Model-Based Approach for Recognizing Human Activities From Video Sequences *

Shawn Dettmer, Avinash Seetharamaiah, Lin Wang and Mubarak Shah
Computer Vision Laboratory
Computer Science Department
University of Central Florida
Orlando, FL 32816

June 22, 1998

Abstract

In this paper, we propose a system for recognizing activities, e.g., walking, running, marching, skipping, etc. Our system is model-based: we use fourteen cylinders to model the human body and joint curves to model human motion in 3-D. Our system uses a single camera and a 3-D model to recognize activities (using whole body motion). The system will be able to deal with any arbitrary motion, not necessarily motion parallel to the image plane. We do not assume to know the height of the person performing activities, nor his or her distance from the camera. We will be able to deal with the motion of multiple people in a sequence, who are performing combinations of activities.

1 Introduction

Automatically detecting and recognizing human activities from video sequences is a very important problem in motion-based recognition. There

*This work was supported by a grant from DoD STRICOM under Co ntract No. N61339-96-K-0004. The content of the information herein does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

are several possible applications of the proposed research. One possible application is in automated video surveillance and monitoring, where human visual monitoring is too costly, too risky, and otherwise impractical. One human operator at a remote host workstation may supervise many automated video surveillance systems. This may include monitoring of sensitive sites for unusual activity, unauthorized intrusions, and triggering of significant events. Another area is detection and recognition of animal motion, with the primary purpose of discriminating it from the human motion in surveillance applications. Video games could be made more realistic using an activity recognition system, where the players control navigation by using their own body movements. A virtual dance or aerobics instructor could be developed that watches different dance movements or exercises and offers feedback on the performance [12]. Other applications include athlete training, clinical gait analysis, traffic monitoring, digital libraries (most videos are about people) and human-computer interface.

In this paper, we propose a model-based approach for recognizing activities. The proposed approach for recognizing various human activities is broken down into several basic steps (see Figure 1). First, the person, or persons, moving through the scene must be located. Once located, a sub-image around each person is taken. For each sub-image, edges are detected using the Canny edge detector. Straight lines are then fitted to the edge pixels using a recursive splitting scheme [3].

We have a set of joint angles for each activity (e.g. walking, running, skipping, etc.) to be recognized. These joint angles were obtained from Rohr [26] and Goddard [15]. Each set represents the angles of the various joints of the human body during a single cycle of the activity that it models. The *posture* is a value that indicates a point in the cycle which lies between 0 and 1 inclusive. Given the set of lines found in the sub-image, we need to find the *posture* and *pose* of the model that will fit the lines, where *pose* is the rotation and translation.

Given initial posture, we generate our 3-D model for each activity using the joint angles of *posture*. Given the 3-D points on the model, the *pose* can be estimated. In pose estimation, given a 3-D model and its 2-D projection from a particular view, the goal is to estimate the rotations and translations. We use Faugeras' [14] linear constraint minimization method for camera calibration for our pose estimation problem to estimate three rotations and translations of the object (torso) with respect to the camera.

Once the best *pose* for that *posture* is computed, the 3-D model is projected into 2-D space using both that *posture* and *pose*. Based on the differences between model lines and input lines, a likelihood measure can be assigned to this particular *pose* and *posture* for each activity.

Possible matches in *pose*, *posture* and the activity being performed can be found by taking those values that produced the best likelihood measures obtained during the search. Since this may produce several matches, we need to monitor the scene over time to distinguish the correct activity. We use a Kalman filter for each activity. Each filter will predict what the model should look like at a particular time for that activity. Based upon how well this prediction matches the scene, we can determine if a filter is allowed to continue or not. In the end, only one filter will remain, and that filter gives the correct activity.

2 Human Body Modeling Using Cylinders

In our approach, we represent the human body by a volume model consisting of 14 cylinders connected by joints. One cylinder is used for each body part we are modeling, which are the head, torso, upper arms, forearms, hands, thighs, calves, and feet. Each cylinder is described by two parameters: its length and the radius of its circular cross-sections. Each cylinder in the model has its own coordinate system, so all 3-D points on the cylinder correspond to that cylinder's coordinate system. The center of the torso is used to describe the location of the body, and acts as the origin for the 3-D coordinate system used for all points on all the cylinders. The location of cylinders attached to the torso cylinder (the head, the upper arms, and the thighs) are defined by transformations from the origin of the torso coordinate system. From there, the locations of the remaining cylinders are given as transformations from the origin of the coordinate system of the cylinder to which they attach, e.g. the forearms are attached to the upper arms, the calves are attached to the thighs, and so forth. Thus, all 3-D points on the cylinders can be computed based upon the location of the center of their cylinder and its orientation.

