Third IEEE International Workshop on Visual Surveillance, July 1, 2000, Dublin, Ireland

Multi-Camera Multi-Person Tracking for EasyLiving

John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, Steve Shafer
Microsoft Research Vision Technology Group
Microsoft Corporation
Redmond, WA, USA

Abstract

While intelligent environments are often cited as a reason for doing work on visual person-tracking, really making an intelligent environment exposes many real-world problems in visual tracking that must be solved to make the technology practical. In the context of our EasyLiving project in intelligent environments, we created a practical person-tracking system that solves most of the real-world problems. It uses two sets of color stereo cameras for tracking multiple people during live demonstrations in a living room. The stereo images are used for locating people, and the color images are used for maintaining their identities. The system runs quickly enough to make the room feel responsive, and it tracks multiple people standing, walking, sitting, occluding, and entering and leaving the space.

Keywords: multi- person tracking, multiple stereo/color cameras, intelligent environment

1. Practical Problems of Tracking People in Rooms

We are developing an intelligent environment called EasyLiving. Our goal is to create a software architecture and supporting technologies that aid everyday tasks in indoor spaces with unobtrusive computing. For instance, one of our demonstrations has a person sitting on a couch watching a movie. When the person leaves the couch, the movie pauses until he or she comes back. Our project requires work in distributed computing, geometric modeling, and sensing. Our main accomplishments to date have been shown in a series of live demonstrations in our offices and living room lab, shown in Figure 1. Our project's main sensing modality is computer vision, which we use to determine the location and identity of people in a room. This paper describes our vision system, which uses multiple color stereo cameras to track multiple people simultaneously.

Knowing the location and identity of people in the room is a vital prerequisite for many of the most compelling services that an intelligent environment can provide. These services include:

- Triggering events based on location, such as the couch example and above.
- Locating the right device to play an instant message, either audio or video, to a particular person.
- Invoking a particular user's preferences, such as lighting or audio, in a certain room.
- Understanding a person's behavior in order to assist him or her.

With the output of our visual tracking system, we wrote a number of programs to demonstrate EasyLiving. One program is a game, called "Hotter/Colder", in which a person uses a mouse to secretly select a point on a map of the room. Another person enters the room and tries to find the point by walking from place to place, with the room issuing spoken clues such as "You are cold" and "You're getting warmer". Another program lets a user carry a wireless mouse to different tables in the room. On each table, the mouse's moves and clicks will be rerouted to the computer that controls the display nearest that table. A third program projects on the wall one pair of cartoon eyes for each person in the room. The eyes follow the person as he or she moves around. A fourth program automatically starts and stops a VCR or DVD movie when a person sits on or stands up from a couch. The movie is automatically rerouted to different displays in the room depending on where the person sits.

Our live demonstrations let us experience what an

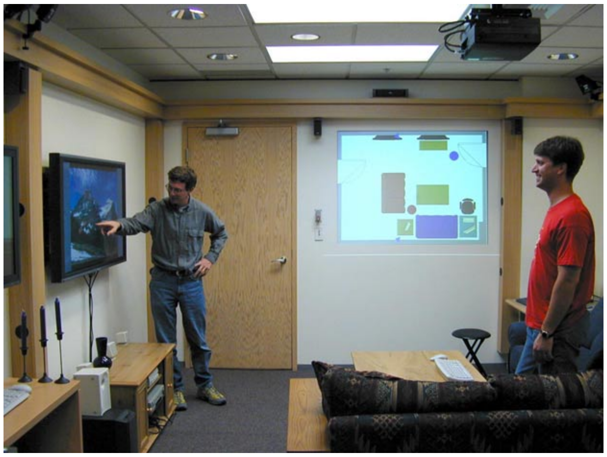

Figure 1: We track multiple people using stereo cameras in the EasyLiving lab, which is an intelligent environment set up to look like a living room.

intelligent room really feels like, and they also force us to confront many practical issues simultaneously. For a vision-based tracking system to support a real-life intelligent environment, it must:

