Real-Time Tracking of Multiple People Using Continuous Detection [*]

David Beymer and Kurt Konolige
Artificial Intelligence Center
SRI International
Menlo Park, CA 94025
E-mail: {beymer, konolige}@ai.sri.com

Abstract

Recent investigations have shown the advantages of keeping multiple hypotheses during visual tracking. In this paper we explore an alternative method that keeps just a single hypothesis per tracked object for computational efficiency, but displays robust performance and recovery from error by employing continuous detection during tracking. The method is implemented in the domain of people-tracking, using a novel combination of stereo information for continuous detection and intensity image correlation for tracking. Real-time stereo provides extended information for 3D detection and tracking, even in the presence of crowded scenes, obscuring objects, and large scale changes. We are able to reliably detect and track people in natural environments, on an implemented system that runs at more than 10 Hz on standard PC hardware.

1 Introduction

Our goal is to simulataneously track a number of people in crowded natural environments. A system of this sort would be useful in a number of applications, such as human-computer interaction, surveillance, and mobile robots that work among people. The challenge is to find efficient methods for detecting and tracking people under fairly difficult natural conditions. Such a system should be robust enough to operate with partial occlusions of the subjects, and recover from tracking errors where a subject is temporarily lost. Finally, the techniques must be implemented on standard available hardware, and run fast enough to track in real time.

A well-known and efficient technique for tracking objects appearing in an image is to correlate a template of the object against the image. Robust tracking under lighting and object orientation change can be achieved by adapting the template, but the problem of *template drift* occurs: the adapted template moves off the desired object, either by acquiring background noise, or because the object is temporarily occluded. To some extent this problem can be ameliorated by keeping multiple hypotheses about the image location of the object, and recovering from errors as an object emerges from occlusion or is viewed at a previously-seen orientation [8]. But there is an obvious expense and complexity involved in keeping multiple hypotheses. This expense is compounded as the problem size is increased from tracking a single object to tracking multiple objects.

An obvious question arises as to how the template is acquired in the first place. In the best case, it is determined automatically by a *detection method* based on a model of the object class to be tracked [11, 14]. Detection can be an expensive operation, involving search over the image and matching against the model. The problem is complicated by the size of the search space, since the object may appear at different sizes (scale change) and orientations. Generally, once detected, an object is tracked more quickly and at a reduced computational expense.

In this paper we explore an alternative technique to multihypothesis tracking, *continuous detection and tracking*, that keeps just a single hypothesis about the image location of an object being tracked. In this method, detection and tracking play complementary roles. Detection signals the presence of a desired type of object, but does not distinguish its identity or relation to objects previously seen. Tracking determines the spatiotemporal coherence of an object, but is prone to misjudgements about the actual presence of the object. Such misjudgements occur, for example, when an adaptive tracking template picks up background information, either because the background itself is distracting, or because the object is partially obscured. The main conceptual contribution of the work is the idea that continual detection can provide enough information about object presence and location to overcome the problems associated with adaptive template tracking of a single hypothesis.

Continuous detection depends on the availability of a fast detection method. For this, we exploit the properties of video-rate stereo [9, 10, 17], which for the following reasons is a powerful modality for detecting moving objects in a cluttered scene.

- Stereo provides a stable basis for segmentation. Using background subtraction to detect change is more reliable with stereo than other methods, because the

[*] This research was supported by DARPA contract N00014-97-C-0146 through the Office of Naval Research.

presence of distracting shadows, lighting changes, and camera dynamics has little effect [2]. Because stereo returns range information, it can discriminate the three-dimensional shape of objects, a useful feature in the detection process.

- Occluding surfaces can be found and dealt with, reducing their effect. This is one of the most important properties of stereo, since occluding surfaces are a difficult problem for other techniques.
- Stereo also helps minimize the problem of scale in detection, since the apparent size of objects can be adjusted based on their perceived range.
- Stereo can provide useful state information to the tracking phase, since relevant properties such as real-world velocity can be estimated.

To demonstrate the power and practicality of continuous detection and tracking, we have implemented a system that uses stereo and adaptive correlation to detect and track many people at the same time in a crowded environment. Since the detection phase of the system is continuously active, it can detect new people that enter a scene, or re-detect people that cannot be tracked because of excessive occlusion or other failure modes. The system operates at reasonable data rates (> 10 Hz) on standard PC hardware, depending on the number of people tracked. We perform experimental validation of the system to highlight its excellent detection rate in the presence of distractors, and compare its performance to adaptive correlation without continuous detection.

