Tracking Humans using Prior and Learned Representations of Shape and Appearance

Jongwoo Lim
Department of Computer Science
University of Illinois at Urbana, Champaign
Urbana, IL 61801

David Kriegman
Department of Computer Science and Engineering
University of California at San Diego
La Jolla, CA 92093

Abstract

Tracking a moving person is challenging because a person's appearance in images changes significantly due to articulation, viewpoint changes, and lighting variation across a scene. And different people appear differently due to numerous factors such as body shape, clothing, skin color, and hair. In this paper, we introduce a multi-cue tracking technique that uses prior information about the 2-D image shape of people in general along with an appearance model that is learned on-line for a specific individual. Assuming a static camera, the background is modeled and updated on-line. Rather than performing thresholding and blob detection during tracking, a foreground probability map (FPM) is computed which indicates the likelihood that a pixel is not the projection of the background. Off-line, a shape model of walking people is estimated from the FPMs computed from training sequences. During tracking, this generic prior model of human shape is used for person detection and to initialize a tracking process. As this prior model is very generic, a model of an individual's appearance is learned on-line during the tracking. As the person is tracked through a sequence using both shape and appearance, the appearance model is refined and multi-cue tracking becomes more robust.

1. Introduction

The goal of our work is to develop techniques for robustly tracking walking people over long sequences of images in which the person may be seen from many directions, the lighting may vary across the scene and over time, and where there may be occasional occlusion and other moving objects. Our approach is essentially to treat person tracking as incremental appearance-based recognition in which we have an appearance model for the class of objects that we are tracking along with the object's state in the previous frames. We start tracking with a shape model that essentially captures the detectable silhouette of people and is learned off-line from training sequences. On-line as a person is first detected and then tracked just using the 2-D shape model, the tracker automatically learns the appearance of the tracked person. Combining two cues, generic person shape and person-specific appearance, the tracker is able to continue to track the person in more difficult situations than just using shape alone. An overview of the proposed algorithm is depicted in Figure 1.

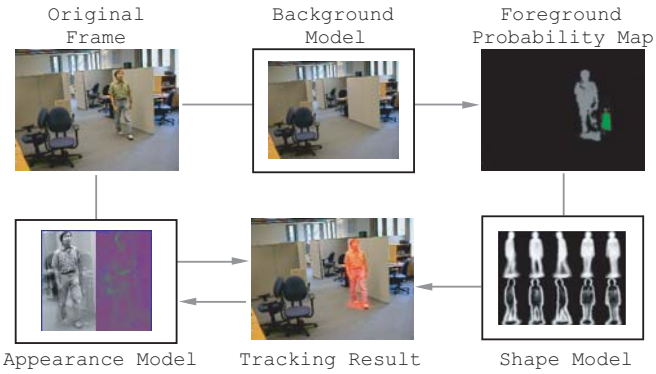


Figure 1: Overview of the proposed tracking algorithm; From the acquired image frame, the foreground probability map is built using the background model, and both image and foreground probability map are used to compute the updated location of people.

When the camera is fixed and the scene is static or slowly varying, a simple representation of the background can be used, and it can be updated on-line during the process of tracking. The background model is used to determine the probability that a pixel is an element of background (or likewise the probability that it is part of a foreground object – the foreground probability map (FPM)). The main advantage of background modeling is that it provides information about the shape and location of the object that we are interested in. Early tracking algorithms used a background model to find blobs in the foreground [9, 6]. But often this approach fails due to image noise, shadows, sudden illumination changes, or movement of background objects.

In our approach, we train off-line on sequences of images of walking people who assume different poses and configurations. Silhouettes (shape) are detected using the background model, and these are clustered; each cluster corresponds to a different pose. Hence, the offline model is a collection of FPMs representing distinctive poses. This shape model can be used in detecting and tracking a person in the

image. When a connected component of pixels in the FPM with high probability (a blob) is found, it is compared with the silhouettes in the model, and if it is similar enough, we initialize a tracking process. While tracking, we continuously find the best affine warp that aligns the current blob with one of the shapes in the model.