1. Maintain the location and identity of people. In general, behaviors of the room are more compelling when this data is known accurately. Our system measures location to roughly 10 cm on the ground plane, and it maintains the identity of people based on color histograms taken as they move around the room.
2. Run at reasonable speeds. A vision update rate below 1 Hz, combined with other processing delays, makes the room feel sluggish and error-prone. Our system, running on three PCs, updates location and identity at about 3.5 Hz.
3. Work with multiple people. With perhaps only one exception, all types of rooms are sometimes occupied by more than one person. We regularly track two people simultaneously for demonstrations, and the system works well with three.
4. Allow creation and deletion of people representations. It is generally impossible to predict who will be entering a room. Our system automatically creates new instances of people as they cross through a special region in the room. It deletes instances of people that have not been seen for a certain period.
5. Work with multiple cameras. No camera can see around corners, so multiple cameras are required for tracking people in general rooms. We use two sets of color stereo cameras to visually cover our living room lab.
6. Use cameras in the room. Using cameras looking down from high overhead simplifies the tracking problem, but is impractical in most rooms. Our cameras are mounted on the wall at a height of approximately 2.3 meters.
7. Work for extended periods. It is not enough to process just a few thousand frames of video to show practical tracking. Our tracker can track multiple people indefinitely.
8. Tolerate partial occlusions and variable postures. People sometimes walk behind tables and chairs, and they sit and stand. Our system maintains tracking despite these wide variations in people's appearance.

2. The State of the Art

One way to achieve all the requirements of practical tracking in real-world environments is with active badges. Originating with work on infrared-transmitting badges at Olivetti Research[1] and Xerox PARC[2], active badges are small, electronic devices worn by people. The badges transmit an ID signal to receivers placed around the building. The ID signal corresponds to the identity of the badge's wearer, and the received signals are used to compute the wearer's location. The Olivetti initiative is continuing with ultrasonic badges at AT&T Laboratories in Cambridge, UK[3]. There are also commercial asset-tracking systems, like the radio frequency tags from PinPoint (http://www.pinpointco.com/) and wired and unwired motion trackers from Ascension Technology (http://www.ascension-tech.com/) and Polhemus (http://www.polhemus.com/). However, it is not clear that regular consumers would be willing to don any sort of device to interact with their intelligent environment. Tracking based on cameras, while not yet as reliable, has the advantage of leaving the users unencumbered. In addition, having cameras in a room is also valuable for modeling the room's geometry, cataloging its contents, and detecting unbadged occupants.

The area of vision-based tracking is very active, with work in face tracking, gesture understanding, body-part tracking, and whole-body tracking. We concentrate here on work in multi-person whole-body tracking. The work most closely related to ours is that of Haritaoglu and Davis, with a series of outdoor person-trackers named W^4[4] (separated people, grayscale camera), W^4S[5] (separated people, stereo camera), and Hydra[6] (clumped people, grayscale camera). Combined, these trackers fulfill most elements in our list of requirements in the section above, with the only exceptions being tolerating variable postures and using multiple cameras. Their systems include grayscale appearance modeling, the ability to add and delete people over time, and the ability to track people whose silhouettes overlap (Hydra).

Another closely related system is that of Darrell *et al.*[7] who, in addition to using color and stereo like our system, also use face detection. They were able to evaluate the relative effectiveness of these three cues.

In the tracking system presented by J. Orwell *et al.*[8], multiple cameras track multiple people walking in a parking lot. A software agent is created for each person detected in each camera. Reasoning about trajectory geometry in the ground plane of the parking lot, the agents communicate to determine whether or not they are assigned to the same person being seen from different cameras.

In their part of the DARPA VSAM project, CMU has created an elaborate system for video-based surveillance[9]. Using multiple pan/tilt/zoom cameras, their system classifies and tracks multiple people and vehicles as they move about outdoors.

