MIT Media Laboratory, Perceptual Computing Technical Report #363
January 1996, Appears in: ICPR'96 Vienna, Austria August 1996

# Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features

Ali Azarbayejani and Alex Pentland
Perceptual Computing Section, MIT Media Lab, Cambridge, MA, USA

## Abstract

*We describe a method for estimation of 3-D geometry from 2-D* blob features. *Blob features are clusters of similar pixels in the image plane and can arise from similarity of color, texture, motion and other signal-based metrics. The motivation for considering such features comes from recent successes in real-time extraction and tracking of such blob features in complex cluttered scenes in which traditional feature finders fail—scenes containing moving people, for example.*

*We use nonlinear modeling and a combination of iterative and recursive estimation methods to recover 3-D geometry from blob correspondences across multiple images. The 3-D geometry includes the 3-D shapes, translations, and orientations of blobs and the relative orientation of the cameras.*

*Using this technique, we have developed a real-time wide-baseline stereo person tracking system which can self-calibrate itself from watching a moving person and can subsequently track people's head and hands with RMS errors of 1–2 cm in translation and 2 degrees in rotation. The blob formulation is efficient and reliable, running at 20–30 Hz on a pair of SGI Indy R4400 workstations with no special hardware.*

## 1. Introduction

This paper describes a method for estimation of 3-D geometry from *blob features* and evaluates the performance of a real-time self-calibrating stereo person tracker that uses the estimator.

The notion of "blobs" as a representation for image features has a long history in computer vision [4, 6, 1, 12]. The term "blob" is somewhat self-explanatory ("something of vague or indefinite form"), but a useful definition from a computational point of view might be that a blob is defined by some visual property that is shared by all the pixels contained in the blob and is not shared by surrounding pixels. This property could be color, texture, brightness, motion, shading, a combination of these, or any other salient spatio-temporal property derived from the signal (the image sequence).

Our current interest in blob models is motivated by our discovery that they can be reliably tracked even in complex, dynamic scenes, and that they can be extracted in real-time without the need for special purpose hardware. These properties are particularly important in applications that require tracking people, and recently we have used 2-D blob tracking for real-time whole-body human interfaces [12, 5] and real-time recognition of American Sign Language hand gestures [11].

Our success at 2-D tracking motivates this paper's investigation into recovering useful 3-D geometry from these features. We begin by addressing the basic mathematical problem of estimating 3-D geometry from blob correspondences in displaced cameras. The relevant unknown 3-D geometry includes the shapes and motion of 3-D objects and the relative orientation of the cameras. The observations consist of the corresponding 2-D blobs, which can in general be derived from any signal-based similarity metric; in our experiments we use chrominant spectral similarity.

We will then present experimental results on two sub-problems required for human motion interpretation, namely self-calibrating the stereo rig, and then using it to reconstruct 3-D hand/head shape and motion. All of the experimental apparatus described here is real-time, at 20 to 30 frames per second, and runs on a pair of SGI Indy workstations without any special-purpose hardware.

## 2. Background: Blob features

The notion of grouping atomic parts of a scene together to form blob-like entities based on proximity and visual appearance is a natural one, and has been of interest to visual scientists since the Gestalt psychologists studied grouping criteria early in this century [2].

In modern computer vision processing we seek to group pixels of images together and to "segment" images based on visual coherence, but the "features" obtained from such efforts are usually taken to be the boundaries, or contours, of these regions rather than the regions themselves. In very complex scenes, such as those containing people or natural objects, contour features often prove unreliable and difficult to find and use.

The blob representation that we use was developed by Kauth *et al* and Pentland [7, 4], for application to multispectral satillite (MSS) imagery. In this method feature vectors at each pixel are formed by adding $(x, y)$ spatial coordinates to the spectral components of the imagery. These are then clustered so that color and spatial similarity combine to form coherent connected regions or "blobs". Essentially the same technique has been used recently in Wren *et al* [12] for real-time tracking of people in color video. The spatial coordinates are combined with color and brightness channels to form a five-element feature vector at each point $(x, y, Y, U, V)$. These are clustered into blobs which drive a "connected-blob" representation of the person. A broadly similar shape-color model has also been investigated by Schuster [10], and has been used to achieve fast tracking of human hands in a cluttered environment.

