Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 638073, 14 pages
doi:10.1155/2008/638073

Research Article

Robust Real-Time 3D Object Tracking with Interfering Background Visual Projections

Huan Jin[1, 2] and Gang Qian[1, 3]

[1] *Arts, Media and Engineering Program, Arizona State University, Tempe, AZ 85287, USA*
[2] *Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287, USA*
[3] *Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287, USA*

Correspondence should be addressed to Huan Jin, huan.jin@asu.edu

Received 2 November 2007; Revised 8 March 2008; Accepted 9 May 2008

Recommended by Carlo Regazzoni

This paper presents a robust real-time object tracking system for human computer interaction in mediated environments with interfering visual projection in the background. Two major contributions are made in our research to achieve robust object tracking. A reliable outlier rejection algorithm is developed using the epipolar and homography constraints to remove false candidates caused by interfering background projections and mismatches between cameras. To reliably integrate multiple estimates of the 3D object positions, an efficient fusion algorithm based on mean shift is used. This fusion algorithm can also reduce tracking errors caused by partial occlusion of the object in some of the camera views. Experimental results obtained in real life scenarios demonstrate that the proposed system is able to achieve decent 3D object tracking performance in the presence of interfering background visual projection.

1. INTRODUCTION

Movement-driven mediated environments attract increasing interests many interactive applications, including mediated and interactive learning, performing arts, and rehabilitation, just to name a few. The well-known immersive virtual reality system CAVE [1] is a good example of such movement-driven interactive environments. A mediated environment has both a movement sensing/analysis module and a feedback module. Users interact with the environment through their body movements (e.g., 2D/3D locations, facing direction, gestures), and/or by manipulating objects being tracked. Based on the movement analysis results, the feedback module produces real-time video and audio feedback which correlates with the users' movement. Figures 1(a) and 1(b) show examples of object tracking in mediated environments.

Although a large amount of effort has been made to develop video-based human movement analysis algorithms, (e.g., [2, 3] see for recent literature survey), reliable tracking and understanding of human movement in a mediated environment remain a significant challenge to computer vision. In this paper, we focus our discussion on robust 3D object tracking in complex environment with dynamic, interfering background visual projections. Objects can be tracked using different sensing modalities. In spite of many commercially available object tracking systems (e.g., InterSense IS-900 Precision Motion Tracker [4] built on hybrid ultrasound-inertial tracking technology, and Flock of Birds electromagnetic tracking from Ascension, Vt, USA [5]), video-based systems pose as an attractive solution to object tracking mainly due to its low cost. However, reliable and precise tracking of multiple objects from video in visually mediated environments is a nontrivial problem for computer vision. Visual projections used as part of the real-time feedback in a mediated environment often present a fast-changing and dynamic background. Moreover, in many applications, the background projections contain visual patterns similar to the appearance of objects being tracked in terms of color, brightness, and shape. We coin such background visual patterns *interfering background projection* since they interfere with vision-based tracking. False objects introduced by such interfering background can easily cause tracking failure if no enough care is taken. Multiple users interacting with

Table 1: Characteristics of tracking systems.

	Flock of birds [5]	InterSense IS-900 [4]	Vision-based tracker (proposed)
Data acquisition	Electromagnetic fields	Ultrasound/inertia sensors	Color and IR cameras
Degree of freedom (DOF)	6 (location+orientation)	6 (location+orientation)	3 (location only)
Maximum tracking area (m^2)	36	140	225
Accuracy (location) (mm)	1-2	2-3	4-5
Sensitivity (jitter)	Sensitive to ambient Electromagnetic environment	Sensitive to number and Distribution of emitters	Sensitive to IR sources
Update rate (*frames per second*)	144	180	40–60 (number of objects dependent)
cost	Medium	High ($29,900)	Low ($6,000 with 6 cameras+1 PC)

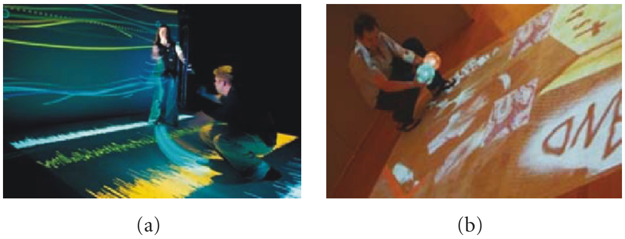

(a) (b)

Figure 1: Examples of mediated environments with interfering background visual projections ((a) photo courtesy of Ken Howie, Studios, copyright 2007 Arizona State University, (b) photo courtesy of Tim Trumble Photography, copyright 2007 Arizona State University).

each other in the environment often make the objects being tracked partially or fully occluded in some of the camera views. In addition, to increase the visibility of the visual projection, the lighting condition of the environment is often dimmed and suboptimal for video tracking. Reliable object tracking in 3D space with dynamic interfering background presents a challenging research problem. Many existing object tracking algorithms focus on robust 2D tracking in cluttered/dynamic scenes [6–8]. For example, [9] utilized the epipolar geometry in a particle filtering framework to handle object occlusions in crowded environments using multiple collaborative cameras. Some 3D object tracking algorithms utilize background subtraction and temporal filtering techniques to track objects in 3D space with the assumption of simple or stationary background [10, 11]. References [12, 13] used the planar homography to establish the object correspondence in overlapping field of view (FOV) between multiple views. Thus, tracking is limited in 2D plane coordinate systems essentially. Different from these existing methods, our approach uses the planar homography to remove the 2D false candidates on the known planes, and objects are tracked in 3D space. Moreover, to the best of our knowledge, no existing vision-based systems have been reported to be able to reliably track objects in 2D or 3D in environments with the interfering background projections that we are dealing with in this paper.

To overcome the aforementioned challenges, in this paper we present a working system we have developed for real-time 3D tracking of objects in a mediated environment where interfering visual feedback is projected onto the ground and vertical planes. The ground plane covered by white mats using an overhead projector through a reflecting mirror. The vertical plane visualizes the projections from a projector outside the tracking space. To deal with the dimmed lighting conditions, we use custom-made battery-powered glowing balls with built-in color LEDs as the objects to be tracked by the system. Different balls emit different color light spectrums. To alleviate the ambiguity caused by visual projections, we adopt a multimodal sensing framework using both color and infrared (IR) cameras. IR cameras are immune to visual projections while color cameras are necessary to maintain the target identities. IR-reflective patches were put on the balls to make them distinct from the background in the IR cameras. Homography mappings for all camera pairs with respect to the planes are recovered and used to remove false candidates caused by the projections on those planes. To better handle occlusions and minimize the effects on the objects' 3D locations caused by partial occlusion or outliers, we use a mixture of Gaussian to represent the multimodal distribution of all objects being tracked for each frame. Each mixture component corresponds to a target object with a Gaussian distribution. The kernel-based mean-shift algorithm is deployed for each object to find the optimal 3D location. Kalman filtering finally smoothes 3D location and provides a predicted 3D location for outlier rejection and kernel bandwidth selection. The proposed tracking system has been tested in various real-life scenarios, for example, for embodied and mediated learning [14]. Satisfactory tracking results have been obtained. Table 1 summarizes comparison of the proposed vision-based tracking with the state-of-the-art tracking techniques, namely Ascension's Flock of Birds and the IS-900 system from InterSense, Mass, USA. Clearly the proposed vision-based tracking system is much cheaper than the other two popular tracking systems, while with comparable location tracking performance. As mentioned in the future work, we are extending the proposed system by including inertial sensors for the orientation recovery of the objects.

2. OBJECT APPEARANCE MODEL

To ensure the visibility of the visual feedback, the ambient illumination needs to be on a dim level. As a result, a color object is very hard to be seen by color cameras. The object identity cannot be maintained in such low ambient illumination. Therefore, we use custom-designed battery-driven glowing balls with sufficient built-in color LEDs as tracking objects. Sufficient small IR-reflective patches are attached on them evenly so that they can be detected as bright blobs in IR cameras lightened by infrared illuminators. The reason that we use infrared illuminators instead of plugging in infrared LEDs in glowing balls is due to the fact that infrared LEDs consume much more power than color LEDs. A fully charged battery usually can last about two hours to power a glowing ball at sufficient brightness level for tracking. In fact, the tracking objects can be in any shape since the 2D shape is not exploited in tracking. In our experiments, we use balls as tracking objects since they are easy to be manipulated by subjects, for example, tossed between two interacting subjects. Multiple objects can be tracked by our proposed system. One assumption made in the system development is that different objects are identifiable by their unique colors, that is, two objects cannot share the same color.

