MULTI-PERSON TRACKING FROM SPARSE 3D TRAJECTORIES IN A CAMERA SENSOR NETWORK

Kyle Heath

Stanford University
Department of Electrical Engineering
Stanford, CA 94305

Leonidas Guibas

Stanford University
Department of Computer Science
Stanford, CA 94305

ABSTRACT

We describe and evaluate a vision-based technique for tracking many people with a network of stereo camera sensors. The technique requires lightweight local communication and is suitable for crowded scenes where targets are frequently occluded and where appearance based modeling techniques fail. In our approach, multiple stereo sensors individually estimate the 3D trajectories of salient feature points on moving objects. Sensors communicate a subset of their sparse 3D measurements to other sensors with overlapping views. Each sensor fuses 3D measurements from nearby sensors using a particle filter to robustly track nearby objects. We evaluate the technique using the MOTA-MOTP multi-target tracking performance metrics on real data sets with up to 6 people and on challenging simulations of crowds of up to 25 people with uniform appearance. Our method achieves a tracking precision of 10-30 cm in all cases and good tracking accuracy even in crowded scenes of 15 people.

Index Terms— Distributed tracking, camera sensor network, particle filtering

1. INTRODUCTION

Multi-person tracking has applications in many fields including security, human computer interaction, entertainment, and even health care. Robust tracking is a basic building block for a wide range of higher-level applications. Existing techniques have not found widespread use because they rely on assumptions that are too strong for many real-world applications. For this reason, the problem of tracking many people in dynamic, cluttered, and uncontrolled environments is an active area of research. We address some of the difficulties posed by unconstrained environments with the goal of bringing robust people tracking out of the lab and into the real world.

Our technique is designed for a network of stereo camera sensors with limited communication. During operation, each sensor independently estimates 3D trajectories of a small set of different surface features on moving objects as illustrated in Figure 1. Sensors do not compute dense depth images, but only estimate depth for a sparse set of the most reliable image features. Each sensor communicates the positions of features it observes to other sensors with overlapping sensing volumes. At each sensor, a data association module assigns measurements to independent people trackers. Each tracker uses a particle filter to estimate a person's 2D position and velocity in the world coordinate frame. We propose a formulation of a particle filter that takes into account the shape of a human, the visibility constraints of the camera, and the velocity of the observed features.

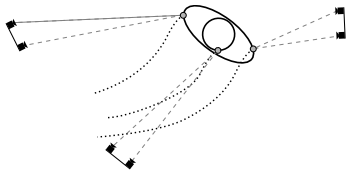


Fig. 1. Multiple stereo-pair sensors independently estimate 3D trajectories of different sets of features on moving objects. Sensors communicate trajectories of features they observe to neighbor sensors with overlapping sensing volumes. Each sensor fuses 3D trajectories using particle filter based person trackers.

Many existing techniques make assumptions which greatly restrict the generality of the approach in real-world settings. For example, common background modeling techniques assume that the appearance of non-targets can be predicted well from past observations. Furthermore, they assume that changes in the scene are caused primarily by the presence of targets. Yet in many interesting real-world settings, these assumptions do not hold. Scenes often include many other dynamic objects, fast changes in lighting, and complex object interactions like shadows and reflections that greatly influence the image.

In single-view tracking techniques, it is common to assume that distinct targets have distinct appearances with respect to color, texture, size, or contour features. This assumption does not hold in many settings, for example when people may be wearing uniforms. Single-view approaches also face a fundamental limitation caused by occlusions. Such techniques assume that static and dynamic occlusions are rare

978-1-4244-2665-2/08/$25.00

and can be handled as short-term special cases. However in many real-world settings, it is not possible to place a camera in the ideal location that minimizes occlusions (like a very high overhead view) and a robust technique must handle frequent and prolonged occlusions.

Multi-view techniques that perform correspondence between views assume that the appearance of a feature in one view will be similar to its appearance in another view. This assumption fails for widely separated views where the scene geometry and lighting can result in a lack of commonly observed features and very different appearances of the same feature. Furthermore, communicating a description of a sufficient number of potential features to establish reliable correspondences may require a significant amount of communication. Thus the wide-baseline correspondence problem is difficult when using many cameras in a bandwidth-limited distributed setting.

Contribution

Our contribution is a distributed vision technique for tracking moving objects in real-time given a sparse set of 3D feature trajectories as observations. To the best of our knowledge, we are the first to propose this formulation of a particle filter on sparse 3D trajectories.

Our technique does not require the assumptions discussed above which cause problems in unconstrained environments. It uses stereo depth measurements which are generated without modeling the appearance of targets or non-target objects. Furthermore, it handles frequent and prolonged occlusions by fusing observations from multiple viewpoints. By performing short-baseline stereo correspondence within the sensor, the technique does not need to communicate to establish correspondences between wide-baseline views. This makes the approach scalable in a distributed setting because it requires relatively little communication to send a small set of 3D trajectories compared to image data.

The performance of this technique was evaluated using the standard Multi-Object Tracking Precision (MOTP) and Multi-Object Tracking Accuracy (MOTA) performance metrics described in [1]. The particle filters we propose perform well using fewer than 100 particles and can track multiple targets in real time on commodity PC hardware.

