Distributed Metric Calibration of Ad-Hoc Camera Networks

DHANYA DEVARAJAN, RICHARD J. RADKE and HAEYONG CHUNG
Rensselaer Polytechnic Institute

We discuss how to automatically obtain the metric calibration of an ad-hoc 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 performs a local, robust bundle adjustment over the camera parameters and scene points of its neighbors in an overlay "vision graph". We analyze the performance of the algorithm on both simulated and real data, and show that the distributed algorithm results in a fairer allocation of messages per node while achieving comparable calibration accuracy to centralized bundle adjustment.

Categories and Subject Descriptors: I.4.1 [**Image Processing And Computer Vision**]: Digitization and Image Capture—*camera calibration, imaging geometry*; I.4.8 [**Image Processing And Computer Vision**]: Scene Analysis—*sensor fusion*; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*3D/stereo scene analysis*

General Terms: Algorithms, Measurement, Performance

Additional Key Words and Phrases: camera calibration, metric reconstruction, distributed algorithms, sensor networks, bundle adjustment, structure from motion

1. INTRODUCTION

Existing computer vision research on integrating images from a large number of cameras generally assumes that the images are all available at a powerful, central processor that may have a priori partial knowledge of the cameras' configuration. In contrast, we are motivated by computer vision problems in large (e.g. outdoor) ad-hoc camera sensor networks, in which processing is decentralized and each camera has little a priori knowledge about its relationship to other cameras or its environment. Such networks will be essential for 21st century military, environmental, and surveillance applications [Akyildiz et al. 2002], but pose many challenges to traditional computer vision. In a typical scenario, camera nodes are distributed throughout an environment (e.g. a building or a battlefield) and their initial positions and orientations are unknown. The nodes are unsupervised after deployment,

This paper has been accepted for publication in *ACM Transactions on Sensor Networks*, in press as of April 2006. Author's addresses: D. Devarajan, R. Radke, and H. Chung, Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180. Please address correspondence to R. Radke. This work was supported in part by the US National Science Foundation, under the award IIS-0237516.

and generally have no knowledge about the topology of the broader network [Estrin et al. 2001]. Such networks must have the ability to self-calibrate using only observations of their environment in order to perform various higher-level vision tasks such as event detection, object tracking, volume reconstruction, change detection, or navigation. Moreover, as in any sensor network, the network must cope with power limitations and short-range antennas that restrict the amount and distance of communication between the nodes.

Calibration of a fixed configuration of cameras is an active area of computer vision research, and good results are achievable when the images are all accessible to a powerful, central processor. In contrast, here we model a set of uncalibrated cameras as nodes in a communication network, and propose a distributed algorithm for estimating networked camera calibration parameters in which each camera only communicates with other cameras that image some of the same scene points. Viewed another way, we describe a sensor localization problem for camera networks and address it using structure-from-motion techniques from the computer vision literature. We analyze the performance of the distributed calibration algorithm using examples that model real-world camera networks, and discuss its messaging behavior compared to a centralized calibration scheme. We show that the distributed algorithm achieves a high-accuracy result using about the same number of messages as in the centralized case, but distributes these messages more fairly across the nodes.

The paper is organized as follows. Section 2 reviews the type and method of calibration for existing multicamera networks at research institutions today. Section 3 describes our distributed, metric reconstruction algorithm, and Section 4 analyzes the performance of the algorithm on both simulated and real datasets, in terms of both calibration accuracy and messaging overhead. We conclude in Section 5. A shorter, earlier version of this work appeared in the 2004 Workshop on Broadband Advanced Sensor Networks (BASENETS '04) [Devarajan and Radke 2004].

2. PRIOR WORK

This paper concentrates largely on issues related to computer vision, as opposed to explicitly modeling the physical layer of the communication network. We work at the level of abstraction of the messages that are passed between nodes, and analyze our algorithms with respect to the number of messages per node that are required for camera calibration. We implicitly make several assumptions based on active research problems with good preliminary solutions in the wireless networking community. For example, we assume that 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 communication [Akyildiz et al. 2002; Priyantha et al. 2000]. If each node knows its one-hop neighbors, we assume that a message from one specific node to another can be delivered efficiently (i.e. without broadcasting to the entire network) [Blazevic et al. 2001; Capkun et al. 2001; Chu et al. 2002; Niculescu and Nath 2003]. Finally, 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 [Pottie and Kaiser 2000], so that messaging should be kept to a minimum.

In the remainder of this section we discuss prior work related to multi-camera calibration and active vision networks. 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 number of cameras is large and the cameras have very different perspectives.

2.1 Multicamera Systems

Systems comprising a large number of cameras are generally contained in highly-controlled laboratory environments, such as the Virtualized Reality project at Carnegie Mellon University [Kanade et al. 1997] and similar stage areas at the University of California at San Diego [Moezzi et al. 1997] and the University of Maryland [Davis et al. 1999]. Such systems are typically carefully calibrated using test objects of known geometry and an accurate initial estimate of the cameras' positions and orientations. For example, Hörster et al [Hörster et al. 2005] described a system for calibrating an in-room multi-camera system using reference images displayed on multiple flat-panel displays.

There are relatively more systems in which a single camera acquires many images of a static scene from different locations (e.g. [Gortler et al. 1996; Levoy et al. 2000]), 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 include [Debevec et al. 1998] and [Teller et al. 2001]. 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), in which each node has an image or video sequence that is to be shared/combined/interpreted by other nodes [Choi et al. 2004; Obraczka et al. 2002; Wu and Abouzeid 2004a; 2004b]. 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, the position of the principal point, and the skew. The external parameters describe the position of the camera in a world coordinate system using a rotation matrix and a translation vector. The classical problem of determining the rigid motion relating a pair of cameras is well-understood [Tsai 1992]; 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 [Hartley and Zisserman 2000]. N-camera calibration can be accomplished by minimizing a nonlinear cost function of the camera parameters and a collection of unknown 3-D scene points projecting to matched image correspondences; this process is called bundle adjustment [Triggs et al. 2000].

Several algorithms have been proposed for calibration of *image sequences* through the estimation of fundamental matrices [Zhang and Shan 2001] or trifocal tensors [Fitzgibbon and Zisserman 1998] 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.

[Antone and Teller 2002] and [Sharp et al. 2002] 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 2002] 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. [Rahimi et al. 2004] described a centralized method to simultaneously track an object and calibrate an indoor camera network. Their algorithm required prior knowledge of the motion dynamics of the object, and only considered non-overlapping images. [Sinha et al. 2004] used dynamic silhouettes for the centralized calibration of a camera network. We emphasize that unlike these methods, we explicitly model the underlying communication network, and analyze the performance of the vision algorithm with respect to both calibration accuracy and messaging overhead. [Choi et al. 2004] described a distributed calibration algorithm using an approach similar to ours. However, they used a few anchor nodes equipped with GPS to fix absolute position, and a checkerboard test object inserted into the environment for calibration purposes.

3. DISTRIBUTED METRIC CALIBRATION

We model a camera network with two undirected graphs: a *communication graph* and a *vision graph*. Figure 1 illustrates the idea with a hypothetical network of ten nodes. Figure 1a shows a snapshot of the locations and orientations of the cameras. Figure 1b illustrates a possible communication graph for the network; an edge appears between two cameras in this graph if they have one-hop direct communication, a common abstraction in wireless ad-hoc networks [Haas et al. 2002]. 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). Several examples of communication and vision graphs for realistic camera networks are illustrated in Section 4.