2 Related Work

There is an impressive body of literature on model-based detection and tracking of people. Because video-rate stereo has only been achieved recently, only a few groups use it. Darrell et al. [1] present a system that uses different sensing modalities, including stereo range, to robustly segment and track people, concentrating on faces. Like us, they continuously segment the image on the basis of stereo range. However, unlike in our work, they do not attempt to model and detect human torsos on the basis of their shape, nor do they use correlation and Kalman filtering to track people. Instead, they rely on multiple cues such as color and face recognition to reduce errors, and various heuristics to track from frame to frame. In terms of performance, our work differs primarily in that we are addressing the harder problem of tracking multiple people with partial occlusions and distractions, while the experiments in [1] are restricted to the closest face to the cameras, which is unoccluded and not usually distracted by adjacent surfaces at a similar depth. We also track at far distances (up to 20 meters) from the cameras, where depth disambiguation can be problematic. Finally, they consider the problem of long-term re-identification of previously seen individuals, which we do not.

Other non-stereo, model-based methods for tracking people in image sequences use color [18, 3], motion [6, 16], and/or contour tracking [8, 13]. For example, the ”silhouette” method in [6] a background model is used to segment motion contours, which are then matched against an explicit model of human shape. Interestingly, the system tends to fail during occlusions generated by multiple subjects near each other, and under shadow or other lighting changes. Both these effects can be mitigated using stereo information, and the most recent implementation of the system adds this capability.

All of these systems must deal with the difficulties of occlusion and distraction. Multihypothesis methods [8, 13, 12], already mentioned in the introduction, keep enough information to recover from locally bad situations, but at the cost of added complexity and the inability to adequately recognize and deal with occluded situations as they occur. For color-based trackers, occlusion is a real problem; at least one paper attempts to explicitly model and recover from occlusion events [3]. Finally, methods such as wavelet matching [11] are inherently insensitive to occlusion, although their at the present time their performance is not as good as other systems.

3 System Overview

The system architecture, shown in Fig. 1, exhibits a tight coupling between the tracker and a continuously-operating detector. Information from a monochrome stereo head is fed into a stereo processing unit, which performs efficient area-based correlation to extract disparity information [10] at some cyclical rate, typically greater than 10 Hz. A further operation of background subtraction isolates objects that differ from a learned background. This information is fed to a recursive segmentation process, and the results are scaled at different resolutions in the stereo pyramid. Pyramid representations are used to compensate for scale differences, and to increase the efficiency of processing for certain operations.

A detector, running at each cycle on all levels of the pyramid, matches an appropriate shape model template to detect person-like objects. Detection is used in two ways: to find new people and insert them into the current state; and to register tracking templates that are following already-detected people.

The tracker utilizes intensity information, again with a pyramid structure to help with scale changes. As new people are found and presented to the tracker, it forms an adaptive tracking template for the person. At each cycle, it updates their state using a Kalman filter and information obtained by correlation of the tracking template. The tracker relies on the detector to compensate for drift as it adapts the tracking templates. Additionally, the tracker will eliminate people from the set of states if they are no longer recognized in the scene.

There are a few limitations of the system. Because we rely on range background subtraction, the camera

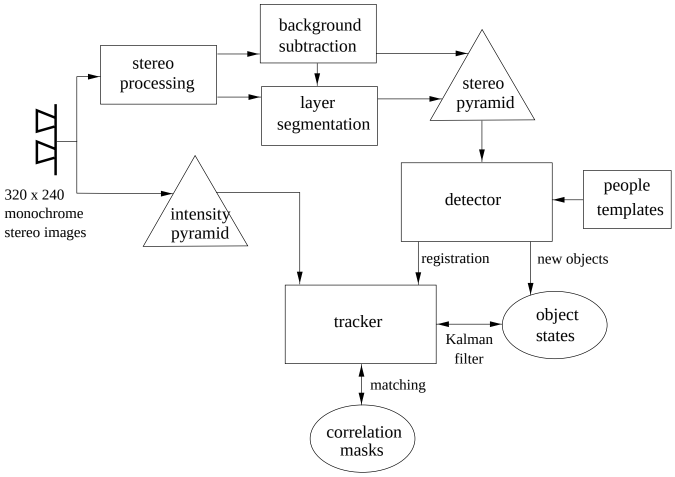


Figure 1: Flowchart for our detection and tracking system.