The paper is structured as follows. Section 2 describe how our approach compares to other distributed visual tracking techniques. In Section 3, we give an overview of the system by describing the input and output of the system components and then describe each component in detail: sparse stereo feature tracking in Section 4, data association and track initialization in Section 5, and particle filter person tracking in Section 6. In Section 7, we present an evaluation of the system on real and synthetic data sets.

2. RELATED WORK

Vision based multi-target tracking has been studied extensively in the literature. Previous work is vast and is summarized well in [2]. The following is a discussion of only the most closely related work for tracking multiple targets in real-world settings.

Several techniques use multiple short-baseline stereo cameras. A real-time people tracking system for an interactive environment is presented by Krumm *et al.*[3] which uses depth-based background subtraction. The offline technique of Darrell *et al.*[4] also uses depth based background subtraction but proposes delaying segmentation until after sensor fusion which is done by rendering 3D foregrounds from multiple sensors in a common "plan view" image. The techniques proposed by Harville [5] and Zhao *et al.*[6] also use the idea of "plan views" of dense stereo data for tracking but not multi-sensor fusion. These techniques are all different from ours because they require that the objects in the scene have enough texture for dense stereo reconstruction and build background models that assume a static environment.

Like our technique, the M2Tracker system described by Mittal and Davis [7] is designed to work in crowded settings where targets are partially occluded in every view. Their region-based stereo technique avoids many problems with wide-baseline correspondence by matching regions instead of points. Yet their approach is very different from ours because it requires background modeling and assumes that everyone in the scene is wearing uniquely colored clothing to perform the region based correspondence.

Osawa *et al.*[8] track people with a particle filter by generating a 3D model of the environment and use a 3D ellipsoid human model. The likelihood of a hypothesized state is evaluated by rendering images of the ellipsoid shape model in the environment model to capture the effect of known occlusions. They demonstrate tracking in a cluttered office environment with two people but do not discuss the cost of rendering an image from a 3D model per particle per time step. Unlike our method, this technique requires a static scene and uses background modeling to segment the target.

Khan and Shah [9] and Lopez *et al.*[10] propose tracking methods based on the visual hull. In [9] a 2D slice of the visual hull around people's feet is calculated and tracking is done offline. In [10] particle filters track people using a voxel representation of the 3D visual hull. Visual hull techniques are sensitive to errors in foreground segmentation and are not suited for environments with many occlusions because the visual hull becomes loose and can not resolve individuals.

The method of Tsutsui *et al.*[11] was used to track a single person through occlusions using optical flow. Though they estimate temporal correspondence and use multiple cameras, they do not estimate 3D trajectories of surface features as in our method. Our sparse stereo tracking technique amortizes the expense of an initial correspondence search over many

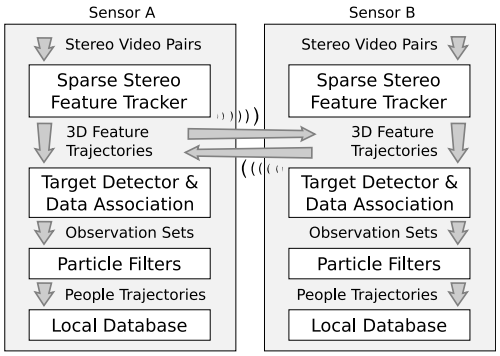


Fig. 2. Each sensor captures and processes a stereo image pair to produce 3D trajectories of a sparse set of features. Multiple sensors collaborate by sharing 3D feature trajectories. The feature trajectories are assigned to a particle filter tracker and new trackers are generated. The people tracks generated by the particle filter trackers are stored in a local database for use by higher-level applications.

cheaper correspondence checks and could be considered a simplification of the technique proposed by Tang *et al*.[12]. Our technique differs because it uses only images as input and does not rely on the correct tracking of previous 3D features.

3. OVERVIEW

In the following, we give an overview of the technique by describing the flow of information through the system components shown in Figure 2. A sensing cycle begins when a sensor captures a stereo image pair. The *Sparse Stereo Feature Tracker* module takes the image pair as input and outputs the current 3D position of a small set of previously tracked surface features. Each sensor then updates neighbor sensors with the current position of features that fall inside their sensing volume. The set of all 3D trajectories generated locally or received from neighbors is then fed as input to the *Target Detector and Data Association* module. This module detects new targets and generates as output an assignment of the 3D trajectory measurements to each tracker called an *Observation Set*. Next each *Particle Filter* takes the 3D trajectories in its *Observation Set* as input and generates an estimate of the target's 2D position and velocity in the world coordinate frame. Finally, the tracking results are stored locally in a database. Higher-level applications can then query the local databases to perform their tasks.