It is preferable that the vision graph be estimated automatically, rather than constructed manually [Sharp et al. 2002] or specified a priori [Antone and Teller 2002]. Arcs in the vision graph can be automatically established by detecting and matching corresponding features between images; we discuss one approach to this problem in Section 3.5. While we consider the static case here, vision and communication graphs in real sensor-networking applications would be dynamic due to the changing presence, position and orientation of each camera in the network, as well as time-varying channel conditions.

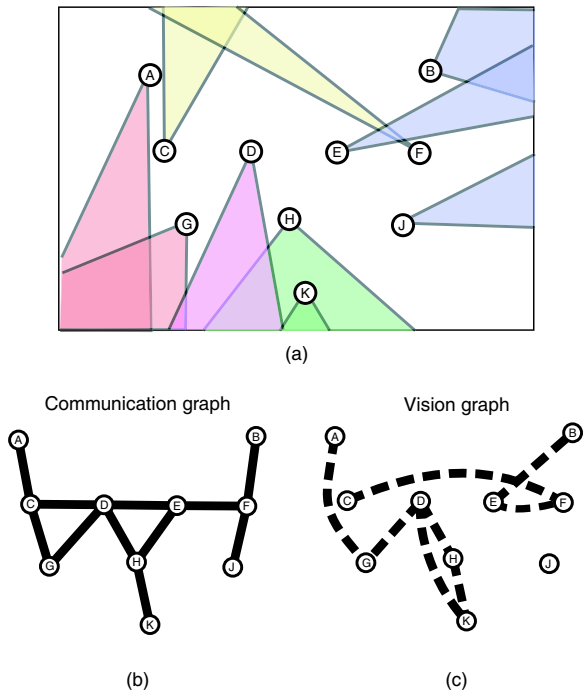


Fig. 1. (a) A snapshot of the instantaneous state of a camera network, indicating the fields of view of ten cameras. (b) One possible associated communication graph. (c) The associated vision graph. Note that the presence of an edge in one graph does not imply its presence in the other graph.

In our approach, each camera node calibrates itself independently based on information shared by cameras adjacent to it in the vision graph. This local calibration would allow cameras that image part of the same scene to exchange and interpret visual information necessary for a higher-level task, such as tracking or estimating the shape of a target that moves through the field of cameras, without requiring any single node to know the global configuration of the entire network. The result of the calibration will be that each camera has an estimate of 1) its own location, orientation, and focal length, 2) the corresponding parameters for each of its neighbors in the vision graph, and 3) the 3D positions of the scene points corresponding to the image features it has matched with its neighbors. It is important to obtain the reconstruction in a *metric* framework, where the recovered geometry of the cameras/scene differs from the truth only by an unknown rotation, translation, and scale.

We now discuss the calibration process in more detail. Good general references on cameras and calibration that go deeply into the issues below are [Hartley and Zisserman 2000; Faugeras et al. 2001].

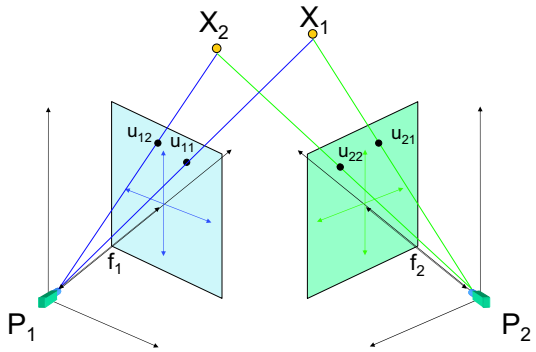


Fig. 2. Notation and geometry of the imaging system.

3.1 Notation and Problem Statement

We assume that the vision graph contains M nodes, each representing a perspective camera described by a 3×4 matrix P_i:

$$P_i = K_i R_i^T \left[I_{3\times 3} \;\; -C_i \right]. \tag{1}$$

Here, $R_i \in SO(3)$ and $C_i \in \mathbb{R}^3$ are the rotation matrix and optical center comprising the external camera parameters. K_i is the intrinsic parameter matrix, given by

$$K_i = \begin{bmatrix} f_i\alpha_x & s & p_x \\ 0 & f_i\alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f_i is the focal length, α_x and α_y represent the width and height of each pixel, s is the skew, and $[p_x, p_y]$ is the principal point. In this paper, we assume that K_i can be written as $diag(f_i, f_i, 1)$. The simplification is justified for the real cameras we studied in Section 4.2, and is a reasonable assumption for standard focusing and zooming cameras [Pollefeys et al. 1998]. We also note that one can always make the principal point of the camera the image origin by recentering, and that the pixel sizes and skew are frequently known or can be easily estimated prior to deployment [Heyden and Åström 1999; Clarke and Fryer 1998]. However, the mathematics below is easily generalizable to a camera with a generic K matrix.

Each camera images some subset of a set of N points $\{X_1, X_2, \ldots, X_N\} \in \mathbb{R}^3$. This subset for camera i is described by $V_i \subset \{1, \ldots, N\}$. The projection of X_j onto P_i is given by $u_{ij} \in \mathbb{R}^2$ for $j \in V_i$:

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

where λ_{ij} is called the projective depth [Sturm and Triggs 1996]. This image formation process is illustrated in Figure 2.

Arcs in the vision graph are formed when two cameras share a sufficient number of points (i.e. they jointly image a sufficiently large part of the environment). This process is discussed further in Section 3.5. We define an indicator function χ_{ij} where $\chi_{ij} = 1$ if a vision graph edge exists between nodes i and j. Each node then forms a cluster $\mathcal{C}_i$ on which the local calibration is performed. Initially, this cluster is formed as $\mathcal{C}_i = \{j|\chi_{ij} = 1\}$. However, nodes that share only a few corresponding points with node i are removed from the cluster in order to ensure a minimum *nucleus* of corresponding points seen by all cameras in the cluster. The neighborhood sufficiency criteria that determine whether calibration at a node is mathematically viable are described in Section 3.3.

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, $\{\hat{P}_j^i, j \in \mathcal{C}_i\}$. The 3D scene points reconstructed at i are given by $\{\hat{X}_k^i\}$, which are estimates of all points seen by at least three cameras in the cluster. In general, our primary interest is in the recovery of the cameras' positions, orientations, and focal lengths, which would be needed for subsequent vision tasks on the network.

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 nucleus of 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 $\{u_{ij}, i = 1, \ldots, m, j = 1, \ldots, n\}$. This problem is called "structure from motion" in the computer vision community.