Tracking with silhouettes is very effective, but it cannot distinguish between multiple people in the scene, and sometimes fails when the foreground probability map becomes noisy or unavailable. To help the tracker to focus on the correct target, we learn the appearance of the person on-line. We adopted the WSL model [4] to represent the pixel intensities of each pixel. The appearance model of the tracked person consists of the sets of WSL models for different poses in the shape model. The learned appearance is used in estimating the warp parameter jointly with the shape model.

More than simply using two cues in tracking, we want to emphasize that these two models have different modes. The shape model is learned off-line from various individuals and represents the general shape of people. The appearance model is learned on-line from the person being tracked and represents its own intensity patterns, which can be changing over time. The shape model gives prior information about people in general, and the appearance model gives detailed information about an individual that distinguishes it from other individuals. Combining these two features, we show that the tracking process becomes more robust.

2. Related Work

In this section we review representative previous works and relate it to our approach.

Recently Toyama and Blake [7] developed a contour-based human body tracker. This algorithm uses exemplars of contours to model the shape of possible targets, and a probabilistic framework to maintain the model efficiently and effectively. In this work, since only contour information is used for tracking, it may fail when the edges in the frame become unavailable or inaccurate, for example, due to background clutter or occlusion.

When the camera is static, the background image can be modeled to indicate whether or not each pixel is similar to the modeled background, and this information directly gives the location of the foreground objects. Most previous work used rather simple background models for performance reasons, and focused on choosing a probability or color model to handle difficult situations like cast shadows or moving background objects [8, 3]. Compared to hard thresholding, we adopted a probabilistic representation of the foregroundness of each pixel. This probabilistic representation enables more robust and accurate object detection and tracking.

One effective technique to model the appearance of objects was introduced by Jepson et al. [4]. The WSL model is able to maintain a probabilistic representation for noisy and time-varying data. Tracking with only the WSL model may suffer drifts due to the lack of prior information about the object being tracked; the background may be mistakenly incorporated into the object model, and the tracker may then drift.

3. Models for Tracking

To track an individual, we maintain both a representation of the person as well as the background. We use the background model to identify pixels that are likely to be part of the moving person, and use two models to represent people: the shape model for people's general shapes and the appearance model for an individual's texture and color.

3.1. Background Model

We assume that the camera is static, and learn the background scene to segment people from a stationary or slowly varying background. Background pixel color is modeled as a multivariate normal distribution with a mean $\boldsymbol{\mu}$ and a diagonal covariance matrix Σ at each pixel at $\mathbf{x}$ (the background model is then $B = \{(\boldsymbol{\mu}_{\mathbf{x}}, \Sigma_{\mathbf{x}})\}$). The YUV color space is used since with the proper distribution on pixel values, the effect of shadows can be reduced. We defined the probability that a pixel at $\mathbf{x}$ belongs to the foreground as

$$p_{fg}(\mathbf{x}|I,B) \;=\; 1 \;-\; p_{bg}(\mathbf{x}|I,B) \;=\; 1 \;-\; e^{-d(\mathbf{x}|I,B)^2}$$

where $d(\mathbf{x}|I,B)^2$ is the squared Mahalanobis distance $(I(\mathbf{x}) - \boldsymbol{\mu}_{\mathbf{x}})^\top \Sigma_{\mathbf{x}}^{-1} (I(\mathbf{x}) - \boldsymbol{\mu}_{\mathbf{x}})$. For every pixel $\mathbf{x}$ in an input image, $p_{fg}(\mathbf{x}|I,B)$ can be evaluated to produce a foreground probability map I_{fg}. See Figure 2 for an example.

At each frame, the mean and variance of each pixel in the background model are updated on-line [6].

