Distributed Metric Calibration of Large Camera Networks

Dhanya Devarajan and Richard J. Radke[*]
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY, 12180 USA
devard@rpi.edu,rjradke@ecse.rpi.edu

Abstract

Motivated by applications in surveillance sensor networks, we present a distributed algorithm for the automatic, external, metric calibration of a network of cameras with no centralized processor. We model the set of uncalibrated cameras as nodes in a communication network, and propose a distributed algorithm in which each camera only communicates with other cameras that image some of the same scene points. Each node independently forms a neighborhood cluster on which the local calibration takes place, and calibrated nodes and scene points are incrementally merged into a common coordinate frame. The accurate performance of the algorithm is illustrated using examples that model real-world sensor networking situations.

Keywords: camera calibration, metric reconstruction, distributed algorithms, sensor networks, bundle adjustment, structure from motion.

1. Introduction

Existing computer vision research on collections of tens or hundreds of cameras generally takes place in a controlled environment with a fixed camera configuration. Traditional vision tasks such as tracking, 3D visualization, or terrain mapping are typically undertaken in situations where images from all cameras are quickly communicated to a central processor. In contrast, we are motivated by vision problems in wireless sensor networks. Such networks will be essential for 21st century military, environmental, and surveillance applications [1], but pose many challenges to traditional vision. In a typical scenario, camera nodes are randomly distributed in an environment and their initial positions are unknown. Even if some nodes are equipped with GPS receivers, these systems cannot be assumed to be highly accurate and reliable [29], and GPS gives no information about the orientation of a directed sensor such as a camera. The nodes are unsupervised after deployment, and generally have no knowledge about the topology of the broader network [8]. Most nodes are unable to communicate beyond a short distance due to power limitations and short-range antennas, and communication must be kept to a minimum because it is power-intensive.

Furthermore, a realistic camera network is constantly in motion. The number and location of cameras changes as old cameras wear out and new cameras are deployed to replace them. Precipitation, wind, and seismic events will jolt the cameras, or remote directives may reposition or reorient them for a variety of tasks. Mobile cameras mounted in vehicles will provide images from time-varying perspectives. Even in a controlled environment like an airport, surveillance cameras may be moved randomly to deter terrorists who look for patterns.

This paper is concerned with the problem of *camera calibration*: that is, the estimation of each camera's 3D position, orientations, and focal length. Calibration is an essential prerequisite for many computer vision algorithms, such as multicamera tracking or volume reconstruction. Calibration of a fixed configuration of cameras is an active area of vision research, and good results are achievable when the images are all accessible to a powerful, central processor. On the other hand, here we present a *distributed* algorithm that can extend to *dynamic* networks of cameras. We view this work as the first step in a new research domain of dynamic camera networks that incorporates state-of-the-art algorithms in both computer vision and wireless networking.

We model a set of uncalibrated cameras as nodes in a communication network, and propose a distributed algorithm in which each camera only communicates with other cameras that image some of the same scene points. Each node independently forms a neighborhood cluster on which the local calibration takes place, and calibrated nodes and scene points are incrementally merged into a common coordinate frame. The accurate performance of the algorithm is illustrated using examples that model real-world sen-

[*] This work was supported in part by the US National Science Foundation, under the award IIS-0237516.

sor networking situations. Section 2 reviews prior work on multicamera systems and their calibration. Section 3 describes our distributed, metric reconstruction algorithm, and Section 4 demonstrates the results of the algorithm for realistic test data. We conclude in Section 5.

2. Prior Work

This paper concentrates on issues related to computer vision, as opposed to explicitly modeling the communication network. We implicitly make several assumptions based on active research problems with good preliminary solutions in the wireless networking community:

1. Nodes that are able to directly communicate can automatically determine that they are neighbors. In a real sensor network, these links are formed by radio, infrared, or optical media [1, 28].
2. If each node knows its one-hop neighbors, a message from one specific node to another can be delivered efficiently (i.e. without broadcasting to the entire network) [3, 4, 5, 24].
3. If necessary, a message can be sent efficiently from one node to all the other nodes [15, 19].

Also, we assume that data communication between nodes has a much higher cost (e.g. in terms of power consumption) than data processing within a node [27], so that messages between nodes should be compact.

In the remainder of this section we discuss prior work related to multi-camera calibration. While there has been a substantial amount of work for 1-, 2-, and 3-camera systems where the cameras share roughly the same point of view, we are primarily interested in relatively wide-baseline settings where the cameras number in the tens or hundreds and have very different perspectives.

2.1. Multicamera Systems

There are only a few research systems in which tens or hundreds of cameras simultaneously observe a scene, and these systems are usually housed in a highly controlled laboratory environment. Such systems include the Virtualized Reality project at Carnegie Mellon University [17] and similar stage areas at the University of California at San Diego [23] and the University of Maryland [6]. Such systems are typically carefully calibrated using test objects of known geometry, and an accurate initial estimate of the cameras' positions and orientations.