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

$$W = \begin{pmatrix} \lambda_{11}u_{11} & \lambda_{12}u_{12} & \cdots & \lambda_{1n}u_{1n} \\ \lambda_{21}u_{21} & \lambda_{22}u_{22} & \cdots & \lambda_{2n}u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m1}u_{m1} & \lambda_{m2}u_{m2} & \cdots & \lambda_{mn}u_{mn} \end{pmatrix} = \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{pmatrix} \begin{pmatrix} X_1^h & X_2^h & \cdots & X_n^h \end{pmatrix}.$$

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

Sturm and Triggs [Sturm and Triggs 1996; Triggs 1996] 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 camera matrices and scene point positions are recovered by SVD factorization of the best rank-4 approximation to the measure-

ment matrix W:

$$\begin{aligned} W &\approx U_{3m\times 4}\Sigma_{4\times 4}V_{4\times n} \\ &= \left(U_{3m\times 4}\sqrt{\Sigma}\right)\left(\sqrt{\Sigma}V_{4\times n}\right) \\ &= \begin{pmatrix} \hat{P}_1 \\ \hat{P}_2 \\ \vdots \\ \hat{P}_m \end{pmatrix}_{3m\times 4} \begin{pmatrix} \hat{X}_1^h & \hat{X}_2^h & \cdots & \hat{X}_n^h \end{pmatrix}_{4\times n}. \end{aligned}$$

However, there is substantial ambiguity in this projective reconstruction, since

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

for any 4×4 nonsingular matrix H. This means that while some geometric properties of the reconstructed configuration will be correct compared to the truth (e.g. the order of 3D points lying along a straight line), others will not (e.g. the angles between lines/planes or the relative lengths of line segments). In order to make the reconstruction useful (i.e. to recover the correct configuration up to an unknown rotation, translation, and scale), we need to estimate the matrix H that turns the projective factorization into a metric factorization, so that

$$\hat{P}_i H^{-1} = K_i R_i^T \left[I_{3\times 3} \;\; -C_i\right].$$

and K_i is in the correct form (e.g. diagonal). This H can be estimated using properties of projective geometry that are too complicated to fully describe here. The key concept is that there exists a special 4×4 symmetric rank-3 matrix Ω^* called the absolute dual quadric that satisfies the equation

$$\hat{P}_i \Omega^* \hat{P}_i^T \propto K_i K_i^T$$

for every camera i. Since $\hat{P}_i$ is obtained from the projective factorization and we have assumed a diagonal form for each (unknown) K_i, each camera puts four linear constraints on the elements of Ω^*. Once Ω^* is estimated, the desired matrix H can be extracted by factoring

$$\Omega^* = H diag\{1,1,1,0\} H^T.$$

Detailed descriptions of metric-from-projective recovery based on this approach are given in [Seo et al. 2001] and [Pollefeys et al. 1998; Pollefeys et al. 2002]. The resulting reconstruction is related to the true camera/scene configuration by an unknown similarity transform that cannot be estimated without additional information about the scene.

Once an initial estimate of the camera/scene geometry is obtained, the local calibration result can be improved by using a nonlinear minimization scheme called bundle adjustment [Triggs et al. 2000]. If $\hat{u}_{jk}$ represents the projection of $\hat{X}_k^i$ onto $\hat{P}_j^i$, then the cost function that is minimized at each cluster i is given by

$$\min_{\substack{\{P_j^i\},j\in\{i,\mathcal{C}_i\} \\ \{X_k^i\},k\in\cap V_j}} \sum_j \sum_k (\hat{u}_{jk} - u_{jk})^T \Sigma_{jk}^{-1} (\hat{u}_{jk} - u_{jk}) \tag{3}$$

where Σ_{jk} is the 2×2 covariance matrix associated with the noise in the image point u_{jk}. The quantity inside the sum is called the Mahalanobis distance between $\hat{u}_{jk}$ and u_{jk}. The minimization is taken over the 3D points in the nucleus as well as the focal lengths, rotation matrix parameters and the translation vectors of all cameras in the cluster. Furthermore, points that are seen by at least three cameras but were not in the nucleus are reconstructed by triangulation [Andersson and Betsis 1995]. While triangulation is mathematically possible with a minimum of two calibrated cameras, this can be very unstable depending on the camera/scene configuration. A second bundle adjustment is then performed over all the reconstructed points and the camera parameters to refine the estimate.

At this point, each camera has an estimate of its relative location and orientation with respect to its neighbors. While the entire camera network could be explicitly brought to the same common coordinate system by estimating and applying chains of similarity transformations [Umeyama 1991], this is not strictly necessary for neighboring cameras to trade visual information, since each camera should have a good estimate of its position and orientation in its neighbors' coordinate systems.

3.3 Neighborhood Sufficiency

The minimum cluster and nucleus size required for viable calibration depend on the number of parameters to be estimated at each node [Pollefeys et al. 1998; Pollefeys et al. 2002]. In our experiments, we use a minimum cluster size of 3 cameras and a nucleus of at least 8 corresponding points. We use 3 cameras per cluster as a minimum since we found that metric calibration does not always result in stable results with fewer cameras. Similarly, we use 8 nucleus points as a minimum since the projective factorization is not mathematically possible with fewer points. We note that even if the sufficiency conditions are not met at node i, it can still obtain estimates of its camera parameters from one or more of its neighbors $j \in \mathcal{C}(i)$.

3.4 Outlier Rejection

As is true for many computer vision algorithms, camera calibration is sensitive to the presence of outliers in the feature correspondences. Outliers can arise when images are noisy or contain repetitive patterns (e.g. windows in buildings). Though the focus of the current paper is on the calibration procedure, we still need to ensure that the correspondences presented to the algorithm are as accurate as possible. For this purpose, we perform outlier rejection in two stages. First, we remove correspondences that are grossly inconsistent with the epipolar geometry for each image pair [Hartley and Zisserman 2000] using the RANSAC robust estimation algorithm [Fischler and Bolles 1981], though we note that several other approaches for rejecting outliers have been proposed (e.g. [Zhang et al. 1995; Garcia and Solanas 2004]).

Second, to account for more subtle outliers that remain and may disturb the projective-to-metric factorization process, we use a second RANSAC-type algorithm based on reprojection errors, as follows.

(1) Choose a minimal set of 8 correspondences.
(2) Obtain a metric reconstruction using these matches as described above.
(3) Triangulate the remaining points using the calibrated cameras.

(4) Calculate the reprojection error of each point (i.e. the inner term in (3) with $\Sigma = I_{2\times 2}$).

(5) Repeat (1)-(4) K times to achieve a desired probability of obtaining at least one minimal set containing only inliers.

(6) Select the set of camera parameters that gave the minimum total reprojection error.

(7) Remove points for which the reprojection error is greater than 3σ, where σ is the assumed standard deviation of the error in pixel locations.

(8) Re-estimate the calibration parameters from the set of inliers.

For our experiments with real images, we used $K = 1000$ trials, corresponding to an estimated outlier probability of 0.05 and a desired likelihood of success of 0.99.

3.5 Obtaining the Vision Graph

While our emphasis on this paper is on the distributed calibration algorithm, we cannot ignore the critical initial step of establishing the vision graph, which involves accurately detecting multi-image correspondences. This is a difficult problem in computer vision, even in the centralized case, especially when the underlying cameras are far apart. Here, we summarize an algorithm we have found to work well in practice.

