Self Localizing Smart Camera Networks

Babak Shirmohammadi, GRASP Laboratory, University of Pennsylvania
Camillo J. Taylor, GRASP Laboratory, University of Pennsylvania

This paper describes a novel approach to localizing networks of embedded cameras and sensors. In this scheme the cameras and the sensors are equipped with controllable light sources (either visible or infrared) which are used for signaling. Each camera node can then automatically determine the bearing to all of the nodes that are visible from its vantage point. By fusing these measurements with the measurements obtained from onboard accelerometers, the camera nodes are able to determine the relative positions and orientations of other nodes in the network.

The method is dual to other network localization techniques in that it uses angular measurements derived from images rather than range measurements derived from time of flight or signal attenuation. The scheme can be implemented relatively easily with commonly available components and scales well since the localization calculations exploit the sparse structure of the system of measurements. Further, the method provides estimates of camera orientation which cannot be determined solely from range measurements.

The localization technology can serve as a basic capability on which higher level applications can be built. The method could be used to automatically survey the locations of sensors of interest, to implement distributed surveillance systems or to analyze the structure of a scene based on the images obtained from multiple registered vantage points. It also provides a mechanism for integrating the imagery obtained from the cameras with the measurements obtained from distributed sensors.

Categories and Subject Descriptors: I.4.8 [**Image Processing**]: Scene Analysis; C.3 [**Special-Purpose and Application-Based Systems**]: Miscellaneous

General Terms: Location infrastructure establishment services

Additional Key Words and Phrases: Smart Cameras, Localization, Optical Signaling, Bundle Adjustment

ACM Reference Format:
Shirmohammadi, B., Taylor, C. Self Localizing Smart Cameras ACM Trans. Embedd. Comput. Syst. 8, 2, Article 39 (August 2011), 26 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION AND RELATED WORK

As the prices of cameras and computing elements continue to fall, it has become increasingly attractive to consider the deployment of smart camera networks. Such networks would be composed of small, networked computers equipped with inexpensive image sensors. These camera networks could be used to support a wide variety of applications including environmental modeling, 3D model construction and surveillance. For example, in the near future it will be possible to deploy small, unobtrusive smart cameras in the same way that one deploys lightbulbs, providing ubiquitous coverage of extended areas. We could imagine using such a system to track passengers at an airport from the time that they arrive at curbside check in to the time that they board their flight.

Author's address: Computer and Information Science Dept., 3330 Walnut St, Philadelphia PA 19104-6389.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

A number of research efforts at a variety of institutions are currently directed toward realizing aspects of this vision. The Cyclops project at the Center for Embedded Networked Sensing (CENS) has developed small low power camera modules and has applied them to various types of environmental monitoring applications [Rahimi et al. 2005]. Kulkarni et al. describe the SensEye system which provides a tiered architecture for multi camera applications [Kulkarni et al. 2005]. Hengstler and Aghajan describe a smart camera mote architecture for distributed surveillance [Hengstler and Aghajan 2006]. The Panoptes system at the Oregon Graduate Institute [Feng et al. 2005] and the IrisNet project at Intel Research [Nath et al. 2002; Nath et al. 2002; Gibbons et al. 2003] both seek to demonstrate applications based on networks of commercial off-the-shelf web cameras. Bhattacharya, Wolf and Chellapa have also investigated the design and utilization of custom smart camera modules under the aegis of the Distributed Smart Camera Project [Yue et al. 2003; Lin et al. 2004].

One critical problem that must be addressed before such systems can be deployed effectively is that of localization. That is, in order to take full advantage of the images gathered from multiple vantage points it is helpful to know how the cameras in the scene are positioned and oriented with respect to each other.

In this paper we describe a novel deployment scheme where each of the smart cameras is equipped with a colocated controllable light source which it can use to signal other smart cameras in the vicinity. By analyzing the images that it acquires over time, each smart camera is able to locate and identify other nodes in the scene. This arrangement makes it possible to directly determine the epipolar geometry of the camera system from image measurements and, hence, provides a means for recovering the relative positions and orientations of the smart camera nodes.

Much of the work on localization in the context of sensor networks has concentrated on the use of time of flight or signal strength measurements of radio or audio transmissions [Bulusu et al. 2004; Moore et al. 2004; Newman and Leonard 2003]. The Cricket ranging system developed at MIT is one example of such an approach. Image measurements derived from the envisioned smart camera systems would provide a complementary source of information about angles which can be used in conjunction with the range measurements to better localize sensor ensembles.

There has been a tremendous amount of work in the computer vision community on the problem of recovering the position and orientation of a set of cameras based on images. Snavely, Seitz and Szeliski [Snavely et al. 2006] describe an impressive system for recovering the relative orientation of multiple snapshots using feature correspondences. This work builds on decades of research on feature extraction, feature matching and bundle adjustment. An excellent review of these methods can be found in [Hartley and Zisserman 2003]. These techniques typically work in a batch fashion and require all of the imagery to be sent to a central location for processing.

Antone et al. [Antone and Teller 2002] and Sinha et al. [Sinha and Pollefeys 2006] both describe schemes for calibrating collections of cameras distributed throughout a scene. Sinha et al. [Sinha and Pollefeys 2006] discuss effective approaches to recovering the intrinsic parameters of a pan tilt zoom camera while Antone et al. [Antone and Teller 2002] discuss approaches that leverage the rectilinear structure of buildings to simplify the localization procedure.

Devarajan et al. [Devarajan et al. 2006; Devarajan et al. 2008; Cheng et al. 2007] describe an interesting scheme which distributes the correspondence establishment and bundle adjustment process among the cameras. The scheme involves having the cameras communicate amongst themselves to detect regions of overlap. This approach can be very effective when sufficient correspondences are available between the frames.

Recently two interesting algorithms have been proposed which address the smart camera localization problem using distributed, consensus style schemes driven by mes-

sage passing. Piovan et al. [Piovan et al. 2008] describe a scheme for recovering the relative orientation of a set of cameras in the plane. Their method converges over time to an estimate that is close to the global least squares estimate. Tron and Vidal describe a scheme that recovers the position and orientation of a set of cameras in 3D. Their method relies on standard algorithms from computer vision that are employed to recover the relative position and orientation of pairs of cameras based on correspondences between the two images. Their approach uses a series of message passing steps to recover an estimate for the relative orientation of the cameras, then another set of message passing steps to recover the translation between the cameras. Lastly the pose estimates are refined by a final set of iterations which adjust both the position and orientation of the nodes. Both methods are effectively distributed forms of gradient descent which seek to optimize agreement between the predicted image measurements and the observed values at each node. In contrast, in the method proposed in this manuscript the nodes that perform the localization procedure collect the sighting measurements from all of the nodes they wish to localize and run a computation to determine the relative configuration of the ensemble.

Several researchers have developed algorithms to discover spatio-temporal correspondences between two unsynchronized image sequences [Tuytelaars and Gool 2004; Caspi et al. 2006; Wolf and Zomet 2002; Carceroni et al. 2004]. Once these correspondences have been recovered, it is often possible to recover the epipolar geometry of the camera system. The idea of using correspondences between tracked objects to calibrate networks of smart cameras has also been explored by Rahimi et al. [Ali Rahimi and Darrell 2004] and by Funiak et al.[Funiak et al. 2006]. These approaches can be very effective when the system can discover a sufficient number of corresponding tracks.

Another interesting approach to smart camera localization has been presented by Sinha, Pollefeys and McMillan [Sinha et al. 2004] who describe a scheme for calibrating a set of synchronized cameras based on measurements derived from the silhouettes of figures moving in the scene.

The scheme described in this paper avoids the problem of finding corresponding features between frames by exploiting active lighting which provides unambiguous correspondence information and allows us to recover the relative orientation of the cameras from fewer image measurements. Early versions of the proposed scheme were described in [Taylor 2004; Taylor and Cekander 2005] subsequent works that built on these ideas were presented in [Taylor and Shirmohammadi 2006] and in [Barton-Sweeney et al. 2006]. The concepts were also adapted for use on small self assembling mobile robots as discussed in [Shirmohammadi et al. 2007].

This paper describes a novel variant of the scheme which leverages the measurements from three axis accelerometers onboard the cameras. These measurements allow the cameras to gauge their orientation with respect to gravity and greatly simplify the problem of recovering the relative orientation of the cameras. Once the camera orientations have been estimated, the localization problem is effectively reduced to the problem of solving a sparse system of linear equations. A subsequent, optional bundle adjustment stage can be employed to further refine the position estimates. Here again we show how one can exploit the sparse structure of the measurement system and perform this optimization efficiently even on networks involving hundreds of cameras.

Importantly, the proposed scheme allows us to develop smart camera systems that can be deployed and calibrated in an ad-hoc fashion without requiring a time consuming manual surveying operation.

