Wide Area Camera Calibration Using Virtual Calibration Objects

Xing Chen, James Davis, Philipp Slusallek*

Computer Graphics Lab
Stanford University, Stanford, CA 94305
{xcchen, jedavis, slusallek}@graphics.stanford.edu

Abstract

This paper introduces a method to calibrate a wide area system of unsynchronized cameras with respect to a single global coordinate system. The method is simple and does not require the physical construction of a large calibration object. The user need only wave an identifiable point in front of all cameras. The method generates a rough estimate of camera pose by first performing pair-wise structure-from-motion on observed points, and then combining the pair-wise registrations into a single coordinate frame. Using the initial camera pose, the moving point can be tracked in world space. The path of the point defines a "virtual calibration object" which can be used to improve the initial estimates of camera pose. Iterating the above process yields a more precise estimate of both camera pose and the point path. Experimental results show that it performs as well as calibration from a physical target, in cases where all cameras share some common working volume. We then demonstrate its effectiveness in wide area settings by calibrating a system of cameras in a configuration where traditional methods cannot be applied directly.

1 Introduction

Many applications of tracking and observation require operation over a wide area, such as monitoring the traffic flow of vehicles in a parking structure or people in a building. In such cases, a single camera is unlikely to be sufficient. Rather, a network of interconnected cameras is required, each of which functions over only a small subset of the total area. In order to build such a system, a number of issues need to be addressed. The cameras must be calibrated in some global coordinate system; distributed components may need to communicate with each other; and some estimate of the system state which integrates all available sources of information must be computed. In this paper we address the issue of calibrating cameras in a wide area sensing environment.

Wide area system calibration is much more challenging than calibrating a single camera. In single camera calibration, the usual method involves placing a carefully instrumented calibration target in the field of view. Based on correspondences between known 3D features on the target and their 2D locations in the image, calibration can be obtained. If multiple cameras are active in the same working volume, then each can be calibrated individually using an identical process.

The case of wide area calibration introduces a number of difficulties. Cameras each cover only a small subset of the total working volume. A calibration target can be moved so that each camera is calibrated separately. However, a global calibration requires knowledge of relative target motion. This is difficult to obtain without expensive instrumentation. Simultaneous activation of cameras poses an additional problem. In large systems with many possibly heterogeneous cameras it becomes difficult to ensure that all cameras record observations at exactly the same moment. Many algorithms rely on simultaneity as a fundamental constraint.

In this paper we introduce a method that brings a system of unsynchronized cameras into calibration in a single global coordinate system. A rough estimate of each camera's pose (i.e. location and orientation) is obtained using standard structure-from-motion techniques. The rough camera calibration can be used to track the path of a point moving through the entire working volume. This path defines a virtual calibration object, which can be used to improve the estimate of camera pose in the global coordinate space. Iterating the above process results in the convergence to both the point path defining the virtual calibration object and a precise estimate of camera pose. We evaluate our method by comparing it to traditional calibration techniques. Furthermore, we demonstrate its effectiveness in wide area settings by calibrating a multi-camera indoor tracking system where cameras cover

* Current address: University of Saarbruecken, P.O.Box 15 11 50 66041 Saarbruecken, Germany.

disjoint viewing regions, a more challenging situation where traditional methods cannot be easily applied.

The rest of the paper is organized as follows. Section 2 introduces the example application to which our calibration technique is applied. Section 3 discusses previous work. Section 4 describes our proposed method and Section 5 gives experimental results. We conclude in Section 6.

2 Application

The wide area calibration technique presented in this paper can be generalized to a wide variety of applications, sensors, and environments. However, an understanding of the specific application in our lab may prove illustrative. Our experimental system was designed to track people in front of a large screen, multi-projector display [1]. In one application, the head position of the user is acquired to provide the correct viewpoint for images rendered on the display. The tracking system has ten ceiling-mounted cameras oriented to observe a 4.0 x 4.5 meter area. Coverage extends from approximately a half meter to 2 meters from the floor. The wide-angle cameras in the corners cover the volume. In addition, since our application requires higher tracking resolution right in front of the display, a few more narrowly focused cameras are installed to increase the resolution in that area. Individual cameras observe only a portion of the volume. In aggregate, however, they cover the space. Figure 1 shows a photograph of the tracking space, and a plan view of camera placement. Note that in order to ensure correct estimates of observed object position, it is required that at least two cameras observe any given region in space. However, there is no single point that is observed by all cameras.