There are relatively more systems in which a single camera acquires many images of a static scene from different locations (e.g. [12, 20]), but the cameras in such situations are generally closely spaced. Notable cases in which many images are acquired from widely spaced positions of a single camera are include Debevec et al. [7] and Teller et al. [38]. However, in these cases, rough calibration of the cameras was available *a priori*, from an explicit model of the 3-D scene or from GPS receivers.

From the networking side, various researchers have explored the idea of a Visual Sensor Network (VSN), where each node has an image or video sequence that is to be shared/combined/interpreted by other nodes [11, 25, 42, 43]. However, most of these discussions have not exploited the full potential of the state of the art in computer vision.

2.2. Multicamera Calibration

Typically, a camera is described by two sets of parameters: internal and external. Internal parameters include the focal length, position of principal points and the skew. The external parameters describe the placement of the camera in a world coordinate system using a rotation matrix and a translation vector. The classical problem of externally calibrating a pair of cameras is well-understood [40]; the parameter estimation usually requires a set of feature point correspondences in both images. When no points with known 3-D locations in the world coordinate frame are available, the cameras can be calibrated up to a similarity transformation [14]. N-camera calibration can be accomplished by minimizing a nonlinear cost function of the calibration parameters and a collection of unknown 3-D scene points projecting to matched image correspondences; this process is called *bundle adjustment* [39].

Several algorithms have been proposed for calibration of image sequences through the estimation of projective transformations [18, 30], fundamental matrices [44] or trifocal tensors [10] between nearby images. However, such methods operate only on closely-spaced, explicitly ordered sequences of images, as might be obtained from a video camera, and are designed to obtain a good initial estimate for bundle adjustment to be undertaken at a central processor. Svoboda et al. [36] proposed a multi-camera system where a person carrying a laser pointer walks around the room and calibration is obtained by tracking the pointer across all images. This is again a centralized calibration problem where calibration depends on presence of common scene points across all images.

Teller and Antone [37] and Sharp [33] both considered calibration of a number of unordered views related by a graph similar to the vision graph we describe in Section 3 below. Schaffalitzky and Zisserman [31] recently described an automatic clustering method for a set of unordered images from different perspectives, which corresponds to constructing a vision graph containing several complete subgraphs. We emphasize the main point that sets our work apart from these systems: none of them view the collection of cameras as the nodes in a communication network, and none of their algorithms are distributed.

3. Distributed Metric Calibration

We propose to model the sensor network with two undirected graphs: a *communication graph* and a *vision graph*. We illustrate the idea in Figure 1 with a hypothetical network of ten nodes.

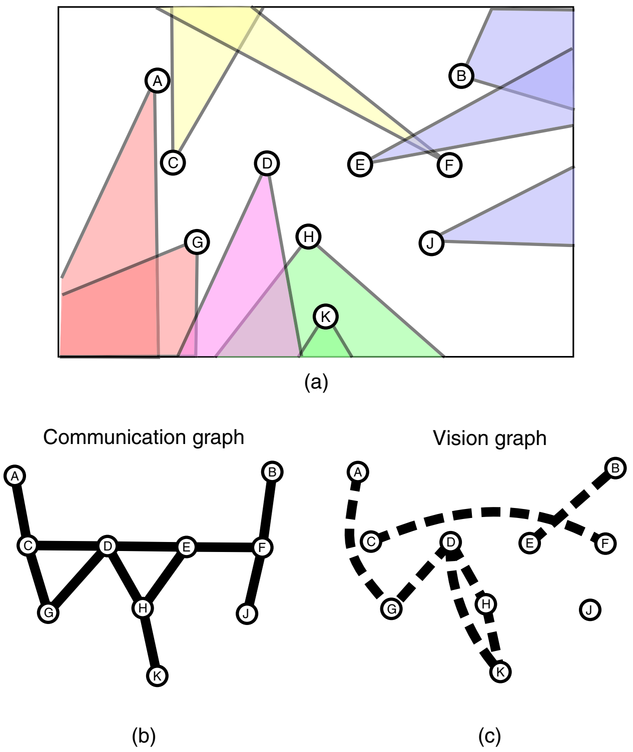


Figure 1. (a) A snapshot of the instantaneous state of a camera network, indicating the fields of view of ten cameras. (b) The associated communication graph. (c) The associated vision graph. Note the presence of an edge in one graph does not imply the presence of the same edge in the other graph.

Figure 1a shows a snapshot of the locations and orientations of the cameras. Figure 1b illustrates the communication graph for the network; an edge appears between two cameras in this graph if they have one-hop direct communication. This is a common abstraction in wireless ad-hoc networks (see [13] for a review). The communication graph is mostly determined by the locations of the nodes and the topography of the environment; in a wireless setting, the instantaneous power each node can expend towards communication is also a factor.

Figure 1c illustrates the *vision graph* for the network; an edge appears between two cameras in this graph if they observe some of the same scene points from different perspectives. We note that the presence of an edge in the communication graph does not imply the presence of the same edge in the vision graph, since the cameras may be pointed in different directions (for example, cameras A and C). Conversely, an edge can connect two cameras in the vision graph despite a lack of physical proximity between them (for example, cameras C and F). In networking terminology, the vision graph is a called an *overlay graph* on the communication network.

