

characteristics of mixed scale motion, and achieves a resolution that is greater than would be possible if all cameras had a single scale.

The system of pan-tilt cameras described in this dissertation is not the only design which follows from a hierarchical paradigm. Suppose for a moment that we chose to cover a hypothetical stadium with thousands of statically arranged cameras, each zoomed in to cover only a very small fraction of the field. With the decreasing cost CMOS imagers, this is becoming a feasible possibility. Initially, such a system seems to be at a single scale since equal resolution is provided everywhere, however when we more carefully consider how to make such a system efficient in terms of computational resources we are led back to a hierarchical paradigm. Processing the video from all of these thousands of cameras would be expensive, so instead we might use an overview of the field to select a subset of the cameras in which the scene is currently active. Processing the video from just this subset would result in improved efficiency. We can see that even in this rather different scenario, efficiency concerns lead to the same hierarchical paradigm that guided the general design of our pan-tilt recovery system. Rough knowledge of the entire scene is used to select a subset or sub-volume to focus on. This sub-volume is then processed in more detail in order to recover finer scale motion.

Another possible system design would be to use a few super-high-res cameras that cover the entire field. At this time, cameras of the needed resolution are not available, nevertheless let us consider the design. As in the previous scenario, processing all of the data would be inefficient. Instead, we might choose to process a lower resolution version of each image in order to locate sub-regions of interest. These sub-regions could then be processed at a more detailed resolution. In fact, this sort of hierarchical or mixed scale processing of a single image can be found in the existing image processing literature [103, 115].

Motions at multiple scales are a fundamental characteristic of mixed scale motion recovery. By treating these scales within a hierarchical paradigm we can design recovery algorithms that make efficient use of available resources. The system design presented in this dissertation has sub-systems which explicitly mimic the scales of motion inherent in the problem domain.

2.3 Interface requirements

The interface between components or sub-systems in a large system design often presents the main challenges. The specific mixed-scale motion recovery system presented in this dissertation is shown in Figure 14. In this case, the sub-systems for large scale motion recovery, sub-region selection, and small scale motion recovery have been expanded to reveal slightly more detailed components.

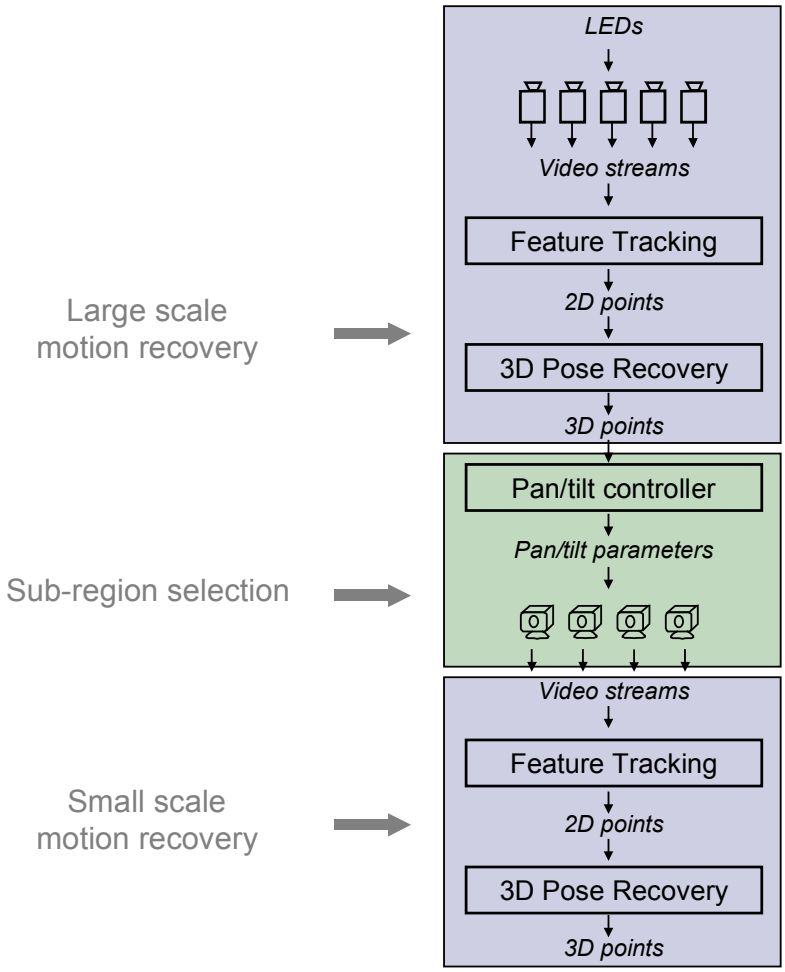


Figure 14 Expanded system overview. Cameras observe LED point features in the environment. These point features are detected and merged to recover the large scale pose of the target. This large scale pose is used to guide pan-tilt cameras to point at the target. These cameras in turn observe features which are detected and merged to recover the small scale pose of the target.

Looking at the components of the entire system from top to bottom we can now understand its structure in a little more detail. LED targets in the environment are observed by a system of cameras. Machines hooked to these cameras track the features in the environment. These features are then merged into a single estimate of the current state of the environment by a pose recovery module. The current pose of the target allows a pan-tilt controller to guide each of the pan-tilt cameras to point towards the target region. The video from each of these cameras is digitized, and features are tracked. These features are then merged to recover the small scale motion of the target.

From the preceding description of the entire system, the similarity of the large scale and small scale subsystems should be apparent. In both cases, targets are observed in the environment, located and then merged to recover the three dimensional pose of the observed target. Yet, as will be described, these subsystems are not identical. An important difference that governs the solutions available are the interface requirements between a sub-system and the rest of the design. For example, the large scale motion recovery system must run in real time, because the location of the recovered target is needed to guide the pan-tilt cameras. The small scale motion recovery system has no such requirement, since in this pipeline there is no following stage which is dependant on timely input. On the other hand, the small scale motion recovery sub-system receives video from cameras that are in motion. The dynamic nature of this video may place restrictions on the set of algorithms available for feature extraction and pose recovery.

In building end-to-end systems such as the one described in this dissertation, the interfaces are critical. Each system component could be chosen from many existing research results. Certainly many feature detectors and pose recovery subsystems exist. However these components are often developed separately, outside of any larger system design. Consequently their designers have often paid little attention to the boundaries, the interfaces where each algorithm connects to the next. These interfaces are precisely where the challenge lies in building an end-to-end system. Since the components do not match exactly, the pieces must be carefully chosen and assembled to build a new system.

2.4 Data flow analysis

Since interfaces can be seen as the major challenge in building any large system, it is instructive to view the entire system in terms of these interfaces, rather than as a sequence of components. Figure 15 shows the same system, but with emphasis on the interfaces, the data that flows between components, rather than on the components themselves. In this view a stream of video images is transformed into a stream of two dimensional feature point locations. This data is merged to form a stream of three dimensional feature points. These

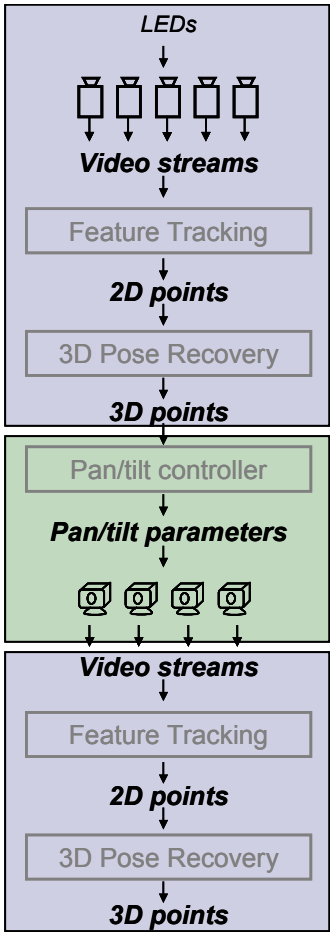


Figure 15 System overview when viewed as a sequence of data flow. Video streams are transformed into streams of image feature points. These data points are converted into a stream of target poses. The pose data is used to derive a sequence of camera parameters, which in turn generate a new set of video sequences. The video from each pan-tilt camera is transformed into a new sequence of feature points, and then into small scale object pose.

features are used to derive a stream of camera control parameters, which in turn produces more video. This new video from the pan-tilt cameras is transformed into image feature points. Finally these small scale features are used to derive the target's small scale pose over time. Each of these data types acts as the interface between components.

I have suggested that the challenge in building an end-to-end system is in ensuring that the interfaces between components match, that each component is getting the data it needs. This can be rephrased as ensuring that the data flow is clean. Noisy or missing data may be unsuitable for the next component. A key task of system design is to ensure that the data flowing between stages is suitable. This should be viewed as both a challenge and an opportunity. While it is important that the data flow be of adequate quality to drive the next stage, it does not need to be any better than that. This implies that perfect data is neither necessary nor desirable. This fact can serve to reduce processing requirements in some cases.

Most of the seemingly unrelated challenges to building a system can be seen as instances of poor data. In Figure 16 a number of challenges have been labeled near the stage of data flow they inflict. Each of these is an example of inadequacy, corruption, or noise in the data at that stage.

In order to better understand how these challenges are instances of inappropriate data, let us consider some examples. The video streams coming from cameras represent the current state of the scene. The feature tracking component of our system seeks to determine the image plane location of each target in the scene. However this is impossible for some target locations, because this data is missing from the video stream due to occlusion. This occlusion is an example of imperfect data. Ideally every target would be visible in each video stream.

At the next stage of the data flow, we again run into difficulties. Suppose that every target is in fact visible and that the image plane position of each target has been extracted. These image plane positions will be merged in order to derive the three dimensional scene coordinates of each target by a pose estimation component. This merging component expects accurate estimates of the actual image plane location of each target. In practice there will be some noise in these estimates, due to imager resolution, changing target appearance, or other reasons. This noise is an imperfection in the data that makes it difficult to accurately reconstruct the targets pose. If the noise is too high, meaningful reconstruction will not be possible.

Consider an example later in the pipeline, an incorrect camera model can cause incorrect pan-tilt parameters to be derived. The imperfect data stream of pan-tilt parameters makes it impossible for the cameras themselves to actually point at the correct target.

Similarly, each system challenge can be seen as an instance of inadequate data flowing in the interface between two components. The difficulties with the data may be inherent in the problem domain, or they may be due to a particular component choice. However, by viewing each challenge as an instance of a single class, rather than as individual and separate challenges to be faced, a framework is available for addressing the overall complexity of system design.

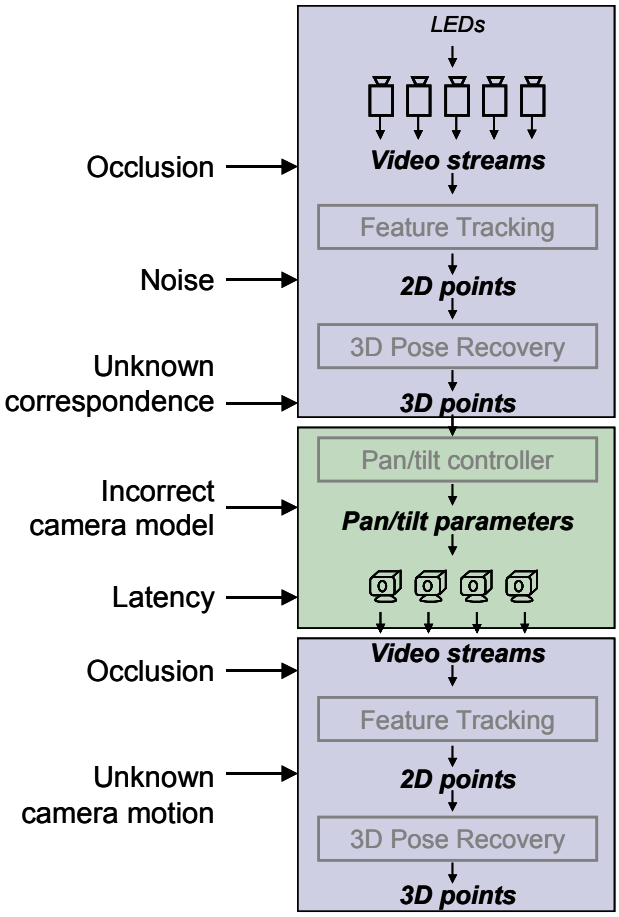


Figure 16 System challenges as instances of poor quality or noisy data. Each challenge is a place where the data needs to be cleaned up before it is suitable as an interface to the next component in the system pipeline.

2.5 Model based solutions

One method of improving noisy or poor data is to use a model to constrain the range of allowable values. Since the challenges in building a complete system can be seen as inadequacies in the data flowing between components, these challenges can be addressed by building models to constrain the data at each interface. These models may construct a subspace and restrict the data range explicitly, filter or regularize the data, or use some algorithm to deal with missing information, but always the goal is to make the data flow suitable for the next component in the system pipeline.

Figure 17 shows a system diagram together with a few specific models used to address challenges that arise in the dataflow. In some cases a single model is used to address issues at multiple stages in the pipeline. For example, a Kalman filter is a model that regularizes noisy measurement data. In this case it is also used to provide a temporal model of motion so that correspondence of data across time can be established.

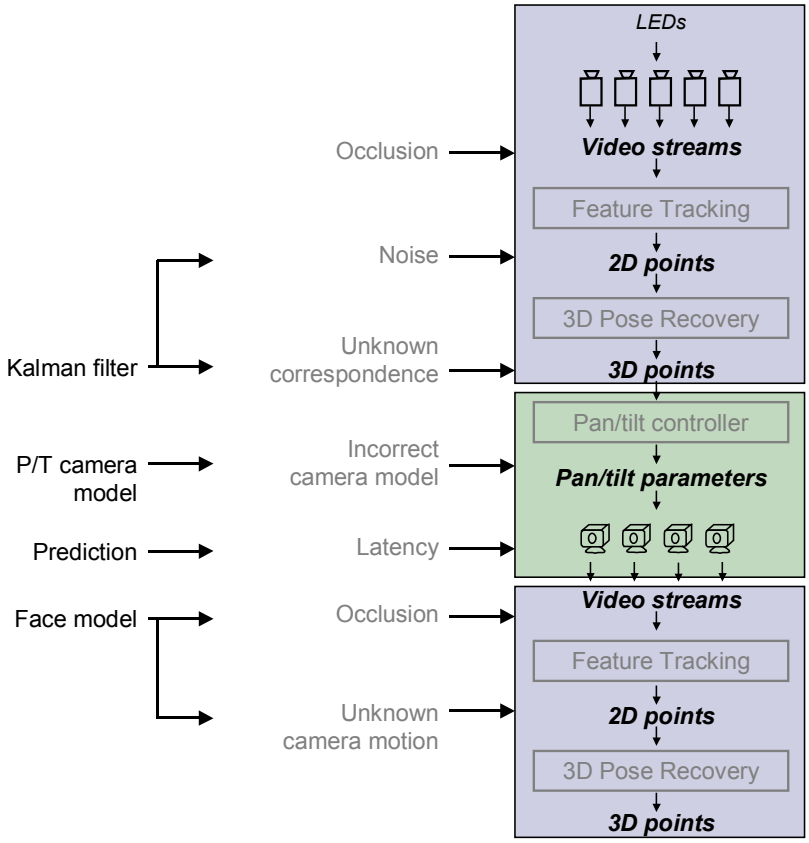


Figure 17 Models can be used to address challenges that arise due to inadequacies in the data flowing between interfaces in the system pipeline. These models constrain, filter or regularize the data so that it is suitable for the next processing component.

Other examples of models used in this system design include a new model of pan-tilt camera motion. The new model improves the accuracy of predicted image plane observations over the standard simplistic model. Also a prediction model is used to counteract latency in the data stream used to control the orientation of pan-tilt cameras. Finally a face model is used to constrain the space of possible face shapes, effectively resolving difficulties with occlusion and unknown camera motion.

These and other models used in the system design comprise an important element of the actual system. They are described in detail in the following sections relating to each system component and interface.

In this chapter a framework for thinking about mixed scale motion recovery systems has been presented. Most importantly, characterizing motion as mixed scale leads to an explicit *hierarchical paradigm* of system design. In addition, it is useful to interpret a system as a sequence of *interfaces and data flow*. Finally, system challenges can be addressed with *model based solutions*.

3 Large scale recovery

Mixed scale motion can be recovered using a system that explicitly mirrors the hierarchy inherent in the motion itself. The system presented in this work uses a large scale motion recovery subsystem to recover motion that occurs at the scale of a laboratory room. This system is used both to capture the body motion of an actor, and to localize a region of interest for more detailed tracking. This chapter details the design of this system component.

The large scale motion recovery subsystem is responsible for localizing targets so that sub-region selection can occur. This goal requires robustness, since if the target is lost, the remainder of the recovery pipeline, sub-region selection and small scale recovery, will be impossible. In addition, the large scale subsystem must be scalable since we would like to recover targets in very large environments. In order to achieve these goals, this subsystem makes use of many cameras. These cameras can be added as necessary to increase the range of possible recovery. Robustness is primarily related to occlusion, and using many cameras also serves to increase robustness since the number of viewpoints from which the scene is observed increases.

The large scale system uses many statically mounted cameras with wide angle lenses to observe the motion of target features. For example, the system in our lab use cameras mounted on the ceiling as in Figure 18. These cameras cover a wide area from many viewpoints, observing a set of features as they move through the volume. While many different features could be detected, the system in our laboratory uses easily detectable point features such as the LED in Figure 19. Using image processing techniques, the location of features is extracted from each video stream. Each observed point feature represents a ray in space. By triangulating these rays, 3D point locations can be obtained as shown in Figure 20. These 3D point locations are exactly the information needed to recover both body motion and the parameters that guide pan-tilt cameras to look at specific targets.

A number of difficulties complicate the process of recovering large scale motion: features are often difficult to detect in the presence of background clutter, wide area configurations of cameras can not be easily calibrated, the required correspondence used to integrate camera observations is unknown a priori, new targets requiring initialization may appear while the system is running, and communication between distributed machines involves unpredictable latencies. The remainder of this section is primarily devoted to discussion of the specific architectures, methods, and solutions employed by this system, in order to address these difficulties.

Figure 18 Cameras mounted on the ceiling of a laboratory are used to capture large scale motion.

Figure 19 An LED light source is a useful feature to track, since detecting it against a wide variety of backgrounds is easy and robust.

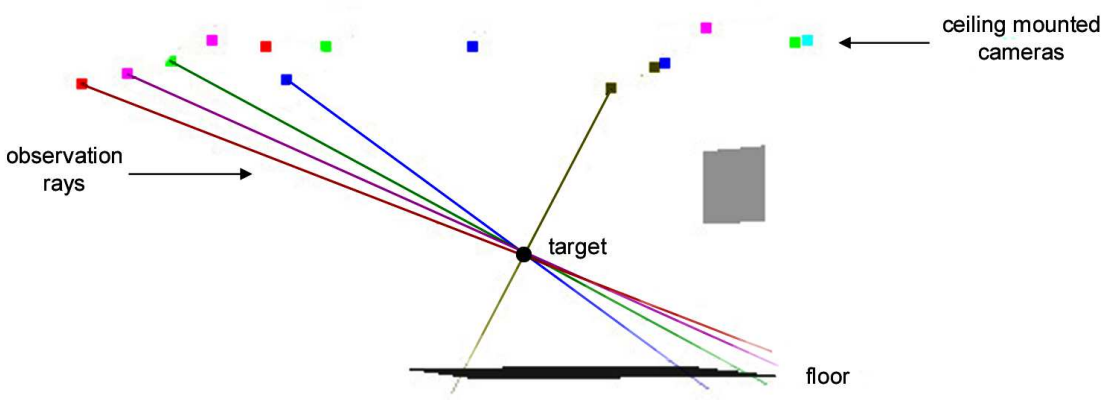


Figure 20 Diagram of cameras observing a single target. This is a view from the side, ceiling mounted cameras appear at the top of the figure. Each camera generates a ray in space. These rays can be triangulated to determine the target's 3D location.

3.1 Physical setup

Large scale recovery makes use of many cameras which observe the scene from a wide variety of viewpoints. The system in our lab uses eighteen NTSC cameras mounted near the ceiling. In Figure 18 we saw a photograph of some of these cameras. Figure 21 contains a diagram of the camera arrangement, shown from above. Notice that cameras are arranged to cover a wide working volume from a variety of viewpoints. The coverage area in which objects can reliably be tracked is approximately 3.5m × 5m, and extends from roughly 0.5m to 2m in height. Cameras can be added in a scalable fashion to this arrangement if a larger or different working volume is needed.

Each camera is connected to a video digitizing board. The current arrangement uses both SGI Indy workstations and Wintel PCs with Matrox Meteor II capture cards. In order to provide flexibility with respect to which cameras are connected to which digitizing boards, all equipment is wired through a patch panel that allows easy reconfiguration, shown in Figure 22. The video stream from each camera is digitized by an individual host machine. This machine is also responsible for locating features within video frames. The recovered features from each camera are sent over ethernet to a SGI Onyx which aggregates the information to recover 3D pose.

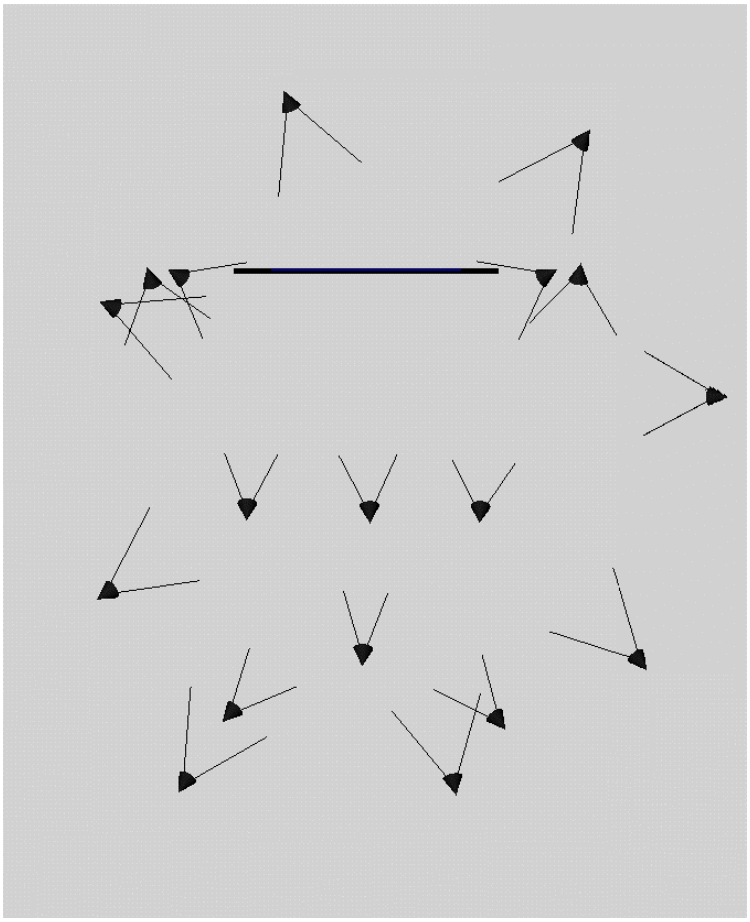


Figure 21 The arrangement of cameras used for large scale motion recovery, shown from above.

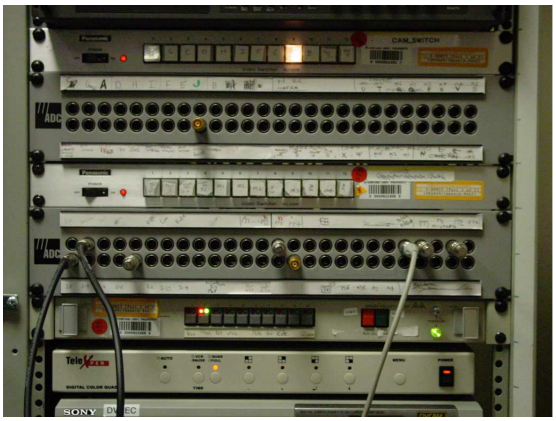


Figure 22 (Left) Cameras are connected to machines through a patch panel in order to provide configuration flexibility. (Right) The stacks of Indy's shown here, as well as Wintel PCs, are used to digitize video and locate features.

The arrangement of cameras need not be identical to the arrangement shown here. In fact the optimal arrangement will vary depending on the coverage area and task. For example, the arrangement shown is actually a composition of two sets of cameras which were originally placed with different goals in mind. Figure 23 shows these two arrangements. One was designed to allow head tracking for people moving in front of a large wall size display. In this case, the cameras were arranged to provide better resolution near the display than further away. In the other case, cameras were arranged in a roughly symmetric circle, such that all cameras had a view of a single working volume. This arrangement was used for experimentally discovering statistical models of occlusion as people move through space. The best arrangement of cameras is determined by the desired task. A metric for determining the best placement is discussed by Chen [25].