2.1. Color model

Color histogram provides an effective feature for object tracking as it is computationally efficient, robust to partial occlusion, and invariant to rotation and 2D size scaling. We adopt hue, saturation, value (HSV) color space because it separates out hue (color) from saturation and brightness channels, and hue channel is relatively reliable to identify different color objects under varying illuminations. A color histogram H_j for the target object j is computed in the tracking initialization stage using a function $b(\mathbf{q}_i) \in \{1, \dots, N_b\}$ that assigns the hue value $\mathbf{q}_i$ to its corresponding bin:

$$\mathrm{H}_j = \{h^{(u)}(R_j)\}_{u=1\dots N_b} = \lambda \sum_{i=1}^{N_{R_j}} \delta[b(\mathbf{q}_i) - u], \qquad (1)$$

where δ is the Kronecker delta function, λ is the normalizing constant, and N_{R_j} is the number of pixels in the initial object region R_j. In our practice, we divide the hue channel into $N_b = 16$ bins in order to make the tracker less sensitive to color changes due to visual projections on the objects. We observe that a glowing ball emits stable color spectrums in two hours given that the battery is fully recharged. The histogram is not updated during the tracking in that the visual feedback might be projected on objects.

2.2. Gray-scale threshold

We use gray-scale thresholding method to detect the reflective objects that have bright blobs in IR camera views. Since all objects being tracked share the same reflective material, they have the same gray scale lower- and upper-bound thresholding parameters $\mathbf{T} = \{\mathrm{T}_{\min}, \mathrm{T}_{\max}\}$. The gray-scale thresholding parameters are determined in the tracking initialization stage and need to be adjusted only when the infrared spectrum of the ambient illumination is substantially changed.

3. MULTIVIEW TRACKING

3.1. System overview

An overview of the proposed tracking system is given by the diagram shown in Figure 2. We will briefly introduce each module.

Initialization

The tracking initialization is to manually obtain camera projection matrices, homogeneous plane coefficients, and the histogram of objects, and the gray-scale thresholds for IR cameras. It is a one-time process, and all the parameters can be saved for the future use.

2D localization and outlier rejection

Object histogram and gray-scale thresholds are used to locate the target in the color and IR camera view, respectively. 2D search region predicted by Kalman filtering helps removing 2D false candidates.

2D pairwise verification

A pair of 2D candidates from two different views is examined by epipolar constraint and planar homography test. Label information is considered in this step.

3D localization

Each valid pair corresponds to a 3D triangulation result. The unlabeled pair from two different IR cameras might be obtained from 2D false candidates such as reflective or bright spots. The predicted 3D location and velocity from Kalman filtering help detecting those false pairs.

Multiview fusion

The distribution of all the objects being tracked is modeled as a mixture of Gaussian. Each mixture component corresponds to a target object. The kernel-based mean-shift algorithm is employed for each target to find the optimal 3D location.

Kalman filtering and occlusion handling

Each target is assigned a Kalman filter that plays the role of a smoother and predictor. The partial occlusions are alleviated by the mean-shift algorithm while the complete occlusions that occur in some camera views are automatically compensated by other nonoccluded camera views.

3.2. System calibration and tracking initialization

System calibration consists of camera calibration, scene calibration, and object template extraction. The cameras

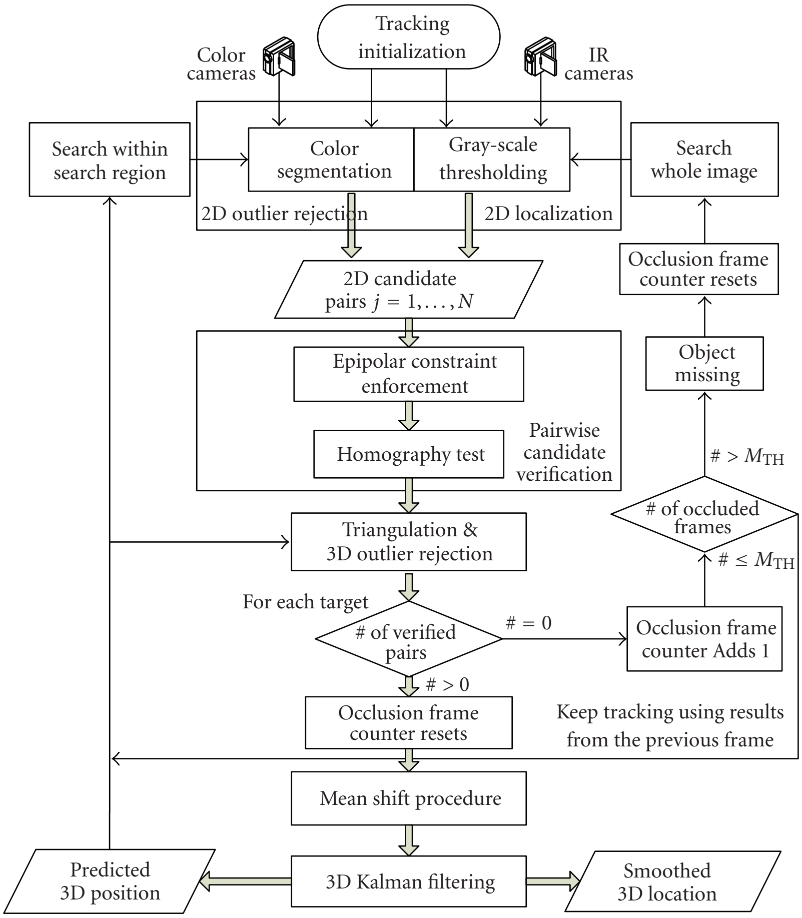


Figure 2: Diagram of the proposed object tracking system.

are calibrated using the multicamera self-calibration toolkit developed by Svoboda et al. [15]. We use a small krypton flashlight bulb as the calibration object that is visible in both color and IR cameras. Given N camera projection matrices $\{\mathbf{P}_i\}_{i=1}^N$, the fundamental matrices, $\{\mathbf{F}_{ij}\}_{i,j=1;i\neq j}^N$, which are useful to enforce the epipolar constraint, can be computed easily [16].

The purpose of the scene calibration is to obtain the equations of the 3D planes on which visual feedback is projected. In our case, they are the ground plane and vertical plane. The plane equation is used to get the planar homography which removes visual projections on the plane. The reason that we do not use 2D feature correspondences to compute homography is that the color points projected on the plane are invisible in IR cameras. Thus, we fit a plane based on the 3D point locations computed by triangulation using the 2D feature correspondences in color cameras. Once we have the 3D plane equations and camera projection matrices, planar homography $\{\mathbf{H}_{ij}\}_{i,j=1;i\neq j}^N$ can be computed between any different cameras.

An object template includes object's gray-scale threshold, color histogram, and 2D size range in each camera view. Currently, the object templates are obtained manually as part of the system calibration process. The color histogram is computed for each color camera by choosing the area of an object in that camera view, while the gray-scale thresholds are learned by waving a reflective ball evenly in 3D mediated space and then extracting blobs' minimum and maximum brightness values in IR cameras. The 2D size of an object is useful to reject 2D false candidates in each camera view. The 2D size range (i.e., minimum and maximum size values) for each object is retrieved by counting the number of pixels in the object's corresponding segmented region in each camera view.

System calibration and object templates can be saved in the system configuration, from which tracking can be restored efficiently and bypasses the initialization stage whenever it restarts. An object template needs to be reinitialized only if its color is changed.

3.3. 2D localization and outlier rejection

In our mediated environment, the background subtraction is not helpful to detect desired objects because of the projected visual feedback and human interaction in the space. Therefore, we directly segment out the object candidates based on its color histogram and gray-scale thresholds for the color and IR cameras, respectively. Given a color image, we identify all the pixels with hue values belonging to the object's color histogram. Given a gray-scale image from an IR camera, we segment out all the pixels whose brightness values are within $\mathrm{T}_{\min}$ and $\mathrm{T}_{\max}$. Then, we employ connected component analysis to get all pixel-connected blobs in both gray-scale or color image. A blob is valid only if its size is within the corresponding object's size range. Finally, we take the blob's center to represent its 2D coordinates. Size control is useful to combat the background projection, especially when a ball is submerged by a large projection with similar color in some color camera views. After size control, the projection will be removed from the list of valid 2D candidates. Essentially, the ball is considered to be "occluded" by the projection in such cases.

For color images, we further verify each valid blob l by comparing its histogram H_l, with the object template histogram H using the Bhattacharyya coefficient metric, which represents a similarity metric with respect to the template histogram. The metric is given by

$$\rho(\mathrm{H}_l, \mathrm{H}) = \sum_{b=1}^{N_b} \sqrt{h_l^{(b)} h^{(b)},} \tag{2}$$

where N_b is the number of bins, and $h_l^{(b)}$ is the normalized value for bin b in the lth blob's histogram H_l. $\rho(\mathrm{H}_l, \mathrm{H})$ ranges from 0 to 1, with 1 indicating a perfect match. A blob is a valid object candidate only if its Bhattacharyya coefficient is greater than a certain similarity threshold T_s. A small threshold $\mathrm{T}_s = 0.8$ was taken because of high occurrence of uniformly colored targets in our applications.

