$$3\sum_{i=2}^{9}\left(\frac{\boldsymbol{\alpha}_i^{\#t}\mathbf{F}\boldsymbol{\alpha}_1^{\#}}{\lambda_1^{\#}-\lambda_i^{\#}}\right)^2 \cong \|\text{perturbed version of } \mathbf{R}^{\#}-\mathbf{R}^{\#}\|_2^2 \geq 6-2\sqrt{3}\text{tr}(\mathbf{S}_1). \quad (7)$$

From $\lambda_9 \geq \cdots \geq \lambda_2$, and (6) and (7), we have the following result from which LB_2 can be obtained:

$$\Xi(\mathbf{R}^{\#},\mathbf{F}) \geq 3\lambda_1+3\lambda_2\sum_{i=2}^{9}\left(\frac{\boldsymbol{\alpha}_i^{\#t}\mathbf{F}\boldsymbol{\alpha}_1^{\#}}{\lambda_1^{\#}-\lambda_i^{\#}}\right)^2 \geq 3\lambda_1+(6-2\sqrt{3}\text{tr}(\mathbf{S}_1))\lambda_2.$$

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their valuable comments and help. This work was supported by the Republic of China National Science Council under Grant NSC-86-2213-E-009-114.

REFERENCES

[1] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision*, vol. 2. Reading, Mass.: Addision Welsey 1993.
[2] T.S. Huang and A.N. Netravali, "Motion and Structure from Feature Correspondences: A Review," *Proc. the IEEE*, vol. 82, no. 2, pp. 252-268, 1994.
[3] Y. Liu, T.S. Huang, and O.D. Faugeras, "Determination of Camera Location from 2D to 3D Line and Point Correspondences," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 28-37, Jan. 1990.
[4] M. Dhome, M. Richetin, J.T. Lapreste, and G. Rives, "Determination of the Attitude of 3D Objects from a Single Perspective View," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 12, pp. 1,265-1,278, Dec. 1989.
[5] S.Y. Chen and W.H. Tsai, "A Systematic Approach to Analytic Determination of Camera Parameters by Line Features," *Pattern Recognition*, vol. 23, no. 8, pp. 859-877, 1990.
[6] R. Kumar and A.R. Hanson, "Robust Methods for Estimating Pose and a Senitivity Analysis," *Computer Vision and Graphic Image Processing: Image Understanding*, vol. 60, no. 3, pp. 313-342, 1994.
[7] T.Q. Phong, R. Horaud, A. Yassine, and P.D. Tao, "Object Pose from 2D to 3D Point and Line Correspondences," *Int'l J. Computer Vision*, vol. 15, pp. 225-243, 1995.
[8] C.N. Lee and R.M. Haralick, "Statistical Estimation for Exterior Orientation from Line to Line Correspondences," *Image and Vision Computing*, vol. 14, pp. 379-388, 1996.
[9] K. Cho, P. Meer, and J. Cabrera,, "Performance Assessment through Bootstrap," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 11, pp. 1,185-1,198, Nov. 1997.
[10] S. Yi, R.M. Haralick, and L.G. Shapiro, "Error Propagation in Machine Vision," *Machine Vision and Applications*, vol. 7, pp. 93-114, 1994.
[11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second ed. Boston, Mass.: Academic Press, 1990.
[12] J.R. Taylor, *An Introduction to Error Analysis*. Mill Valley, Oxford Univ. Press, 1982.
[13] B.K.P. Horn, "Relative Orientation," *Int'l J. Computer Vision*, vol. 4, pp. 59-78, 1990.
[14] R.A. Horn and C.R. Johnson, *Matrix Analysis*. New York, NY: Cambridge Univ. Press, 1985.
[15] K. Kanatani, *Geometric Computation for Machine Vision*. New York, NY: Oxford Univ. Press, 1993.
[16] G.G. Roussas, *A Course in Mathematical Statistics*. second ed. New York, NY: Academic Press, 1997.

Tracking Human Motion in Structured Environments Using a Distributed-Camera System

Q. Cai and J.K. Aggarwal, *Fellow, IEEE*

Abstract—This paper presents a comprehensive framework for tracking coarse human models from sequences of synchronized monocular grayscale images in multiple camera coordinates. It demonstrates the feasibility of an end-to-end person tracking system using a unique combination of motion analysis on 3D geometry in different camera coordinates and other existing techniques in motion detection, segmentation, and pattern recognition. The system starts with tracking from a single camera view. When the system predicts that the active camera will no longer have a good view of the subject of interest, tracking will be switched to another camera which provides a better view and requires the least switching to continue tracking. The nonrigidity of the human body is addressed by matching points of the middle line of the human image, spatially and temporally, using Bayesian classification schemes. Multivariate normal distributions are employed to model class-conditional densities of the features for tracking, such as location, intensity, and geometric features. Limited degrees of occlusion are tolerated within the system. Experimental results using a prototype system are presented and the performance of the algorithm is evaluated to demonstrate its feasibility for real time applications.

Index Terms—Tracking, human modeling, motion estimation, multiple perspectives, Bayesian classification, end-to-end vision systems.

1 INTRODUCTION

TRACKING human motion is of interest in numerous applications such as surveillance, analysis of athletic performance, and content-based management of digital image databases. Recently, growing interest has concentrated upon tracking humans using distributed monocular camera systems to extend the limited viewing angle of a single fixed camera [1], [2], [3]. In such a setup, the cameras are arranged to cover a monitored area with overlapping vision fields to ensure a smooth switching among cameras during tracking. We present a comprehensive framework for automatically tracking coarse human models across multiple camera coordinates and demonstrate the feasibility of an end-to-end person tracking system using a unique combination of motion analysis on 3D geometry in different camera coordinates with existing techniques in motion detection, segmentation, and pattern recognition. The nonrigidity of the human body is addressed by matching points of the middle line of the human image, spatially and temporally, using Bayesian classification schemes. The key to successful tracking in the proposed work relies on our unique method of 3D motion prediction and estimation from different perspectives. Experimental studies using a three-camera prototype system show its efficiency in computation and potential for real time applications.