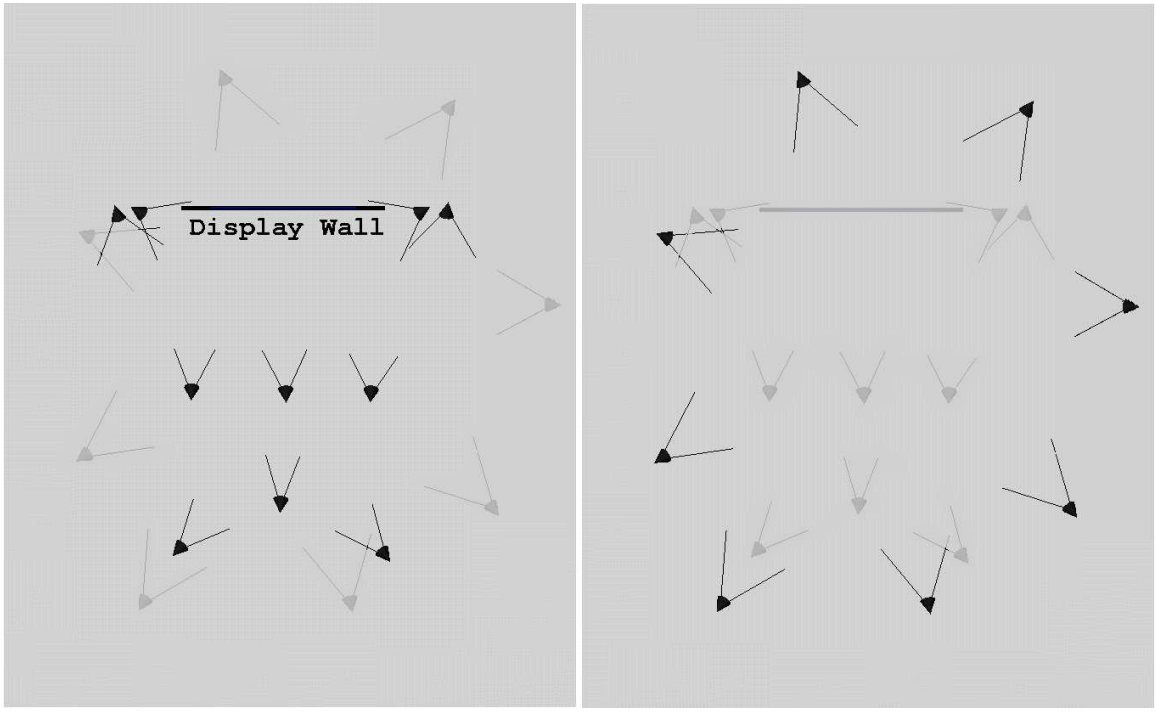


Figure 23 Two independent arrangements of cameras were combined to produce the current configuration. On the left is an arrangement designed for head tracking in front of a display wall. On the right an arrangement used for experiments on occlusion probability.

3.2 Calibration

Reconstruction of target positions requires knowledge of the geometric relationship between cameras. This is typically referred to as calibration information. While many methods exist for calibrating cameras, most existing research has focused on a single imager, .e.g. [57, 104, 105, 116]. Existing techniques can of course be applied to multiple cameras that observe the same working volume, and in fact commercial motion capture systems typically follow this model, requiring placement of a large calibration target in a place that can be seen by all cameras. However, existing techniques do not scale well as the range of multi-camera arrangements increases. This is primarily because building adequately large calibration targets becomes unmanageable as the size of the working volume grows. Since the motion recovery system described here uses many cameras, not all of which have overlapping working volumes, a new method of camera calibration was developed.

From the viewpoint of the framework presented in Chapter 2, camera calibration can be understood as a model that explains the relationship between 2D feature observations and 3D target locations. This model is estimated prior to its application in the data flow pipeline of this system. While existing models are adequate, existing methods for determining model parameters are not. As will be described, the method developed for this project uses a technique that essentially estimates model parameters directly from the data flow itself, rather than attempting to obtain them from some separate process. Complete details of this method are available elsewhere [25, 26], however a brief summary follows.

This method replaces a physical calibration object with a virtual calibration object that is used to calibrate the cameras. This virtual object is created by taking a single detectable target and moving it around the working volume. Suppose for the moment that the exact path of this target through the space is known. The geometric relationship between target locations and image plane observations can be determined. These relationships are conceptually identical to the relationship of physical target feature locations and image plane observations. Thus calibration can be determined via standard methods.

Unfortunately the exact path of the calibration feature through the working volume is not known. However, suppose for a moment that an existing set of cameras is already calibrated. In this case, the cameras themselves can be used to track the motion of the target feature. The known cameras can be used to triangulate the position of the feature in space. Figure 24

shows an example of one such path in space. Note that the path covers a substantial portion of the working volume.

Camera calibration can be obtained if the location of the feature path is known. Similarly the feature path can be found if the cameras are known. With even a poor initial guess, iterating between the two formulations will result in a solution to both sets of parameters. That is, given a ‘poor’ estimate of camera calibration, solve for a ‘poor’ estimate of the feature path. Given this estimate of feature path, solve for a ‘better’ estimate of camera locations, and so on. Figure 25 shows a plot of image plane projection error on an independent data set. It is clear that this method converges to a good estimate of camera calibration.

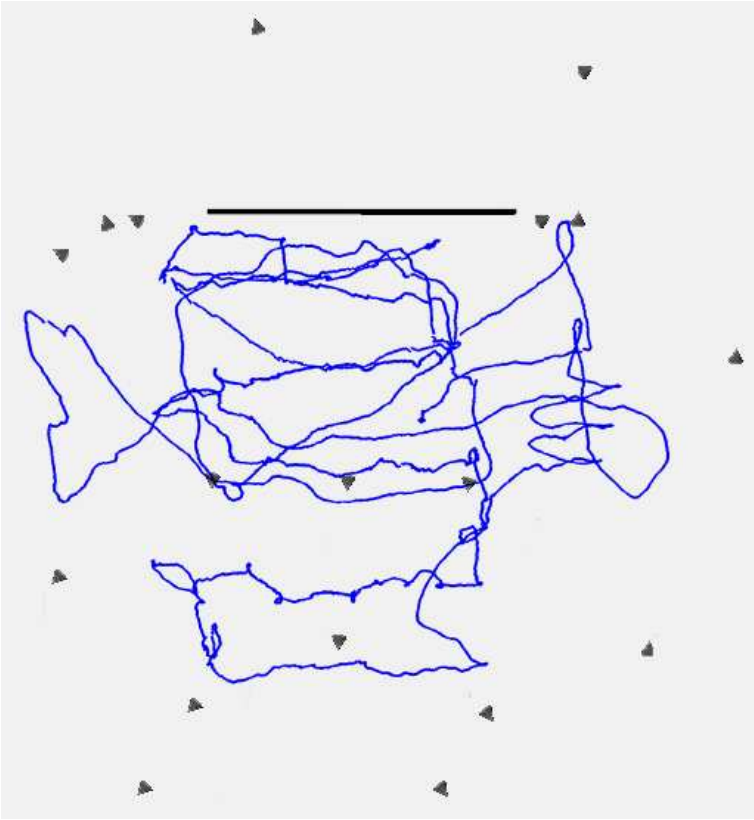


Figure 24 Path of a feature used to create a virtual calibration object. Note that this path covers the entire working volume in a way that would be difficult using a physical calibration object.

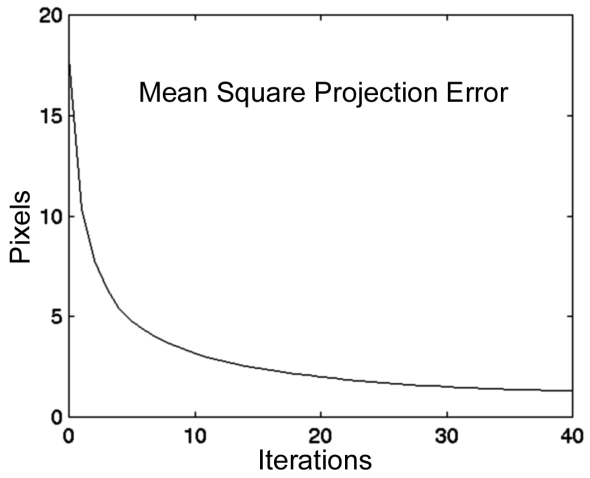


Figure 25 Convergence of proposed calibration method. Iterations vs. projection error. It is clear that this iterative method of camera calibration converges to a good estimate of camera calibration.

3.3 Feature detection

Each camera in the large scale recovery system produces a video stream from which target feature locations must be determined. Ideally features should have some visually distinguishing characteristics so that they can be tracked robustly over a variety of backgrounds. Researchers have proposed a wide variety of methods for tracking naturally occurring features, however these systems are generally not suitably robust [14, 40, 41, 68, 104]. All existing commercial motion capture systems make use of specially designed markers in order to increase robustness. In the work presented here, LED light sources are used as features because they are relatively easy to detect in a video sequence, leading to robust tracking results.

In order to measure the motion of a person or object, targets are attached at key locations so that the combined motion of all target features adequately represents the more complex underlying motion of the person. In Figure 26 a person is shown, along with the locations at which eight features were attached, in this case on the shoulders, elbows, wrists, head and waist of the subject. These targets are observed by many cameras. Since the viewpoint of each camera differs, there will be variation in the set of detected features. For example, Figure 27 shows the feature sets detected by six different cameras for one possible subject pose.

Despite careful design, robustly detecting features in a natural environment remains a great practical difficulty. Feature locations are not determined exactly due to image noise. Scene clutter produces false feature observations, and real targets produce multiple observations. All of these are examples of noise in the 2D feature data. The remainder of this section describes the techniques used by this system to locate features while minimizing error.

Figure 26 Example target locations. Features are attached to a person at positions that will allow recovery of the desired motion.

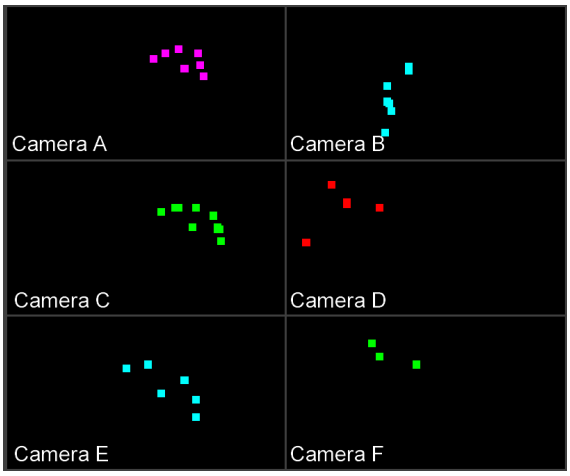


Figure 27 Feature observations from several views. Each camera reports the point features it observes. Since the camera viewpoints vary, the number of observed points and their locations also vary.

It should be noted that after applying these techniques considerable noise and false observation information remains in the 2D data flow. The downstream components in the data processing pipeline need to account for the poor quality information they receive. As will be discussed later, models are applied later in the pipeline that address many of the inadequacies. For example, a Kalman filter is used to model motion during 3D reconstruction. This filter has several advantages, one of which is reducing positional noise introduced by observations with poor accuracy.

It would of course be possible to apply models directly to the low level feature detection algorithms in order to improve their robustness. However future research may evolve this system such that it uses naturally occurring features, rather than engineered markers. It would be desirable to replace the feature detection module of the system while minimally affecting other portions of the pipeline. Most general purpose feature detection algorithms produce data with a great deal of noise. Thus is seems sensible to apply noise regularizing models later in the pipeline rather than directly during feature detection.

Theresholding

Light sources are fairly easy features to detect since they typically appear much brighter than the surrounding scene. The system presented here uses simple thresholding to locate targets [56]. All image intensity values greater than T are labeled as features, and all values less than T are labeled as background. Figure 28 shows a sample image, along with the pixels that have been labeled as features. As can be seen each light may project onto more than one pixel in the image plane. The exact size of this spot varies with LED brightness, LED directionality, distance from camera, and camera iris settings. All pixels illuminated by a single feature are aggregated using a connected component search. The centroid of each group is reported as the feature location.

Ideally the threshold used to identify features could be set to virtually any value. The lights should be bright, and the rest of the scene should be relatively dark. By adjusting the camera iris, the overall image gain can be set so that the background is nearly black. Initial versions of this system did in fact merely set the threshold value to half the maximum image intensity, with adequate results. Although this simplistic thresholding works, a number of factors reduce the quality of feature detection. As features move further from a camera, they get

dimmer. After an LED feature passes a certain distance its brightness falls below the threshold and the feature is no longer detected. Clearly, a threshold that is as low as possible will be advantageous. However, as lighting changes in the work area, the background illumination level also changes. Every time background lighting changes, the ideal threshold level changes. In addition, since lighting in our laboratory is uneven the ideal threshold value varies from camera to camera. Even worse, the ideal threshold can vary between regions within a single camera view.

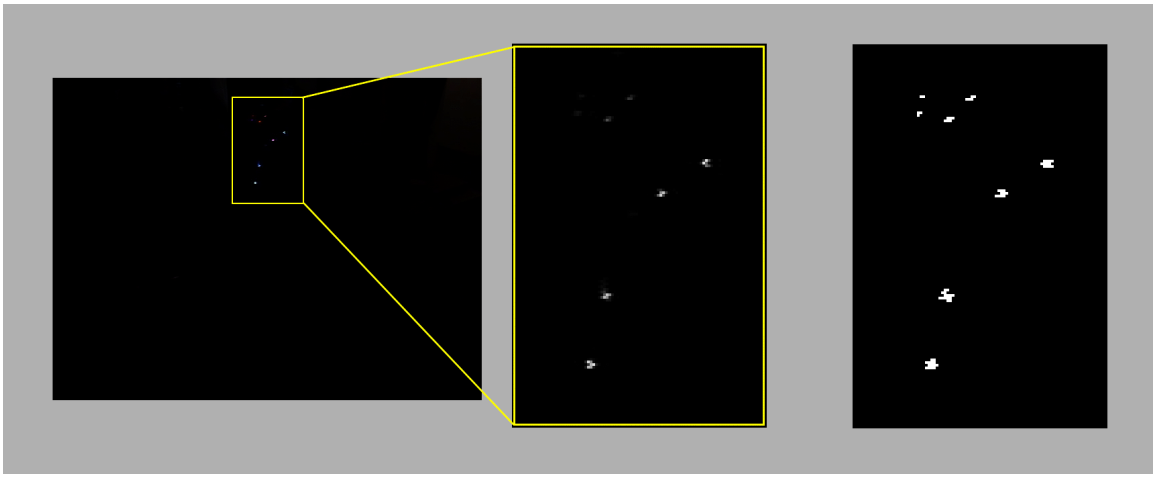


Figure 28 Thresholding to detect target features. Image pixels brighter than some threshold value are marked as belonging to a feature to be tracked. Left. Observed image frame with enlarged image in the region of target features. Right. Thresholded image indicating feature locations.

The fact that background illumination varies spatially and temporally led to a more complex strategy for setting thresholds. A maximum background value is calculated for each pixel. When the LED feature is not present, this value represents the brightest intensity we expect the background to obtain. This maximum background can then be used as a per pixel thresholding value. Brighter parts of the background will have higher thresholds, and darker areas will have lower threshold values. In this way the optimum threshold is obtained in each area. To account for temporal changes in lighting, this background threshold is recalculated whenever a significant fraction of imager pixels exceed the old threshold values. Since real features are fairly sparse relative to the whole workspace, a change in many pixels represents a change in lighting rather than many features being detected.

Despite estimating the maximum background illumination over a number of frames, the background illumination occasionally exceeds the threshold. To prevent unwanted false features from being detected, the background threshold level is further adjusted by both an additive offset above the calculated threshold, and a minimum threshold value regardless of observed background illumination. These two values can be modified for each camera, interactively from a command console, while the system is running.

Figure 29 shows a diagnostic window for one camera. Although individual values are calculated for each pixel, in order to better visualize the intensity levels, the two dimensional image is reduced to a one dimensional function by plotting the maximum value in each image column. The red line represents the calculated maximum background illumination level. The green line shows the threshold value, this value is calculated as a function of maximum background illumination, additive offset, and minimum threshold. The blue line shows the current intensities observed by this camera. Wherever the current intensity exceeds the threshold line, a feature will be identified. In this figure a single LED feature is present in the camera view. We can see that this bright spot causes a peak in image intensities that will be identified as a feature, while all other parts of the image are below the threshold level.

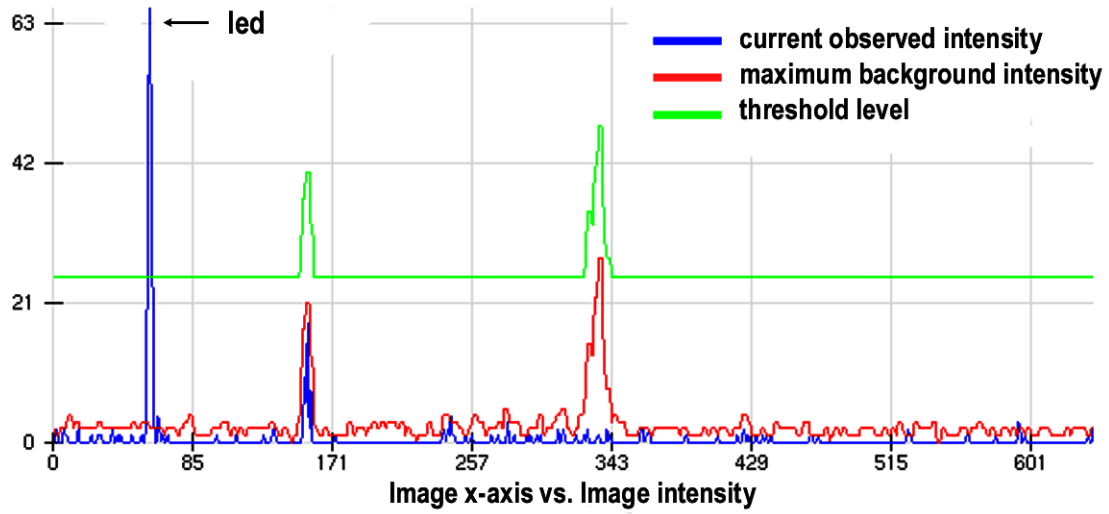


Figure 29 A screenshot from an interactive visualization of current threshold parameters and camera observed intensities. In this case a single feature in the camera view exceeds threshold parameters.

The various threshold values do not need to be adjusted during every use of the motion recovery system. However, under particularly difficult lighting conditions, it is difficult to guess at good threshold settings. In these cases, interactive visualization and control of these parameters is critical.

Masking invalid areas

Sometimes scene lighting is so irregular and/or noisy that thresholding can not account for it. This often happens in the case of computer monitors. These light sources are very bright and irregular. If for instance a user opens a new window with a white background, this represents a new very bright object in the working volume that was not previously accounted for by the adaptive thresholding. This window will be mistaken for an observed feature. Clearly, eliminating the false features that often appear in these areas is desirable. Commercial motion capture systems typically address this issue by requiring the capture area be located in a room with carefully controlled lighting. This is obviously a disadvantage when one wishes to recover motion in natural environments.

Our system allows interactive selection of invalid regions from each cameras viewpoint. Figure 30 shows the features reported from a particular camera viewpoints over the course of a short capture session. This camera exhibits many false features in addition to the correctly identified target feature path. These regions of false features correspond to monitors. By interactively marking these regions as invalid, as shown in Figure 31, the operator of the system can eliminate many false features. We have found that marking these regions is very easy for a user, typically requiring only a few seconds. Furthermore, since monitors are typically static, these invalid regions need be specified only occasionally.

Although marking invalid regions results in a net gain in robustness, it reduces the effective region of coverage from each camera. That is, if a target feature moves into the invalid region, it will of course not be reported. Fortunately, the system as a whole uses many cameras from many viewpoints. Although a feature may be disregarded from one viewpoint, cameras from other directions will still observe the target. It is unlikely that a target will fall within the invalid zone of every observing camera.

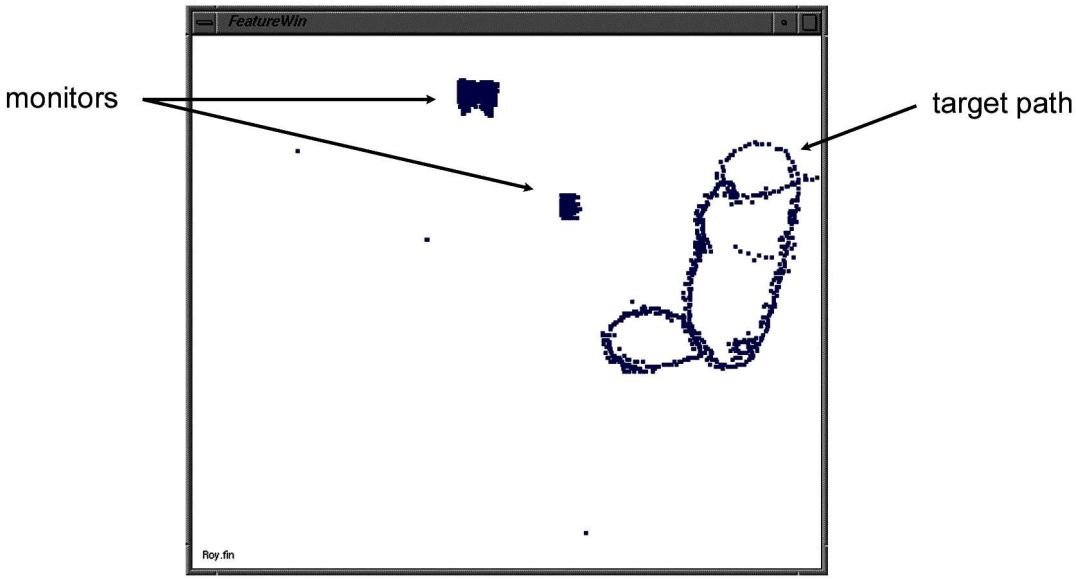


Figure 30 Features reported by a particular cameras. This camera is reporting many false features in the region of a monitor. By selecting this region as invalid, false features can be eliminated.

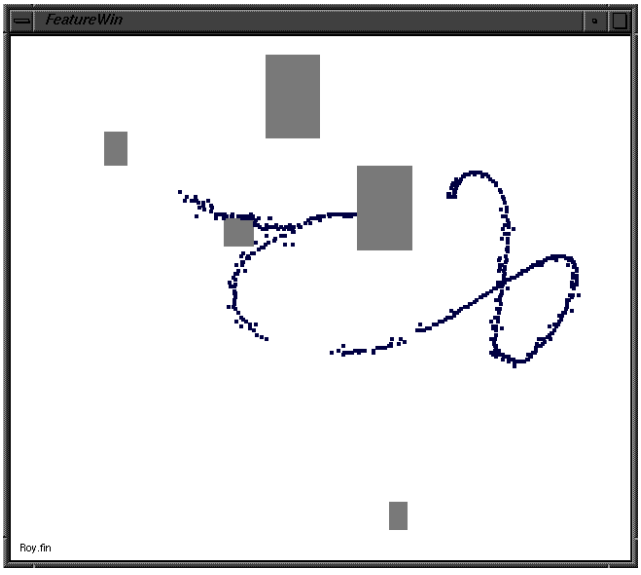


Figure 31 Marked invalid regions. Regions of the camera view frustum which correspond to sources of noise have been marked as invalid, and false features are no longer present.

Subsampled feature detection

Detecting features in each video stream generates a sustained and fairly heavy computational load. Most commercial motion capture systems employ a specialized piece of hardware which searches each video stream for target features. Our system uses standard desktop computers to digitize video and search for image features. A modern processor can in fact, search an entire video field in the 1/60 of a second available before another field becomes available for processing. However, computational resources can be minimized by sub-sampling each video frame, rather than searching the entire frame.

There are a number of reasons to investigate methods of reducing computational load. The first is merely practical, our system uses primarily SGI Indy's for digitizing which have 133 Mhz MIPS R4600 processors. These processors are not quite fast enough for real time performance. Some cameras are connected to newer Intel machines, in these cases the machines are also performing other tasks so that feature detection can not be allocated the entire CPU budget.

A second reason to minimize computation load is overall system latency. After a video field is digitized, features are detected, and then merged to obtain an estimate of three dimensional location. This estimate is obtained sometime after the target feature was actually at a given location in space. For real-time applications of motion recovery, such as ours, minimizing latency is an important concern. If the number of milliseconds required to detect features can be reduced, the overall system latency is also reduced.

One method for reducing computational load is sub-sampling. A number of strategies are available for sub-sampling video frames as they are digitized. The most obvious pattern is a regular grid. Since the spot created by the LED on the camera imager is often larger than a single pixel, merely searching every N pixels will usually identify all features. Of course very small features might slip through the sampling grid. If the feature is not moving then it may remain in one of these gaps for an extended period. In order to improve robustness over time, the sampling grid is jittered on each field, so that features in all locations will eventually be reported. After identifying pixels on the sub-sampling grid that pass the feature detection threshold, a small neighborhood is also searched in order to provide an accurate estimate of the feature centroid.

Most features move in predictable patterns across each cameras image plane. This regularity can be exploited by searching for features more carefully in regions where they are expected. In our system a small window around predicted locations can optionally be searched. The image plane location in which each feature was detected in the previous frame is a likely location to find that same feature in the current frame. Predicting the expected feature location based on estimated image plane velocity is another good method for finding features. Figure 32 shows a visualization of pixels that have been searched in an example video field. The predicted window and background sampling pattern are both apparent. In our system, good results have been obtained with very low computational requirements by using predicted search windows of size 40x40 pixels in which every pixel is searched, and reducing the overall frame sampling to one sample in each 5x5 pixel block. Using these values, a 640x240 pixel field can be searched for features in under 8 ms.