Rosales and Sclaroff[10] describe a multi-person tracking system that unifies object tracking, 3D trajectory estimation, and action recognition from a

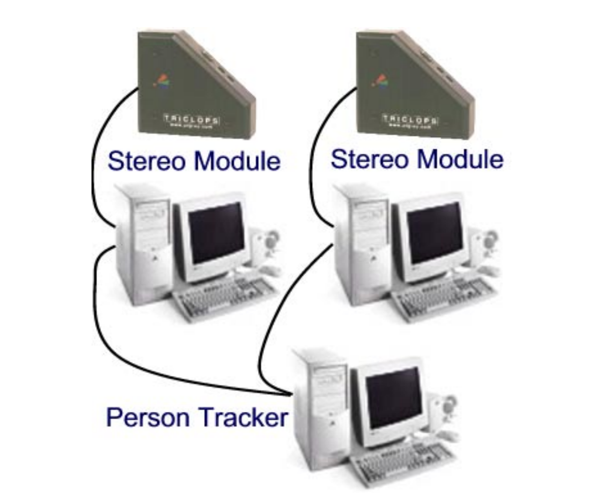

Figure 2: Hardware for tracking consists of two Triclops stereo cameras each connected to a PC running our stereo module. The two Stereo Modules both communicate with our Person Tracker on a third PC.

single video camera. It uses an extended Kalman filter for computing trajectories, which are in turn used to reason about occlusion. Kettnaker and Zabih[11] have developed a system that reasons about trajectories at a higher level using a Bayesian formulation to compute likely paths of multiple people as seen occasionally from separate cameras in a building's hallways.

In their "Closed-World" scheme for tracking multiple people, Intille and Bobick[12] and Intille, Davis, and Bobick[13] maintain local contexts that help track individual blobs and a global context for understanding the state of the whole space. A context is a set of constraining assumptions covering a specific time period and region that aid tracking.

Rehg, Loughlin, and Waters[14] present a multi-person tracking system for an interactive kiosk that uses a pair of widely space color cameras. Like us, they use color and stereo for tracking.

Omnidirectional cameras are attractive for tracking because of their wide coverage. Boult *et al*.[15] track multiple, camouflaged soldiers from an omnidirectional camera. They maintain tracks from frame to frame using spatial proximity and similarity of simple features. Stiefelhagen *et al*.[16] track multiple meeting participants around a table using skin color to detect faces from an omnidirectional camera placed on the table.

A statistical representation of tracking gives a principled method of dealing with uncertain data and multiple targets. MacCormick and Blake[17] describe a modification of CONDENSATION tracking that incorporates an exclusion principle to keep multiple head tracks from coalescing onto one head. Cai and Aggarwal[18] use a Bayesian technique to match features on human figures between frames and between multiple cameras.

Halevi and Weinshall[19] present a novel tracking algorithm called "motion of disturbances" which uses temporal differencing to create an image of "waves" showing tracks of multiple moving objects.

Using an articulated 3D model of the human body, Gavrila and Davis[20] show how to track a pair of people dancing closely using image sequences from multiple viewpoints.

Clearly there are many different approaches to tracking multiple people. Each researcher works under a different set of constraints, and there are few widely agreed-upon principles.

3. The EasyLiving Tracker

We designed our tracking system to support demonstrations of an intelligent environment. Our laboratory is set up like a living room, with two couches, a coffee table, an entertainment center, a PC, and various flat panel displays. The room is shown in Figure 1. The output of the person tracker is the locations and identities of people in the room. By "identity" we do not necessarily mean the absolute identity of the person, but rather that the tracker maintains a consistent, internally generated ID for each person during each run of the program. This ability to do identity maintenance means that the same person is recognized as the same person wherever he or she is in the room.

The remainder of this section describes the parts of our tracker and how they work together. In summary, our tracker uses two color Triclops stereo cameras (Point Grey Research, http://www.ptgrey.com/), each connected to its own PC. Using the registered depth and color images from the cameras, we do background subtraction to locate 3D blobs in each camera's field of view. These blobs, which are normally broken up over the regions of peoples' bodies, are merged into person shapes by examining the space of blob clusterings. The program that takes the stereo camera output and locates people-shaped blobs is called the Stereo Module, and there is one instance of this program running for each of the two Triclops cameras. Each Stereo Module reports the 2D ground plane locations of its person-shaped blobs to a tracking program, called the Person Tracker, on a third PC. The relationship between the cameras, PCs, and programs is shown in Figure 2. The Person Tracker uses knowledge of the two cameras' relative locations, fields of view, and heuristics on the movements of people to produce a final report on the locations and identities of people in the room. We also maintain color histograms of the person-shaped blobs, which the tracking program uses to disambiguate people when they are close together.