Ideally, the vision graph should be estimated automatically, rather than constructed manually [33] or specified *a priori* [37]. However, for the purposes of this initial investigation, we assume that the feature correspondences used in the calibration procedure are given. We note that establishing robust feature correspondences across many images is itself a difficult problem, even in the centralized case. In the future, we plan to automatically establish arcs in the vision graph by detecting and matching invariant features [21, 22]. Furthermore, while we consider the static case here, vision and communication graphs in real sensor-networking applications will be dynamic due to the changing presence, position and orientation of each camera in the network, as well as time-varying channel conditions.

Our goal is to design a calibration system in which each camera only communicates with (and possesses knowledge about) those cameras connected to it by an edge in the vision graph. We assume that each camera node calibrates independently of the rest by estimating a metric reconstruction based on a local cluster formed by neighboring cameras in the vision graph. We refer to the calibration at a node as the *local calibration*. Arbitrary coordinate frames arising from local calibrations are then aligned iteratively to a common Coordinate frame. The following sections describe the details of each of these phases.

3.1. Notation

We assume that the vision graph contains M nodes, each representing a perspective camera described by a 3x4 matrix P_i:

$$P_i = K_i \left[R_i,\ t_i\right]. \tag{1}$$

Here, $R_i \in SO(3)$ and $t_i \in \mathbb{R}^3$ are the rotation matrix and translation vector comprising the external camera parameters. K_i is the intrinsic parameter matrix and is assumed here to be $diag(f_i, f_i, 1)$, where f_i is the focal length.[1] Each camera images some subset of a set of N points $X \in \mathbb{R}^3$. This subset is described by $V_i \subset \{1, \ldots, N\}$.

[1] Aspect ratio, center of projection and image skew can be easily incorporated into the K matrix and the following algorithm. Lens distortion may also be a factor in real camera networks, but this distortion does not need to be estimated with reference to other cameras.

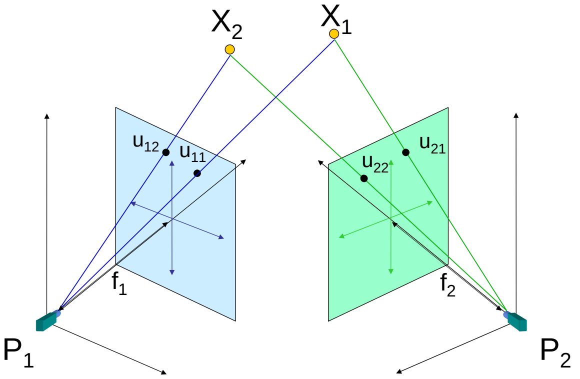


Figure 2. Notation and geometry of the imaging system.

The projection of X_j onto P_i is given by $u_{ij} \in \mathbb{R}^2$ for $j \in V_i$:

$$\lambda_{ij} \begin{bmatrix} u_{ij} \\ 1 \end{bmatrix} = P_i \begin{bmatrix} X_j \\ 1 \end{bmatrix}, \qquad (2)$$

where λ_{ij} is called the projective depth [35]. This image formation process is illustrated in Figure 2. The simplified K matrix assumption for the camera is just to provide an example model for the scheme. The mathematics of the technique does accommodate for more generalized models (see below).

Edges are culled from the vision graph based on neighborhood sufficiency conditions (i.e. there must be a minimal number of neighbors per node that must jointly image some minimal number of points). We define a characteristic function χ_{ij} where $\chi_{ij} = 1$ if node j satisfies the sufficiency conditions at node i. Thus each node is associated with a cluster, $\mathcal{C}_i$, on which the local calibration is carried out:

$$\mathcal{C}_i = \{j \mid \chi_{ij} = 1\}$$

In our experiments, we chose a minimum cluster size of 4 nodes that must share 12 corresponding points. At each node i, the local calibration results in an estimate of the local camera parameters $\hat{P}_i^i$ as well as the camera parameters of i's neighbors, $\{\hat{P}_j^i, j \in \mathcal{C}_i\}$. (Conversely, this means that each of camera i's neighbors will have a slightly different estimate of where i is; we discuss how to reconcile these differences below.) The 3D scene points reconstructed at i are given by $\{\hat{X}_k^i\}$, which are estimates of $\{X_k | k \in \bigcap_{j\in\{i,\mathcal{C}_i\}} V_j\}$. If neighborhood sufficiency conditions are not met at a node, then local calibration does not take place (though an estimate can still be obtained; see below).

3.2. Local Calibration

Here, we describe the local calibration problem at node i. We denote $\{P_1, \ldots, P_m\}$ as the cameras in i's cluster, where $m = |\{i, \mathcal{C}_i\}|$. Similarly, we denote $\{X_1, \ldots, X_n\}$ as the 3D points used for calibration, where $n = |\{X_k | k \in \bigcap_{j\in\{i,\mathcal{C}_i\}} V_j\}|$. Node i must estimate the camera parameters P as well as the unknown scene points X using only the 2D image correspondences $\{u_{ij}, i = 1, \ldots, m, j = 1, \ldots, n\}$.