To remove 2D outliers introduced by cluttered and dynamic scenes, we specify a search region. It is given by a circular region centered at the predicted 2D location which is reprojected from 3D Kalman predicted location (3.7). The radius of the search region $T_{\mathrm{2D}}^{(k,i)}(t)$ for object k in camera view i is determined by

$$T_{\mathrm{2D}}^{(k,i)}(t) = \begin{cases} \mathrm{TH}, & \Delta_{\mathrm{2D}}^{(k,i)}(t) \leq \mathrm{TH}, \\ \alpha \cdot \Delta_{\mathrm{2D}}^{(k,i)}(t), & \text{otherwise}, \end{cases} \tag{3}$$

where $\Delta_{\mathrm{2D}}^{(k,i)}(t)$ is the 2D pixel distance, the object is expected to travel from $t-1$ to t. Given the final location estimates of the object at $t-1$ and $t-2$, the 2D locations of object k in camera view i in the two previous frames can be found. $\Delta_{\mathrm{2D}}^{(k,i)}(t)$ is then computed as the pixel distance between these two 2D locations. α is a scaling factor. TH is the lower bound for the search radius. In our implementation, we set $\alpha = 1.5$ and $TH = 10$ pixels. A 2D blob for the target is identified as an outlier if it is out of the target's search region. After this 2D outlier rejection step, for each object k a candidate list of 2D locations $\{X_{k,i}^{(n)}(t)\}$ is formed in every camera view i at time t. n is the index of the valid blob in 2D candidate list.

In an IR camera view, the 2D candidate list of an object consists of the blobs within the search region of the object. In a color camera view, the valid 2D candidates of an object are blobs inside the search region with a histogram similarity (Bhattacharyya distance) above a threshold to that object. Please note that when two objects are close to each other in an IR camera view with overlapping search areas, a 2D candidate is allowed to be included in the candidate lists of both objects.

3.4. 2D candidate pair verification

Point correspondences from two or more camera views are needed to compute 3D locations using triangulation. An initial set of candidate pairs $\{X_{k,i}^{(n)}(t), X_{k,j}^{(m)}(t)\}_{i<j}$ is formed by pairing up 2D candidates in different views. When these pairs are formed, 2D candidates associated with different objects are not allowed to be paired up. The resulting list of 2D candidate pairs might include false candidate pairs not corresponding to any object, such as pairs related to floor projections, or pairs not related to any physical objects in the space.

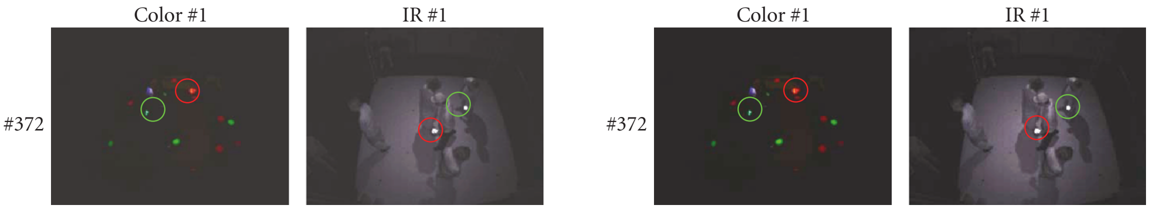
To remove such false pairs, we first verify each pair by the epipolar constraint, that is, $X_2^T \mathbf{F}_{21} X_1 = 0$. Pairs with epipolar distance $ED(X_2, X_1) < \mathrm{T}_{ED}$ are classified into the valid pair set $\mathcal{X}^e$, where T_{ED} is the epipolar distance threshold.

Due to the visual feedback projected on the ground plane or vertical plane, some projections sharing a similar color histogram with the target may be observed in two color camera views. Such projections satisfy the epipolar constraint. To remove projections, we apply the planar homography test against the pairs in $\mathcal{X}^e$. The pair that has passed the homography test, that is, $\|X_2 - \mathbf{H}_{21} X_1\| > \mathrm{T}_H$, is not corresponding to projections and will be put into the final valid pair set $\mathcal{X}$, where T_H is the homography test threshold.

The object, however, may actually be laid on or close to one of the planes for visual projection. To prevent valid pairs from being removed by the homography test in this case, we first search through all the color-IR pairs to see if there is any color-IR pair $\{X_2^e, X_1^e\} \in \mathcal{X}^e$ satisfying $\|X_2 - \mathbf{H}_{21} X_1\| < \mathrm{T}_H$. If there are such color-IR pairs, there is a good chance that the ball is on the floor. All the indices of related color blobs in those color-IR pairs are recorded in a valid blob list B. If a color-color candidate pair fails the homography test (i.e., they satisfy the homography constraint w.r.t. the ground plane), but one of the blobs is in B, meaning that the ball is on the plane and it is expected for the related color blobs to fail the homography test, this color-color pair is still regarded as a valid pair and put into the final valid pair set $\mathcal{X}$. In our experiment, we set both T_{ED} and T_H to a small value (3 pixels). After this step of 2D candidate verification, $\mathcal{X} = \{X_n\}_{n=1}^{N_p}$ is established, where N_p is the number of valid 2D pairs.

3.5. Triangulation and outlier rejection

The 3D positions of the objects need to be computed using the filtered list of 2D pairs. Triangulation is commonly used to localize a 3D point from its 2D projections in two or more camera views. Although multiple 2D pairs of the same object can be triangulated to find the corresponding 3D location, it is challenging to reliably segment IR-IR pairs



(a) Avg. 2D reproj. error = 1.472 pixels

(b) Avg. 2D reproj. error = 0.413 pixel

Figure 3: Comparison of two fusion methods. (a) 3D location is computed by taking the mean of all 3D points; (b) 3D location is obtained by kernel-based mean-shift algorithm.

to multiple objects. In our proposed approach, 2D pairs on the list are first triangulated separately to generate a list of 3D candidates. Then, the final position estimates of the objects are obtained through a multiview fusion procedure as detailed in Section 3.6.

For each pair $X_n \in \mathcal{X}$, the corresponding 3D candidate location Z_n is obtained using triangulation. A set of 3D candidates $\{Z_n\}_{n=1}^{N_p}$ can be created. Recall that each object is identified or labeled by a unique color emitted from the built-in LEDs. A 3D candidate obtained from a non IR-IR pair (i.e., a color-color pair or a color-IR pair) carries the object label because of the availability of color information. Such color 3D candidates are also called *labeled candidates*. On the other hand, IR cameras are color-blind. As a result, a 3D candidate obtained from an IR-IR pair has no color labels associated to it. These candidates are referred to as *unlabeled candidates*. Although an IR-IR pair does associate to an object based on the 2D search regions, this association is not exploited to label IR-IR pairs since one IR-IR pair might associate to more than one objects if these objects are close to each other in the 3D space. Instead of explicitly assigning multiple labels to 3D candidates from such ambiguous IR-IR pairs, all IR-IR pairs and the resulting 3D candidates are simply treated as unlabeled candidates. Section 3.6 shows how these unlabeled candidates are fused with the labeled candidates through mean shift to find the final estimate of the 3D object locations.

Although the 2D pair list has been filtered using the epipolar and homography constraints, it is still possible that some of the 3D candidates do not actually relate to any objects being tracked. A false 2D pair candidate appears to be valid if the two points happen to be on their epipolar lines. A false 3D candidate can also be formed by projections on human body. Similarly reflective surfaces and spots, such as watches and glasses, may form as false IR-IR pairs. These pairs are still alive through the pairwise candidate verification. To eliminate the outliers caused in these scenarios, we perform the following 3D outlier rejection step based on the Kalman predication introduced in Section 3.7.

Let $Z_{k,t}^{(n)}$ be the nth 3D candidate for object k at time t, and let $\widetilde{Z}_{k,t}$ be the corresponding prediction from the Kalman filter. $Z_{k,t}^{(n)}$ is considered to be within a search sphere of object k if $\|Z_{k,t}^{(n)} - \widetilde{Z}_{k,t}\| \leq \mathrm{T}_{3\mathrm{D}}^{(k)}(t)$, where $\mathrm{T}_{3\mathrm{D}}^{(k)}(t)$ indicates the tolerance of outliers at time t for object k. A labeled 3D candidate outside the search sphere centered at the associated object is rejected as an outlier. So is an unlabeled candidate outside the search spheres of all the objects. Similar to the 2D outlier rejection by a circular search region, a spherical search region is used to remove outliers. $\mathrm{T}_{3\mathrm{D}}^{(k)}(t)$ is set to be $\beta \cdot \Delta_{3\mathrm{D}}^{(k)}(t)$, where $\Delta_{3\mathrm{D}}^{(k)}(t)$ is the expected displacement of the object between $t-1$ and t. $\Delta_{3\mathrm{D}}^{(k)}(t)$ is estimated as the distance between the final 3D location estimates of the object at $t-1$ and $t-2$. Theoretically, the scaling parameter β depends on the frame rate of the system and the motion of the object. However, since the frame rate of the system is nearly constant within a short period of time, it is reasonable to use a constant scaling parameter. To cope with possible abrupt object motion, we set $\beta = 2$ which has proven to be working well in our experiments.

