MODEL-BASED 3D TRACKING IN MULTIPLE CAMERA VIEWS

Itai Katz
National Institute of Standards and Technology, Gaithersburg, MD
itai.katz@nist.gov

Kamel Saidi
National Institute of Standards and Technology, Gaithersburg, MD
kamel.saidi@nist.gov

Alan Lytle
National Institute of Standards and Technology, Gaithersburg, MD
alan.lytle@nist.gov

Abstract

We present a method for tracking objects in 3D by comparing views from multiple video cameras against a 3D model. Camera-based object tracking is a vast field that until recently has been limited to the 2D (image) domain. With the falling cost of sensors in recent years, the research community has been shifting its focus to 3D tracking using networks of calibrated cameras. The standard approach has been to run traditional 2D tracking in each video and fuse the results. The algorithm described below tracks objects directly in 3D to give increased precision and to overcome limitations inherent in the standard approach. We demonstrate the effectiveness in tracking construction personnel in video sequences of a construction site mock-up, resulting in a reduction in tracking error over current methods.

KEYWORDS: camera network, tracking, multiple cameras, 3D

INTRODUCTION

The ability to automatically track personnel on a construction site could have significant implications for project safety and productivity. Consider the following use cases:

- Workers operating under crane hooks or behind heavy machinery are exposed to considerable risk. A robust tracking capable of localizing workers and equipment could provide real-time incident detection and mitigation.

- On large scale sites, personnel transportation time between work stations takes significant time. By analyzing the distribution of workers over time, project managers could optimize site layout to reduce this time.

Several radio-based technologies have been suggested to address this topic, including radio frequency identification (RFID) and ultra-wideband (UWB), but these have the disadvantage of requiring personnel to carry tracking tokens. Thus active tracking requires a higher degree of compliance from each worker and may raise the barrier to entry. Additionally, large

numbers of transient sub-contractors make the distribution and management of tokens impractical.

With the growing ubiquity of cameras, video has also been explored as a possible sensor. Localizing pedestrians in camera images is not novel. While human tracking has long been a mainstay of the computer vision literature, the construction domain provides unique challenges. The high degree of visual clutter that is typical of construction sites, as well as the constantly evolving environment, lead to a deployment-ready tracking system whose requirements are substantially different than those of a laboratory prototype.

In this paper we present a novel method for 3D tracking using multiple, calibrated camera views. By comparing the hypothesized position of a model and reprojecting into each camera, tracking can occur directly in the decision space. This represents a departure from the traditional 2D approach, where a model is identified in each camera's feature space, and the resulting positions are fused to produce a 3D result. Using a simulated construction site, we show the effectiveness of our proposed method and compare it to the existing 2D one.

APPROACH

We first describe the baseline method, followed by the proposed method.