2. TECHNICAL APPROACH

Figure 1 illustrates the basic elements of our vision based localization system. In this localization scheme each of the embedded camera systems is equipped with a con-

trollable light source, typically an infrared Light Emitting Diode (LED), a three-axis accelerometer and a wireless communication system. Each smart camera uses its signaling LED as a blinker to transmit a temporally coded sequence which serves as a unique identifier. The cameras detect other nodes in their field of view by analyzing image sequences to detect blinking pixels and, hence, are able to determine the relative bearing to other visible nodes. Figure 1 shows the simplest situation in which two nodes can see each other. Here we note that the accelerometer measurements provide another independent source of information about the orientation of the cameras with respect to the vertical axis. These measurements allow two smart cameras to determine their relative position and orientation up to a scale factor. When a collection of smart cameras is deployed in an environment, these visibility relationships induce a sparse graph among the cameras as shown in Figure 2. These measurements can be used to localize the entire network. The scheme provides a fast, reliable method for automatically localizing large ensembles of smart camera systems that are deployed in an ad-hoc manner.

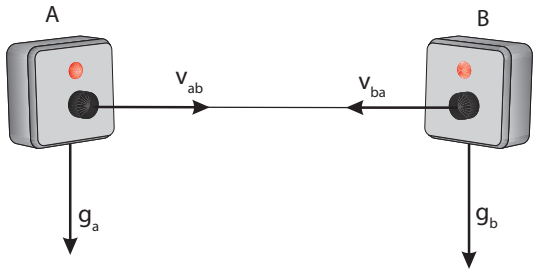

Fig. 1. This figure shows the basic elements of the proposed localization scheme. It depicts two smart camera nodes equipped with controllable light sources and accelerometers. The camera nodes are able to detect and identify other nodes in the scene by analyzing their video imagery. They can then determine their relative position and orientation up to a scale from the available measurements. Larger networks can be localized by leveraging this relative localization capability.

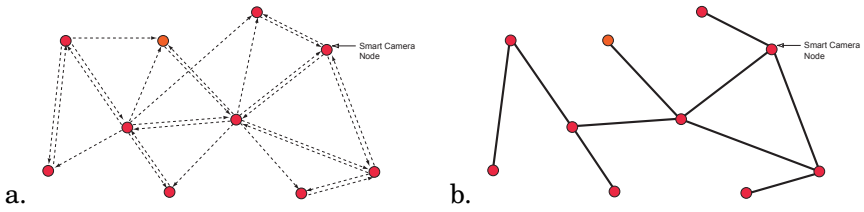

Fig. 2. The visibility relationships between the nodes can be represented with a directed graph as shown on the left. If we consider only pairs of nodes that are mutually visible we end up with the undirected variant shown on the right.

One advantage of the proposed localization scheme is that it can also be used to detect and localize other smaller, cheaper sensor nodes that are simply outfitted with blinking LEDs. Figure 4 shows the result of localizing a constellation of 4 smart cameras and 3 blinker nodes. The ability to automatically survey the locations of a set of sensor motes distributed throughout a scene could be used to enable a variety of application. We could imagine, for example, using the smart camera system to localize a

set of audio sensors in an environment. Once this has been accomplished the signals from the microphone sensors could be correlated to localize sound sources in the scene as was done by Simon et al. [Simon et al. 2004]. The locations of these sound sources could then be related to the images acquired by the cameras so that appropriate views of the sound source could be relayed to the user.

Various components of the proposed localization scheme are described in more detail in the following subsections.

2.1. Blinker Detection

In the first stage of the localization process, the nodes signal their presence by blinking their lights in a preset pattern. That is, each of the nodes would be assigned a unique string representing a blink pattern such as 10110101, the node would then turn its light on or off in the manner prescribed by its string. Similar temporal coding schemes are employed in laser target designators and freespace optical communication schemes.[1]

The blink patterns provide a means for each of the camera equipped nodes to locate other visible nodes in their field of view. They do this by analyzing the images to locate pixels whose intensity varies in an appropriate manner. This approach offers a number of important advantages, firstly it allows the node to localize and identify neighboring nodes since the blink patterns are individualized. Secondly, it allows the system to reliably detect nodes that subtend only a few pixels in the image which allows for further miniaturization of the camera and sensor nodes.

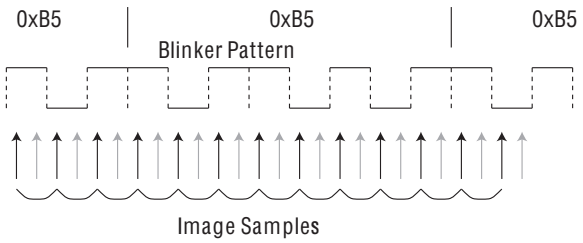

Fig. 3. The optical intensity signal in the imager is sampled at twice the bit period which ensures that either the odd or even sample set will correctly sample the message regardless of the offset between the sampling and encoding clocks.

Figure 3 depicts the timing of the blink pattern and the image acquisition process. As described earlier, the blinkers continuously repeat a prescribed bit sequence at a fixed frequency. In the current implementation, this blinking function is carried out on each node by a microcontroller based subsystem which controls an LED array.

Our current optical detection scheme *does not* seek to synchronize the image acquisition process on the cameras with the blinkers. This implementation decision significantly reduces the complexity of the system and the amount of network traffic required.

In our detection scheme we assume that the exposure time of the images is small compared with the bit period of the optical signal being transmitted. For example in our current implementation the bit period is (1/6)th of a second while the exposure

[1]One could argue that freespace optical communication dates back to classical antiquity when the invading Greeks signaled to their hidden fleet using torches once they had successfully breached the gates of Troy.

time of each camera is approximately 10 microseconds. This means that each pixel in the camera effectively functions as a sample and hold circuit sampling the value of the intensity signal at discrete intervals. In general, if we were to sample a binary signal with a sampling comb of the same frequency it would correctly reproduce the binary signal on almost every occasion, the only exception being when the sampling comb happens to be aligned with the transitions in the binary signal. In that case since the samples are being taken while the input signal is transitioning between a high value and a low value, the resulting sample can take on any intermediate value and the decoded result will typically not correspond to a valid code. This problem can be overcome by sampling the signal at twice the bit encoding frequency. We can then divide the samples into two sets corresponding to odd and even numbered samples as shown in Figure 3 and can guarantee that at least one of these sets correctly samples the binary signal.

More specifically if the even set of samples happens to be aligned with the bit transitions we can be sure that the odd samples which are offset by precisely half a bit period will safely sample the middle of each bit and vice versa. That is, while one or the other of the sets of samples may be corrupted by an accidental alignment with the bit transitions at least one set of samples must sample the bit pattern cleanly.

In our implementation, as each new image is obtained it is compared with the previous odd or even frame, each pixels intensity measurement is compared with its previous value and if it has changed by more than a specified amount we push a 1 bit on a shift register associated with that pixel otherwise we push a 0 bit. By testing the change in intensity values between frames rather than the intensity values themselves we avoid setting an absolute threshold on intensity values which makes our implementation more robust to varying illumination conditions.

As an example, if the system observed the following sequence of intensity values from a given pixel { 108, 110, 113, 70, 20, 68, 98, 58, 18, 60, 105, 108, 112, 55, 12, 54, 100, 101 }, it would produce the following 8 bit values from the odd and even samples respectively 01111011 , 00000100 assuming that the sample indices start at 1 and a threshold value of 50 is used to test the changes between consecutive samples. These 8 bit patterns are then compared to the transition patterns that would result from the signal patterns that the smart camera is interested in detecting to see if a match exists. For example a blink code of 0xB5 would produce the following pattern of transitions 11011110. This decoding can be accomplished simply and efficiently by using a lookup table. Note that since the samples can start at any point in the sequence the bit transitions can correspond to any cyclic permutation of the pattern of interest. Similarly negating a given pattern produces the same sequence of transitions in the intensity measurements only inverted. These issues are easily handled by appropriately tagging all equivalent patterns of transitions in the lookup table with the same base code. In the example above, the transition pattern 01111011 would map to the code 0xB5 in the lookup table. Because of these equivalences, the number of unique codes that this recognition scheme can distinguish is on the order of $\left(\frac{2^{(n-1)}}{n}\right)$ where n denotes the number of bits in the code.

After eight even or odd samples have been acquired each additional frame adds another transition which can be used to confirm the presence of a detected code. That is, given a complete set of 8 samples one can predict what the next transition will be if the pixel is in fact exhibiting the suspected blink pattern. This means that the system can confirm the presence of a blink code by monitoring the pixel over a specified number of samples to be sure of its identity. For example if a code of 0xB5 is detected at a particular pixel based on either the odd or the even samples. The system would monitor that location for an additional 20 frames which would provide 10 more odd

or even samples which should follow the proscribed pattern before the detection is confirmed. In practice this simply means that the transition patterns detected at the pixel in question should be mapped by the lookup table to the same target code over an extended sequence of frames. This is a very effective approach for removing false detections caused by spurious sampling alignments since these false detections do not recur reliably.

More sophisticated decoding schemes are certainly possible. One could, for example imagine a coding scheme which used a unique preamble to delineate the start of the bit sequence. The advantage of the scheme described here is the fact that it is amenable to real time implementation using straightforward per-pixel operations. With our current system we are able to process and decode 3 Mpix images at 12 frames per second on an embedded processor. Note that this scheme returns all of the relevant blinker patterns detected in the image so the camera can simultaneously detect multiple targets. Figure 4 shows the results of the blinker detection phase on a typical image. Here the detected locations in the image are labeled with the unique codes that the system found.