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{x}}^{(t+1)} &\leftarrow (1-\rho)\,\boldsymbol{\mu}_{\mathbf{x}}^{(t)} + \rho\, I^{(t)}(\mathbf{x}) \\ \sigma_{\mathbf{x},i}^{(t+1)} &\leftarrow (1-\rho)\,\sigma_{\mathbf{x},i}^{(t)} + \rho\, (I_i^{(t)}(\mathbf{x}) - \boldsymbol{\mu}_{\mathbf{x},i}^{(t)})^2 \end{aligned}$$

where the superscript (t) represents the data at time t, $\sigma_{\mathbf{x},i}$ is the i-th element on the diagonal of $\Sigma_{\mathbf{x}}$, $I_i(\mathbf{x})$ and $\boldsymbol{\mu}_{\mathbf{x},i}$ are the i-th element of $I(\mathbf{x})$ and $\boldsymbol{\mu}_{\mathbf{x}}$, and ρ is the learning rate, which is usually a small constant. We only keep diagonal terms of the covariance matrix for simplicity and efficiency, and to avoid instability, set a lower bound on the variance.

Due to the noise from the image acquisition process, the foreground probability map is often noisy and needs to be smoothed to get a robust estimate of object status. We can suppress the noise by augmenting our assumption of pixel-wise independence in the background model by considering spatial coherence [1]. The spatial coherence on pixels in an

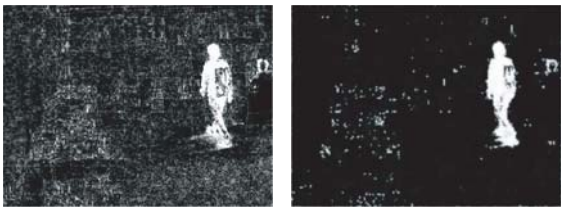


Figure 2: Foreground probability maps before and after spatial coherence smoothing.

image I can be expressed as minimizing the error function

$$E_{sc}(\hat{I} \mid I) = \sum_{\mathbf{x}} \left(\mid \hat{I}(\mathbf{x}) - I(\mathbf{x}) \mid^2 + \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x})} c \mid \hat{I}(\mathbf{x}) - I(\mathbf{x}') \mid^2 \right)$$

where $\mathcal{N}(\mathbf{x})$ is the horizontal/vertical neighbors of $\mathbf{x}$, and c is a weighting constant. We can calculate $\hat{I}_{fg}$ using an iterative gradient descent algorithm. After convergence, the smoothed foreground probability map $\hat{I}_{fg}$ replaces I_{fg}.

3.2. Shape Model

The background model allows us to build the foreground probability map of the current frame. In the foreground probability map, there may be false positive pixels like moving background objects or shadows. To distinguish human shapes from other shapes, we use a prior model of the possible shapes of people.

3.2.1. Learning the Shape Model

The shape model is learned off-line from training video sequences of people walking around, which are taken in favorable imaging conditions. From training images and using background modeling, we can obtain cropped silhouettes of people to build a large collection of foreground probability maps of humans in a wide variety of poses. In most of the frames, the blob detector finds the correct bounding boxes of humans, and when the region does not represent a human or includes false positives or excessive noise, an operator can ignore or modify the detected region. After this step, all patches are scaled and aligned to have the same size and center, so that they represent normalized human shapes.

Since the shape of each pose may be significantly different, the global mean image (FPM) of all cropped, scaled and aligned patches is too blurry to be used in detection and tracking. Instead of one global mean, we need a more precise representation of each pose which then leads to greater discriminative power. All of the normalized patches are clustered into k sets using the K-means clustering algorithm. For each set of patches, we build the mean image as the representation of the pose. We will denote the shape model S as $\{\mathbf{s}_k\}$, where $\mathbf{s}_k$ is the mean image of the cluster k. The result of clustering 453 images of a walking person into four clusters is shown in Figure 3. We used this shape model for all tracking experiments in this paper. Each mean image represents the pose of people clearly, and the variance

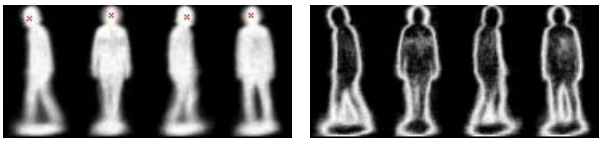


Figure 3: The shape model; the mean and variance images of 4 clusters of normalized foreground probability map patches from the training sequence.

images show significant variation only along the boundary of the shape. Recall that we are not clustering intensity images, but the foreground probability maps.