Each camera is connected to a digitizing board and a CPU. Interlaced fields are captured and processed at 60 Hz. At capture time, fields are time stamped by the local CPU, and all CPUs use a standard network time daemon to ensure consistent time within 3ms. However, the cameras themselves are not synchronized for simultaneous capture. After digitization the local CPU extracts features and sends these over the network to a central estimator that uses an extended Kalman filter [2] to integrate data from all cameras into a single estimate of object position and orientation.

3 Previous work

There has been a great deal of research in the area of accurate camera calibration. Most previous methods use a known calibration pattern that is imaged by the camera. Features are extracted from the image, and the best fit of intrinsic camera parameters and extrinsic camera pose is obtained. Tsai proposed a widely used model, but other more robust models are used as well [3, 4].

Azarbayejani and Pentland propose a method for calibrating the relative position of cameras [5]. An identifiable object is waved in front of a synchronized stereo pair of cameras, and the per-camera image location of the object at each time step is recorded. A standard structure from motion system is used to derive the relative pose of the two cameras. Their focus is not wide area tracking, and synchronized cameras with a common viewing volume are required.

Stein proposes a system of cameras to track vehicles in an outdoor environment [6]. By observing the motion of objects in video sequences from multiple cameras, an

(a) (b)

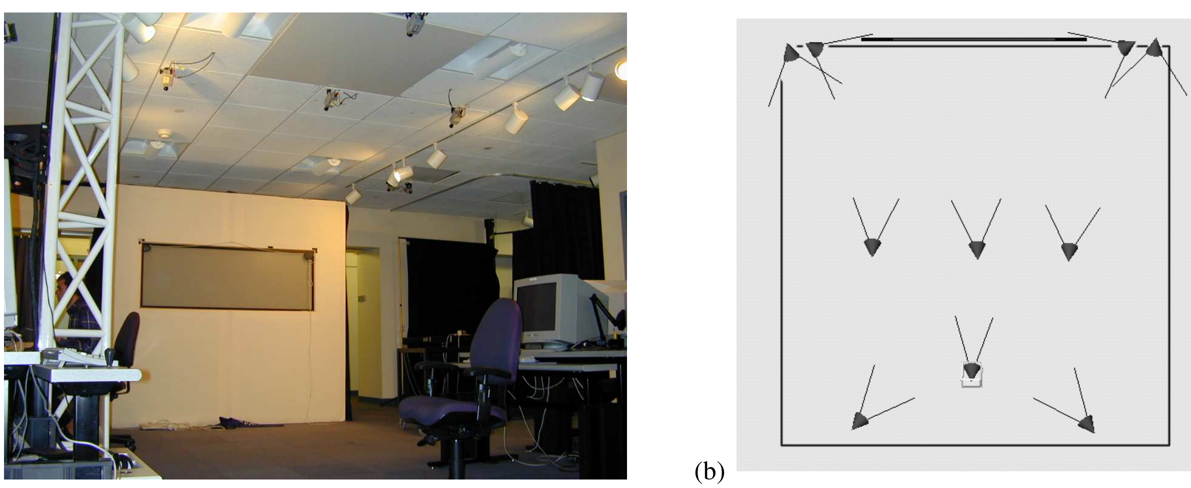


Figure 1: (a) Ceiling mounted cameras are used to track users around a wide area environment in front of the large display. (b) Cameras are arranged so that observation of the entire space is possible, although no single camera observes the entire working volume.

approximate camera pose and time offset can be recovered from several unsynchronized cameras. Image features are used to refine the calibration estimate. This system requires a flat ground plane in all the images and solves the homography relating objects on this 2D plane.

Rander and Kanade have a system of approximately 50 cameras arranged in a dome to observe a room sized space. In order to calibrate these cameras, a large calibration object is built and then moved precisely to several locations, in effect building a virtual calibration object that covers the room [7]. While this works well, it can be quite costly to ensure the precise movement of a calibration object, and it is not easily adaptable when the shape or size of the working volume changes.

