Towards Vision-based 3-D People Tracking in a Smart Room

Dirk Focken, R. Stiefelhagen
Interactive Systems Laboratories
Universität Karlsruhe (TH), Germany
focken@ira.uka.de, stiefel@ira.uka.de

Abstract

This paper presents our work on building a real time distributed system to track 3-D locations of people in an indoor environment, such as a smart room, using multiple calibrated cameras. In our system, each camera is connected to a dedicated computer on which foreground regions in the camera image are detected. This is done using an adpative background model. These detected foreground regions are broadcasted to a tracking agent, which computes believed 3-D locations of persons based on the detected image regions. We have implemented both a best-hypothesis heuristic tracking approach as well as a probabilistic multihypothesis tracker to find the object tracks from these 3-D locations. The two tracking approaches are evaluated on a sequence of two people walking in a conference room recorded with three cameras. The results suggest that the probabilistic tracker shows comparable performance to the heuristic tracker.

1 Introduction

Keeping track of people is a vital topic in smart environment research. The context knowledge gained from peoples' positions can help a lot to predict what people might expect from a smart environment or to enable other sensors to focus on areas where people are. As an example a person standing next to a white board and several people located around a table provides enough evidence to guess that people attend a presentation or lecture. Furthermore the location and number of people in a room also is a useful feature for activity classification.

For indoor environments several tracking approaches are possible: Vision based methods [7, 4, 6], speaker localization [12], or simply attaching tracking badges to people [10].

As attaching physical devices to people is undesirable and speaker localization only works if people are talking, vision based tracking still is and will be an important problem to solve.

Most of vision based tracking research has been done with sequences from a single perspective: Haritiaoglu's W^4 tracked several people in real time with a single camera [4], Darell [3] used a stereo camera to track faces and the well known pfinder system by Wren [11] tracked body parts of single users with one camera.

Considerably less work was published on tracking humans with multiple cameras. This might be due to the fact that the correspondence problem among features from different perspectives creates a lot of difficulties for tracking algorithms. On the other hand, multiple perspectives help to solve ambiguities caused by occlusion or segmentation errors and provides 3-D information.

To help solve the correspondence problem most of the research on tracking with multiple cameras used fully calibrated sensors. Cai and Aggarwal [2] track points on the medial axes of humans which are brought to correspondence using epipolar line constraints between multiple perspectives.

Mikic et al. [7] used in the AVIARY project multiple calibrated cameras to track people in real time by matching foreground regions obtained from background subtraction exploiting geometric constraints between multiple views.

In more recent work Krumm et al. [6] used two stereo cameras to track people in a living room identifying people with color histograms.

2 Overview of the tracking system

Our tracking system was mainly inspired by the work in [7] using the same triangulation technique from centroids of foreground regions. However, our system uses a probabilistic tracking approach instead of a best-hypothesis Kalman filter based method. In addition, a more sophisticated background subtraction algorithm is used.

Our system is designed as a distributed sensor network consisting of several low level and one high level component connected through the network:

The low level component is the background subtraction module. At each camera one instance of this module produces a feature stream which is sent to the high level component, the tracking agent. The tracking agent collects these feature streams to produce 3-D tracks of objects in the scene. Figure 1 shows a typical set up of the tracking system.

To ensure a consistent view of the scene, all the data from the lower level components is time stamped and the participating machines in the sensor network are synchronized using NTP [8].

Furthermore, in an initial setup step, the cameras of the sensor network were calibrated using the camera calibra-

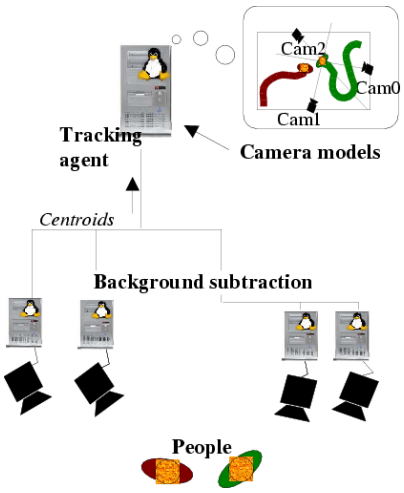

Figure 1. Components of the tracking system

tion toolbox by Bouguet [1]. In order to extract the intrinsic parameters and extrinsic parameters of each camera this method only needs several images of a checker board in various arbitrary positions. As the cameras are not moved, panned or tilted, the obtained camera models do not change during the operation of the tracking system.