The main advantage of using the shape model is that it gives a simple and concise representation of the possible shapes of people regardless of their clothing or appearance. Also this shape model is general, in the sense that it does not depend much on the identity of individuals. Since we will allow the model to deform in an affine way, the size or aspect ratio (which can be related to each person's characteristics) can be easily handled by this shape model.

3.2.2. Tracking with the Shape Model

In our formulation, tracking amounts to finding a set of warp parameters which map the foreground blob of the current frame onto one of the silhouettes in the shape model. Throughout this work, we use 2D affine warps as our motion model, and this captures 2D translation, rotation, scaling with fixed or changing aspect ratio, and skewing. We denote an affine warp of the coordinates $\mathbf{x}$ with the parameter $\boldsymbol{\theta}$ as $w(\mathbf{x}; \boldsymbol{\theta})$. We can formulate the error of the current warp with parameter $\boldsymbol{\theta}$ as follows:

$$E(\boldsymbol{\theta}) \;=\; \min_k \sum_{\mathbf{x}} \mid \mathbf{s}_k(\mathbf{x}) - I_{fg}(w(\mathbf{x}; \boldsymbol{\theta})) \mid^2 \qquad (1)$$

Usually finding the global minimum of this error function is very hard, but for tracking, we can expect the motion of people between two consecutive frames to be small enough to be able to assume that a local minimum close to the current parameter is the correct warp. We use a gradient-descent method to find the warp, and a Taylor series expansion for efficient computation [5, 2]. We decompose $\boldsymbol{\theta}$ into $\boldsymbol{\theta}_0 + \delta\boldsymbol{\theta}$ where $\boldsymbol{\theta}_0$ is the initial warp parameter for the current frame, and $\delta\boldsymbol{\theta}$ is the update for the warp. The error function can be written as a function of $\delta\boldsymbol{\theta}$ instead of $\boldsymbol{\theta}$ as

$$\begin{aligned} E(\delta\boldsymbol{\theta}) \;&=\; \min_k \sum_{\mathbf{x}} \mid \mathbf{s}_k(\mathbf{x}) - I_{fg}(w(\mathbf{x}; \boldsymbol{\theta}_0 + \delta\boldsymbol{\theta})) \mid^2 \\ &\simeq\; \min_k \sum_{\mathbf{x}} \mid \Delta I_{fg}(\mathbf{x}, k; \boldsymbol{\theta}_0) - M_k^\top \delta\boldsymbol{\theta} \mid^2 \end{aligned}$$

where $\Delta I_{fg}(\mathbf{x}, k; \boldsymbol{\theta}_0) = \mathbf{s}_k(x) - I_{fg}(w(\mathbf{x}; \boldsymbol{\theta}_0))$, and M_k is a matrix whose i-th column is $\frac{\partial \mathbf{x}}{\partial \theta_i} \frac{\partial \mathbf{s}_k}{\partial \mathbf{x}}$. The update $\delta\boldsymbol{\theta}$ for the pose k is given as

$$\delta\boldsymbol{\theta}_k \;=\; (M_k M_k^\top)^{-1} M_k \, \Delta I_{fg}(\mathbf{x}, k; \boldsymbol{\theta}_0) \qquad (2)$$

and since the matrix $(M_k M_k^\top)^{-1} M_k$ remains constant unless s_k changes, it can be precomputed at the program start

and used there after. The pose k and the corresponding $\boldsymbol{\theta}_k$ giving the minimum error are picked as the estimated pose and motion parameters of the current frame.