The earliest work in this area is, perhaps, by Sato et al. [1]. They considered the moving human image as a combination of various blobs. All distributed cameras were calibrated in the world coordinate system, which corresponds to a CAD model of the

- *Q. Cai is with the Consulting Group, Realnetworks, Inc., 2601 Elliott Ave., Seattle, WA 98121. E-mail: qcai@real.com.*
- *J.K. Aggarwal is with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712-1084. E-mail: aggarwaljk@mail.utexas.edu.*

Manuscript received 16 Oct. 1997; revised 3 Sept. 1999.
Recommended for acceptance by R. Szeliski.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107749.

indoor environment. The blobs of body parts were matched through image sequences using the area, average brightness, and rough 3D position in the world coordinates. Kelly et al. [2] adopted a similar strategy [1] to construct a 3D environmental model using the voxel feature. The depth information contained in the voxel is obtained using height estimation. Moving humans were tracked as a group of these voxels from the "best" angle of the viewing system. Neither of these methods considered the particular body structure and shape characteristics of a human being. In addition, both need to model the environment in 3D and establish a world coordinate. They are computationally expensive and do not adapt to changes in dynamic environments. In our work, only neighboring cameras are calibrated to their relative coordinates and background images are updated periodically to capture the changes in the environment. Based on studies on human geometric structures, we distinguish moving human figures from other nonhuman objects by modeling the human body. Matching the subject image between consecutive frames involves motion estimation in a spatial-temporal domain under a Bayesian classification scheme.

Tracking is done from a single camera view until the system predicts that the active camera soon will no longer have a good view of the subject of interest. Tracking then switches to the camera that will provide a better view and require the least switching to continue tracking. Thus, the tracking paradigm consists of three basic modules: Single View Tracking (SVT), Multiple View Transition Tracking (MVTT), and Automatic Camera Switching (ACS).

2 Single View Tracking

Tracking from a single view includes two major components: preprocessing and feature correspondence between consecutive frames. Three stages of preprocessing are performed:

1. Segmenting the moving objects from the still background,
2. Distinguishing human subjects from other segmented nonbackground objects, and
3. Extracting features from the segmented human subjects.

Feature correspondence is established by applying a Bayesian classifier to locate the most likely match of the subject image in the next frame. The feature vector consists of location, intensity, and geometric information. Multivariate Gaussian models are formulated to parameterize the class conditional probability density of the feature vector. Thus, tracking is reduced to finding the minimum sum of the corresponding *Mahalonobis* distances of the feature given the estimated feature parameters.

2.1 Preprocessing

Preprocessing is critical to the success of high-level processing stages. If a moving object is missed at the preprocessing stage, the system will be unable to track this particular object at later stages. The major task of preprocessing is to segment human images from the rest of the image objects. To the best of our knowledge, there are still no satisfying and robust general solutions. Here, we apply efficient standard motion detection and segmentation techniques to take the advantage of the fact that the viewing system is still. More robust and complicated segmentation schemes could be applied if computational cost is not a consideration. The key to the proposed motion segmentation is to dynamically recover the background by grouping regions of still pixels in time. Then, we detected moving blobs by differencing and focused on the upper half body of the blobs using a coarse 2D human model. This procedure is followed by human segmentation, where moment invariants are used as the shape feature for distinguishing between human and nonhuman moving objects based on Principal Component Analysis (PCA). More details are found in [4], [5].

Due to their robustness for matching in different views, N points belonging to the middle line of the upper body are selected and aggregated as the feature to track. The line segment is extracted by finding the middle points of the blobs. Using multiple feature points instead of a single point [6] makes matching the subject image more reliable. We have elected to use six points based on the trade-off between the need to use fewer points to reduce computation cost and the need to use more points due to the nonrigidity of moving human figures. To ensure the robustness of the feature matching, we incorporate three types of features: location, intensity, and geometry. The location feature is defined as the horizontal and vertical position of the feature points: $\mathbf{X}_t = [(u_{1t}, v_{1t}), (u_{2t}, v_{2t}), \ldots, (u_{Nt}, v_{Nt})]^T$, where t is the time index. We define the intensity feature as $\mathbf{Y}_t = [y_{1t}, y_{2t}, \ldots, y_{Nt}]^T$, in which y_{mt} is the average intensity of the neighborhood of the mth feature points. Another type of feature is the image height ratio between consecutive frames (the height of a candidate image in the current frame divided by the subject height in the previous frame) as the geometric feature (g_t), where the image height is computed as the height of the upper body using a coarse 2D geometric human model at the segmentation stage. This feature is essential for tracking in narrow corridor scenes where the location and intensity features most likely fail.