3 Background subtraction

At each camera a background subtraction module extracts foreground regions from the live camera images. The centroids of foreground regions are later used by the tracking agent in connection with camera models to discover the 3-D position of objects of interest.

The key idea in the implemented segmentation algorithm is to subtract the still background from the current image yielding people or recently moved objects.

The critical part of this approach is to adapt the background estimation over time in such a way that gradual lighting changes or moved objects do not result in a seriously erroneous background estimate.

Most of the background subtraction algorithms estimate the background color of each pixel in the image continously. Our tracking system uses the background estimation process developed in [9]. The process uses a mixture of Gaussians to estimate the background color per pixel. This provides a more robust foreground region extraction compared to single Gaussian approaches.

Background models that estimate the background color per pixel tend to detect shadows as foreground (false positives), if the underlying color space has an intensity component as in Figure 2(b). To counter this problem the background color can be estimated only on a chromatic color space. But this does not always solve the problem, since a number of foreground objects might not be detected (false negatives) or an object dissolves into several regions as in Figure 2(c).

The implemented segmentation algorithm uses both chromatic and intensity information to ensure a low number of false positives and negatives. The process of segmentation is illustrated in Figure 2:

From the camera image (Figure 2(a)) the adaptively estimated background image on the intensity channel Y of the YUV color space is subtracted. This yields foreground regions as shown in Figure 2(b). In the same manner the background image estimated on the RG color space is subtracted from the current image (Figure 2(c)). Then each RG region is matched to a Y region, if it is inside this Y region. For each Y foreground region the bounding box of its matched or interior RG regions is computed (Figure 2(d)). The bounding boxes are filled and pixel-wise intersected with the Y foreground regions (Figure 2(e)).

This approach cuts off most of the shadows due to the use of chromatic information while exploiting intensity information to obtain smoother silhouettes.

From these foreground regions the RG color histogram, the bounding box, the centroid, and the size are computed and broadcasted appropriately packaged and time stamped.

4 Tracking

The tracking agent as the higher level component collects data streams from the image processing components (centroids of foreground regions). From this data the tracking agent computes 3-D locations and tracks of objects with the help of the previously obtained camera models.

Two tracking agents were implemented. One tracking agent is a re-implementation of the work in [7] which basically holds a Kalman filter per tracked object and can therefore be depicted as a best-hypothesis approach.

In order to have a better understanding which tracks correspond to real objects a probabilistic tracking algorithm was implemented. A probabilistic tracker naturally provides a confidence measure for each track by the corresponding a posteriori probability. As well our probabilistic tracker was designed to use multiple hypotheses per track with the intention to track objects more consistently. In the following we will refer to this way of tracking as the multi-hypothesis probabilistic approach or just the probabilistic approach.

In section 4.2 and 4.3 the two different tracking algorithms are described in more detail. Both agents use the same preprocessing of foreground and camera model data to generate hypothetical 3-D measurements of objects. Section 4.1 explains this localization process.

4.1 Creating 3-D measurements from low level data

To compute candidate 3-D locations of objects an agent guesses which foreground regions from different cameras belong to the same object. We depict such a grouping of foreground regions as a correspondence guess. In order to tell valid correspondence guesses from invalid guesses a measure is used that calculates how accurately an object can be localized with that guess using the camera models. The following paragraphs give a brief explanation how this measure is obtained:

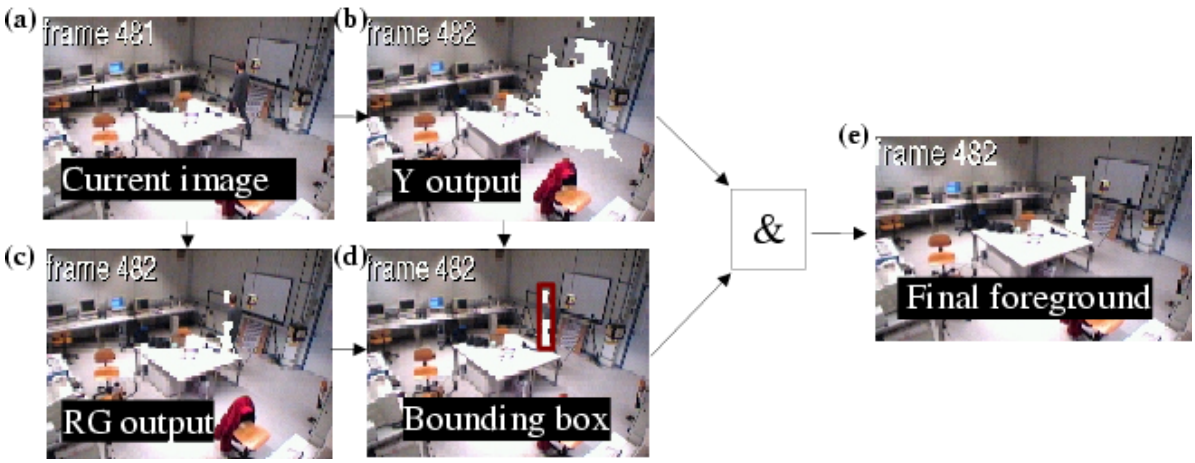

Figure 2. Extracting foreground (see text for details)