3.2.3. Detection using Shape Model

To initialize the tracker, we need to find the image region that contains a person who has entered the field of view. The shape model can be used for tracker initialization, since it can be used as a filter which ignores non-human objects based on their similarity to the shape model. We can detect people most of the time without rotating or skewing the image. This helps to keep the detection process simple, since we only translate and scale the model according to the bounding box of the foreground blob. To minimize the false-positives in detection, we restrict the scale to be in a reasonable range and the threshold for similarity high enough not to detect moving non-human objects as being human. Due to this restriction, a person may not be detected for some period until the size and shape are within the specified range to initialize the tracker.

3.3. Appearance Model

Though silhouettes are a powerful representation of person shape, in many cases, shape itself is not enough, for example, when two people cross each other or when the background model fails (like shadow or sudden illumination change). To compensate for this weakness, we learn on-line the appearance of people that we are tracking, by remembering the pixel intensities at each pixel of each pose.

Due to the huge variety of colors or textures of human clothing, skin and hair, learning a general appearance model a priori for people is very difficult, but once we track one specific individual who steps in view, the color or texture does not change much, so on-line appearance modeling is possible.

The statistics of pixel intensities are learned using the WSL model [4]. (Note that in our current implementation, we only use intensity not color as initial experiments did not yield much benefit from color for the added computational cost). It maintains three probability distributions, Wandering, Stable and Lost models. The Stable model gives representative values of the pixel, and the weights among the three models show the confidence given to each model. The probabilistic mixture model for the data d_t at time t is

$$p_{wsl}(d_t \mid \mathbf{m}_t, \mu_{s,t}, \sigma^2_{s,t}, d_{t-1}) = m_w p_w(d_t; d_{t-1}, \sigma^2_w) + m_s p_s(d_t; \mu_{s,t}, \sigma^2_{s,t}) + m_l p_l(d_t)$$

where $\mathbf{m}_t = (m_w, m_s, m_l)_t$ are the mixing probabilities, $\mu_{s,t}$, $\sigma^2_{s,t}$ are the mean and variance for the Stable model, σ^2_w is a fixed variance for the Wandering model. p_w and p_s are normal distributions, and p_l is a uniform distribution. Among these parameters, $\mathbf{m}_t$, $\mu_{s,t}$ and $\sigma^2_{s,t}$ are updated on-line. More details are described in [4].

We assign one WSL model per each pixel of each pose, and all models of the estimated pose are updated on-line with the current frame. The template of the appearance for tracking is defined as $\mathbf{a}_k = \{\, \mu_s \text{ at pixel } \mathbf{x} \text{ of pose } k \,\}$.

3.4. Tracking with Shape and Appearance

With the shape model and the appearance model, we need to rewrite the objective function in (1) to use both models. We use the Stable model as the template of appearance $\mathbf{a}_k$ of the pose k.

$$E(\boldsymbol{\theta}) = \min_k \sum_{\mathbf{x}} \Big(|\, \mathbf{s}_k(\mathbf{x}) - I_{fg}(w(\mathbf{x};\boldsymbol{\theta}))\,|^2 + \alpha |\, \mathbf{a}_k(\mathbf{x}) - I_{gr}(w(\mathbf{x};\boldsymbol{\theta}))\,|^2 \Big)$$

where I_{gr} is the current frame image converted into grayscale. With a formulation similar to the one in Section 3.2.2, we can compute the update parameter $\delta\boldsymbol{\theta}$ as

$$\delta\boldsymbol{\theta}_k = (M_k M_k^\top + \alpha\, N_k N_k^\top)^{-1} \big(M_k\, \Delta I_{fg}(\mathbf{x}, k; \boldsymbol{\theta}_0) + \alpha\, N_k\, \Delta I_{gr}(\mathbf{x}, k; \boldsymbol{\theta}_0) \big) \quad (3)$$

where N_k is, similar to M_k, a matrix whose i-th column is $\frac{\partial \mathbf{x}}{\partial \theta_i} \frac{\partial \mathbf{a}_k}{\partial \mathbf{x}}$, and $\Delta I_{gr}(\mathbf{x}, k; \boldsymbol{\theta}_0) = \mathbf{a}_k(x) - I_{gr}(w(\mathbf{x}; \boldsymbol{\theta}_0))$.