Once the blinkers have been detected and localized in the images, we can derive the unit vectors, $\mathbf{v}_{ab}$ and $\mathbf{v}_{ba}$, that relate the nodes as shown in Figure 1. Here we assume that the intrinsic parameters of each camera (focal length, principal point, distortion coefficients) have been determined in a previous calibration stage. These parameters allow us to relate locations in the image to direction vectors relative to the camera frame.

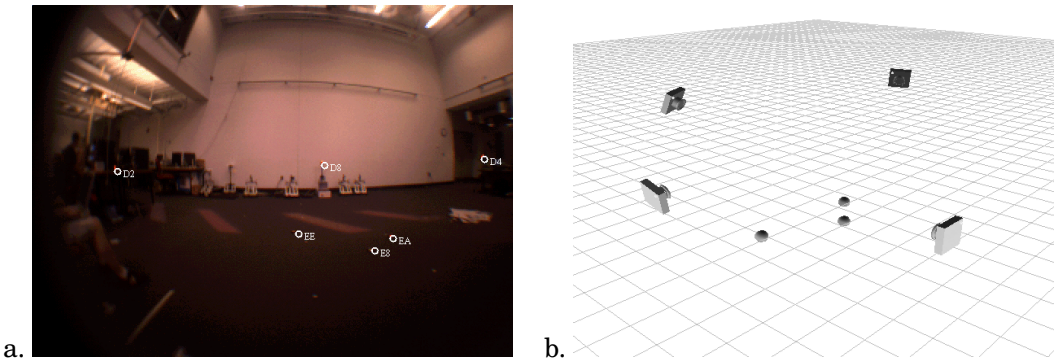

Fig. 4. This figure shows the results of automatically localizing a constellation of 4 smart cameras and 3 blinker nodes. The image obtained from one of the smart cameras is shown in (a) while the localization results are shown in (b).

2.2. Recovering Orientation

Each of the smart camera nodes is equipped with an accelerometer which it can use to gauge its orientation with respect to gravity. More specifically given a unit vector $\mathbf{g}_C$ denoting the measured gravity vector in the cameras frame of reference we can construct an orthonormal rotation matrix $R_{CW} \in SO(3)$ which captures the relative orientation between the cameras frame of reference denoted by C, and a local gravity referenced frame centered at the camera denoted by W where the z axis points upwards as shown in Figure 5.

From the vector $\mathbf{g}_C$ we can derive a second vector $\mathbf{n}_C$ which represents a normalized version of $(\mathbf{e}_x \times \mathbf{g}_C)$ where $\mathbf{e}_x$ denotes the unit vector along the x-axis, that is $\mathbf{e}_x = (1, 0, 0)^T$. We use the vector $\mathbf{n}_C$ to define the y axis of the gravity reference frame, $\mathbf{y}_W$,

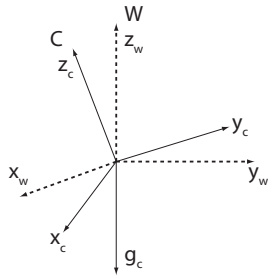

Fig. 5. Each smart camera uses the measurements from an onboard accelerometer to gauge its orientation with respect to gravity.

in Figure 5. From the two perpendicular unit vectors, $\mathbf{g}_C$ and $\mathbf{n}_C$, we can construct the rotation matrix $R_{CW} \in SO(3)$ as follows: $R_{CW} = [\, (\mathbf{g}_C \times \mathbf{n}_C) \;\; \mathbf{n}_C \;\; -\mathbf{g}_C \,]$. Note that the columns of R_{CW} correspond to the coordinates of the x, y and z axes of the world frame in the cameras frame of reference. These equations can easily be modified in situations where the gravity vector is aligned with the x-axis.

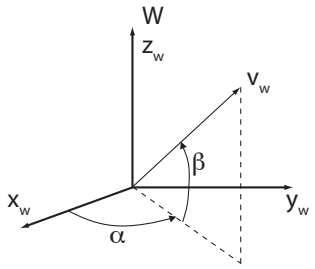

Fig. 6. Sighting vectors in each camera can be transformed to a local, gravity referenced frame and represented in terms of azimuth, α, and elevation, β, angles.

This rotation matrix can be used to transform the sighting vectors recovered in the camera frame into the local gravity referenced world frame where they can be conveniently represented in terms of azimuth and elevation angles, α and β, as shown in Figure 6. More specifically, once a blinker has been detected in the image, one can use its position in the frame along with the intrinsic parameters of the camera, which are recovered in a prior calibration phase, to compute a 3D vector, $\mathbf{v}_C$, which represents the ray from the center of projection of the camera to the blinker [2]. The rotation matrix R_{CW} can then be used to transform this vector from the cameras frame of reference to the local gravity referenced frame as follows: $\mathbf{v}_W = R_{CW}^T \mathbf{v}_C$. Equation 1 shows how the resulting vector, depicted in Figure 6, is related to the azimuth and elevation angles, α and β.

[2]This is a standard operation in many Computer Vision codes and one can find a thorough description of the procedure by consulting the Matlab Calibration Toolbox which is freely available online. See also [Heikkila and Silven 1997].

$$\mathbf{v}_W = \begin{pmatrix} \mathbf{v}_W^X \\ \mathbf{v}_W^Y \\ \mathbf{v}_W^Z \end{pmatrix} \propto \begin{pmatrix} \cos\beta\cos\alpha \\ \cos\beta\sin\alpha \\ \sin\beta \end{pmatrix} \tag{1}$$

These equations allow us to recover the azimuth and elevation angles from the components of the vector $\mathbf{v}_W$ as follows: $\alpha = \text{atan2}(\mathbf{v}_W^Y, \mathbf{v}_W^X)$, $\beta = \text{asin}(\mathbf{v}_W^Z)$. This change of coordinates simplifies the overall localization problem since we can use the azimuth angle measurements to localize the nodes in the horizontal plane and then recover the vertical displacements between the nodes using the elevation angles in a second phase.

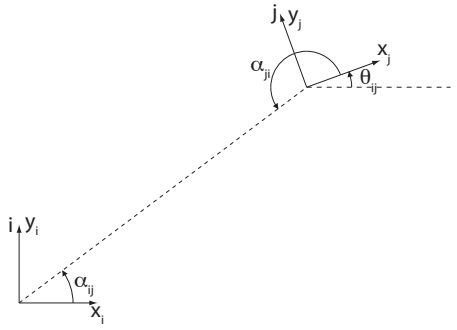

Fig. 7. The relative yaw angle between two camera frames in the plane can easily be recovered from the measured azimuth angles if the cameras can see each other.

While we can construct a gravity referenced frame for each of the cameras from the accelerometer measurements, the relative yaw between these frames is initially unknown. However, when two smart cameras can see each other as depicted in Figure 7 it is a simple matter to estimate their relative orientation from the available azimuthal measurements α_{ij} and α_{ji} which are related by the following equation.

$$\alpha_{ji} = \alpha_{ij} - \theta_{ij} + \pi \tag{2}$$

Here the parameter θ_{ij} captures the yaw angle of camera frame j with respect to camera frame i as shown in Figure 7.

More generally, the visibility relationships between the smart camera nodes can be captured in terms of a directed graph where an edge between nodes i and j indicates that node i can measure the bearing to node j as shown in Figure 2. Any smart camera node can construct such a graph by querying its neighbors for their sighting measurements. From this directed visibility graph we can construct an undirected variant where two nodes are connected if and only if they can see one another. If there is a path between two nodes in this undirected graph, they can determine their relative orientation. This allows any smart camera node to estimate the relative orientation of its neighbors via a simple breadth first labeling.

Once this has been done, all of the bearing angles can be referenced to a single frame of reference, that of the root node. What remains then is to determine the position of the nodes relative to the root. This can be accomplished by concatenating all of the available azimuthal measurements into a single homogenous linear system which can be solved using singular value decomposition (SVD).

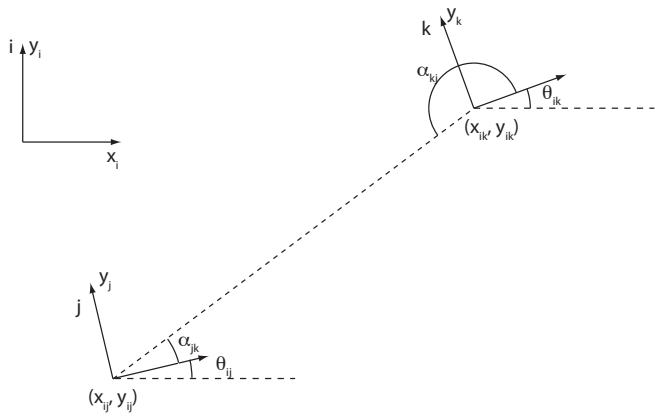

Fig. 8. In this figure the positions and orientations of the cameras j and k are referenced to the root node, camera i. The bearing measurements α_{jk} and α_{kj} induce linear constraints on the coordinates (x_{ij}, y_{ij}) and (x_{ik}, y_{ik}) .