Each foreground region in a correspondence guess has a centroid. Assuming that the projected centroid of an object is close to the centroid of its foreground regions, a ray can be cast from the center of projection of the camera through the centroid of the foreground region towards the object's centroid in the scene (see Figure 3).

If the correspondence guess is correct as in Figure 3(b), the rays intersect nearly at one point, the location of the real object.

If the correspondence guess is not correct, the rays will not intersect at one point as in Figure 3(a).

Using the camera models the intersection of the mentioned rays can be expressed as an overdetermined linear equation. The residual r of the equation is the above mentioned measure for telling correct from incorrect correspondence guesses (see [7] for details).

The tracking agent computes for each frame all possible correspondence guesses and discards all guesses whose residual r is above a threshold (for instance 50mm per camera). The remaining correspondences are sorted by the number of foreground regions supporting it and secondarily by their residual r.

As each correspondence provides a believed object position, this produces an ordered list of hypothetical 3-D object positions or measurements ($\mathbf{z}_t = (x_t, y_t, z_t)$):

$$\mathbf{Z}_t = (\mathbf{z}_t^1, \mathbf{z}_t^2, \mathbf{z}_t^3, \ldots) \tag{1}$$

4.2 Best-hypothesis heuristic tracker

The best-hypothesis tracker is a re-implementation of the tracking approach as described in [7].

In this approach each object is tracked with a Kalman filter estimating velocity and position. Assuming that the tracking agent already has some valid tracks of the objects in the scene, the task is to match each measurement in $\mathbf{Z}_t$ to one of the already existing tracks.

To associate measurements with tracks several techniques can be used such as nearest fit or methods that minimize a penalty function defined for an assignment guess of the entire set of tracks to measurements as described in [7]. A threshold V is used to discard measurements that are too far away from a current object location for a given track.

For measurements that are not matched to tracks a new Kalman track is started. Tracks are only considered as valid tracks, if measurements were matched to them over some amount of time (for instance more than a half a second). If no measurements are matched to a valid track for a certain time (more than a second), the track is discarded.

4.3 Multi-hypothesis probabilistic tracker

The multi-hypothesis tracker uses a probabilistic approach to update and create tracks from the list of measurements $\mathbf{Z}_t$.

Assuming for the moment that the tracker has already produced some valid tracks, the task is to update the tracks given some new hypothetical 3-D locations of objects $\mathbf{Z}_t$. The tracker has to keep the most promising and to discard the most unlikely tracks.

As the tracker uses multiple hypothesis per track, it is important to understand that each track consists of several track paths.

Assigning probabilities to track paths

In order to tell good from bad track paths the a posteriori probability $P(\mathbf{X}^t|\mathbf{Z}^t)$ for each track path $\mathbf{X}^t$ is computed. To be more specific each track path is a time stamped sequence of 3-D locations (x_t, y_t, z_t):

$$\mathbf{X}^t = \{(\mathbf{x}_1), (\mathbf{x}_2), ..., (\mathbf{x}_t)\} \tag{2}$$

$\mathbf{Z}^t$ are all the measurements $\mathbf{Z}_t$ seen up to time t:

$$\mathbf{Z}^t = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_t\} \tag{3}$$

The a posteriori probability for a track path given a history of observations is formally:

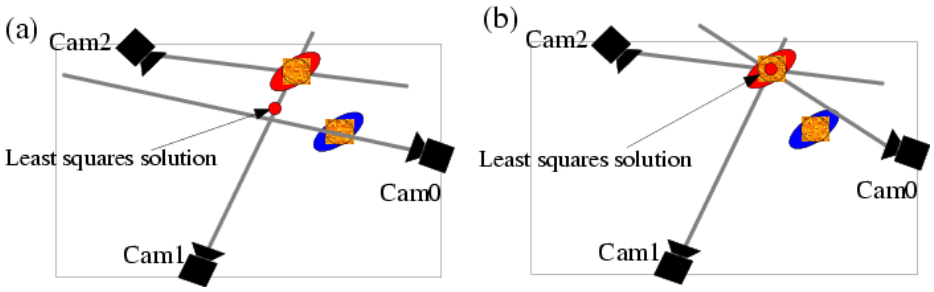

Figure 3. (a) A wrong correspondence (b) A correct correspondence

$$P(\mathbf{X}^t|\mathbf{Z}^t) = \frac{P(\mathbf{Z}^t|\mathbf{X}^t)P(\mathbf{X}^t)}{P(\mathbf{Z}^t)} \quad (4)$$

Assuming that $\mathbf{Z}_t$ only depends on the current track position hypothesis $\mathbf{x}_t$ and that prior measurements $\mathbf{Z}^{t-1}$ does not depend on $\mathbf{x}_t$, the above equation yields:

$$P(\mathbf{X}^t|\mathbf{Z}^t) = \frac{P(\mathbf{Z}^{t-1}|\mathbf{X}^{t-1})P(\mathbf{Z}_t|\mathbf{x}_t)P(\mathbf{x}_t, \mathbf{X}^{t-1})}{P(\mathbf{Z}^t)} \quad (5)$$

We compute the probability $P(\mathbf{Z}_t|\mathbf{x}_t)$ in the following way: The probability distribution that a measurement is seen at location $\mathbf{z}_t$ given the current position of the track $\mathbf{x}_t$ is modeled as a Gaussian distribution:

$$p(\mathbf{z}_t^i|\mathbf{x}_t) = \frac{1}{(2\pi\sigma^2)^{\frac{3}{2}}} \exp^{-\frac{1}{2\sigma^2}(\mathbf{x}_t - \mathbf{z}_t^i)^2} \quad (6)$$

,where σ is a value between 20 to 40cm.

As measurements are not equally likely, the overall probability $P(\mathbf{Z}_t|\mathbf{x}_t)$ can be seen as a weighted sum of $P(\mathbf{z}_t^i|\mathbf{x}_t)$ probabilities. The weights $P(\mathbf{z}_t^i)$ are modeled to be dependent on the triangulation error and the number of foreground regions supporting the corresponding measurement $\mathbf{z}_t^i$. This yields for $P(\mathbf{Z}_t|\mathbf{x}_t)$:

$$P(\mathbf{Z}_t|\mathbf{x}_t) = \sum_{i=1}^{n} P(\mathbf{z}_t^i) * P(\mathbf{z}_t^i|\mathbf{x}_t) \quad (7)$$

$P(\mathbf{x}_t, \mathbf{X}^{t-1})$ can as well be described with a Gaussian distribution:

$$p(\mathbf{x}_t, \mathbf{X}^{t-1}) = \frac{1}{(2\pi\sigma^2)^{\frac{3}{2}}} \exp^{-\frac{1}{2\sigma^2}(\mathbf{x}_t - \mathbf{x}_{t-1})^2} * p(\mathbf{X}^{t-1}) \quad (8)$$

Updating and creating tracks

For the updating process several possible actions how to continue a path have to be evaluated: A track path can be updated with a measurement using a Kalman filter. As well a path can be updated by reinitializing - jumping directly to the position of a measurement, and finally from a Kalman filter model a new position can be guessed without a measurement. For each action the overall $P(\mathbf{X}^{t+1}|\mathbf{Z}^{t+1})$ is computed.

As the tracker uses multiple hypotheses for a given track, the algorithm keeps the best n resulting track paths per track.

After the update process there might be measurements which were not used to extend any tracks. Such a measurement creates a new track whose initial position is the location of the unused measurement.

To ensure that track paths do not accumulate at the same 3-D position, paths with a smaller confidence measure are deleted if they come closer than the exclusion threshold E to a track path with a higher confidence measure. The value of E was varied in the experiments between 100mm and a 500mm.

Last but not least during the tracking process there will be tracks that were created earlier than other tracks. In order to make the $P(\mathbf{X}^t|\mathbf{Z}^t)$ values comparable among these, a penalty per missing frame is added to the confidence measure $P(\mathbf{X}^t|\mathbf{Z}^t)$. The value of the penalty corresponds to the probability that the younger track jumped by one meter and that the track was one meter away from the nearest measurement. This rather large penalty ensures that older tracks are kept unless there is no data supporting them.