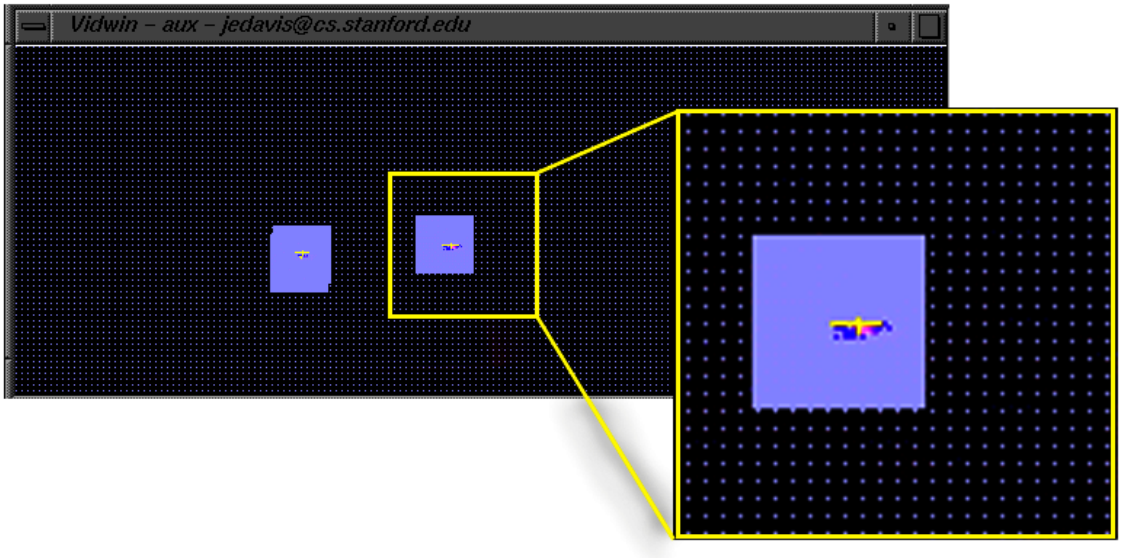


Figure 32 Visualization of which pixels have been searched for target features. Every pixel is checked within a window around the predicted feature location. Sparse subsampling of the entire frame is used to identify new features.

3.4 Integrating observations

The large scale tracking system has many cameras each of which generates observations. Individually, these observations are insufficient to determine the location of targets in three dimensions. Observations from multiple cameras are required. Unfortunately neither

correspondence between camera observations nor across time is known. Ideally every observation would come with a tag that uniquely identified the target. In practice this labeling must be inferred from the positional data in order to integrate observations.

Commercial motion capture systems choose to solve only part of this problem. Systems such as those from Motion Analysis and Vicon have been engineered such that cameras are synchronized. This means that observations from all cameras occur simultaneously and correspondence between cameras can be determined separately at each time instant. This simplifies the situation since correspondence across time is no longer critically intertwined with correspondence between cameras. Each observation can be projected to a ray in space as shown in Figure 33. All points at which several rays intersect are determined to be the 3D location of a target.

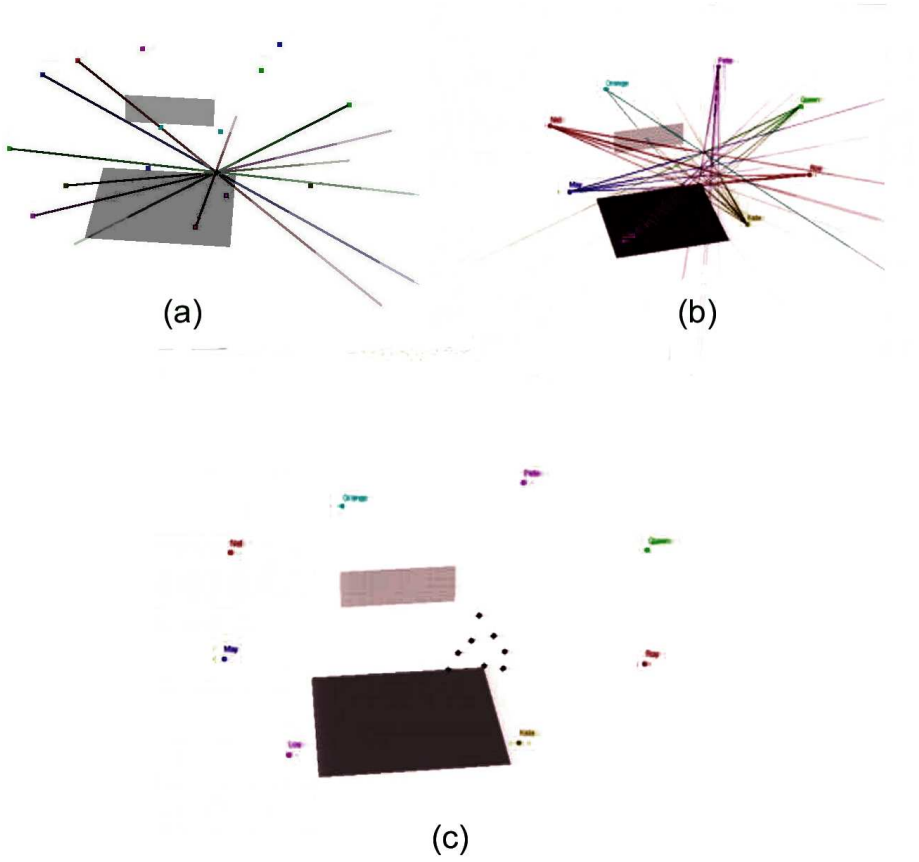


Figure 33 Intersecting rays can be used to determine the three dimensional location of targets. (a) A 3D view of a set of camera observation rays for just one target. Colored dots are cameras mounted on the ceiling, and each line represents a camera observation. (b) The set of observing rays for multiple targets. (c) The target locations extracted by intersecting these rays.

Target locations extracted by direct ray intersection are independent at each time instant. This creates a difficulty when attempting to use the data in a real time system such as the one described in this work. Consider the extracted target locations in Figure 33(c). There is no labeling to differentiate targets, only an unorganized collection of points. Suppose that the pan-tilt cameras in the mixed scale recovery system are to be directed to observe the target attached to an actors head. There is no information available to suggest which target this is. Furthermore, and more importantly, this determination must be made independently at each time instant. If inconsistent decisions are made then the pan-tilt cameras will appear to jitter between targets. Of course commercial systems do attempt to determine some labeling and correspondence between these targets, but it occurs as a time-consuming user assisted offline process, unsuitable for use in a real time system.[1]

One approach to obtaining consistent labels for observed targets is to maintain a dynamically updated motion model for each target. This motion model keeps track of the current estimate of a target's state, for example its position and velocity. Now rather than obtaining an independent set of target locations at each time instant, there is a known set of targets. The motion of these targets is estimated over time based on observations. One benefit of using a motion model is that targets are implicitly labeled. As seen in Figure 34, each target has an identifier, which stays with it over time. Because of this we need only decide to direct pan-tilt cameras to observe a particular target, and consistent behavior will be possible.

While commercial motion capture solutions have not made mainstream use of a motion model, the benefit of using a motion model has not gone unnoticed in the academic community. Many possible formulations for maintaining and updating a motion model have been explored by the computer vision research community. Kalman filtering [21] essentially estimates the current target motion using a Gaussian probability distribution. Many researchers have adapted it to work with edge, contour, and texture features, usually in conjunction with a kinematic body model [63, 76, 98, 107]. Mixtures of Gaussian probabilities have also been used as representations for motion state [23]. Recently, the

[1] Very recently several commercial optical motion capture systems have begun selling 'real-time options' in which they attempt to maintain target correspondence across time. In theory this can provide the real time motion data needed to drive an animated model, however personal conversations with several professionals who have attempted to use these options suggest that they are still extremely fragile.

CONDENSATION algorithm proposed by Isard and Blake has become a popular method for tracking motion using a sampled probability distribution function [61]. As with the other methods, a wide variety of specific feature, motion, and pose models have been explored [79, 95, 96]. Nearly all of this work uses a complex geometric model of articulated object pose, and a relatively straightforward motion model which estimates position and velocity. In addition, it is common in much of this work to track only a single object. In contrast, the radar tracking community solves a problem more similar to the one we consider, that of tracking multiple point features in the presence of clutter [10, 11, 15]. However in both communities the methods for updating a chosen motion model are similar.

Figure 34 Rather than determining target location separately at each time instant, a motion model can be used to maintain an estimate of target location. One of the benefits of using a motion model is that consistent target IDs can be maintained. Here we see target locations with their associated IDs.

Another benefit of using a motion model is that it allows for unsynchronized observations. The motion model provides a constraint across time. Many systems rely on simultaneous observations, so that constraints can be employed between viewpoints. A motion model effectively acts as a regularization term, so that observations from different points in time can influence and constrain one another. Most methods for updating the state of a motion model can deal naturally with an observation that occurs at any point in time. For example, Welch and Bishop leverage this property of a Kalman filter to obtain improved tracking performance [110].

One last benefit of using a motion model is increased robustness to occlusion. When target observations are completely independent in time, an occlusion will cause the target to be completely lost. When a motion model is used, the model acts as a predictor of future positions and velocities. If the target is not observed due to occlusion for a short time interval, then the model may be sufficient to accurately predict the target's location. When the target again becomes visible, observations will again influence the estimated target state. Thus, the model improves robustness to short intervals of occlusion.

As observations are made by cameras, the central estimator needs to associate each observation with the most likely target model. We have found that projecting each target's current position to the camera image plane, and merely associating each target with the nearest observed feature works adequately. The use of a gate or verify region further improves reliability. More complex data association strategies have been developed, and an introduction can be found in textbooks on target tracking [10, 15, 87].

The system described in this thesis uses a Kalman filter to estimate the position and velocity of each target in the working volume. This method was chosen due to its relative efficiency, making it ideal for online tracking. The details of the Kalman model are not relevant to this discussion, however a comprehensive description of the filter details specifically within the context of this system, is presented by Chen [25]. In addition several good introductory texts on Kalman filtering exist [21, 109].

3.5 Time and communication

Anticipating and managing potential difficulties with time, latency, and communication explicitly can result in improved tracking performance. The Kalman filter used to integrate observations from multiple cameras assumes that these observations are accurately time stamped and instantaneously delivered to the filter. In practice it is impossible to keep precise time in a widely distributed network, and latencies are likely to vary both between cameras and over time. Naively applying the theoretical update model for each observation will result in poor tracking performance.

Since observations are recorded on a distributed set of machines, it is possible for unexpected system or network load to greatly delay an observation report. These reports are received on a

central machine which collects and distributes these observations to the appropriate Kalman filters. Suppose that an observation was delayed by two seconds. By the time it could potentially be used to update the expected state of a tracked target, the information is no longer relevant. In the case of an offline system, observations can be reordered according to the time of observation, but this is not feasible for a real time system such as ours. Each Kalman filter maintains an internal time indicating the most recent update. Blackman suggests that simply ignoring late observations results in only a marginal increase in tracking error [15]. However this result implicitly assumes that accurate time keeping is available.

Unfortunately a policy of discarding old observations, coupled with an imperfect notion of time can have adverse results. For example, suppose that a set of cameras labeled A,B,C is tracking a target. Further suppose that the clocks on these machines are synchronized to within a few milliseconds and that all other processing and network latency is equal. It is possible that only one camera will ever contribute observations to the filter, while the other cameras starve. This can happen with even very small clock offsets. Figure 35 shows a diagram of one possible sequence of events. Note that camera A has only a small clock discrepancy, but because it is systematic, it results in starvation for the remaining cameras. Due to starvation, target pose estimation can not proceed, since it requires multiple viewpoints.

In practice it is impossible to insure either completely consistent time stamps or delay through the entire processing pipeline. Even using a standard and well studied time synchronization tool [72] resulted in clock drift of up to 100 ms, in our experience. This is a very large clock discrepancy considering the fact that video observations occur every 16 ms. Managing time explicitly within the code of the various servers and distributed observation machines in this system brings the discrepancy down to the level of network latency, 2-6 ms in the case of this system. This level of discrepancy can still lead to starvation as we have seen. Thus a strict strategy of discarding old observations must be abandoned. Observations that are only slightly delayed, within the expected and systematic range of time discrepancy, should be used to update the current estimate of target state. Observations which are delayed for an unexpectedly long time should be discarded since using very old information will not enhance the current estimate.

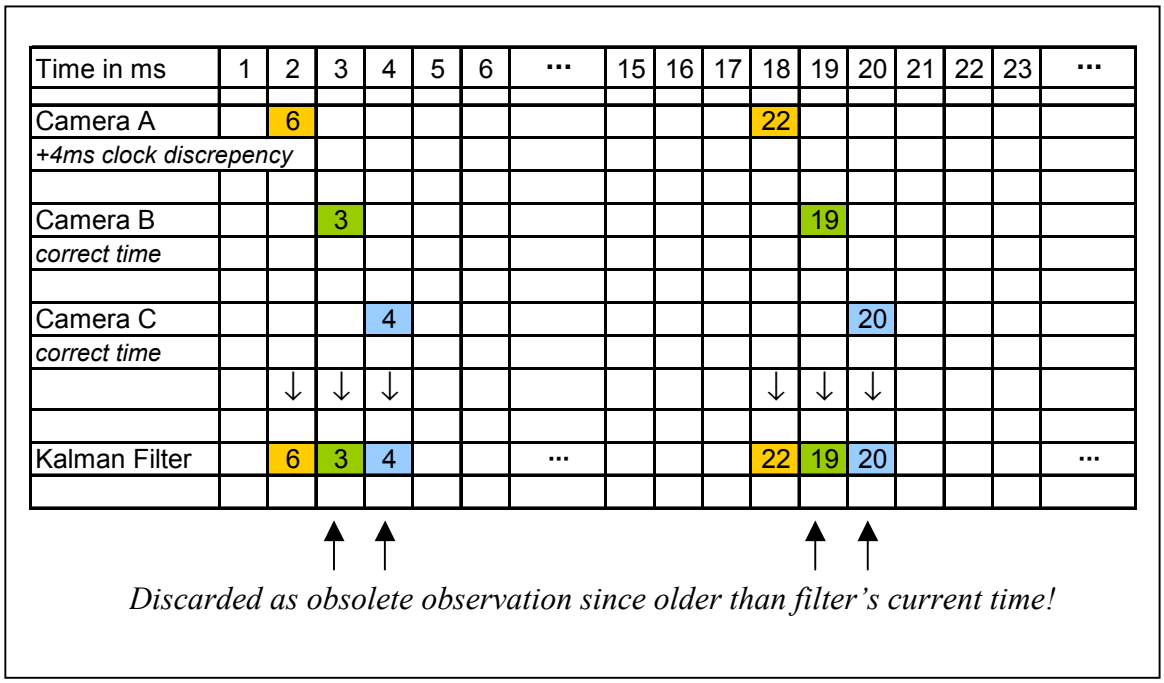


Figure 35 Camera starvation. This sequence illustrates how only a small clock discrepancy can create a situation where all remaining cameras are effectively starved. Camera A observes the target at time 2ms, but because of a 4ms clock discrepancy tags the observation as occurring at time 6ms. This observation is then integrated into the Kalman filter. Camera B observes the feature 1ms later at time 3ms, and correctly time stamps the observation. An attempt is made to integrate this observation into the state of the Kalman filter, but because it is tagged as older than the current state, it is discarded. A similar situation arises with Camera C. Since all cameras will continue to make observations at precisely 60Hz, the situation will repeat and cameras B and C will effective starve, due to a relatively small clock discrepancy on the part of camera A.

When using many fast cameras and many targets it may be true that the central estimator receives observations faster than they can be integrated with each Kalman filter. Since a large latency is not desirable, some observations will have to be ignored. Using an appropriate method to choose which observation to next integrate with each Kalman filter is important in order to avoid camera starvation.

After processing an observation, the central estimator is faced with one queue per camera, each of which may have several waiting observations. One possible policy would be to drain all the queues and process only the most recent observation discovered. However this can lead to starvation for reasons similar to the previous discussion.

In order to prevent starvation, round robin selection could be employed, choosing from each camera in turn. In this case each camera would be insured service. Since the queue could have several observations waiting, only the most recent observation from the given camera

would be used. While this does prevent starvation, it leads to a situation where unnecessarily old observations are used to update the target state. Suppose that one camera has a slow observation rate, but not latency, relative to the other cameras. For the sake of example, let us suppose normal cameras update every 16 ms, while the slow camera updates every 100 ms. When this camera's time slot arrives in the round robin progression, its most recent observation may have been sitting in the queue long enough to become quite old relative to the current system state. Ignoring the observation will lead to camera starvation. Using the observation at this point will potentially lead to 100 ms old data being integrated into the current estimate.

The dilemma regarding old observations can be avoided by only considering observations that occurred during the most recent processing interval. Each time an observation is chosen for processing, all camera queues should be emptied so that the next observation is chosen only from those that have arrived during the most recent processing interval. Round robin processing under these conditions can lead to starvation of slower cameras. Instead, a camera should be chosen randomly from the set of available observations. A slow camera will less often produce an available observation, but when it does, it will have a chance to be selected so that starvation does not occur.

While the above analysis may seem overly complicated for a theoretically homogenous set of cameras, in practice, there is nearly always one camera or machine which is behaving abnormally. Given the difficulty in insuring that every component is functioning correctly all the time, it is imperative the system behave well in the presence of non-ideal data flow, timing and processing requirements. The discussion above represents the evolution of this system as problems that actually occurred were diagnosed and corrected.

3.6 Adding and removing subsystems

When a new object enters or leaves the working volume, targets need to be added or removed from the active set. The system as described so far assumes that a known number of targets, and thus Kalman filters, are active and remain active throughout the entire tracking sequence. Even if this were true, the system would need to be initialized with the number of targets currently in use. A much better solution is to automatically add and subtract systems from the active set as they appear or depart from the observable volume.

Removing subsystems

When a target leaves the working volume it should be removed from the active set of targets and associated Kalman systems. Unfortunately no signal that this has occurred is available. Since not all cameras observe all targets at all times, simple lack of observation by a camera is insufficient to conclude that a target has left the working volume. In fact, due to random failures of the tracking module and temporary occlusions, even when no cameras is able to observe a target, it may still be active. However, targets that remain occluded for an extended period are likely to have moved considerably from their last known position, making data association and thus continued tracking difficult or impossible. Thus, even if a target is in fact still active, if it hasn't been observed for an extended period, it is better to remove it, and let a new target be created when it again becomes observable.

If a target is unobserved by any camera for some period of time, it should be removed from the active set. A timeout of 100ms-300ms empirically seems to work well with this system. Setting the timeout period too short will cause targets to inadvertently be removed, only to be recreated immediately. In addition to unnecessarily using computational resources, this causes the target ID to change, which is undesirable since some applications require consistent IDs to function correctly. Setting the timeout period too long will leave phantom targets floating in space. These targets will continue to move according to their last known velocity. The phantom targets are not only incorrect, they act as potential confusing influences during data association.

The best timeout period of course will vary depending on the details of a given tracking system and the dynamics of the targets being tracked. The timeout period needs to be long enough that every camera has a chance to declare no observation. For example, in the case of this system, cameras sometimes run as slow as 30Hz, so the period should not be less than 33ms. In addition, since the tracking and low level feature detection is not completely reliable in every frame, it is advantageous to wait several cycles to determine if the target is really unobserved. In this example, a 100ms timeout period is equivalent to believing that the low level tracking succeeds in observing a target within all sliding windows of three consecutive frames.

In addition to failure of low level tracking, a target could genuinely be occluded. Even when this is true, if the occlusion is short lived, then the predicted target motion will be nearly

identical to the actual target motion. When the occlusion ends, the target will again be observed and actual tracking will be resumed. Robustness to short occlusions is in fact one of the principal advantages of using a temporal model for target motion, the predicted velocity and position allow a target to be reacquired for continued tracking after a brief disappearance. In order to make use of this advantage, the target should of course not be removed from the active set before the occlusion ends. As mentioned previously, if the occlusion is extended the predicted motion is likely to poorly match the actual motion and the target will not be recovered in any case. The length of occlusion that can be tolerated is related to target acceleration. In the absence of acceleration, prediction will perform perfectly over any time period. Under extremely high acceleration the predicted position will diverge almost immediately. The timeout period should be set long enough that targets can be recovered after brief occlusions, but not so long that incorrect diverging targets remain in the working set. The given range of 100ms-300ms empirically works well with the kinds of motion encountered by this particular system.

Merely describing a target as observed or unobserved is insufficient. It is possible that only a single camera observes a target. For a brief duration this is the expected state. Since the cameras in this system are not synchronized no two cameras observe the target simultaneously. Typically however, another camera observes the target from a new viewpoint after only a very brief interval. If an extended period passes without a second observing viewpoint, the estimate of position will be degraded. A single observing viewpoint is not sufficient to fully constrain the 3D position of a target, instead the observations from a single viewpoint constrain the target to lie on a ray in space. Conceptually, additional viewpoints determine the target's depth along this ray. In the absence of additional viewpoints the target will drift arbitrarily along the ray.

Since incorrectly reported target positions cause greater difficulty than missing targets, systems are removed from the active set when they are observed by only one camera for an extended period. The same timeout threshold used for completely unobserved targets seems to also work well for targets observed by only one camera.

Adding subsystems

When a new target enters the working volume, a Kalman system dedicated to estimating its motion must be instantiated. Similarly, after a target is occluded and removed from the active set, and then again becomes visible, a new Kalman system must be created.

Whenever a camera observes a feature that can not be associated with any existing target, this indicates that a new target may have appeared. Unfortunately, this is not always the case. Cameras also observe a great deal of sporadic noise. Temporary bright reflections from objects, and multiple reports of a single feature occur often. It is not desirable to create a new system in these cases since no corresponding target exists. In order to robustly create systems when an actual target appears, while ignoring noise, confirmation from multiple observers is required.

A real target will be observed by multiple cameras in a mutually consistent manner. Each observation indicates a ray in space upon which the target must lie. A stationary target will generate multiple observing rays which intersect. Accidental features due to specular reflections and other causes will generally not result in multiple intersecting rays, since the effects which give rise to this noise are often viewpoint dependant. The more viewpoints that confirm an observation with intersecting rays, the more certain it is that an actual target is present. Empirically, requiring only two confirming observations appears to work well.

Requiring even more rays to cross in space theoretically would further increase the robustness of the system. However, since the system described here has only eighteen cameras in the wide area system, it is not expected that many cameras could possibly observe a target, even under ideal conditions. If a system was built with many more cameras, then a greater amount of confirmation would be desirable.

In the absence of noise, observation rays intersect exactly in space. In reality sensor noise prevents this ideal condition. Rather than exact intersection of observation rays, a intersection threshold tolerance is used, such that rays that come within epsilon of coincidence are considered intersecting. The correct setting of this threshold depends on expected imager resolution and noise. If the threshold is too small, matching observations will not be found to intersect and new targets will never be created. If the threshold is too large, the likelihood of unrelated observations accidentally being found to intersect will increase. In this system, a threshold of 15 mm was empirically found to work well.

Since cameras are not synchronized, observations do not occur simultaneously. Conceptually, targets should be created whenever multiple observations confirm the location of a target. However, observations from the distant past should not be allowed to influence decisions about the current location of targets. Ideally, only current observations should be used to locate and verify the position of a new target. However, it is not strictly possible to know which observations are current at all times. In addition to using only the most recent observations from any given camera, a threshold observation period is used, after which an observation is considered stale. Stale observations are not used to determine or verify ray intersections. As with the thresholds for removing targets, the threshold for determining a stale observation depends on the particular system design. If the period is too short, then not all cameras will have an opportunity to report observations that may be relevant. In this case targets will not get created properly. If the threshold period is too long, then old observations may be used to determine intersections, resulting in incorrect phantom targets being created. A threshold of 70 ms empirically works well with the system described here.

A moving object observed at different times will appear at different locations in space, thus the rays will not intersect. However, if observations are relatively close in time, the object will not have moved far and approximate intersection will occur.

The above discussion has assumed that targets are stationary. In practice most targets are moving. Since ray observations are not simultaneous, rays observing moving targets will not intersect. This provides an additional motivation for the intersection threshold tolerance described above. If the motion of the target has a low velocity, the distance between rays will be small and will fall within the threshold. Thus this threshold also controls the maximum velocity at which a target can move and still be detected and added to the active set of targets. While it would be possible to reformulated the target addition rules to account for motion, fast moving targets are often difficult to track robustly. It is desirable for targets to be robustly observed when they are newly created, since a number of internal Kalman system state parameters are estimated based on these observations. Consequently, adding only slow moving targets to the active set is a desirable policy.

Only observations that are unmatched with any existing active targets are considered when creating new targets. Thus there is a relationship between the data association subsystem which matches observations to targets, and the subsystem described here which creates new targets. If errors are made during data association, either too few or too many new targets are

likely to be created. In particular, if the Kalman gate or verify region is set too high, all observations will be associated with some existing target and no observations will ever be considered by the target addition subsystem.

3.7 Evaluation

The performance of the large scale system can be evaluated quantitatively. Some metrics, such as feature detection accuracy, can be measured in a relatively straight forward manner. An analysis of both image plane and spatial feature detection error is given. However, the measured accuracy is sensitive to several parameters of the Kalman filter used for observation integration. While quantifying these effects is difficult, a qualitative discussion of some of the effects of these parameters is provided.

Each camera produces a video stream from which features must be extracted. The accuracy with which these features can be extracted affects the performance of the rest of the system. In order to evaluate the accuracy of feature detection, an LED was mounted on a linear translation stage, and the path of this LED was measured as it was moved along the controlled path. Each camera observes this motion from a unique perspective, and the observed path from nine of the cameras is shown in Figure 36.