Consider the situation shown in Figure 8 where node j measures the relative bearing to node k. Since we have already recovered the relative orientation between camera frame j and the root camera node i, θ_{ij}, each bearing measurement induces a homogenous linear equation in the unknown coordinates of the following form.

$$(x_{ik} - x_{ij})\sin(\alpha_{jk} + \theta_{ij}) - (y_{ik} - y_{ij})\cos(\alpha_{jk} + \theta_{ij}) = 0 \tag{3}$$

Here (x_{ij}, y_{ij}) and (x_{ik}, y_{ik}) denote the coordinates of nodes j and k with respect to camera frame i. The collection of homogenous linear equations can be aggregated into a row sparse system of the form $A\mathbf{p} = \mathbf{0}$ where p is a vector with $2n$ entries formed by concatenating the coordinates of the n camera frames with respect to the root, $\mathbf{p} = (x_{i1}, y_{i1}, x_{i2}, y_{i2}, ..., x_{in}, y_{in})^T$. The matrix A will have one row for each bearing measurement.

Singular value decomposition can be employed to find the null space of the matrix A. More specifically, it can be used to find the vector corresponding to the minimal singular value of A or the minimum eigenvalue of $A^T A$. Because of the sparse structure, such problems can be solved efficiently using modern matrix codes even for systems involving hundreds of cameras [Golub and Loan 1996]. This approach subsumes and improves upon earlier approaches to localizing larger collections of cameras based on repeated triangulation [Taylor and Shirmohammadi 2006]. If the structure of the network cannot be completely determined from the available measurements the dimension of the null space of A will be two or more. This can be detected by considering the ratio between the smallest and second smallest singular values.

Since this linear system is homogenous we can only resolve the configuration of the nodes up to a positive scale factor. In other words, the camera systems provide us with angular measurements which allow us to perform localization via triangulation. They do not provide distance measurements directly so the overall scale of the reconstruction is undetermined. This ambiguity can be resolved with a single distance measurement, that is, knowing the distance between any two nodes in the network determines the scale of the entire constellation.

If additional position measurements are available for some of the nodes, via GPS or a prior survey, such information can easily be incorporated into the localization process. For example if (x_j^w, y_j^w) denote the easting and northing GPS coordinates of node j we

can add the following two equations which relate the coordinates recovered from the homogenous system to the GPS measurements.

$$x_j^w = \lambda(x_{ij}\cos\gamma - y_{ij}\sin\gamma) + t_x \quad (4)$$

$$y_j^w = \lambda(x_{ij}\sin\gamma + y_{ij}\cos\gamma) + t_y \quad (5)$$

Where t_x, t_y and γ denote the position and orientation of the root node with respect to the geodetic frame of reference and λ denotes the overall scale parameter that relates the two frames. If we let $c = \lambda\cos\gamma$ and $s = \lambda\sin\gamma$. We end up with two linear equations in the unknowns c, s, t_x and t_y. Given two or more such GPS measurements one can solve the resulting linear system to recover these unknown parameters and, hence, recover the geodetic locations of all of the nodes in the system.

$$x_j^w = cx_{ij} - sy_{ij} + t_x \quad (6)$$

$$y_j^w = sx_{ij} + cy_{ij} + t_y \quad (7)$$

2.3. Recovering Vertical Displacements

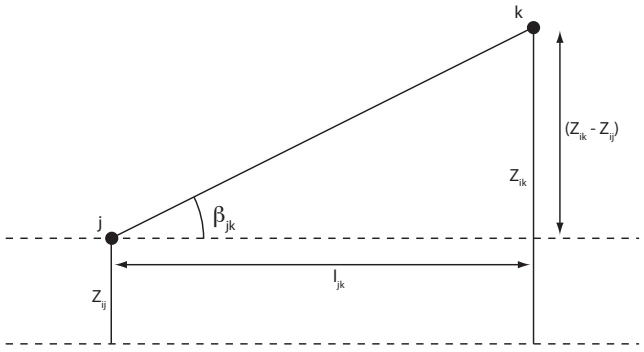

Fig. 9. The elevation measurements induce a linear constraint on the relative heights of the nodes once the locations in the plane have been estimated.

Once the (x, y) locations of the nodes in the horizontal planes have been estimated, it is a simple matter to recover the relative heights of the nodes. Figure 9 shows how an elevation measurement, β_{jk}, relates the heights of two nodes z_{ij} and z_{ik}. From each such elevation measurement one can construct a linear equation.

$$\frac{z_{ik} - z_{ij}}{\sqrt{(x_{ik} - x_{ij})^2 + (y_{ik} - y_{ij})^2}} = \tan\beta_{jk} \quad (8)$$

These constraint equations can be aggregated into a single linear system of the form $B\mathbf{z} = \mathbf{c}$ where the vector $\mathbf{z}$ represents the aggregates of all of the unknown vertical coordinates, $\mathbf{z} = (z_{i1}, z_{i2}, ..., z_{in})^T$. Once again the root node i defines the origin so its z coordinate is 0.

2.4. Refining Pose Estimates

If necessary, the estimates for node position and orientation produced by the linear process described in the preceding sections can be further refined. In this refinement step the localization process is recast as an optimization problem where the objective is to minimize the discrepancy between the observed image measurements and the measurements that would be predicted based on the estimate for the relative positions and orientations of the sensors and cameras. This process is referred to as Bundle Adjustment in the computer vision and photogrammetry literature [Hartley and Zisserman 2003].

In the sequel we will let $\mathbf{u}_{jk} \in \mathbb{R}^3$ denote the unit vector corresponding to the measurement for the bearing of sensor k with respect to camera j. This measurement is assumed to be corrupted with noise. The vector $\mathbf{v}_{jk} \in \mathbb{R}^3$ corresponds to the predicted value for this direction vector based on the current estimates for the positions and orientations of the sensors. This vector can be calculated as follows:

$$\mathbf{v}_{jk} = R_{ij}(\mathbf{t}_{ik} - \mathbf{t}_{ij}) \tag{9}$$

In this expression $R_{ij} \in SO(3)$ denotes the rotation matrix which relates camera frame j to the root frame i while $\mathbf{t}_{ik}, \mathbf{t}_{ij} \in \mathbb{R}^3$ denote the positions of nodes j and k relative to node i.

The goal then is to select the camera rotations and sensor positions so as to minimize the discrepancy between the vectors $\mathbf{u}_{jk}$ and $\mathbf{v}_{jk}$ for every available measurement. In equation 10 this discrepancy is captured by the objective function $\mathcal{O}(\mathbf{x})$ where $\mathbf{x}$ denotes a vector consisting of all of the rotation and translation parameters that are being estimated.

$$\mathcal{O}(\mathbf{x}) = \sum_{\mathbf{i,j}} \left\| \mathbf{u_{ij}} - \frac{\mathbf{v_{ij}}}{\|\mathbf{v_{ij}}\|} \right\|^{\mathbf{2}} \tag{10}$$

Problems of this sort can be solved very effectively using variants of Newton's method. In these schemes the objective function is locally approximated by a quadratic form constructed from the Jacobian and Hessian of the objective function

$$\mathcal{O}(\mathbf{x} + \delta\mathbf{x}) \approx \mathcal{O}(\mathbf{x}) + (\nabla\mathcal{O}(\mathbf{x}))^{\mathbf{T}}\delta\mathbf{x} + \frac{\mathbf{1}}{\mathbf{2}}\delta\mathbf{x}^{\mathbf{T}}(\nabla^{\mathbf{2}}\mathcal{O}(\mathbf{x}))\delta\mathbf{x} \tag{11}$$

At each step of the Newton algorithm we attempt to find a step parameter in the space, δx, that will minimize the overall objective function by solving a linear equation of the form.

$$\delta\mathbf{x} = -(\nabla^{\mathbf{2}}\mathcal{O}(\mathbf{x}))(\nabla\mathcal{O}(\mathbf{x})) \tag{12}$$

Here we can take advantage of the fact that the linear system described in equation 12 is typically quite sparse. More specifically, the Hessian matrix $\nabla^2\mathcal{O}$ will reflect the structure of the visibility graph of the sensor ensemble. This can be seen by noting that the variables corresponding to the positions of nodes j and k only interact in the objective function if node j observes node k or vice versa. For most practical deployments, the visibility graph is very sparse since any given camera typically sees a relatively small number of nodes as depicted in Figure 2. This means that the computational effort required to carry out the pose refinement step remains manageable even when we consider systems containing several hundred cameras and sensor nodes.

The optimization problem given in Equation 10 can be further simplified by restricting the problem to recovering the relative positions of the camera in the horizontal

plane. This can be accomplished simply by projecting the bearing measurements into the plane perpendicular to the gravitational vector. In this case Equation 9 would be modified, the rotation matrix R_{ij} would be an element of $SO(2)$ and the vectors $\mathbf{t}_{ik}, \mathbf{t}_{ij}$ and $\mathbf{v}_{jk}$ would be in $\mathbb{R}^2$.