5 Results

In operation our tracking system is able to track two people using three cameras at a frame rate of 3-7 frames per second on four Pentium II 400 MHz machines. The tracking agent has a smaller run time for a tracking step than the vision components. This is the reason why the frame rate mainly depends on the background subtraction modules that update their background models continously on a 160x120 image.

To evaluate the two tracking algorithms a fifty second long sequence of two people walking in a conference room was recorded at 5 frames per second with three cameras and segmented manually to produce ground truth data. In the sequence one of the two people is always in the field of view of all cameras, while the other person leaves the observation area for about 5 seconds.

The foreground region data from the background processing modules were logged in real time during the recording of the sequence (see Figure 4). From these logs the

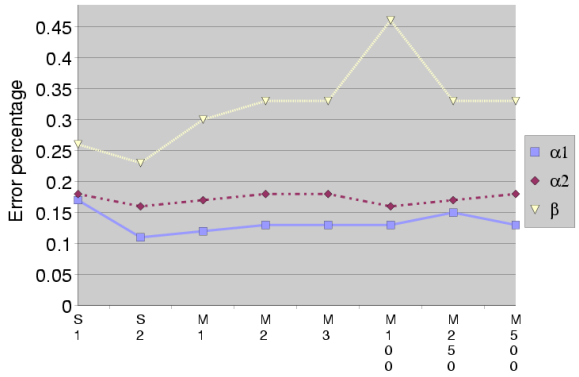

Figure 5. Percentage of tracking errors for different tracking variants

tracking algorithms produced their tracks which were compared to the ground truth data.

For each person the tracking error was computed in millimeters and a track of a person was considered to be lost, if the error was larger than 750mm. The mean μ and standard deviation σ of the 2-D position for each person was computed for frames that had a tracking error smaller than 750mm. As well we calculated for each person the percentage α how often its track was lost.

For the best-hypothesis tracker two different variants were used. The first one S_1 is a re-implementation of the approach described in [7]. Though this method worked well in general, it tended to lose track when a person turned around a corner; the Kalman filter continued on a straight line. To counter this problem the tracker variant S_2 adaptively increased the V threshold for those tracks. This enabled tracks to search for valid measurements in a larger area and solved the above described problem in most cases.

The probabilistic tracker was evaluated varying the number of allowed hypotheses per track between one and three (tracker variants M_1, M_2, M_3). As well variants of the tracker were evaluated with differing exclusion thresholds E ranging from 100mm to 500mm (variants $M_{100}, M_{250}, M_{500}$ where the number of hypotheses per path was three).

For each tracker those tracks were considered valid which had been updated in the last second and that were created at least half a second ago.

If less than two tracks were valid, the tracker only reported its single valid track or no track at all. If two or more tracks were valid, the best-hypothesis tracker reported the two longest valid tracks and the probabilistic tracker reported the two valid tracks with the highest probability.

In order to reflect the amount of false alarms produced by the trackers, we compute an additional evaluation measure β. It is the percentage how many times a tracker failed to judge the exact number of valid tracks in the room.

Figure 5 and Table 1 summarize the result for the various tracker variants.

Table 1. Tracking accuracy of different algorithms (Values in mm)

	μ_1	μ_2	σ_1	σ_2
S_1	193	210	127	138
S_2	194	216	120	145
M_1	196	215	120	139
M_2	197	207	123	140
M_3	197	210	122	142
M_{100}	192	207	127	145
M_{250}	188	203	120	142
M_{500}	197	210	122	142

5.1 Discussion

By inspecting both tracker outputs frame by frame, we found that they consistently kept track of the two subjects as long as the subjects were more than a 1 meter apart in the scene. The trackers failed, when the subjects came to close to each other. In this case the background subtraction modules merged the subjects into one foreground region and thus one track out of two was lost. Although in one instance of a closer encounter the track was only lost for a second: The Kalman filter was able to dead reckon the person's location since the subject did not change its direction or velocity. But in most cases the subjects changed either their direction or their velocity (turning around the Table or stopping temporarily) which caused the loss of their track during the encounter.

It is interesting to mention that both approaches managed to detect that one subject left the observation area for five seconds.

The similar performance of the two tracking approaches are also reflected in table 1. The mean and standard deviation of the tracking accuracy does not differ a lot among the algorithms. This is probably due to the way μ and σ were computed and to the fact that the triangulation process itself is identical for the trackers.

Additionally, the percentages of lost tracks (measures α_1 and α_2) are of the same order for both tracking approaches. For the best hypothesis tracker S_2 (S_1) $\alpha_1 = 0.11(0.17)$ and $\alpha_2 = 0.16(0.18)$, while for the probabilistic tracker variants $M_1, M_2, .., M_{500}$ α_1 ranges from 0.12 to 0.15 and α_2 ranges from 0.16 to 0.18.