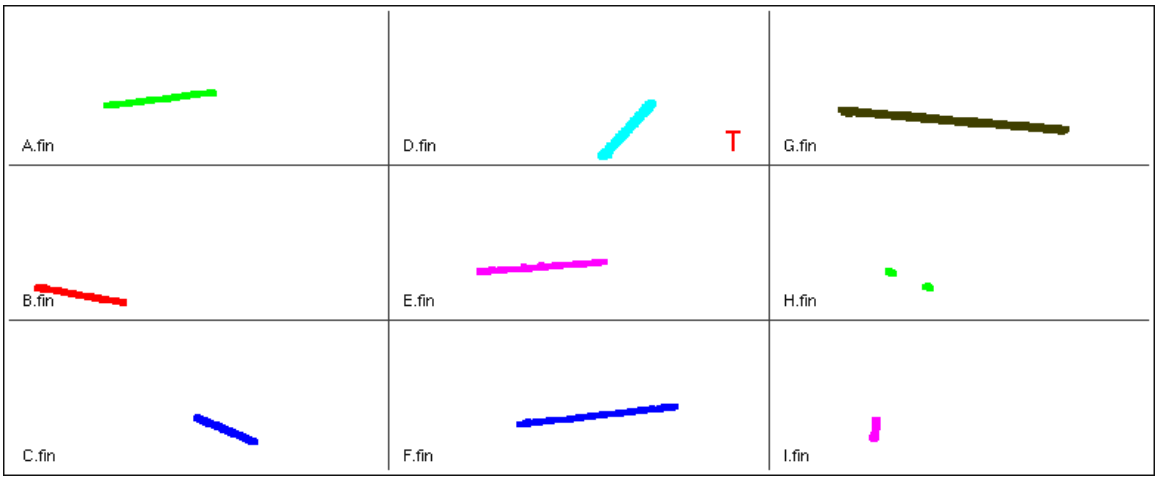


Figure 36 Observed path of an LED from nine camera viewpoints as it was moved along a controlled linear path.

Under ideal conditions when a target is moved along a linear path, the observations of the target will also precisely lie along a line. However limited imager resolution and imaging noise cause deviation from the desired path. Figure 37 shows the observation line produced on two of the observing cameras. It is clear that while the observations are approximately linear, there are significant random perturbations in the path. Since this plot is in the space of the image plane, it is not immediately obvious where the errors originate. Rather than plotting the observed feature locations along a parametric line on an X-Y plane, the observations can visualized as time series in which the X-axis and Y-axis measurements have been separated, as shown in Figure 38. From this plot it is more obvious that a significant portion of the error is due to pixel quantization. Since sub-pixel feature estimation is not employed, the observed feature location oscillates between adjacent pixel locations.

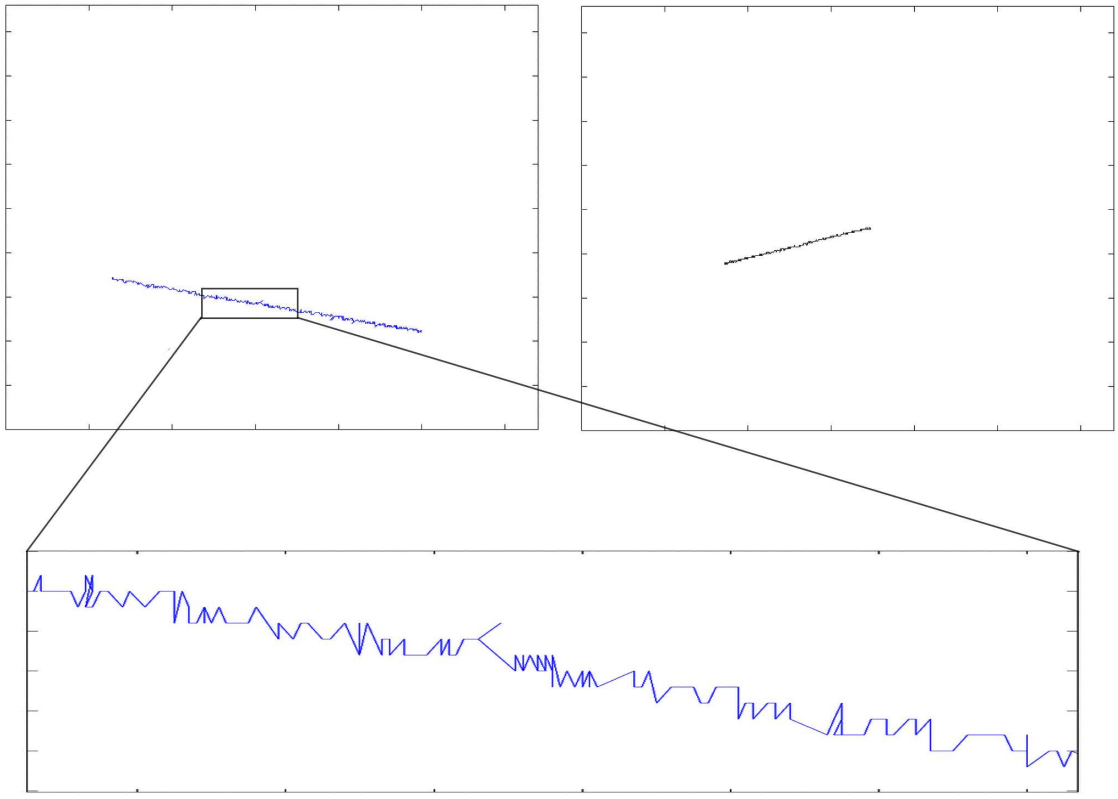


Figure 37 Noticeable noise inflicts the observed path of an LED moved along a linear path. These plots are in the space of the XY image plane.

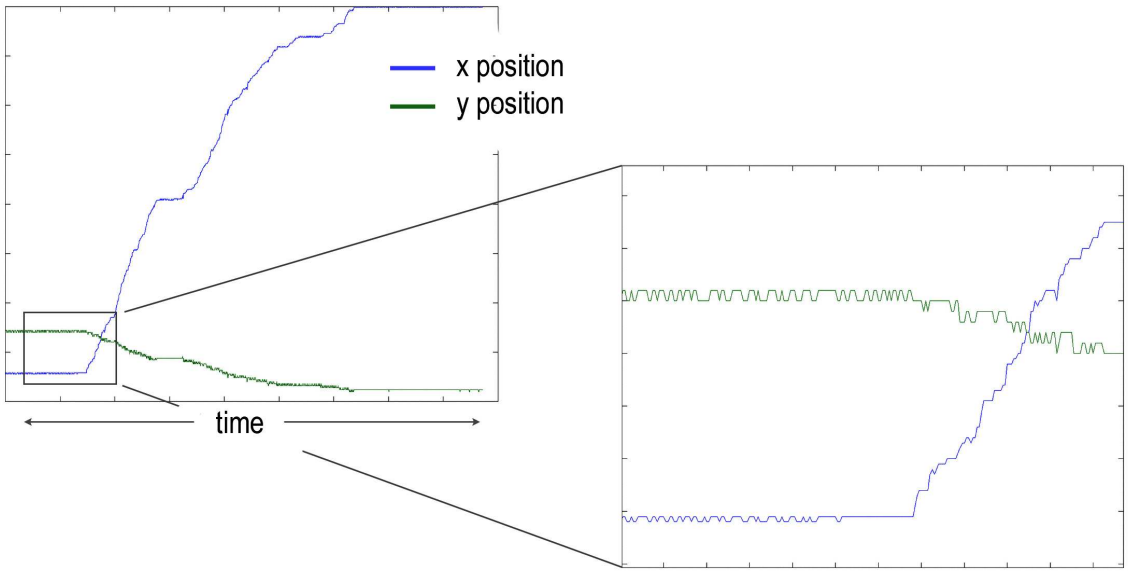


Figure 38 Plot of observed LED position. X-axis and Y-axis motion have been separated into separate time series. It is clear that a great deal of the noise is due to integer quantization of pixel position.

Image plane observation error can be quantified by fitting a line to the observed measurements, and determining the distance in pixels that each observation lies from this ideal line. Figure 39 shows plots of the fitting error for each of the two lines considered above. Ideal lines were determined by fitting the data in a least squares sense, and error is the Euclidean distance between an observation and the closest point on the line. It is interesting to note that the character of these plots is quite different. The first line exhibits quite irregular noise characteristics, while the observations from the second camera have much more uniform error. This can be attributed to the natural variability of the process. The LED itself has directionally dependant output, and cameras are mounted in varying orientations and distances from the motion in this trace. Some cameras are likely to observe the LED more robustly than others. Despite the fact that the first trace has occasional observations as distant as five pixels from the ideal position, both traces have excellent average behavior. The mean distance from observation to the ideal line is 0.94 pixels and 0.71 pixels respectively.

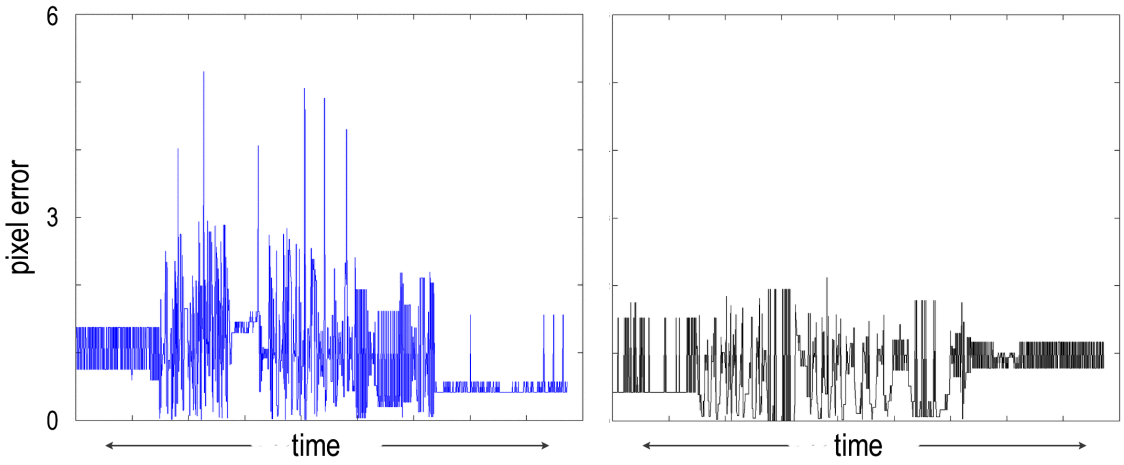


Figure 39 Pixel error of observed target locations relative to ideal linear motion. Observations from the camera on the left clearly have more irregular noise characteristics than the camera on the right. This is due to natural variation in viewing angle and distance.

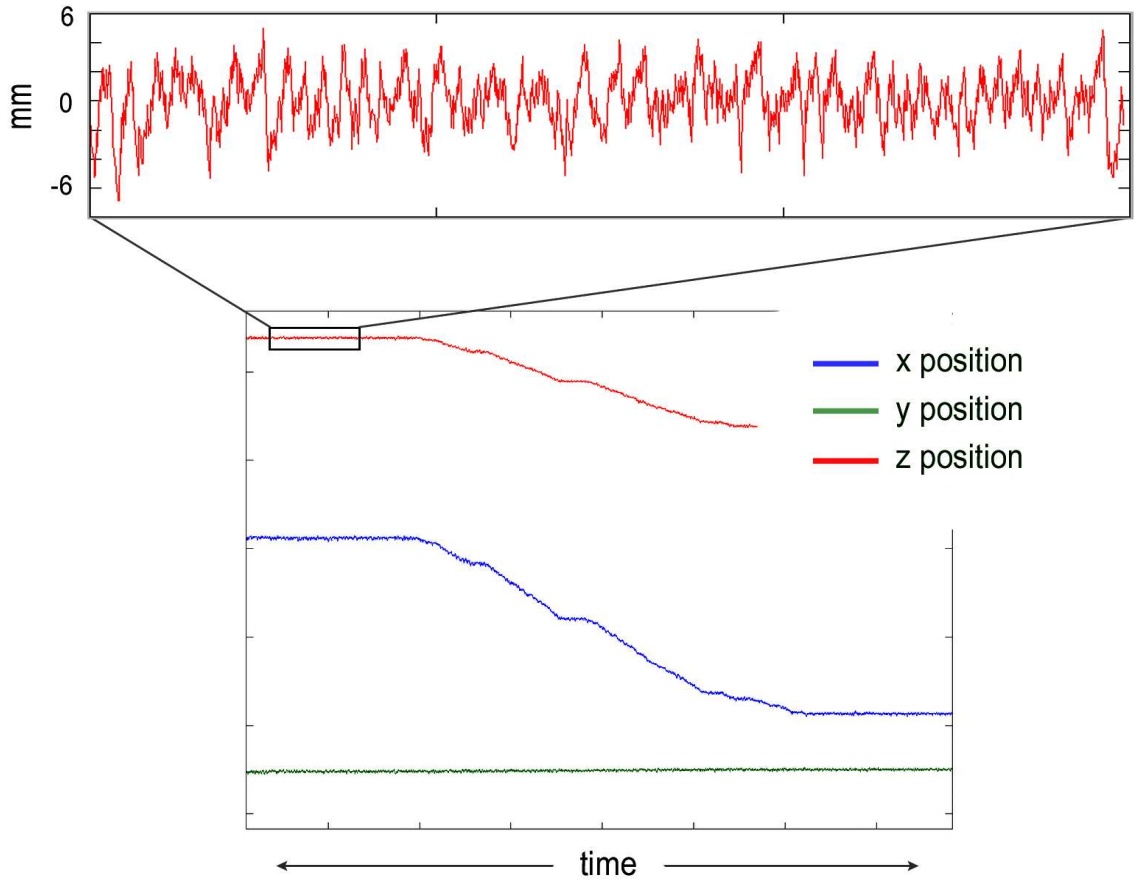


Figure 40 Time series plot of target position reconstructed by the large scale capture system. The enlarged section of the graph shows jitter in the z-axis direction.

While image plane analysis gives an indication of feature detection quality, a more important metric in terms of system usefulness is accuracy of target location in 3D space. The observations from all cameras while an LED target was moved along a linear path were integrated to determine the motion of the target. A time series plot of this linear motion through space is shown in Figure 40. The zoomed in portion of the graph shows the Z-axis motion during a time period in which the LED was not moving. It is clear that the estimated point position jitters in space by as much as 10 mm.

The expected accuracy of the system is dependant on the working volume and the camera resolution. Since the viewing volume of each camera is different, the expected resolution varies spatially. However an approximate calculation using average values is possible. Cameras individually observe an area averaging about 2.0m wide, and an imager has 640 pixels in the horizontal direction. As shown above, observations commonly jitter by 3 pixels, and occasionally as much as 5 pixels. Thus we might expect spatial accuracy in the range of 3-5 pixels × 2000 mm ÷ 640 pixels = 9-16 mm. Thus, observed accuracy is roughly equivalent to the expected range.

The spatial accuracy of the system can be further characterized by fitting a line to the determined target locations. Just as was done previously in the case of 2D observations, target locations are compared to a best fit line and the residual error of each target location is shown in Figure 41. As expected, the target jitters around the line within about 10 mm. The mean deviation from the ideal line is 2.2 mm.

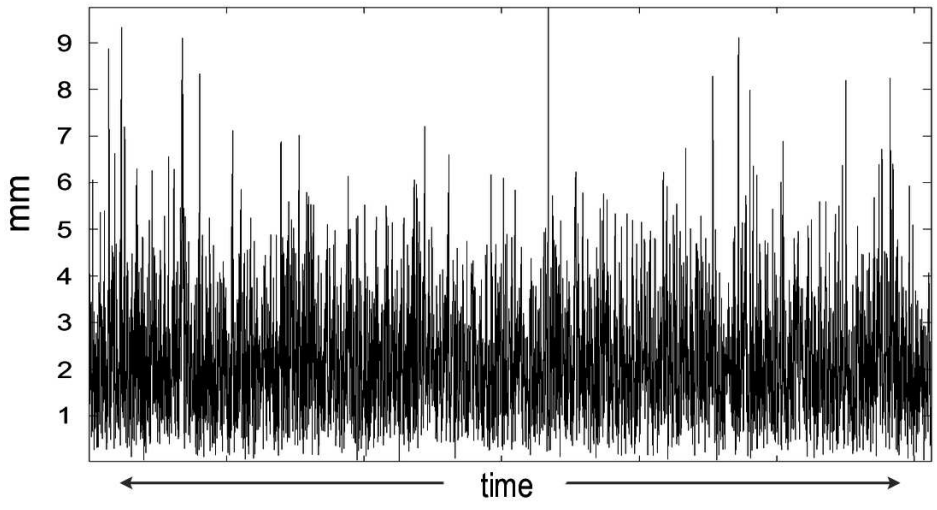


Figure 41 Target position reconstructed by the large scale system was fit to ideal linear motion. This plot shows deviation from the ideal line in millimeters.

Although it is informative to measure the large scale systems overall accuracy, it is not as straightforward as it might seem. The large scale system employs a Kalman filter to estimate target position, and one use of such a filter is to regularize or smooth noise in a data stream. The filter has a number of parameters which can be tuned to provide a responsive but noisy signal, or a highly damped but very smooth signal. For example, Figure 42 shows the position of a target calculated with two different settings for the Kalman filter parameters. In one case the signal jitters substantially in response to noisy observations, in the other the signal is less responsive, instead providing an estimate of the targets average path. It is clear that calculations of the system accuracy will be greatly affected by the choice of filter parameters.

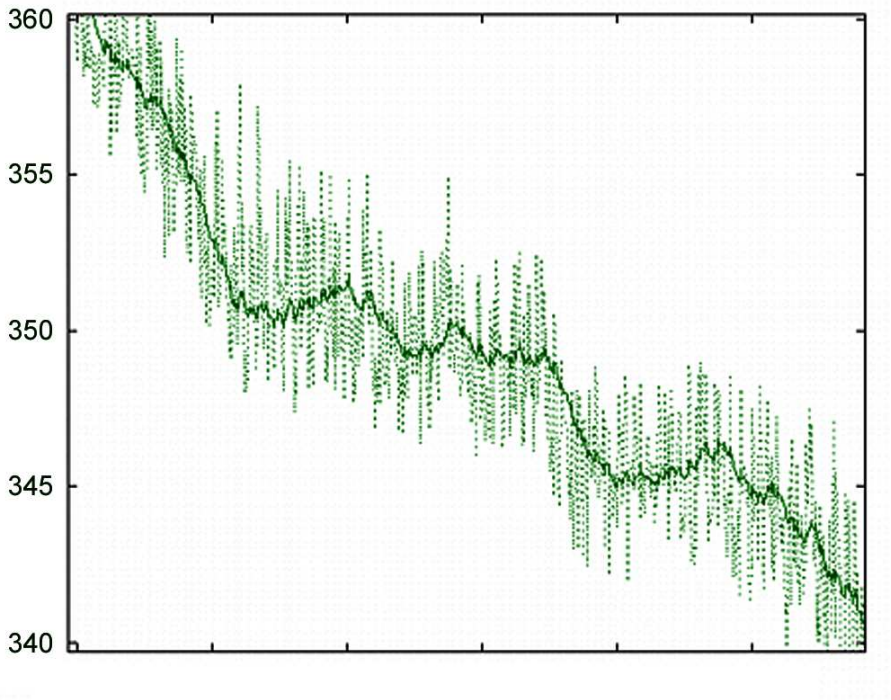


Figure 42 Target position in millimeters, calculated with two different settings of Kalman filter parameters. By changing filter parameters it is possible to provide a range of estimated motion, from noisy but responsive, to smooth but over-damped.

While it is well known that the Kalman filter parameters affect the shape of the estimated motion signal, the parameters have a number of less obvious influences. For example, in addition to accuracy, the filter parameters can also have an affect on the robustness of the system. Figure 43 shows the estimated path of a target that was moved in a circular motion.

The figure on the left shows the path when estimated with reasonable Kalman filter parameters settings. On the right is the position estimate when the filter parameters are set to apply a high degree of smoothing to the signal. In this case target tracking fails. The target becomes lost to the tracker, and continues along a path of constant velocity until it becomes sufficiently stale and is deleted. At this point a new system is created to match the observed target and the process repeats.

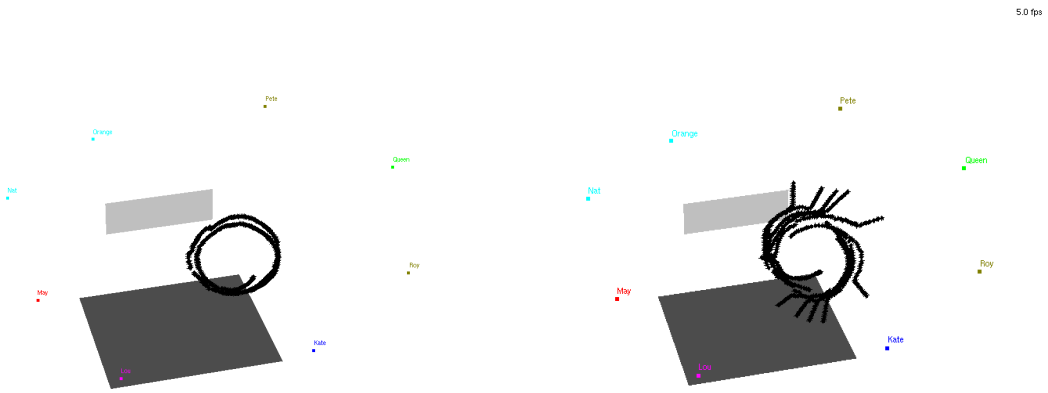


Figure 43 Estimated path of a target moved in a circular motion. (Left) The correctly estimated path obtained with good settings of Kalman filter parameters. (Right) The target path is lost when Kalman filter parameter settings are configured with too strong a belief in a constant velocity motion model. The target is repeatedly lost, deleted, and re-instantiated.

The above behavior can be understood by visualizing the motion predicted by the Kalman filter motion model, in addition to the observed target positions. Figure 44 shows the circular target path from the viewpoint of one camera. The three figure sub-panels show target prediction when different filter parameters are applied. Blue dots are the observed feature location. These are connected via lines to black dots representing the predicted target position. If the motion model exactly described the actual motion, the two dots would always lie directly upon one another and no lines would be visible. The top panel shows the case in which filter parameters are set to apply a relatively weak motion prior. That is, measurements are believed to be very reliable, and the estimated motion is adjusted accordingly. Although the estimated path is sensitive to noise, the estimated motion closely matches the observed behavior. The middle panel represents a slightly stronger belief that the motion model is

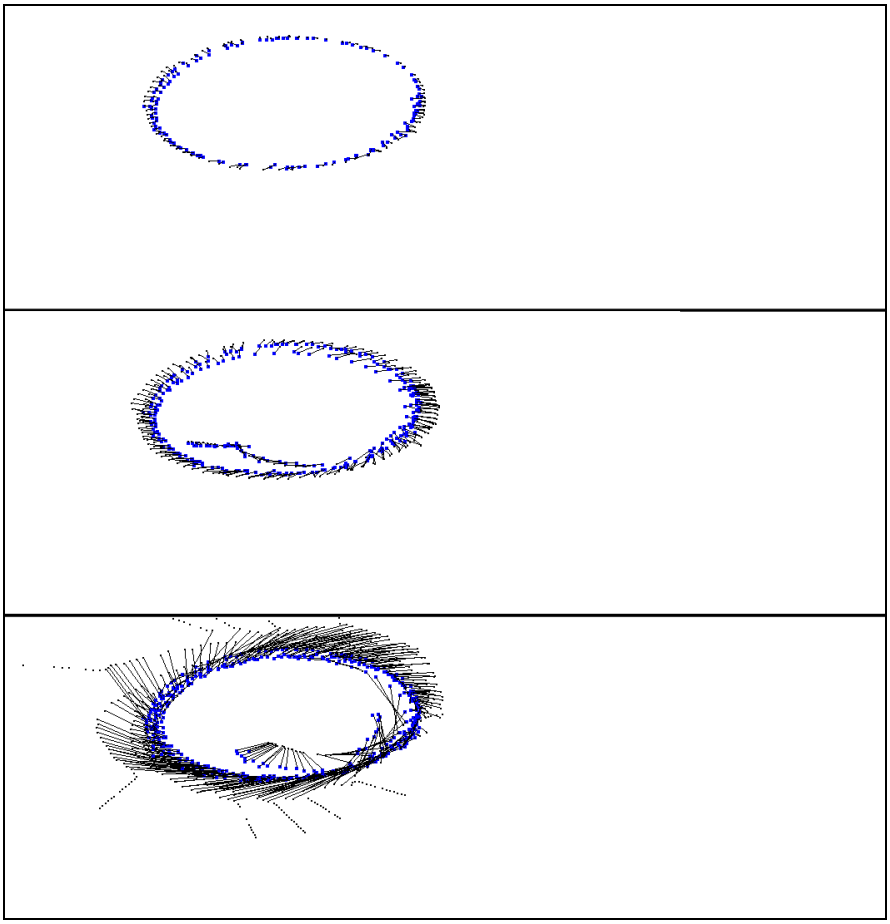


Figure 44 Observed path of a target along with target locations predicted by the Kalman filter motion model. Blue dots are observations, and black dots with connecting lines are associtated predicted target location. (Top) Low strength motion model. The predicted path is noisy, but positions stay close to actual observations. (Middle) Medium strength motion prior. The predicted path is somewhat smoothed, but predicted positions more strongly follow prior velocity and lie further outside the observed motion. (Bottom) High strength motion model. Predicted target positions too strongly follow prior velocity and tracking is lost, with targets continuing under constant velocity.

reliable. In this case, inconsistent observations will be regarded as noise and the model will more slowly adapt itself to reflect new position and velocity estimates. This is visible in terms of the predicted target location. Since the motion model applied is a constant velocity model, the assumption is that the next observed target location should be in a straight line with respect to the previous observations. Thus all of the predicted positions lie outside of the targets circular path. However the predictions are still close enough to reality that the target can be successfully tracked. The bottom panel shows the extreme case in which the motion prior is too strong. In this case the motion model is considered highly reliable and observations extremely noisy. Observations can still cause changes in the estimate of target

position and velocity, but the motion model is more ‘stiff’, requiring more observational data before changing its belief in the target state. Under these conditions the predicted target position lies even further outside the circle, along the path of constant velocity. When the predicted position moves too far from the actual observed data, it is no longer matched during data association, and the estimated target continues under existing motion with no further influence from observations. The old target becomes stale and is deleted, and a new target is created that matches the existing observations.