Taking into account all the image projections, (2) can be written as

$$\begin{aligned} W &= \begin{pmatrix} \lambda_{11}u_{11} & \lambda_{12}u_{12} & \ldots & \lambda_{1n}u_{1n} \\ \lambda_{21}u_{21} & \lambda_{22}u_{22} & \ldots & \lambda_{2n}u_{2n} \\ \vdots & & & \\ \lambda_{n1}u_{n1} & \lambda_{n2}u_{n2} & \ldots & \lambda_{nn}u_{nn} \end{pmatrix}. \\ &= \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{pmatrix} \begin{pmatrix} X_1^h & X_2^h & \ldots & X_n^h \end{pmatrix} \end{aligned} \qquad (3)$$

Here, X^h denotes X represented in homogeneous coordinates, i.e. $X^h = [X^T, 1]^T$.

Sturm and Triggs [35] suggested a factorization method that recovers the projective depths as well as the structure and motion parameters from the above equation. They used relationships between fundamental matrices and epipolar lines in order to recover the projective depths λ_{ij}. Once the projective depths are recovered, the structure and motion are recovered by SVD factorization of the best rank-4 approximation to the measurement matrix W:

$$\begin{aligned} W &= U_{3m\times4}\Sigma V_{4\times n} \\ &= \left(U_{3m\times4}\sqrt{\Sigma}\right)\left(\sqrt{\Sigma}V_{4\times n}\right) \\ &= \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{pmatrix}_{3m\times4} \begin{pmatrix} X_1^h & X_2^h & \ldots & X_n^h \end{pmatrix}_{4\times n}. \end{aligned}$$

However, there is a projective ambiguity in the reconstruction, since

$$\left(\hat{P}H^{-1}\right)\left(H\hat{X}\right) = \hat{P}\hat{X}. \qquad (4)$$

for any 4×4 nonsingular matrix H. There exists some H_i relating the projective factorization to the true metric (i.e. Euclidean) factorization, given by

$$\hat{P}_i H_i^{-1} = K_i \begin{bmatrix} R_i & t_i \end{bmatrix}. \qquad (5)$$

This H_i can be estimated by using the dual to absolute conic, which remains invariant under rigid transformations

[14]. The dual to the absolute conic, Ω, satisfies the equation

$$P_i \Omega P_i^T \propto K_i K_i^T = \alpha_i \omega_i. \tag{6}$$

where ω_i is the dual to the image of the absolute conic and α_i is the constant of proportionality. The homography H_i satisfies

$$\Omega = \tilde{H}\tilde{H}^T. \tag{7}$$

where $\tilde{H}$ is the first 3 columns of H_i. Seo, Heyden and Cipolla [32] and Pollefeys [26] proposed methods for transforming a projective factorization into a metric one based on (6)-(7). The Euclidean reconstruction so recovered is related to the true camera/scene configuration by an unknown similarity transform that cannot be estimated without additional measurements of the scene. As mentioned before the above schemes also accommodate more generalized camera models.

The above scheme might fail under two conditions. First, there may be a small fraction of outliers in the correspondences, e.g. caused by repetitive patterns in the images, that can cause the parameter estimates to be inaccurate. Standard rejection algorithms such as RANSAC [9] should be able to detect and remove such outliers. It is also possible that the camera cluster might be close to a critical configuration of the metric reconstruction and might result in a poorly conditioned problem. For example, cameras whose centers are collinear or that have a common center of focus are critical for metric calibration [34]. Hence, the local calibration procedure may yield unreliable estimates. Such failures typically can be detected automatically from large reprojection errors or disproportionately large values of the estimated internal parameters. However, the failure of calibration at a particular node can be easily compensated for by obtaining one of the neighboring estimates, e.g. node i can calibrate its camera based on node j's estimation, $\hat{P}_i^j, j \in \mathcal{C}_i$.

3.3. Frame Alignment

The estimates obtained by local calibrations have different coordinate frames, each offset from the true frame by an unknown similarity transformation (i.e. rotation, translation, and scaling). In order for cameras on the network to coordinate higher-level vision tasks, we require a distributed algorithm to align all nodes to a common coordinate frame.

We assume each node has a unique identifier $\mathrm{id}_i \in \mathbb{R}$, such as a factory serial number, and an alignment index a_i that is initially set to id_i. Each node i then continually aligns its frame to the *available* neighbor with lowest alignment index $a_j, j \in \mathcal{C}_i$. By "available", we mean a neighbor node that is not currently aligning its frame to that of node i. Ultimately, if the vision graph is connected, each frame will be aligned to the frame of the camera with the lowest identifier; a similar scheme was presented and analyzed in [16]. The procedure at node i is as follows.

1. Let $j_{\min} = \arg\min_{j \in \mathcal{C}_i} a_j$, and $a_{\min} = a_{j_{\min}}$.
2. If $a_{\min} < a_i$, then
   (a) Align node i to the frame of node $j_{\min}$.
   (b) Set $a_i = a_{\min}$.
3. Iterate.