First, we automatically locate a large number of feature points in each image using two methods: a multiscale Harris-Laplace detector [Mikolajczyk and Schmid 2004] (which generally finds corners), and a scale- and orientation-invariant keypoint detector [Lowe 2004] (which generally finds high-contrast blobs). Each detector is well-suited to finding features that can be reliably and robustly extracted from different views of the same scene. Typically, thousands of features are generated for each image. Each feature point is described with a 128-dimensional SIFT descriptor [Lowe 2004], formed from orientation histograms of gradient images centered at the point. This descriptor has been shown to be robust to orientation, affine distortion, and illumination changes [Mikolajczyk and Schmid 2003].

Once the features in each image are obtained, putative matches between each image pair are formed by determining, for each feature in the first image, the nearest neighbor (in the sense of Euclidean distance between descriptors) in the second image. If the descriptor-distance ratio of the nearest neighbor to that of the second-best match is below a threshold, the feature correspondence is accepted as a putative match [Lowe 2004]. This scheme reduces the number of false matches that could otherwise be generated by scenes with many repetitive structures.

Next, we perform outlier rejection as described in Section 3.4. The matches that remain after this process are extremely reliable, but there may be too few of them upon which to base the camera calibration algorithm. Accordingly, we grow additional matches based on the robustly-estimated epipolar geometry to increase the number of matches. That is, for each feature in the first image, we search along the corresponding epipolar line in the second image to find the best match. If the descriptor distance is below a threshold, the match is accepted. This feature-growing scheme typically results in an order-of-magnitude increase in the number of correct matches while keeping the number of false matches near zero.

This approach to feature correspondence is much the same as the methods described in [Hartley and Zisserman 2000; Schaffalitzky and Zisserman 2001], and it is important to note that such methods are fundamentally centralized and network-unaware. However, we have also developed a power-aware algorithm for estimating the vision graph in which each node broadcasts a fixed-length feature digest to the network to establish putative vision graph edges, followed by variable-length messaging along putative edges to refine correspondences. The details of this method will be described in a future submission, since the focus here is on calibration. An alternate method for distributed feature matching was proposed in [Avidan et al. 2004], which used a probabilistic argument based on random graphs to analyze the propagation of wide-baseline stereo matching results obtained for a small number of image pairs to the remaining cameras.

3.6 Algorithm Summary

In summary, the overall algorithm at each node operates as follows, assuming that the vision graph has been established (Section 3.5).

(1) Form a local calibration cluster from adjacent cameras in the vision graph satisfying the neighborhood sufficiency conditions (Section 3.3), i.e. each cluster must have a minimum of 3 nodes with 8 common scene points.

(2) Use RANSAC to remove outlier correspondences in the nucleus (Section 3.4).

(3) Obtain a robust metric reconstruction based on projective factorization, estimation of the dual absolute quadric, and further outlier rejection (Sections 3.2 and 3.4).

(4) Bundle adjust using the initial estimate obtained above (Section 3.2).

(5) Triangulate the non-nucleus scene points and bundle adjust over all recovered points (Section 3.2).

If the calibration process at a node fails at any point above, this node can simply check if one its neighbors in the vision graph has an estimate of the necessary parameters and borrow these. In a practical situation, calibration simply may not be possible for nodes that are unluckily pointed (e.g. at the sky or a featureless wall, or at a region not imaged by any other cameras in the network). In such cases, there may be nothing to do but wait for new overlapping cameras to join the network, or to use the unfortunate camera as a communication relay.

4. EXPERIMENTS

We studied the algorithm's performance for both simulated and real datasets, in terms of both performance on the calibration task and analysis of message-passing on the underlying communication graph.

First, we describe the calibration performance metrics. For the simulated experiments, we can directly compare the calibration parameters estimated using the distributed algorithm to ground truth. The scene points reconstructed by each camera are aligned via a similarity transformation to the ground-truth coordinate system before comparison. The error metrics between real and estimated focal

lengths, camera centers, camera orientations and scene points are computed as:

$$d(f, \hat{f}) = |f - \hat{f}|/f \quad (4)$$

$$d(C, \hat{C}) = \|C - \hat{C}\| \quad (5)$$

$$d(R, \hat{R}) = 2\sqrt{1 - \cos\theta} \quad (6)$$

$$d(X, \hat{X}) = \|X - \hat{X}\| \quad (7)$$

where θ in (6) is the relative angle of rotation between the rotation matrices R and $\hat{R}$. We can also compute the RMS image reprojection error measured in pixels, defined as

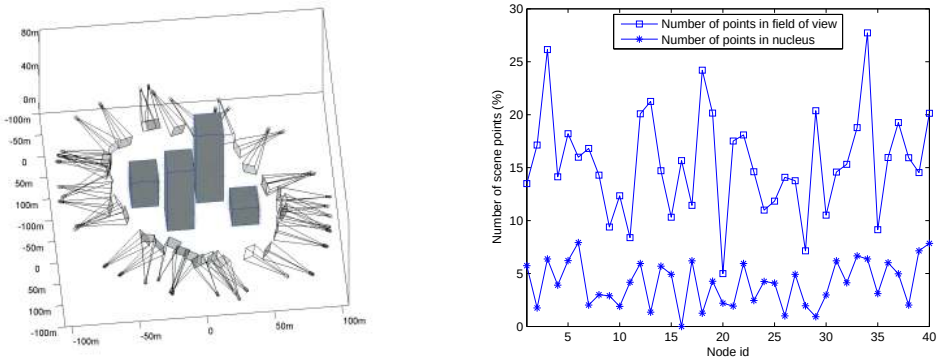
$$u_{err} = \sqrt{\frac{1}{T}\sum_j \sum_k (\hat{u}_{jk} - u_{jk})^T (\hat{u}_{jk} - u_{jk})}, \quad (8)$$

where T is the total number of matched image points. Finally, we can compute the Mahalanobis error, defined as the reprojection error divided by the estimated standard deviation of noise in the image correspondences, to give a relative measure of performance.

The same types of quantitative comparisons for the camera parameter and scene point estimates are usually difficult or infeasible to obtain for real imagery of outdoor scenes, as in our second experiment below; we can only judge the results based on reprojection errors and empirical observations of the quality of the recovered cameras/structure.

Second, we analyze the communication cost of the distributed communication scheme in terms of the number of messages that must be handled (i.e. sent or received) by each node, compared to a centralized calibration scheme. While the calibration algorithm described above passes messages along edges of the vision graph, messages are actually delivered via a series of hops along edges of the underlying communication graph. In the distributed case, each node sends a list of its features to each of its neighbors in the vision graph, which forms the input to the local calibration processes. In the centralized case, all nodes communicate their features to a designated sink node (we chose the node with maximum degree), which computes the global calibration using bundle adjustment and sends the results back to each node in the network. Since the focus here is on calibration, we do not include the cost of vision graph formation in the communication cost analysis.

We model all the nodes as equipped with identical communication systems and equal-range antennas. We say that an edge exists in the vision graph if the cameras image at least 8 common scene points, while an edge appears in the communication graph if the cameras are at most r meters apart. For simplicity we assume that the communication cost for both transmission and reception of messages is the same. For each experiment, we generated a family of communication graphs based on varying the antenna range r. We set the minimum r as the range below which the communication graph is disconnected, and increase r until the communication graph is fully connected. We calculate the shortest path between two nodes using Dijkstra's algorithm [Cormen et al. 2001], choosing randomly between all paths with the shortest length. We record the number of times each node is used either for transmission or reception of message signals during both calibration processes, since nodes connected to links that handle more messages will consume more power



(a) (b)

Fig. 3. (a) The field of view of each of the simulated cameras. Focal lengths have been exaggerated. (b) The percentage of scene points in the field of view and nucleus at each local calibration cluster.

and hence have shorter lifetimes [Estrin et al. 1999], a particularly important issue for battery-operated or otherwise power-constrained wireless sensor networks.