2.5. Scaling Up

The proposed linear and non-linear localization schemes which exploit the sparse structure of the relevant matrices can be used to localize hundreds of nodes at a time. However, when we consider networks of the future we may ultimately want to handle systems that cover extended areas such as the airport scenario mentioned in the introduction. Such systems may involve thousands of camera and sensor nodes which are added and removed continuously. Here it may not be feasible or desirable to have each node recover its position with respect to every other node in the ensemble. The proposed scheme can be employed to allow each smart camera node to estimate its position with respect to all of its neighbors within a specified radius. Each node would then have an estimate for the configuration of a subset of the total ensemble. This is however, sufficient to allow all of the nodes to agree on locations of salient objects via a process of coordinate transformation.

Consider a situation where camera node j wants to inform its neighbor k of the coordinates of some event. Let $R_{jk} \in SO(3)$ and $\mathbf{t}_{jk} \in \mathbb{R}^3$ denote the estimates for the position and orientation of node k with respect to node j which is maintained by node j. Similarly let $R_{kj} \in SO(3)$ and $\mathbf{t}_{kj} \in \mathbb{R}^3$ denote node k's estimate for the relative position of node j. Let l_{jk} denote the distance between j and k in node j's frame of reference while l_{kj} denotes the length of the same vector in k's reference frame. Notice that since the two nodes localize each other independently there is no reason that these lengths should be the same in the absence of absolute distance measurements. Given the location of a point in j's reference frame, P_j, one can transform that coordinate to k's reference frame using the following expression.

$$P_k = \left(\left(\frac{l_{kj}}{l_{jk}} \right) R_{kj} P_j \right) + \mathbf{t}_{kj} \tag{13}$$

Here the ratio $\left(\frac{l_{kj}}{l_{jk}}\right)$ accounts for the change in scale factor between the two coordinate frames. Since the procedure does not require the nodes to agree on a common scale factor it can be employed even when no absolute distance measurements are available.

These transformation can be chained so that events detected by one smart camera node can be relayed to other nodes through a sequence of transformations so that all of the events are referenced to a common frame where they can be compared and correlated.

One can imagine embedding these coordinate transforms into the communication and routing protocol so that position information is seamlessly transformed into the prevailing coordinate frame of reference as it is sent through the network.

3. EXPERIMENTAL RESULTS

In order to characterize the efficacy of the proposed localization scheme a number of experiments were carried out both in simulation and with our custom built smart camera network. Section 3.1 briefly describes the Argus smart camera nodes that we designed and built and Section 3.2 recounts the localization experiments that were carried out with those nodes. Section 3.3 describes the results of a set of simulation experiments that were designed to further characterize the behavior of the method.

3.1. Smart Camera Node

Figure 10 shows a picture of the current generation of the Argus Smart Camera System. Each smart camera system is powered by a dual core 600 MHz Blackfin processor from Analog Devices. This Digital Signal Processor was designed to support high performance image processing operations in low power devices such as cameras and cell phones. The smart camera board can be interfaced to a range of Aptina CMOS imagers, the configuration shown in the figure is outfitted with a 3 megapixel imager and a fisheye lens which affords a field of view of approximately 180 degrees. The system is also outfitted with a Zigbee wireless communication module, an Ethernet controller, an 8 bit PIC microcontroller, a three axis accelerometer and an 850 nm high intensity infrared signaling light. When properly aligned, the smart cameras can detect the infrared signaling lights at distances in excess of twenty meters.

In this realization, the center of the lens and the center of the LED array are offset by 5.5 cm. Ideally, they should be colocated. This could be accomplished by surrounding the lens with the LEDs. In practice we assume that the modeling error introduced by this offset will be negligible if the distance between the cameras is relatively large, on the order of 2 meters or more.

The unit can, optionally, be equipped with a GPS receiver and/or a three axis magnetometer which would allow it to gauge it's absolute position and orientation. The unit consumes less than 3 watts of power in operation and can be powered for 6 hours with a 6 ounce Lithium Ion battery pack.

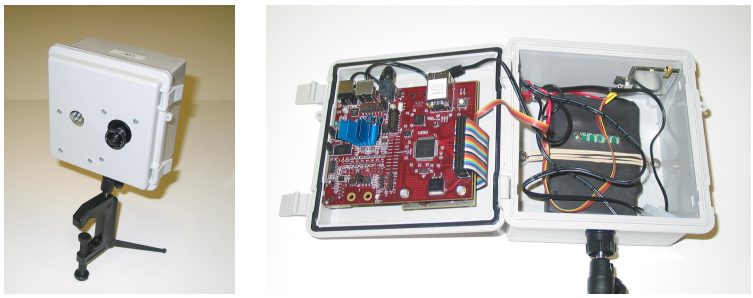

Fig. 10. Argus Smart Camera Node used in our experiments.

3.2. Indoor Experiments

These smart camera nodes were deployed in an ad-hoc manner in various locations in and around our laboratory facility. The linear localization and bundle adjustment process were carried out on the bearing measurements obtained from the sensors. The results of this procedure were then compared to measurements for the distances between the nodes obtained with a Leica Disto D3 handheld range finder. Because of the distances involved and the geometry of the camera nodes the errors in these ground truth distance measurements are on the order of 10 centimeters. In each experiment an appropriate scale factor was chosen to account for the scale ambiguity in the localization result.

For each of the deployment scenarios we show pictures of the environment along with a sketch indicating the dimensions of the space and the recovered locations of the cameras. Three dimensional renderings of the camera positions recovered by the method are also presented.

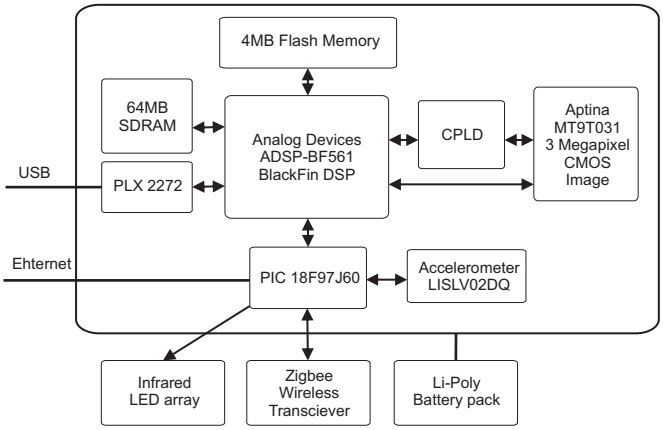

Fig. 11. Block Diagram showing the major components of the Smart Camera node.

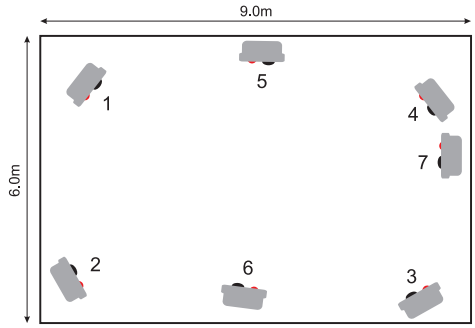

Fig. 12. Floor Plan of the High Bay area showing the dimensions of the space and the recovered locations of the cameras

3.2.1. High Bay. This experiment was conducted in the High Bay portion of our laboratory in an area 6.3 meters by 9 meters on side. The results obtained by the localization scheme were compared with 23 inter node distance measurements ranging from 3.62 meters to 9.98 meters. For the linear method the average absolute error in the recovered range measurements was 5.36 cm while the average relative error in the measurements was 0.91 %. After bundle adjustment the average absolute error was 6.24 cm and the average relative error was 1.05%.

3.2.2. GRASP Laboratory. This experiment was conducted in one of the main office areas of our laboratory in an area 20 meters by 16 meters on side. The results obtained by the localization scheme were compared with 16 inter node distance measurements ranging from 3.58 meters to 16.19 meters. For the linear method the average absolute error in the recovered range measurements was 13.17 cm while the average relative

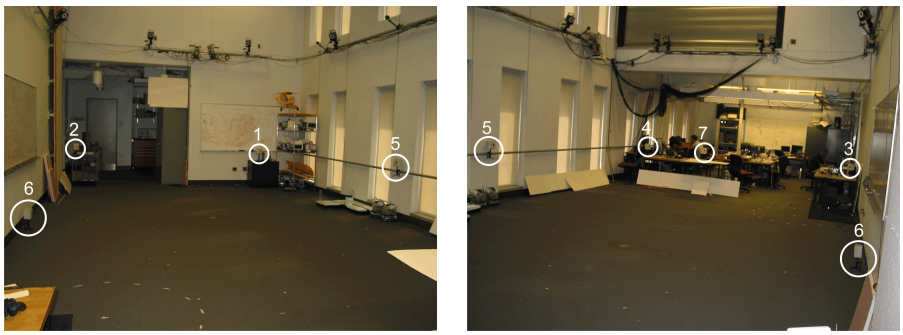

Fig. 13. Snapshots of the High Bay area showing the deployed cameras

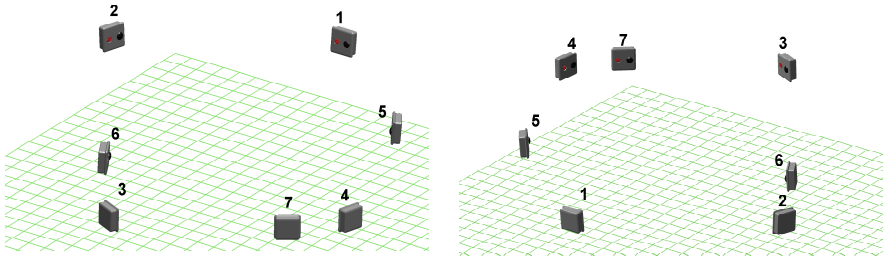

Fig. 14. Localization results returned by the proposed localization method showing the relative positions and orientations of the nodes.