The alignment process in step 2a is accomplished by solving

$$\min_S \sum_{k \in V_i \cap V_{j_{\min}}} \|S\hat{X}_k^i - \hat{X}_k^{j_{\min}}\|^2,$$

where S is constrained to be a similarity transform. This minimization can be accomplished in closed form by Umeyama's method [41]. After convergence, there may still be some disagreement between $\hat{X}_k^i$ and $\hat{X}_k^j, j \in \mathcal{C}_i$, which we deal with by averaging all local estimates for a given point X_k in the common coordinate frame. We note that this step is not strictly necessary when the main goal is to calibrate the cameras, not to recover the structure.

An interesting question is how to determine, in a distributed manner, the node that requires the fewest number of alignment transformations (thus minimizing error accumulation). This node (i.e. the barycenter of the vision graph) could be assigned to have the lowest identifier in the above scheme.

3.4. Summary

The entire algorithm for distributed calibration is summarized below:

1. Form the vision graph from the set of correspondences contained in M views.
2. At each node i,
   (a) Form a subnet $\mathcal{C}_i$ with at least 12 common image measurement points and 4 nodes.
   (b) Estimate a projective reconstruction on the normalized points [35].
3. Calculate the residual reprojection errors. If they are large, discard the local calibration process. Otherwise,
   (a) Form the equation matrix for metric reconstruction and normalize the rows to have unit norm.
   (b) Estimate a metric reconstruction based on the projective cameras [32].

4. Estimate the focal lengths and also the error in the principal points (we assume principal points are known). If the error is large or if the estimated focal lengths seem unreasonable, then discard the local calibration process.

5. Incrementally align each node to a common frame.

   (a) Determine the lowest-labeled available neighbor.

   (b) Estimate a similarity transformation to align the frames [41].

   (c) Perform multiple iterations of alignment until all nodes are initialized and there are no further changes to the frame alignment.

   (d) Average all local estimates of the same scene point to ensure all nodes share the same estimated structure, if desired.

4. Experiments

We studied the algorithm's performance for simulated data by comparing the structure and calibration parameters estimated using the distributed algorithm to ground truth. Both datasets are aligned to the same similarity frame before comparison. The structural error (i.e. error in X's) is calculated as the average distance between the the ground truth and the reconstructed points, relative to the scene diameter. Calibration errors are described by orientation, focal length, and camera center errors. Orientation error is measured in terms of the angle between corresponding optical axes, and focal length error as the deviation from unity of the ratio of the true and estimated values. Error in the camera center is calculated as the average distance between the the ground truth and the estimated centers, relative to the scene diameter.

The following section discusses the details of two experiments: one with many realizations of random scene points and image noise, and one with a more realistic model of camera/scene placement with occlusions.

4.1. Experiment 1

500 scene points were uniformly (randomly) distributed in a 5m-radius sphere and 40 camera nodes (focal length 5cm, zero skew and principal points on image center) were positioned randomly on an elliptical band around the sphere. Each camera was oriented to view a random location inside the scene. Due to limited field of view, each camera images only a portion of the scene (see Figure 3). The local calibration cluster at each camera consists of yet fewer scene points due to the neighborhood sufficiency conditions.

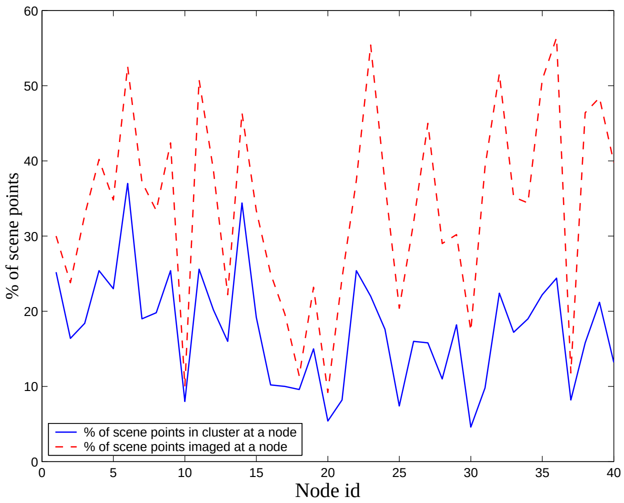


Figure 3. A typical example of the percentage of scene points imaged at each camera (dashed line) and the percentage of scene points in each local calibration cluster (solid line).

Scene points were projected to image planes, and then perturbed with Gaussian noise of standard deviation of 0.5, 1, 1.5, 2 and 2.5 pixels. For each node, the median reconstruction error was calculated. The experiment was repeated with 10 different scene configurations and 10 different perturbations for each noise level. The median reconstruction errors were then averaged over these multiple realizations of noise and scene configurations. Quantitative analysis of the calibration procedure is summarized in Table 1, including the reconstruction error in the 3D points, the Mahalanobis reprojection errors, and camera position, orientation, and focal length errors (recovered from the decomposition (1)).

The overall reconstruction accuracy of scene points is quite good, even in the presence of noise: less than 0.7% median relative error. The Mahalanobis reprojection error, defined as

$$\|u - \hat{u}\| = \left((u - \hat{u})^T \, \Sigma^{-1} (u - \hat{u}) \right)^{1/2}$$

where Σ is the covariance of the image pixels, is also low (< 5), indicating reasonably good estimates.