3.1. Stereo Cameras and Calibration

We chose to use stereo cameras rather than regular color cameras to make it easier to segment people in the room. If the regions projected onto the image from two people overlap, it is very difficult to segment them correctly using only a color image, while a depth image from stereo makes it relatively easy. Each of our two Triclops cameras contains three small color cameras. The PC software bundled with the stereo cameras computes dense disparity images of size 320x240 pixels at a rate of about 4 Hz on a 450 MHz PC. The Triclops software also reports the textureless regions of the image for which disparity could not be reliably computed. Since the cameras inside each Triclops are color, we also get a color image that is registered with the disparity image. We describe later how we use histograms from the color image for identity maintenance.

Two stereo camera units are required to adequately cover the demonstration space in the room, which requires calibrating the relative position and orientation of the cameras. Since the Stereo Module reports only the ground plane locations of blobs, it is only necessary to know the cameras' relative position and orientation in the ground plane. We have two techniques for making this measurement. One is an interactive program that lets a user click on points in images from the two cameras to establish correspondences and ground plane points. The other technique starts by recording, from each camera, the 2D ground plane locations of a person's path as he or she walks around the room. A calibration program then computes the translation and rotation that give the best overlap between the two recorded paths. A typical pair of paths is shown in Figure 3 both before and after calibration. Since the path can sometimes go beyond the field of view of either of the cameras, this program is robust to missing data. The result of both of these calibration techniques is a translation and rotation that is applied to ground plane reports from one camera to put them in the same ground plane frame as the other camera.

3.2. Background Subtraction

We segment human figures from the background with background subtraction in both depth and color. One of the main benefits of using depth images is that they are relatively insensitive to shadows and other changes in illumination, which tend to confound purely color-based background subtraction. We model the background by computing the mean and variance for each pixel in the depth and color images over a sequence of 30 frames with the room empty. For the color pixels these statistics are calculated separately for each of the RGB components.

To compute a foreground image we consider the depth data first. All pixels with invalid depth values (generally due to lack of texture) in the current live image are not considered part of the foreground. This is because people generally have enough texture on their clothes and skin to give valid depth values. The remaining pixels are only considered part of the foreground if at least one of the following applies:

- The mean depth of the corresponding pixel in the background model is not valid. This is the case where a valid depth pixel appears over an invalid depth pixel in the background.
- The mean depth of the corresponding pixel in the background model is valid and the depth pixel in the live image is outside of the tolerance range of the background depth pixel. The tolerance range is set based on a multiple of the standard deviation in depth. This is the case of normal depth image background subtraction.
- Any one of the color components in the live color image is outside of the tolerance range of the corresponding color component of the background color pixel. This is the case of normal color background subtraction.

These rules are designed to work in the nominal case of a person walking around in a static room, and also in the frequent case of a person sitting on a couch and sinking back into the cushions, thus becoming unnoticeable in depth. In this case the color component of the background subtraction will ensure that the person is still considered part of the foreground. The rules are not as good in the case of a person walking in front of an animated video display screen. We are careful to turn on the room's display screens during background modeling in order to get valid depth values in those regions. According to our rules above, a display showing animation will be considered part of the foreground even though its depth values are identical to the background's. We can normally eliminate this problem in our blob processing, described in Section 3.3. We note that the rules above work well for displays

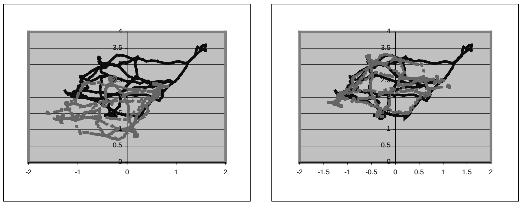

Figure 3: Left plot shows path of person on ground plane from two cameras before calibration. Right plot shows calibration that best aligns the two paths.