A lower bound for $\mathrm{T}_{3\mathrm{D}}^{(k)}(t)$ is also set to secure a valid search region for slowly moving or nearly static objects. In our experiments, this lower bound is set to be 22.5 cm. After rejecting outliers, the list of verified 3D candidates $\mathcal{C}$ is generated. At the same time, each individual object also has its own sublist of 3D candidates based on the color (only applicable to labeled candidates) and position of the 3D candidates. Please also note that one unlabeled candidate can be included in the sublists of two or more close-by objects.

3.6. Multiview fusion

Given the list of 3D candidates $\mathcal{C}$, we need to estimate the 3D locations of the objects being tracked. A straightforward solution is to compute the mean of the 3D candidates on the sublist of an object and use that as the position estimate of the object. However, this method is error-prone since the sublist of 3D candidates of an object and the 3D candidates are noisy. Some of the candidates are biased due to partial occlusion when the object is only partially visible in one or more video cameras, resulting in an inaccurate 2D centroid extraction. In addition, when two objects (e.g., A and B) are close in 3D space, a 3D unlabeled candidate of A might be in the neighborhood of B, and vice versa. Thus, sublist of candidates of an object might contain some 3D candidates which actually belong to some other nearby objects.

To tackle this issue, the location distribution of all the objects being tracked is considered to be a mixture of Gaussian (MoG), with each mixture component corresponding to one object. Object tracking is cast into a mode seeking problem using the 3D candidates. Consequently, the effect of missing 3D candidates due to the complete occlusion in some views, inclusion of unlabeled candidates of other objects, and biased 3D localization caused by partial occlusion on the tracking results will be significantly reduced by selecting proper forms of kernel and weight functions. Since the goal is to locate the modes from the 3D candidates, instead of learning the complete parameters of the MoG using the expectation-maximization (EM) algorithm, a fast mode-seeking procedure based on mean shift is taken in our proposed approach. Figure 3 shows comparison of two fusion methods with associated 2D reprojection errors, namely, the mean of 3D candidates and the mode obtained by kernel-based mean-shift algorithm. It can be seen that mean shift provides smaller reprojection error.

The mean-shift algorithm is an efficient and nonparametric method for clustering and mode seeking [17–19] based on kernel density estimation. Let S be a finite set of sample points. The weighted sample mean at $\mathbf{x}$ is

$$m(\mathbf{x}) = \frac{\sum_{\mathbf{s}\in S} K_{\mathbf{h}}(\mathbf{s}-\mathbf{x})w(\mathbf{s})\mathbf{s}}{\sum_{\mathbf{s}\in S} K_{\mathbf{h}}(\mathbf{x}-\mathbf{s})w(\mathbf{s})}, \qquad (4)$$

where $K_{\mathbf{h}}(\cdot)$ is the kernel function, $\mathbf{h}$ is the bandwidth parameters, and $w(\cdot)$ is the weight function. Mean shift recursively moves the center of the kernel to a new location by the mean-shift vector $\Delta\mathbf{x} = m(\mathbf{x}) - \mathbf{x}$. The mean-shift procedure is guaranteed to be convergent if the kernel $K(\mathbf{x})$ has a convex and monotonically decreasing profile $k(\|\mathbf{x}\|^2)$ [18]. An important property of the mean-shift algorithm is that the mean-shift vector computed using the kernel G is an estimate of the normalized density gradient computed using the kernel K, where G satisfies the relationship $g(\|\mathbf{x}\|^2) = -ck'(\|\mathbf{x}\|^2)$, c is a normalizing constant. g and k are the profiles of kernel G and K, respectively. In order to facilitate real-time implementation, we take advantage of the intermediate results, namely, the prediction and covariance matrix of Kalman filtering (see Section 3.7) to approximate the initial center and bandwidth.

Integration of mean shift and Kalman filter has been introduced in [18], where mean shift is used to locate the optimal target position in an image based on the prediction from a Kalman filter. Then, the result of mean shift is used as the measurement vector to the Kalman filter. Our proposed approach for multiview fusion follows [18] in spirit in terms of the relationship of mean shift and Kalman filter. The major difference between our approach and is that we use mean shift as a fusion mechanism to find optimal object 3D locations using both labeled and unlabeled 3D position candidates obtained from triangulation of point pairs.

Center initialization

A good initial center will expedite the convergence of the mean-shift procedure. For each object being tracked, there is one corresponding center initialized. In our approach, $\mathbf{y} = (\mathbf{y}_x, \mathbf{y}_y, \mathbf{y}_z)$, the predicted location by the Kalman filter at the previous frame is taken as the initial center since $\mathbf{y}$ gives a good guess of object location at the current frame. Note that another possible choice for the initial center is taking the weighted mean of the 3D candidates. We compare the mean-shift iterations with these two initial centers and the same bandwidth using a 1420-frame video sequence (sequence 1). In the histogram shown in Figure 4, x-axis indicates the number of mean-shift iterations while y-axis indicates the number of frames corresponding to each iteration. For instance, to converge to optimal points, about 750 frames take 4 iterations using Kalman prediction as initial center, while about 600 frames take the same iterations using weighted mean as initial center. The iteration histogram shows that our approximation takes slightly more mean-shift iterations. But the approximation gains the benefit of no extra computation for initial center.

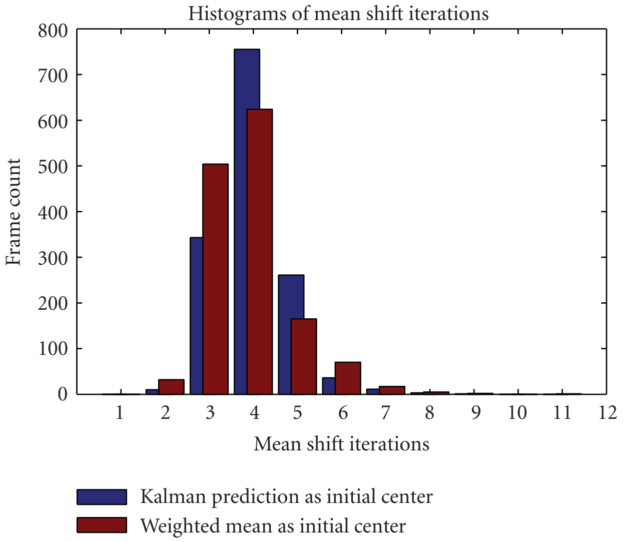


Figure 4: Histogram comparison of mean-shift iterations with different initial centers.

Bandwidth selection

A proper bandwidth for the kernel function is critical to the mean-shift algorithm in terms of estimation performance and efficiency [20]. Reference [21] lists four different techniques for bandwidth selection. In [22], the bandwidth selection theorem is shown that if the true underlying distribution of samples is a Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, for the fixed-bandwidth mean shift, the optimal bandwidth for a Gaussian kernel $K_{\mathbf{h}}$ that maximizes the bandwidth-normalized norm of the mean shift vector is given by $\mathbf{h} = \boldsymbol{\Sigma}$.

In our approach, MoG is used to represent the joint position distribution of all the object, and the marginal distribution of a single object is described by a single Gaussian. In practice, $\boldsymbol{\Sigma}$ the true covariance matrix of the Gaussian related to an object is unknown. However, $\widetilde{\boldsymbol{\Sigma}}$ the corresponding covariance matrix of the predicted object position in Kalman filter provides a good approximation to $\boldsymbol{\Sigma}$ since it resembles the uncertainty of the sample point

Table 2: Coefficients of fitted planes.

Plane	Homogeneous coefficients
Ground plane	$[-0.0022, 0.0039, 0.9999, 0.0132]$
Vertical plane	$[0.9889, -0.1338, 0.0628, 0.0144]$

distribution. Thus in our proposed approach, $\mathbf{h} = (h_x, h_y, h_z)$ is formed by the diagonal terms of $\tilde{\mathbf{\Sigma}}$.