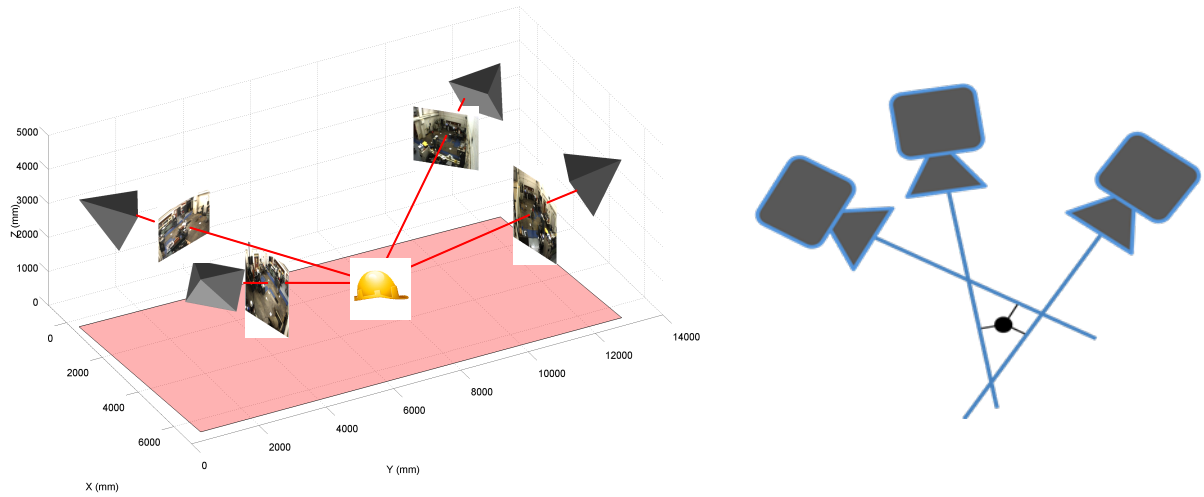
Baseline method

As is typical in the computer vision literature, a lowercase **x** represents a two-dimensional homogenous point on the image plane, while an uppercase **X** represents a three-dimensional homogenous point in space. The two points are related through

$$w\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X} \tag{1}$$

where w is an unknown scaling factor and $\boldsymbol{P}$, the 3 x 4 projection matrix, describes the pose and geometric parameters of the camera and lens system. This matrix can be computed through standard calibration techniques such as (Zhang, 1998).

The standard approach to track objects in 3D, which we will define as the *baseline* method, is a two-step process. In the first step, traditional 2D tracking algorithms are applied to localize the object in each image plane. Common techniques include blob tracking with optical flow, Kernel-based methods, such as the well known Mean Shift algorithm (Comaniciu, 2002), and contour-based methods (Kass 1988). The second step takes the resulting positions $\boldsymbol{x}^{\boldsymbol{i}}$ (nominally the projected object's centroid) for each camera i and combines them using a least means squared method to recover **X**. In other words, **X** is the 3D position which minimizes the sum of squared distances (in Euclidean space) between itself and the associated projected rays from each image plane. This process is illustrated in Figure 1. For greater mathematical detail the interested reader is directed to (Liu, 2002).



(a) (b)

Figure 1: Schematic representation of the baseline method. (a) The object is tracked in each camera view and the 3D position is reconstructed using knowledge of the camera positions. (b) To reconstruct the 3D point in space, an optimization function finds the point that minimizes the sum of perpendicular distances (residuals).

From the nature of the optimization it is clear that a tracking failure in any camera would result in a large error after reconstruction. As the number of cameras increase (e.g., to expand the work volume), the likelihood of these errors is expected to increase. Specifically, the baseline method is subject to:

1) Occlusion. When the object is obscured by the environment or exits the frame edge, the target will be lost. Care must be taken to identify "bad" cameras and exclude them from the reconstruction until the target is reacquired. On a construction site occlusion is of particular concern.
2) Scale selection. A classic issue in 2D tracking is determining scale, which is a function of the object's distance to the camera. With independent tracking, it is possible that scale will be inconsistent across cameras.
3) Rotation-dependent error. If the target is not a perfect sphere, then as the target rotates the apparent centroid in the image projection will drift relative to the true centroid.
4) Temporal synchronization. The baseline method assumes all images are captured simultaneously. Slight deviations in triggering are not corrected and could result in significant error: If the target is moving at 2 m/s (typical walking speed), a temporal uncertainty of 50 msec results in a 10 cm tracking error.

Proposed method

An alternative solution offers an approach that is conceptually simple: given a point cloud model of the object, M_{target}, and an initial 3D position, a synthetic target at a hypothetical position is projected into each camera view, and compared with the observed data. The position of the synthetic target undergoes iterative translation until the projection best matches each camera view. The proposed approach is depicted below in Figure 2.