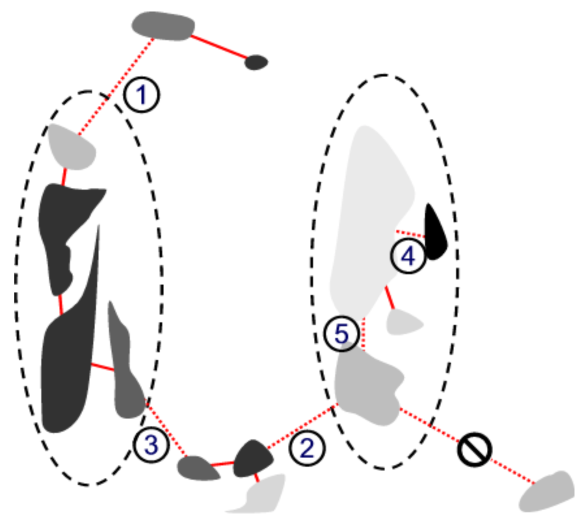

Figure 5: Blobs are grouped to form people-shaped clusters of blobs.

that turn uniform (including completely dark) during a live run. In this case the live depth values will become invalid and the screen will be kept in the background.

Beymer and Konolige[21] use only disparity (and not color) for background modeling. Gordon *et al.*[22] give a sophisticated algorithm for combining both.

3.3. Making People-Shaped Blobs

Background subtraction gives a list of pixels, some subset of which likely correspond to people in the image. We grow blobs by connecting four-connected pixels whose disparities are within a certain, small range. There are usually several blobs falling on each person on the image, rather than the more favorable case of one large blob per person. This breakup is due to image noise and missing disparities from textureless regions on people.

We group blobs into people-shaped clusters of blobs by searching through the space of possible groupings. A sketch of this process appears in Figure 5. We test each hypothesized grouping against a person-model, which is simply the expected height and width of a person.

Rather than test all possible blob groupings, which would take too long, we start with a grouping based on the minimum spanning tree, shown in Figure 5. The minimum spanning tree is the graph that spans all the blobs with the minimum sum of arc lengths. To measure arc length between two blobs, we connect the 3D centroid of the blobs with a line segment. The length is the Euclidian distance between the point where the line leaves one blob and enters the other.

The first step in generating hypothesized clusters is to eliminate all links whose length is greater than a threshold, as shown in Figure 5 with the linked marked ⊘. The next step is to mark the n longest links, shown in the figure as ①-⑤. We chose $n=5$ as a good compromise between speed and correctness. This set of n longest links are broken in each of its 2^n possible combination to generate 2^n hypothesized blob clusterings.

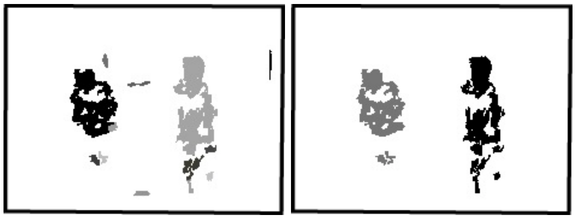

Figure 4: Left image shows raw blobs after background subtraction, and right image shows blobs grouped into two people-shaped clusters with spurious blobs eliminated.

We evaluate a hypothesized clustering by first computing a 3×3 covariance matrix of the 3D (x, y, z) coordinates of each of the clustering's constituent connected blobs. For instance, one of the 2^n hypothesized blob clusters breaks all the marked links in the figure. In this case there would be seven blob clusters, and we would compute a covariance matrix for each of the seven. We evaluate each cluster of blobs based on the two largest eigenvalues, λ_1 and λ_2, of the covariance matrix, which are in rough proportion to the length of the two largest axes of an ellipsoid surrounding the 3D points. The eigenvalues are insensitive to the rotation of the ellipsoid and easy to compute. If the product of the eigenvalues, $\lambda_1\lambda_2$, is below a threshold, the cluster is judged as too small and eliminated from further consideration. This tends to eliminate random outlier blobs and blobs that come from small changes in the background like a moved pillow. The clusters that survive this test are likely to represent people. In our example figure, the clusters outlined in dotted ellipses would come up as surviving clusters after breaking links ①, ②, and ③ (and ⊘). The first two eigenvalues of the c surviving clusters are compared with ideal values λ_1^* and λ_2^* using

$$d=\sum_{i=1}^{c}\left[\left(\lambda_{i,1}-\lambda_1^*\right)^2+\left(\lambda_{i,2}-\lambda_2^*\right)^2\right]$$