2.2 Feature Correspondence

Tracking a subject between adjacent frames can be achieved by finding the closest match of features in the next frame based on constraints such as continuous position, instantaneous velocity, similar intensity, etc. We apply a Bayesian classifier to locate the most likely match of the subject image in the next frame. For simplicity of computation without loss of generality, we assume that a prior probability function $P(\Theta)$ is uniformly distributed, where Θ is the feature parameters of the subject to track. In a multivariate Gaussian model, it represents the mean and covariance of the feature vector $\mathbf{Z}_t = [\mathbf{X}_t, \mathbf{Y}_t, g_t]$, where $\mathbf{X}_t$, $\mathbf{Y}_t$, and g_t are assumed to be independent of each other since they are different types of features. So, we define

$$p(\mathbf{Z}_t|\Theta) = p_x^{w_x}(\mathbf{X}_t|\Theta_x)p_y^{w_y}(\mathbf{Y}_t|\Theta_y)p_g^{w_g}(g_t|\theta_g),$$

where w_x, w_y, and w_g are the the weights associated with $p_x(.)$, $p_y(.)$, and $p_g(.)$. Based on Bayes theory, the closest match is found by searching the minimum of $D_t = -\log p(\mathbf{Z}_t|\Theta)$. The weights for each feature are computed based on the $1/w_x : 1/w_y : 1/w_g = [-\log p_x(\mathbf{X}_t|\Theta_x)] : [-\log p_y(\mathbf{Y}_t|\Theta_y)] : [-\log p_g(g_t|\Theta_g)]$ and $w_x + w_y + w_g = 1$ during training. We assume that $p_x(.)$, $p_y(.)$, and $p_g(.)$ are normally distributed to reduce the computational cost.

Since the subject of interest has a nonrigid form, we assume that one feature point is independent of another. Under such assumptions, the mean vector of $p_x(.)$ $\mu_x = \bar{\mathbf{X}}_t$ and the covariance Σ_x is a diagonal matrix with the mth component of $\sigma_{x,m}^2$. Therefore, we have

$$p_x(\mathbf{X}_t|\Theta_x) = \prod_{m=1}^{N} \frac{1}{2\pi\sigma_{x,m}^2} \exp\left[-\frac{(u_{mt} - \bar{u}_{mt})^2 + (v_{mt} - \bar{v}_{mt})^2}{2\sigma_{x,m}^2}\right]. \quad (1)$$

The estimation of $(\bar{u}_{mt}, \bar{v}_{mt})$ is computed using perspective projection and the assumption that velocity direction for three consecutive frames is unchanged [5],

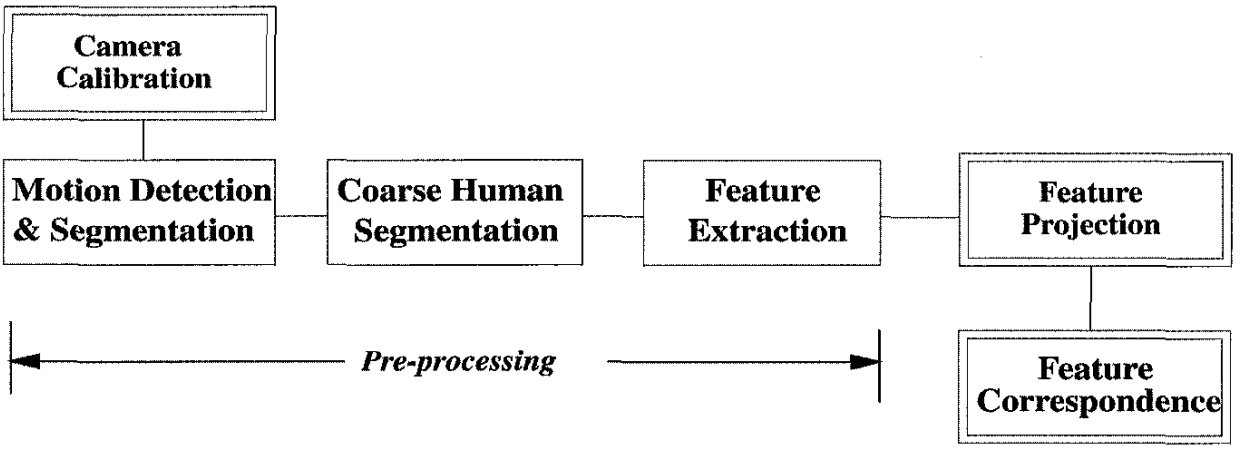

Fig. 1. The basic procedure of transition tracking.

$$\bar{u}_{mt} = \frac{(r_t r_{t-1} - 1)\bar{u}_{m(t-1)} - (r_t - 1)\bar{u}_{m(t-2)}}{r_t r_{t-1} - r_t},$$
$$\bar{v}_{mt} = \frac{(r_t r_{t-1} - 1)\bar{v}_{m(t-1)} - (r_t - 1)\bar{v}_{m(t-2)}}{r_t r_{t-1} - r_t},$$

where $r_t = 1/g_t$, and $r_0 = 1$. We define $\sigma_{x,m} = \lambda_x h_x$ so that $\sigma_{x,m}$ is proportional to the scale of the image height h_x, where λ_x is only a scaling factor in order to obtain a universal scaling for the *Mahalonobis* distances for different types of features. The definition and estimation for intensity $p_y(.)$ and geometric features are similar and are given in [5].

Finally, we have

$$\begin{aligned} D_t &= w_x \sum_{m=1}^{N} \frac{(u_{mt} - \bar{u}_{mt})^2 + (v_{mt} - \bar{v}_{mt})^2}{\sigma_{x,m}^2} \\ &\quad + w_y \sum_{m=1}^{N} \frac{(y_{mt} - \bar{y}_{mt})^2}{\sigma_{y,m}^2} + w_g \frac{(g_t - \bar{g}_t)^2}{\sigma_g^2} \\ &= w_x D_{x,t} + w_y D_{y,t} + w_g D_{g,t}. \end{aligned} \tag{2}$$

These $D_{x,t}$, $D_{y,t}$, and $D_{g,t}$ are *Mahalonobis* distances for each individual feature. The most likely match should satisfy two conditions: D_t must be 1) less than certain threshold T, and 2) the minimum value among the candidates. Although this threshold is currently preset, its value could be adapted according to different tracking environments.