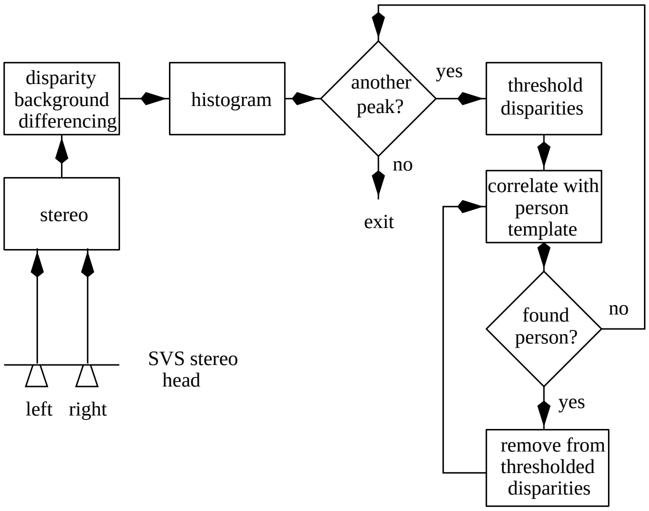


Figure 2: Flowchart for the person detection module.

cannot translate, although rotations and zooming can be accomodated [2]. Surprisingly, the system is not severly limited by the decrease of depth resolution with distance, even for small stereo baselines. For example, we are able to use stereo effectively at 20 m, with a baseline of only 11 cm; the excellent stability of background subtraction accounts for this.

In the next few sections we give detailed descriptions of the detection and tracking algorithms.

4 People Detection using Disparity Templates

The goal of the person detection module is to automatically initialize person tracks for the tracker. The main idea behind the detection module is to segment the foreground image into layers of near constant disparity. People are then located within these layers by correlating with a bank of person templates. The main steps in our person detection module are shown in Fig. 2.

4.1 Stereo background differencing and layers

Given a left and right image pair from a stereo head, we first compute stereo using the area correlation method described in [10]. The disparity image is a dense image slightly smaller than the original intensity image, where the disparity is inversely related to depth according to the formula

$$d = bf/z \tag{1}$$

where d is the disparity, b is the stereo head camera baseline, f is the camera focal length, z is the normal distance from the image plane to the object. We work with disparities rather than range because the error statistics are constant over the range of disparities.

In order to detect foreground objects in the scene, background differencing is applied to the stereo disparities [2]. While stereo involves some computational expense over using color or intensities, it does offer some advantages. First, stereo disparities are insensitive to shadows and changes in lighting conditions. Second, for two people who are adjacent in the image but at differing depths, their differing disparities will allow the system to properly segment the two. In our backgrounding computation, pixels that have a *larger* disparity than the background (i.e. closer to the camera) are initially marked as foreground. The morphological opening operator (erode followed by dilate) is applied to suppress noise and connected components are computed to retain blobs of a minimum size.

In our system, the background disparity image is computed by averaging the stereo results from an initial background learning stage where the scene is assumed to contain no people. While much work has been done on adaptive background models [2, 5], the insensitivity of stereo to changes in lighting mitigates to some extent the need for adaptation. We plan to add adaptation to deal with long-range changes such as adding/subtracting objects from the scene.

Next, the foreground disparity image is further segmented into layers of dominant disparity. This takes advantage of stereo's ability to segment objects at different depths. A histogram of the foreground disparities is computed, smoothed, and then histogram peaks are extracted as seeds for forming layers. When focusing attention on a particular layer, we gate the foreground disparities in a disparity-dependent range around the histogram peak. Fig. 3 shows (a) the left image from a stereo pair, (b) the stereo disparities, (c) the background disparities, (d) the foreground disparities, and (e) the two foreground layers.

4.2 Handling scale variation

In person finding systems that utilize no range information, image scale is an important issue. Detecting people at a range of scales typically involves search over that range (e.g. [11]) or estimation of scale from segmented blobs, which is difficult. Stereo disparity information al-