4.1 Simulated image experiment

First, we studied the performance of the algorithm with simulated data modeling a real-world situation. The scene consisted of 40 cameras surveying four simulated (opaque) structures of varying heights. The cameras were placed randomly on an elliptical band around the "buildings". The dimensions of the configuration were chosen to closely model a real-world configuration. The buildings had square bases $20m$ on a side and are $2m$ apart. The cameras have a pixel size of $12\mu m$, a focal length of 1000 pixels and a field of view of 600×600 pixels. The nearest camera was at $\approx 88m$ and the farthest at $\approx 110m$ from the scene center. Figure 3a illustrates the setup.

A total of 4000 scene points were uniformly distributed along the walls of the buildings and captured by the 40 cameras. Each camera images only a fraction of the total number of scene points due to the finite field of view and the assumption that the buildings are opaque (see Figure 3b). For the simulated experiment, we assume the correspondences and vision graph are known (since there are no images in which to detect correspondences). The median cluster size for this experiment was found to be 4, with a maximum size of 7. The projected points were then perturbed by zero-mean Gaussian random noise with standard deviations of 0, 0.5, 1, 1.5, and 2 pixels. The calibration was estimated for seven realizations of noise at each level, and the results are reported in Table I. Each statistic is averaged over all realizations at each noise level and over all cameras/points in the configuration.

As Table I indicates, most of the errors were very small. The cameras are localized fairly well (e.g. 45 cm error in camera centers compared to a scene width of 220m, for 1 pixel noise variance). In comparison, commercial GPS have accuracy ranges in meters [Bajaj et al. 2002]. The point reprojection error from the distributed estimate is generally slightly smaller than the standard deviation of

Table I. Summary of the calibration errors for the simulated experiment, measured using the metrics in (4)-(8). The width of the scene in the experiment was 220m.

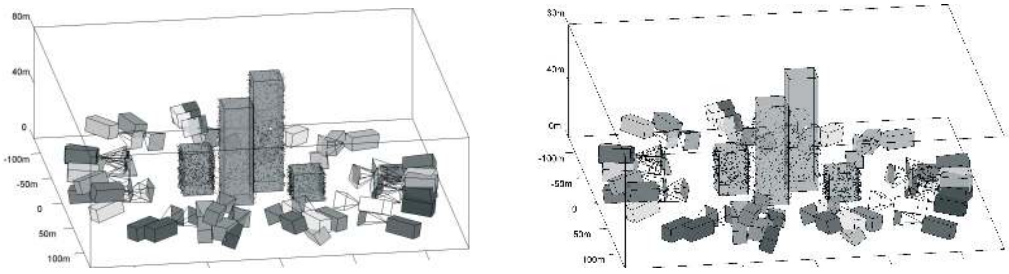
Noise σ	Method	C_{err} (cm)	R_{err}	f_{err}	X_{err} (cm)	u_{err} (pix)	*Mahal. err*	*cputime* (min)
0.5	Distributed	24.2	0.056	0.0025	21.9	0.39	0.79	30
	Centralized	20.8	0.039	0.0020	15.0	0.46	0.96	427
1	Distributed	45.3	0.075	0.0052	23.3	0.76	0.76	54
	Centralized	43.9	0.100	0.0031	20.2	0.83	0.83	433
1.5	Distributed	82.6	0.10	0.0091	25.8	1.15	0.77	83
	Centralized	75.1	0.12	0.0059	27.2	1.14	0.81	449
2	Distributed	120.1	0.12	0.012	37.9	1.53	0.76	186
	Centralized	117.9	0.14	0.010	41.9	1.11	0.80	443

the noise (i.e. the Mahalanobis error is less than 1), as desired, and this relative measure of performance is roughly constant as the noise increases.

We also compared the distributed algorithm against centralized bundle adjustment, to determine if central bundling can create a substantial improvement. To initialize the centralized estimate, we registered all the cameras' structure to a common frame, averaged the multiple estimates of each scene point, and chose the camera estimate that gave the lowest reprojection error. Importantly, Table I demonstrates that the average accuracy of the distributed algorithm is about the same as centralized bundle adjustment; for example, the error in camera center localization for the distributed case is at most 8cm more than the error in the centralized case (compared to an overall scene width of 220m). However, the distributed estimate is computed much more quickly than the centralized estimate, as evidenced by the last column in Table I, which compares the average Matlab cputime required for each method and noise level (note that the centralized timings do not include the time required to obtain the initial estimate). While the centralized bundle adjustment takes roughly the same amount of time, regardless of the noise level, the distributed algorithm is faster at lower noise levels. While the distributed timings represent the total cputime required to estimate the structure and camera parameters at all nodes, in practice, these computations would be executed in parallel by all nodes independently, so the actual time required for calibration would be substantially less than what Table I suggests. A major reason for the speed difference between the distributed and centralized algorithms is that the centralized bundle adjustment problem is an optimization over approximately 9000 parameters, whereas each local bundle adjustment problem in the distributed scheme has an average of 1589 parameters (2423 in the worst case).

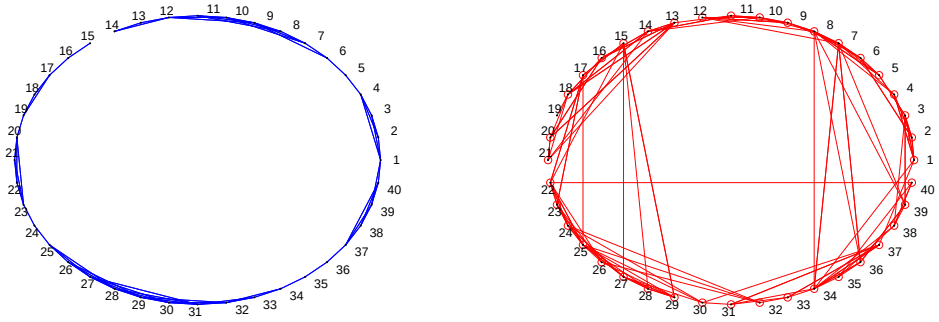
Figures 4a and 4b show the ground-truth configuration and the recovered configuration, respectively, for one noise realization with 1 pixel noise variance. The quality of the shape recovery is evident. We emphasize that the only data used to obtain this result were the positions of matching points in the images taken by the cameras.