Gottschalk and Hughes propose a framework for auto-calibration in wide area spaces [8]. Head mounted sensors observe precisely synchronized beacons mounted on the ceiling of their UNC lab. Data gathered from the sensors can be used to estimate both the moving head location and orientation, and to refine the initially available position estimate of the beacons. Like the system in this paper, they also employ the principle of iterative calibration. Welch later proposed a refined estimation method [9]. However their tracking architecture is quite different from the multi-camera environments we consider.

This paper's contribution is a wide area calibration method that addresses several previously ignored difficulties. A large number of unsynchronized cameras can be calibrated in a single consistent coordinate system. This can be achieved even when some cameras are arranged with non-overlapping working volume and when no initial estimate of camera pose is available. In addition, the method requires no complex instrumentation, and is easily adaptable to working volumes of variable size and shape.

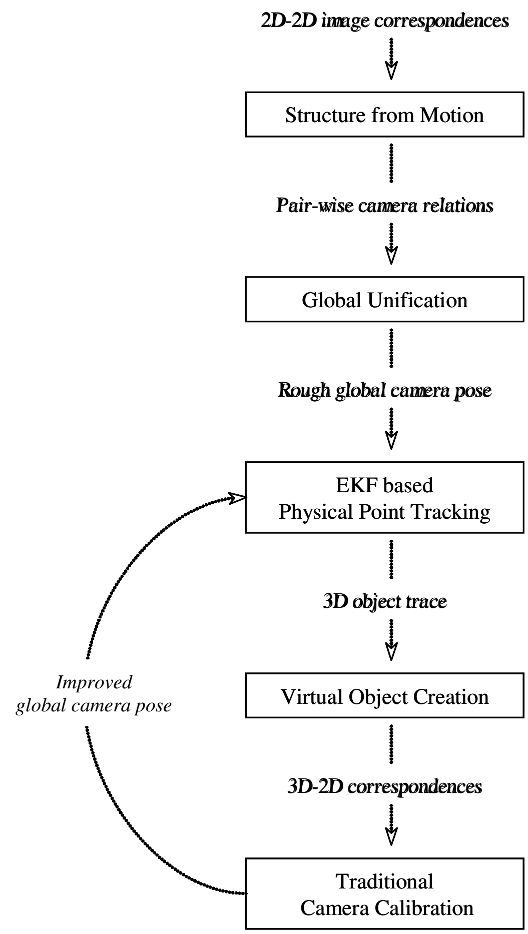


Figure 2: The main stages in our calibration method. Boxes represent computational stages. Italic text shows data flow.

4 Proposed Method

An outline of our method is shown in Figure 2. After the separate calibration of intrinsic camera parameters, our method begins by obtaining 2D image correspondences. The pairwise relative pose between cameras can be found using structure from motion. Then, a unification process brings these pairwise relationships into a single global space. The rough estimate of global pose calculated by the preceding steps can be used to initialize the following iterative procedure. A 3D trace of an object moving through space is estimated using an extended Kalman filter (EKF). This trace can be used as a virtual calibration object by correlating it with camera observations. Using traditional camera calibration, a new set of camera pose estimates is obtained. Iteration produces a globally consistent camera calibration.

4.1 Intrinsic calibration

Camera calibration is typically divided into two parts: intrinsic and extrinsic parameter calibration. The intrinsic parameters usually consist of lens distortion, image center, and focal length. For a short baseline stereo pair the relative pose between cameras in the pair could also be included. Extrinsic parameters define how the local camera coordinate system relates to a global coordinate system, i.e. the six parameters defining position and orientation. We propose that intrinsic calibration is best performed on cameras individually since it is not dependent on the global coordinate system. As mentioned previously, many calibration methods exist that are appropriate for a single camera. We use the intrinsic models proposed by Heikkila [4]. Finding the extrinsic parameters of each camera in a way that is globally consistent is the focus of the remaining portion of this paper.