We assume that the sensors' intrinsic and extrinsic parameters have been estimated using existing techniques [13, 14]. Also, we assume the sensors have detected their 1-hop communication neighbors. Using this information, sensors can discover other sensors that may have overlapping fields of view. Because of the limited resolution of the stereo sensors, we assume the effective sensing volume is bounded. Thus

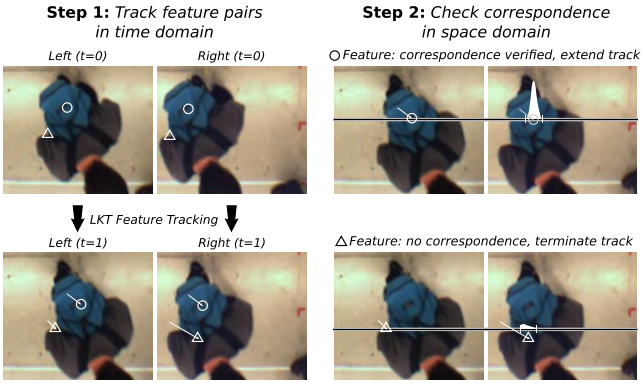


Fig. 3. A feature track update has two steps: In *Step 1*, the positions of image features in the left and right images are updated independently using LKT feature tracking in the time domain. The feature in the circle is correctly tracked while the feature in the triangle is not. In *Step 2*, a very limited correspondence search is performed in the spatial domain (the scores indicated as a white graph). In *Case 1*, the good correspondence is verified. In *Case 2*, no good correspondence is found so the track is terminated.

sensors that observe the same region of space are likely to be 1-hop communication neighbors. This suggests locality in the communication pattern between sensors.

4. TRACKING SPARSE 3D FEATURES

In this section, we describe a method for estimating the 3D trajectories of a sparse set of surface features. The cost of establishing image correspondences in the time domain is offset by a more efficient stereo correspondence search in the space domain. We model the stereo depth estimation error as Gaussian and track the features using a Kalman filter [15].

The feature tracking component starts by identifying good features to track using the Harris corner detector [16]. To get uniform coverage of features on moving objects, it restricts the search to regions that are changing and are not close to other tracked features. In each frame, it chooses the best Harris corner in the left image and searches for a corresponding point in the right image. To make this efficient, it rectifies the images so that the epipolar lines correspond to the same rows in the images. The correspondence search uses a forward-backward constraint (mutual best match) and the sum of absolute difference (SAD) error metric with a window size of 9×9 pixels on the RGB images. If it finds a match, it creates a new feature-pair.

Next the feature tracking component extends the tracks for existing feature-pairs by performing a time domain and a space domain correspondence as shown in Figure 3. First it tracks the left and right image features independently in the time domain using Lucas-Kanade-Tomasi (LKT) feature

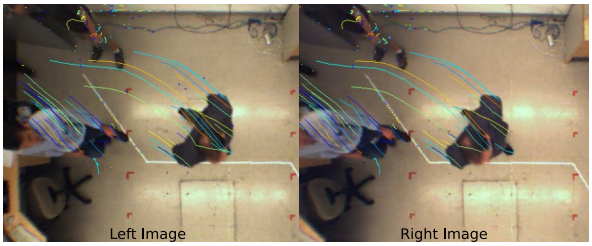


Fig. 4. Example feature-pair tracking results where the trajectory of a feature is indicated by the same color path in the left and right images. Cross-eyed stereoscopic viewing is possible.

tracking [17]. Because of accumulated errors, the features tracked in the left and right images may drift over time. To verify the same feature is being tracked, the component does a limited correspondence search on a small window ± 6 pixels at the image location predicted by the LKT feature tracking. The feature track is extended if the check succeeds and is terminated otherwise. The method is efficient because the cost for finding correspondences in the temporal dimension is offset by the reduced cost of a stereo correspondence verification instead of a full correspondence search. Figure 4 shows stereo feature tracks generated by this technique.

Because of error in estimating the corresponding image points, there will be error in the triangulated 3D positions. If we model the error in the image points as Gaussian, the error for the triangulated point is not Gaussian because of the non-linearity of the triangulation function. However a normal distribution is a convenient approximation and can be calculated as described in [18]. Using the calculated distribution for each triangulated measurement, the feature tracking module performs Kalman filtering to estimate the 3D position and velocity of the feature. The posterior distribution produced by the Kalman filter will be used later in a particle filter to effectively fuse measurements from different sensors with very different noise characteristics.

5. DATA ASSOCIATION AND TARGET DETECTION

Our technique uses a separate particle filter to track each person and thus requires a data association step to assign 3D feature measurements to individual trackers. In the next section, we describe how to assign measurements to trackers and how new trackers are instantiated.

5.1. Data Association

To track a target, our technique first computes a mapping of measurements to targets. To generate this mapping, the *Data Association Module* calculates the probabilities $p(m|T_i)$ that a measurement m was generated by each existing track T_i. It assigns a measurement to the most likely track, as long as the probability of this assignment is above a threshold. If none of the existing tracks explain the observation well, it marks the observation as unassigned and considers it later during the new track detection step.