While the camera orientation error is small, less than 0.14 radians, the relative positional error in the camera centers are quite large for larger values of noise variance (e.g. 9% in the worst case). This indicates that the original and recovered optical axes are nearly parallel, but that the reconstructed cameras are mis-positioned. We note that this sensitivity in camera center estimation is an expected phenomenon and not a failure of the algorithm; moving a camera along its optical axis has a relatively small effect

Noise variance	X_{err}	C_{err}	f_{err}	Ang_{err}	Reprojection error	
					Mahalanobis	Euclidean
0	0	0	0	0	0	0
0.5	0.0897	1.4027	0.0023	0.0261	3.5760	1.7880
1	0.1993	3.1593	0.0059	0.0527	3.7597	3.7597
1.5	0.3370	4.7759	0.0070	0.0789	3.4578	5.1868
2	0.5109	6.1991	0.0133	0.1120	3.5863	7.1726
2.5	0.6710	9.0693	0.0215	0.1324	4.9618	12.4046

Table 1. Error Table : X_{err} : Distance errors in scene recovery (percentage of scene width); C_{err} : Errors in camera centers (percentage of scene width); f_{err}: Focal length error expressed as a relative fraction; Ang_{err}: Orientation error expressed as angular difference between the optical axes (in rad); Reprojection error : Error distance between the original image and the reprojected image points : Mahalanobis distance (dimensionless) and Euclidean distance (in pixels)

on the position of projected points in the absence of severe perspective effects. Indeed, this error in camera centers has little influence on the structure estimation or on the reprojection error that is the basis for the local calibrations.

As for the internal camera parameters, focal lengths are estimated quite accurately (less than 0.03% error). Here we have assumed known principal points. If the estimated principal point of the local calibration at node i is far from the known value, the local estimate is rejected and an estimate obtained from one of node i's neighbors.

Only a portion of the 3D points jointly imaged by all the cameras are reconstructed, due to the neighborhood sufficiency constraints. However, after all nodes have been calibrated, it is straightforward to estimate the missing scene points via a triangulation procedure [2]. Alternately, it is straightforward to include the scene points imaged by at least 2 cluster cameras directly in the optimization function for bundle adjustment.

4.2. Experiment 2

We also studied the performance of the algorithm with data modeling a real-world situation with reasonable dimensions. The scene consisted of 20 cameras surveying two simulated (opaque) structures. The cameras were placed randomly on an elliptical band around the "buildings'". The configuration of the setup is shown in Figure 4. Since the image plane is finite and the "buildings" are opaque, each camera sees only about 25% of the scene points. A total of 4000 scene points uniformly distributed along the walls of the buildings were captured by the 20 cameras and the imaged points were then perturbed by Gaussian random noise with a standard deviation of 1 pixel.

Figures 5a and 5b show the ground-truth configuration and the recovered configuration, respectively. The quality of the shape recovery is evident. Most of the cameras are well-recovered; the average orientation error is 0.008 radians, while the relative focal length error is 0.015. The mean error in estimation of camera positions, relative to the scene width, is 2.1%, while the median error is 0.7%. Two cameras (marked by circles in Figure 5b) are not calibrated due to failing the neighborhood sufficiency constraints, while another camera is particularly poorly calibrated (marked by a square in Figure 5b). We emphasize that the only data used to obtain this result were the positions of matching points in the images taken by the cameras.