Kernel setup

Labeled sample points have explicit identity information while unlabeled ones do not. Two different kernel functions are applied to labeled and unlabeled sample points. Both kernel functions share the same bandwidth $\mathbf{h}$. Since a labeled 3D candidate is expected to be within an ellipsoid centered at the mode, and all the 3D candidates within the ellipsoid are considered equally important from the perspective of the kernel function, a truncated flat kernel (6) is used for the labeled 3D candidates. Let $Q_{\mathbf{h}}$ be an ellipsoid centered at $\mathbf{y}$ with three axes given by $\mathbf{h}$:

$$Q_{\mathbf{h}}(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x}_x - \mathbf{y}_x)^2}{h_x^2} + \frac{(\mathbf{x}_y - \mathbf{y}_y)^2}{h_y^2} + \frac{(\mathbf{x}_z - \mathbf{y}_z)^2}{h_z^2} - 1. \tag{5}$$

The truncated flat kernel for labeled samples is given by

$$K_{\mathbf{h}}(\mathbf{x} - \mathbf{y}) = \begin{cases} 1, & \text{if } Q_{\mathbf{h}}(\mathbf{x}, \mathbf{y}) < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

On the other hand, we assume that the contribution to the mode estimation made by an unlabeled 3D candidate is less than that of a labeled candidate. The contribution of an unlabeled 3D candidate to the mode estimation is computed according to the distance from the unlabeled point to the mode so that a distant point has small contributions. Hence, a truncated Gaussian kernel (7) is applied to all unlabeled sample points as follows:

$$K_{\mathbf{h}}(\mathbf{x} - \mathbf{y}) = \begin{cases} \exp\left\{ -\dfrac{\|\mathbf{x} - \mathbf{y}\|^2}{\max^2(h_x, h_y, h_z)} \right\}, & \text{if } Q_{\mathbf{h}}(\mathbf{x}, \mathbf{y}) < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Weight assignment

All the sample points in the final set are the survivors through the 2D pairwise verification process and 3D outlier rejection. Each 3D point is associated with a small epipolar distance $d_{ED}(\mathbf{x}) < \mathrm{T}_{ED}$. Thus, the epipolar distance is a good indicator to represent the weight of each sample point. A Gaussian kernel (8) is used to compute the weight for each sample point:

$$w(\mathbf{x}) = e^{-d_{ED}^2(\mathbf{x})/\alpha}, \tag{8}$$

where $\alpha = 2\mathrm{T}_{ED}^2$.

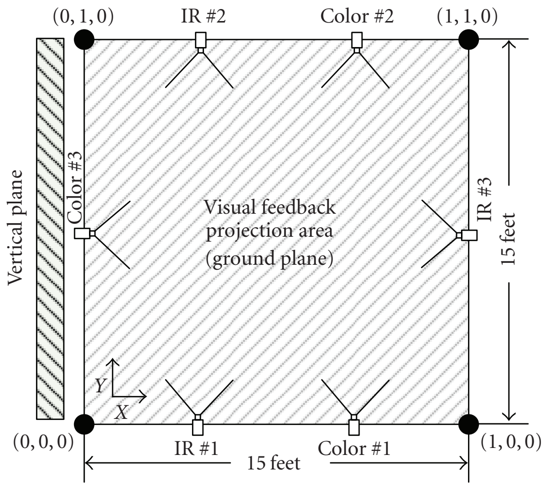


Figure 5: Illustration of the real system setup in top-down view.

The above mean-shift procedure iteratively updates the position estimates of all objects being tracked. The position of one object is considered converged when the norm of corresponding mean-shift vector is less than a prechosen threshold (e.g., 0.5 mm in our implementation).

3.7. Smoothing and prediction using kalman filter

In the proposed tracking system, Kalman filters are used to smooth position estimates obtained using mean shift and predict object positions for the next time instant. Each target object is assigned with a Kalman filter to perform smoothing and prediction based on a first-order motion model. The position estimates obtained using mean shift are treated as measurements and input to the Kalman filters. The smoothed 3D locations are then used as the final tracking results of the objects. It is possible, and perhaps desirable, to estimate the measurement noisy covariance matrix based on the results of the mean shift. In our implementation, to reduce computational cost a constant diagonal measurement noisy covariance matrix is used instead. The results obtained are still good enough for our applications. The predicted object locations are projected onto all camera views to form 2D search region for the next time instant. These 2D search regions can substantially reduce the number of 2D false candidate blobs as described in Section 3.3. Similarly, these location predictions also serve as centers of searching spheres for 3D outlier rejection as discussed in Section 3.5. In addition, the covariance matrices of the predicted positions also inform the bandwidth selection in Section 3.6.

3.8. Handling complete occlusion and tracking failure

When complete occlusion occurs in some camera views, the 2D candidates in other nonoccluded camera views can still be coupled to form pairs for triangulation. As long as an object is visible in at least two camera views, the tracking of the object can be consistently achieved. Sometimes due

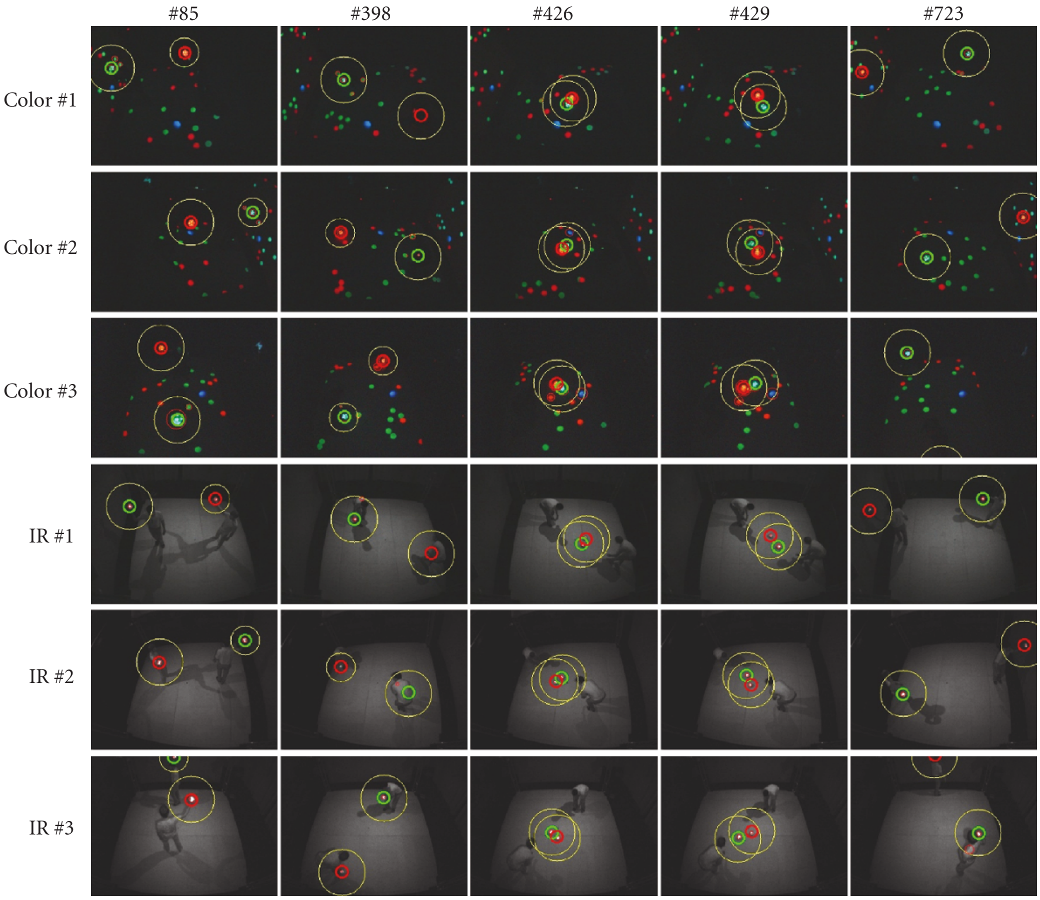


FIGURE 6: Two-object tracking results using sequence 1. A large amount of interfering visual patterns was present in the background. Bold green (red) circles indicate the reprojected 2D locations of the green (red) object. The slim red circles show the 2D candidates in each camera view. The large slim yellow circles are the predicted search regions for the removal of 2D false candidates. The mean (μ) and std (σ) of the reprojection errors in pixels are as follows. For the green object, $\mu_1 = 0.823$ and $\sigma_1 = 0.639$. For the red object, $\mu_2 = 0.758$ and $\sigma_2 = 0.607$. Note that the blue points are deliberately added as the interfering background projections, which are different from tracking objects' colors, to test the robustness of our tracking approach.

to severe occlusions or fast and abrupt moving direction changes, an object is only detected in one camera view or completely invisible by all the cameras. Consequently, no valid 2D pair of the object can be formed in such scenarios and the tracking system issues a tracking failure event for this object as no pair exists. To recover the tracking of the missing object, we search valid color-IR pairs in the whole image in both color and IR camera views, without enforcing distance-based outlier rejection scheme according to search regions. During the detection phase of a lost object, the existence of a color-color pair matching, the color of the object, or an IR-IR pair alone cannot claim the detection of the object since a color-color pair may be caused by background visual projection, and an unlabeled IR-IR pair does not provide any identity information and it may not actually correspond to any objects to be tracked. In this case, most of outliers are removed by color-IR pair selection, and the final result is refined by mean-shift-based multiview fusion. Once a lost object has been detected, the tracking can be resumed.