# 3. Estimating 3-D blobs

## 3.1 Blobs: moment-based shape model

We can represent shapes in both 2-D and 3-D by their low-order moments. Clusters of 2-D points have 2-D spatial means and covariance matrices, which we shall denote $\bar{q}$ and $C_q$, while 3-D shapes have 3-D spatial means and covariance matrices, denoted $\bar{p}$ and $C_p$.

The blob statistics can be interpreted as representing uniform, Gaussian, or some other second-order distribution of occupancy. The distribution is not terribly important because the physical location and orientation are independent of distribution and are encoded in the mean and covariance. The scale of the shape parameters with respect to the variance will vary across distributions, but the relative shape will remain the same. A Gaussian interpretation is chosen, therefore, for computational convenience.

Like other representations used in computer vision and signal analysis, including superquadrics, modal analysis, and eigen-representations, blobs represent the global aspects of the shape and can be augmented with further moments to attain more detail if the data supports it. The reduction of degrees of freedom from individual pixels to blob parameters is a form of regularization which allows the ill-conditioned problem to be solved in a principled and stable way.

For both 2-D and 3-D blobs, there is a useful physical interpretation of the blob parameters. The mean represents the geometric center of the blob area (2-D) or volume (3-D). The covariance, being symmetric, can be diagonalized via an eigenvalue decomposition

$$C = \Phi L \Phi^T \tag{1}$$

where $\Phi$ is orthonormal and $L$ is diagonal.

The diagonal $L$ matrix represents the size of the blob along independent orthogonal object-centered axes and $\Phi$ is a rotation matrix that brings this object-centered basis in alignment with the coordinate basis of $C$.

This decomposition and physical interpretation is important for estimation, because the shape $L$ is constant (or slowly varying) while the rotation $\Phi$ is dynamic. The parameters must be separated so they can be treated appropriately.

## 3.2 Parameterization of observation and state

To estimate 3-D geometry from 2-D images, we consider the nonlinear forward model and try to solve the inverse problem. That is, we consider the function

$$y = f(x) \tag{2}$$

where $y$ consists of 2-D observations and $x$ consists of the 3-D state. We observe $y$ and try to recover $x$.

The observation vector $y$ consists of a $(\bar{q}, C_q)$ pair for each observation:

$$y = \{(\bar{q}, C_q)_{t,k}\}, t = 1 \ldots M, k = 1 \ldots N \tag{3}$$

where $t$ is the frame index and $k$ is the blob index.

The covariance matrix $C_q$ has three free parameters, thus any pertubation $\delta y$ is of dimension $5MN$ where $M$ is the number of blobs and $N$ is the number of images being observed ($N = 2$ for stereo, e.g). Thus, the corresponding observation perturbation is

$$\begin{aligned} \delta y \;\; = & \;\; \{(\delta\bar{q}_u, \delta\bar{q}_v, \delta\sigma_u^2, \delta\sigma_v^2, \delta\sigma_{uv})_{t,k}\} \\ & t = 1 \ldots M, k = 1 \ldots N \end{aligned} \tag{4}$$

The state vector $x$ consists of the 3-D blob parameters $(\bar{p}, C_p)$ and the 3-D transformation $(T, R, \beta)$ to each camera, where $T$ is a 3-D translation vector, $R$ is a 3-D rotation (unit quaternion), and $\beta$ is inverse effective focal length $(1/f)$. In order to facilitate the physical interpretation, the 3-D covariance is decomposed as in Equation 1. Thus, for $M$ blobs and $N$ cameras, $x$ is

$$x = \{(\bar{p}, \Phi, L)_k, (T, R, \beta)_t\}, t = 1 \ldots M, k = 1 \ldots N \tag{5}$$

and the corresponding state perturbation is

$$\begin{aligned}\delta x &= \{(\delta\bar{p}_x, \delta\bar{p}_y, \delta\bar{p}_z, \delta\omega_x, \delta\omega_y, \delta\omega_z, \delta l_x, \delta l_y, \delta l_z)_k, \\ &\quad (\delta t_x, \delta t_y, \delta t_z, \delta\omega_x, \delta\omega_y, \delta\omega_z, \delta\beta)_t\}, \\ &\quad t = 1 \ldots M, k = 1 \ldots N \end{aligned} \tag{6}$$

where the $\delta\omega$ terms are angular perturbations on the respective rotations and the $\delta l$ terms are perturbations on the elements of the diagonal matrix $L$.

