Camera Tracking for Augmented Reality Media

Bolan Jiang, Suya You, and Ulrich Neumann

Integrated Media Systems Center
University of Southern California
Los Angeles, CA 90089-0781
{bjiang|suyay|uneumann}@usc.edu

Abstract

This paper presents a camera tracking system for the spatial stabilization of Augmented Reality (AR) media. Our approach integrates both artificial landmarks (fiducials) and natural features for camera tracking. Artificial landmarks are used for system initialization and computation of initial camera pose. Robust and extendible tracking is achieved by dynamically calibrating the 3D positions of a priori uncalibrated natural features. Analysis and experimental results demonstrate the effectiveness of this approach for presenting stabilized AR media in long camera-motion video sequences.

1. Introduction

The estimation of image motion and camera pose plays a key role in advanced multimedia and human-computer interaction systems. Accurate and robust camera pose estimates are a prerequisite for a variety of applications including dynamic scene analysis and interpretation; 3D scene structure extraction and modeling; and image stabilization and video data compression. In our work, we extend motion estimation techniques to the video-based augmented-reality tracking problem. By extracting 2D image motion information directly from a live video stream, we estimate or constrain the real camera 3D motion parameters to maintain dynamic alignment between a real and virtual camera. Our goal is to reduce the dependence of camera tracking on artificial landmarks, as many existing systems do, and to extend the application space of augmented reality systems, even to natural outdoor environments.

An Augmented Reality (AR) system projects virtual 3D media into the user's view of the real world in real time. Ideally, it appears to the user as if the virtual 3D objects (text, models, or images) actually exist in the real environment. One of the key requirements for accomplishing this illusion is a tracking system that accurately measures the position and the orientation of the observer's location in space. Without accurate tracking and registration, the virtual objects will not appear in the correct location at the correct time, diminishing the illusion that they coexist with the real objects.

Many tracking systems utilizing vision techniques have been developed to achieve accurate registration [1]. However, such systems only work in prepared environments where the system designer has sufficient control over the environment to place calibrated fiducials in the regions of interest. A camera pose estimate is obtained by tracking the pre-designed artificial landmarks (fiducials) whose 3D coordinates are pre-calibrated prior to system operation. This approach work well if the landmarks can be calibrated, detected, and identified reliably [1-4]. However, calibrating fiducials in a large environment is difficult, and even when accomplished, the operating range of the tracking system is limited to areas where some minimum number of fiducials are in view. Even partial occlusion of fiducials can cause failures or errors in tracking. If robust tracking can be achieved with less calibration and over wider areas, this would be a major step in reducing the hurdles to setting up and operating an AR media display system.

In this paper, we present a robust tracking system that achieves a significant step towards reducing the need for wide area fiducial calibration and extending the tracking range into regions beyond where fiducials are visible. Our approach integrates both fiducials and natural features to obtain very robust and accurate camera tracking. Fiducials are used for system initialization and computation of initial camera pose. Robust and extendible tracking is achieved by dynamically calibrating the 3D positions of a priori uncalibrated natural features. This work is the extension of our pervious work [3,4] in terms of describing increases in accuracy and addressing the correspondence problem. Additionally, we introduce our capability to create approximate 3D models of the operating scene.

2. System overview

Four main components are included in our tracking system:

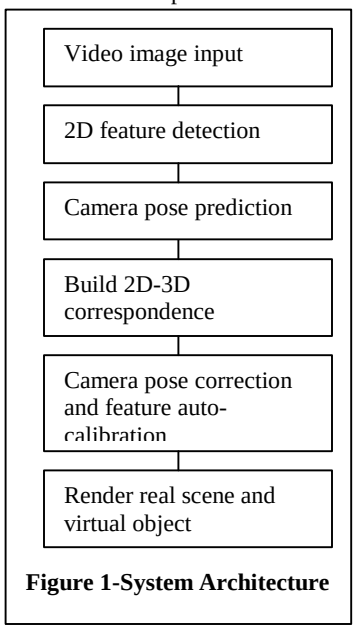


Figure 1-System Architecture

- *Camera pose computing*: estimating camera pose based on the correspondences of tracked features.

- *2D feature tracking*: tracking the inter-frame motion of natural features for pose estimation.

- *2D-3D correspondence*: corresponding 2D image measurements with their calibrated features.

- *Natural feature calibration*: dynamically calibrating the 3D positions of a-priori uncalibrated natural features.