Occasionally, an occluded object will reappear again shortly at a position close to where it got lost. In such cases, there is no need to search the object in the entire image. To accommodate such scenarios, some extra steps are taken in our current system implementation as shown in the right column in Figure 2. In the current implementation, a missing-frame counter (MFC) is associated to each object being tracked. When an object is not visible in at least two cameras views, the corresponding MFC is triggered to count the number of frames that the object is continually missing. When the reading of the MFC is less than a certain threshold M_{TH} , for example, 25 frames in our implementation, the tracking system will keep trying to search for the object in all of the camera views within search regions according to its 3D location at the last time instant when the object was successfully tracked. If the object reappears in the search regions of two of more camera views during this period, the MFC will be set to zero and the tracking of the object continues. If the object cannot be found during

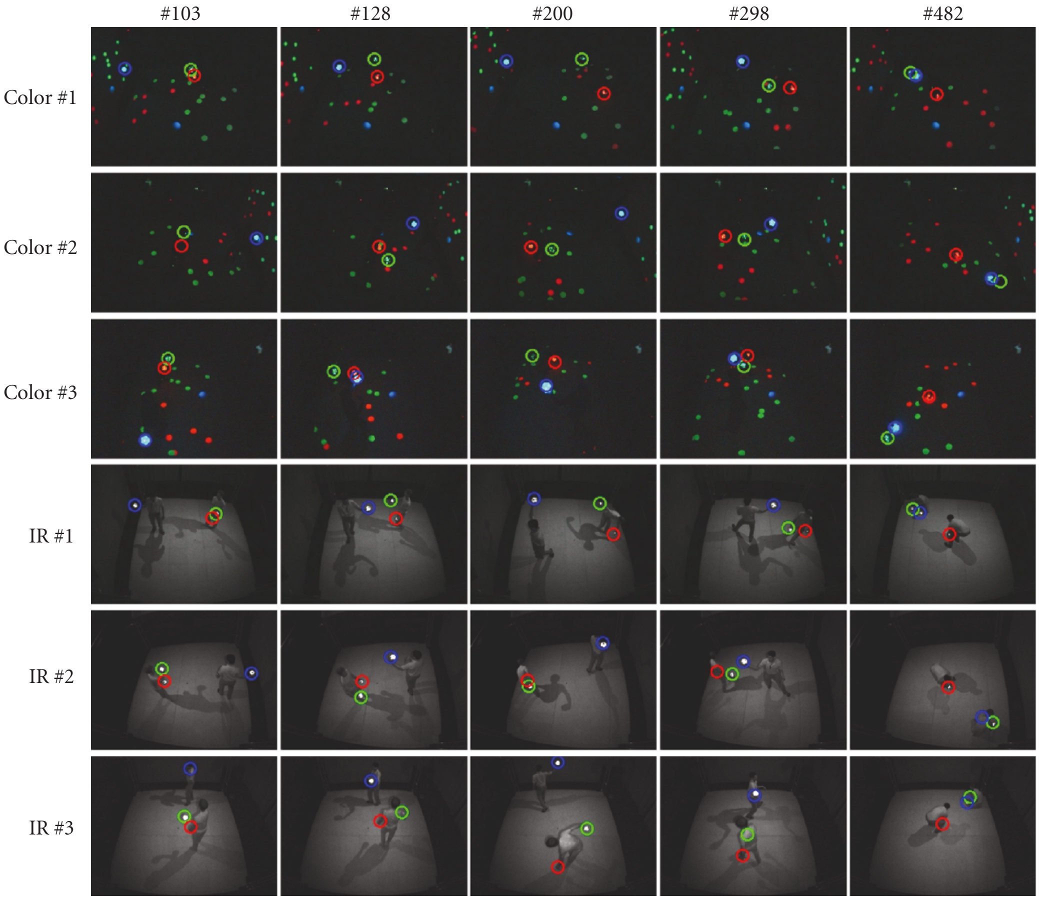


Figure 7: Three-object tracking results using sequence 2. The statistics of the reprojection errors in pixels are $\mu_1 = 0.788$ and $\sigma_1 = 0.609$ for the green object, $\mu_2 = 0.891$ and $\sigma_2 = 0.681$ for red object, and $\mu_3 = 0.884$ and $\sigma_3 = 0.523$ for the blue object.

this period, the MFC gets reset and the systems start to search for the object in the whole images from all the camera views according to the procedure described in the previous paragraph.

4. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

The proposed tracking system has been implemented using six CCD cameras (Dragonfly 2, Point Grey Research, Ariz, USA), including three color cameras and three IR cameras. Figure 5 illustrates the real system setup in top-down view. The cameras are mounted about 14 feet high from the ground on a steel truss. The tracking system software is programmed using C++ with multithreading on a PC (Intel Xeon 3.6 GHz, 1 GB RAM) running Windows XP. Image resolution is set to 320 × 240. All cameras are synchronized and calibrated in advance. The dimension of the space is aligned to predefined coordinate systems with 1 unit = 15 feet. To better cover the activity space, large FOVs are applied to all the cameras. So, lens distortion [16] is also recovered for each frame using the camera calibration toolbox [23]. Visual projections are projected onto two planes—the ground plane and a vertical plane parallel to y-axis at the boundary of the capture volume. Table 2 shows the homogeneous coefficients of the two planes. This system runs in real-time at 60 fps when tracking three objects, and 40 fps when tracking six objects. In this six-camera setting, approximately 10~18 pairs per object will appear in the final 2D pair set. The number of pairs varies based on occlusions and the number of neighboring objects. Occlusions cause missing pairs while neighboring objects result in more unlabeled IR-IR pairs appearing in the final set.

To evaluate and analyze the performance of the tracking system, a large amount of fast time changing visual patterns was projected onto the ground and vertical planes. These visual patterns often carry similar colors to the objects being tracked. In this section, tracking results obtained using three sequences are presented to demonstrate the robust tracking performance of the proposed system in the presence of interfering visual projection, occlusion, and multiple subjects in the space. Sequence 1 shows reliable tracking of two objects with interfering visual projection on the multiple projection planes. Sequence 2 has three objects and