Some more sophisticated primitives like super-quadrics etc, can also be used for modeling human body, which may be more accurate from the graphics point of view. But, we feel that from the vision point view, the cylinders are sufficient and simple to deal with.

3 Human Activity Modeling Using Joint Angels

We use a kinematic approach to model the movement of people. The kinematic approach explicitly specifies the geometry of objects, for example, position and orientation. The joint curves with respect to posture are used for representing the geometric changes.

The joint curves of the shoulder, elbow, hip, and knee in one motion cycle are used to recognize different activities. The joint curves of the shoulder, hip, elbow, and knee represent the orientation of the upper arm with respect to torso, upper leg with respect to torso, lower arm with respect to upper arm, and lower leg with respect to upper leg respectively.

For modeling walking, we used the data from Rohr [26]. Sixty normal men ranging in age from 20 to 65 years have been analyzed to obtain the basic elements of walking. For each of the joints, Rohr used the angle positions at 10 discrete times instants in one cycle.

For modeling running, skipping, and jogging, we used the data from Goddard [15]. For each activity, three persons are analyzed. Each person has four samples of the same activity, which means each activity has 12 samples.

Each activity has been standardized to one cycle. The joint angels are expressed as function of posture, which varies from zero to one. Since the movement patterns of the body parts are very similar for different persons, the average data is used. Since all the activities considered here are symmetric movements, the motion curves of the joints are only needed for one side of the human body. In order to be able to calculate the joint angles in any postures, these values are interpolated by periodic cubic splines. Given the length of a cycle, n, for a given activity from the activity detection stage, we divide the joint curves into n equal intervals to be used in the activity recognition. This way we will know angels for each frame of a cycle.

Four joint curves are created for each activity. Fast re-projection of these motion states on a screen reveals that our motion model appears to be fairly realistic.

Using the proposed framework, we can recognize almost any activity (whole body movement) or gesture (upper body movement) as far as the model of activity in terms of joint curves is available. For example, we have modeled the *increase speed gesture*, widely used in the Army, using joint

angels involving right upper arm and forearm.

4 Line Correspondence

When we have the 2-D projection of our model, we need to be able to match the model lines to lines found in the scene. To establish this correspondence, the lines in both model and scene are represented by their direction, midpoint and length. For example, the line $x + ay + b = 0$ is represented by a vector $[a, b, y, l]^T$, where y is the ordinate of the midpoint and l is the length.

To determine what 3-D points in the model create edges when projected, the model is generated for some given rotation and translation around the camera coordinate system. Based upon the camera's view, all 3-D points whose surface normal is perpendicular to the camera angle are the 3-D points which create edges. These points are then grouped into straight lines.

When we match, we consider a line from the model, represented by the vector $\mathbf{m}_0$, which is $[a, b, y, l]^T$, and its covariance matrix $\mathbf{M}_0$ (The covariance matrix can be computed from the uncertainty in edge points, when fitting straight lines). All lines found in the scene are considered for matching. Each scene line is represented as a vector corresponding to one of the mappings, denoted $\mathbf{r}_i$ and its covariance matrix $\mathbf{R}_i$.

To find corresponding lines, we consider all scene lines to be possible matches. For an ideal match between a model line and a scene line, the equation $\mathbf{r}_i - \mathbf{m}_0 = \mathbf{0}$ would be satisfied. However, such an ideal match is practically non-existent. So, we need to find which scene line minimizes this equation. We compute the covariance matrices $\mathbf{\Lambda}_i = \mathbf{M}_0 + \mathbf{R}_i$, and the Mahalanobis distances $d_i = (\mathbf{r}_i - \mathbf{m}_0)^T(\mathbf{\Lambda}_i)^{-1}(\mathbf{r}_i - \mathbf{m}_0)$ for all lines in the scene. The scene line with the minimum Mahalanobis distance is considered to be the match.