Since we assume that a feature tracked over time was generated by the same moving object, information about past assignment likelihoods should be incorporated when predicting the current assignment likelihood. For example if two tracks A and B become very close, a feature that has always been assigned to track A should not be assigned to track B even if track B appears a more likely candidate at that instant. Thus to compute the probability $p_t(m|T_i)$ for time t, the data association module forms a weighted average of the probability at the previous time $p_{t-1}(m|T_i)$ and the probability $p_t(m|s_{T_i})$ of the measurement given a person in the current state of tracker i as

$$p_t(m|T_i) = \alpha p_{t-1}(m|T_i) + (1-\alpha)p_t(m|s_{T_i}). \quad (1)$$

Here α controls how fast the previous assignment probabilities are forgotten. The probability $p_t(m|s_{T_i})$ is calculated using Equation 4 described in the next section.

5.2. New Track Detection

The *New Track Detection Module* clusters measurements that were marked as "unassigned" during the data association step to find sets with similar position and velocity. These clusters may represent a new person entering the sensing volume. It searches for clusters by randomly choosing a 3D feature and finding the set of neighbors with a similar velocity within a vertical cylinder of radius r. We use $r = 0.5$ m to approximate a human shoulder span. The module instantiates a new track for the largest cluster with at least 10 measurements.

6. PARTICLE FILTER PERSON TRACKING

Particle filtering is a sequential Bayesian inference technique which represents non-parametric posterior distributions by a set of random samples from the true distribution. See [19] for an introduction to particle filtering. Isard and Blake demonstrated the utility of this technique for 2D visual tracking in the well known Condensation algorithm [20]. The robustness of this technique in the face of ambiguity results from its ability to implicitly maintain multiple hypotheses through a non parametric representation of the posterior distribution.

Let s_t represent the state of the target at time t, which we wish to estimate. Here $s_t = \{x_t, y_t, \dot{x}_t, \dot{y}_t\}$ is the target's 2D position and velocity in world coordinates. Let z_t represent the observation at time t. Then $z_t = \{m_t^1, m_t^2, ..., m_t^{n_t}\}$ consists of a set of feature measurements m_t^i. Each measurement m_t^i is a multivariate normal distribution with mean $\mu_t^i = \{x, y, z, \dot{x}, \dot{y}, \dot{z}\}$ and covariance Σ_t^i. The measurements are obtained from the sparse stereo feature tracking algorithm described in Section 4.

The particle filter tracker operates as follows. Each tracker has N particles which represent the potential states of the target at time t (i.e. a possible 2D position and velocity). The tracker simulates the evolution of each particle to time $t+1$ according to a stochastic motion model described in Section 6.1. This is the prediction step. Next the tracker considers the 3D measurements and assigns an importance weight to each particle where higher weights are given to particles that are more likely given the measurements. The importance weights are calculated from the observation model described in Section 6.2. Finally the tracker resamples the particles by randomly drawing (with replacement) N particles where each particle is chosen with a probability proportional to its importance weight. This is the correction step. Next we define the motion model and observation models used in the Bayesian prediction and correction steps respectively.

6.1. Motion Model

We define a motion model $p(s_t|s_{t-1})$ to describe how the target's state evolves from one time step to the next. We use a simple dynamical model with Gaussian noise

$$p(s_t|s_{t-1}) \sim \mathcal{N}(\mu_m, \Sigma_m), \tag{2}$$

$$\mu_m = (x_t + \dot{x}_t \Delta t,\ y_t + \dot{y}_t \Delta t,\ \dot{x}_t,\ \dot{y}_t)^T,$$

where the covariance of the noise Σ_m is selected to reflect how fast the targets maneuver.

6.2. Observation Model

The observation model $p(z_t|s_t)$ gives the probability of a particular observation z_t given the object's state is s_t. We define it as the product of the probabilities of the individual 3D feature measurements in the observation each taken to the power α:

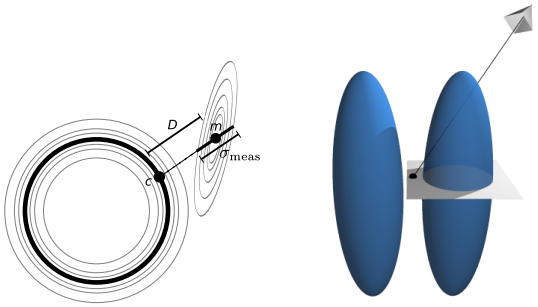
$$p(z_t|s_t) = \Pi_{i=1}^{n_t} p(m_t^i|s_t)^\alpha. \tag{3}$$

This formulation is useful for combining measurements that are not truly independent in the computationally convenient form of a product as discussed in [21]. Using $\alpha < 1$ accounts for redundant information by discounting the contribution of each measurement. This makes the model more accurate and improves performance in practice. We chose $\alpha = 0.07$ empirically.

Next we define the probability of an individual measurement given the state of a person:

$$p(m|s_t) = p_{\text{shape}}(m|s_t)\ p_{\text{vis}}(m|s_t)\ p_{\text{vel}}(m|s_t). \tag{4}$$

This probability is the product of three component probabilities: shape, visibility, and velocity.