### 3.3 Forward process model

The forward model begins with the perspective projection equation:

$$q = c(p) \tag{7}$$

$$\begin{pmatrix} q_u \\ q_v \end{pmatrix} = \frac{1}{1 + p_z\beta} \begin{pmatrix} p_x \\ p_y \end{pmatrix} \tag{8}$$

where $p$ is a 3-D point, $q$ is the corresponding image point, the origin is at the center of the image plane, and the center of projection is at coordinates $(0, 0, -1/\beta)$.

Three-dimensional Gaussian distributions do not perspectively project exactly to 2-D Gaussian distributions, but if we have a rough estimate of the 3-D mean, we can use the linearized projection equation which does project 3-D Gaussians to 2-D Gaussians. It is easy to get a good initial value $p_0$ for the 3-D mean from the 2-D observations $\{\bar{q}_i\}$ to facilitate this linearization; this is covered in the next section.

(Possible drawbacks to using the linearized projection equation are that it will introduce a systematic modeling error in the results and that it requires iterative re-computation of $p_0$. However, the modeling error is small and most applications, particularly the human interface applications we are interested in, do not need higher precision. Our experimental results and applications developed using our system support this position. With regards to calculating $p_0$, it is an extremely cheap computation that we currently solve using a linear technique.)

Here we assume we have computed $p_0$ and linearize the projection equation

$$q = c(p_0) + J_c(x)(p - p_0) + \cdots \tag{9}$$

where $J_c$ is the Jacobian matrix of partial derivatives $(\partial c/\partial\delta p)$.

With the linearized system the 3-D mean projects to the 2-D mean, thus we have

$$\bar{q} = c(\bar{p}) \tag{10}$$

To get the covariance observations, define $\delta q = q - \bar{q}$ and $\delta p = p - \bar{p}$. Since $\bar{q}$ is $c(\bar{p})$ we have the linear relationship

$$\delta q = J_c \delta p \tag{11}$$

where now $\delta q$ describes a zero-mean Gaussian distribution with covariance $C_q$ and $\delta p$ describes a zero-mean Gaussian distribution with covariance $C_p$.

A Gaussian distribution with covariance $C_p$ undergoing a linear transformation $J_c$ will have covariance $J_c C_p J_c^T$. The Jacobian $J_c$ is a function of the state, $J_c(x)$, as is $C_p(x)$, thus we can write $C_q(x)$ as a composite function of the state, completing the specification of the forward model $y = f(x)$.

For the numerical solution, we shall also require the Jacobian of the forward model, $J(x) = [\partial f(x)/\partial\delta x]$, which can be determined analytically and evaluated numerically when needed.

### 3.4 Initialization and estimation

For performing the inversion of this nonlinear system, we need an initial state value at which to start. From stereo, we can recover an excellent starting state using linear techniques.

The initial 3-D position is obtained via linear least squares from the 2-D means (the closest approach of the two image rays). The initial rotation is taken as a rotation about the $z$-axis only, in concordance with the image-plane rotation of one of the 2-D blob observations, say the left image. Finally, the initial shape is matched in the $x$–$y$ plane to that of the left image (remember the initial 3-D blob is only rotated in $z$) and the $z$ size is chosen to be the smaller of the two principal dimensions of the $x$–$y$ plane.

Conceptually, then, we have back-projected the 2-D blob from the left image into a plane parallel to the left image at a depth determined by the correspondence of the 2-D blob means from the left and right images. We have then given the initial shape based on one projection and a heuristic notion of regularity.

The nonlinear estimation can now proceed in many ways. We have a forward model, $y = f(x)$, so we can develop a cost function based on the error $e(x, y) = y - f(x)$. We also have the Jacobian $J(x)$ analytically, so we have available to us many forms of nonlinear estimation techniques [8, 9, 3].

We use a form of the Levenberg-Marquardt (LM) algorithm for the static self-calibration problem and a form of the extended Kalman filter (EKF) for the dynamic problem of tracking motion and shape. We have also experimented with combining the iterative properties of LM with the recursive modeling of EKF to perform the two tasks together, but these results are still forthcoming.

## 4. Experimental analysis

The motivating application for us has been tracking people in 3-D in uncontrived environments, such as an office environment. We have developed a simple blob tracker based on the ideas presented in [12] that gives us reliable 2-D blobs of a person's face and hands in real-time. This tracker operating independently on two cameras in a wide-baseline configuration gives us correspondences that we can use to self-calibrate the stereo system and track people's movement and gestures at the same time. We present here some experimental results taken from our real-time system, which operates at 20–30 Hz using two SGI Indy workstations.