Figure1 depicts the architecture of our system. In each new video image we detect 2D fiducials and natural features. Fiducials are used to initialize the camera pose. In subsequent frames, prior pose estimates are used to predict the current pose that will be progressively refined. More robust and accurate pose estimates are obtained by using the tracked 2D features whose 3D positions are auto-calibrated dynamically

from multiple prior images. The result is refined iteratively until the estimate error converges. The final output of the tracker is an accurate estimate of camera pose that specifies a virtual camera used to project the augmented reality media into the scene. Besides the camera pose, our system also produces the set of of auto-calibrated natural features that are useful for automatic scene modeling. Through autocalibration, the range of tracking is extended beyond the initially-calibrated area into unprepared areas of the enviornment.

3. Camera tracking and pose estimate

3.1 2D feature detection and tracking

Two types of features, artificial landmarks and natural features are used as tracking primitives.

Landmark feature

We adopted a multi-ring color fiducial originally designed by [3]. Since these artificial landmarks are concentric rings of different colors and sizes, their unique appearance simplifies detection and correspondence to their unique calibrated 3D positions.

Natural feature tracking

A robust motion tracking approach is used for natural feature tracking. The novel part of the approach [7] is its integration of three motion analysis functions, feature selection, tracking, and verification, in a closed-loop cooperative manner to cope with complex imaging conditions. Firstly, in the feature selection module, 0D (point) and 2D (region) features are selected for their tracking and motion estimation suitability. This selection and evaluation process also uses data from a tracking evaluation function that measures the confidence of a feature's prior tracking estimates.

Once selected, features are ranked according to their evaluation scores and fed into the tracking module. The tracking method is a differential-based local optical-flow calculation that utilizes normal-motion information in local neighborhoods to perform a least-squares minimization to find the best fit to motion vectors. Unlike traditional single-stage implementations, the approach adopts a multi-stage robust estimation strategy. For every estimated result, a verification and evaluation metric assesses the confidence of the estimation. If the estimation confidence is low, the result is refined iteratively until the estimation error converges.

To achieve robust tracking, two different verification strategies are used for the point and region tracking and motion models. Basically, in both cases, an estimated motion field generates a predicted frame that is used to measure the estimation residual. The difference between the predicted frame and the true target frame measures the error of the estimate. This error information is fed back to the tracking module for motion correction and to the feature detection module for feature re-evaluation. The process acts as a "selection-hypothesis-verification-correction" strategy that make it possible to discriminate between good and poor estimation features, which maximizes the quality of the final motion estimations.

3.2 Build 2D-3D Correspondence

For tracking and computing camera pose related to a world coordinate frame, we need to correspond a number of calibrated 3D features and their 2D projections on the image plane. As stated before, two kinds of features, artificial landmarks and natural features are used as tracking primitives in our system. Since the artificial landmarks have their 3D coordinates pre-calibrated, the tracking system has to reliably detect them and build the correspondence between the detected 2D projections and their tabulated 3D positions. Unlike in [3], where the correspondences are built with a cluster identification method that makes use of the spatial relationship of at least three visible fiducials, we employ a simple but effective prediction approach. For each new video image, the 2D projections of fiducials are detected. At the same time, the 3D fiducial positions are also projected onto the image plane using a predicted camera pose derived from the previous frame. We note the fact that inter-frame motion is often small relative to the larger distances between fiducials. The nearest detected 2D fiducial and projected fiducials are corresponded. Combining this method with simple color matching and cluster identification to provide the initial correspondences produces a very robust system. Even with only one or two observable fiducials, the system nominally determines the correct 2D-3D correspondences.

Natural features are also used for camera tracking. Since we do not know their 3D positions initially, it is critical to track them robustly in the 2D image sequence to facilitate calibration of their 3D positions. Once their 3D positions are estimated, 2D-3D correspondence is established by 2D-2D correspondence of its 2D image measurements between current frame and last frame.

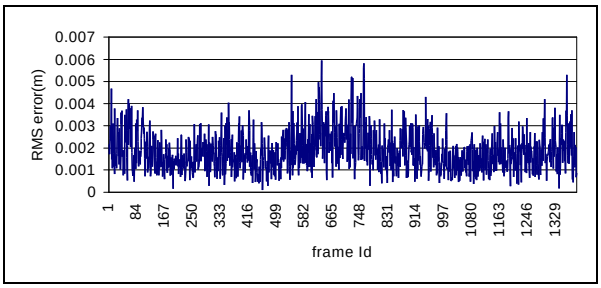
3.3 Camera Tracking and Natural feature auto-calibration

To track camera pose and dynamically estimate the 3D positions of natural features, we developed an auto-calibration approach based on an iterative Extended Kalman Filter (iEKF). Basically, this Kalman Filter [5] consists of two main processes: pose prediction and measurement correction. In the prediction process, the 2D feature motion and "history" information from pervious estimates are combined to predict the current camera pose. This predicted pose is used to establish the 2D-3D correspondences, as described above. With correspondences, the new image feature measurements are used to correct the pose. This measurement correction step is iterative in that we refine the estimate by applying corrections from one feature at a time. By processing the measurement correction in a recursive way we overcome the requirement for a minimum number of available features in order to obtain a pose and we accommodate corrections from features with varying position certainty. In order to calibrate the 3D positions of features we also maintain each feature's state. The feature database holds a position and error covariance matrix that represents the uncertainty of each feature.