4.2 Initial extrinsic calibration

Pairwise calibration using structure from motion. To obtain a globally consistent extrinsic calibration of cameras, we start by searching for pairwise registration between nearby cameras in our system. By finding corresponding 2D image points in a pair of camera views we can employ any structure from motion system to recover both the 3D location of these corresponding points, and more importantly for our application, the relative pose of the camera pair. We use a publicly available structure from motion implementation from Zhang [10]. An easily identifiable object is moved so that over time it covers the working volume of our system. We use an LED or flashlight in a darkened room. Since each camera sees only a subset of the working area, not all cameras observe the object at any given location. At this stage, however, only pairwise registration is required. The corresponding 2D observations for all relevant pairs of cameras are recorded.

It should be noted that since the cameras are not synchronized for simultaneous input, no pair of cameras will actually observe the point at exactly the same location. At this stage we make an approximation that will be refined in a later part of our algorithm. Since the object is known to move continuously, we discretize time into small intervals. We use 36ms, since this is approximately the time required for two NTSC video fields to be processed by our 60Hz cameras. Observations occurring during the same time interval are approximated as collocated both temporally and spatially. Given this approximation and the resulting set of pairwise image correspondences, we can employ structure from motion to obtain a set of pairwise camera registrations.

Global unification. The pairwise camera registration that has been obtained provides only the relative rotation and translation up to an unknown scale factor. The desired global calibration will place all cameras in a single global coordinate system.

Starting with an arbitrary pair of cameras, we define a global coordinate system. New cameras can be added incrementally until all available cameras have been included in the global framework. Figure 3 shows an example of a new camera being added to the global framework.

In this example the translation and orientation of cameras A and B are known in the global coordinate system. In addition, we have pairwise relationships giving the translation and orientation of camera C, relative to those of A and B. Note that the translation of C given by its pairwise relationships is a vector with unknown scale.

However, intersection of the rays $\overrightarrow{AC}$ and $\overrightarrow{BC}$ is sufficient to acquire the scale factors α and β. Due to errors, the rays may not intersect. We use the point with the minimum distance to both rays as the approximate intersection point.

Once the scale factor α or β is computed, the global location and orientation of camera C can be derived based on the global location and orientation of either camera A or B. We use the following notation. The translation of camera A with respect to B's coordinate system is denoted by ${}^{B}T_{A}$. The normalized vector from B to A is ${}^{B}\hat{T}_{A}$. Similarly the rotation of A relative to B's coordinate system is ${}^{B}R_{A}$. Using W to notate the world, or global coordinate system, and arbitrarily picking camera A as a base, we have:

$$\begin{aligned} {}^{W}T_{C} &= {}^{W}T_{A} + \alpha\, {}^{W}R_{A} \cdot {}^{A}\hat{T}_{C} \\ {}^{W}R_{C} &= {}^{W}R_{A} \cdot {}^{A}R_{C} \end{aligned}$$

Incrementally applying the above procedure locates all cameras in a single global coordinate system. Naturally this global coordinate system has an orientation based on the initial camera and an arbitrary scale. We take a few real world measurements in order to determine the transform to a physically meaningful coordinate system.

It should be noted that the pairwise registration obtained using structure from motion is not of equal quality in all cases. Camera pairs placed in degenerate positions are likely to cause errors, as are solutions

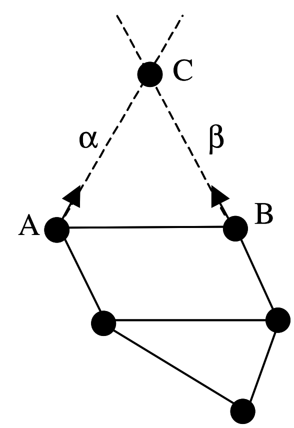


Figure 3: A set of globally calibrated cameras is shown connected by solid lines. The location of camera C can be calculated using the pairwise relationships with A and B, whose global poses are already known.

determined from only a few image point correspondences. Using the residual error returned by structure from motion, we can rank the quality of pairwise information. This ranking can in turn be used to add cameras to the global set in preferential order, thus improving the estimate of global calibration.

Sources of error. Our initial global calibration process contains a number of approximations and sources of error. The discretization of time results in errors bounded by the velocity of the object and the discretization interval. The incremental method in which cameras are added to the global set is also a source of error. Since the global pose of each additional camera is dependent on the previously estimated global pose of other cameras, small errors made at each step can accumulate. Even though the errors made at each stage may be small, cameras added near the end of a large collection are likely to suffer from a great deal of accumulated error. However, the approximate calibration obtained is a good initial guess for the iterative method that is the core of our algorithm.