(a) Shape component (b) Visibility component

Fig. 5. *(a) shows the iso-contours of a horizontal slice through the shape model distribution (circles) and measurement distributions (ellipses). The variance σ_{meas} in Equation 5 is the variance of m in the direction to c. (b) The visibility component incorporates knowledge of the camera position to assign the ellipsoid on the right a low probability since the black measurement point would not be visible to the camera because of a self occlusion.*

6.2.1. Shape

The shape component of the observation model encodes the constraint that measured features should be near the surface of the tracked object. Any geometric shape model could be used, but models that allow efficient distance queries are desirable. In this work, we use a simple axis-aligned ellipsoid. A parametrized shape model could be used to fit variations in the target class (e.g. height, width) by adding variables to the estimated state. However in this work, we used a fixed size model based on average adult proportions.

Given a shape model, we define a distance function $d_{\text{meas}}(m, s_t)$ from the 3D measurement m to the closest point c on the shape model positioned at state s_t. The random variable D is the distance from the measurement m to the point c as shown in Figure 5(a). We model the true distance to the surface D as the calculated distance plus normally distributed noise from error in the shape model and error in the measurement as

$$D = d_{\text{meas}}(m, s_t) + \mathcal{N}(0, \sigma_{\text{shape}}) + \mathcal{N}(0, \sigma_{\text{meas}}). \tag{5}$$

Note that the variance σ_{meas} is the variance of the measurement in the direction to the closest point on the shape model as shown in Figure 5(a). Finally the shape component of the observation model is

$$p_{\text{shape}}(m|s_t) = p(D = d_{\text{meas}}) \sim \mathcal{N}(0, \sigma_{\text{shape}} + \sigma_{\text{meas}}). \tag{6}$$

6.2.2. Visibility

The visibility component incorporates knowledge that the camera can not observe a surface if the line of sight to the surface is broken. For example in Figure 5(b), a measurement located

between the two hypothesized states is equally likely to have come from either state according to the shape component of the observation model. However, the visibility component assigns the state on the right a low probability since that point would not be visible to the camera because of a self occlusion.

First we define a visibility indicator function $v(p, s_t)$ which is 1 when the shape model positioned at s_t does not obstruct the line of sight from the camera to the point p and zero otherwise. Figure 5(b) shows a horizontal slice of the visibility indicator function which corresponds to the shadow that would be cast by the shape model if there was a light source at the camera. Our technique approximates the visibility probability by generating k samples p_i from the measurement distribution m and calculates the average number of visible sample points

$$p_{\text{vis}}(m|s_t) = \frac{1}{k}\sum_{i=1}^{k} v(p_i, s_t). \tag{7}$$

6.2.3. Velocity

The velocity component of the observation model states that velocity of the 3D measurement should be the velocity of the tracked object's centroid plus Gaussian noise. The 3D measurement velocity is projected to 2D for comparison to the tracked object's velocity and its probability is evaluated with respect to the 2D Gaussian distribution

$$p_{\text{vel}}(m|s_t) \sim \mathcal{N}\left(\left(s_{\dot{x}_t}, s_{\dot{y}_t}\right)^T, \Sigma_{\text{vel}}\right). \tag{8}$$

This very simple model assumes a rigid body undergoing pure translation, but is sufficient to disambiguate feature trajectories that are close but have different velocities.

7. EVALUATION

To evaluate the proposed method, we generated a series of real and synthetic data sets to simulate various configurations of a stereo camera sensor network. Videos of the data sets and tracking results are available at: `http://www.stanford.edu/~heathkh/smpt`.

We used the *Multiple Object Tracking Precision* (MOTP) and *Multiple Object Tracking Accuracy* (MOTA) metrics proposed in [1] to evaluate the performance of our tracker. The metrics were designed to give a compact and intuitive sense of a tracker's performance and are used in large tracking research initiatives including the CHIL [22] and Vace [23] projects. The MOTP metric measures the trackers ability to estimate precise positions. To calculate MOTP, a scoring program establishes a mapping between measured and ground truth tracks and calculates total position error for matched object-hypothesis pairs over all frames, divided by the total number of matches made across all frames:

$$\text{MOTP} = \frac{\Sigma_{i,t} d_{i,t}}{\Sigma_t c_t}. \tag{9}$$

The MOTA metric expresses the ability to correctly estimate the number of objects and give trajectories consistent labels. The scoring program calculates MOTA by calculating the average number of targets missed $\overline{e}_{\text{fn}}$, average number of targets falsely detected $\overline{e}_{\text{fp}}$, and average number of times targets have their labels mismatched $\overline{e}_{\text{mm}}$. The sum $\Sigma_t g(t)$ in the denominators is the sum over all frames of the number of people present in each frame. In all evaluations, we use a radius parameter of 0.25 m in the computation of the MOTA/MOTP metrics:

$$\text{MOTA} = 1 - (\overline{e}_{\text{fp}} + \overline{e}_{\text{fn}} + \overline{e}_{\text{mm}}) \tag{10}$$