Here $\lambda_{i,1}$ and $\lambda_{i,2}$ are the two eigenvalues from the i^{th} cluster. Of the 2^n hypothesized clusterings, the one with the smallest value of d is chosen for further processing. An actual instance of this clustering procedure is shown in Figure 4.

The centroids of the people-shaped clusters are projected into the camera's ground plane and then reported from the Stereo Modules to the Person Tracker, described in Section 3.5.

3.4. Histograms for Identity Maintenance

Each of the two Stereo Modules maintains a color histogram model for each person in the room. These color histograms are used by the Person Tracker to disambiguate certain configurations of people when spatial tracking is not enough. (The Person Tracker's use of color histograms is described in Section 3.5.)

The Triclops camera produces spatially and temporally registered images of disparity and RGB. We use a coarsely quantized version of the RGB image for color histograms. For a person-shaped cluster of depth blobs, we histogram the color pixels corresponding to the blob regions. The RGB axes are each quantized into four equal-length ranges, giving a $4\times4\times4$ color cube and a 64-bin color histogram. This coarse quantization reduces the effects of spatially varying illumination color in the room.

Each Stereo Module maintains a grid of 10×10 square cells on the ground plan. It receives reports of people's identity and location from the Person Tracker. When one of these reports places a person in a cell they have not yet visited, the Stereo Module stores a new histogram for that person in that cell. Thus there are several histograms stored for each person in different parts of the room, reducing the effect of spatially varying illumination color and intensity.

Each Stereo Module compares histograms from the current image to stored histograms using histogram intersection[23], which measures how well one histogram accounts for the counts in another histogram. The intersections are normalized such that the maximum possible intersection is one. These normalized comparisons are reported to the Person Tracker.

CMU's VSAM tracker[9] uses frame-to-frame matching of blobs based on the blob's trajectory, image template, size, and histogram, all in image coordinate frames of each camera. While we also maintain separate sets of histograms for each camera, we maintain trajectories in a single global frame, as described in the next section.

3.5. Tracking People

As described in Section 3.3, the Stereo Module reports the 2D ground plane location of people-shaped clusters of blobs to the Person Tracker module. The Person Tracker runs on a PC that is separate from the two PCs that run the two Stereo Modules, and communication between the PCs is via DCOM.

An incoming report from a Stereo Module triggers the Person Tracker to perform an update. In order to be able to compare reports, the Person Tracker transforms each one into a common 2D coordinate frame based on the calibration described in Section 3.1. The Person Tracker maintains a list of tracks, each corresponding to one person in the room. Each track contains a history of the person's past locations, from which we compute a velocity that is used to predict the person's current location. The Person Tracker searches the area around the predicted location for new reports from the Stereo Modules. If there is more than one report in the area, then the Stereo Module is called to match each reported cluster of blobs against its stored histograms. The response is used to disambiguate the reported clusters. This process iterates until all active person tracks are matched to reported clusters.

Occasionally, there are person tracks for which there are no reported clusters nearby. Since the lack of a match is likely caused by occlusions or errors from the Stereo Module (rather than a person actually disappearing from the middle of the room), any unsupported person tracks are left active until they have been unsupported for a long period. This makes the Person Tracker robust to occasional dropouts from the Stereo Modules.

Once the Person Tracker has finished the matching phase, it reports the matches back to the reporting Stereo Module. This allows the Stereo Module to update any stored histogram information with the actual person's identity.