See Figure 2 for results of line correspondence.

5 Pose Estimation

In order to accurately track a person and identify the activity being performed, it is necessary to obtain the pose of the person, that is, his rotation and translation with respect to the camera. To do this, we need to relate the

3-D points in our model to the 2-D points in the scene.

To accomplish the task of pose estimation, a method very similar to Faugeras' camera calibration [14] can be used. Where as Faugeras determines the position and rotation of the camera with respect to some world image coordinate center, we determine the position and rotation of an object (in this case, a person) with respect to the camera coordinate system.

For any 3-D point, $\mathbf{M}$, there is a transformation, $\tilde{\mathbf{P}}$, that will give us the 2-D image point, $\mathbf{m}$. This can be written as $\mathbf{m} = \tilde{\mathbf{P}}\mathbf{M}$, where each point is given in a homogeneous coordinate system.

We can define the matrix $\tilde{\mathbf{P}}$ as $\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{q}_1^T & q_{14} \\ \mathbf{q}_2^T & q_{24} \\ \mathbf{q}_3^T & q_{34} \end{bmatrix}$. For every point 3-D point, $\mathbf{M}_i = (X_i, Y_i, Z_i)$, on our model and its corresponding 2-D point, $\mathbf{m}_i = (u_i, v_i)$ in the scene, we can obtain two linear equations using above equations:

$$\begin{aligned} \mathbf{q}_1^T\mathbf{M}_i - u_i\mathbf{q}_3^T\mathbf{M}_i + q_{14} - u_i q_{34} &= 0 \\ \mathbf{q}_2^T\mathbf{M}_i - v_i\mathbf{q}_3^T\mathbf{M}_i + q_{24} - v_i q_{34} &= 0. \end{aligned}$$

Thus, for N points, we have $2N$ linear equations that can be written as $\mathbf{Aq} = 0$, where $\mathbf{A}$ is of the form $A = \begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_iX_i & -u_iY_i & -u_iZ_i & - \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_iX_i & -v-IY_i & -v_iZ_i & - \end{bmatrix}$ two rows for every point, i, and $\mathbf{q}$ is the 12×1 vector $[\mathbf{q}_1^T, q_{14}, \mathbf{q}_2^T, q_{24}, \mathbf{q}_3^T, q_{34}]^T$.

Using Faugeras' constraint minimization method, which constrains such a system of equations to avoid the meaningless solution of $\mathbf{q} = \mathbf{0}$, the transformation matrix $\tilde{\mathbf{P}}$ can be found. This is formulated as: $\min_{\mathbf{q}} ||\mathbf{Aq}||$ subject to $||\mathbf{q_3}||^2 = 1$. This involves finding the eigenvectors of a 3×3 matrix and inverting a 9×9 matrix. See Figure 2.g for results of pose estimation.

6 Activity Recognition

When the attempt is made to match the scene information with the various activities, it may be possible for several activities at various points during their cycle to be considered possible matches. In order to distinguish which of the possible matches is the *actual* activity, we use a Kalman filter for each of the potential activities.

To match the scene information to a particular activity, we have a state vector
$(X, Y, Z, R_x, R_y, R_z, p, \dot{X}, \dot{Y}, \dot{Z}, \dot{R}_x, \dot{R}_y, \dot{R}_z, \dot{p})$, where X, Y and Z are the 3-D coordinates of the person (torso) being tracked with respect to the camera, (R_x, R_y, R_z) is the rotation of the person around the camera axes, and p is the *posture* (i.e., the point in the cycle), and the remainder are their velocities.

For the Kalman filter, we need a transition matrix, ϕ, a weight (covariance) matrix, $\mathbf{Q}$, and a measurement matrix, $\mathbf{H}$.

To initialize each Kalman filter, the data obtained from pose and posture estimation is collected over several frames. This allows us to estimate the velocities. So, for each potential activity match, we have an initial state vector $\mathbf{a}_i^0$.

In order to determine which of the potential matches is actually the correct one, all "bad" matches must be eliminated. A Kalman filter for each activity is used. The idea is that the Kalman filter can estimate what the pose and posture of the model should be at the next time instant. Based upon how well that filter's prediction matches what is in the scene, we can determine if that filter, and its corresponding activity, is still a possible match or not.