Figure 5a illustrates the communication graph for the simulated experiment, with an antenna range of 37m (the lowest range for which the communication graph is connected). Figure 5b illustrates the vision graph. While there are many edges



(a) (b)

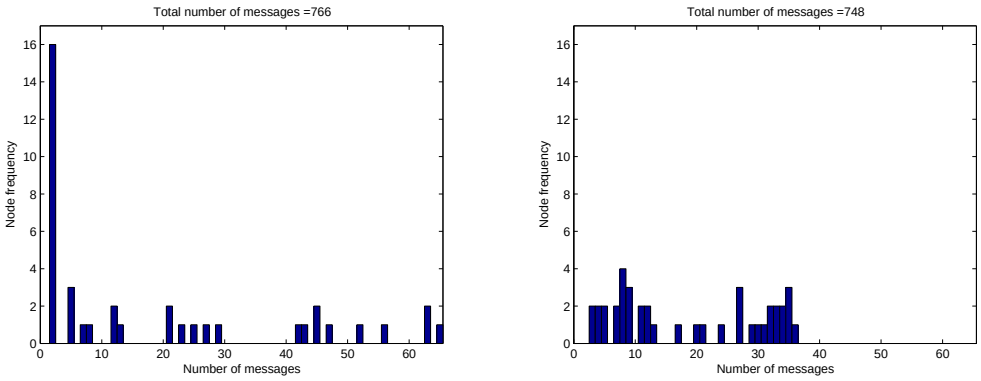
Fig. 4. (a) Ground truth structure and camera positions for the simulated experiment. (b) Recovered structure and camera positions for a noise perturbation of 1 pixel. Focal lengths have been exaggerated.



(a) (b)

Fig. 5. (a) Communication graph and (b) vision graph for the simulated experiment. The pictured communication graph was generated using an antenna range of $37m$ (the minimally connected case).

shared between the two graphs, there are also many edges in the vision graph that require multiple hops to accomplish in the communication graph. Figure 6 shows the number of messages handled at each node for the centralized (Figure 6a) and distributed (Figure 6b) calibration algorithms, for the same antenna range value of $37m$. The total number of messages handled is comparable in both cases (766 for centralized vs. 748 for distributed). However, there is a relatively fairer utilization of the nodes in the distributed algorithm compared to the centralized algorithm (i.e. the centralized message distribution has a heavier tail). Nodes that are frequently used will have lower lifetime due to the power consumed in sending and receiving messages; the failure of such nodes would be catastrophic for the centralized algorithm. On the other hand, failure of a node in the distributed case is relatively less costly, since the calibration is maintained in a distributed state across the network.



(a) (b)

Fig. 6. Histogram of messages per node in the communication graph. (a) centralized calibration vs. (b) distributed calibration in the simulated experiment, for an antenna range of 37m. The cost for sending and receiving messages is assumed to be the same. The centralized message distribution has a much heavier tail than the distributed case.

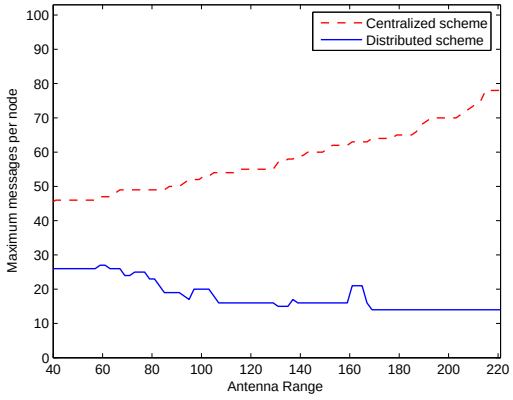


Fig. 7. Maximum number of messages handled per node as a function of increasing antenna range, in the centralized (dashed line) and distributed (solid line) cases in the simulated example. The distributed algorithm always required a smaller worst-case number of messages per node.

Figure 7 shows the maximum number of messages handled by any node in the network during the calibration algorithms as a function of increasing antenna range. In both cases, increasing the range makes the communication graph more connected. A more connected network generally places higher communication requirements on the sink node for the centralized case.[1] However, the maximum usage of any node

[1]Increasing the antenna range also increases the power required to send the messages, which we do not model here.

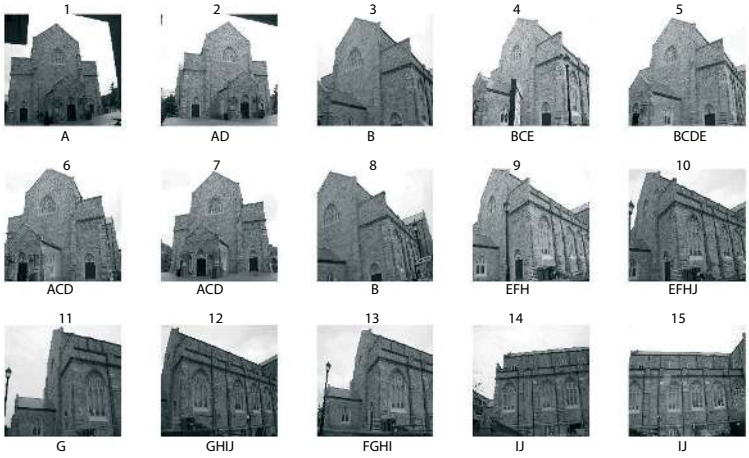


Fig. 8. The 15-image data set used for the experiment on real images. The letters A-J at the bottom of each image indicate the clusters to which the image belongs.

actually decreases in the distributed case, indicating an advantage of the distributed algorithm.

4.2 Real image experiment

We also approximated a camera network using 15 real images of a building captured by a single camera from different locations (Figure 8). The images were taken with a Canon G5 digital camera in autofocus mode (so that the focal length for each camera is different and unknown). A calibration grid was used beforehand to verify that for this camera, the skew was negligible, the principal point was at the center of the image plane, the pixels were square, and there was virtually no lens distortion. Therefore, our pinhole projection model with a diagonal K matrix is justified in this case.

For the real experiments, we used the automatic multi-image correspondence algorithm described in Section 3.5 to generate the vision graph. In this experiment, ten unique clusters were detected in the original set of fifteen images, indicated by the letters $A - J$ in Figure 8.

Figures 9 and 10 illustrate two views of the reconstructed 3D scene and camera configuration obtained from applying the distributed calibration algorithm to the 15 images in Figure 8. This result was generated by aligning each camera's reconstructed scene points to the same frame. No single camera would have full knowledge about the entire scene as shown, and each camera really only knows its location relative to its neighbors and reconstructed scene points. The quality of the structure recovery is apparent. For example, the right angles of the building faces are clearly well-estimated in Figure 9. The average Euclidean reprojection error is 0.58 pixels (compared to 0.34 pixels for centralized bundle adjustment).