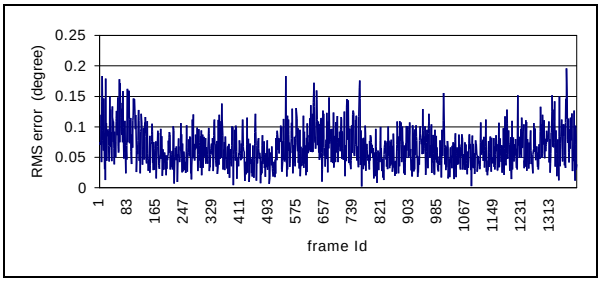
A piecewise-constant white acceleration model is chosen to model target motion. This assumes the target undergoes a constant acceleration during period δt, and the acceleration is uncorrelated from period to period. This model is expressed as

$$\vec{x}(t) = A\vec{x}(t - \delta t) + \Gamma\vec{v}(\delta t) \qquad (1)$$

where A is the state transition matrix implementing $x(t) = x(t - \delta t) + \dot{x}(t - \delta t)\,\delta t$ and $\dot{x}(t) = \dot{x}(t - \delta t)$. Γ is the noise gain, and $\vec{v}$ is the zero-mean white vector whose elements are independent of each other.



(a) RMS error for estimated camera position



(b) RMS error for estimated camera orientation

Figure 2- synthetic data test result

Besides the dynamic model we also need a measurement model to correct the state prediction. For our tracking application we use the projections of given 3D features on the image as the measurement model. Our tracking and feature calibration algorithm is introduced as follows:

1) Calculate elapsed time from the last frame and use the camera state in the last frame to predict the current camera state and error covariance matrix. The camera state includes position, orientation, and their derivatives $[x, y, z, \dot{x}, \dot{y}, \dot{z}, \Delta f, \Delta q, \Delta y \ , \dot{f}, \dot{q}, \dot{y}]$. Similar to SCAAT [2] we maintain incremental orientation in the state vector and the externally accumulated absolute orientation is represented as a quaternion.

2) After computing the 3D-2D correspondence, sort the observable features in increasing order of 3D position uncertainty. For each feature *f*, augment camera state with feature state and augment camera state error covariance matrix with feature error covariance (equation 2). Then use the current camera state to project the 3D feature on to the image and use the difference from the corresponding 2D real image measurement to update the camera state and feature state of f.

$$\hat{x}^- = [\hat{x}^-, x_f, y_f, z_f]$$
$$\hat{P}^- = \begin{bmatrix} P^- & 0 \\ 0 & P_f \end{bmatrix} \quad (2)$$

By iterating the features in increasing order of uncertainty the feature error is small while the camera error is large at the start of the measurement correction iterations. This leads to fast convergence of the estimated camera state. After camera state convergence, the features with relative large errors do not affect camera state significantly. Accurate camera state also leads to rapid feature position convergence. By integrating camera tracking and feature calibration together we can use all of the observable features in the image to obtain smooth and accurate camera pose tracking.

3) For new features without corresponding 3D positions in the database, use the 2D-2D correspondence of feature tracking and the tracked camera pose to estimate their 3D positions and add them to the database. The intersection of two rays connecting the estimated camera pose and the 2D image measurement of the features in two frames create an initial estimate of the 3D position for the feature [6].

Camera pose for the first frame is computed by a 3-point method as an initial value for iEKF. More details about the iEKF can be found in [4].

4. Results and evaluation

The proposed approach has been integrated in a real time AR system. This section gives the performance of the tracking on both a simulation test-set and a real scene.

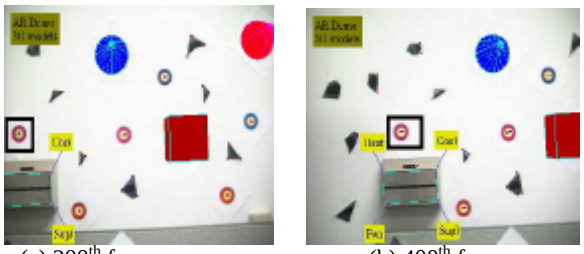
4.1 Simulation data

In the simulation test, we distribute 100 features in a volume of 100x30x20 inches in which six features are precalibrated and 94 are unknown. The camera starts from the area where six calibrated features are visible and then it pans and rotates while the precalibrated features move out of sight. Guassian noise of 0.5-pixel standard deviation was added to image feature measurements. The estimated camera pose was compared with the true camera pose. The RMS error of camera position and orientation are shown in Fig.2 (a) and Fig.2 (b). We also reprojected the tracked features back onto the image plane and calculated their reprojection error. The average reprojection error is about 0.6 pixel with a standard deviation of 0.08 pixel.