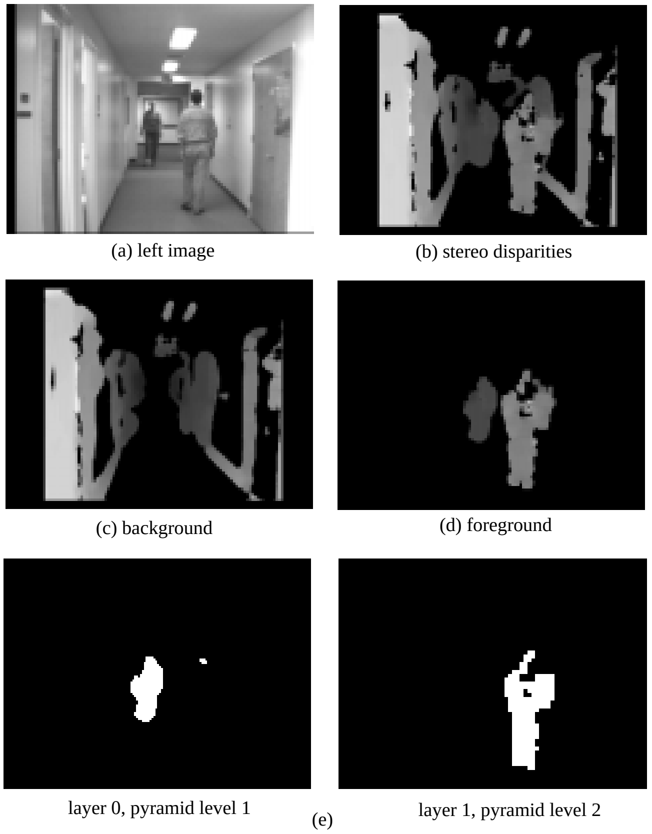


Figure 3: Computation of stereo and foreground layers.

lows us to compute the appropriate scale from a layer's dominant disparity directly. In fact, person scale is proportional to disparity.

Consider a person with width w standing a distance z from the camera as shown in Fig. 4. The person projects to a width w' in the image plane, so by similar triangles

$$\frac{z}{w} = \frac{f}{w'}$$

Cross multiplying, we get $zw' = fw$. The right hand side of this equation is constant for a given width w, so we get $zw' = \mathbf{const}$. Combining with Equation (1) to eliminate z, we get

$$w' = dK \tag{2}$$

where K is a constant that can be measured in a simple calibration step.

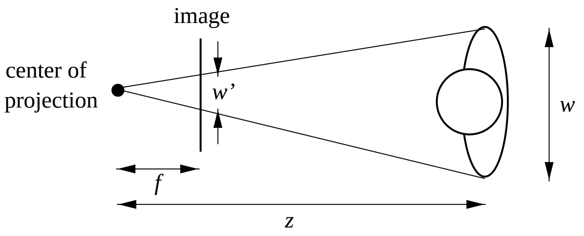


Figure 4: Person width w' in the image plane is proportional to disparity. See text for details.

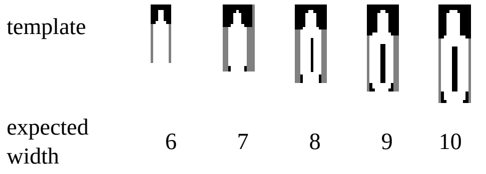


Figure 5: Templates used for person detection. The larger templates have zeros in the center since stereo may fail in the center region if the person is wearing clothing with little texture. Grey pixels are don't cares.

Since we want to operate on a large range of scales (scale change of 10:1) in real time, the detection and tracking modules operate on a 4 level pyramid. Stereo and background subtraction are performed at the highest level, but as soon as we focus on processing a layer at disparity d_0, we switch processing to **pyramid-level**(d_0), where the pyramid level is chosen to make the expected person width 8 pixels (using Equation (2)). This keeps our person templates for detection and tracking to about 12x28 pixels, which is important for speed.

4.3 Person templates

Given disparity layers capturing foreground objects at different disparities, the next step is to detect the people present in the foreground layers. This is done using correlation with binary person templates. Notice that we are doing binary correlation so the templates simply capture the 2D shape of people. As compared to, say, intensity-based appearance modeling (e.g. eigenfaces [15]), there is much less variation in the coarse 2D shape of people, especially when one operates at lower pyramid resolutions. Thus, we use a simple person model based on a small set of binary templates.

Shown in Fig. 5, the set of binary person templates used for detection differ from one another primarily in scale. These templates are designed to cover the expected scale variation in one octave of our pyramid representation. More precisely, given a foreground layer at disparity d_0, we downsample the later to pyramid level **pyramid-level**(d_0). Using Equation (2), we predict the expected width of the person and hence which template from Fig. 5 to correlate with.