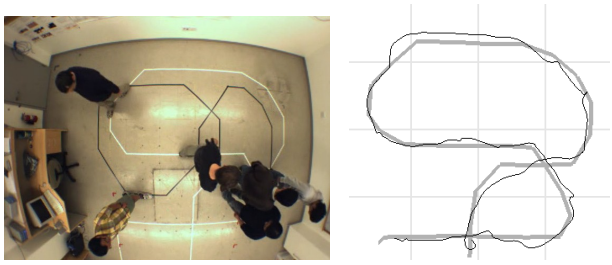
$$\overline{e}_{\text{fn}} = \frac{\Sigma_t e_{\text{fn}}(t)}{\Sigma_t g(t)}, \quad \overline{e}_{\text{fp}} = \frac{\Sigma_t e_{\text{fp}}(t)}{\Sigma_t g(t)}, \quad \overline{e}_{\text{mm}} = \frac{\Sigma_t e_{\text{mm}}(t)}{\Sigma_t g(t)}.$$

Experiment A - Single sensor in real scene

We collected real data sets by building stereo heads from unsynchronized VGA firewire cameras with wide-angle lenses (135° HFOV) arranged as stereo pairs with a 20 cm baseline. Note that using a wide-angle lens provides a large sensing volume but low depth resolution with a VGA image. We obtained ground truth for the real data sets by marking the position of people's feet on the ground plane from a calibrated overhead view of the observation region. To achieve sufficiently accurate ground truth, we asked the participants to follow paths marked on the floor with colored tape. The annotation tool overlays the video with these ground plane paths which allowed more accurate annotation and guided annotation when the feet were occluded from view (which occurred frequently).

In Experiment A, we mounted a single stereo camera sensor facing down 3 m above the floor in a 6 m×7 m room. We performed 6 four minute long trials with increasing numbers of people walking at various speeds and continually entering and leaving the sensing volume. Note that many of the simplifying assumptions discussed in the introduction do not apply for this data set. In particular, four of the six people are wearing nearly identical dark colored clothes, significant occlusions occur frequently between multiple people for several seconds, and the specularity of the floor causes the background appearance to change significantly as people move making background modeling difficult. An image from the last trial with six people is shown in Figure 6(a) and a measured track is compared with ground truth in Figure 6(b).

Table 1 shows the MOTA and MOTP metrics for these trials. The tracking precision (MOTP) is around 20 cm for all trials. The tracking accuracy (MOTA) degrades in trials with more people. We found that much of the false positive and false negative errors that reduce the MOTA score result from the delay in instantiating and terminating tracks when people enter or leave the scene. These tracking results compare favorably to those reported in [10] (MOTP=18.5 cm, MOTA=81.2%) for a similar scenario with up to 6 people which used 5 higher-resolution cameras.



(a) Camera view (b) Example track

Fig. 6. *(a) In Experiment A, a single stereo sensor tracks up to 6 people. This frame shows an occlusion between four people. (b) A comparison of a track from Experiment A (thin black line) with ground truth (thick gray line). The grid indicates 1 m squares.*

Table 1. Tracking performance in Experiment A

People	MOTP cm	$\bar{e}_{\text{fn}}\%$	$\bar{e}_{\text{fp}}\%$	$\bar{e}_{\text{mm}}\%$	MOTA %
1	19	7.51	6.81	0	85.7
2	18	6.37	4.81	0.39	88.4
3	16	9.45	5.91	0.23	84.4
4	19	8.61	6.53	1.21	83.7
5	17	6.96	6.18	0.97	85.9
6	17	9.80	9.19	2.50	78.5

Experiment B - Four sensors in real scene

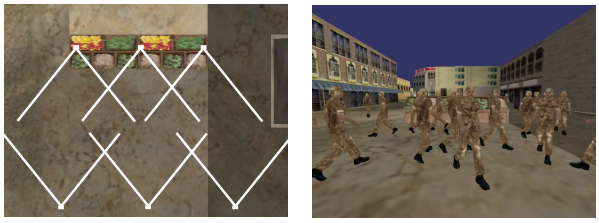
In Experiment B, we mounted four stereo camera sensors in the corners of a 6 m×8 m room at an angle approximately 45° from horizontal. We chose a cluttered office environment with occlusions caused by desks and bookshelves. Figure 7 shows images from the four sensors. We performed a one minute long trial where four participants brush by each other while walking a narrow circuit marked on the floor. The MOTP and MOTA metrics are shown in Table 2.

The tracking precision (MOTP) is about 10 cm worse than in Experiment A. This is expected because the targets were farther from the stereo sensors and noise in the depth estimation has a greater effect on the 2D position estimate when the sensors have a more horizontal view compared to the downward view of the scene in Experiment A. The tracking accuracy also decreased because of a pair of false positive and false negative tracks.

Table 2. Tracking performance in Experiment B

People	MOTP cm	$\bar{e}_{\text{fn}}\%$	$\bar{e}_{\text{fp}}\%$	$\bar{e}_{\text{mm}}\%$	MOTA %
4	28	14.8	12.48	1.76	70.1

Fig. 7. In Experiment B, four stereo sensors observed people walking in a cluttered office environment.