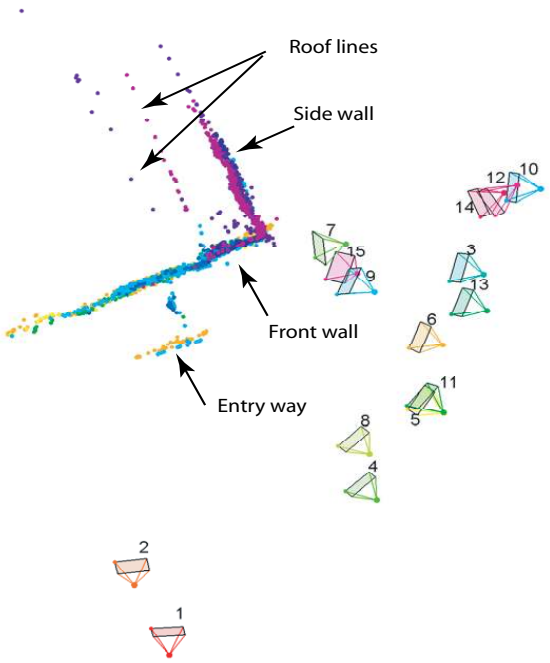
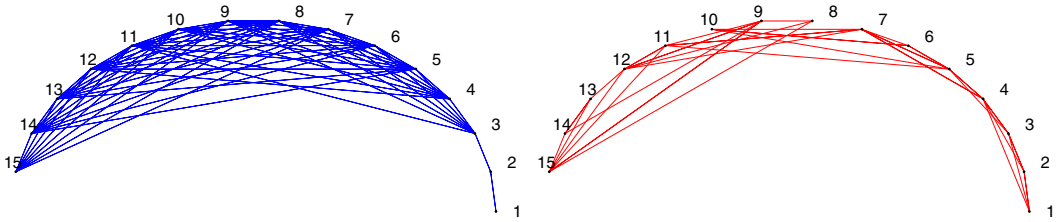


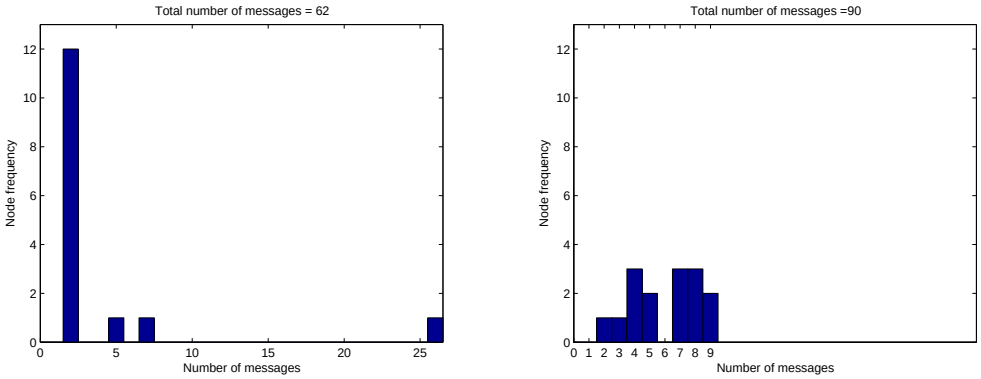
Fig. 9. Top view of the reconstructed 3D scene and camera configuration for the real experiment. The color of each scene point indicates to which of the clusters it belongs. The view illustrates that points from different clusters are correctly aligned in the final scene. Arrows indicate features of the building that are recognizable in Figure 8.

Fig. 10. Another view of the reconstructed 3D scene and camera configuration for the real experiment. Reference lines are superimposed on the image for better perception. The viewpoint is roughly similar to image 9 in Figure 8.



(a) (b)

Fig. 11. (a) Communication graph and (b) vision graph for the real experiment. The pictured communication graph was generated using a antenna range of 0.4 on a normalized scale in which the communication graph is first fully connected at range 1.0.



(a) (b)

Fig. 12. Histogram of messages per node in the communication graph for the real experiment. (a) Centralized calibration, (b) distributed calibration. The communication graph was generated using a normalized range of 0.4. As in the simulated example, the distributed algorithm makes fairer use of the nodes.

As above, we computed and analyzed the number of messages handled by each node during both calibration algorithms. Since the ground truth camera locations are unknown in this case, we built communication graphs for varying antenna ranges based on the estimated positions of the camera nodes obtained at the end of the calibration. The corresponding communication and vision graphs are shown in Figure 11. Figure 12 shows the node usage for the centralized and distributed cases. In the centralized case, the sink node is very heavily used while the rest of the nodes are used sparingly. The distributed scheme requires all the nodes to bear an approximately equal amount of communication load. Just as in the simulated experiment, the maximum number of messages handled by any node is much less for the distributed algorithm (a constant value of 9) than for the centralized one (between 25 and 27 depending on the antenna range).

5. CONCLUSIONS

We presented a distributed algorithm for the automatic, external, metric calibration of an ad-hoc network of cameras with no centralized processor. Each camera performs a local, robust bundle adjustment over the camera parameters and scene points of its vision graph neighbors, starting from an initial point obtained by projective factorization. We illustrated the accurate performance of the calibration algorithm using examples that modeled real-world camera networks, and showed that the distributed algorithm results in a fairer allocation of messages per node while achieving comparable accuracy to centralized bundle adjustment. There is still much room for future work at this intersection of computer vision and sensor network research.

It is likely that real ad-hoc camera networks (e.g. cameras that are randomly deployed over a battlefield) might have a non-negligible fraction of nodes that cannot be calibrated due to too little visual overlap with other nodes. In this case, time-of-flight or GPS-based localization schemes could be incorporated to get at least coarse estimates of such nodes' positions. Also, the smaller a local cluster in the vision graph, the higher the possibility of entering a "critical" (i.e. mathematically unviable) configuration for the calibration problem. One could study how the local cluster formation process could be adjusted for the purposes of avoiding critical configurations.

As discussed in Section 3.2, at the end of the calibration process described above, each camera has an estimate of its location and orientation relative to its neighbors. However, without additional processing, each of camera i's neighbors may have a slightly different estimate of where i is, which may adversely affect vision tasks that require very high precision. Currently, we would deal with inconsistent estimates and achieve consensus in such cases simply by averaging multiple estimates when they occur (e.g., camera i's final estimate of its location would be $\frac{1}{|\{i,\mathcal{C}_i\}|}\sum_{j\in\{i,\mathcal{C}_i\}}\hat{C}_i^j$, after the estimates have been registered by a similarity transformation). However, this is only the statistically "correct" answer when all the estimated parameters have the same covariance. Since this is never true in practice, we are developing more principled information fusion methods based on the underlying probability densities of the estimated structure-from-motion parameters. The distributed framework makes it easy to incorporate graphical message-passing models in order to achieve global consistency [Murphy et al. 1999]. The use of graphical models for similar purposes has been recently investigated in the context of robotic simultaneous location and mapping (SLAM), e.g. [Dellaert et al. 2005; Paskin and Guestrin 2004]. We note that while we expect methods based on graphical models to achieve better consensus, there may be tradeoffs with higher messaging overhead. Furthermore, the experiments in this paper indicate the agreement between nodes can already be quite good without enforcing global consensus.

To demonstrate the viability of the proposed distributed calibration scheme, the emphasis of this paper is on computer vision, and we have assumed idealized networking conditions. We plan to build more realistic network models that model the power required to send or receive a message of a given length, and investigate node lifetimes under realistic assumptions. We could further study the effects of unequal or time-varying antenna ranges or channel conditions and dynamic net-

work topologies. Since a realistic camera network is always changing, we could model calibration as a continuous, efficient background process on the network to ensure accurate performance on vision tasks. The graphical-model based approach we are investigating will naturally handle the case of moving cameras. Eventually, we plan to build wireless camera nodes to test the performance of our algorithms in real situations. We will adopt more advanced camera models (e.g. including lens distortion) as the situation warrants.

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

ACKNOWLEDGMENTS

Thanks to Dr. A. Abouzeid for valuable comments about the message-based analysis. Also, thanks to Dr. M. Pollefeys for providing projective-to-metric factorization code.