4.3 Iterative refinement

EKF based physical point tracking. Given a globally calibrated set of cameras, an object can be tracked continuously through the entire working volume. A number of techniques exist for integrating information from diverse sensors into a single estimate of object behavior [2, 11]. We chose to use an extended Kalman filter (EKF) because it is simple and efficient. Our EKF is configured with a constant velocity model and estimates the position and velocity of the tracked point. As an object moves through space, observations from cameras are used to update the EKF estimate. The resulting trace is a continuous estimate of object motion over time. Note that there is no requirement that cameras provide observations at simultaneous moments. The EKF parameters and internal dynamic model provide the temporal constraints normally derived from simultaneous observation. Details of appropriate EKF parameters and models are application dependent, but well known [2, 12, 13].

The initial estimate of camera pose using structure from motion requires that an identifiable object be moved so that the working space is covered. Since the requirements are the same for tracking, the data gathered earlier can be reused. Rather than discretizing observations into time intervals, an EKF processes the observations into a continuous 3D object path.

Virtual object creation. Tracking a 3D object over a wide area provides a method for obtaining a large virtual calibration object. A conventional calibration object has features distributed spatially. Ideally these features are distributed in such a way that they are easily identifiable and cover the working volume. For large working volumes it is impractical to build a precisely measured physical object for the purposes of calibration. The virtual calibration object defined by the estimated 3D object path has features distributed temporally. Each temporal moment relates to a single position in space.

Even with correctly calibrated cameras, the object trace obtained previously will not be perfect. Inadvertent motion into a region observed by a single camera will leave the system under-constrained. Occlusions can cause complete loss of the object, and sudden acceleration will not fit our EKF model. In order to ensure an accurate virtual calibration object, we discard trace points that are seen by only one camera. In addition, we discard trace points for two seconds after time periods in which no camera observes the point. This provides a chance for the system to settle back into more reliable state.

Since almost any location on the path can be used, a large number of calibration features can be constructed. If the path of the physical object through space traverses all portions of the desired working volume, then excellent coverage is obtained as well.

Recalibrating camera extrinsics. The virtual calibration object can be used to individually calibrate each camera with respect to the global coordinate system. A given camera reports a sequence of 2D image observations. The time of observation relates this 2D observation to a corresponding 3D feature point in the virtual calibration object. The resulting set of 2D to 3D correspondences can be used to find the external camera pose. As in calibrating the internal characteristics of our camera, we use a standard method [4].

Iterating the above stages improves the estimate of both camera pose and virtual object location. More accurate virtual objects provide better camera calibration, and better camera calibration allows the virtual object path to be determined more accurately. We have obtained convergence for highly over-constrained environments in five iterations, with more general wide area settings requiring up to forty iterations.

5 Results

Evaluating the effectiveness of a calibration method is not trivial. It is sometimes difficult to obtain ground truth data for the camera in question. In addition, since we have proposed a calibration method for use in wide area environments, general comparisons with existing techniques are impossible.

We first discuss an appropriate metric with which to evaluate our results. Next we consider calibration in a restricted setting, in which comparison with existing techniques is possible. Finally, we show that our proposed

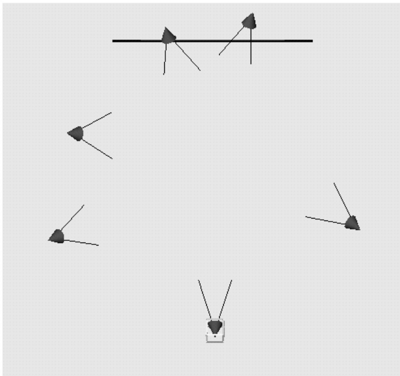


Figure 4: Restricted setting in which all cameras can see a common area. A configuration such as this allows comparison with existing calibration techniques.

method functions as expected on a more general wide area calibration task.

5.1 Evaluation metric

Camera calibration is often defined in terms of projection error. In a traditional calibration task, known 3D locations are projected onto the camera image plane. The distance from the projected point to the observed image location is known as the projection error. Given a set of correspondences, the best camera calibration is the one that minimizes this error.