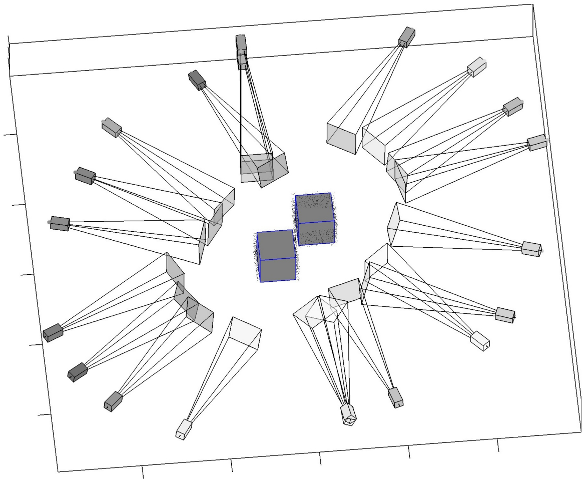
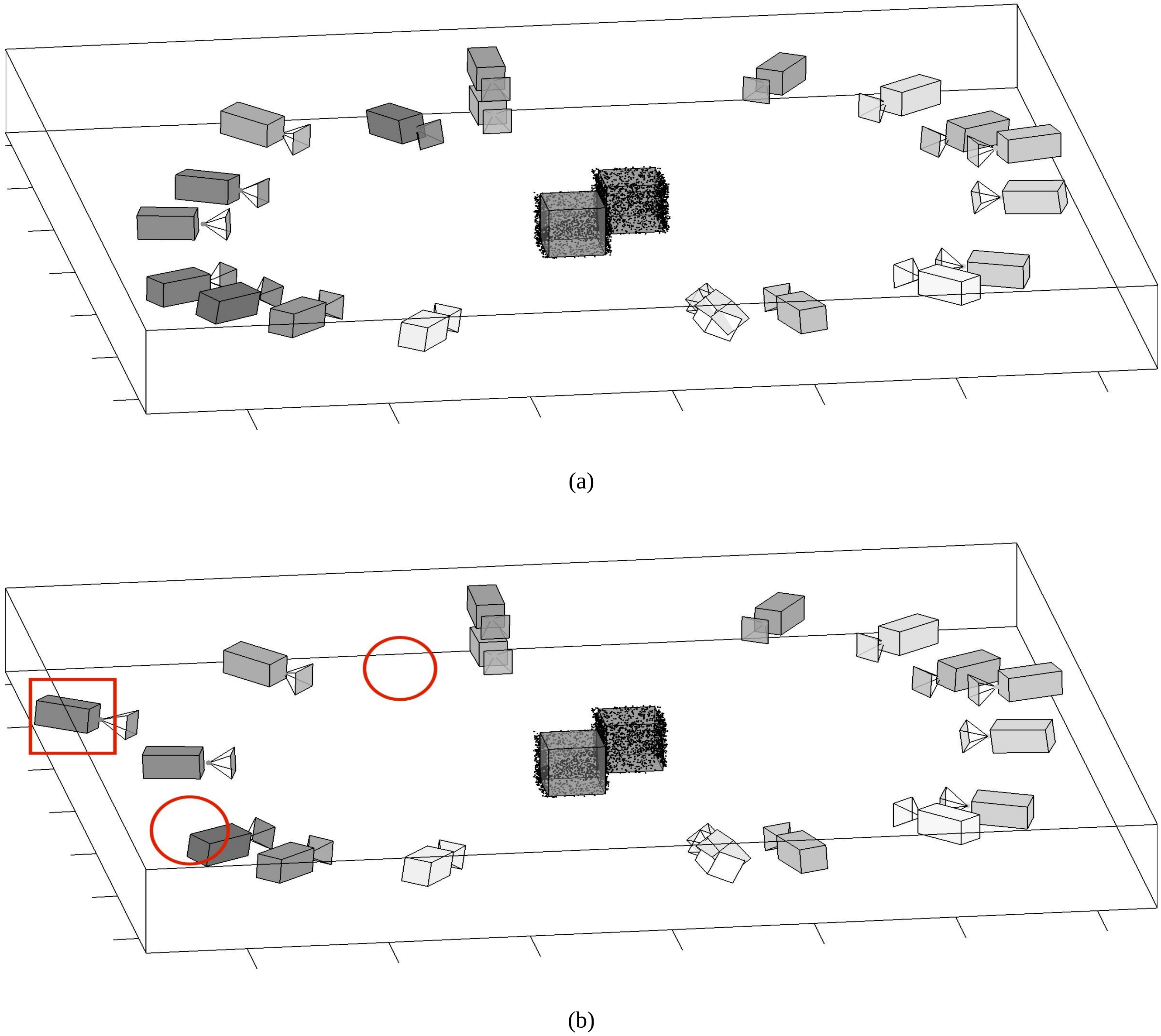


Figure 4. The field of view of each of the simulated cameras.

5. Conclusions

We have demonstrated a distributed metric calibration algorithm whose results are comparable to centralized algorithms. Furthermore, the algorithm is asynchronous in



(a)

(b)

Figure 5. (a) Ground truth structure and camera positions (b) Recovered structure and camera positions. The two circles indicate cameras that are not calibrated and the square indicates a camera that is particularly poorly calibrated.

the sense that more than one node can process at a time, and there need be no ordering on the node processing.

Since the emphasis on this paper is on demonstrating the workability of the distributed calibration scheme, we have assumed ideal networking conditions. While such assumptions do simplify the simulations, they do not change the overall approach in more complicated cases. Analysis of the associated communication issues under varying conditions and topologies (e.g. using a network simulator) is the next logical step towards developing a more realistic model of the distributed network. Eventually, we plan to build wireless camera nodes to test the performance of our algorithms in real situations.

As mentioned above, we plan to generate the vision graph automatically based on invariant feature matching. We are currently working to develop more principled versions of the frame alignment process based on the underlying probability densities of the estimated camera parameters.

Finally, we note that distributed camera calibration is only the first step towards additional distributed computer vision algorithms, such as view synthesis or image-based query and routing.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[2] M. Anderson and D. Betsis. Point reconstruction from noisy images. *Journal of Mathematical Imaging and Vision*, 5:77–90, 1995.

[3] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. Hubaux, and J. L. Boudec. Self-organization in mobile ad-hoc networks: the approach of terminodes. *IEEE Communications Magazine.*, June 2001.

[4] S. Capkun, M. Hamdi, and J. P. Hubaux. GPS-free positioning in mobile ad-hoc networks. In *34th HICSS*, January 2001.

[5] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int'l J. High Performance Computing Applications*, 2002. Also Xerox Palo Alto Research Center Technical Report P2001-10113, May 2001.

[6] L. Davis, E. Borovikov, R. Cutler, D. Harwood, and T. Horprasert. Multi-perspective analysis of human action. In *Proceedings of Third International Workshop on Cooperative Distributed Vision*, November 1999. Kyoto, Japan.