The main difference between the two approaches is the amount of false alarms. The measure β shows that the S_1 and S_2 estimate the number of tracks more accurately than the probabilistic tracker (0.23,0.26). The probabilistic tracker produces 10 percent more false alarms: For instance M_3 has a $\beta = 0.33$. This is why the best heuristic approach might be preferable at the moment, but we think that refining the simplistic probabilistic model might improve the false alarm rate. On the long run a more sophisticated probabilistic approach should be preferable, because its posterior probabilities should provide a more precise confidence measure of tracks. The comparable tracking performance results for the so far simplistic probabilistic model and the best hypothesis approach suggest that the subject of prob-

Figure 4. Snap shot of the evaluation sequence

abilistic tracking in this domain of application is a worthwhile for future research.

Finally, we found that multiple hypotheses did not provide an improvement on this test sequence. Multiple hypotheses on this sequence even produced a slightly higher false alarm rate. Inspecting the trackers output frame by frame showed that the extracted foreground region provided a rather clear foreground region signal that did not give much opportunity for alternate paths. With such a clear sequence a multiple hypotheses tracking approach does not seem to be worthwhile. A single hypothesis approach either based on heuristics or a probabilistic model seems to suffice.

6 Conclusion

We have build a real time 3-D tracking system using multiple calibrated cameras to locate and track objects and people in a conference room. The system is designed as a distributed sensor network and relies on a quite sophisticated adaptive background subtraction algorithm whose key feature is the fusion of chromatic and intensity information to suppress shadows as false positives.

To have a better understanding which tracks are likely to correspond to real objects in our system, a probabilistic tracker was implemented. The implemented probabilistic tracker leads to tracking results comparable to the results achieved with a best-hypothesis tracker while providing additional confidence measures for each track.

Improvements of the probabilistic tracker can certainly be made by refining the rather simplistic probability model, i.e. to incorporate the velocity from the Kalman filters, as well as color and size information of the matched foreground regions.

Currently, we are evaluating our tracker on several longer test sequences and we are trying to improve the tracker using a more sophisticated probabilistic tracking model to provide better confidence measures for tracks.

In the future we plan to also use stereo cameras in the sensor network to be able to fuse geometric, color and depth information to increase the robustness of our system.

7 Acknowledgments

The work on the 3-D tracking system was conducted at the University of Massachusetts at Amherst and at Universität Karlsruhe. We want to thank Roderic Grupen and Allen R. Hanson for their support at University of Massachusetts and Alex Waibel at Universität Karlsruhe.

Part of this work was carried out within the FAME project and has been funded by the European Union as IST project No. IST-2000-28323. This research is also partially supported by the German Research Foundation (DFG) as part of the Sonderforschungsbereich 588.

References

[1] J.-Y. Bouguet. Camera calibration toolbox for matlab, 2001.

[2] Q. Cai and J. Aggarwal. Tracking human motion using multiple cameras, 1996.

[3] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175–185, 2000.

[4] I. Haritaoglu, D. Harwood, and L. Davis. W4: Who? when? where? what? a real time system for detecting and tracking people. *Face and Gesture Recognition*, pages 222–227, 1998.

[5] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. In *EVVC (1)*, pages 893–908, 1998.

[6] J. Krumm, S. Harris, B. Meyers, B. Brumitt, and M. H. amd Steve Shafer. Multi-camera multi-person tracking for easy living. In *Third IEEE International Workshop on Visual Surveillance, July 1*, 1997.

[7] Mikic, Santini, and Jain. Tracking objects in 3d using multiple camera views. Technical report, University of California at San Diego, USA, 2000.

[8] D. Mills. David mills. network time synchronization project. web site http://www.eecis.udel.edu/ mills/ntp.htm, 2001., 2001.

[9] C. Stauffer and W. Grimson. Adaptive background mixture models for realtime tracking. In *Proc. of CVPR, 1998, 333–339*, 1998.

[10] A. A. Ward and A. Hopper. A new location technique for the active office. In *IEEE Personal Communications, vol.4, pp. 42-47*, 1997.

[11] 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.

[12] T. Yamada, S. Nakamura, and K. Shikano. Robust speech recognition with speaker localization by a microphone array. In *Proc. ICSLP '96*, volume 3, pages 1317–1320, Philadelphia, PA, 1996.