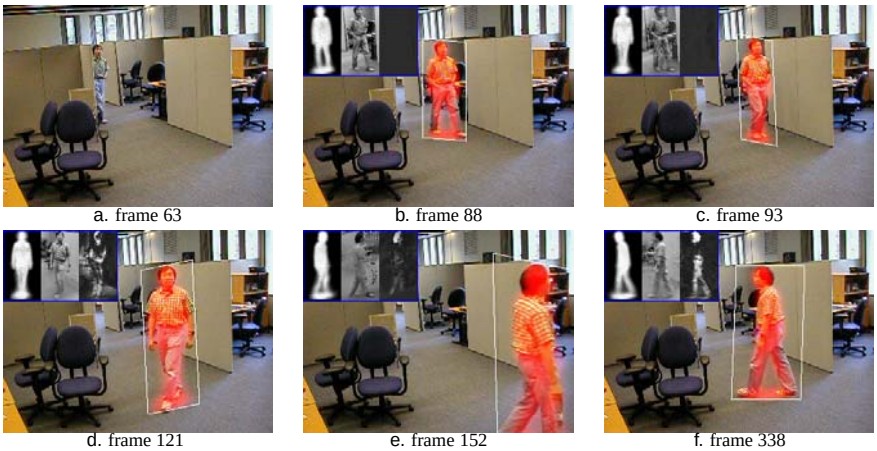
The contribution of the two models to the parameter update can be controlled by the constant α. In this paper we used a fixed α, but α could be determined automatically according to the likelihoods of the shape and appearance models to the current frame.

4. Experimental Result

The algorithm is implemented in C++, and is tested on both recorded sequences and live video inputs. On a modest desktop machine, it runs at 3-4 frames per second (fps) with the shape model only, and at 1-2 fps with both models. The main reason for the difference in performance is that, for the appearance model, we cannot precompute the matrix required in the parameter update step (Eq. (2) & (3)), since the appearance model is updated on-line at each frame.

In Figure 4, the window in the upper left of each image shows the current status of the appearance model, the left subwindow is the Stable model, and the right subwindow is the weight of the Stable model in the WSL model. The tracking result is overlaid over the original frame in a red color with a white bounding box.

Figure 4 shows frames from a video which has one person in it. The person in the sequence walks around in the room, and his pose, location and scale in the image change significantly. The person in this sequence has not been shown in the training sequences for the shape model, which shows the extensibility of our shape model. In Figure 4.a, the person appears in the scene, but the shape does not match anyone in the shape model. After some frames, the



a. frame 63 b. frame 88 c. frame 93

d. frame 121 e. frame 152 f. frame 338

Figure 4: Tracking result for a one-person sequence. In the window at the upper left corner of each frame, the left subwindow shows the estimated pose, the center subwindow shows the appearance model learned for the pose, and the right subwindow shows the weight of the Stable model in the appearance model. a. A person is not detected at this time instance because the size and shape of the target is not within the detection range. b. A tracker is allocated for him, and his appearance is learned. c,d. Tracking is successful, and the appearance model becomes more stable. e,f. Since appearance is learned per pose, the appearance model for each additional pose is also learned (note different pose in the subwindows).

detector finds the human shape and initializes tracking (Figure 4.b). In Figure 4.c the representative pose has changed due to changes of the shape of the person's silhouette. After some period of time, the appearance model learns the pixel intensities, and the weight of the Stable model has become higher (Figure 4.d). Since each pose has a separate appearance model and the pose in Figure 4.e has never been seen before, the pixel intensities for this pose start to be learned in the appearance model.