The selected person template is then correlated against the downsampled layer image, using a Hamming distance metric. If the maximum correlation value is above a threshold, then we place a new person detection at the location of the correlation peak. The threshold we use for detection is approximately 75% of the number of template pixels. Since there might be multiple people present in the foreground layer, the detected person is subtracted from the layer and the correlation process repeated. In the flowchart of Fig. 2, this loop shows up in the three boxes in the lower right. Naturally, when no additional people are detected at the current disparity layer, the next disparity layer from the disparity his-

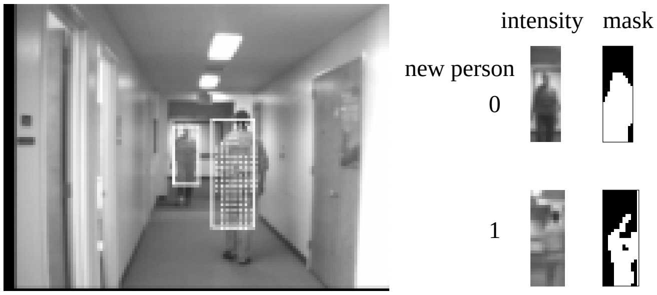


Figure 6: Results of person detector on image of Fig. 3. Two templates are extracted for each person, an intensity template and a stereo-based mask.

togram is processed.

Person detections for the image in Fig. 3 are shown in Fig. 6. When a person is detected – and hence initialized for the tracker – two templates are extracted:

1. *Intensity template.* This is used for intensity correlation in the tracker.

2. *Foreground mask.* This is a "probability" mask for pixels belonging to the person.

Exactly how thes templates are used is explained in the next section on the tracker.

5 People tracking

Over the last few years, multimodal trackers have emerged as a technique for dealing with clutter and occlusion [8]. In an ambiguous clutter or occlusion related event, the probability distribution spreads out and becomes multimodal, essentially allowing the tracker to maintain multiple tracking solutions. As the sequence evolves and the tracking becomes less ambiguous, the probability distribution ideally becomes unimodal again around the true answer.

Our tracker is multimodal in a different way, using multiple input modalities, namely intensities and stereo disparities (see also [1, 7]). While it is unimodal in terms of tracker state for an individual, it takes advantage of the segmentation aspects of stereo to achieve some of the same results as probabilistic trackers. Background clutter can be ignored by using predicted disparities to narrow search to a particular foreground disparity layer. Another advantage of using stereo is that template drift, a problem for adaptive templates, can be avoided by using the segmentation information.

5.1 Kalman filtering

Our tracker works in a Kalman filter [4] framework where the measurement process is intensity correlation and stereo. To measure person location, we use a world coordinate system (X, Z) as shown in Fig. 7, where the X-Z plane is parallel to the floor and Z measures the distance from the stereo head. This world coordinate system is preferable to an image-based $(x, y, \text{disparity}(x, y))$ since our constant velocity model for system dynamics is more appropriate for the world coordinate system (i.e. people actually move about in the world coordinate system). It is easy to map back and forth between world and image coordinates. Assuming that the stereo head is parallel to the floor, Z is related to disparity(x,y) using Equation (1) and x is related to X using the familiar perspective projection equation $x = fX/Z$.

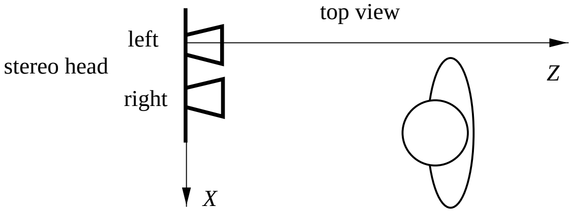


Figure 7: The world coordinate system (X, Z) is used in the state of the Kalman filter.

Our Kalman filter state is a vector $(X, Z, \dot{X}, \dot{Z})^T$, and our model for system dynamics is a constant velocity model; acceleration is modeled as noise. In the description of the tracking algorithm below, d is the depth Z mapped into image space as a disparity value. The image measurement y is also maintained for each person outside of the Kalman state; the variance on this measurement is so high we felt that it might detract from the filter performance if included.

5.2 Tracking Algorithm

In this section we give the details of the tracking algorithm. This procedure is iterated, of course, over all the people being tracked by the system. The primary novel feature here is how stereo is being used to modulate the intensity correlation and re-center the templates to avoid drift. We assume that stereo and disparity background subtraction have already been applied to the image. For the person P being tracked, let P_{templ} be the intensity template and P_{mask} be the stereo mask (as in Fig. 6).

tracking procedure
input: image intensities I_{left}
stereo foreground I_{fgnd}
person P to track
algorithm:

1. Kalman prediction. This estimates (X, Z) for the current frame. The image measurement y is predicted separately using a linear predictor.