For each state vector we have at time t, the filter predicts what the next state should be by computing $\mathbf{a}_i' = \phi \mathbf{a}_i^{t-1}$, and the covariance matrix $\mathbf{P}_i' = \phi \mathbf{P}_i^{t-1} \phi^T + \mathbf{Q}$. The parameters of the predicted state (i.e., the location, rotation and posture) are used as estimates for generating the 3-D model. This model is projected, line correspondence is performed, and the pose is estimated. This gives the measurement vector, $\mathbf{r}_t$, which contains the actual posture and pose of the person in the scene. Also, the covariance matrix, $\mathbf{R}_t$, is obtained from pose estimation.

Based upon the error in the line correspondence, we can determine if a filter should be allowed to continue, or be eliminated. The filters whose predictions produced a poor line correspondence (i.e. the error is above a threshold) are eliminated. For those matches which fall below the threshold, an update of the state (and its corresponding gain and covariance matrices) is computed using:

$$\mathbf{a}_i^t = \mathbf{a}_i' + \mathbf{K}_i^t(\mathbf{r}_t - \mathbf{H}\mathbf{a}_i')$$

where $\mathbf{K}_i^t = \mathbf{P}_i'\mathbf{H}^T(\mathbf{H}\mathbf{P}_i'\mathbf{H}^T + \mathbf{R}_t)^{-1}$, $\mathbf{P}_i^t = \mathbf{P}_i' - (\mathbf{K}_i^t\mathbf{H}\mathbf{P}_i')$, $\mathbf{r}_t$ is the measurement from the current frame and $\mathbf{R}_t$ is its covariance matrix.

After several frames of continuous processing, the filters corresponding to incorrect activities should give very poor line correspondences, and can be dismissed. The filter of the correct activity should consistently give good line correspondences with its prediction and the scene, and, thus, consistently give little error. Eventually, only one such filter should remain, and its corresponding activity declared the match. Thus, the Kalman filter doesn't actually perform the recognition, but gives us the means by which we can distinguish activities. See Figure 3 for preliminary results of activity recognition with synthetic scenes.

7 Comparison With Other Approaches

Several systems for detecting, recognizing and tracking human activities have been proposed in the literature (see [6] for a comprehensive review or earlier work). A brief summary of representative approaches in a tabular form is given in Figure 4. There are two main classes of approaches: 2-D approaches, and 3-D model-based approaches. In 2-D approaches no model of a 3-D body is used, only 2-D motion, e.g. optical flow, is employed to compute features in a sequence of frames to recognize activities. In 3-D approaches, a 3-D model of the human body and joint angles are employed. In the model-based approach, one to four cameras have been used. The types of tasks, which have been reported include: detection of activities [23], detection of cyclic motion [2, 23, 27] tracking walkers using a model-based approach [26, 17], recognizing people by their gait [1, 20], recognizing activities [23], and recognizing gestures [8, 18, 25, 30, 7].

So far, to our knowledge, there is only one approach, proposed by Polana and Nelson, for activity recognition. This approach is 2-D, does not use a model of the human body and motion, and it assumes motion is parallel to the image plane and assume to know the height of the person.

Davis et. al. [13] employ multiple cameras to compute the joint angels and a person's pose using search in the high dimensional space. Metaxas et al [21] also use multiple cameras to compute *both* a 3-D body model and a 3-D motion model. In Metaxas et al's work the problem of recognition is not addressed. In both of these approaches, the complexity of the system is very high because they deal with very large degrees of freedom (22 in Davis et al's system, even more in Mataxas's system). Therefore, these systems are

highly prone to noise.

On the other hand, in the proposed approach, we employ single camera and use a priori known 3-D body model and 3-D motion model to recognize human activities, thus increasing robustness.

References

[1] Adelson, E.H. and Niyogi, S. A. Analyzing and recognizing walking figures in XYT. In *IEEE CVPR-94*, pages 469–474, 1994.

[2] M. C. Allmen and C. R. Dyer. Cyclic Motion Detection Using Spatiotemporal Surfaces and Curves. In *Proc. 10th Int. Conf. Pattern Recognition*, pages 365–370, 1990.

[3] Ballard, D.H., and Brown, C.M. *Computer Vision*. Englewood Cliffs, N.J.: Prentice Hall, 1982.