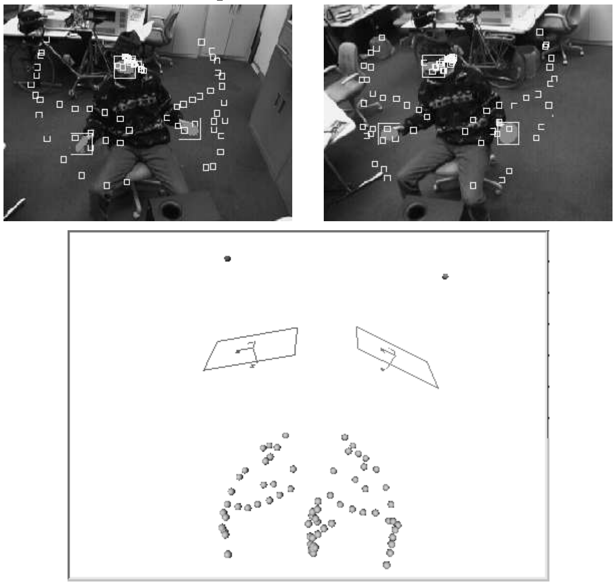
### 4.1 Self-calibration

When a person first enters the space, the stereo calibration is obtained by collecting a set of three blob correspondences (face, left hand, right hand) over a number of frames (50–100 total correspondences) and computing the stereo calibration using the LM estimator on the batch of correspondences. Figure 1 shows a typical data set at system startup that calibrates the stereo rig.

The stereo pair shows the first image with overlayed blobs and large white boxes marking the current feature locations, and small white boxes representing the subsequent feature tracks.

The calibration points are collected in a time span of roughly 5 sec and the estimation requires less than 2 sec. In this case, the subject waved his arms up and down to generate data and the system quickly converged to the state shown in the bottom portion of Figure 1, which is a roughly overhead view showing the location of the cameras (COP and virtual image plane for each) and the 3-D trajectories of the hands and head.

Stereo sequence for self-calibration



3-D view of self-calibration sequence

**Figure 1.** The blob representation can be used to facilitate stereo self-calibration. Here we illustrate the self-calibration of a stereo rig in real time, simply from watching a person moving. The stereo pair shows the feature tracks on the person. The 3-D view shows a roughly overhead view of the space including the recovered cameras and the 3-D feature tracks. RMS residual error is 1.5 pixels; RMS 3-D errors are on the order of 2.25cm.

#### 4.1.1 Error Analysis

There is no absolute 3-D ground truth for self-calibration, but residual error can be analyzed in the image plane and relative error can be evaluated in 3-D.

The residual error approach consists of using the recovered camera parameters and blob locations to re-synthesize the feature locations and compare to the actual measurements. Over dozens of calibration sequences, the RMS image plane residual is 1.5 pixels. The image size is (320,256) and the major blob axis diameters range from 20 to 35 pixels. The sources of residual errors include both measurement error (noise) and modeling error.

Relative 3-D error is evaluated after self-calibration by using the right hand as a 3-D pointer and traversing a trajectory of known shape and dimensions. In our case, a user moved his hand linearly along the edge of a table back and forth completing 4 cycles. The recovered 3-D trajectory is depicted in Figure 2, in which the three coordinates are plotted against frame number. The digitized hand locations are clustered along a 3-D line segment with known length (120cm). A line segment is fit to the cluster using 3-D regression and the coordinates are scaled to the known length.

The resulting RMS error from this analysis is 2.25cm. The relative error of reconstruction of hand position over this trajectory is therefore 1.8 percent. The sources of error include not only noise and modeling error, but also hand movement error, since the trajectory followed by the person was not exactly linear and the hand shape changes. Thus only a fraction of this relative error should be counted as computational error.