REFERENCES

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

Andersson, M. and Betsis, D. 1995. Point reconstruction from noisy images. *Journal of Mathematical Imaging and Vision 5*, 77–90.

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

Avidan, S., Moses, Y., and Moses, Y. 2004. Probabilistic multi-view correspondence in a distributed setting with no central server. In *Proceedings of the 8th European Conference on Computer Vision (ECCV)*. 428–441.

Bajaj, R., Ranaweera, S. L., and Agrawal, D. P. 2002. GPS: Location-tracking technology. *IEEE Computer 35,* 4 (April), 92–94.

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

Capkun, S., Hamdi, M., and Hubaux, J. P. 2001. GPS-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Hawaii Iternational Conference On System Sciences (HICSS '01)*.

Choi, H., Baraniuk, R., and Mantzel, W. 2004. Distributed Camera Network Localization. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*. Pacific Grove, CA.

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

Clarke, T. and Fryer, J. 1998. The development of camera calibration methods and models. *Photogrammetric Record 16,* 91 (April), 51–66.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. 2001. *Introduction to Algorithms*, 2 ed. MIT Press.

Davis, L., Borovikov, E., Cutler, R., Harwood, D., and Horprasert, T. 1999. Multi-perspective analysis of human action. In *Proceedings of the 3rd International Workshop on Cooperative Distributed Vision*. Kyoto, Japan.

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

Dellaert, F., Kipp, A., and Krauthausen, P. 2005. A multifrontal QR factorization approach to distributed inference applied to multirobot localization and mapping. In *Proceedings of National Conference on Artificial Intelligence (AAAI 05)*. 1261–1266.

Devarajan, D. and Radke, R. J. 2004. Distributed metric calibration for large camera networks. In *Proceedings of the 1st Workshop on Broadband Advanced Sensor Networks (BASENETS) 2004 (in conjunction with BroadNets 2004)*. San Jose, CA.

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

Estrin, D., Govindan, R., Heidemann, J. S., and Kumar, S. 1999. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of ACM/IEEE Conference on Mobile Computing and Networking (MobiCom 99)*. Seattle, WA, USA, 263–270.

Faugeras, O., Luong, Q.-T., and Papadopoulou, T. 2001. *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press, Cambridge, MA, USA.

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

Fitzgibbon, A. W. and Zisserman, A. 1998. Automatic camera recovery for closed or open image sequences. In *Proceedings of the 5th European Conference on Computer Vision (ECCV '98)*. Springer-Verlag, London, UK, 311–326.

Garcia, M. A. and Solanas, A. 2004. Simultaneous localization and modeling from stereo vision. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*.

Gortler, S., Grzeszczuk, R., Szeliski, R., and Cohen, M. 1996. The lumigraph. In *Proceedings of the Computer Graphics (SIGGRAPH '96)*. 43–54.

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

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

Heyden, A. and Åström, K. 1999. Flexible calibration: Minimal cases for auto-calibration. In *Proceedings of the 7th IEEE International Coference on Computer Vision (ICCV '99)*. 350–355.

Hörster, E., Lienhart, R., Kellermann, W., and Bouguet, J.-Y. 2005. Calibration of visual sensors and actuators in distributed computing platforms computing platforms. Tech. Rep. TR2005-11, University of Augsburg, Institute of Computer Science, University of Augsburg, Germany.

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

Levoy, M. et al. 2000. The Digital Michelangelo Project: 3D scanning of large statues. In *SIGGRAPH*.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60,* 2, 91–110.

Mikolajczyk, K. and Schmid, C. 2003. A performance evaluation of local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Madison, Wisconsin, 257–264.

Mikolajczyk, K. and Schmid, C. 2004. Scale and affine invariant interest point detectors. *International Journal of Computer Vision 60,* 1, 63–86.

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

Murphy, K. P., Weiss, Y., and Jordan, M. I. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*. 467–475.

Niculescu, D. and Nath, B. 2003. Trajectory based forwarding and its applications. In *Proceedings of the ACM/IEEE Conference on Mobile Computing and Networking (Mobicom'03)*. ACM Press, New York, NY, USA, 260–272.

Obraczka, K., Manduchi, R., and Garcia-Luna-Aceves, J. 2002. Managing the information flow in visual sensor networks. In *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communication (WMPC '02)*.

Paskin, M. A. and Guestrin, C. E. 2004. Robust probabilistic inference in distributed systems. In *Proceedings of the twentieth Conference on Uncertainty in Artificial Intelligence (UAI '04)*. AUAI Press, Arlington, VA, USA, 436–445.

Pollefeys, M., Koch, R., and Van Gool, L. J. 1998. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings of the 6th IEEE International Conference on Computer Vision (ICCV '98)*. 90–95.

Pollefeys, M., Verbiest, F., and Gool, L. J. V. 2002. Surviving dominant planes in uncalibrated structure and motion recovery. In *Proceedings of the 7th European Conference on Computer Vision-Part II (ECCV '02)*. Springer-Verlag, London, UK, 837–851.

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

Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. 2000. The Cricket location-support system. In *Proceedings of the 6th Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*.

Rahimi, A., Dunagan, B., and Darrell, T. 2004. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 04)*. 187–194.

Schaffalitzky, F. and Zisserman, A. 2001. Viewpoint invariant texture matching and wide baseline stereo. In *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*. Vancouver, Canada, 636–643.

Schaffalitzky, F. and Zisserman, A. 2002. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision (ECCV '02)*. Vol. LNCS 2350. 414–431. Copenhagen, Denmark.

Seo, Y., Heyden, A., and Cipolla, R. 2001. A linear iterative method for auto-calibration using DAC equation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*.

Sharp, G., Lee, S., and Wehe, D. 2002. Multiview registration of 3-D scenes by minimizing error between coordinate frames. In *Proceedings of the 7th European Conference on Computer Vision (ECCV '02)*. Vol. LNCS 2351. 587–597. Copenhagen, Denmark.

Sinha, S. N., Pollefeys, M., and McMillan, L. 2004. Camera network calibration from dynamic silhouettes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR))*. 195–202.

Sturm, P. and Triggs, B. 1996. A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the 4th European Conference on Computer Vision (ECCV '96)*. 709–720.

Teller, S., Antone, M., Bodnar, Z., Bosse, M., Coorg, S., Jethwa, M., and Master, N. 2001. Calibrated, registered images of an extended urban area. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '01)*.

Triggs, B. 1996. Factorization methods for projective structure and motion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '96)*. IEEE Comput. Soc. Press, San Francisco, CA, USA, 845–51.

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

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

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

Wu, H. and Abouzeid, A. 2004a. Energy efficient distributed JPEG2000 image compression in multihop wireless networks. In *Proceedings of 4th Workshop on Applications and Services in Wireless Networks* (August 8-11). Boston, Massachusetts, USA.

Wu, H. and Abouzeid, A. 2004b. Power aware image transmission in energy constrained wireless networks. In *Proceedings of The 9th IEEE Symposium on Computers and Communications (ISCC'2004)* (June 28-July1). Alexandria, Egypt.

Zhang, Z., Deriche, R., Faugeras, O. D., and Luong, Q.-T. 1995. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence 78,* 1-2, 87–119.

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

Received February 2005; September 2005; accepted May 2006