[4] Bobick, A. F. and Campbell, L.W. Recognition of human body motion using phase space constraints. In *IEEE ICCV-95*, pages 624–630, 1995.

[5] Cédras, C., and Shah, M. A survey of motion analysis from moving light displays. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 1994.

[6] Cédras, C., and Shah, M. Motion-based recognition: A survey. *Image and Vision Computing*, 13(2):129–155, March 1995.

[7] Cui, Y., Swets, D., and Weng, J. Learning-based hand sign recognition using shoslif-m. In *IEEE-ICCV*, pages 631–636, 1995.

[8] Darrell, T., and Pentland, A. Space-time gestures. In *CVPR*, pages 335–340. IEEE, 1993.

[9] Davis, J., and Shah, M. Recognizing hand gestures. In *ECCV*, pages 331–340, May 1994.

[10] Davis, J., and Shah, M. Three-dimensional gesture recognition. In *Asilomar Conference on Signals, Systems, And Computers*, 1994.

[11] Davis, J., and Shah, M. Visual gesture recognition. *IEE Proceedings Vision, Image and Signal Processing*, 141(2):101–106, 1994.

[12] Davis, Jim. Appearnce-based motion recognition of human actions. Technical Report 387, MIT Media Lab Perceptual Computing Group, Cambridge: MIT, 1996.

[13] Davis, L. S. and Gavrila, D. M. 3-d model-based tracking of human upper body movement: A multi-view approach. In *IEEE CVPR-96*, pages 73–80, 1996.

[14] Oliver Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993.

[15] N. H. Goddard. *The Perception of Articulated Motion: Recognizing Moving Light Displays.* PhD thesis, University of Rochester, 1992.

[16] Gould, K., and Shah, M. The trajectory primal sketch. In *Conference on Computer Vision and Pattern Recognition*, pages 79–85, San Diego: IEEE Computer Society, June, 1989.

[17] D. C. Hogg. *Interpreting Images of a Known Moving Object.* PhD thesis, University of Sussex, 1984.

[18] J. Schlenzig, E. Hunter and R. Jain. Recursive identification of gesture inputs using hidden markov models. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 187–194, 1994.

[19] Li Nan, Shawn Dettmer, and Mubarak Shah. Visual lipreading using eigensequences. In *Proc. International Workshop on Automatic Face and Gesture Recognition*, pages 30–34, 1995.

[20] Little, J. and Boyd, J. Describing motion for recognition. *Int. Symposium on Computer Vision-95*, pages 235–240, November 1995.

[21] Metaxas, D. and Kakadiaris, I.A. Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In *IEEE CVPR-96*, pages 81–87, 1996.

[22] Pentland, A. and Azarbayejani, A. Real-time self-calibrating stereo person tracking using 3-d shape estimation from blob features. Technical Report 363, MIT Media Laboratory, 1996.

[23] R. Polana and R. C. Nelson. Detecting Activities. In *Proc. Conf. on Computer Vision and Pattern Recognition*, pages 2–7, New York, NY, June 15–17 1993.

[24] Rangarajan, K., Allen, Bill, and Shah, M. . Matching motion trajectories. *Pattern Recognition*, 26:595–610, July, 1993.

[25] Rehg, J., and Kanade, T. Visual tracking of high dof articulated structures: an application to human hand tracking. In *ECCV*, pages 35–46, May 1994.

[26] K. Rohr. Towards Model-Based Recognition of Human Movements in Image Sequences. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 1993.

[27] Seitz, S. M. and Dyer, C. R. Affine invariant detection of periodic motion. In *Proc. of IEEE CVPR-94*, pages 970–975, 1994.

[28] M. Shah and R. Jain. *Motion-Based Recognition.* Kluwer Academic Publisher, 1996. (to appear).

[29] Tsai, Ping-Sing, Keiter, K., Kasparis, T., and Shah, M. Cyclic motion detection. *Pattern Recognition*, 27(12), 1994.

[30] Wilson, A. D. and Bobick, A. Learning visual behavior for for gesture analysis. In *Proc. IEEE Intl. Symp. on Computer Vision.* IEEE Computer Society, 1995.