In the above paradigm, if the subject of interest is occluded by another subject, the system might select the occluding subject as the best match, even though the intensity or geometry features might not agree. If we do not "memorize" the correct features in the previous frame, the target might be switched after occlusion. In such cases, we use estimated features instead of the ones computed directly from the current frame. Details of the computation are addressed in [5].

3 Multiple View Transition Tracking

In our system, tracking continues in the single view (SVT) mode until the active camera no longer has a good view of the subject of interest, when tracking switches to a video stream captured from another nearby camera. At that point, the system enters the mode of Multiple View Transition Tracking (MVTT). Fig. 1 shows the overall diagram of the module. The double-framed rectangular boxes represent the processes which differ from SVT. In MVTT, the tracking feature in consecutive frames must be adjusted to the same spatial coordinates. Preprocessing starts with camera calibration, which measures the intrinsic and extrinsic parameters of the system cameras by using the methods in [7] and [8], respectively. These parameters are used to establish the relationships between various camera coordinates. Then, we go through the same procedure as in the preprocessing in the SVT module. The last step before feature correspondence is to project the location feature into the same camera coordinates.

We again apply multivariate Gaussian models to represent the class-conditional distributions of the feature $p(\mathbf{Z}_t|\Theta)$, including only location and intensity information, since there is no longer a valid criteria for estimating geometric features from different camera views without knowing the relative distances between the subject and the viewing cameras. However, feature correspondence using the location feature differs significantly.

3.1 Tracking Based on the Location Feature

Tracking across different perspectives in time is equivalent to matching feature points from I_t and J_{t+1}, where I_t is the frame imaged by camera C_i at time t and J_{t+1} is the frame imaged by camera C_j at time $t+1$. It involves both spatial and temporal motion estimation. Typical methods, like Kalman filtering, could be used in this case. To reduce computational cost, we apply a simpler prediction and estimation method instead. Two basic models are addressed: the class-conditional distribution for spatial matching $p_{x1}(\mathbf{X}_t|\Theta_{x1})$ and that for spatial-temporal matching, $p_{x2}(\mathbf{X}_t|\Theta_{x2})$.

3.1.1 Spatial Matching

Spatial matching is based on the correspondence between a 2D point and its corresponding epipolar line. To establish correspondence between frames imaged by camera C_i and camera C_j at time t, the multivariate Gaussian model for position is modified from (1) to

$$p_{x1}(\mathbf{X}_t|\Theta_{x1}) = \prod_{m=1}^{N} \frac{1}{2\pi\sigma_{x1,m}^2} \exp\left[-\frac{d_{mt}^2}{2\sigma_{x1,m}^2}\right],$$

where d_{mt} is the distance between the mth feature point (u_{mt}, v_{mt}) and its expected 2D epipolar line $a_{mt}x + b_{mt}y + c_{mt} = 0$, all in the view of C_j. The 2D epipolar line is projected from the point $(\bar{u}_{mt}, \bar{v}_{mt})$ in the view of C_i, and t is the time index. Since the distance between an image point $(x_0/z_0, y_0/z_0)$ and a 2D line $ax + by + c = 0$ is

$$\frac{|x_0/z_0 + by_0/(az_0) + c/a|}{\sqrt{1 + b^2/a^2}},$$

we define $\sigma_{x1,m}$ as $\sigma_{x1,m} = \lambda_{x1} 1/\sqrt{1 + b_{mt}^2/a_{mt}^2}$, with λ_{x1} again as a scaling factor.

3.1.2 Spatial-Temporal Matching

Spatial-temporal matching involves estimating the projection of a 3D point in the view of camera i at time t (denoted as $(\bar{u}_{it}, \bar{v}_{it})$), given $(\bar{u}_{i(t-1)}, \bar{v}_{i(t-1)})$, $(\bar{u}_{j(t-1)}, \bar{v}_{j(t-1)})$, and $(\bar{u}_{jt}, \bar{v}_{jt})$. Using the pinhole projection model, we have

$$\alpha_1 [\bar{u}_{i(t-1)}\ \bar{v}_{i(t-1)}\ f]^T = \alpha_2 R_{ij}\ [\bar{u}_{j(t-1)}\ \bar{v}_{j(t-1)}\ f]^T + T_{ij}$$

and

$$\beta_1 \alpha_1 [\bar{u}_{it}\ \bar{v}_{it}\ f]^T = \beta_2 \alpha_2 R_{ij} [\bar{u}_{jt}\ \bar{v}_{jt}\ f]^T + T_{ij},$$

where R_{ij} and T_{ij} is the rotational matrix and translational vector between the camera coordinate i and j, α_1 and α_2 are scaling factors, β_1 and β_2 are the depth ratio of the point at times $t-1$ and t, in C_i and C_j, which can be calculated using the height ratio of the subject images between adjacent frames. Finally, we arrive at:

$$\alpha = \frac{(\beta_1 - 1)f}{r_{31}U + r_{32}V + r_{33}(\beta_2 - 1)f}$$

$$\bar{u}_{it} = \frac{\alpha}{\beta_1}(r_{11}U + r_{12}V + r_{13}(\beta_2 - 1)f) + \frac{\bar{u}_{i(t-1)}}{\beta_1}$$

$$\bar{v}_{it} = \frac{\alpha}{\beta_1}(r_{21}U + r_{22}V + r_{23}(\beta_2 - 1)f) + \frac{\bar{v}_{i(t-1)}}{\beta_1},$$

where $\alpha = \alpha_2/\alpha_1$, $U = \beta_2 \bar{u}_{jt} - \bar{u}_{j(t-1)}$, $V = \beta_2 \bar{v}_{jt} - \bar{v}_{j(t-1)}$, and r_{kl} is the kth row and lth column element of R_{ij}. When occlusion is detected by thresholding, similar to the module of SVT, only y_{mt} is modified. More details could be found in [5].