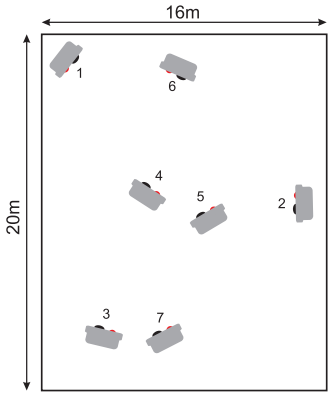

Fig. 15. Floor Plan of the GRASP Lab area showing the dimensions of the space and the recovered locations of the cameras

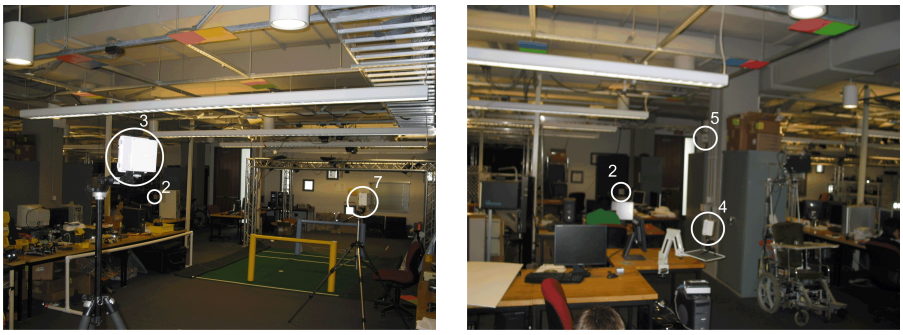

Fig. 16. Snapshots of the GRASP Lab area showing the deployed cameras

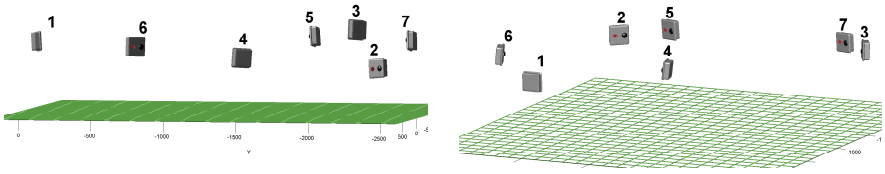

Fig. 17. Localization results returned by the proposed localization method showing the relative positions and orientations of the nodes.

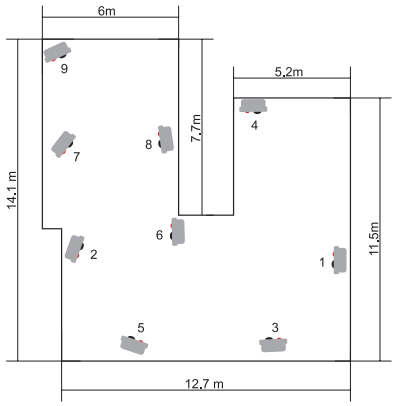

Fig. 18. Floor Plan of the first floor area showing the dimensions of the space and the recovered locations of the cameras

error in the measurements was 1.56 %. After bundle adjustment the average absolute error was 12.90 cm and the average relative error was 1.35%.

3.2.3. First Floor CS Building. In this experiment the cameras were deployed to cover the entire first floor of the Computer and Information Science building, an area ap-

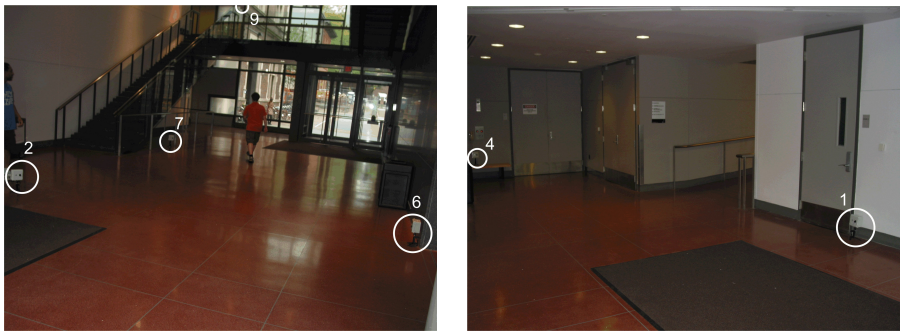

Fig. 19. Snapshots of the first floor area showing the deployed cameras

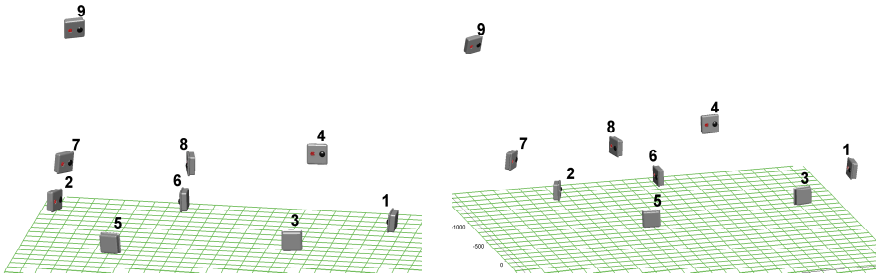

Fig. 20. Localization results returned by the proposed localization method showing the relative positions and orientations of the nodes.

proximately 20 meters by 16 meters on side. The results obtained by the localization scheme were compared with 16 inter node distance measurements ranging from 5.06 meters to 17.48 meters. For the linear method the average absolute error in the recovered range measurements was 41.40 cm while the average relative error in the measurements was 4.23 %. After bundle adjustment the average absolute error was 32.75 cm and the average relative error was 3.31%. In this experiment camera 9 was actually mounted on the mezzanine overlooking the entranceway which accounts for its vertical displacement.

3.3. Simulation Experiments

A series of simulation experiments were carried out to investigate how the proposed scheme would perform on networks that were considerably larger than the ones we could construct with our available hardware. Figure 21 shows the basic elements of these simulation experiments. The horizontal plane was divided into a grid where the cells were unit length on side. Each grid was populated with a number of virtual smart cameras which were randomly positioned and oriented within that area. In these experiments, the number of smart cameras per cell is referred to as the camera density. Limitations on the cameras field of regard were modeled by stipulating that each camera could observe all of the cameras in its own grid cell and the adjoining cells but no others. The cameras were assumed to be effectively omnidirectional so they could measure the bearing to all of the other cameras within their field of regard.

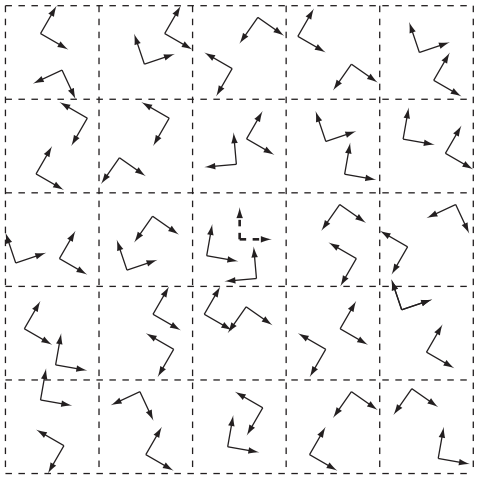

Fig. 21. In the simulation experiments the virtual smart camera nodes were randomly placed within various grid cells in the plane. Each cell is unit length on side and the number of cameras in each cell is referred to as the camera density. One camera frame at the center defines the base frame of reference.

One camera was placed at the origin of the coordinate system and this node defined the coordinate frame of reference. The proposed localization schemes were employed to recover the positions and orientations of all of the other smart cameras with respect to this base frame. The localization was restricted to the horizontal plane since vertical displacements could easily be recovered once the horizontal locations were determined.

The bearing measurements recovered by the smart cameras were corrupted by uniformly distributed random noise. The maximum error in the bearing measurements is referred to as the bearing error, so a bearing error of 2 degrees would indicate that the measured bearing could differ from the true value by up to 2 degrees.