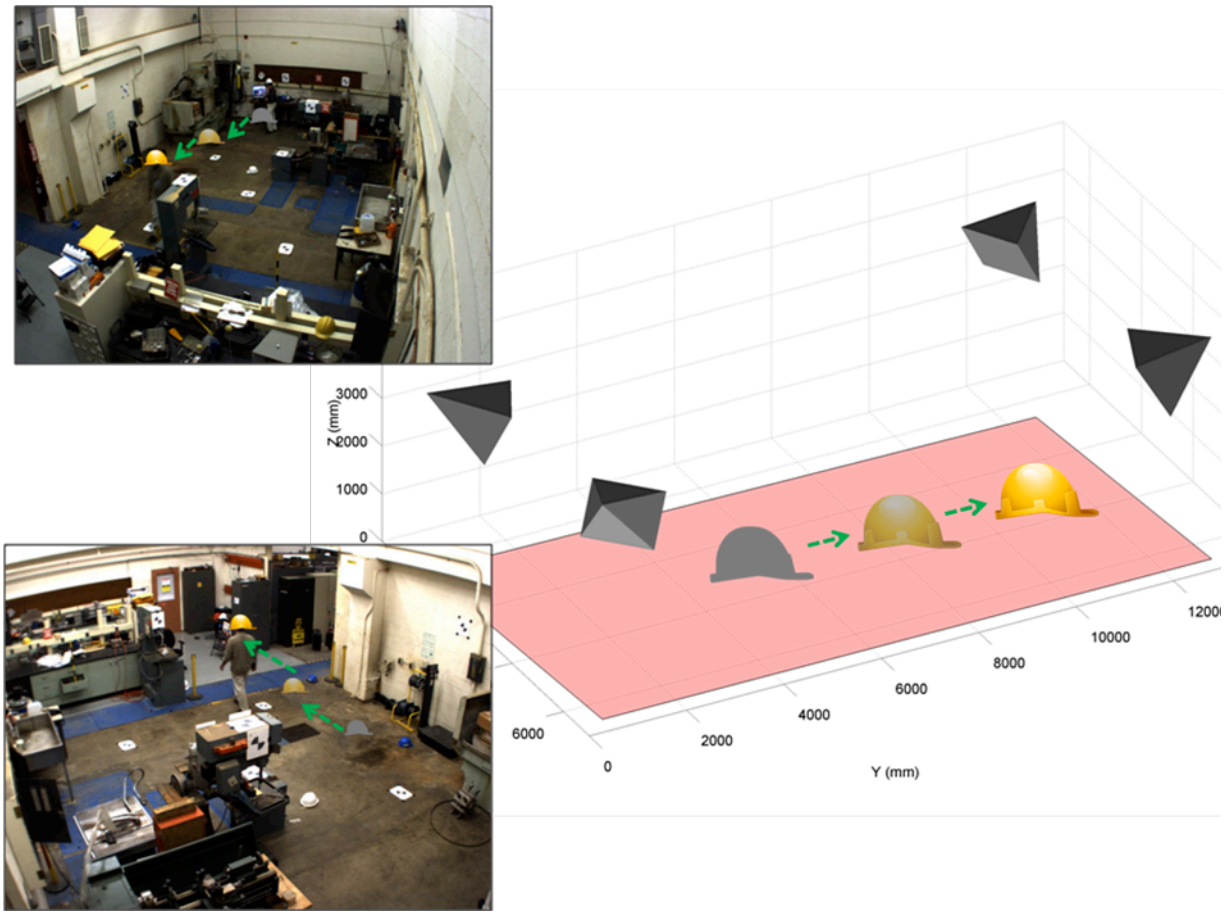


Figure 2: In the proposed method, a hypothetical position is iteratively updated until its projection matches the camera observations.