2. Threshold the foreground disparity map I_{fgnd} in a range around the predicted disparity d. This generates a disparity layer I_{layer} containing the person P and possibly other people at disparity d (similar to Fig. 3(e)).

3. Correlate P_{templ} against the input image reduced to $\textbf{pyramid}-\textbf{level}(d)$. Each pixel in P_{templ} is weighted by P_{mask}, which eliminates the influence of background clutter from the template.

4. Refine the location from step (3) by correlating a person template against the layer image I_{layer}. This re-centers the template on the person and avoids template drift. The predicted width of the person determines which person template we select from Fig. 5.

5. Update step. Recursively update P_{templ} using I_{left} and probability mask P_{mask} using I_{layer}. Update disparity d by taking a weighted average of I_{fgnd} using P_{mask}. Also perform standard Kalman filter update steps.

6. Remove person P from the foreground mask I_{fgnd} to keep the person from being detected by the detection module.

Steps (1), (3), and (5) constitute a fairly standard correlation-based Kalman filter. Our stereo enhancements are in steps (2) and (4). Step (3) does most of the work in updating the person position, but is susceptible to template drift. Step (4) emphasizes depth discontinuity information to try to keep the templates centered. Notice also in step (3) how the stereo mask is used to focus attention on the foreground object pixels in the template.

In the recursive update step (5), the template P_{templ} is updated using the formula

$$P_{templ}(x,y) = \alpha P_{templ}(x,y) + (1-\alpha) I_{left}(x+x_p, y+y_p),$$

where (x_p, y_p) is the location of the upper left corner of the template as computed by step (4) and α is 0.75. The mask P_{mask} is updated in the same way. Also, disparity d is updated as follows:

$$d = \frac{\sum_{(x,y)} P_{mask}(x,y) I_{fgnd}(x+x_p, y+y_p)}{\sum_{(x,y)} P_{mask}(x,y)}.$$

In our detection and tracking system, new people are always being added by the detection module. Likewise, the tracker needs a criterion for eliminating people who have been occluded. When a person A is occluded, A will no longer claim portions of the foreground disparity map, so A's probability mask P_{mask} will trend down to zero. Thus, the tracker eliminates a person when

$$\sum_{(x,y)} P_{mask}(x,y) < threshold.$$

In terms of tracking multiple people, we have found that sorting and processing people from front to back helps with occlusion events. If person B hides behind person A, the algorithm will update A before B, which helps maintain the continuity of A. Since person A "claims" the disparities from the foreground image I_{fgnd} as B disappears, B will lose support and will be properly eliminated by the tracker.

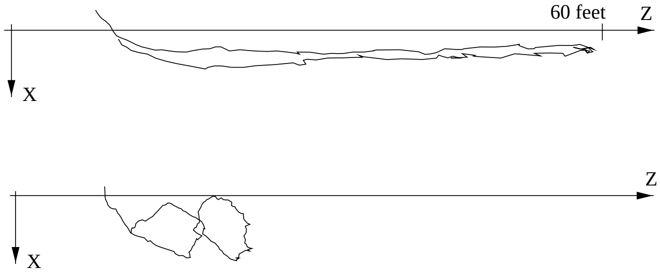


Figure 8: Two tracking results for a single person. In the upper track, the person is running up and down a hallway. In the lower track, a figure eight pattern is tracked.

5.3 Tracker plots

In this section we show plots of some tracking results; a quantitative evaluation of the detection and tracking modules is given in the next section. First, Fig. 8 shows two separate tracks for a single person. The upper sequence, a sequence where the person runs up and down a hallway, demonstrates two nice features of our tracker: (1) we can track out to 20 meters from the stereo head with a camera baseline of only 11 cm, and (2) the tracker can keep up with a running person. In the lower track, the person is walking in a figure eight pattern.

Fig. 9 demonstrates the system's ability to track multiple people. The figure shows six frames covering approximately 2 seconds from sequence of four people, three of whom are visible at any one moment. Person ID's are indicated in the upper right corner of a tracking box. Person 1 is correctly tracked, but person 2 is temporarily occluded by person 1 and then redetected as person 4 in frame 104.