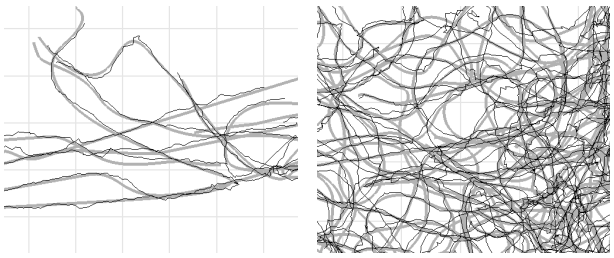
(a) Sensor placement in Exp. C (b) Sensor view

Fig. 8. *(a) In Experiment C, a game engine is used to simulate video from six sensors arranged in a corridor in an outdoor market setting. (b) A view from a sensor in Experiment C.*

Experiment C - Tracking simulated crowds with limited communication

In Experiment C, we used the Delta3D game engine to generate simulated video of crowds of 5, 15 and 25 people with uniform appearance walking in an outdoor market. Using a simulation approach allowed us to obtain accurate ground truth for larger crowds than was feasible in our physical testbed. Figure 8(a) shows the configuration of the 6 sensors used in this experiment and Figure 8(b) shows a view from one of the sensors. The sensors were positioned horizontally at eye level and configured with a 15 cm baseline, an 80 degree field of view, and a resolution of 640 × 480.

In this experiment we evaluated the effect of a limited communication budget on tracking performance for different target densities. We use the following simple communication protocol. At each time step, a sensor is allowed a fixed number of bits to transmit. For example a sensor allocated 50 kbps transmission rate could send roughly 100 feature position updates at 15 hz (where we assume features are encoded as pixel coordinates + disparity in 33 bits). A sensor selects which subset of observed features to share by drawing randomly from the set of features that fall inside the sensing



(a) 5 people (b) 25 people

Fig. 9. *Comparison of generated tracks (thin black lines) with ground truth tracks (thick gray lines) from Experiment C. The grid indicates 1 m squares.*

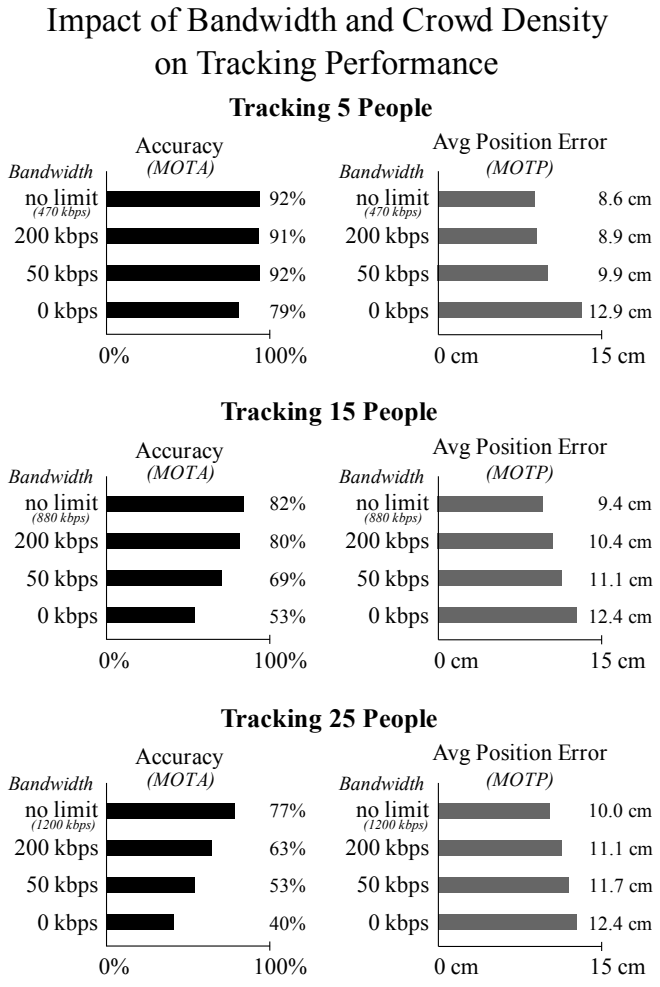


Fig. 10. Average tracking accuracy and precision in Exp. C for scenarios with 5, 15, and 25 people under 4 different communication budgets.

volume of its neighbors. For each crowd size of 5, 15, and 25 people we performed tracking under four communication budgets: no communication, 50 kbps, 200 kbps, and an unlimited budget where all tracked features are communicated to each neighbor. Figure 9 shows example tracks and Figure 10 shows the MOTA and MOTP performance metrics for each trial.

Tracking precision ranged from 12.9 cm down to 8.6 cm and consistently improved as the communication budget was increased. As expected, tracking accuracy improved with larger communication budgets and decreased as the number of targets increased. For the scenario with 5 people, a good tracking accuracy of 92% was achieved with a small communication budget of 50 kbps. For the scenario with 15 people, a tracking accuracy of 80% was achieved with a communication budget of 200 kbps. This compares favorably to the accuracy reported in [10] in which only 6 people were tracked. It is interesting to note that nearly the same tracking accuracy was obtained when sensors exchanged only a much smaller random subset of their observations as when exchanging all their observations in the 5 and 15 person scenarios.