This can be represented analytically as

$$\boldsymbol{X} = \min_{\tilde{\boldsymbol{X}}} \sum_{i} \rho\left(\boldsymbol{P}^{i}\tilde{\boldsymbol{X}}, \boldsymbol{x}^{i}\right) \tag{2}$$

Where ρ is an error function. If ρ is the commonly used L^2-norm, equation (2) becomes

$$\boldsymbol{X} = \min_{\tilde{\boldsymbol{X}}} \left\|\boldsymbol{f}(\boldsymbol{P}^{1}\tilde{\boldsymbol{X}}) - \boldsymbol{f}(\boldsymbol{x}^{1})\right\| + \cdots + \left\|\boldsymbol{f}(\boldsymbol{P}^{N}\tilde{\boldsymbol{X}}) - \boldsymbol{f}(\boldsymbol{x}^{N})\right\| \tag{3}$$

The function $\boldsymbol{f}(\cdot)$ is a descriptor that describes the local neighbourhood at a point $\boldsymbol{x}$ using some low-level feature. This allows two image regions to be directly compared. Empirical testing found that a HSV histogram works well.

The proposed approach has several benefits over the baseline:

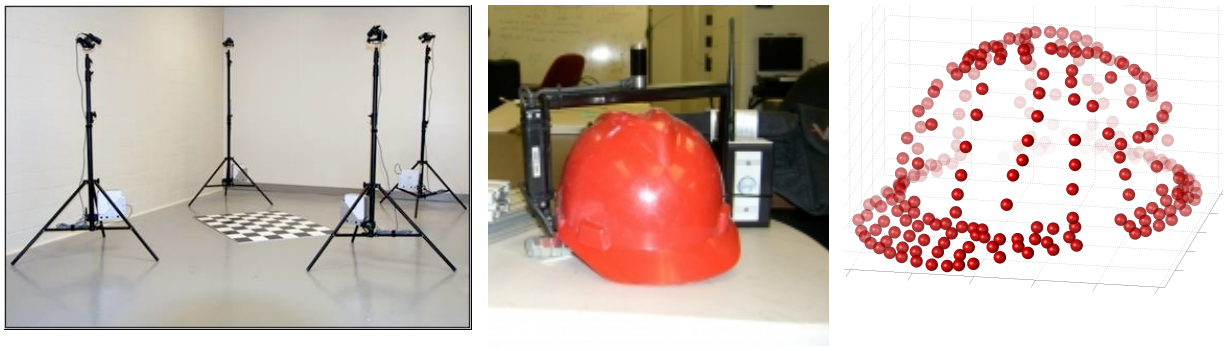
1) Scale selection. The scale of the object is only a function of the model's physical size. Since the apparent scale in the image is implicitly handled by the projection matrix, no scale parameter needs to be manually set.
2) Location consistency. Tracking directly in 3D avoids the consistency problem that occurs when 2D tracking returns erroneous results in one or more cameras.
3) Parameters. This method has no environment-specific parameters.

As an extension, we consider the case where cameras are not triggered (i.e., frames are not captured simultaneously across cameras). In deployment, this scenario is common as providing hardware triggering is costly. In our experiments, two cameras exhibited as much as a 50 msec discrepancy between corresponding frames. This time difference, which we define to be the latency, can affect performance substantially. The solution is to combine the position estimate **X** with a velocity estimate $\dot{\boldsymbol{X}}$. When projecting the hypothesis position into the camera images in the optimization step, the location in the image is adjusted to account for velocity and latency (the time between corresponding frames). For camera $\boldsymbol{i}$ and latency $\boldsymbol{\delta}$ Equation (1) becomes

$$w\boldsymbol{x}^{\boldsymbol{i}} = \boldsymbol{P}^{\boldsymbol{i}}\boldsymbol{X} + \boldsymbol{\delta}^{\boldsymbol{i}}\dot{\boldsymbol{X}} \tag{4}$$

METHODOLOGY

At a NIST facility, we installed a network of four cameras around a 100 m^2 work space. A simulated construction site was set up and an actor was recorded moving throughout the work space in six separate trials. The object being tracked was a hardhat, with ground truth provided by an indoor GPS (iGPS) system[1]. This device is theoretically capable mm-level precision and up to 40 Hz update rate, and was chosen as it was expected to provide precision of 1-2 orders of magnitude above the camera network being tested.