The above analysis leaves the question of system accuracy and robustness with somewhat unsatisfying answers. Changes in system parameters have an affect on system performance in a variety of subtle ways. Nevertheless it is possible to make a number of assertions about the qualitative performance of the system. Multiple targets can be tracked robustly. Target paths can be initiated and deleted as they move in and out of the working volume. The working volume is scalable, such that cameras can be simply added in order to cover more space. Targets can be tracked in this working volume with at least 10 mm accuracy, which is sufficient to guide proper sub-region selection.

4 Sub-region selection

After locating target features at the large scale, some way of providing more detailed information about these features to a small scale system is needed. This process is called sub-region selection. In the case of the system presented here, sub-region selection occurs by directing a set of pan-tilt cameras with telephoto lenses to look at the target. These cameras provide high resolution imagery to the small scale motion recover system The chief concern of the sub-region selection sub-system is accurately determining camera parameters that do in fact direct these camera to view the target.

Directing the pan-tilt cameras to point in the correct direction is complicated by several issues. Complex actuated components require an appropriate geometric model and method for calibration. In addition, since these cameras are used in a real time system, some method must be employed to counteract latency.

This section describes the details of the methods employed in order to insure that cameras are oriented correctly and appropriate high resolution imagery is available to the small scale recovery system.

4.1 Physical setup

The system constructed in our laboratory makes use of four pan-tilt cameras. Each of these cameras is connected to an SGI O2 which digitizes, compresses, and saves the video frames for later offline small scale processing. We use Sony EVI-D30 integrated pan-tilt cameras which can be controlled with a RS-232 serial protocol. The sub-region selection process determines the correct camera parameters and communicates via ethernet with a pair of wintel PCs that manage serial communication with the cameras. Figure 45 shows a schematic of this setup.

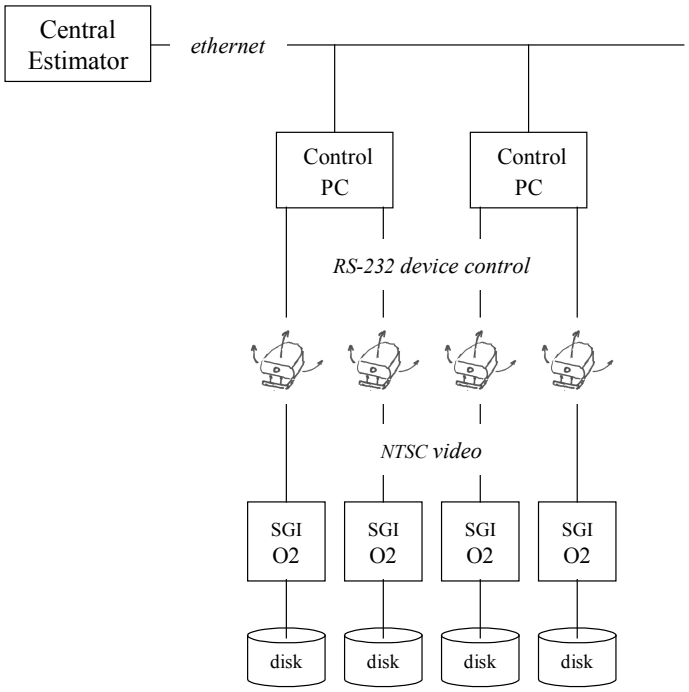


Figure 45 Schematic of the control setup for the pan-tilt cameras that perform sub-region selection.

Figure 46 shows some of these cameras mounted in our laboratory. A top down view of the pan-tilt camera configuration is shown in Figure 47, along with the statically placed cameras used in the large scale system. These cameras are mounted such that they can be directed to view any sub-volume within the total working volume of the large scale tracking system. The four pan-tilt cameras are all mounted on one side of the working volume. In order to allow stereo triangulation, these cameras need to be mounted such that they have view points with a small baseline. In a production system, more pan-tilt cameras would be used to cover the scene from all possible angles.

Figure 46 Pan-tilt cameras mounted in our laboratory motion recovery area.

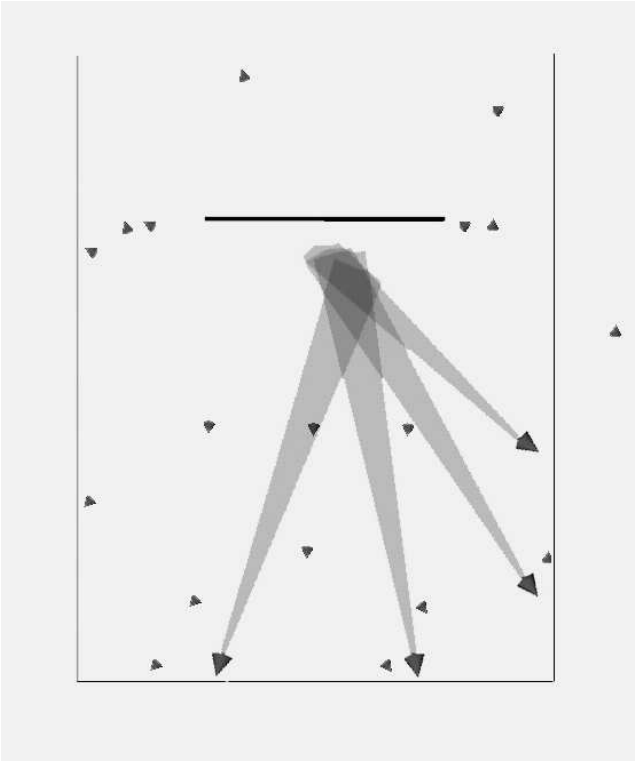


Figure 47 Top down view of pan-tilt camera placement. Cameras can be directed such that their view frustrums intersect at the desired target feature location.

4.2 Calibration

The pan-tilt controller module must direct each camera to look at the desired target. In order to do this, the geometry and parameters of the camera must be determined. The hardware control of our cameras allows two free parameters, pan and tilt. The rest of the camera geometry is fixed and can be determined in advance. As with static cameras, a geometric model that relates camera image plane observations to actual target positions is needed. In the case of pan-tilt cameras this model includes some description of the mechanical motion as the two parameters change.

Most existing systems that make use of a pan-tilt camera assume a rather simplistic geometric model of motion, in which axes are orthogonal and aligned with the camera imaging optics [9, 12, 28, 43, 113]. This simplistic model is insufficient to account for the motion of the cameras used in this system. Of course more complex models have been proposed by the robotics community, e.g. Shih gives the details of calibrating a stereo head with multiple actuated degrees of freedom, however orthogonally aligned axes are still assumed [94]. The following sections introduce the simplistic camera geometric model used by most researchers and then describe the more complex model used in this system.

Simple pan-tilt camera model

Most previous researchers using pan-tilt cameras have used a fairly simple camera model. Since the camera pans and tilts, the camera motion is modeled as two rotations around the origin, followed by a perspective camera transformation, as shown in Figure 48. This transformation can be written as a sequence of matrix operations

$$\begin{bmatrix} I_x \\ I_y \\ I_w \end{bmatrix} = \mathbf{C}\,\mathbf{R_y}\,\mathbf{R_x} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad (1)$$

Where $[x\ y\ z]^T$ is a point in world coordinates. A rotation around the x-axis, $\mathbf{R_x}$, and then around the y-axis, $\mathbf{R_y}$, correspond to tilt and pan respectively. Finally a perspective camera

transform, **C**, results in the image plane coordinates at which the point is observed, (I_x/I_w , I_y/I_w).

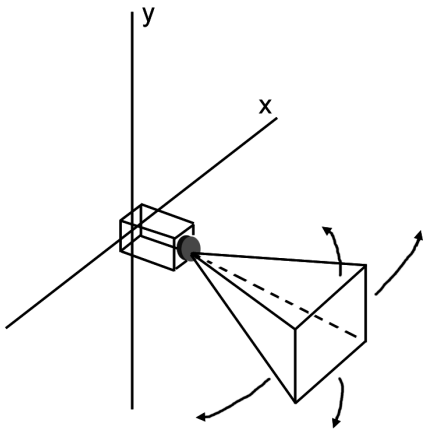


Figure 48 Simple pan-tilt camera motion model. Pan and tilt are modeled as axis aligned rotations around the origin.

For an ideal pan-tilt camera this model is sufficient. However, the assumption that the pan and tilt axes of rotation are orthogonal, aligned with the image plane, and through the camera's nodal point are frequently violated. The usual solution to this problem is careful engineering, so that the assumptions are as nearly true as possible [102]. Since our system uses commercially integrated pan-tilt cameras that significantly violate the assumptions and can not be easily modified, a better model is needed.

Improved pan-tilt camera model

When a pan-tilt camera is assembled, it is difficult to ensure that the axes of rotation intersect the optical center of the camera imaging system. In addition, the axes are unlikely to be exactly aligned. In order to model the actual motion of the camera, new parameters can be introduced into the camera model. As shown in Figure 49 rather than being coordinate frame aligned, the pan and tilt degrees of freedom can be modeled as arbitrary axes in space. The imaging plane and optics are modeled as a rigid element that rotates around each of these

axes. This model more closely approximates the actual camera geometry, and can be written as

$$\begin{bmatrix} I_x \\ I_y \\ I_w \end{bmatrix} = \mathbf{C}\ \mathbf{T_{pan}}\ \mathbf{R_{pan}}\ \mathbf{T_{pan}^{-1}}\ \mathbf{T_{tilt}}\ \mathbf{R_{tilt}}\ \mathbf{T_{tilt}^{-1}} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad (2)$$

where $\mathbf{R_{tilt}}$ is a 3x3 rotation matrix which rotates *tilt* angles around a vector in the direction of the tilt-axis. $\mathbf{R_{pan}}$ is analogous. These axes do not necessarily pass through the origin, and $\mathbf{T_{pan}}$ and $\mathbf{T_{tilt}}$ represent translation vectors from the origin to each axis. Thus the projection of a point in space onto the image plane can be found as a function of current camera *pan* and *tilt* parameters.

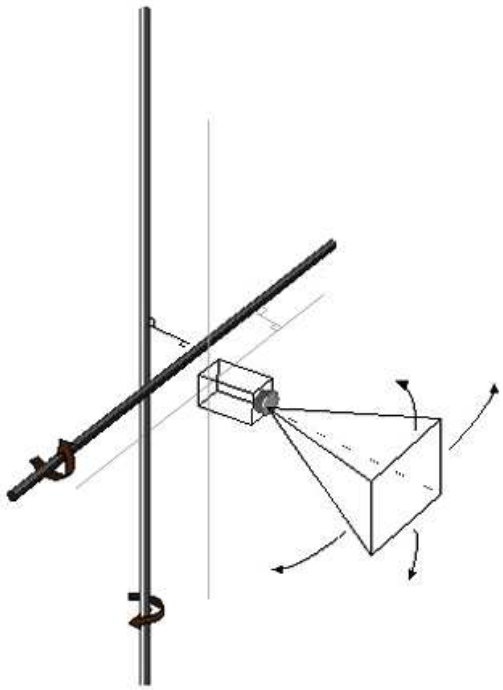


Figure 49 Improved model of camera motion. Pan and tilt motions are modeled as rotations around arbitrary axes in space.

The above models assume that the angular rotation around each axis is known. Of course the motor control of each axis occurs in terms of some unknown units, and a calibrated mapping to angular units is required. In the case of the cameras used in this system, rotations are

specified in degrees, and empirical tests suggest that angular rotation was in fact well calibrated by the manufacturer.

Calibrating the model

The above model specifies the generic relations between geometric components. Each camera will vary within this general model, and determining the specific parameters of the model is typically referred to as calibration. In the case of this model, the parameters to be determined are the two axes of rotation in addition to the standard set of camera intrinsic and extrinsic parameters.

One reason that the simple camera model described above is often used despite its inadequacies is the ease of calibration. Since the axes of rotation are given relative to the camera image plane, it suffices to calibrate the camera in a single configuration and then apply the model when a new set of pan-tilt rotations is required. In the more general case of arbitrary axes, a great deal of additional data needs to be collected in order to robustly determine all parameters.

Traditional camera calibration proceeds by arranging a set of known 3D features in the camera's view frustum. These are typically attached to some calibration target. The projection of each 3D feature is then observed on the camera image plane. These observations are 2D data points. Camera parameters can be determined by solving

$$\arg\min_{\Phi} \{ \mathbb{C}(\Phi, \mathbf{X_{3D}}) - \mathbf{X_{2D}} \} \qquad (3)$$

where $\mathbb{C}$ is the camera model that defines the projection of points onto the image plane, Φ is the set of camera parameters to be determined, $\mathbf{X_{3D}}$ is the vector of 3D feature locations, and $\mathbf{X_{2D}}$ is the vector of corresponding image plane observations. Since Φ often includes radial lens distortion terms in addition to extrinsic geometric pose, minimizing this equation is usually framed as a non-linear search problem.

In order to obtain a good estimate of the parameters, Φ, it is necessary that the spatial features, $\mathbf{X_{3D}}$, cover the working volume, and that the observations, $\mathbf{X_{2D}}$, cover the image

plane. In the event that coverage is not comprehensive, parameters that fit the observed space will be obtained. If a target later moves outside of the observed calibration range, the model will extrapolate into the new area. Since data extrapolation is known to be unreliable relative to interpolation, it is beneficial to obtain maximum coverage [86].

In the case of the pan-tilt camera model considered here, two axes of rotation are added to the set of parameters, Φ. Since the current pan and tilt parameter settings must also be included, the equation now becomes

$$\arg\min_{\Phi} \{ \mathcal{C}(\Phi, \boldsymbol{pan}, \boldsymbol{tilt}, \mathbf{X_{3D}}) - \mathbf{X_{2D}} \} \quad (4)$$

where ***pan*** and ***tilt*** are vectors specifying the setting corresponding to each feature-observation pair. As before, good calibration requires adequate coverage of the space. In this case, coverage of the range of pan and tilt parameters is implied, as well as coverage of space.

Thus the procedure for calibrating this camera model is an iteration of the procedure used for a static camera. The pan-tilt parameters are set to some value, .e.g. (0º, 0º). A set of feature-observation pairs is obtained. Then the pan-tilt parameter values are changed to some new value, .e.g. (100º, 50º) and an additional set of feature-observation pairs is obtained. The procedure is repeated until the range of camera motion has been covered. All vectors are concatenated and the equation is minimized.

Features and observations must be placed to cover the volume observed by the pan-tilt camera. Since the camera can turn, this is potentially a very wide area. As with the previous case of wide area calibration, it is cumbersome to build a sufficiently large calibration object. If a wide area tracking system is already in place, as it is in this case, the tracking system itself can instead be used to build a virtual calibration object by tracking the motion of a single feature over time.

In our system, calibration was obtained by first calibrating the camera intrinsic properties, lens distortion, focal length, etc., since these do not change as the camera is rotated about its axes. Then the extrinsic camera placement and location of rotation axes is calibrated by observing features in each of several camera parameter configurations. Observations were

gathered in 10º increments in the range (-50º - 50º pan, 0º tilt) and again at 5º increments in the range (0º pan, -20º - 15º tilt). All observations were then stacked into vectors and equation (4) was minimized using a non-linear equation solver. This method was found to adequately insure that cameras can be directed to observe a feature at any point in the working volume.

4.3 Directing pan-tilt cameras

Given a target in the working volume, it is important to direct each pan-tilt camera such that it observes the target. Thus for a target position in space we must find the correct (*pan*, *tilt*) parameter settings that orient the optical axis of the camera such that it intersects this point. Equation (2) gives the projection of a point onto the camera image plane for a known pan and tilt setting. In this case it is desired that the target point projected onto the optical center of the image plane. Inverting these equations unfortunately yields multiple solutions for the (*pan*, *tilt*) parameters.

If inconsistent parameter choices are used from one time instant to the next, the pan-tilt cameras will jitter between the possible solutions. Rather than resolve this ambiguity analytically, this system determines camera orientation by means of a gradient decent search. This insures that parameters are chosen in a nearby local minima, thus smoothing possible jitter. Starting from the current camera configuration, the projection of the target onto the image plane is calculated. The difference between the projected position and the center of the image plane is calculated. The projected error after the camera undergoes a small pan or tilt change in each direction is also calculated. If the error is reduced for any of the hypothesized camera settings, the new parameter settings are accepted and the search is iterated until the error is below a pixel. Usually fewer than ten iterations are required and the process takes negligible CPU time. After optimal (*pan*, *tilt*) parameters have been determined, the camera itself is directed to reorient into the desired configuration.

If the pan-tilt cameras were issued orientation commands at the same rate that target state estimation proceeds, their command queues would quickly overflow. The actuated cameras simply can not respond quickly enough. It is necessary to direct the cameras at some lower rate. In this system the orientation of pan-tilt cameras are renewed at between 15Hz and 30Hz.

It is desirable for camera motion to proceed as quickly and smoothly as possible. Unfortunately brief fast motor motions cause the camera to vibrate and distant slow motion makes the camera insufficiently responsive. One can of course set the desired motor velocity in terms of angular change, but noise in target positioning can create difficulties in estimating the actual angular change. In addition the step motors which rotate these cameras have discretized positioning and one wishes to avoid camera jitter between adjacent settings. For these reasons a simple damping filter is used to regularize the estimated target position, beyond the filtering already inherent in Kalman filtering. This smooth motion is used to direct the cameras, using an adaptive velocity proportional to required change.

4.4 Latency

The pan-tilt cameras used for sub-regions selection do not respond instantaneously. This latency in response can severely compromise the ability of the system to generate useful high resolution images for use by the small scale motion recovery sub-system. A number of reasons contribute to observed latency.

The Sony EVI-D30 cameras used in this system are consumer grade cameras intended for teleconferencing applications. They have not been designed for rapid response. Between the time that a command is received by the camera and the motors reorient the camera into the desired configuration, there is significant delay. Of course dedicated pan-tilt heads with better accuracy and response rate can be obtained, but the fundamental issue of latency can not be completely eliminated. Although camera physical mechanics is one of the primary contributors to latency, other parts of the tracking pipeline contribute. The large scale motion recovery system takes some amount of time to determine the location of targets. After the target has been located the position is transmitted to a separate machine, which is dedicated to camera control. This machine in turn controls the camera via a low bit rate serial connection. Each of these stages introduces latency to varying degrees.

If latency is too long, targets moving at high velocity will move to a new location before the camera can be oriented correctly. This is observable as the camera apparently following behind the target with the target always out of the video frame. End-to-end latency through the entire pipeline of this system is approximately 300ms. At least 250ms of this is due to the

camera mechanism. With this amount of latency, targets moving at normal human walking speed will fall out of the pan-tilt video frame.

Latency in a data stream is an instance of inadequate data. This difficulty can be compensated for by using a model of motion to predicting the future position of targets. Given a correct prediction of the future, cameras can be directed such that their orientation will match the current location of the target, rather than the target's previous location. In this case, an appropriate model allows a correction to be made to the data stream.

If target motion is modeled as position and velocity, predictive estimates will be possible. Conveniently, the large scale motion recovery framework already includes a temporal model that estimates both position and velocity. These estimates can be used directly to predict future target positions. This prediction is regularly performed as part of the update procedure for each Kalman filter, and is used similarly here. The predicted position in space of a target at a future time is just $\mathbf{P'} = \mathbf{P} + \Delta t * \mathbf{V}$ where Δt is the time offset into the future, **P** is position, and **V** is velocity. In the case of this system Δt should be set to 300ms to match the camera motion latency.

Prediction makes a dramatic difference in the ability of the pan-tilt cameras to robustly keep a target centered in their field of view. Figure 50 shows still frames from a video sequence showing the effect. The first section of the video shows target motion from the viewpoint of a pan-tilt camera without using prediction. Note that the target moves out of the frame. The second section of the sequence shows target motion when prediction is enabled. The target is now kept approximately centered at all times. Although only empirical evidence that prediction is valuable is provided here, Azuma provides an analytic analysis of prediction that describes its expected value [8].

The sequence in Figure 50 also illustrates one of the primary motivations for a layeredmotion recovery system. In parts of this sequence the target moves out of the video frame. If only the pan-tilt camera was in use, then the target would be lost, with no apparent means to reorient the camera. However, the large scale recovery subsystem robustly maintains an estimate of target position. Even when the target falls out of the frame, and is thus lost to the small scale recovery system, it is possible to reorient the pan-tilt cameras such that the target again becomes visible at a later time.

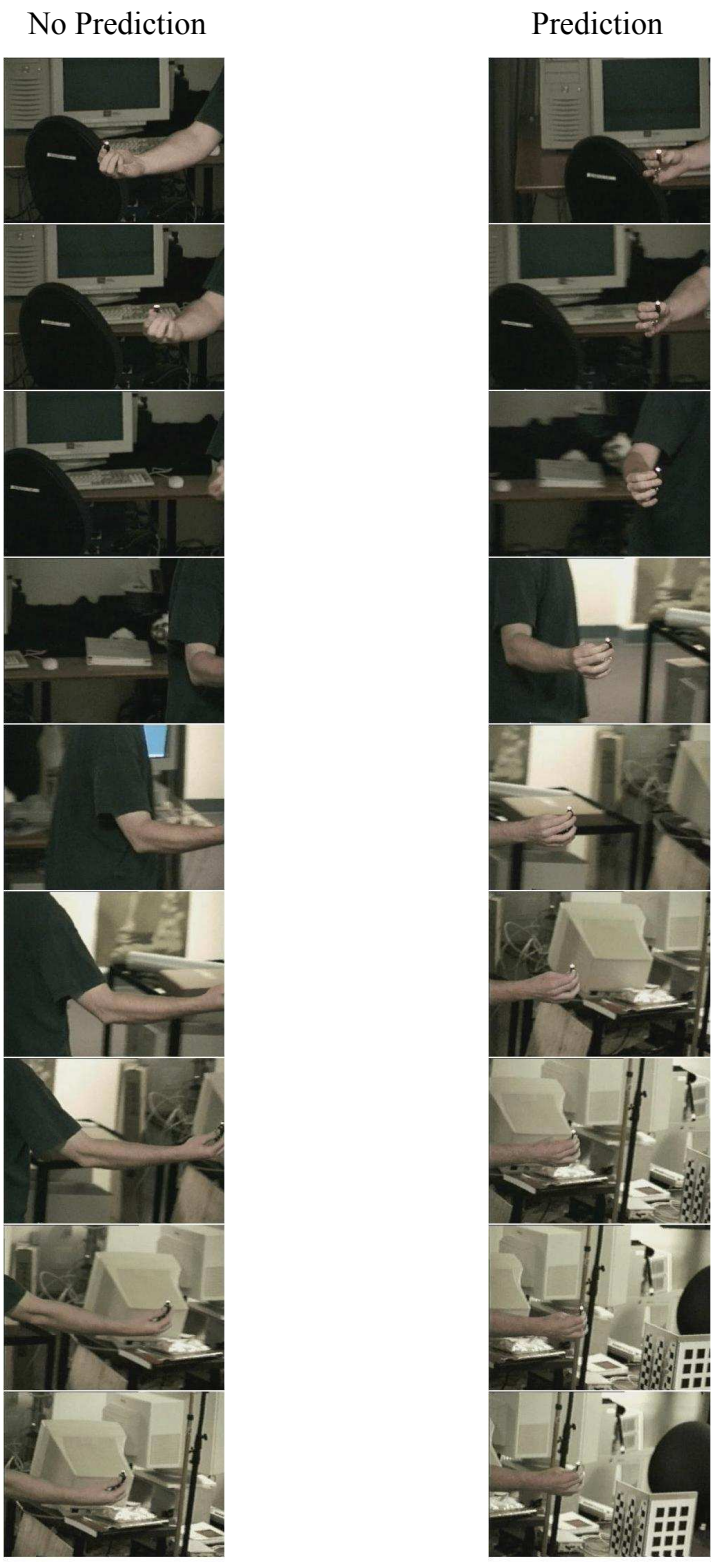


Figure 50 *Video figure* This sequence shows the video stream from a pan-tilt camera as it is directed to follow a moving target. In the first column, the camera is directed without prediction. Note that the target often falls out of the video frame. In the second column prediction is used. In this case the target stays approximately centered in the video frame at all times.

4.5 Evaluation

The primary task of the sub-region selection system is to accurately direct pan-tilt cameras such that they provide high resolution imagery of the target. These cameras must be calibrated and an improved model of camera geometry was introduced. The quality with which this model fits observed data can of course be evaluated and an analysis is provided below. Camera motor latency degrades actual performance relative to the ideal calibrated performance. An additional measure of quality is the system's ability to maintain correct orientation while a target is moving with high acceleration.

Camera calibration can be measured in terms of image plane projection error. This is essentially a measure of the camera model's ability to explain data. If the model is of high quality, then the projection of target locations onto the image plane will fall close to actual observations. In order to evaluate the pan-tilt camera model proposed in this work, a set of 361 data points was collected consisting of world space target location and observed image plane coordinates. This data was collected over the full range of the camera (*pan, tilt*) parameter space. In order to isolate calibration from potential bias due to the tracking system, a Faro Arm coordinate measuring device was used to determine world space coordinates.

The collected data was used for camera calibration under three conditions. In all cases the intrinsic camera parameters including focal length and lens distortion were calculated as a preprocess using the method of Heikkila [57]. In the first condition, only the data from the (pan = 0, tilt = 0) parameter setting was used. Under this condition the external pose was calculated using standard camera calibration, and the rotational axes were assumed to be orthogonal to and aligned with the optical imaging plane. Since data from only a limited portion of the configuration space was used, it is expected that errors in extrapolated regions of the space will be relatively high. In the second condition, data from all pan-tilt parameter settings was aggregated and used to find the best fit external pose of the camera, still under the assumption that rotations are around orthogonal aligned axes. This condition is similar to the simple model of pan-tilt motion used in nearly all other research. Finally, the model proposed in this work was investigated. All data was used to calibrate both the external pose of the camera, and a pair of arbitrary rotational axes.