4.2 Real scene

In a real experiment, a video sequence was captured and digitized. We started from an area with 15 multi-ring color fiducials on the wall and ended in the area where none of the a priori known fiducials can be seen. We tracked 30 natural features on every frame. The frame rate is 30frames/s and there are 1251 frames in the over 40 second long sequence. Fig 3 shows four snapshots of the video sequence showing some virtual object/media rendering. In (a) camera pose was estimated using the multi-ring color fiducials. In (b) and (c) camera pose was estimated by multi-ring color fiducials and dynamically calibrated features. In (d) camera pose was estimated solely from dynamically calibrated features. The fiducial in the black rectangle was also dynamically calibrated and the accuracy is shown in Fig 4

We also output the calibrated features whose uncertainty is under a threshold. Two different views are shown in Fig 5. From the image we can see that the features on the wall roughly compose a plane when viewed from the side. Additional results including Mpeg files can be found at our website http://deimos.usc.edu/~bjiang



(a) 200th frame (b) 400th frame

(c) 600th frame (d) 900th frame

Figure 3- camera tracking result

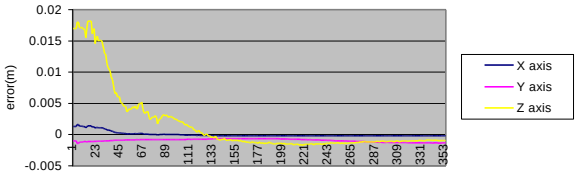
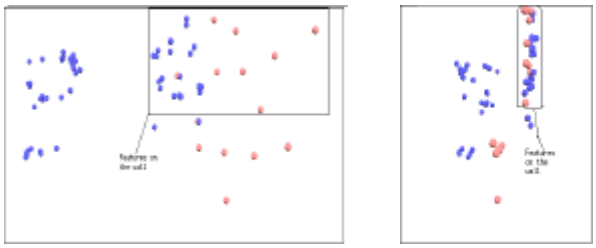


Figure 4 -Calibration Error for feature inside the black rectangle



(a) Front view (b) Side View

Figure5 - visulization for dynamic calibrated features

5. Conclusion

In this paper we present a new camera-tracking method that can extend tracking from a small prepared area to larger unprepared regions of the environment. Camera-tracking and feature auto-calibration are integrated into a single framework that unifies the structure of the whole approach, making it fast, simple, and flexible. In addition, the use of pose prediction leads to an effective method for finding 2D-3D correspondences, thereby eliminating the need for unique fiducial appearances or patterns. Compared with previous work, the propagated error is greatly reduced, which means the system can effective operate for a longer duration and over a larger area.

6. Acknowledgments

This work was largely supported by the Defense Advanced Research Project Agency (DARPA) “Geospatial Registration of Information for Dismounted Soldiers.” We acknowledge the members of AR tracking group of the University of Southern California. We also thank Intel, SGI, and the Integrated Media Systems Center (IMSC) for support.

7. Reference

[1] R. T. Azuma. A survey of augmented reality. Presence: Teleoperators and Virtual Environment, 6(4), pp.355-385, Aug. 1997

[2] G. Welch, G. Bishop, SCAAT: Incremental Tracking with Incomplete Information, Proceedings of Siggraph97, computer graphics, pp. 333-344

[3] YoungKwan Cho, Scalable Fiducial-Tracking Augmented Reality, Ph.D. Dissertation, Computer Science Department, University of Southern California, January 1999

[4] J. Park, B. Jiang, and U. Neumann. "Vision-based Pose Computation: Robust and Accurate Augmented Reality Tracking," IEEE International Workshop on Augmented Reality, Oct. 1999, pp. 3-12

[5] Y. Bar-Shalom X. Li, Estimation and Tracking: Principles, Techniques, and Software, Artech House, Norwood MA, 1998

[6] Emanuele Trucco, Alessandro Veri, Introductory Techniques for 3-D Computer Vision, Prentice Hall, ISBN: 0132611082

[7] Ulrich Neimann and Suya You, "Integration of Region Tracking and Optical Flow for Image Motion Estimation", International Conference on Image Processing (ICIP),1998.

[8] B. Jiang, http://deimos.usc.edu/~bjiang