The first simulation experiment was designed to explore how the error in the reconstruction varied as a function of distance from the reference camera. Here we explored various camera configurations within a 7 by 7 grid. For each trial we recorded the error in the rotational and translational error in each position estimate and segregated these errors based on distance. In Figure 22 the first error bar in each plot reports the mean and standard deviation of the error for cameras between 0 and 1 units from the reference camera, the second bar reports the error for cameras between 1 and 2 units from the origin and so on. The bearing error for these experiments was fixed at 2 degrees. The reconstruction procedure first recovered an estimate for the camera pose using the linear method and then refined that estimate with a bundle adjustment stage.

The graphs indicate how the rotational and translational error increase as the distance from the reference node grows. This is similar to the effect observed in robotic localization systems where small errors accumulate over time as the robot moves further from its point of origin.

These experiments were repeated for camera densities varying from 1 camera per cell up to 5 cameras per cell as shown in Figure 22. These plots indicate that as the camera density increases, the reconstruction error decreases. Effectively, adding more cameras to each cell increases the number of bearing measurements available and further constrains the reconstruction improving the accuracy.

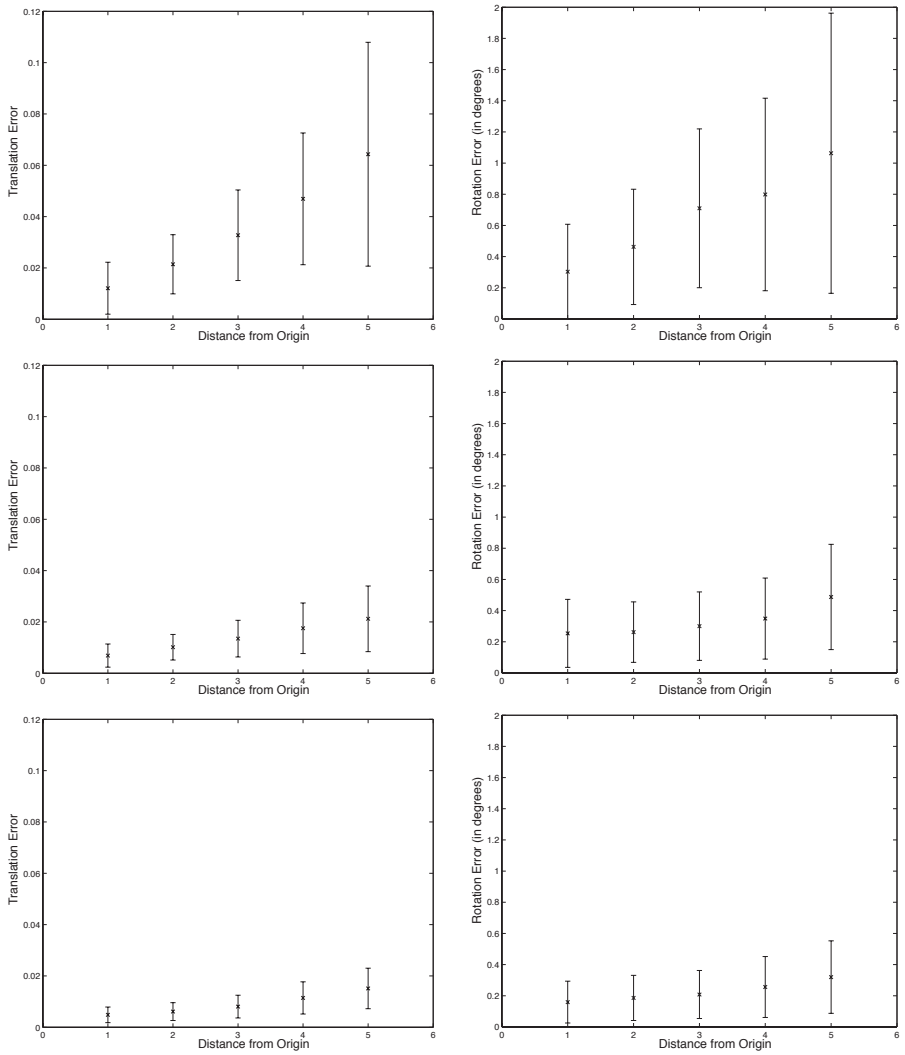

Fig. 22. This figure shows how the position and orientation errors vary as the distance from the reference frame increases and the camera density changes. The first second and third rows of graphs correspond to camera densities of 1, 3 and 5 cameras per cell respectively. The error bars in the graph indicate the mean and standard deviation of the errors in the reconstruction.

The second set of experiments was designed to explore how the error in the reconstruction varied as a function of the bearing error. Several trials were carried out on a 7 by 7 grid with a camera density of 2 as the bearing error was varied from 0.5 degrees up to 3 degrees. Figure 23 shows how the mean rotational and translational error in the reconstruction were affected as the simulated measurement error grew.

The third set of experiments characterize the improvement afforded by the bundle adjustment phase of the reconstruction procedure. The plots on the left hand side indicate the rotational and translational error as a function of distance in the estimate provided by the linear estimation stage over several trials. The plots on the right record the error after those estimates have been refined by the bundle adjustment stage. In

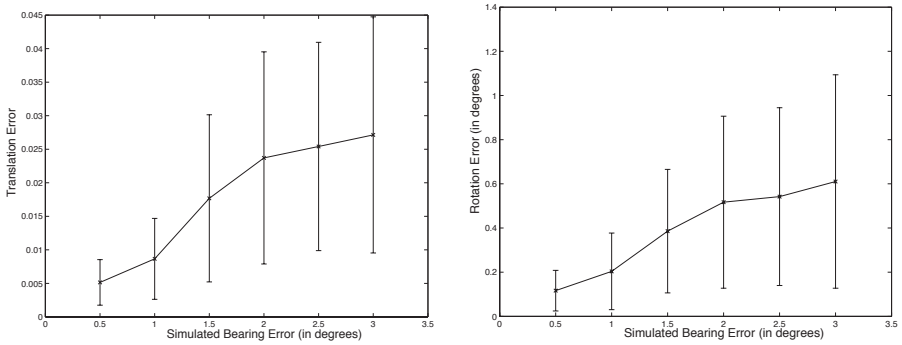

Fig. 23. This figure shows how the position and orientation errors vary as the magnitude of the bearing error increases. The error bars in the graph indicate the mean and standard deviation of the errors in the reconstruction.

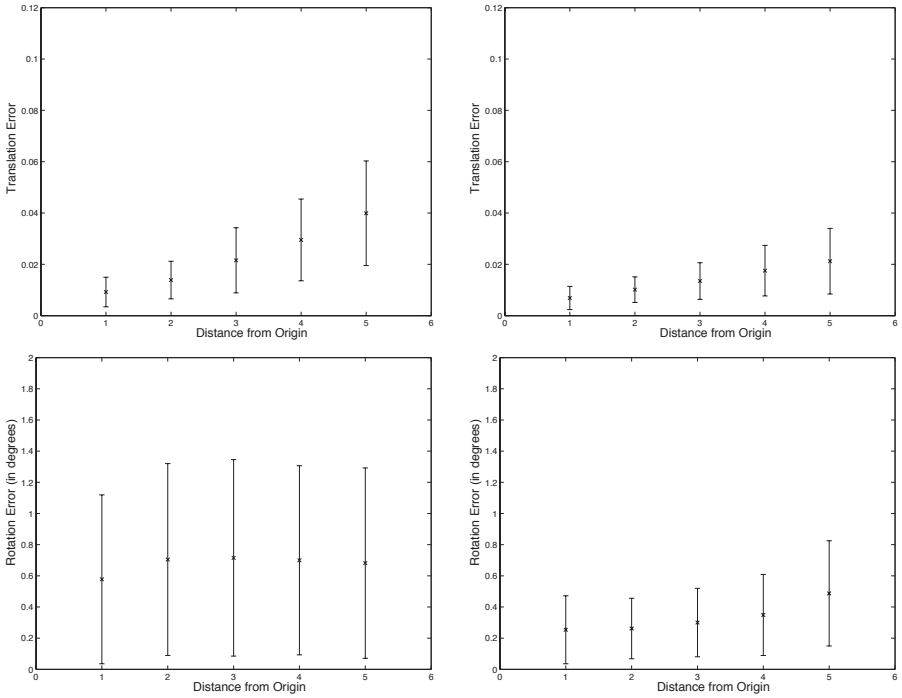

Fig. 24. The plots on the left hand column of the figure depict the error in the pose estimates after the linear phase of the reconstruction procedure while the plots on the right depict the errors after the bundle adjustment phase.

these experiments we employed a 7 by 7 grid with a camera density of 2 and a bearing error of 2 degrees. In these experiments the bundle adjustment phase typically reduces the errors in the estimates by about 50%.