Another formulation provides only 2D image correspondences between multiple cameras. In this case, in addition to camera parameters, the actual 3D point location is unknown. The best fit of these variables is often defined as a minimization of projection error. If cameras are poorly calibrated in relation to one another, then one expects projection error to be quite high.

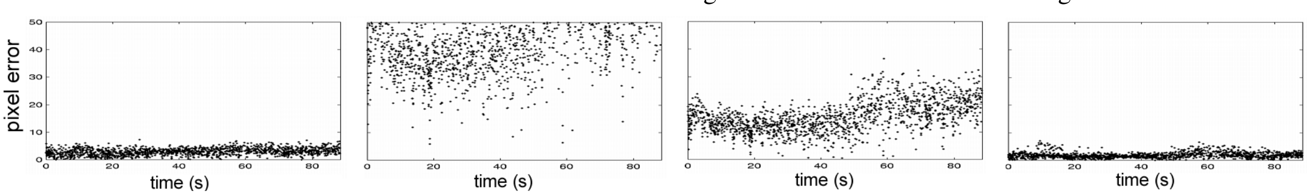
Since the cameras used in our application are not synchronized, we slightly modify this method. An object is tracked using an EKF as previously described. At the time of each camera observation, the EKF provides a predicted object location. The predicted location can be projected onto the camera image plane. The distance between the predicted location and the observed location is recorded as projection error. Note that this error may be caused in part due to an inadequate EKF system dynamic model. In our case we use a constant velocity model; thus, any acceleration applied to the object will appear as projection error. However, this error will be present only during acceleration. Extended, consistent bias in the projection error can be attributed to poor camera calibration.

5.2 Comparison with existing techniques

In order to compare our proposed calibration to existing methods, we have constructed an example in a more restricted setting. By arranging multiple cameras so that all may observe a single region of space, traditional target based calibration can be performed (Figure 4).

We calibrate the camera system using two methods, the one proposed in this paper, and one that uses a physical target. In order to isolate potential noise, the target based calibration uses the same LED feature as was used to build a virtual calibration object. This LED feature is now placed at the end of a Faro digitizing arm [14] that returns the position of the tip of the arm with sub-millimeter accuracy. By arranging for simultaneous triggering of the digitizing arm and camera, we can obtain correspondences between the global 3D location of the feature and the observed 2D image location. While only a few correspondences are theoretically needed, we use approximately 50 to obtain robust external calibration. By repeating this process for each camera a complete set of calibrations in a single coordinate space is obtained.

To evaluate our method fairly, we gathered a new object trace unrelated to any previous traces used during calibration. It is important not to reuse previous traces during evaluation, since we want to ensure against overfit solutions that match the input data set, but do not actually provide a calibrated system of cameras. Figure 5 contains a set of graphs showing projection error in the tracking process for a particular camera. (Data from other cameras in the system is similar.) Using the calibration obtained with a traditional physical target results in trace (a). The average pixel error in a mean squared sense over all cameras is relatively low, only 4.4 pixels. The rough global calibration obtained using structure from motion



(a) (b) (c) (d)

Figure 5: Comparison of projection error of a point trace using (a) traditional target based calibration, (b) structure from motion only, (c) one iteration of virtual object calibration, (d) five iterations of virtual object calibration. Note that a virtual calibration object performs as well as traditional calibration.

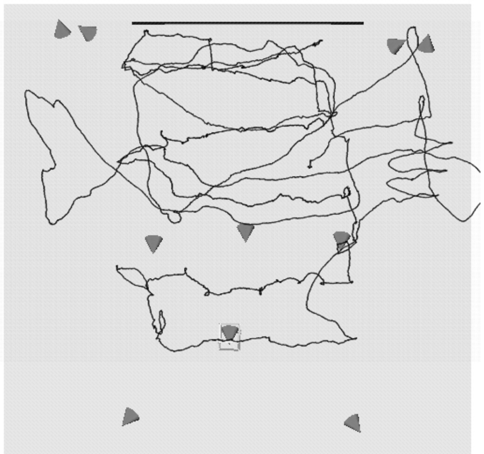


Figure 6: Top view of camera positions after applying structure from motion only. The object path shown defines a virtual calibration object that can be used to improve camera calibration.

results in trace (b). Note that the mean error has greatly increased to 33.8 pixels. After building a virtual calibration object and recalibrating the cameras, we obtain trace (c). Using a virtual calibration object has reduced the mean error to 14.4 pixels. After five iterations of building virtual calibration objects and recalibrating the cameras, trace (d) results. The mean error has been further reduced to 3.9 pixels, approximately equal to the known reliable calibration obtained with a target. Some error remains, but as mentioned earlier this may be due to object acceleration or image feature extraction noise.