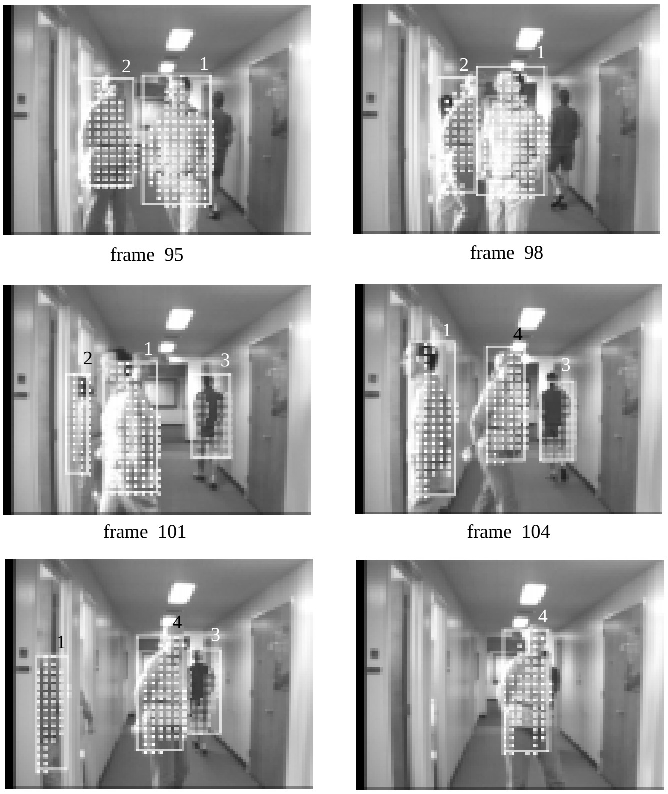
6 Experimental Results

In this section, we describe our experimental setup and quantitatively evaluate the our tracker on several sequences.

6.1 Experimental setup

Our tracker runs on standard PC hardware, a dual Pentium II processor at 400 MHz. For stereo, we use SRI's Small Vision System [10] with a camera baseline of 11 cm and FOV of 50 degrees. The disparity search is over 24 pixels in a 320x240 image, with 1/4 pixel interpolation of the results.

The stereo head is aimed down a corridor in our office building; Fig. 3 shows a typical view. The hallway extends 60 feet before reaching a wall, and we have tracked out the full 60 feet as shown in Fig. 8. At the end of its range, we are seeing disparity differences from the background of less than 1/2 pixel, showing the excellent background discrimination and the ability of stereo to work at far distances. When capturing sequences, we asked people to walk up and down the hall and in and out of offices.



frame 95 frame 98

frame 101 frame 104

frame 107 frame 110

Figure 9: Tracking results on approximately 2 seconds of video containing 4 people (one person is barely visible).

Seq	$\#People$	$\#Occl$	TR	FP	MTD
1	1	0	96%	0%	6.0
2	1	0	98%	0%	4.0
3	1	0	96%	0%	10.0
4	2	0	89%	10%	2.5
5	2	0	92%	6%	11.0
6	2	1	86%	0%	9.0
7	3	2	79%	3%	7.7
8	4	2	85%	2%	5.0
9	3	6	84%	4%	5.8
10	5	10	78%	1.3%	6.6
11	4	9	69%	5.6%	7.0
12	5	20	68%	3.2%	5.4
13	5	28	70%	6.7%	6.2

Table 1: Evaluation statistics for our tracker, including tracking rate TR, false positive rate FP and mean time to detect MTD.

In terms of distributing the computational load across both Pentium processors, we have placed the stereo computation on one and the stereo backgrounding, detection and tracking modules on the other. When tracking no people (stereo and detection processes only), the system runs at 20 Hz. When tracking one person, performance goes down to 15 Hz, and with four or five people performance drops to around 8 Hz.

6.2 Evaluation

To evaluate the tracker, we compiled statistics related to person detection, tracking, and false positives. To this end, we first collected a number of test sequences and manually defined ground truth for each sequence by specifying the center of the torso section of each person in every other frame. Then, to evaluate a sequence, we run the tracker and match up the resulting tracks with ground truth. This matching process follows a simple greedy algorithm: compute all distances between tracker centroids and ground truth, pair up the track and ground truth with smallest distance, and repeat until the smallest distance is above a threshold of about 75% of body width. Distances, of course, are normalized using disparity information from the tracker and Equation 2.

Given these matches, the *tracking rate* is defined as the percentage of all ground truths that have corresponding matches from the tracker. Likewise, the *false positive rate* is defined as the portion of tracks that have no corresponding ground truth. To evaluate the detection module, we measure the *mean time to detect*. This is the mean number of frames from the first appearance of a ground truth to the time it is detected by the tracker. Since occlusion occurs frequently in our sequences, the mean time to detect also includes the time taken to reacquire ground truths after they have re-emerged from being occluded.