Finally, the Person Tracker computes a new location for the person. The non-uniform size of the stereo disparities and the low rate of reporting results in

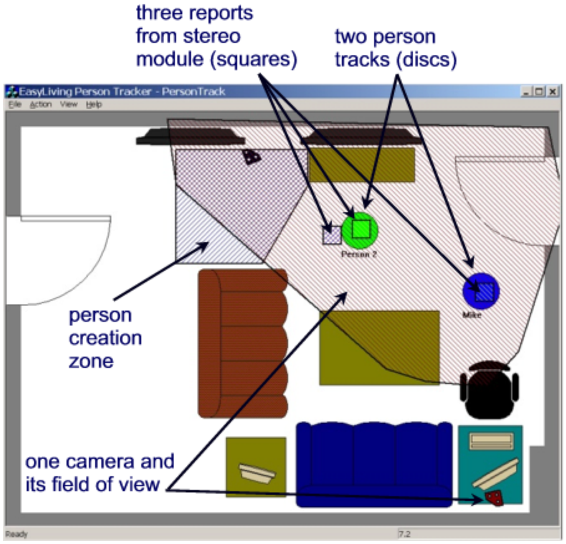

Figure 6: The Person Tracker displays the locations and identities of tracked people on a hand-drawn map of the room. It also shows where the cameras' fields of view intersect the ground plane and the locations of the people-shaped blob clusters reported from the two Stereo Modules.

discontinuities in the location data. To account for this the Person Tracker filters noise and averages each person's velocity. It then uses the average velocity to update the person's location information instead of the actual reading.

In order for the system to start tracking new people, it uses a person creation/deletion zone. This area represents valid routes for people to enter and leave the room. The person creation zone may be at the door or even just the edge of the field of view of the camera. This region is evaluated for any reported person clusters. If the cluster doesn't match any of the existing person tracks, then a new temporary track is created and monitored over time. If it continues to be supported by reports from the Stereo Module, it is converted into a new person track, otherwise it is deleted. Likewise, if a valid track enters the zone and then disappears from view, it is removed from the list of active tracks. This zone keeps the system from mistakenly creating new person tracks from extraneous blobs due to partial occlusions, moved furniture, or a coat left on the couch. Our person-creation zone is similar in concept to Intille *et al*'s[13] closed-world assumption of only allowing new people to enter and exit the space through a door.

4. Performance

Our person tracking system supports live demonstrations in our EasyLiving lab, which is set up to look like a residential living room. These demonstrations typically last about 20 minutes, with the person tracking software running continuously over the duration. The Person Tracker produces new results at a rate of about 3.5 Hz, limited ultimately by the speed of the stereo processing in the Stereo Modules. During the demonstration, people enter and leave the living room, with their tracks being created and deleted appropriately. Tracking works well with up to three people in the room, depending on how they behave. With more than three people moving around, the frequent occlusions cause enough poor clusterings in the Stereo Module that the Person Tracker cannot maintain coherent tracks. We do not require the demonstrators to wear special clothes, although similarly colored outfits can cause tracks to be misassigned due to indistinguishable histograms. The demonstrators can walk around, stand still, sit, and brush against each other without the system loosing track of them. There are also large areas of moving video in the cameras' fields of view that the tracking system tolerates easily.

5. Conclusion

This paper describes our person-tracking system for our EasyLiving research project. The system reports the location and identity of people in a typical living room environment using images from two sets of color stereo cameras mounted on the room's walls. Because we give live demonstrations, and because we are trying to create practical scenarios, we were prevented from making many simplifying assumptions. The system runs fast enough to make the room feel responsive. It works with multiple people as they walk, stand, and sit. People can enter and leave the space as the demonstration proceeds. Overall, our tracking system performs reliably, and we have used it to build demonstrations that let us explore issues of software architecture, geometric representation, and user interfaces for practical intelligent environments.

6. References

[1] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems*, vol. 10, pp. 91-102, 1992.

[2] N. Adams, R. Gold, B. N. Schilit, M. Tso, and R. Want, "An Infrared Network for Mobile Computers," presented at USENIX Symposium on Mobile and Location-independent Computing, Cambridge, MA, 1993, pp. 41-52.