Figure 51 shows the results of the experiment above. The image plane projection error of each data point is shown in a scatter plot, for each of the calibration conditions. It is clear that

assuming a single static camera during calibration produces high error, as expected. The median error is 46.3 pixels. The middle figure shows calibration using the simple pan-tilt model of camera motion. As can be seen, calibrating using the full range of camera pan-tilt parameters greatly improves the quality of model fit. The median projection error is now only 19.1 pixels. On the right is a plot of the error when camera pose and axes of rotation are allowed to vary independently. There is a noticeable improvement in this model's ability to explain the data. Error has been reduced to a median 14.2 pixels.

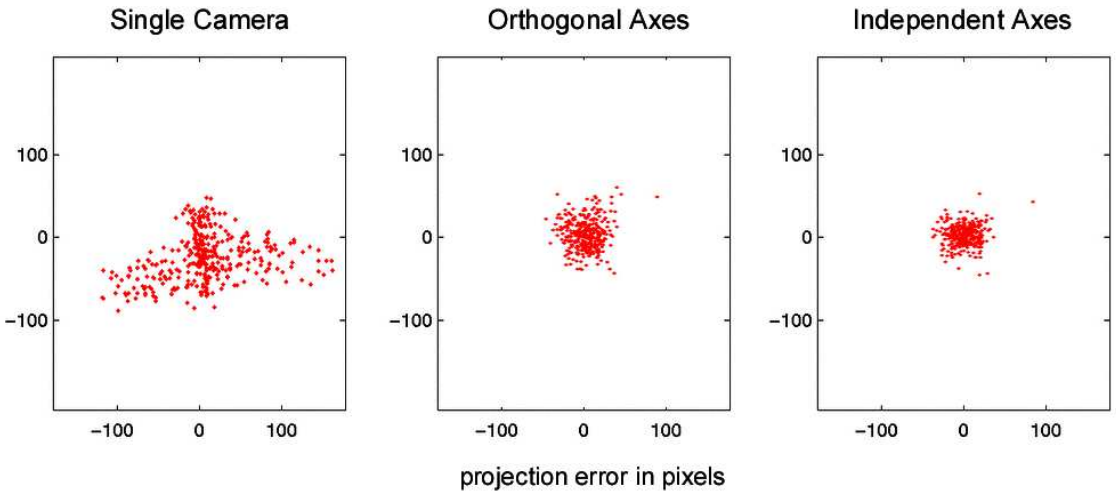


Figure 51 The image plane projection error of measured data diminishes as the complexity of camera calibration increases. The pan-tilt calibration model proposed in this work has the smallest error. (Left) Error when calibrating using only observations and world space coordinates collected while the camera was in its home position. (Middle) Projection error using all data in conjunction with the traditional simple model of pan-tilt calibration. (Right) The error when using the model proposed in this work.

Since it is somewhere difficult to understand the distribution of data points in a scatter plot, projection error has been expressed as a histogram in Figure 52. It is clear that progressively more complex camera models do in fact improve the quality of fit. The median projection error in terms of pixels, under all of these conditions, is somewhat higher than should be expected. This can be explained by measurement noise. Both the world space coordinate measurements, and image plane feature observations were afflicted with higher than normal amounts of noise. If the l_2-norm used to fit this data was replaced with a robust statistical estimator improvement would be expected. However, the quality of fit was sufficient for this application.

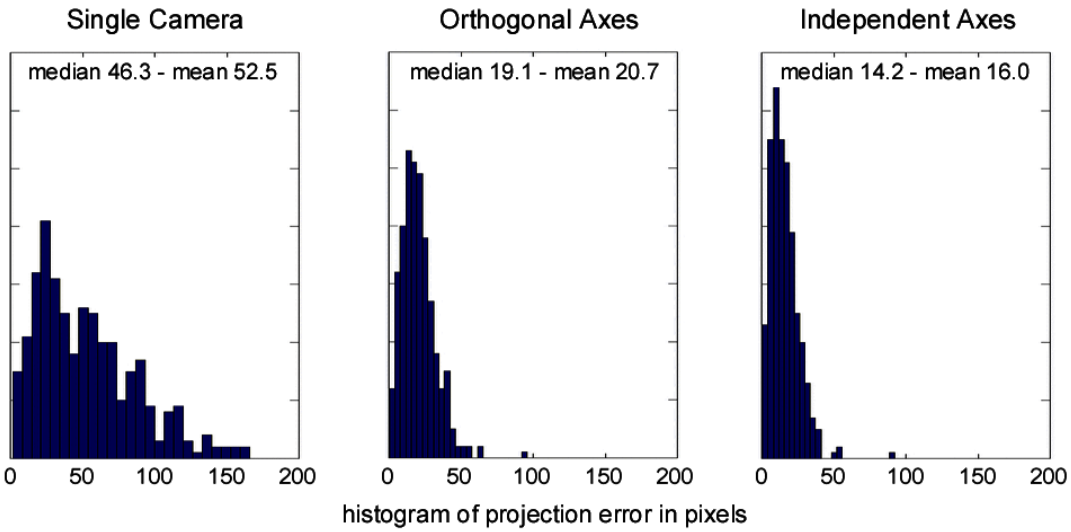


Figure 52 Histograms of projection error. The use of independent rotation axes, rather than axes orthogonally aligned with the image plane improves the fit between modeled projection and observation. (Left) Calibration of a single camera position, extrapolated to pan-tilt motion. (Middle) Calibration of camera while requiring aligned orthogonal axes. (Right) Calibration allowing camera and axes to move independently.

Projection error gives some indication of calibration accuracy when cameras are in a stationary pose. However in this system the cameras are continuously in motion. The predictive model used to offset latency makes an assumption of zero mean acceleration, and will be inaccurate whenever this assumption is violated. In order to understand the behavior of the system under real world conditions, the pan-tilt cameras were directed to stay oriented towards a target undergoing motion. Ideally, the target will stay centered in the image frame of each pan-tilt camera. In practice the target will deviate from this position in a manner correlated with the targets deviation from the expected model. Thus targets moving with constant velocity will stay well centered, while targets with high acceleration will deviate from the image center. In order to understand the worst case behavior of the system, a target undergoing relatively large acceleration was chosen. A target was moved rapidly from side to side as shown in VideoFigure 50. The deviation from image center was measured on a 900 frame sequence lasting 30 seconds. Figure 53 shows a plot of this deviation over time. The peaks in deviation correspond to the high acceleration experienced by the target as it reversed direction. Average deviation was 104 pixels, 1/6 of the image width, from the image center. Although this deviation is relatively high, it is sufficient for this application in which

maintaining the target within frame is more important than motion within the frame. Better performance would be expected from higher quality pan-tilt mechanisms.

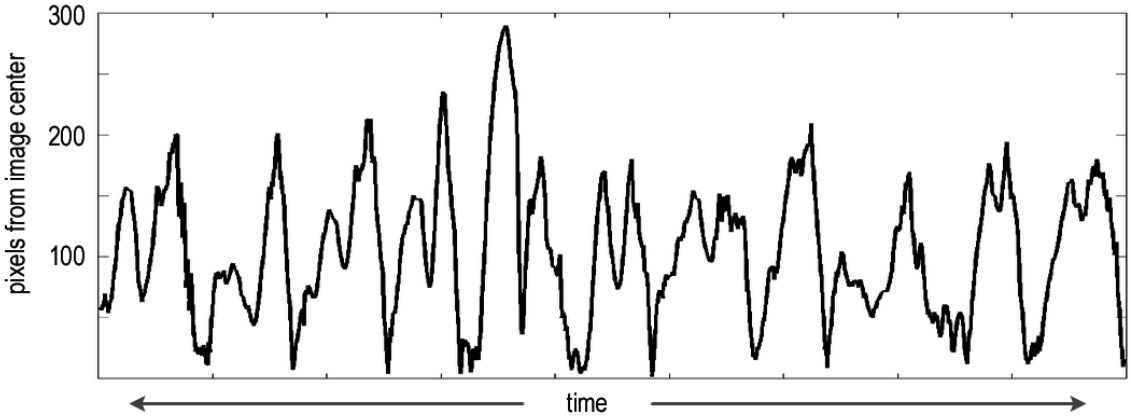


Figure 53 Deviation of a target from the center of the pan-tilt image frame as a function of time. Peaks in target deviation correspond to periods of high target acceleration.

5 Small scale recovery

Features and movements too small to be recovered by the large scale system can be reliably recovered with the small scale recovery subsystem. This system makes use of the guided pan-tilt cameras introduced in the last section. By using a narrow field of view, high resolution images can be used to track features. For example, Figure 54 shows a tracked target from each of the four viewpoints. The same sub-volume is visible in each of the four views. Conceptually this subsystem performs the same tasks as the large scale system. Features are identified, tracked, and integrated, allowing 3D motion to be recovered. However, the interface with surrounding system components is different, thus the details have changed.

The change in system details between large and small scale motion recovery brings a new set of challenges. Due to their dense packing, small scale features must be less active in modifying the object to be tracked. In addition, these features must be detected in video captured on a moving camera. The combination of these attributes makes errors in feature integration more noticeable, and missing markers due to occlusion more likely. The remainder of this section discusses the details of the small scale recovery system, as well as suggesting a model for facial acquisition that can be used to regularize the somewhat noisy data recovered.

Figure 54 The view from each of the four pan-tilt cameras. The cameras all look at the same point, but provide images from different angles, allowing triangulation of features.

5.1 Feature detection

Features in the small scale system are conceptually similar to large scale features but differ in the details. Because it is not practical to mount many LED lights close together in a small space, painted markers are used. Figure 55 shows a close up of a face with hand painted marks. The red trail is marks the path of one feature over time. On the left is an enlarged view of the region surrounding this feature. Since it is not necessary to track the small scale features in real time, video from the pan-tilt cameras is recorded to disk for offline processing.

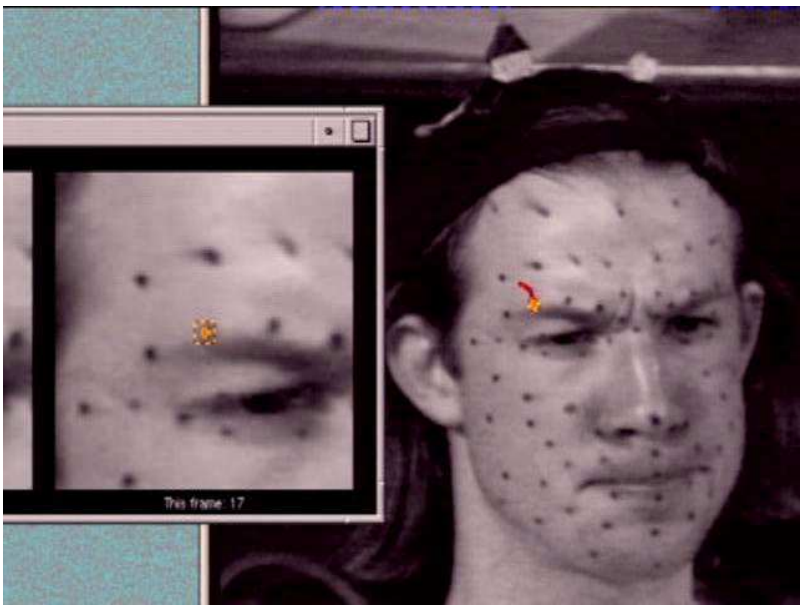


Figure 55 *Video figure* The small scale recovery system uses painted marks as features for motion recovery. A close-up of features on the face is shown here.

Features are tracked using Lucas-Kanade style tracking [68, 101]. A user manually identifies features in the first frame of video from each camera that observed the target. These features are then tracked throughout the rest of the sequence based on a windowed image similarity function. This method of tracking is well known to suffer from failure and drift on long sequences. In this system, tracking was empirically found to be stable for between 50-200 frames, which is consistent with existing literature. In order to restart tracking after these intervals, a user interface is provided that makes it relatively simple to monitor and guide the

tracking process, Figure 56. A promising area of future work would be to use other features, such as retro-reflective markers which can be reliably tracked across many frames.

The pan-tilt cameras used to capture video streams are in motion while video is being captured. This presents several difficulties relative to standard tracking in which high quality video, with relatively small marker motion is assumed. Because of the camera motion, features may be subject to relatively large motion in the frame. It is not unusual for a feature to move by as much as 50 pixels from one frame to the next. In addition, when the camera undergoes fast motion, some frames may have excessive motion blur, such that no features are visible.

When features undergo frame motion larger than the size of the search window, image gradient style tracking such as this will be unable to locate the feature. In order to minimize feature motion during tracking, image stabilization is applied [55, 69]. The relative translation between frames is calculated such that the dominant object position stays fixed in the frame. After stabilization, the only remaining marker motion is due to facial deformation. This greatly increases the likelihood that features will be successfully tracked.

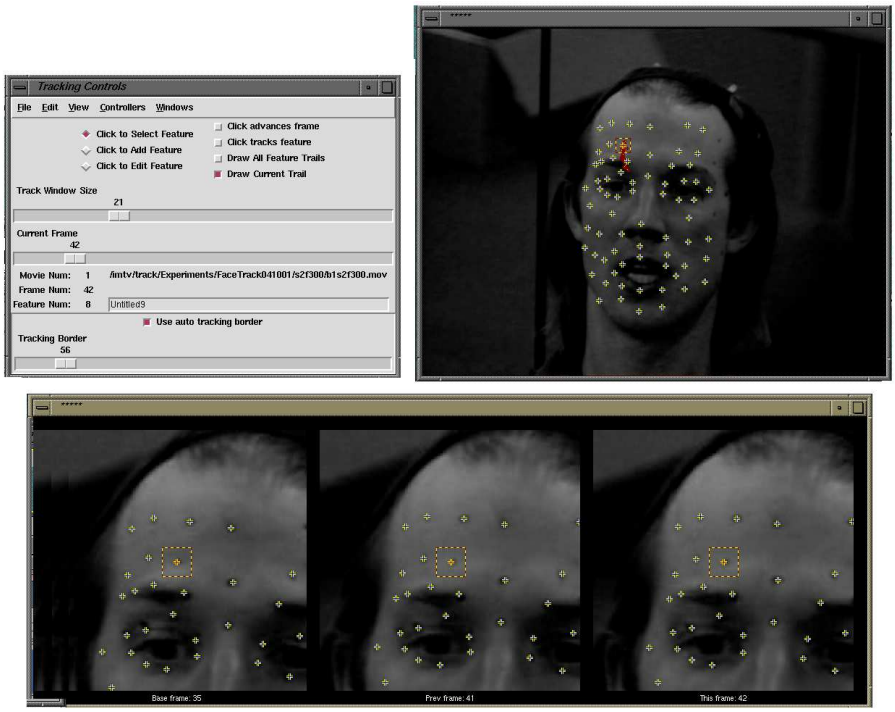


Figure 56 A screen shot of a tracking session. Automated tracking sometimes fails, so a user interface is provided to guide the tracking process in these cases. Above are the controls and an overview of the current tracking state. Below is an enlarged view so that the tracking progress of a particular feature can be monitored.

When the face is quickly deforming, features may still have too great a motion to be tracked. For example, this happens often with markers on the chin during rapid speech. In order to further increase the probability that a marker will be tracked a search for the feature is performed not just near the previous feature location, as in standard tracking, but also at an offset based on the predicted feature motion. The predicted feature location is calculated via a finite difference velocity model. The addition of predicted search regions further improves the reliability of tracking.

Occasionally the pan-tilt cameras are required to move at a high velocity in order to keep the tracked object in the frame. When this happens the video frames may have excessive motion blur, from which no features can be recovered. Standard tracking implementations will stop whenever a feature is lost. This is a reasonable behavior when the video stream is high quality and the feature loss is likely due to a change in image appearance. Since this change is likely to persist, the feature will not be recovered. In the case of motion blur, one of the following video frames is likely to have improved clarity. Therefore when a feature is lost, tracking is attempted on the following few frames. In many cases the feature can be recovered, and this method saves many instances of reinitializing feature tracks.

After tracking features in all camera views, these observations can be integrated in order to recover 3D pose.

5.2 Integrating observations

Views from several different pan-tilt cameras can be integrated to recover a targets 3D pose. Each feature's 3D location can be triangulated from two viewpoints as shown in Figure 57. In theory, the pan-tilt cameras are moving under known calibrated motion. Therefore camera parameters are known and any two cameras can be chosen to triangulate each feature. In practice, the particular pan-tilt cameras in use have precision of only 0.1 degree. Furthermore, although the expected camera pose after successfully completing a camera directive is known, the velocity and motion model of the camera is unknown, so that the pose is unknown while the camera is actually undergoing motion. Since the camera is constantly undergoing motion, the precise pose of the camera is never known. Therefore, only an approximate triangulated 3D pose can be determined.

When using a single pair of cameras to triangulate all features on an object, the errors introduced by the approximation above are correlated and therefore not of practical consequence. The resulting overall pose is just as would be expected. However, when different pairs of cameras are used to triangulate different features, the errors are not correlated. This results in a situation in which features from one pair of cameras have a noticeable positional bias relative to features reconstructed from a second pair of cameras. This can result in a very noisy target pose reconstruction.

The practical consequence of this is that, given the low quality cameras in use, all features should be reconstructed using a single pair. Of course the chosen pair can and should vary across time, since the best view of the target will change.

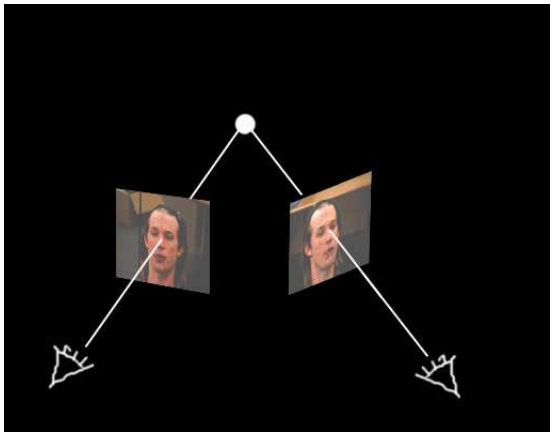


Figure 57 Small scale features can be reconstructed by triangulating observations from two viewpoints.

The positional bias introduced by poor knowledge of camera pose is a form of noise in the data flow, first in terms of the assumed camera parameters, and then later in terms of feature position. It would of course be possible to instantiate some model that would regulate this noise near the source. In this system, I have instead chosen to explicitly regularize the noise later in the pipeline, using a model of facial deformation.

Figure 58 shows an example of a face recovered with the small scale recovery system. The face geometry is shown from several viewpoints. Sixty-nine feature points were marked and tracked on the face. Observed features were integrated as described to recover the 3D feature

locations. The triangulated mesh connectivity was created by hand as an aid to visualization, since visual interpretation of point features is difficult.

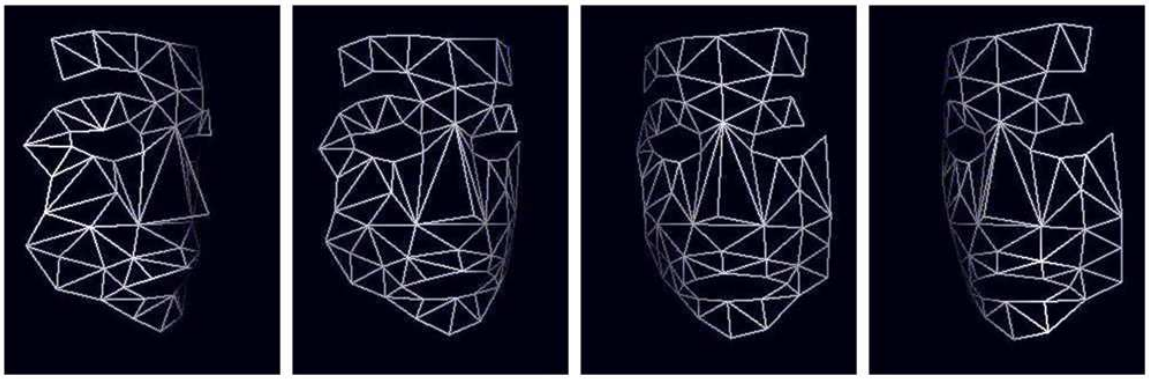


Figure 58 *Video figure* Recovered 3D face geometry shown from several viewpoints.

5.3 Occluded features

Features are often occluded such that they can not be recovered by the small scale recovery system. For example, Figure 59 shows two views of a face, and the recovered features available from these two views. Notice that since half the face is occluded in one of the images, no features can be reconstructed.

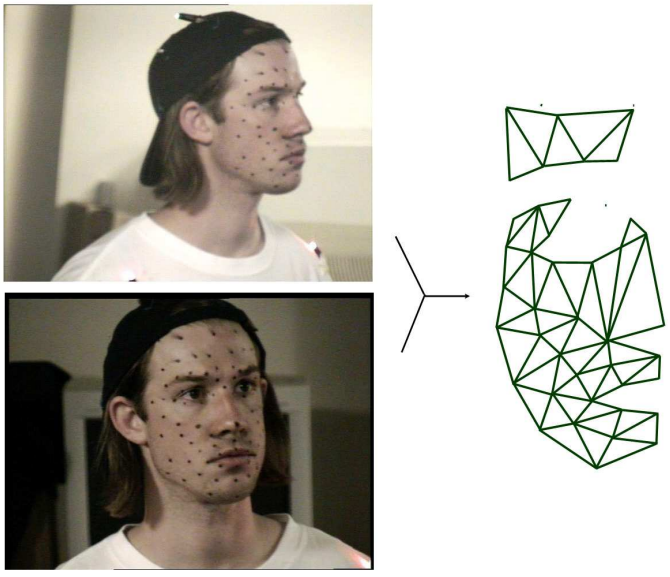


Figure 59 Reconstructed face. Since half of the features are missing from one of the viewpoints, these features can not be reconstructed directly.

Existing facial motion recovery systems insure that all features are visible by restricting head motion with a physical restraint and carefully arranging cameras so that all features are visible. In the system described here the small scale target is free to move in the global working volume, and insuring that all features are visible at all times is impossible. Additional cameras would increase the probability of features being observed, however the possibility of lost features will always exist. Unfortunately, the applications that make use of facial motion capture assume that all feature points are present all the time, so some method must be employed to recover the missing data.

Features can also be occluded during large scale motion capture, so one wonders if similar methods are appropriate. In the large scale system presented here, as well as most commercial motion capture systems, features are relatively sparse and the motion of individual features is largely independent. Sparse independent features can be reconstructed by any pair of cameras that happen to see the feature. Under these conditions the probability of occlusion is relatively low. However, since there is little correlation between the motion of neighboring features, a human artist is typically required to create motion for any marker that is missing. In contrast, facial features of the kind shown in Figure 55 are dense and move in a strongly dependant fashion. As discussed in the previous section, dense correlated features should be reconstructed from a single pair of cameras, in order to minimize visible artifacts. Since only two cameras are used for reconstruction, the probability of occlusion is relatively high. In addition, since a dense feature set of features is present, it is prohibitively expensive for a human to create all of the missing marker data by hand. Fortunately, the strongly correlated motion of these markers can be used to automatically create motion for missing markers. This motion will be consistent with the existing feature motion. One method of automatically creating motion for missing markers will be described in the next section.

5.4 Face model

Features on the face that can not be directly reconstructed due to occlusion can be automatically derived using a face model. Since dense features are present, and these features are highly correlated, existing features can be used to fit the parameter space of a model. The model can then be used to predict the location of missing features. A diagram of this process is shown in Figure 60.

Using a data model to recover missing features is a very explicit example of regularizing an inadequate data flow. The data stream has missing and/or noisy information, i.e. occluded features. Since the interface to the next component in the system requires complete information, a model is used to constrain the data range to an allowable sub-space. Once existing partial data has been fit to the subspace, missing features can be recovered directly.

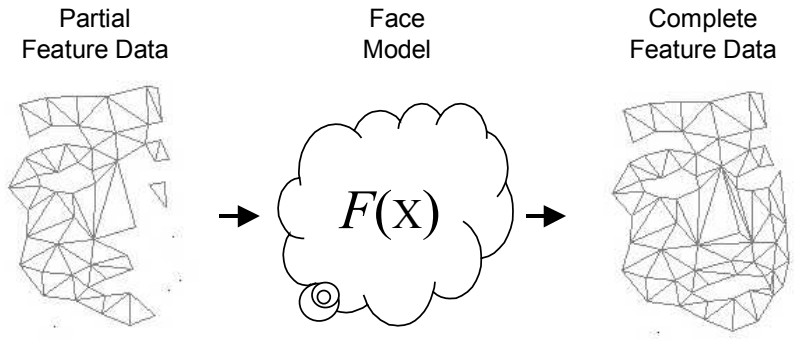


Figure 60 A partial set of features can be used to fit a facial model. This model can be used to reconstruct the location of missing features.

This general method for recovering occluded markers has been applied to human body motion [58]. An articulated model is typically employed which constrains the relation of markers according to the kinematic pose of the figure. Unfortunately the methods used in conjunction with articulated models are not directly applicable to facial data. Since it has traditionally been acceptable to insure that all markers are visible by strictly controlling head pose, other researchers have not investigated models which may be appropriate for recovering occluded features in facial data. However, models related to the one proposed here have been used for compression, image processing, tracking, and rendering of facial information [2, 16, 30, 48, 74, 82]. The remainder of this section gives the details of the particular model used with this system.

Model definition

An ideal model of facial deformation should allow for a full range of expressions, while restricting the space of poses to only facial configurations which are possible. Using a linear

combination of basis shapes as shown in Figure 61 provides a good approximation of these criteria. Additional bases can be added until the full range of expressions are represented. However since only linear combinations of known poses can be created, the space is restricted to a range near valid faces.