4 Automatic Camera Switching

We choose to track the subject of interest in one video stream at one time instant to reduce the computational cost and automatically switch among cameras to keep the subject in view. Automatic camera switching (ACS) consists of two steps: prediction and optimal camera selection. The prediction process reports when camera switching is necessary, which may happen in three cases:

1. when the subject image appears to be moving out of the viewing boundaries of the current camera,
2. when the subject moves too far away, and
3. when the subject becomes occluded by another subject for more than two frames.

In these situations, switching to another camera may result in a more continuous or better view of the subject. The selection of "optimal" camera is considered in terms of three aspects:

1. The candidate camera must be able to image the subject in the future,
2. Spatial matching between different cameras is robust, and
3. The candidate camera will contain the subject image over the longest number of frames, given the subject's current position and velocity.

The third requirement minimizes the amount of camera switching during tracking.

4.1 Prediction

We address three types of prediction for the subject image: location prediction, height prediction, and tracking confidence measurement. Location prediction estimates the location of the subject image in the next frame and judges if it will be within the vision field of the current camera. Image height prediction is,, in a sense, an estimation of the subject's depth using the image's positions in previous frames. Tracking confidence is a measure of robust matching between consecutive frames. It could be lowered due to poor segmentation, occlusion, and ambiguity in the clothing and size of the subject images.

In each process, we assume constant velocity of the subject over three consecutive frames. This assumption is reasonable given the small time period for capturing three frames. The velocity information is refined at each step once the uncertainty of matching is resolved.

4.1.1 Location Prediction

Location prediction is based on the perspective projection and constant velocity [5]. Finally, we have $u_t = u_{t-1} + \triangle u/(2r_{t-1} - 1)$ and $v_t = v_{t-1} + \triangle v/(2r_{t-1} - 1)$ with

$$(\triangle u, \triangle v) = (u_{t-1} - u_{t-2}, v_{t-1} - v_{t-2}).$$

To initialize the prediction process, we assume that $r_{t\ 1} = 1$ and $\triangle u = \triangle v = 0$. If (u_t, v_t) is out of the viewing boundaries of the current camera, camera switching is immediate.

4.1.2 Image Height Prediction

Image height prediction uses the height of the upper body image as a coarse reflection of the subject's depth in the camera coordinate. Compared to width, the height of the subject image more truthfully reflects the distance between the subject and the active camera. For example, a person facing toward the viewing camera will be the same height as he turns 90 degrees away, but a different width. Using the definition of r_t, along with the constant velocity assumption, the height of the subject's upper body in the tth frame is

$$h_t = \frac{h_{t-1}}{r_t} = \frac{h_{t-1}}{2} + \frac{h_{t-1}}{2} \cdot \frac{1}{2r_{t-1} - 1}.$$

If h_t becomes too small, indicating that the subject is moving too far from the viewing camera, then immediate camera switching is necessary.

4.1.3 Tracking Confidence

Tracking confidence is derived from D_t since it is the key to finding the most likely match between two consecutive frames. Two types of confidence are considered: the absolute confidence, ACF_t, and the relative confidence, RCF_t, where t is the time index. ACF_t is defined as $ACF_t = T/D_t$ with T as a threshold addressed before. As D_t decreases, ACF_t increases proportionally, which agrees with the decision criterion that the less the *Mahalanobis* distance is, the more robust the match is. RCF_t is a measure of the relative tracking confidence among multiple candidates for matching. It is defined as $RCF_t = D_t(1)/D_t(0)$, with

$$D_t(0) \leq D_t(1) \leq \ldots \leq D_t(k) \ldots \leq D_t(K),$$

k as the index of subject candidates, and K as the number of subject candidates. A confident match should have both high ACF_t and RCF_t. If only one subject exists, ACF_t is the only quantitative measure for tracking quality. The overall confidence is defined as $CF_t = \min[ACF_t, RCF_t]$. A small CF_t may be caused by occlusion of subject images, poor segmentation, ambiguity between sizes and intensity values of subject images, etc. In such situations, changing the viewing angle of the camera may help to solve some of these problems. The definition of tracking confidence also applies to each individual feature, except that the threshold T has been changed to T_x, T_y, and T_g, respectively.