[3] A. J. Andy Ward, Andy Hopper, "A New Location Technique for the Active Office," *IEEE Personal Communications*, vol. 4, pp. 42-47, 1997.

[4] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^4: Who? When? Where? What?," presented at Third International Conferrence on Face and Gesture Recognition, Nara, Japan, 1998.

[5] I. Haritauglu, D. Harwood, and L. S. Davis, "W^4S: A Real-Time System for Detecting and Tracking People in 2 1/2 D," presented at Fifth European Conference on Computer Vision, Freiburg, Germany, 1998.

[6] I. Haritaoglu, D. Harwood, and L. S. Davis, "*Hydra*: Multiple People Detection and Tracking Using Silhouettes," presented at International Conference on Image Analysis and Processing, Venice, Italy, 1999, pp. 280-285.

[7] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking Using Stereo, Color, and Pattern Detection," presented at IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998, pp. 601-608.

[8] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G. A. Jones, "A Multi-agent Framework for Visual Surveillance," presented at International Conference on Image Analysis and Processing, Venice, Italy, 1999, pp. 1104-1107.

[9] R. T. Collins, A. J. Lipton, and T. Kanade, "A System for Video Surveillance and Monitoring," presented at American Nuclear Society Eighth

International Topical Meeting on Robotics and Remote Systems, Pittsburgh, PA, 1999.

[10] R. Rosales and S. Sclaroff, “3D Trajectory Recovery for Tracking Multiple Objects and Trajectory Guided Recognition of Actions,” presented at IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, 1999, pp. 117-123.

[11] V. Kettnaker and R. Zabih, “Bayesian Multi-camera Surveillance,” presented at IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, 1999, pp. 253-259.

[12] S. S. Intille and A. F. Bobick, “Closed-World Tracking,” presented at Fifth International Conference on Computer Vision, Cambridge, MA, 1995, pp. 672-678.

[13] S. S. Intille, J. W. Davis, and A. F. Bobick, “Real-Time Closed-World Tracking,” presented at IEEE Conference on Computer Vision and Pattern Recogntion, Puerto Rico, 1997, pp. 697-703.

[14] J. M. Rehg, M. Loughlin, and K. Waters, “Vision for a Smart Kiosk,” presented at IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, 1997, pp. 690-696.

[15] T. E. Boult, R. Michaels, X. Gao, P. Lewis, C. Power, W. Yin, and A. Erkan, “Frame-Rate Ominidirectional Surveillance & Tracking of Camouflaged and Occluded Targets,” presented at Second IEEE Workshop on Visual Surveillance, Fort Collins, CO, 1999.

[16] R. Stiefelhagen, J. Yang, and A. Waibel, “Modeling People's Focus of Attention,” presented at IEEE International Workshop on Modelling People, Kerkyra, Greece, 1999, pp. 79-86.

[17] J. MacCormick and A. Blake, “A Probabilistic Exclusion Principle for Tracking Multiple Objects,” presented at Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 572-578.

[18] Q. Cai and J. K. Aggarwal, “Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized Video Streams,” presented at Sixth International Conference on Computer Vision, Bombay, India, 1998, pp. 356-362.

[19] G. Halevi and D. Weinshall, “Motion of Disturbances: Detection and Tracking of multi-Body non-Rigid Motion,” presented at IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico, 1997, pp. 897-902.

[20] D. M. Gavrila and L. S. Davis, “3-D Model-Based Tracking of Humans in Action: A Multi-View Approach,” presented at IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 1996, pp. 73-80.

[21] D. Beymer and K. Konolige, “Real-Time Tracking of Multiple People Using Continuous Detection,” , 1999, http://www.ai.sri.com/~konolige/tracking.pdf.

[22] G. Gordon, T. Darrell, M. Harville, and J. Woodfill, “Background Estimation and Removal Based on Range and Color,” presented at IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, 1999, pp. 459-454.

[23] M. J. Swain and D. H. Ballard, “Color Indexing,” *International Journal of Computer Vision*, vol. 7, pp. 11-32, 1991.