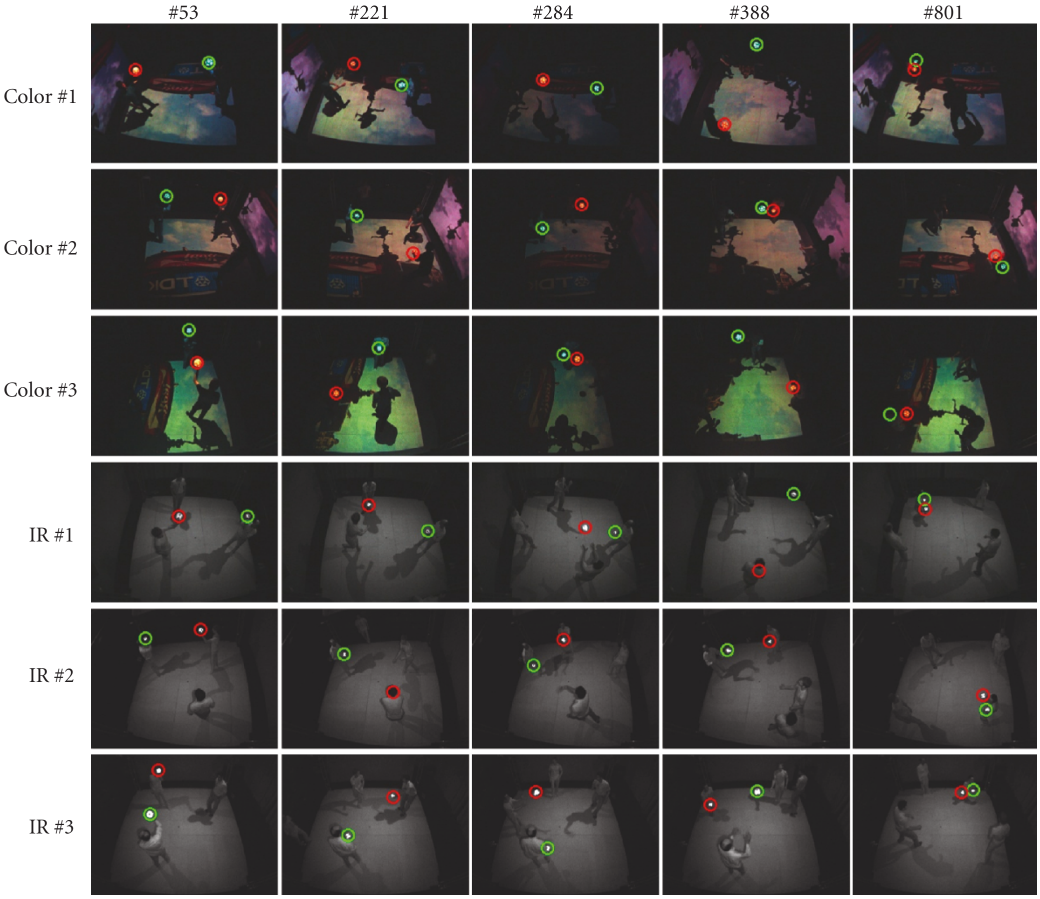
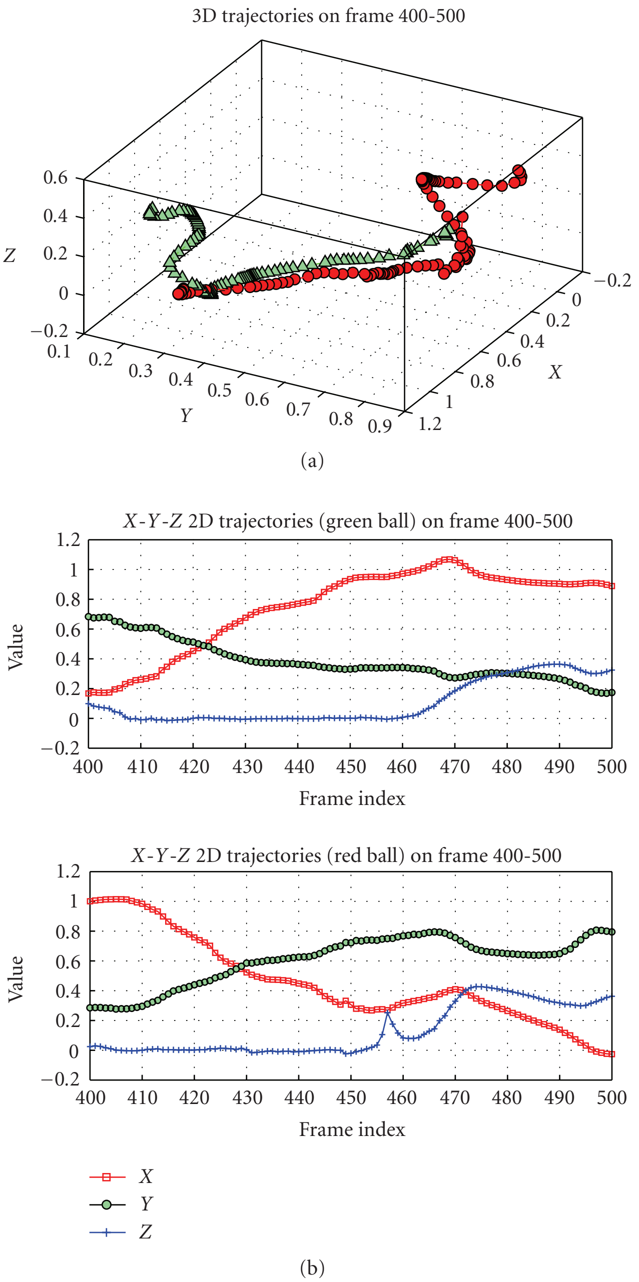


Figure 8: Two-ball tracking results using sequence 3. Four people were moving in the space and interacting with the mediated environment. The statistics of the reprojection errors in pixels are $\mu_1 = 0.732$ and $\sigma_1 = 0.511$ for the green object, and $\mu_2 = 0.755$ and $\sigma_2 = 0.549$ for the red object.

frequent occlusion. In sequence 3, four subjects interact with the mediated environment using two objects. Partial and complete occlusions also frequently occur in sequence 3.

In the first experiment using sequence 1 (1420 frames long), two glowing balls (green and red, 6 inches in diameter) patched with reflective material were tracked with interfering visual patterns of similar color projected onto two planes. Sample frames from all six cameras (frames 85, 398, 426, 429, and 723) are shown in Figure 6 with reprojected 2D locations (small bold circles) and search regions (large slim circles) superimposed for both objects. The bold green and red circles indicate the green and red objects' 2D locations reprojected from the final 3D locations, respectively. The large slim yellow circle is the predicted search region used to remove 2D false candidates. In these frames, a great amount of 2D false candidates did appear in the entire image but they were effectively removed through the predicted search region. The plane projections were successfully eliminated by the homography test. Some bright spots, for example, the spots on the upper body of the subjects in frames 398 and 723 in all three IR cameras, had no much negative effects on tracking due to 3D outlier rejection and mean-shift-based multiview fusion steps introduced earlier. Objects can be continually tracked when they were put on the ground plane without being confused with floor projections. Object identities were successfully maintained in some challenging scenarios, for example, when two objects were rolling on the ground plane and passing by each other in frames 426 and 429. In this case, it can be seen from Figure 6 that the centers of the reprojected 2D locations were exactly on the objects being tracked. We also projected final 3D location onto all camera views and calculated the average reprojection error over all cameras for each frame. In this experiment, the reprojection errors over all the frames are 0.823 pixels with standard deviation 0.639 pixels for the green object, and 0.758 pixels with standard deviation 0.607 pixels for the red object.

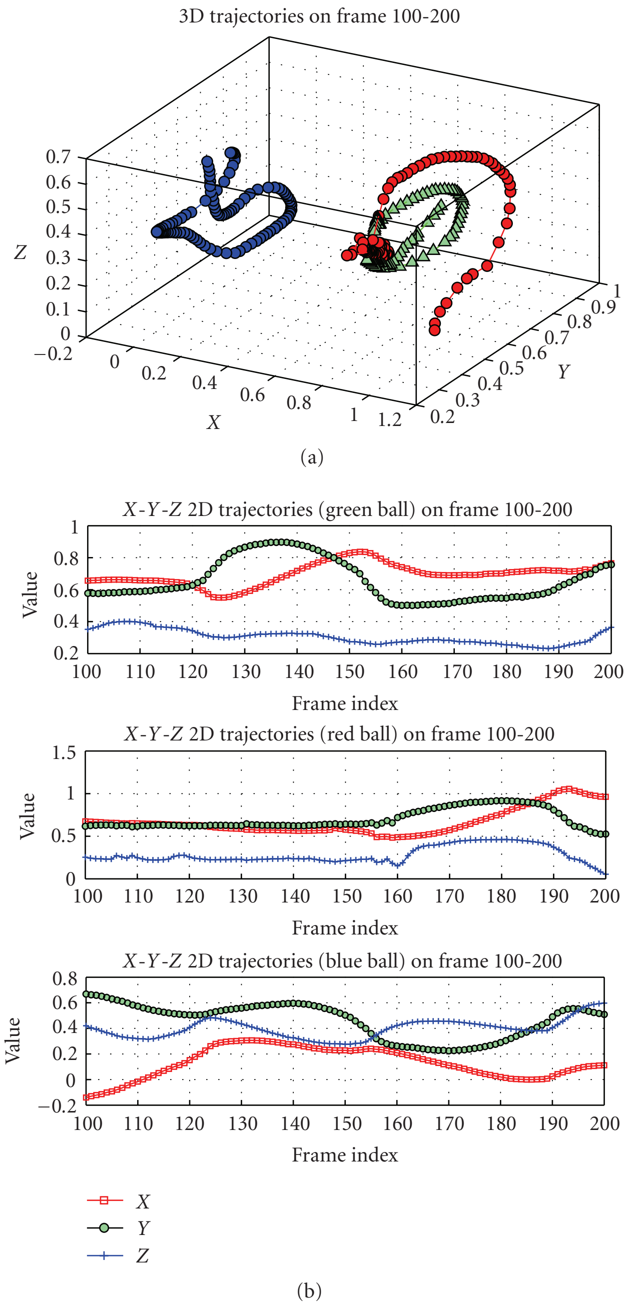
The second sequence (see Figure 7) is 658 frames long, consisting of three moving objects with interfering visual patterns projected on the two projection planes. A person switches green and red objects between his hands around frames 103, 128, 200, and 298. Partial and complete occlusions occurred in these views but the objects were tracked accurately without lose or exchanging of identities. In frame 482, three objects were placed on the ground plane at the same time, and objects were covered by bowing



(a)

(b)

Figure 9: Trajectories of the two objects in sequence 1 between frame 400 and frame 500. The two objects were rolling on the ground plane and passing by each other.



(a)

(b)

Figure 10: Trajectories of the three objects in sequence 2 between frame 100 and frame 200. The green and red objects were manipulated by one subject, and the blue object was handled by another person.

persons in some views. In such severe occlusions with similar color visual projection, all three objects were still tracked accurately. The reprojection errors over all frames in sequence 2 are 0.788 pixels with standard deviation 0.609 pixels for the green ball, 0.891 pixels with standard deviation 0.681 pixels for the red object, and 0.884 pixels with standard deviation 0.523 pixels for the blue object.