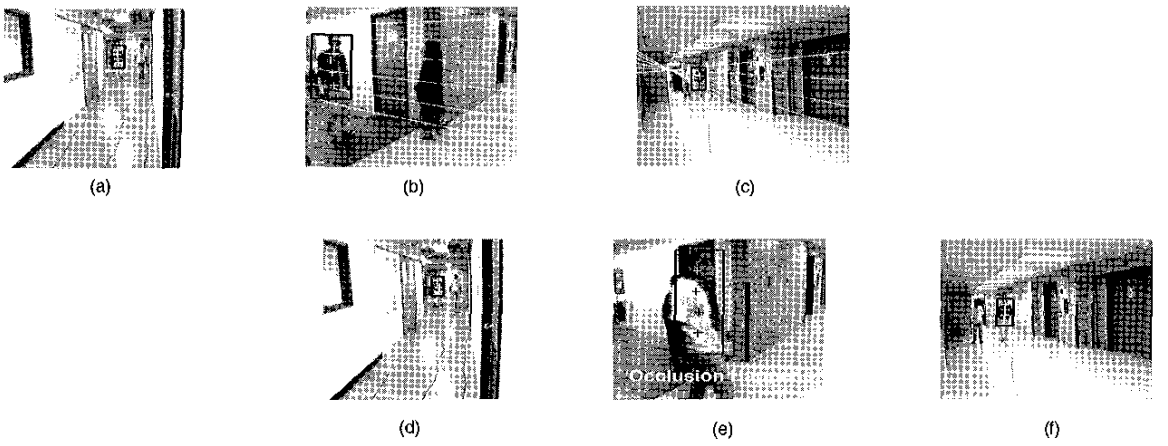

Fig. 2. Tracking a subject around an indoor corner: (a) $C_1, t = 1$, (b) $C_0, t = 2$, (c) $C_2, t = 3$, (d) $C_1, t = 2$, (e) $C_0, t = 3$, (f) $C_2, t = 4$.

4.2 Optimal Camera Selection

We select the optimal camera based on matching robustness and prediction of the subject image position given its current position and velocity. The process of selecting the optimal camera involves two steps, matching evaluation and frame number calculation.

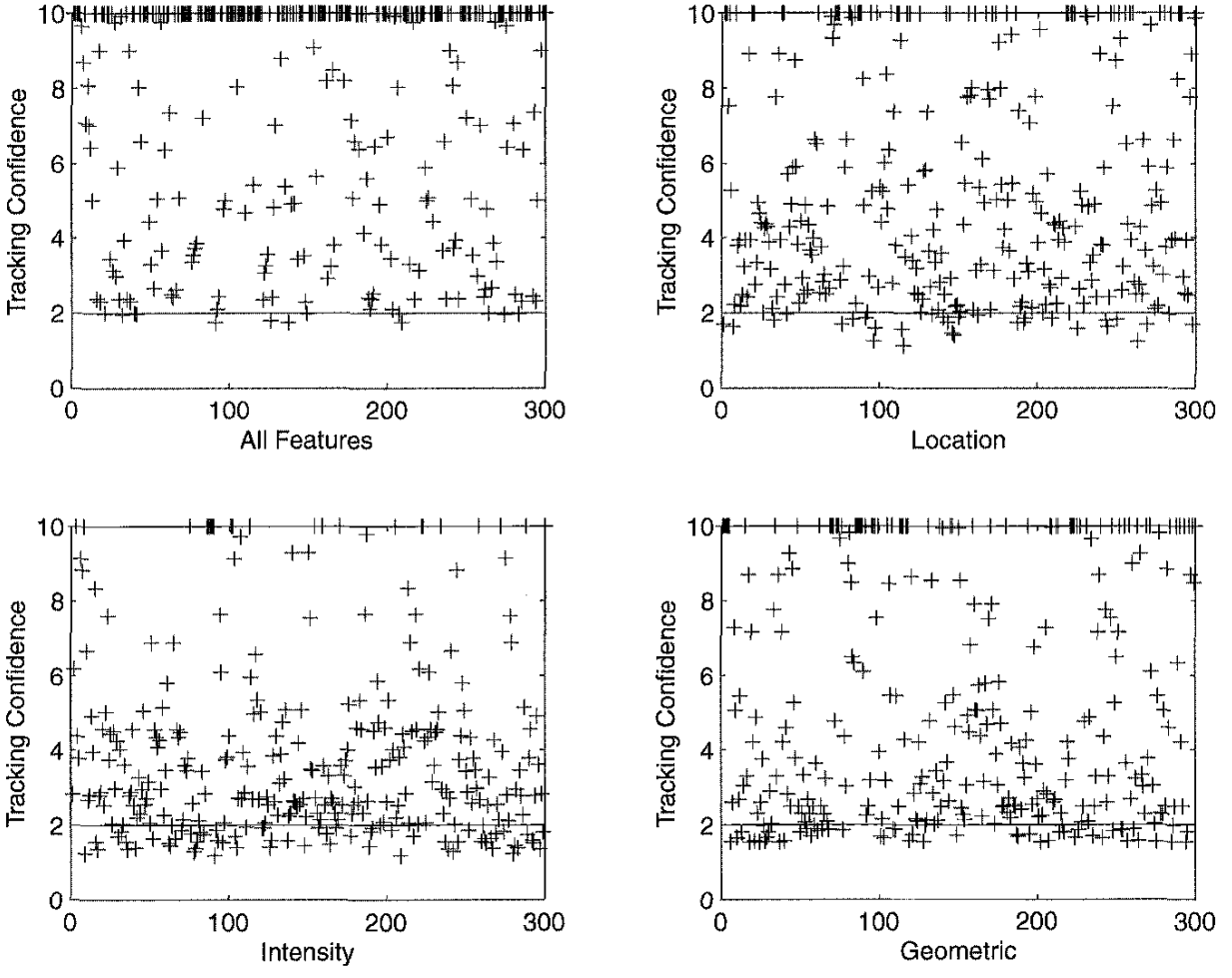

Fig. 3. Tracking confidence measurements in SVT.