In the next sequence, we show various situations related to initialization and termination of trackers. In Figures 5.a and 5.c, the detection results do not cover the whole body due to an incomplete probability foreground map (a) and partial occlusion (c), but when complete information about the targets is available, the trackers recover correct representations of individuals (Figure 5.b,d). Figure 5.e-h show the departure of one person from the view. When the information about the target becomes insufficient or unavailable, the tracker for the individual terminates automatically (Figure 5.g). Figure 5.i-j show another departure without occlusion and the reentry of the previous target. As we do not maintain past models, the returning person is treated as a new individual. A natural extension is to store the individual models and use them in a recognition process when a new person is detected.

4.1. Effectiveness of Appearance Model

In this section, we compare the accuracy of tracking results when only the shape model is used to when both the shape and appearance models are used. Figure 6 shows frames from two tracking results, (a-e) are from the shape-only tracking and (f-j) are from the shape-appearance tracking. To show the accuracy of tracking, the upper left window in each image shows the warped images of the current grayscale frame and the mask. Note that the person is partially out of sight, so there is no information in the missing region to update the warp parameters.

Since the shape-only tracker uses just the foreground probability maps and tries to minimize the error function in (1), the warp parameter starts deviating from the correct position (Figure 6.b). Because the out-of-scene part is not considered in the parameter update, and since the tracker does not have any information about the appearance of the person, the estimation converges to the incorrect local minimum and switches to the pose which gives minimal error (Figure 6.c). In Figure 6.d-e, the pose is again corrected due to the large gap from the person to the right side of the image, but it still failed to recover the correct warp parameters.

The tracking algorithm using both models only suffers small deviations when the person returns to the center from the occluded area (Figure 6.g-h), but due to the information about the appearance, it does not deviate much. When the silhouette of the person becomes fully available (Figure 6.i), the appearance model guides the tracker to converge to the correct point.

5. Conclusion and Future Work

We introduced a tracker that uses two representations of human images. The shape model captures the shapes of various poses, and the appearance model captures the texture of each pose. Due to their intrinsic differences, we trained the shape model off-line, but the appearance is learned online while the target is being tracked. More importantly, they represent different kinds of information about the target, and therefore they offer complementary advantages for tracking. We focused the experiments of this algorithm on tracking people, but there is no restriction in extending this