[7] P. E. Debevec, G. Borshukov, and Y. Yu. Efficient view-dependent image-based rendering with projective texture-mapping. In *9th Eurographics Rendering Workshop*, June 1998. Vienna, Austria.

[8] D. Estrin et al. *Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. National Academy Press, 2001. Washington, D.C.

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.

[10] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV (1)*, pages 311–326, 1998.

[11] A. E. Gamal. Collaborative visual sensors, 2004. `http://mediax.stanford.edu/projects/cvsn.html`.

[12] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *Computer Graphics (SIGGRAPH '96)*, pages 43–54, August 1996.

[13] Z. Haas et al. Wireless ad hoc networks. In J. Proakis, editor, *Encyclopedia of Telecommunications*. John Wiley, 2002.

[14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[15] J. Hromkovic, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (broadcasting and gossiping). In F. Hsu and D.-A. Du, editors, *Combinatorial Network Theory*, pages 125–212. Kluwer, 1995.

[16] R. Iyengar and B. Sikdar. Scalable and distributed gps free positioning for sensor networks. In *IEEE ICC*, May 2003.

[17] T. Kanade, P. Rander, and P. J. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE Multimedia*, 4(1):34–47, 1997.

[18] E.-Y. Kang, I. Cohen, and G. Medioni. A graph-based global registration for 2D mosaics. In *15th International Conference on Pattern Recognition(ICPR)*, 2000. Barcelona, Spain.

[19] D. W. Krumme, G. Cybenko, and K. N. Venkataraman. Gossiping in minimal time. *SIAM J. Comput.*, 21(1):111–139, 1992.

[20] M. Levoy and P. Hanrahan. Light field rendering. In *Computer Graphics (SIGGRAPH '96)*, pages 31–42, August 1996.

[21] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, September 1999.

[22] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, volume 1, pages 128–142, 2002.

[23] S. Moezzi, L.-C. Tai, and P. Gerard. Virtual view generation for 3D digital video. *IEEE Multimedia*, 4(1):18–26, Jan.-March 1997.

[24] D. Niculescu and B. Nath. Trajectory based forwarding and its applications. Technical report, Rutgers University, 2002. `http://www.cs.rutgers.edu/dataman/papers/tbftr.pdf`.

[25] K. Obraczka, R. Manduchi, and J. Garcia-Luna-Aceves. Managing the information flow in visual sensor networks. In *The Fifth International Symposium on Wireless Personal Multimedia Communication (WMPC)*, October 2002.

[26] M. Pollefeys, R. Koch, and L. J. V. Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *ICCV*, pages 90–95, 1998.

[27] J. Pottie and W. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 3(5):51–58, May 2000.

[28] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*, August 2000.

[29] A. Savvides, C. C. Han, and M. B. Srivastava. Dynamic fine-grained localization in ad-hoc wireless sensor networks. In *International Conference on Mobile Computing and Networking (MobiCom) 2001*, July 2001.

[30] H. S. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *European Conference on Computer Vision*, pages 103–119, 1998.

[31] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *European Conference on Computer Vision*, volume LNCS 2350, pages 414–431, June 2002. Copenhagen, Denmark.

[32] Y. Seo, A. Heyden, and R. Cipolla. A linear iterative method for auto-calibration using dac equation. In *CVPR*, 2001.

[33] G. Sharp, S. Lee, and D. Wehe. Multiview registration of 3-

D scenes by minimizing error between coordinate frames. In *European Conference on Computer Vision*, volume LNCS 2351, pages 587–597, June 2002. Copenhagen, Denmark.

[34] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 1100–1105, June 1997.

[35] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV (2)*, pages 709–720, 1996.

[36] T. Svoboda, H. Hug, and L. V. Gool. Viroom – low cost synchronized multicamera system and its self-calibration. In *LNCS 2449*, pages 515–522. Springer, 2002.

[37] S. Teller and M. Antone. Scalable, extrinsic calibration of omni-directional image networks. *International Journal of Computer Vision*, 49(2/3):143–174, September/October 2002.

[38] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master. Calibrated, registered images of an extended urban area. In *IEEE CVPR*, December 2001.

[39] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[40] R. Tsai. A versatile camera calibration technique for high-accuracy 3-D machine vision metrology using off-the-shelf TV cameras and lenses. In L. Wolff, S. Shafer, and G. Healey, editors, *Radiometry – (Physics-Based Vision)*. Jones and Bartlett, 1992.

[41] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[42] H. Wu and A. Abouzeid. Energy efficient distributed JPEG2000 image compression in multihop wireless networks. In *Proceedings of 4th Workshop on Applications and Services in Wireless Networks, Boston, Massachusetts*, 2004.

[43] H. Wu and A. Abouzeid. Power aware image transmission in energy constrained wireless networks. In *Proceedings of The 9th IEEE Symposium on Computers and Communications (ISCC'2004), Alexandria, Egypt, June 2004.*, 2004.

[44] Z. Zhang and Y. Shan. Incremental motion estimation through local bundle adjustment. Technical report, MSR-TR-01-54, 2001.