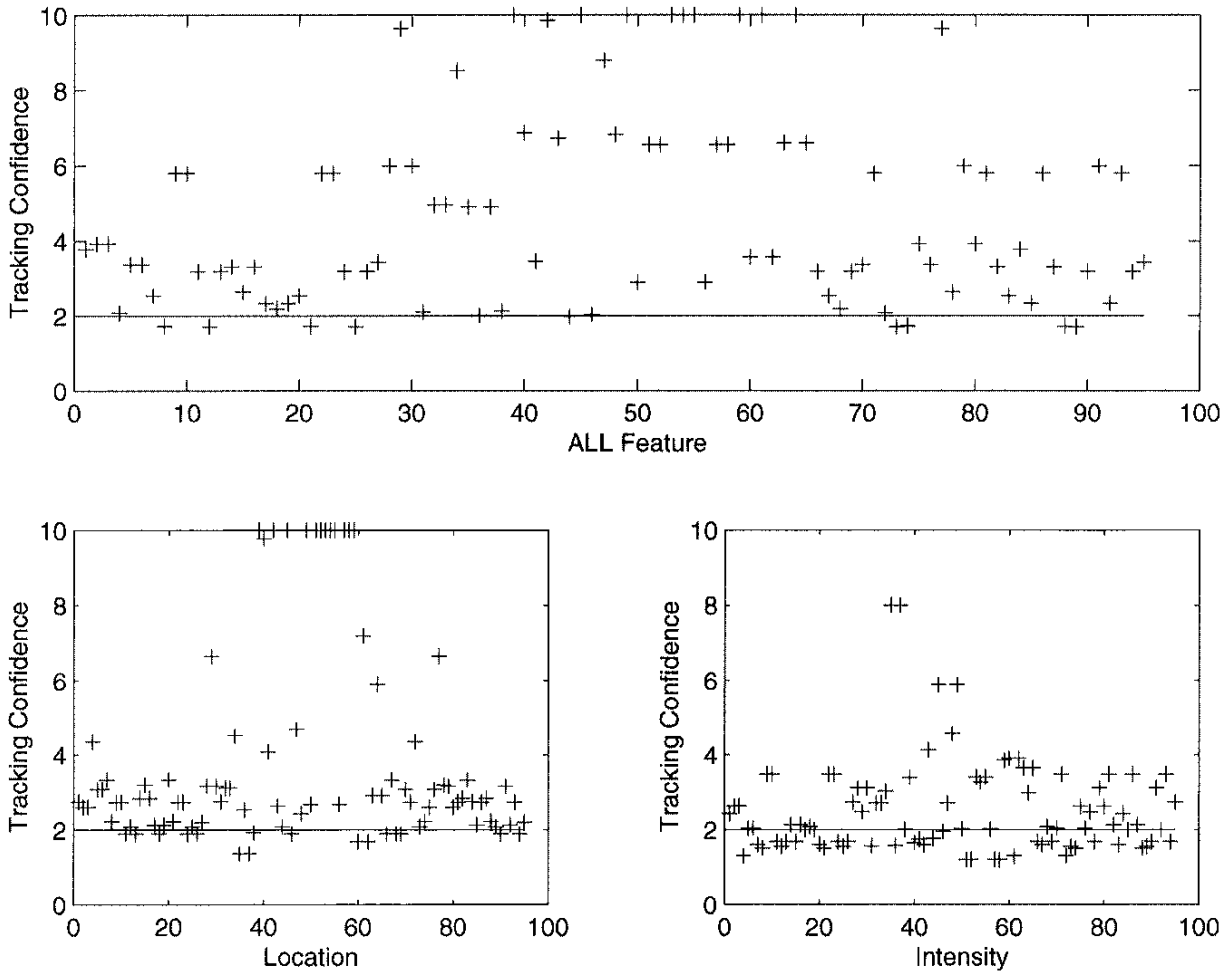

Fig. 4. Tracking confidence measurements in MVTT.

4.2.1 Matching Evaluation

Matching evaluation selects the optimal camera with high tracking confidence, i.e., with CF_i above the corresponding threshold.

4.2.2 Frame Number Calculation

Frame number calculation is used to minimize the amount of camera switching during tracking to reduce the computational cost. If more than one camera has a robust match, we use the current position and velocity of the subject to estimate the number of frames until the subject will move out of the view of the candidate camera or will move too far from the camera to be viewed well. We choose the camera that will image the subject over the most frames as the optimal camera. The detailed derivation of frame number calculation is addressed in [5].

5 Experimental Studies

5.1 A Prototype System

Our prototype system consists of three monocular cameras with partially overlapping fields of view, linked to a synchronization device, a digitizer, and a computer to handle all control and processing. We are interested in tracking moving humans in various indoor scenes, such as a long narrow corridor, an indoor corner, and a room. Complex scenes are considered to be combinations of these three typical scenes.

In the setup, we use three ULTRAK K-500 1/2" solid state b/w CCD cameras mounted with Computar H612FI wide angle lenses. A Matrox MAGIC frame grabber installed in a Compaq 486 PC grabs and digitizes 512×480 pixel images from the cameras. All images are processed by a RISC workstation running AIX (60 MHZ). The images are grabbed from the three cameras in the order of $C_0C_1C_2C_0C_1C_2\ldots$. The time interval between consecutive frames taken by the same camera is about 0.3 seconds, while the interval between consecutive frames taken by adjacent cameras (e.g., C_i and C_{i+1}) is about 0.1 second. The scaling factors for σ are set in such a way that we expect a valid match with D_t, $D_{x,t}$, $D_{y,t}$, and $D_{g,t}$ to be around 1. The thresholds are set as $T = T_x = T_y = T_g = 2$ and the weights are calculated as $w_x = w_g = 0.45$ and $w_y = 0.1$. These parameters were obtained from training on testing data. It takes about 0.3 seconds for the RISC workstation to process the tracking algorithm between consecutive frames.

We used seven data sets captured in a cluttered room, long corridors, corridor corners, a building lobby, and building elevators, with up to six people walking in various directions, and with still people in the background. Fig. 2 shows an example of tracking a subject in a corridor corner that involves all three basic modules: SVT, MVTT, and ACS. The first switching

($C_1 \rightarrow C_0, t = 2$) happens when the subject moves too far and the second switching ($C_0 \rightarrow C_2, t = 3$) is invoked when the subject is about to move out of the right boundary.