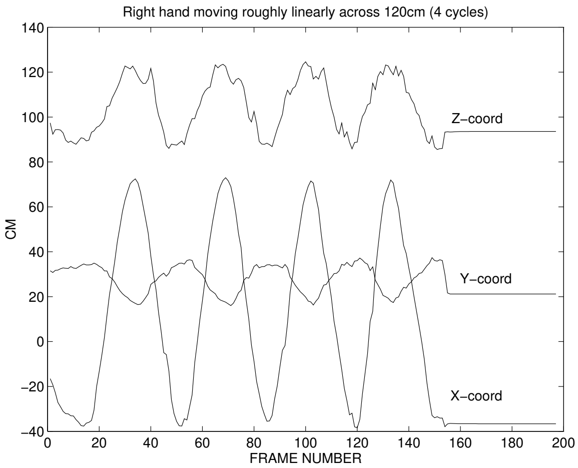



**Figure 2.** Translation of the right hand back and forth along a linear trajectory after self-calibration. Reconstruction of hand position has an RMS error of 2.25cm, resulting in a relative error of 1.8% over this trajectory.

### 4.2 Person-tracking and shape recovery

The self-calibration can in principle take place simultaneously with shape and motion estimation, but for the purposes of performing our experiments they have been implemented separately. This makes it easier to isolate the sources of error for each component of the estimation.

Thus, in this section we are concerned with the steady-state characteristics of shape and motion estimation after self-calibration has converged. Again, since absolute motion parameters are impossible to know, we evaluate relative error by formulating some hand motions with known qualitative parameters.

#### 4.2.1 Error Analysis

The first test case consists of a linear motion where the user tries to keep his hand shape and orientation constant while sliding his hand along the straight edge of a rigid box. This is similar to the previous test case for self-calibration except that rotation and shape are measured.

Figure 3 shows a stereo pair and 3-D blob estimates for one frame of the sequence. Analysis on the translation was performed by fitting the 3-D location estimates to a line and computing the RMS error of translation, which was 1.5cm, resulting in a similar relative error metric to the self-calibration analysis. Analysis on the rotation and shape were performed by computing the mean values and RMS errors, which were about 2 degrees and 5% relative error respectively. Sources of error include measurement noise and modeling error. The comparatively high relative error in shape is probably due to the fact that the shape parameters are the least well-conditioned parameters in the state.

A second test case addresses the dynamic behavior of the rotation parameters. In this case, the user was told to rotate his hand from pointing directly forward to pointing roughly leftward and to cycle through this movement four times. The recovered angle of rotation is depicted in Figure 4, where a qualitative view of the estimation noise can be seen. Since the absolute ground truth is not known, the only way to quantitatively evaluate the error is to assume something about the trajectory.

In this case, we assume the actual trajectory was smooth, so we synthesize a set of error vectors by smoothing the trajectory and taking the difference between the actual and the smoothed trajectories. We use a 7-tap approximate Gaussian low-pass filter to smooth the trajectory. The RMS of the synthesized error is 1.98 deg, which results in a 2.2 percent relative error. The source of this error measure is primarily only the high-frequency components of measurement noise. Additional error may be present and may arise from low-frequency measurement errors or modeling errors.

## 5. Conclusion

We have developed an estimation technique for recovering 3-D object shape and motion and multiple-camera geometry from 2-D blob features. The 3-D objects and 2-D features are both represented using moment-based physical models called blobs. Nonlinear optimization techniques are used for estimation; the Levenberg-Marquardt technique is used for static parameters and the extended Kalman filter is used for dynamic estimation.

Experimental results verify that we can obtain good quantitative 3-D physical descriptions from these coarse 2-D image observations of people. We have experimentally demonstrated that this method can be used to self-calibrate stereo cameras from watching people move and subsequently to determine the location, orientation, and shape of parts of a person to an accuracy of a 2 cm, 2 degrees and a few percent, respectively (RMS errors).

Perhaps the most important performance evaluation, however, is that we have been able to build a person-tracking system using this technique which has run reliably for dozens of hours, with dozens of different subjects, in several different locations, and in real time (20–30 fps) using only standard workstations.