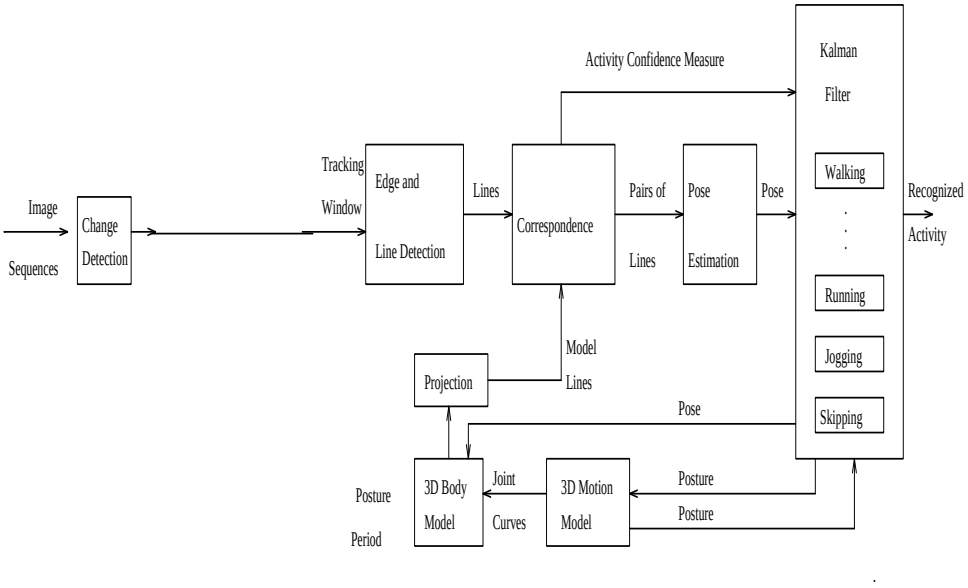

Figure 1: Block Diagram of the proposed approach.

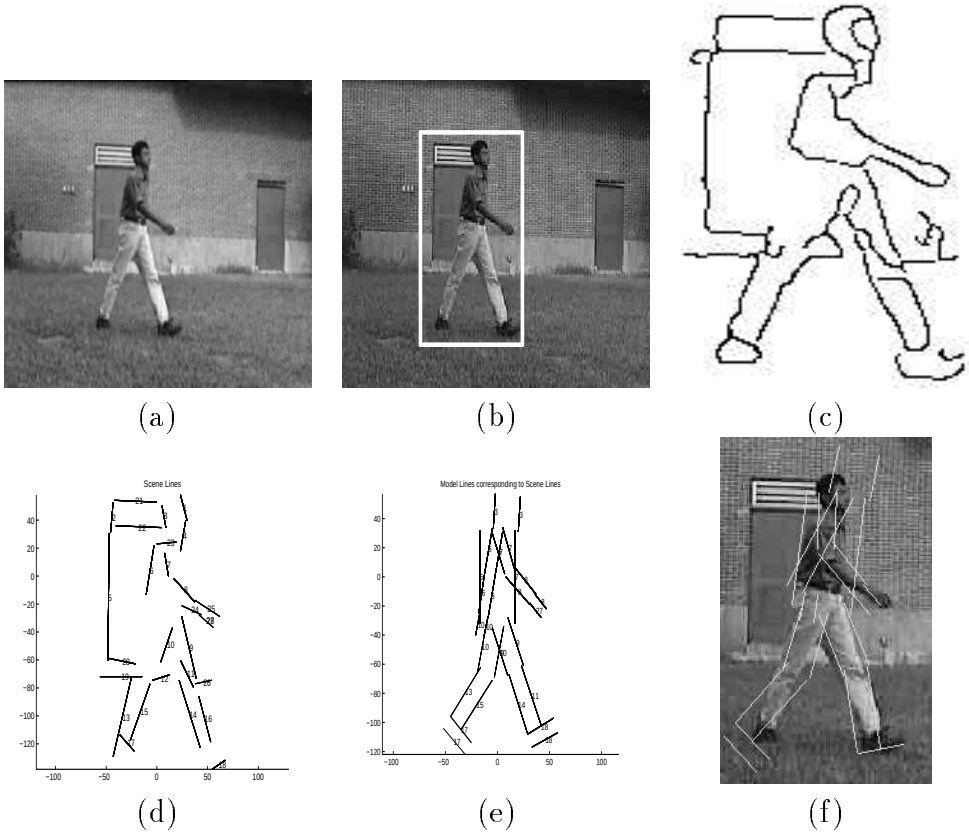

Figure 2: Results of line correspondence and pose estimation. (a) Original image. (b) Tracking window superimposed on the original image. (c) Detected edges in the tracking window. (d) Detected scene lines from (c). (e) Corresponding model lines. (f) Model lines superimposed on the original image. Computed pose (translation and rotation) is applied to the model, then projected on the image plane. The lines closely match with the image.