The best achieved tracking accuracy dropped from 92% in the scenario with 5 people to 77% in the scenario with 25 people. Tracking accuracy was reduced at higher target densities because there were fewer features tracked on targets in the center of the crowd. This suggests that the 6 sensor deployment used in this experiment is suitable for tracking up to about 15 targets but additional sensors with different viewpoints should be deployed to provide accurate tracking at higher target densities.

8. CONCLUSION

We presented a distributed vision-based technique for tracking multiple people with multiple cameras suitable for cluttered and dynamic environments. The approach is designed for scenes where background modeling is difficult and significant occlusions can occur. The particle filter tracking technique we propose performs well using fewer than 100 particles per person and can track multiple people in real-time on commodity PC hardware. In evaluations on real and simulated scenes, it achieves a tracking precision of 10-30 cm and good tracking accuracy even in crowded scenes of 15 people.

In future work, we plan to investigate how sensors can more efficiently share feature trajectory measurements with their neighbors using selection and compression and also how to estimate the number of sensors required to achieve a desired tracking performance in a particular setting. Also it would be interesting to implement the sparse 3D feature tracking module on a real-time smart camera platform like the NXP WiCa mote [24].

Acknowledgments: This work has been supported under ONR grant N000140710747, Multidisciplinary University Research Initiative (MURI), by ARO grant W911NF-06-

1-0275, and by NSG grant CCF-0634803.

9. REFERENCES

[1] Keni Bernardin, Alexander Elbs, and Rainer Stiefelhagen, "Multiple object tracking performance metrics and evaluation in a smart room environment," in *Sixth IEEE International Workshop on Visual Surveillance*, May 2006, Graz, Austria.

[2] Wei Qu, Dan Schonfeld, and Magdi Mohamed, "Distributed bayesian multiple-target tracking in crowded environments using multiple collaborative cameras," in *Journal on Advances in Signal Processing*. 2007, Hindawi Publishing Corporation.

[3] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer, "Multi-camera multi-person tracking for easyliving," in *Third IEEE International Workshop on Visual Surveillance*, 2000.

[4] T. Darrell, D. Demirdjian, N. Checka, and P. Felzenszwalb, "Plan-view trajectory estimation with dense stereo background models," in *Proceedings of the International Conference on Computer Vision*, 2001.

[5] Michael Harville, "Stereo person tracking with short and long term plan-view appearance models of shape and color," in *IEEE International Conference on Advanced Video and Signal based Surveillance*, 2005.

[6] T. Zhao, M. Aggarwal, R. Kumar, and H. Sawhney, "Real-time wide area multi-camera stereo tracking," in *Computer Vision and Pattern Recognition*, 2005, pp. 976–983.

[7] Anurag Mittal and Larry S. Davis, "M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo," in *European Conference on Computer Vision*, 2002, pp. 18–36.

[8] Tatsuya Osawa, Xiaojun Wu, Kaoru Wakabayashi, and Takayuki Yasuno, "Human tracking by particle filtering using full 3d model of both target and environment," in *The 18th International Conference on Pattern Recognition*, 2006.

[9] Saad M. Khan and Mubarak Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *European Conference on Computer Vision*, 2006.

[10] A. Lopez, C. Canton-Ferrer, and J.R. Casas, "Multi-person 3d tracking with particle filters on voxels," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2007, vol. 1, pp. 913–916.

[11] H. Tsutsui, J. Miura, and Y. Shirai, "Optical flow-based person tracking by multiple cameras," in *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, August 2001, pp. 91–96.

[12] CY Tang, YP Hung, SW Shih, and Z Chen, "3d feature-based tracker for multiple object tracking," in *Proceedings National Science Council Republic Of China - Part A*, 1999, vol. 23, pp. 151–168.

[13] Tomas Svoboda, Daniel Martinec, and Tomas Pajdla, "A convenient multi-camera self-calibration for virtual environments," Tech. Rep., Swiss Federal Institute of Technology, 2005.

[14] Stanislav Funiak, Carlos E. Guestrin, Mark A. Paskin, and Rahul Sukthankar, "Distributed localization of networked cameras," in *Fifth International Symposium on Information Processing in Sensor Networks*, 2006.

[15] R. E. Kalman, "A new approach to linear filtering and prediction problems," in *Transactions of the ASME - Journal of Basic Engineering*, 1960, vol. 82, pp. 35–45.

[16] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[17] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[18] L. Matthies and Steven Shafer, "Error modeling in stereo navigation," *IEEE Journal of Robotics and Automation*, , no. 3, pp. 239 – 250, 1987.

[19] A. Doucet, J.F.G. des Freitas, and N.J Gordon, *Sequential Monte Carlo Methods in Practice*, Springer Verlag, New York, 2001.

[20] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," in *International Journal of Computer Vision*, 1998, vol. 28, pp. 5–28.

[21] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics*, MIT Press, 2005.

[22] "CHIL - computers in the human interaction loop," http://chil.server.de.

[23] "VACE - video analysis and context extraction," http://www.ic-arda.org.

[24] R. Kleihorst, B. Schueler, and A. Danilin, "Architecture and applications of wireless smart cameras (networks)," *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, pp. IV–1373–IV–1376, 2007.