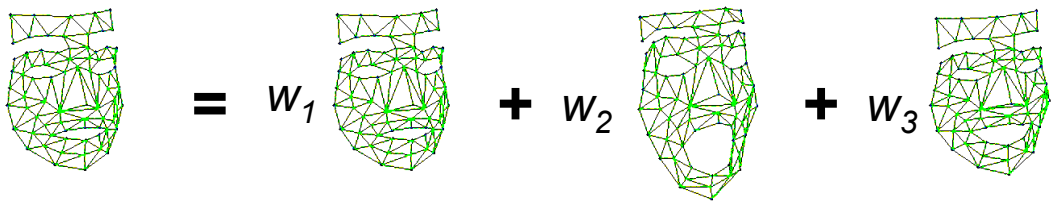


Figure 61 A linear combination of valid basis faces can be used to model the range of all possible feature motion.

This model can be written as

$$\mathbf{F} = \sum_i w_i \mathbf{B_i} \qquad (5)$$

Where each $\mathbf{B_i}$ is a basis vector containing the coordinates of all feature points $\mathbf{B_i}=[x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_n, y_n, z_n]^T$. These bases are combined linearly and w_i is the weight with which each basis should be applied. **F** is the face vector defined by the model. It is most intuitive to conceptualize these basis vectors as actual valid facial configurations. In this case, linear interpolation between bases corresponds to linear morphing between a set of known faces. Other basis sets are possible and will be discussed later.

Fitting data to the model

Given a set of basis faces as a model, and a newly triangulated set of features, these features can be fit to the model. Writing equation (5) as a matrix equation, we have

$$\mathbf{B}\,\mathbf{w} = \mathbf{F} \qquad (6)$$

where $\mathbf{B} = [\mathbf{B_1}, \mathbf{B_2}, \ldots, \mathbf{B_k}]$ and $\mathbf{w} = [w_1, w_2, ..., w_k]^T$. It is unlikely that the weight vector, **w**, can be fit such that equation (6) is satisfied exactly, however it can easily be recovered in a least squares sense.

In the case that some features are missing in the triangulated face vector, **F**, due to occlusion, the model can still be fit. The unknown features are merely removed from both **F** and **B**, and the equation is solved for **w** as before. The full set of facial features can be recovered by multiplying the weights against the set of full basis vectors, **B**.

While a basis weight vector that minimize error in a least square sense can be recovered in this manner, it is not necessarily the best choice. The error will be minimized only across the observed features, and we are interested also in the error of features that were not observed. The weights produced by least squares sometimes correspond to extreme extrapolations from the existing pose data. Although these data extrapolations fit the observed data well, they tend to magnify errors when reconstructing unobserved data. It is well known that interpolating models are better behaved than extrapolating models [86], so in this work the weight vector is constrained to interpolate known data. Weights in the range (0,1) will interpolate existing data, and the equation can be solved subject to this constraint [66]. Other methods for fitting weights to a basis set exist, and discovering the best method is an active area of research [29, 67].

Acquisition process

The facial model described above is easy to acquire. Since the model consists of a set of valid faces, we need only capture a set of faces and use them. These faces can be captured as a preprocess, using the same cameras, markers and actor as will be used in the actual motion capture session. Since the difficulty with occluded features arises primarily due to a moving subject and cameras, we can prevent missing features by holding the cameras in a given configuration and restricting the global motion of the subject. This configuration with fixed geometry is essentially the method used by existing facial motion capture systems. The subject them performs a sequence of facial deformations that fully covers the space of

motions expected in the final sequence. Since this preprocess of capturing expressions is performed under controlled conditions, all features are present and the data is of high quality. The set of frames recovered can be used directly as the basis set of faces.

While other researchers have investigated facial basis sets that are actor independent [16], these apply only to static facial pose. In order to robustly handle dynamic facial expressions, and any desired marker arrangement, this work acquires a new facial basis set for each actor and marker arrangement. Since this process is nearly identical to ordinary use of the motion recovery system, it is not a burden to acquire new face models.

Choosing a basis set

Using an entire video sequence of facial expressions as a basis set can result in thousands of bases. Unfortunately the simple method for determining basis weights used in this work is not properly constrained when too many basis vectors exist. In order to insure an over constrained set of equations, and therefore better numerical behavior, the size of the basis set must be reduced. We consider two methods of reducing the basis set: picking a subset of the existing bases that best spans the space, and using principal component analysis to find an optimal orthogonal set of bases to describe the space.

The primary modes of motion represented by the sequence can be obtained via principal component analysis (PCA) [62]. By projecting each of the facial shape vectors onto the principal components of the motion, a measure of extremeness can be obtained. Choosing the vector with the largest absolute projection onto each principal component results in a reduced basis set that still encodes the possible range of facial expression. Intuitively this removes all the neutral faces from the data set. It also removes duplicate faces with ‘mouth open’ or ‘eyebrows up’, leaving only one basis face for each mode of motion. This method of selecting a subset of basis vectors has empirically been found to work well for this application.

Rather than choosing bases from among valid face vectors in the space, it is possible to use principal component analysis to perform a basis transformation. The resulting vectors are orthogonal and ordered according to the magnitude of variation they contribute to the space. Given a limited number of bases, these vectors optimally represent the variation in the dataset. Furthermore it is easy to threshold the number of bases used in order to insure a

maximum of some epsilon error. For this reason, this transformation is frequently used for compressing data [2, 51, 74].

The chief drawback of the PCA transformation is that the resulting vectors are no longer physically meaningful. They do not represent faces or any other shape directly. They only represent the mathematical notion of linearly spanning a vector space. For this reason it is hard to impose meaningful constraints when fitting data to these bases. For example, we would like to constrain solutions to the range of valid faces, disallowing extreme extrapolation from observed facial expressions. We previously considered constraining basis weights to the range (0,1) in order to insure interpolation between valid points in the space. In the space of eigenvectors, there is no obvious analogous constraint to apply. Without a way to apply constraints against extrapolation, it is possible that recovered missing features will be grossly distorted.

Two potential methods for selecting a basis set have been described. Both methods have successfully been used in conjunction with this work for recovering missing features. A more thorough investigation of the best method of picking basis vectors is warranted, but beyond the scope of this document.

5.5 Evaluation

The small scale recovery system can be evaluated by analyzing the error inherent in determining feature locations. Since motion recovery takes place inside a sub-volume relative to the entire space, and this sub-volume is moving, it is appropriate to measure error in terms of the motion between recovered features. That is, the relative geometry and pose between multiple recovered small scale features should be faithfully recovered, while the large scale motion that the target is undergoing is of lower importance. This section has also proposed a model for reconstructing occluded features. The quality of these reconstructed features should be evaluated. Using triangulated data as ground truth, it is possible to analyze the deviation in target position that is expected when using the model to recreate occluded features.

The primary motivational example for the small scale motion recovery system in this work is a human face. Figure 62 shows still frames from a complete sequence in which the motion of

a face was captured. In addition to the reconstructed 3D geometry, the view from one camera is shown so that a visual comparison can be made. Note that the geometry closely resembles the original. Under normal operating conditions the face would be undergoing large scale motion. In order to increase the ease of visual comparison the pan-tilt cameras were not allowed to move in this example. Rather they were oriented in a constant direction.

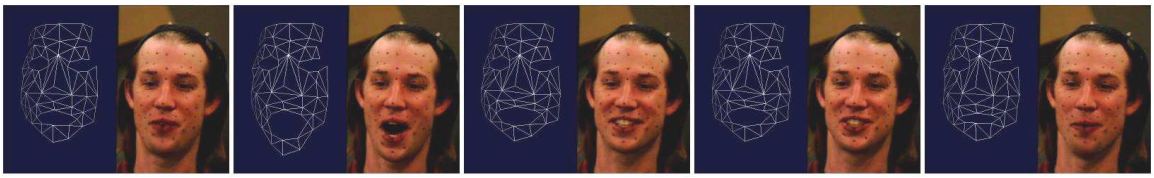


Figure 62 *Video figure* Still frames from a sequence of recovered facial geometry.

Geometric pose can of course be reconstructed when the face is in motion. Figure 63 shows a plot of global facial motion for one such capture session. All frames have been superimposed into a single graph. As can be seen from the figure, the motion in this case spans a two meter range. However, because the motion is represented at the global scale it is difficult to deduce any meaningful measure of quality.

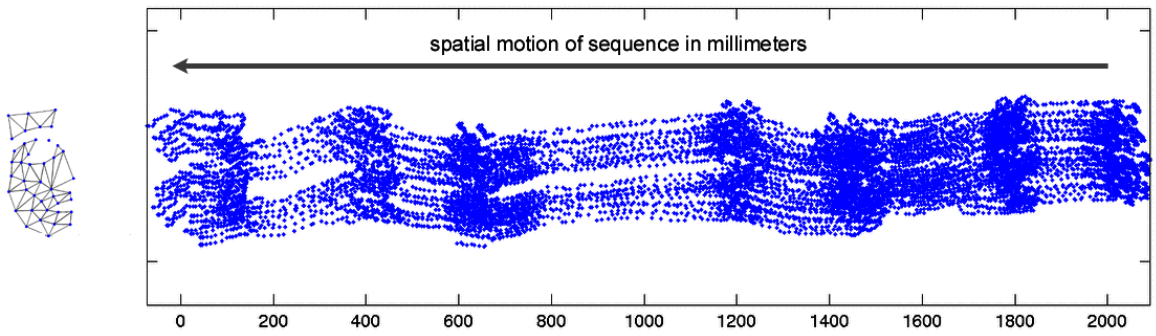


Figure 63 Facial motion recovered by the small scale recovery system, during one capture sequence. The global head motion is on the order of two meters.

A metric for analyzing small scale recovery quality should be invariant to large scale facial motion. One such metric is the distance between triangulated feature points. Features on some parts of the face should maintain a constant separation regardless of facial motion,

while others will clearly vary over time. Figure 64 shows the recovered face from the previous sequence, together with plots of the distance between two different feature pairs. For each pair the mean separation over the course of the sequence is defined as the neutral pose.

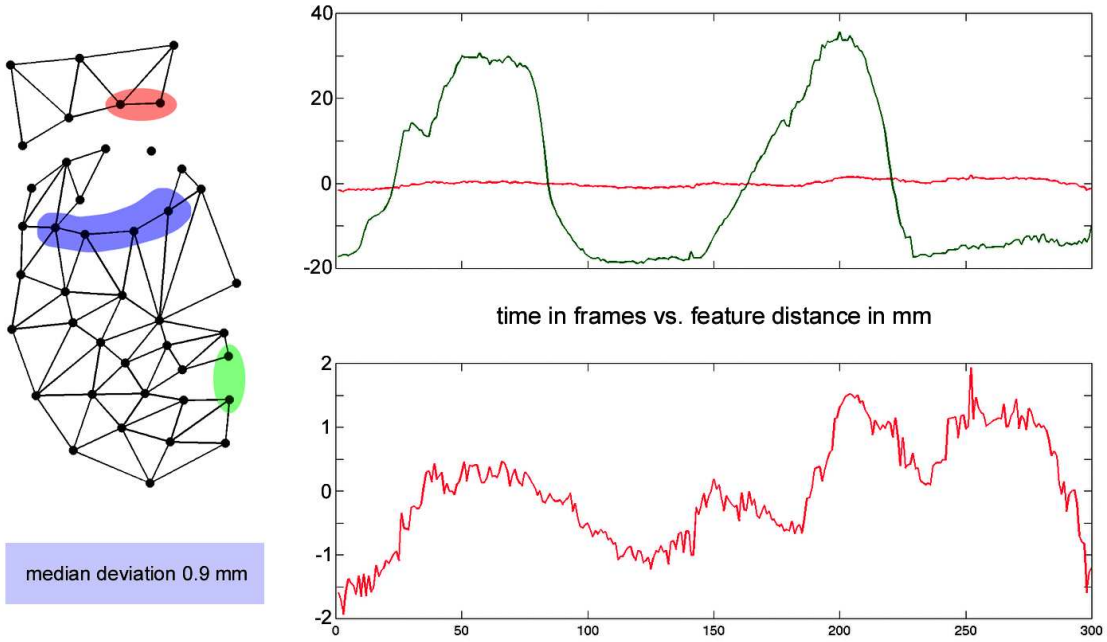


Figure 64 Distance between facial features. The distance between the green pair of features on the lips varies over time, while the distance between the red pair of features on the forehead remains relatively constant. The plots on the right show actual distance deviation during one motion recovery session. In addition to the lower right plot, the median accuracy of the small scale recovery system was quantified to be 0.9 mm by evaluating the distance deviation between all pairs of features marked in blue.

The feature pair marked in green, on opposing lips, is expected to have a large variation in distance over time. As seen from the plot on the upper right, these features do in fact vary in distance by 50 mm. In contrast, the feature pair marked in red, on the forehead, is expected to be relatively static. From the upper right plot it is clear that the distance between these features is in fact nearly constant. Any deviation from a constant distance is likely due to inaccuracies in the small scale recovery system. The lower right plot is an expanded view of the distance between the two red features. It is clear that feature locations can vary on the order of 2 mm from the expected neutral pose. In order to further quantify the expected accuracy of the small scale recovery system, a similar distance measure was applied to the group of features below the eye, marked in blue. These features are not expected to have significant motion relative to one another. The pairwise deviation from the neutral distance

aggregated over all combinations of these four features was computed. The median deviation for these features was found to be 0.9 mm.

Facial features occluded during small scale recovery can be reconstructed using a facial model. The quality of reconstructed missing features can be evaluated as shown in Figure 65. A sequence of 100 frames was captured and all features were triangulated. Eleven feature points were synthetically removed from the sequence, to simulate unobserved features due to occlusion. This new sequence was then fit to a face model. The face model was constructed from a different sequence of 200 frames, that was then reduced to a basis set of 80 vectors. After fitting, the model was used to reconstruct all feature points. The reconstructed location of removed feature points was compared to the original location determined via triangulation. A plot of error projected onto the frontal XY plane is shown in Figure 66. As can be seen, the reconstructed points lie primarily within 1.5 mm of the correct feature locations.

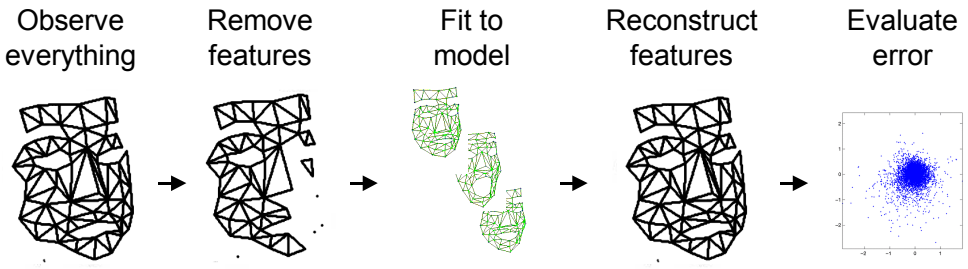


Figure 65 The quality of reconstructed features can be evaluated by synthetically removing features to simulate occlusion. After reconstruction, these features are compared to ground truth data and found to vary by less than 1.5 mm from their original location.

As more features are occluded, the amount of data available to fit and therefore reconstruct the correct face is diminished. In order to investigate the robustness of this method when a large number of features are occluded a sequence was used in which a successively larger number of features disappear as the sequence progresses. The sequence has 431 frames, starting with 42 of the total 76 features observed, and ending with only 16 features. This is much more extreme occlusion than would be expected in an actual capture scenario. The model was created from a sequence of 321 frames, reduced to a basis set of five principal components. Ideally a valid face will be reconstructed by the model, even with minimal data available. A sequence of still frames from this sequence is shown in Figure 67. As can be seen, the reconstructed face behaves as expected even when many features are occluded.

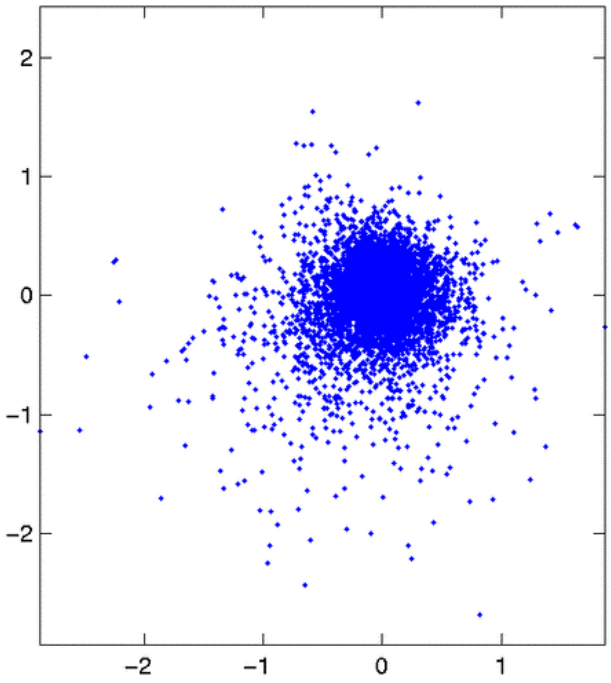


Figure 66 A plot of error, measured as the distance between reconstructed missing features and the ground truth location of those features. The error is shown projected onto the frontal, XY, plane of the face, in mm.

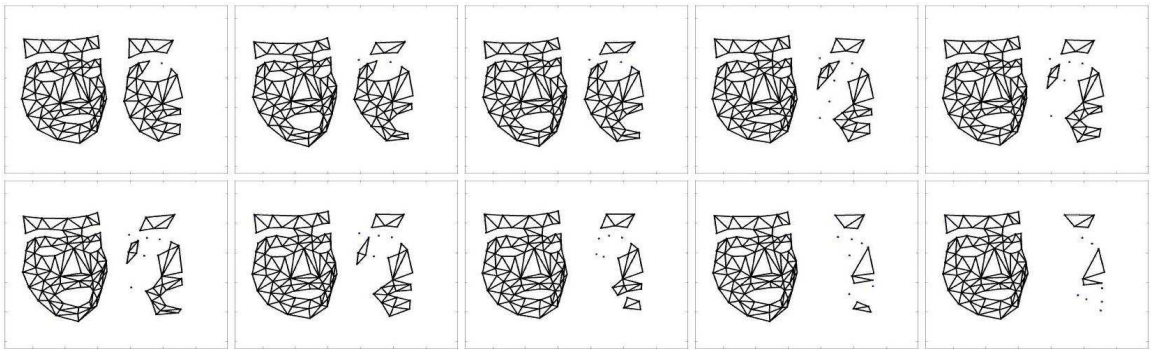


Figure 67 *Video figure* A sequence of frames showing the quality of reconstruction as successively more feature points are occluded. Even with many occluded features, the reconstructed face remains believable. Right. Data acquired via triangulation, many features are occluded. Left. Reconstructed complete facial model.

6 Conclusions

This dissertation has presented a framework and system design for acquiring mixed scale motions. A number of application domains for the technology presented here exist, and a brief overview of a few of these applications is given below. We conclude by reviewing the contributions of this dissertation and discussing possible avenues of future work.

6.1 Applications

This thesis proposes and demonstrates a motion recovery system that is applicable to mixed scale environments. A number of applications exist for this technology. The primary motivating application for this work is simultaneous recovery of facial and body motion. The entertainment industry makes heavy use of motion capture while creating games and movie special effects. Using traditional capture systems, facial and body motion must be independently acquired. Since these two motions are captured separately, their timing will not be well synchronized and significant effort is typically required by a skilled animator to make corrections.

Using this system, face and body motion can be captured simultaneously. The large scale sub-system recovers body motion, while the small scale sub-system recovers detailed facial deformation. Figure 68 shows an example of a person walking across a laboratory while swinging his arms and changing facial pose. Although this motion is not very interesting it is sufficient to demonstrate that facial motion can be recovered despite the fact that the small scale working volume surrounding the head traverses a rather large percentage of the global working volume.

Figure 68 *Video figure*. Live action of simultaneous body and facial motion. The subjects large scale motion causes the sub-volume within which small scale facial motion occurs to traverse most of the global working volume.

The body motion recovered from this sequence is shown in Figure 69. The three dimensional motion of each target has been recovered by the large scale system. As a post process, connecting bones were added to the motion rendering to help visualize the bodies pose. In a complete animation pipeline, this skeletal motion would be used to drive the motion of an animated character.

Figure 69 *Video figure*. Recovered large scale body motion. The three dimensional motion of each target has been recovered, and is used here to visualize the overall motion of the subject.

The facial motion recovered by the small scale recovery system is shown in Figure 70. Individual features on the face were recovered via triangulation, and then the whole face was fit to a facial model to reconstruct any occluded features. As a post process, a triangulation was specified to connect facial features in order to aid in visualizing the recovered motion. As with body motion, the recovered facial motion can be used to drive the motion of animated character.

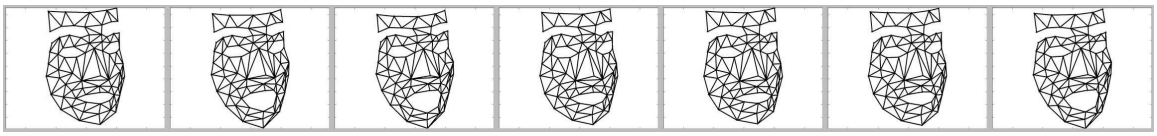


Figure 70 *Video figure*. Recovered small scale body motion. The motion of each facial feature was recovered, and the entire data set fit to a facial model. This model was used to reconstruct the motion of any occluded features.

In addition to recovering face and body motion, the system installed in our laboratory has been used for a few other mixed scale tasks. The system can be used exactly as described to track the deformation of other small scale objects. One example of this is tracking the deformation of a subjects knee as they walk across a room. The ability to capture detailed motion and shape that occurs in a larger space, may in the future help researchers develop biomechanical models that more accurately reflect natural movement conditions. Figure 71 shows still frames from one such sequence. Two camera viewpoints are shown, together with a course model of the recovered knee shape.

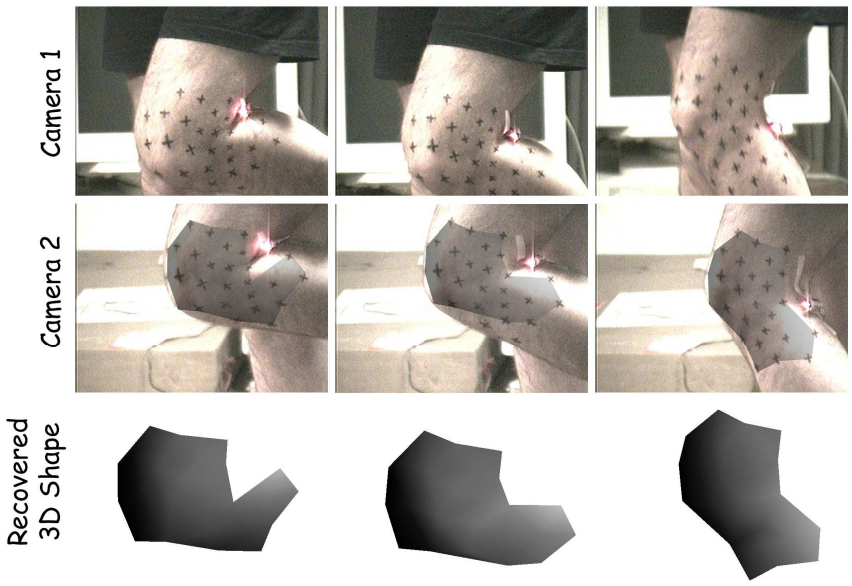


Figure 71 Still frames from the recovered motion of knee deformation, while a subject walks across a room.

Both of the previous examples determine target motion and shape by triangulating directly between several of the pan-tilt cameras used in this system. Other algorithms exist for determining shape, and their range can also be expanded through the use of a hierarchical solution. Figure 72 shows a sequence of frames and corresponding recovered geometry, captured with a mixed scale structured light system. This system captures high resolution dense geometry by triangulating between known patterns of projected light and a single camera [54]. Most structured light systems have a limited range to resolution ratio dependant on both the projector and the camera. This system uses a high resolution wide angle projector, and a relatively low resolution narrowly focused camera. The working volume in

which shape can be captured was expanded by using the large scale motion tracking system to locate a sub-region of interest, and the sub-region selection mechanism to orient a pan-tilt camera towards this sub-region. The video from the pan-tilt camera was then used in conjunction with structured light recovery.

The large scale motion recovery sub-system presented in this work is interesting on its own. It has been used separately for tasks in augmented reality and human computer interaction.

Augmented reality systems present virtual 3D objects to a user but do not entirely replace their field of view with a virtual environment. Well known examples include the CAVE architecture and the Responsive Workbench [31, 64]. A key component of these technologies is the ability to track a users head position so that the projected computer image is rendered from the correct viewpoint. Our laboratory has a high resolution display wall [53, 60] on which images can be rendered. Using the large scale recovery system to track the position of a users head, a virtual object can be rendered such that it appears with the correct geometry, irrespective of the persons position in the room. A sequence of images demonstrating this application is shown in Figure 73.

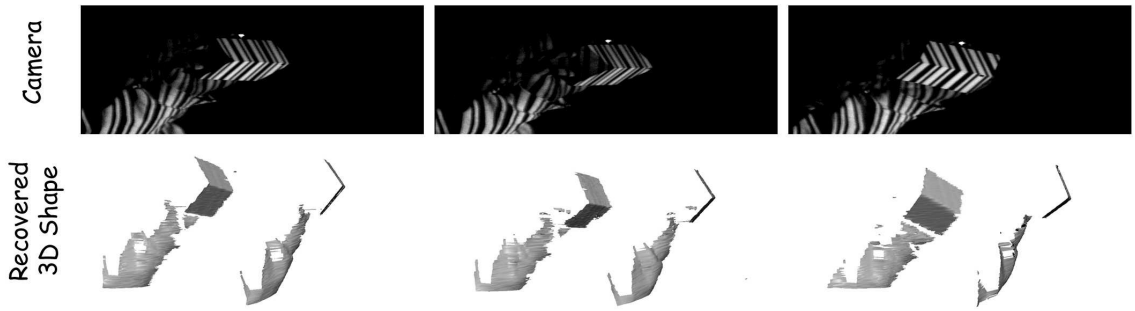


Figure 72 Mixed scale structured light system. A static projector and a dynamically oriented camera are used to capture geometry via triangulation of known projected stripe patterns.