(a) (b) (c)

Figure 3: (a) Camera network testbed prior to configuration (b) Close-up of tracking object. The red color of the hardhat provides a simple visual target for the cameras. Note iGPS receiver (black cylinder) mounted above the hardhat. (c) 3D point cloud model of hardhat used in the proposed method

The cameras are capable of 1024 x 768 pixel resolution at a frame rate of 7.5 fps. These research-grade cameras were chosen for their particularly low noise in the low light levels present in the experimental setting. To ensure consistent data between all five devices (four

[1] Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

cameras and the iGPS tracker), 15 calibration targets were mounted in the work volume to provide a common reference frame. To provide temporal synchronization, all devices were connected to a local NTP server. Although all the devices were *synchronized* they were not *triggered* (i.e., although all the clocks were consistent, the cameras and iGPS did not capture data at the same instant). Five video sequences were recorded, demonstrating motion along the Y-axis (North-South), X-axis (East-West), Z-axis (with a step ladder), and a freeform motion.

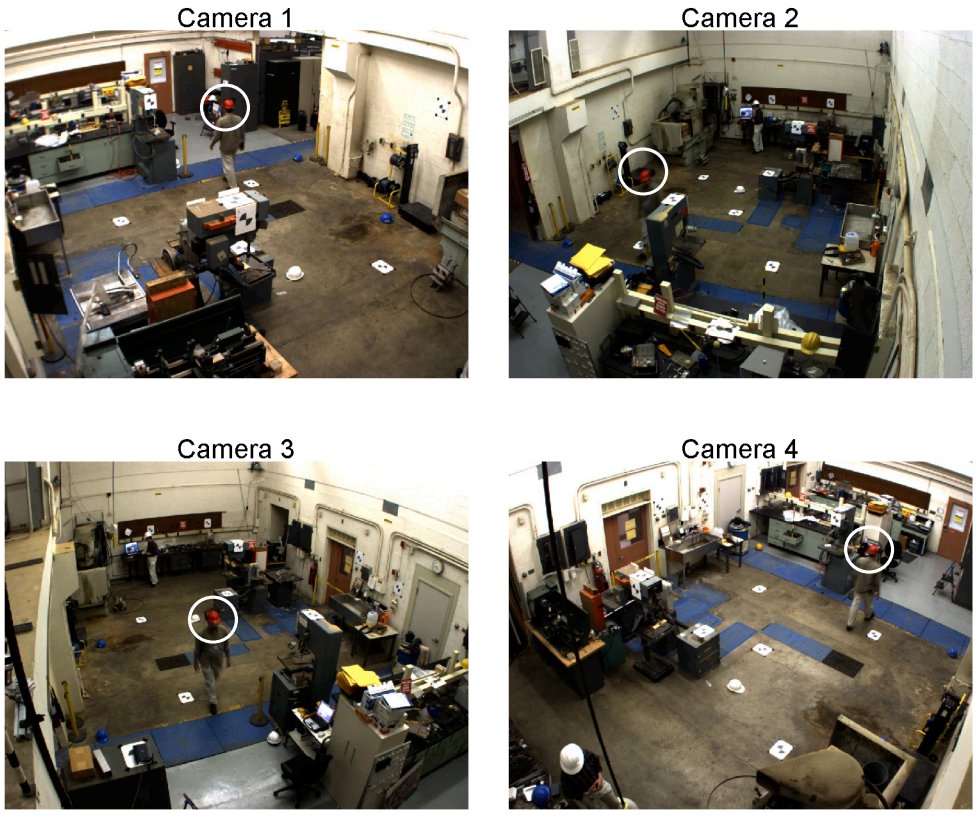
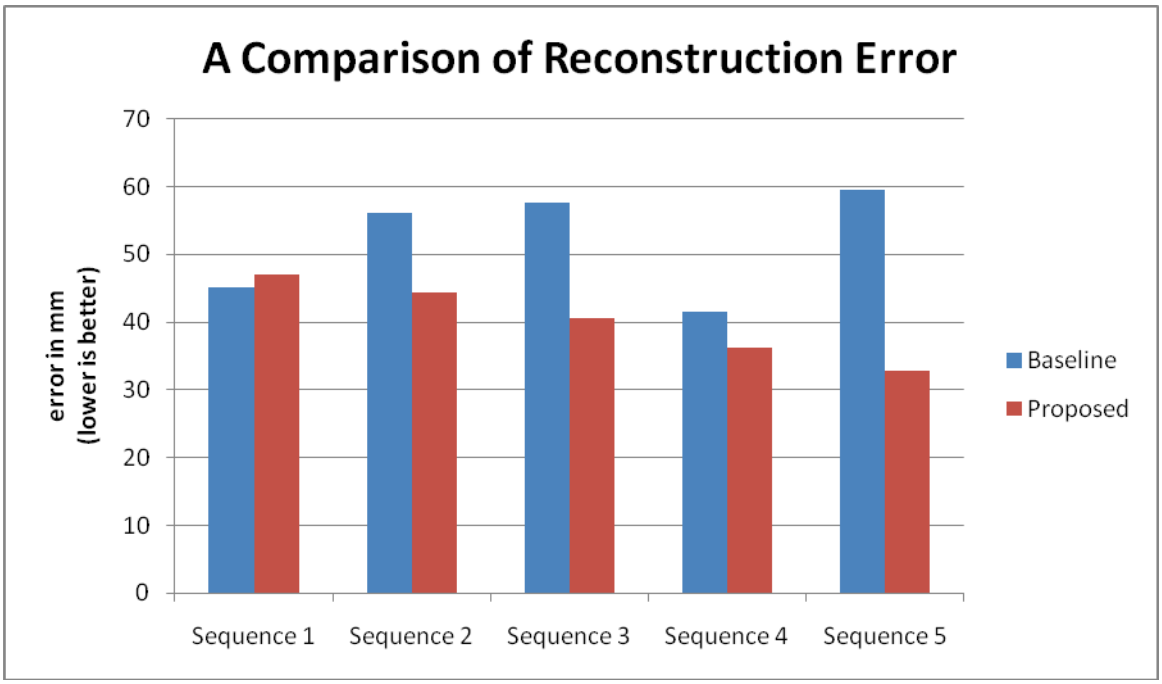