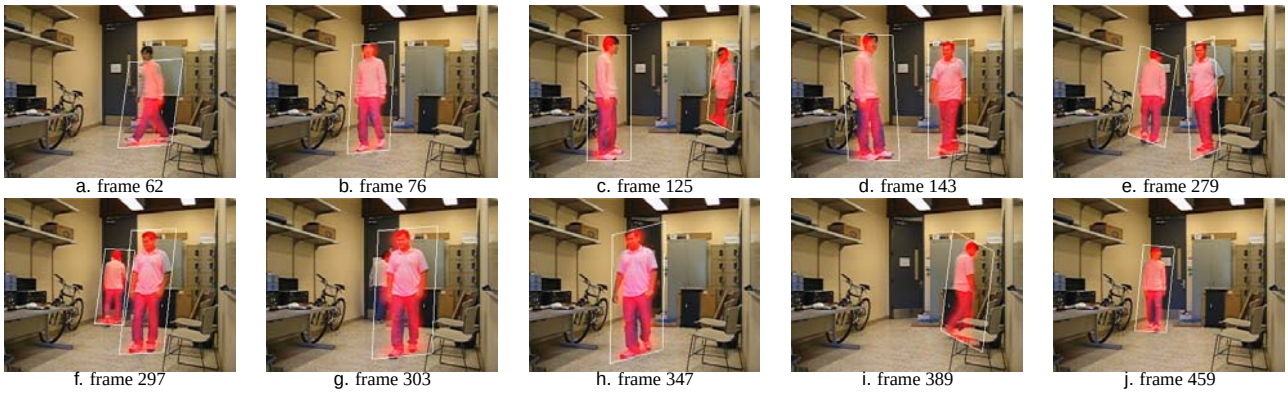


Figure 5: Tracking result for a two-people sequence. a,b. One person enters, and a tracker for him is initialized. Due to incomplete FPM, the head region is not included in the initial estimates, but the tracker recovers after several frames. c,d. The other person enters. The detector finds him very early, but due to partial occlusion, the initialization does not include the shins. The tracker recovers when the partial occlusion disappears. e-h,i. The departure of a person is automatically detected and handled by terminating the tracker for him. j. The person returns into the room, and a new tracker is initialized.

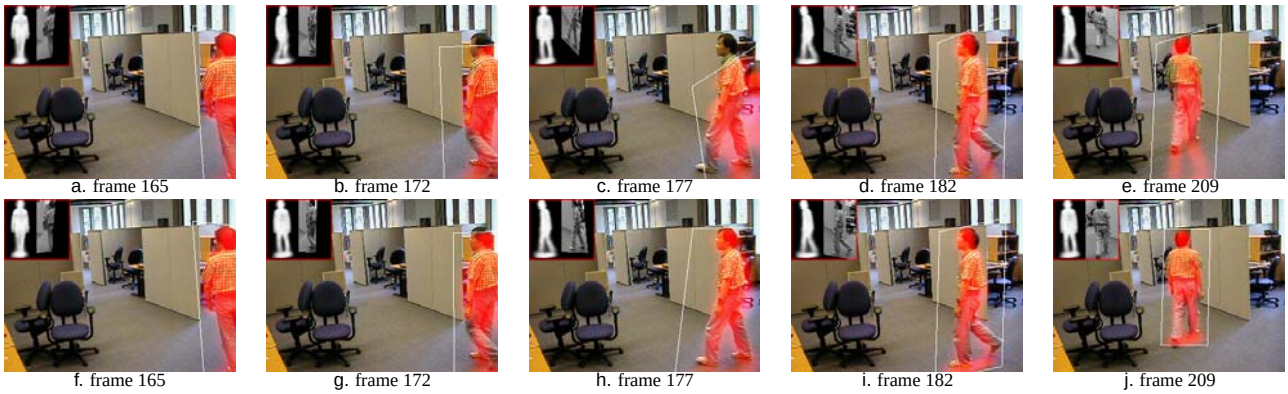


Figure 6: Comparison between shape-only and shape-appearance models (a.-e.: shape-only, f.-j.: shape-appearance). The upper left window in each image shows the pose and warped image patch from the frame. The shape-only model drifts much (c.) and fails to recover correct warp parameters (d.,e.), but the shape-appearance model can robustly track the person (note the accuracy of estimated warp in the windows of f.-j.).

work to general object tracking.

There are a number of ways to improve this approach. Based on the tracking result, we can build a layered representation of the current scene, each layer represents one object moving in the scene or occluding the other objects. The current implementation does not consider the transition probability between poses, but this can be learned during the training process and improve the tracking performance and stability. Also we will extend this work to a multi-camera situation, to track objects in 3-D space and to represent objects more precisely.

Acknowledgments

This work was funded under NSF CCR 00-86094 and the U.C. MICRO program.

References

[1] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[2] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[3] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proc. IEEE ICCV'99 FRAME-RATE Workshop*, 1999.

[4] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust on-line appearance models for visual tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 415–422, 2001.

[5] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, Jun 1994.

[6] C. Stauffer and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[7] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *IEEE International Conference on Computer Vision*, pages 50–59, 2001.

[8] K. Toyama, J. Krumm, B. Brumitt, , and B. Meyers. Wallflower: Principles and practice of background maintenance,. In *IEEE International Conference on Computer Vision*, pages 255–261, 1999.

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