5.3 Example in a wide area setting

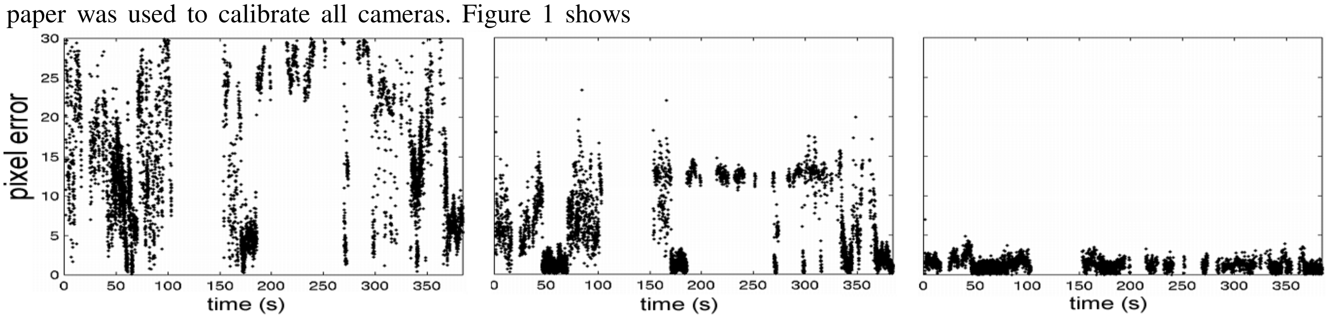
In a wide area setting, we can verify that the iteration process converges in a similar fashion to the more restricted setting described above. The following results consider camera placement designed for tracking people in a room sized environment. The method proposed in this paper was used to calibrate all cameras. Figure 1 shows final camera placement, while Figure 6 shows camera locations after calibration using only structure from motion. Note that these locations are approximately correct. However, refinement is required. For example, the three cameras in the middle are expected to be collinear. Figure 6 also shows a plan view of the temporal path used to build a virtual calibration object. Note that the path traverses the field of view of all cameras, covering the working volume better than a single physical calibration object.

As described previously, projection error is expected to decrease as quality of calibration improves. A new trace unrelated with calibration was captured. Figure 7a shows the projection error of a single camera after globally calibrating all cameras using structure from motion. As before, the mean error across all cameras is quite large, 16.3 pixels. Holes in the graph indicate time periods in which this particular camera did not observe the object, so no error measure is available. (Of course metrics are available for other cameras that do see the point during this time period). Figure 7b shows the improvement after three iterations. Forty iterations of building a virtual calibration object results in a low projection error, 1.3 pixels, shown in Figure 7c. Projection error lower than that obtained in the previous example can likely be attributed to improvements in intrinsic camera calibration. A plot of pixel error vs. iterations can be seen in Figure 8.

6 Conclusions

Calibration of cameras in a wide area environment introduces new challenges to the calibration process. Since individual cameras observe only a small fraction of the whole environment, determination of reliable global relationships is difficult. Typical previous approaches such as building calibration objects that span the observation space do not scale well into wide area environments.

We have introduced a method suitable for calibration



(a) (b) (c)

Figure 7: Projection error in a wide area calibration task. (a) Error after applying structure from motion. (b) Error after three iterations of calibration using a virtual calibration object. (c) Significantly reduced error after forty iterations using a virtual calibration object.