Figure 4: Typical camera views during a sequence. Note the hardhat being tracked (white circle).

RESULTS

The experimental results are presented in Figure 5. Across all sequences, the average 3D reconstruction error for the baseline method is 52.2 mm compared with the average error for the proposed method of 39.4 mm. This represents a 24% reduction.



CONCLUSIONS

This report presented an approach to 3D model-based object tracking in multiple cameras. This method can overcome some of the shortcomings of the standard 2D approach by demonstrating robustness to occlusion, implicit scale selection, and decreased tracking error. The experimental results showed that performance is significantly increased over the baseline method.

Future work will seek to improve the ability to detect object rotation. Good performance was demonstrated for translation, but identifying rotation accurately and consistently remains challenging. Further analysis would also show how tracking performance is affected by frame rates and the number of cameras in the system.

REFERENCES

Comaniciu, D., and Meer, P. (2002) Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5), 603-619.

Kass, M., Witkin, A., and Terzopoulos, D. (1988) Snakes: Active Contour Models, International Journal of Computer Vision, 1(4), 321-331

Liu, B., Yu, M., Maier, D., and Manner, R. (2002) An Efficient and Accurate Method for 3D-Point Reconstruction from Multiple Views, International Journal of Computer Vision, vol. 65, 175-188

Zhang, Z., (1998) A Flexible New Technique for Camera Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, 1330-1334.