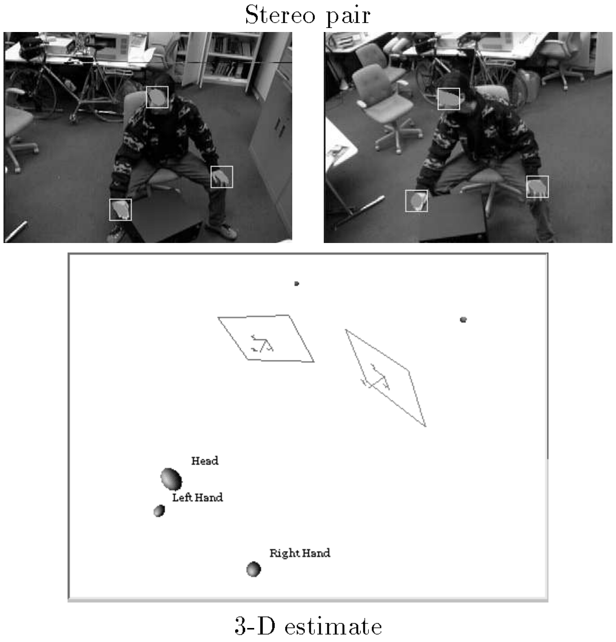



**Figure 3.** Real-time estimation of position, orientation, and shape of moving human head and hands. We find RMS errors of 1.5cm, 2 degrees, and 5% on translation, rotation, and shape, respectively along a linear 3-D trajectory.

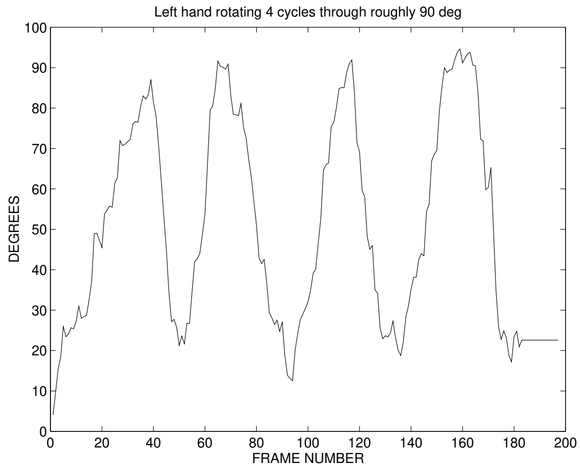



**Figure 4.** Rotation of the left hand back and forth through roughly 90 deg. An analysis of the jitter in the angular signal results in measures of 2 deg RMS error, or roughly 2.2 percent relative error.

The key to the robust, real-time performance is that the 2-D blob features on which the estimation relies can be reliably and efficiently extracted and matched in a bottom-up fashion.

We feel that use of this type of feature is a significant departure from the traditional notions of image features (e.g., points, lines) and image cues (e.g., motion fields, shading), and can lead to a basis for practical 3-D vision systems in application domains where traditional approaches have not had a great deal of success. Although the blob models provide only rigid motion and coarse shape information, they are fast and extremely reliable; thus futher precision and higher levels of details, if desired, can be safely bootstrapped from this level of representation, potentially leading to a powerful "coarse-to-fine" or "subsumption" approach to 3-D shape and motion analysis.

## References

[1] A. F. Bobick and R. C. Bolles. The representation space paradigm of concurrent evolving object descriptions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):146–156, February 1992.

[2] W. D. Ellis. *A Source Book of Gestalt Psychology.* Harcourt, Brace, and Co., 1939.

[3] A. Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, MA, 1974.

[4] R. J. Kauth, A. P. Pentland, and G. S. Thomas. Blob: An unsupervised clustering approach to spatial preprocessing of mss imagery. In *11th Int'l Symposium on Remote Sensing of the Environment*, Ann Arbor, MI, April 1977.

[5] P. Maes, B. Blumberg, T. Darrell, and A. Pentland. The ALIVE system: Full-body interaction with autonomous agents. In *Proceedings of Computer Animation 95.* IEEE Press, April 1995.

[6] D. Marr. *Vision.* Freeman, 1982.

[7] A. Pentland. Classification by clustering. In *Proceedings of the Symposium on Machine Processing of Remotely Sensed Data.* IEEE, IEEE Computer Society Press, June 1976.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: the art of scientific computing.* Cambridge University Press, Cambridge, U.K., second edition, 1992.

[9] L. E. Scales. *Introduction to Non-linear optimization.* Springer-Verlag, New York, 1985.

[10] R. Schuster. Color object tracking with adaptive modeling. In *Workshop on Visual Behaviors*, pages 91–96, Seattle, WA, June 1994. International Association for Pattern Recognition, IEEE Computer Society Press.

[11] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition 1995*, June 1995. (Zurich, Switzerland).

[12] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. *Pfinder*: Real-time tracking of the human body. In *Photonics East, SPIE Proceedings Vol. 2615*, Bellingham, WA, 1995. SPIE.