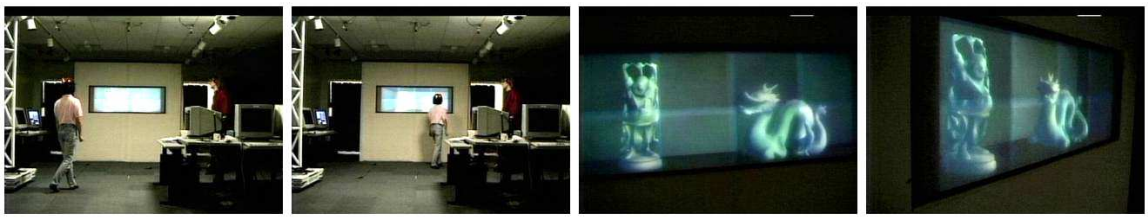


Figure 73 Using the large scale motion recovery sub-system to track a users head position ensures that the correct projection of a virtual object can be rendered on a wall size display.

The large scale system has also been used as a foundation for scalable pointer-based input technology [34]. By mounting a set of wide area static cameras behind the high-resolution display wall, rather than on the ceiling, target motion on the display surface can be recovered. Hand held laser pointers can be used to interact directly with contents on the display. The laser pointers produce bright spots which are tracked by the large scale recovery system. The domain of tracking has been reduced to a two dimensional space, but otherwise the issues are very similar. Figure 74 shows a group of people using this technology to interact with content on the wall size display.

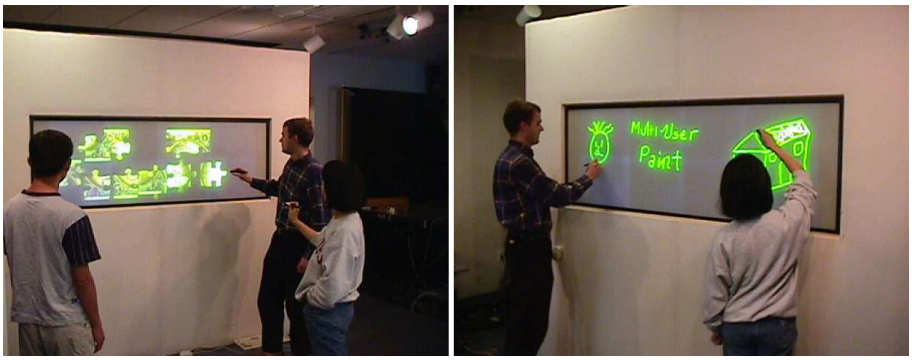


Figure 74 Users interact with a wall size display using a scalable pointer based input technology. The architecture of this input system draws directly from the large scale tracking presented in this work.

There are many other potential applications for mixed scale recovery technologies. The demonstration applications discussed here merely represent the subset implemented in our laboratory.

6.2 Contributions

This thesis has discussed mixed scale motion recovery over a broad range of abstraction, from high level goals to implementation details of particular components. Within this scope three primary contributions have been presented. A framework for thinking about mixed scale motion recovery, a specific system design, and the application of these ideas to simultaneous face and body motion recovery.

Framework for mixed scale motion recovery. A framework for thinking about mixed scale motion recovery was presented. The cornerstone of this framework lies in the

characterization of the problem domain as inherently *mixed scale*. This characterization suggests a hierarchical paradigm of system design. The system presented here is one instance of this paradigm that explicitly mirrors the independent scales of motion using pan-tilt cameras. We have also seen how conceptualizing a large system in terms of interfaces, rather than components can be helpful. This method allows many system challenges to be addressed as instances of inadequate data flow, to which corrective data models can be applied. For example, in this work a Kalman filtering motion model, a pan-tilt camera model, and a facial pose model were used to address specific challenges.

Specific system design. A specific design was presented that demonstrates the feasibility of a mixed scale motion recovery system. This design provides a high resolution to range ratio by using multiple sub-systems. The wide area sub-system uses many cameras and is optimized for coverage. This sub-system is scalable to large environments and robust to occlusion. A set of pan-tilt cameras is optimized for high resolution imaging. This sub-system provides accurate recovery of detailed motion. Specific details of this system relating to feature detection, target integration, track initiation, latency, pan-tilt control, and facial modeling, among others, have been presented. In addition to design and implementation, the system in our laboratory was evaluated using a number of metrics in order to characterize the expected performance.

Application to simultaneous face-body recovery. Many real world applications are inherently mixed scale. In order to demonstrate the potential of mixed scale recovery for addressing these applications, the system described in this thesis has been applied to the particular task of simultaneously recovering face and body motion. This is an application of practical importance to the animation industry that currently must perform these captures separately. A production strength implementation of the ideas presented here would be a substantial benefit to motion capture studios.

6.3 Future Work

A number of interesting additions to this work exist. The most obvious direction in which to gain further experience is in scaling to truly large environments. The laboratory environment in which this work was performed is not the ideal demonstration of the importance of mixed scale technology. Existing motion capture systems already do a reasonable job of capturing

many interesting motions in these environments. However no feasible solution yet exists for capturing human motion in stadium size environments. Conceptually, capturing motion in a stadium is identical to capturing motion in a room, and the technologies demonstrated here could be directly applied.

Another direction of practical importance is recovering multiple small scale targets. A trivial implementation is possible in which the pan-tilt sub-region selection mechanism is merely replicated for each desired small scale target. However, a more interesting direction is with regard to optimal resource allocation. Given a set of small scale targets and a pool of available pan-tilt cameras for sub-region selection, an optimal assignment of resources to targets could be found.

As the working volume grows to the size of a stadium, it may become important to capture motion at more than two scales. While the system demonstrated in this work has only two levels, the framework presented suggests a paradigm for building systems with additional scales. Although this is theoretically straightforward, it is likely of practical difficulty. One of the key challenges in this work is stability of the selected sub-region. Tracking of small scale features is reasonably robust to small rotational errors in the pan-tilt camera's orientation. It is possible that much greater stability would be required if a selected sub-region was required to itself act as a base volume for even further sub-region selection.

Although the specific system presented here makes use of pan-tilt cameras for sub-region selection, it is not clear that this is the optimal method under all conditions. In particular the cost of mechanical components is nearly constant, while the cost of imagers and processing power is decreasing rapidly. Systems that make use of many many cameras will eventually become feasible and investigating the issues involved with that architecture would be of value. In particular there are likely trade-offs that could be analyzed to determine when each architecture is the appropriate choice.

The directions suggested for future work predominantly apply to the high level goal of mixed scale motion recovery. In addition, a great deal of interesting work remains to be done with regard to tracking, feature detection, motion models, camera calibration, marker prediction, and facial modeling.

References

1. Agrawala, M., Beers, A. C., Frohlich, B., Hanrahan, P., McDowall, I., and Bolas, M. *The two-user Responsive Workbench: support for collaboration through individual views of a shared space*. in 24th International Conference on Computer Graphics and Interactive Techniques. 1997. Los Angeles CA USA.

2. Alexa, M. and Müller, W., *Representing Animations by Principal Components*. Eurographics (Computer Graphics Forum), 2000. 19(3).

3. Alexander, E. J. and Andriacchi, T. P., *Correcting for deformation in skin-based marker systems.* Journal of Biomechanics, 2001. 34(3): p. 355-61.

4. Analogous, http://www.analogus.com/

5. Andriacchi, T. P. and Alexander, E. J., *Studies of human locomotion: past, present and future.* Journal of Biomechanics, 2000. 33(10): p. 1217-24.

6. Ascension, http://www.ascension-tech.com/

7. Azuma, R., *A Survey of Augmented Reality.* Presence: Teleoperators and Virtual Environments, 1997. 6(4): p. 355-385.

8. Azuma, R. and Bishop, G. *A frequency-domain analysis of head-motion prediction*. in SIGGRAPH '95. 1995: ACM.

9. Barreto, J. P., Peixoto, P., Batista, J., and Araujo, H. *Tracking multiple objects in 3D*. in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99). 1999.

10. Bar-Shalom, Y., *Multitarget-multisensor tracking*. 1990: Artech House.

11. Bar-Shalom, Y. and Fortmann, T. E., *Tracking and Data Association*. 1988: Academic Press.

12. Basu, A. and Ravi, K., *Active camera calibration using pan, tilt and roll.* IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 1997. 27(3): p. 559-66.

13. Birchfield, S. *Elliptical Head Tracking Using Intensity Gradients and Color Histograms*. in IEEE Conference on Computer Vision and Pattern Recognition (CVPR98). 1998. Santa Barbara.

14. Black, M. J. and Jepson, A. D., *EigenTracking: robust matching and tracking of articulated objects using a view-based representation.* International Journal of Computer Vision, 1998. 26(1): p. 63-84.

15. Blackman, S. and Popoli, R., *Design and analysis of modern tracking systems*. Artech House radar library. 1999: Artech House. 1230.

16. Blanz, V. and Vetter, T. *A morphable model for the synthesis of 3D faces*. in SIGGRAPH 99: 26th International Conference on Computer Graphics and Interactive Techniques. 1999. Los Angeles CA USA: New York, NY, USA : ACM, 1999.

17. Bodenheimer, B., Rose, C., Rosenthal, S., and Pella, J. *The Process of Motion Capture: Dealing with the Data*. in Computer Animation and Simulation, Eurographics Animation Workshop. 1997: Springer-Verlag, Wein.

18. Bradshaw, K. J., McLauchlan, P. F., Reid, I. D., and Murray, D. W., *Saccade and pursuit on an active head/eye platform.* Image and Vision Computing, 1994. 12(3): p. 155-63.

19. Bregler, C. and Malik, J. *Video Motion Capture*, UC Berkeley: UCB//CSD-97-973, 1997.

20. Bregler, C. and Malik, J. *Tracking people with twists and exponential maps*. in IEEE Computer Society Conference on Computer Vision and Pattern Recognition,. 1998.

21. Brown, R. G. and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering. Second Edition*. 1992.

22. Calvert, T. W., Chapman, J., and Patla, A., *Aspects of the kinematic simulation of human movement.* IEEE Computer Graphics and Applications, 1982. Vol. 2(No. 9): p. 41=50.

23. Cham, T.-J. and Rehg, J. M. *A multiple hypothesis approach to figure tracking*. in IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1999.

24. Chauhari, A., Bragg, R., Alexander, E., and Andriacchi, T. *A Video-Based, Markerless Motion Tracking System for Biomechanical Analysis in an Arbitrary Environment*. in Bioengineering Conference. 2001. Snowbird, Utah: ASME.

25. Chen, X. *Design of Many-Camera Tracking Systems for Scalability and Efficient Resource Allocation.* Ph.D. Dissertation from Electrical Engineering, Stanford University, 2002.

26. Chen, X. and Davis, J., *Wide Area Camera Calibration Using Virtual Calibration Objects.* IEEE Computer Vision and Pattern Recognition, 2000.

27. Coco, D., *Motion Capture Advances*, in *Computer Graphics World*. 1997.

28. Collins, R. T. and Tsin, Y. *Calibration of an outdoor active camera system*. in IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1999: Los Alamitos, CA, USA : IEEE Comput. Soc, 1999.

29. Comon, P., *Independent component analysis: A new concept.* Signal Processing, 1994. 36: p. 287-314.

30. Cootes, T. F., Edwards, G. J., and Taylor, C. J. *Active Appearance Models*. in European Conference on Computer Vision. 1998: Springer.

31. Cruz-Neira, C., Sandin, D. J., and DeFanti, T. A. *Surround-screen projection-based virtual reality: the design and implementation of the CAVE*. in SIGGRAPH 20th Annual International Conference on Computer Graphics and Interactive Techniques. The Eye of Technology. 1993. Anaheim CA USA.

32. Cutler, L. D., Frohlich, B., and Hanrahan, P. *Two-handed direct manipulation on the Responsive Workbench*. in Symposium on Interactive 3 D Graphics. 1997. Providence, RI, USA.

33. Darrell, T., Moghaddam, B., and Pentland, A. *Active Face Tracking and Pose Estimation in an Active Room*. in CVPR. 1996: IEEE.

34. Davis, J. and Chen, X., *LumiPoint: Multi-User Laser-Based Interaction on Large Tiled Displays.* Displays (to appear), 2002.

35. DiFranco, D., Cham, T.-J., and Rehg, J. *Recovery of 3D Articulated Motion from 2D Correspondences*, Compaq Cambridge Research Laboratory: CRL 99/7, 1999.

36. Eisert, P. and Girod, B., *Analyzing facial expressions for virtual conferencing.* IEEE Computer Graphics and Applications, 1998. 18(5): p. 70-8.

37. Essa, I., Basu, S., Darrell, T., and Pentland, A. *Modeling, tracking and interactive animation of faces and heads//using input from video*. in Computer Animation '96. 1996.

38. Essa, I. A. and Pentland, A. P., *Coding, analysis, interpretation, and recognition of facial expressions.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997. 19(7): p. 757-63.

39. EVI-D30. *EVI-D30 Sony Color Video Camera Operating Instructions*, Sony

40. Fairhurst, M., *Computer Vision for Robotic Systems An Introduction*. 1988: Prentice Hall.

41. Faugeras, O., *Three-Dimensional Computer Vision, A Geometric Viewpoint*. 1993: MIT Press.

42. Frishberg, B., *An analysis of overground and treadmill sprinting.* Medicine and Science in Sports and Exercise, 1983. 15: p. 478-485.

43. Fry, S., Bichsel, M., Muller, P., and Robert, D., *Tracking of flying insects using pan-tilt cameras.* Journal of Neuroscience Methods, 2000. 101: p. 59-67.

44. Furniss, M. *Motion Capture*. in Media in Transition, An International Conference. 1999. MIT.

45. Gavrila, D. M., *The visual analysis of human movement: a survey.* Computer Vision and Image Understanding, 1999. 73(1): p. 82-98.

46. Ginsberg, C. M. and Maxwell, D. *Graphical marionette*. in ACM SIGGRAPH/SIGART Workshop on Motion. 1983.

47. Gliecher, M., *Animation From Observation: Motion Capture and Motion Editing.* Computer Graphics, 1999. 33(4): p. 51-54.

48. Gokturk, S. B., Bouguet, J.-Y., and Grzeszczuk, R. *A Data-Driven Model for Monocular Face Tracking*. in International Conference on Computer Vision (ICCV). 2001. Vancouver, Canada: IEEE.

49. Goldman, M., *Practical MoCap: Motion Capture for TV*, in *millimeter: The Motion Picture and Television Production Magazine*. 1999.

50. Goldman, M., *Real-Time Optical Motion Capture: Will it Change How You Make TV Animations?*, in *millimeter: The Motion Picture and Television Production Magazine*. 2000.

51. Guenter, B., Grimm, C., Wood, D., Malvar, H., and Pighin, F., *Making faces*, in Computer Graphics. Proceedings. SIGGRAPH 98 Conference Proceedings. 1998, Acm: New York, NY, USA. p. 472.

52. Guernsey, L., *Turning the Super Bowl Into a Game of Pixels*, in *The New York Times*. 2001.

53. Guimbretière, F., Stone, M., and Winograd, T. *Fluid Interaction with High-resolution Wall-size Displays*. in UIST : User Interface Software and Technology. 2001: ACM.

54. Hall-Holt, O. and Rusinkiewicz, S. *Stripe Boundary Codes for Real-Time Structured-Light Range Scanning of Moving Objects*. in ICCV. 2001: IEEE.

55. Hansen, M., Anandan, P., Dana, K., Wel, G. v. d., and Burt, P. *Real-time Scene Stabilization and Mosaic Construction*. in IEEE Workshop on Applications of Computer Vision. 1994.

56. Haralick, R. and Shapiro, L., *Computer and Robot Vision*. 1992: Addison-Wesley.

57. Heikkila, J. and Silven, O., *A Four-Step Camera Calibration Procedure with Implicit Image Correction.* IEEE Computer Vision and Pattern Recognition, 1997: p. 1106-1112.

58. Herda, L., Fua, P., Plankers, R., Boulic, R., and Thalmann, D., *Using skeleton-based tracking to increase the reliability of optical motion capture.* Human Movement Science, 2001. 20(3): p. 313-41.

59. Howe, N., Leventon, M., and Freeman, W. *Bayesian Reconstruction of 3D Human Motion from Single-Camera Video*, MERL - Mitsubishi Electric Research Laboratory: TR-99-37, 1999.

60. Humphreys, G. and Hanrahan, P., *A Distributed Graphics System for Large Tiled Displays.* IEEE Visualization '99, 1999: p. 215-224.

61. Isard, M. and Blake, A., *CONDENSATION-conditional density propagation for visual tracking.* International Journal of Computer Vision, 1998. 29(1): p. 5-28.

62. Jollife, I. T., *Principal Component Analysis*. 1986, New York: Springer Verlag.

63. Kakadiaris, L. and Metaxas, D., *Model-based estimation of 3D human motion.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. 22(12): p. 1453-9.

64. Krueger, W. and Froehlich, B., *The Responsive Workbench (virtual work environment).* IEEE Computer Graphics and Applications, 1994. 14(3): p. 12-15.

65. Kuhn, T. S., *The structure of scientific revolutions*. 1962: University of Chicago Press.

66. Lawson and Hanson, *Solving Least Squares Problems*. 1974: Prentice-Hall.

67. Lee, D. and Seung, S., *Learning the parts of objects by non-negative matrix factorization.* Nature, 1999. 401: p. 788-791.

68. Lucas, B. D. and Kanade, T., *An iterative image registration technique with an application to stereo vision.* Seventh International Joint Conference on Artificial Intelligence (IJCAI-81), 1981.

69. Marcenaro, L., Vernazza, G., and Regazzoni, C. *image Stabilization Algorithms for Video-Surveillance Applications*. in International Conference on Image Processing. 2001: IEEE.

70. Martzke, R., *Instant replay gets "Matrix'-like view of key plays*, in *USA TODAY*. 2001.

71. Maxwell, D. *Graphical Marionette: A Modern Day Pinocchio".* S.M. Visual Studies from Architecture Machine Group, MIT, 1983.

72. Mills, D. L. *Network Time Protocol (ntp) specification, implementation and analysis*, Network Working Group Report: RFC-1305, 1992.

73. Moeslund, T. B. and Granum, E., *A survey of computer vision-based human motion capture.* Computer Vision and Image Understanding, 2001. 81(3): p. 231-68.

74. Moghaddam, B. and Pentland, A. *An Automatic System for Model-Based Coding of Faces*. in IEEE Data Compression Conference. 1995: IEEE.

75. MotionAnalysis, http://www.motionanalysis.com

76. Nickels, K., *Model-Based Tracking of Complex Articulated Objects.* IEEE Trans. on Robotisc and Automation, 1999. 17(1): p. 28-36.

77. Nigg, B., Boer, R., and Fisher, V., *A kinematic comparison of overground and treadmill running.* Medicine and Science in Sports and Exercise, 1995. 27: p. 98-105.

78. Noh, J.-y. and Neumann, U. *A Survey of Facial Modeling and Animation Techniques*, Univeristy of Southern Californina: USC-TR-99-705, 1999.

79. Ong, E.-J. and Gong, S. *Tracking hybrid 2D-3D human models from multiple views*. in IEEE International Workshop on Modelling People. 1999.

80. Peak, http://peakperform.com/

81. Peixoto, P., Batista, J., and Araujo, H. *A surveillance system combining peripheral and foveated motion tracking*. in Fourteenth International Conference on Pattern Recognition. 1998. Brisbane Qld. Australia: Los Alamitos, CA, USA : IEEE Comput. Soc, 1998.

82. Pentland, A., Moghaddam, B., and Starner, T. *View-based and modular eigenspaces for face recognition*. in IEEE Conference on Computer Vision and Pattern Recognition. 1994: Los Alamitos, CA, USA : IEEE Comput. Soc. Press, 1994.

83. Phoenix, http://ptiphoenix.com

84. Plankers, R. and Fua, P., *Tracking and modeling people in video sequences.* Computer Vision and Image Understanding, 2001. 81(3): p. 285-302.

85. Polhemus, http://www.polhemus.com/

86. Press, W., Teukolsky, S., Vetterling, W., and Flannery, B., *Numerical Recipes in C : The Art of Scientific Computing, 2nd Edition*. 1992: Cambridge University Press.

87. Ramachandra, K., *Kalman filtering techniques for radar tracking*. 2000: Marcel Dekker. 241.

88. Rander, P. *A Multi-Camera Method for 3D Digitization of Dynamic, Real-World Events.* from Robotics Institute, Carnegie Mellon University, 1998.

89. Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. *The office of the future: a unified approach to image-based modeling and spatially immersive displays*. in SIGGRAPH 98: 25th International Conference on Computer Graphics and Interactive Techniques. 1998.

90. Rau, G., Disselhorst-Klug, C., and Schmidt, R., *Movement biomechanics goes upwards: from the leg to the arm.* Journal of Biomechanics, 2000. 33(10): p. 1207-16.

91. Rehg, J. M. and Kanade, T. *DigitEyes: vision-based hand tracking for human-computer interaction*. in 1994 IEEE Workshop on Motion of Non-rigid and Articulated Objects. 1994. Austin TX USA.

92. Rehg, J. M. and Kanade, T. *Model-based tracking of self-occluding articulated objects*. in IEEE International Conference on Computer Vision. 1995.

93. Rui, Y., He, L., Gupta, A., and Liu, Q. *Building an Intelligent Camera Management System*. in ACM Multimedia. 2001.

94. Shih, S.-W., Hung, Y.-P., and Lin, W.-S., *Calibration of an active binocular head.* IEEE Transactions on Systems, Man & Cybernetics, Part A (Systems & Humans), 1998. 28(4): p. 426-42.

95. Sidenbladh, H., Black, M., and Fleet, D., *Stochastic Tracking of 3D Human Figures Using 2D Image Motion.* ECCV 2000, European Conference on Computer Vision, 2000: p. 702-718.

96. Sidenbladh, H., De la Torre, F., and Black, M. J. *A framework for modeling the appearance of 3D articulated figures*. in Fourth International Conference on Automatic Face and Gesture Recognition. 2000.

97. Simi, http://www.simi.com/

98. Stenger, B., Mendonca, P. R. S., and Cipolla, R. *Model-based 3D tracking of an articulated hand*. in IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. 2001.

99. Sturman, D. *A Brief History of Motion Capture*. in SIGGRAPH 94 : Course 9 Notes. 1994: ACM.

100. Sturman, D., *The State of Computer Animation.* ACM SIGGRAPH Computer Graphics Newsletter, 1998. Vol. 32 No.1.

101. Tomasi, C. and Kanade, T. *Shape and Motion from Image Streams: A Factorization Method Part 3. Detection and Tracking of Point Features*, Carnegie Mellon University Technical Report: CMU-CS-91-132, 1991.

102. Tomasi, C. and Zhang, J. *How to rotate a camera*. in ICIAP '99 - 10th International Conference on Image Analysis and Processing. 1999: IEEE.

103. Toyama, K. and Hager, G. D. *Incremental focus of attention for robust visual tracking*. in IEEE Conference on Computer Vision and Pattern Recognition. 1996. Los Alamitos, CA, USA: IEEE Comput. Soc. Press.

104. Trucco, E. and Verri, A., *Introductory Techniques for 3-D Computer Vision*. 1998: Prentice Hall.

105. Tsai, R. Y., *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses.* IEEE Trans. Robotics and Automation, 1987. 3(4): p. 323-344.

106. Vicon, http://www.vicom.com/

107. Wachter, S. and Nagel, H. H. *Tracking of persons in monocular image sequences*. in IEEE Nonrigid and Articulated Motion Workshop. 1997.

108. Welch, G. *SCAAT: Incremental Tracking with Incomplete Information*, University of North Carolina at Chapel Hill: TR 96-051, 1996.

109. Welch, G. and Bishop, G. *An Introduction to the Kalman Filter*, University of North Carolina at Chapel Hill, Department of Computer Science: TR 95-041, 1995.

110. Welch, G. and Bishop, G. *SCAAT: incremental tracking with incomplete information*. in 24th International Conference on Computer Graphics and Interactive Techniques. 1997: ACM.

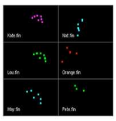
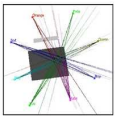
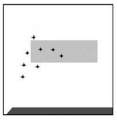
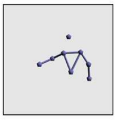
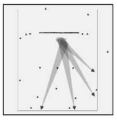
111. Wexelblat, A. *Natural gesture in virtual environments*. in VRST94: Virtual Reality Software and Technology. 1994.

112. White, S., Yack, J., Tucker, C., and Lin, H.-Y., *Comparison of Vertical ground reaction forces during overground and treadmill walking.* Medicine and Science in Sports and Exercise, 1998. 30: p. 1537-42.

113. Woo, D. C. and Capson, D. W. *3D visual tracking using a network of low-cost pan/tilt cameras*. in Canadian Conference on Electrical and Computer Engineering Conference Proceedings. 2000.

114. X-IST, http://www.x-ist.de/

115. Zhang, J. Z. and Wu, Q. M. J., *A pyramid approach to motion tracking.* Real-Time Imaging, 2001. 7(6): p. 529-44.

116. Zhang, Z., *A Flexible New Technique for Camera Calibration.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. 22(11): p. 1330-1334.

mixed scale
motion recovery