In Figure 8, there are 1180 frames. In this experiment, two balls were successfully tracked in the presence of four people moving in the space and interacting with the mediated environment. A large amount of moving visual patterns with varying illumination was projected on both planes. Occlusions of the objects occurred frequently during the tracking. In frame 801, it can be seen that two balls were put close to each other. However, our system can still maintain precise tracking in this challenging case. The objects were successfully tracked in the whole sequence. The average 2D reprojection errors for the green and red ball are 0.732 and 0.755 pixels with standard deviation 0.511 and 0.549 pixels, respectively.

Table 3: Jerk cost of trajectory.

Sequence	Jerk Cost		
	Object 1	Object 2	Object 3
1	0.0355	0.1813	N/A
2	0.0149	0.1266	0.0098
3	0.0184	0.0126	N/A

Our method is able to recover and resume the correct tracking of the same object when it is lost as described in Section 3.8. The false object detection rate after tracking failure in the tested sequences is less than 1% in the recovery phase. A false object detection may happen when the changing background has similar color to the lost object, and there exists an IR-color pair which satisfies both epipolar constraint and homography test coincidentally. The probability of the coincident scenario is very low in practice.

Since ground truth data of the object 3D positions is not available to benchmark our tracking system, the jerk cost was computed to gauge the tracking accuracy and consistency. Reference [24] notes that the smoothness of a trajectory can be quantified as a function of jerk, which is the time derivative of acceleration. Hence, jerk is the third-time derivative of location. For a system $\mathbf{x}(t)$, the jerk of that system is

$$J(\mathbf{x}(t)) = \frac{d^3\mathbf{x}(t)}{dt^3}. \tag{9}$$

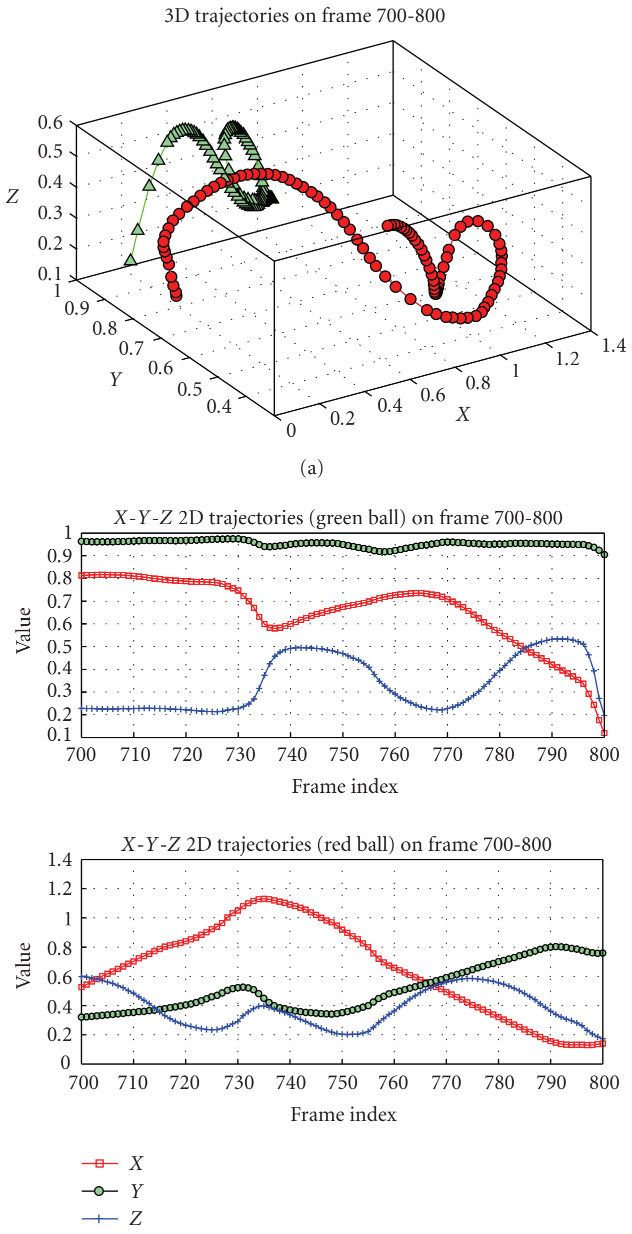
For a particular 3D trajectory $\mathbf{x}(t)$ that starts at t_1 and ends at t_2, the smoothness can be measured by calculating a jerk cost:

$$\text{JC}(\mathbf{x}) = \int_{t_1}^{t_2} \left(\frac{d^3\mathbf{x}_x(t)}{dt^3}\right)^2 + \left(\frac{d^3\mathbf{x}_y(t)}{dt^3}\right)^2 + \left(\frac{d^3\mathbf{x}_z(t)}{dt^3}\right)^2 dt. \tag{10}$$

Three subsequences of 101 frames were selected, one from each video sequence. Table 3 lists the jerk cost of tracked object trajectories in these subsequences. Small jerk cost indicates smooth motion. The 3D trajectories and X, Y, and Z positions over time of objects are plotted. Figure 9 shows these trajectories between frames 400 and 500 of sequence 1. In the first half of this subsequence, the two objects were rolling on the ground plane and passing by each other very closely. Figure 10 shows trajectories between frames 100 to 200 from sequence 2, in the beginning of which the green and red objects were very close to each other. Trajectories of two objects from frame 700 to 800 of sequence 3 are presented in Figure 11.

5. CONCLUSIONS AND FUTURE WORK

This paper reports a real-time system for multiview 3D object tracking for interactive mediated environments with dynamic background projection. The experiment results show that the reported system can robustly provide accurate 3D object tracking in the presence of dynamic visual projection on multiple planes. This system has found important applications in embodied and mediate learning (http://ame2.asu.edu/projects/ameed/). It can also be easily used in many other visually mediated environment to provide reliable object tracking for embodied human computer interaction. We are currently working on integrating built-in inertial sensors such as accelerometers and rotation rate sensors with vision-based tracking. In so doing, the orientation information of the object can also be recovered. In addition, inertial information can improve and maintain



(a)

(b)

Figure 11: Trajectories of the two objects in sequence 2 from frame 700 to 800.

consistent tracking when the object is completely visually occluded in all camera views.

ACKNOWLEDGMENT

This paper is based upon work partly supported by U.S. National Science Foundation on CISE-RI no. 0403428 and IGERT no. 0504647. Any opinions, findings and conclusions or recommendations expressed in thismaterial are those of the author(s) and do not necessarily reflect the views of the U.S. National Science Foundation (NSF).

REFERENCES

[1] Cave Automatic Virtual Environment, http://www.en.wikipedia.org/wiki/Cave_Automatic_Virtual_Environment.

[2] T. B. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.

[3] L. Wang, W. Hu, and T. Tan, "Recent development in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[4] Intersense, http://www.isense.com/.

[5] Ascension, http://www.ascension-tech.com/.

[6] Y. Huang and I. Essa, "Tracking multiple objects through occlusions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 2, p. 1182, San Diego, Calif, USA, June 2005.

[7] J. Pan and B. Hu, "Robust occlusion handling in object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, Minneapolis, Minn, USA, June 2007.

[8] T. Yang, S. Z. Li, Q. Pan, and J. Li, "Real-time multiple objects tracking with occlusion handling in dynamic scenes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 970–975, San Diego, Calif, USA, June 2005.

[9] W. Qu, D. Schonfeld, and M. Mohamed, "Distributed Bayesian multiple-target tracking in crowded environments using multiple collaborative cameras," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 38373, 15 pages, 2007.

[10] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," in *Proceedings of the Workshop on Motion and Video Computing (Motion '02)*, pp. 169–174, Orlando, Fla, USA, December 2002.

[11] F. Jurie and M. Dhome, "Real time tracking of 3D objects with occultations," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '01)*, vol. 1, pp. 413–416, Thessaloniki, Greece, October 2001.

[12] S. M. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1355–1360, 2003.

[13] S. M. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *Proceedings of the 9th European Conference on Computer Vision (ECCV '06)*, vol. 4, pp. 133–146, Graz, Austria, May 2006.

[14] Smallab, http://ame2.asu.edu/projects/ameed/.

[15] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multi-camera self-calibration for virtual environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 14, no. 4, pp. 407–422, 2005.

[16] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2nd edition, 2004.

[17] Y. Cheng, "Mean shift, mode seeking and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[18] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, vol. 2, pp. 142–149, Hilton Head Island, SC, USA, June 2000.

[19] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with application in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[20] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 438–445, Vancouver, Canada, July 2001.

[21] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[22] D. Comaniciu, "An algorithm for data-driven bandwidth selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 281–288, 2003.

[23] Camera calibration toolbox for matlab, http://www.vision.caltech.edu/bouguetj/calib_doc/.

[24] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.