5.2 Performance Evaluation

Next, the system performance is evaluated [10] on about two hours of video (about 1.5 hours for SVT and 0.5 hours for MVTT) in three types of indoor environments. We use tracking confidence CF_t as a measure of the robustness of our algorithm (note only $CF_t \geq 2$ is considered a robust match from the previous description). We plot tracking confidence in both SVT and MVTT modules using all the features as well as each individual feature, as shown in Figs. 3 and 4, where the horizontal axis is the instances of feature correspondence and the vertical axis is the corresponding tracking confidence CF_ts. To have a better view of the low CF_ts, we clip any $CF_t \geq 10$ to be 10. The solid lines in each figure are the threshold of 2. Both figures show that using three types of features achieves a much higher tracking confidence than using any individual feature, and the intensity feature is the least robust. Thus, its weight is set smaller to achieve better tracking. More robust features could be substituted by simply following the defined framework. MVTT tracking confidences are lower than SVT due to the increased complexity of the algorithm and matching ambiguities between a 2D point and its estimated epipolar line from multiple perspectives.

Next, we evaluate the tracking algorithm by the tracking rate, defined as the percentage that the system tracks the right subject image. In SVT, we achieved a 98 percent rate of tracking using all the features. The rates of single feature tracking for location, intensity, and geometric features individually were 93.5 percent, 80.0 percent, and 84.5 percent, respectively. In MVTT, we obtained a rate of 96 percent when using both features, and 95 percent and 68 percent when using the location and intensity features individually. A match with high CF_t usually results in a correct match; wrong matches occur when the CF_t is below the threshold.

Failure of the proposed tracking algorithm is usually due to occlusion, which not only makes the low-level processing more difficult in the first stage, but also increases the matching ambiguity of the feature correspondence. Although we have developed techniques to deal with the problem of occlusion at a certain level, it still remains a major obstacle to the tracking problem. Other factors that degrade performance include reflection on glass and metal surfaces and dramatic changes in scenes viewed through glass doors. All of these factors prevent the system from accurately segmenting the subject image from a still background. MVTT tracking performance is less robust than SVT due to the uncertainty of the depth at the time of matching. Other factors which may deteriorate the algorithm performance are similarities in clothing to the background or in the distance between the subject and the viewing camera, which degrade the contribution of the intensity and geometric features during matching.

6 Conclusion

We have developed a comprehensive framework for tracking coarse human models from sequences of synchronized monocular grayscale images in multiple camera coordinates. Our framework demonstrates the feasibility of an end-to-end person tracking system that uses a unique combination of motion analysis on 3D geometry in multiple perspectives and existing techniques in motion detection, segmentation, and pattern recognition. Bayesian classification schemes associated with a general framework of motion analysis in a spatial-temporal domain are used for feature correspondence between consecutive frames under the same or different spatial coordinates. The performance of the algorithm has been evaluated from a prototype system in various types of indoor scenes and demonstrates the feasibility for real time applications.

Acknowledgments

This work was supported in part by the Texas Higher Eduction Coordinating Board under projects 95-ATP-442 and 97-ARP-275 and by the U.S. Army Research Office under contracts DAAH-04-94-G-0417 and DAAH-04–5-I-0494.

References

[1] K. Sato, T. Maeda, H. Kato, and S. Inokuchi, "CAD-Based Object Tracking with Distributed Monocular Camera for Security Monitoring," *Proc. Second CAD-Based Vision Workshop*, pp. 291-297, Champion, Pa., Feb. 1994.

[2] P.H. Kelly, A. Katkere, D.Y. Kuramura, S. Moezzi, S. Chatterjee, and R. Jain, "An Architecture for Multiple Perspective Interactive Video," *Proc. ACM Conf. Multimedia*, pp. 201-212, 1995.

[3] Q. Cai and J.K. Aggarwal, "Tracking Human Motion Using Multiple Cameras," *Proc. Int'l Conf. Pattern Recognition*, pp. 68-72, Vienna, Austria, Aug. 1996.

[4] Q. Cai, A. Mitiche, and J.K. Aggarwal, "Tracking Human Motion in an Indoor Environment," *Proc. Second Int'l Conf. Image Processing*, pp. 215-218, Washington, D.C., Oct. 1995.

[5] Q. Cai, "Tracking Human Motion in Indoor Environments Using a Distributed-Camera System," PhD thesis, The Univ. of Texas at Austin, 1997.

[6] R. Polana and R. Nelson, "Low Level Recognition of Human Motion," *Proc. IEEE CS Workshop Motion of Non-Rigid and Articulated Objects*, pp. 77-82, Austin, Tex., 1994.

[7] Y. Chang, X. Lebegue, and J.K. Aggarwal, "Calibrating a Mobile Camera's Parameters," *Pattern Recognition*, vol. 26, no. 1, pp. 75-88, 1993.

[8] K. Kanatani, "Constraints on Length and Angle," *Computer Visualization, Graphics, and Image Processing*, vol. 41, pp. 28-42, 1988.

[9] Q. Cai and J.K. Aggarwal, "Automatic Tracking of Human Motion in Indoor Scenes across Multiple Synchronized Video Streams," *Proc. Int'l Conf. Computer Vision*, Bombay, India, Jan. 1998.

[10] S. Pingali and J. Segen, "Performance Evaluation of People Tracking System," *Proc. IEEE CS Workshop Applications in Computer Vision*, pp. 33-38, Sarasota, Fla., 1996.