Table 1 shows the tracking rate TR, false positive rate FP, and mean time to detect MTD for a set of sequences. Each sequence contains between 200 to 300 frames and covers roughly 10 to 20 seconds. Quicktime movies showing the results of 6 of these sequences are available in Movie 1 through Movie 6. The sequences are ordered roughly from easiest to most difficult, where difficulty is measured by the number of people in the sequence $\#People$ and the number of occlusion events in the sequence $\#Occl$. From the table, one can notice a steady degradation in performance as one goes from the easy sequences of tracking a single person with no occlusion to tracking 5 people with 28 occlusion events. We counted occlusions when the object eventually reappeared later in the sequence. We have also run the system continuously for hours at a time, including a demo at November, 1998 Image Understanding workshop.

To evaluate the usefulness of adding the modality of stereo to the tracker, we ran the tracker with the recentering step (4) disabled and with using normal correlation in step (3) instead of weighted correlation. We noticed much more template drift, and the mean tracking rate decreased 4% (people tended to be redetected right after the template drifted off). The mean false positive rate increased significantly – from 3% to 10% – since the tracking often double-tracks a person during drift.

7 Conclusion

In recent years the tracking community has started to emphasize tracking in the face of background clutter and partial occlusion. We have explored how to use stereo in a multimodal approach to the person tracking problem, demonstrating a detection and tracking system that runs in real time on standard PC hardware. Our system employs *continuous detection* to find new people in the scene and help keep templates centered on existing tracks. The detection module uses stereo to segment a foreground image into layers containing people. People are then localized in these layers by correlating with a bank of person templates. The detected people are tracked using correlation on intensity tempates with adjustments from stereo detection to avoid template drift. We have demonstrated the system tracking multiple people in real time while handling a large number of occlusion events.

For our future directions, the most important remaining tracking issue is that of redetection when a person is temporarily occluded. In the current system, the person is detected as a new track when this happens. Our system needs to store distinguishing features of the person, perhaps color, that would enable recognizing a person after occlusion. Also, we hope to have a add color and motion to the input modalities used for tracking – a color version of the Small Vision System will be available shortly. Finally, more detailed person models should be explored, but they will have to fit into our real time constraints.

References

[1] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 601–608, 1998.

[2] Christopher Eveland, Kurt Konolige, and Robert C. Bolles. Background modeling for segmentation of video-rate stereo sequences. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 266–271, June 1998.

[3] Paul Fieguth and Demetri Terzopoulos. Color-based tracking of heads and other mobile objects at video frame rates. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 21–27, San Juan, Puerto Rico, 1997.

[4] Arthur Gelb. *Applied Optimal Estimation.* The MIT Press, Cambridge, MA, 1974.

[5] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 22–29, June 1998.

[6] I. Haritaoglu, D. Harwood, and L. Davis. W4 - Real time detection and tracking of people and their parts. Technical report, University of Maryland, August 1997.

[7] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W^4S: A real-time system for detecting and tracking people in 2.5D. In *Proceedings of the European Conference on Computer Vision*, pages 877–892, 1998.

[8] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Proceedings of the European Conference on Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[9] Takeo Kanade, Atsushi Yoshida, Kazuo Oda, Hiroshi Kano, and Masaya Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 196–202, San Francisco, CA, 1996.

[10] Kurt Konolige. Small vision systems: hardware and implementation. In *Eighth International Symposium on Robotics Research*, pages 111–116, Hayama, Japan, 1997.

[11] Michael Oren, Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomaso Poggio. Pedestrian detection using wavelet templates. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 193–199, June 1997.

[12] Christopher Rasmussen and Gregory D. Hager. Joint probabilistic techniques for tracking multi-part objects. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 16–21, Santa Barbara, CA, 1998.

[13] S. Rowe and A. Blake. Statistical mosaics for tracking. *Image and Vision Computing*, 14(8), 1996.

[14] Kah-Kay Sung and Tomaso Poggio. Example-based learning for view-based human face detection. In *Proceedings Image Understanding Workshop*, volume II, pages 843–850, Monterey, CA, November 1994.

[15] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[16] R.P. Wildes and L. Wixson. Detecting salient motion using spatiotemporal filters and optical flow. In *Proceedings Image Understanding Workshop*, pages 349–356, Monterey, CA, 1998.

[17] J. Woodfill and B. Von Herzen. Real-time stereo vision on the PARTS reconfigurable computer. In *IEEE Symposium on Field Programmable Custom Computing Machines*, pages 242–250, April 1997.

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