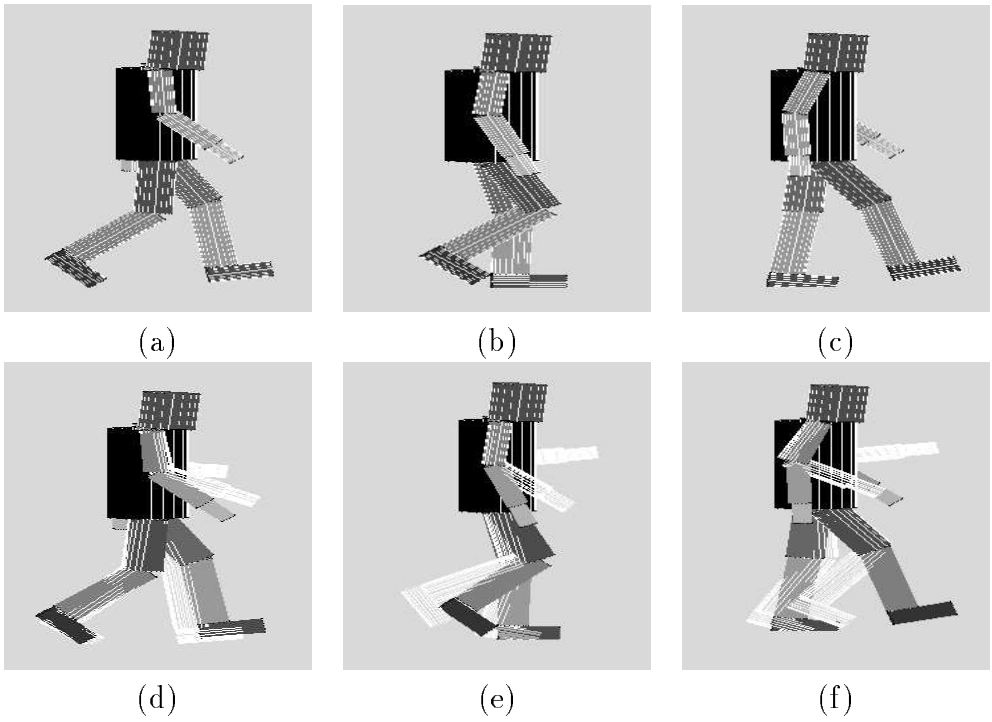

Figure 3: Results for activity recognition using synthetic sequences. An unknown sequence (which happens to be Walking) is attempted to match with Walking (a–c) and Running (d–f) model using Kalman filter. The frames in the unknown sequence are superimposed on corresponding model frames of Walking and Running. In a short time the match stabilizes to walking, the match for running is poor.

author	**persons**	**3D Model**	**segmentation**	**input**	**cameras**	**task**	**motion**
Hogg[17]	two	yes	manual	2-D	one	tracking walker	parallel image p
Rhor[26]	one	yes	automatic	2-D	one	tracking walker	parallel image p
Nelson[23]	multiple	no	automatic	2-D	one	activity recognition	parallel image p
Adelson[1]	multiple	no	automatic	2-D	one	gait recognition	parallel image p
Bobick[4]	one	yes	manual	3-D	six	ballet recognition	general image p
Little[20]	one	no	manual	2-D	one	gait recognition	parallel image p
Davis[13]	one	yes	automatic	3-D	four	tracking, recognition	general
Pentland[22]	one	no	automatic	3-D	two	tracking	general
Metaxas[21]	one	acquired interactively	automatic	2-D	three	tracking arm	general
Proposed	**multiple**	**yes**	**automatic**	**2-D**	**one**	**activity detection & recognition**	**genera**

Figure 4: Comparison of approaches.