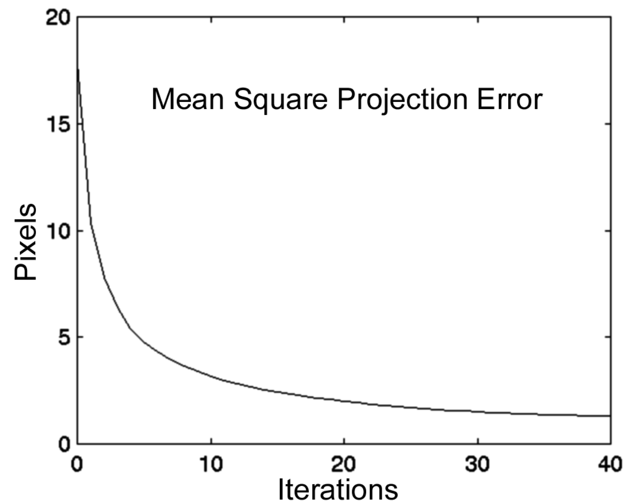


Figure 8: Projection error is greatly reduced after application of the iterative calibration process.

of cameras under these difficult conditions. Intrinsic camera properties are calibrated using existing methods. Next pairwise extrinsic relationships are determined using standard structure from motion techniques. These pairwise relationships allow us to derive an approximate global calibration involving all cameras. The initial estimate of global relationships is not precise. However, it can be used to initialize an iterative calibration technique. Tracking a known physical object as it moves through the environment allows a virtual calibration object to be built. This virtual calibration object can be used as if it were a giant physical calibration object to improve the global camera calibration. Iterating the above process leads to our final camera alignment.

We have presented results that show our calibration method to be competitive with existing techniques in environments where comparison is possible. Further, we show results from an environment where previous calibration methods are unsuitable.

A number of improvements and extensions are candidates for future work. The need for consistent time-keeping on distributed CPUs could be relaxed. In this case the time offset between cameras could be solved as an additional calibration parameter. We would also like to investigate calibration of moving cameras using a similar method, since careful placement of physical targets is impossible in dynamic environments.

Acknowledgements

This work was supported by grants from Intel Corporation, Interval Research, and Sony Corporation. We would like to thank Pat Hanrahan and other members of the Stanford Graphics Lab for helpful comments and discussion.

References

[1] G. Humphreys and P. Hanrahan, "A Distributed Graphics System for Large Tiled Displays," *IEEE Visualization '99*, pp. 215-224, 1999.

[2] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering. Second Edition*, 1992.

[3] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Trans. Robotics and Automation*, vol. 3, pp. 323-344, 1987.

[4] J. Heikkila and O. Silven, "A Four-Step Camera Calibration Procedure with Implicit Image Correction," *IEEE Computer Vision and Pattern Recognition*, pp. 1106-1112, 1997. http://www.ee.oulu.fi/~jth/calibr/Calibration.html.

[5] A. Azarbayejani and A. Pentland, "Real-time Self-calibrating Stereo Person Tracking Using 3-D Shape Estimation from Blob Features," *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, pp. 627-32, 1996.

[6] G. P. Stein, "Tracking from Multiple View Points: Self-calibration of Space and Time," *IEEE Computer Vision and Pattern Recognition*, pp. I:521-527, 1999.

[7] P. Rander, "A Multi-Camera Method for 3D Digitization of Dynamic, Real-World Events," : Robotics Institute, Carnegie Mellon University, 1998.

[8] S. Gottschalk and J. F. Hughes, "Autocalibration for Virtual Environments Tracking Hardware," *Proceedings of SIGGRAPH 93*, pp. 65-72, 1993.

[9] G. Welch, "SCAAT: Incremental Tracking with Incomplete Information," University of North Carolina at Chapel Hill TR 96-051, 1996.

[10] Z. Zhang, "An Automatic and Robust Algorithm for Determining Motion and Structure from Two Perspective Images," *Computer Analysis of Images and Patterns*, pp. 174-81, 1995. http://www.inria.fr/robotvis/personnel/zzhang/.

[11] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *4th European Conference on Computer Vision*, vol. 1, pp. 343-56, 1996.

[12] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*: Academic Press, 1988.

[13] T. J. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive Estimation of 3D-Motion from a Monocular Image Sequence," *IEEE Transactions Aerospace Electronical Systems*, vol. 26, pp. 639-656, 1990.

[14] FaroArm, , http://www.faro.com.