Figure 25 plots the time required to perform both the linear and bundle adjustment phases of the scheme as a function of the total number of smart cameras being localized. The procedure was implemented in Matlab and run on a MacBook Pro laptop. Note that even for 451 camera positions the time required to execute the bundle

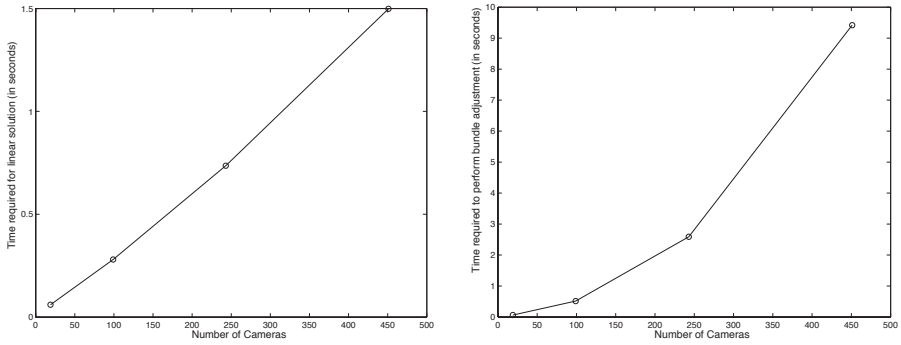

Fig. 25. This figure shows how the time required to perform the linear and bundle adjustment phases of the localization procedure grows as the number of cameras is increased.

adjustment phase was under 10 seconds. The linear phase of the reconstruction is executed in under 1.5 seconds in all cases. These experiments were run with a camera density of 2 and a bearing error of 2 degrees. The number of cameras was increased by increasing the number of grid cells.

4. DISCUSSION

This paper describes a scheme for determining the relative location and orientation of a set of smart camera nodes. The scheme proposed in this paper involves a combination of hardware and software and is, therefore, most applicable in situations where the user has some control over the design of the smart camera nodes. We argue that relatively small additions to the smart camera hardware, namely an accelerometer and a signaling LED, can be leveraged by an appropriate localization algorithm to determine the relative location of the nodes with respect to each other rapidly, reliably and accurately. Experimental results indicate that the scheme provides accurate localization results both in simulation and in practice. The results also provide some indication for how the accuracy of the procedure changes as important configuration parameters are varied.

These experimental results show that the accuracy of the proposed localization scheme compares favorably with the accuracy results reported for other distributed localization schemes on comparable problems. The simulation results provided in [Devarajan et al. 2006] consider the problem of localizing a network of 40 cameras distributed over a circle with a radius of 110 meters. The maximum level of measurement noise considered in this work was 0.1 degrees. Under these conditions their scheme localized the cameras with a mean rotation error of 0.12 degrees and a mean translation err of 120.1 cm. In our simulation experiments the minimum noise level considered was 0.5 degrees. Even with this level of noise the proposed scheme was able to localize a network of 99 cameras distributed over a square 200 meters on side with a mean rotation error of 0.15 degrees and a mean translation error of 14.28 cm.

Funiak et al. [Funiak et al. 2006] describe an experiment which involved localizing a network of 25 cameras distributed over a rectangular area of 50 square meters. Their scheme was able to localize the nodes with a root mean square error of approximately 20 cm. The scheme proposed in this paper was used to localize a network of 6 cameras distributed over 54 square meters with an average error of 6.24 cm.

In the proposed approach the critical problem of establishing correspondences between the nodes is accomplished via optical signaling. This provides a mechanism for reliably identifying nodes in the scene. Other schemes rely critically on the existence

of an appropriate set of stationary or moving targets in the scene that can be matched between views. The advantage of such schemes is that they do not require any modification of the camera hardware, however they can fail in situations where such correspondences are hard to obtain or are not appropriately spaced. Furthermore, the problem of matching objects between views is non-trivial and can require significant computational resources particularly as the number of images grows. Agarwal et al. [Agarwal et al. 2009] describe a state of the art, optimized, distributed scheme for finding correspondences between images. They report computation times of 5 hrs, 13 hours and 27 hours to find correspondences among 57,845, 150,000 and 250,000 images respectively. These computations were performed on a network of 62 dual quad core machines. Cheng et al. [Cheng et al. 2007] propose the use of feature digests to effectively reduce the amount of information that must be sent between smart cameras in a network to establish correspondences. However, this scheme still requires each camera to broadcast approximately 100 kilobytes of information to every other node in the ensemble.

The scheme proposed in this paper provides a more direct approach that can reliably identify neighboring nodes without any prior information in a matter of seconds using embedded processors. Effectively the smart camera nodes act as their own fiducials and the risks associated with relying on an appropriate distribution of feature correspondences in the scene are reduced.

Importantly the resulting sightings directly measure the epipolar structure of the camera network and, therefore, provide more information about the relative location of the nodes than shared point correspondences. This can be seen by noting that two cameras that can see each other can determine their relative position and orientation up to a scale whereas traditional relative orientation schemes require at least 5 correspondences in a non-degenerate configuration to recover the same information. Because of this the number of measurements required to localize the network and the amount of information that must be communicated between the nodes is significantly reduced. Note, however, that the localization algorithm requires an adequate number of line of sight measurements between pairs of cameras in the network so that the resulting visibility graph can be resolved.

The optical signaling scheme employed in this work assumes that the observed intensity of the signaling lights is on the order of the prevailing brightness in the scene so that the signaling changes can be measured by the cameras. For example, it would not be possible to detect a blinking LED in an image if the sun were directly behind it since the sun is so much brighter. Since the observed intensity of a light source falls off with distance, the range at which a particular signaling light can be detected will be concomitantly limited. Our experiments indicate that an array of eight standard high intensity infrared LEDs could be detected reliably at ranges up to 20 meters indoors which proved more than adequate for most deployment scenarios. For outdoor operations, camera systems employing an optical notch filter tuned to the wavelength of the LEDs have been used to detect high intensity LED arrays at distances on the order of 500 meters in direct sunlight.

Another key advantage of the proposed scheme is that it leverages the sparseness inherent in the system of sighting measurements which makes the resulting algorithms much faster than standard vision-based schemes. In recent work Agarwal et al. [Agarwal et al. 2009] describe a state of the art, heavily optimized vision-based localization system intended for city-scale reconstructions. They report reconstruction times of 16.5 hours, 7 hours and 16.5 hours on data sets with 11,868, 36,658 and 47,925 images respectively. Furukawa etal. [Furukawa et al. 2009] applied this reconstruction method to indoor environments they report reconstruction times of 13 minutes for a system with 22 cameras, 76 minutes for a system with 97 cameras, 92 minutes for a system

with 148 images and 716 minutes for a system with 492 cameras on a dual quad-core 2.66 GHz computer. Devarajan et al. [Devarajan et al. 2006] describe a distributed approach to camera localization and report reconstruction times on the order of 54 minutes for a system with 40 images.

In contrast, the method described in this paper can be used to localize networks consisting of hundreds of cameras in a matter of seconds with modest computational effort. This is particularly relevant in the context of smart camera systems where computational effort can be directly related to power consumption and time complexity determines the responsiveness of the system. In our experiments with our ten camera implementation the nodes were typically able to detect each other, communicate their measurements and recover their relative positions within 30 seconds. The most time consuming phase being the blinker detection portion which could be accelerated with faster frame rates. The resulting system is fast enough that it can be used for ad-hoc deployments and can respond quickly when nodes are added, removed or displaced.

The proposed localization scheme is effectively centralized since the sighting measurements from all of the nodes being localized are collected at a central site and then passed to the localization algorithm. However, each sighting measurement is relatively small involving only three numbers: the id of the node spotted along with the azimuth and elevation angles. On a network consisting of 1000 nodes if each camera saw 10 other targets the total amount of measurement information that would need to be collected would be on the order of 30,000 numbers.

Distributed localization methods based on consensus style approaches have the advantage of only requiring communication between neighboring nodes in the network. However, the number of communication steps in these schemes depends critically on the convergence rate of the algorithm. For example Tron and Vidal [Tron and Vidal 2009] report using 1400 rounds of message passing to localize a network of 7 nodes.

It is important to note that in this framework angular measurements derived from images and range measurements derived from other sources are treated as complementary sources of information. Measurements derived from the vision system can be used to determine the relative orientations of the camera systems which is important information that cannot be derived solely from range measurements. On the other hand, range measurements can be used to resolve the scale ambiguity inherent in angle only localization schemes. Similarly angular measurements can be used to disambiguate the mirror reflection ambiguities that are inherent in range only localization schemes. Ultimately it is envisioned that smart camera networks would incorporate range measurements derived from sources like the MIT Cricket system or Ultra Wide Band radio transceivers. These measurements could be used to improve the results of the localization procedure and to localize nodes that may not be visible to the smart camera nodes.

REFERENCES

AGARWAL, S., SNAVELY, N., SIMON, I., SEITZ, S. M., AND SZELISKI, R. 2009. Building rome in a day. In *International Conference on Computer Vision*. Kyoto, 72–79.

ALI RAHIMI, B. D. AND DARRELL, T. 2004. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. Washington D.C., 187–194.

ANTONE, M. AND TELLER, S. 2002. Scalable extrinsic calibration of omni-directional image networks. *Int. J. Comput. Vision 49,* 2-3, 143–174.

BARTON-SWEENEY, A., LYMBEROPOULOS, D., AND SAWIDES, A. 2006. Sensor localization and camera calibration in distributed camera sensor networks. In *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*. San Jose, 1–10.

BULUSU, N., HEIDEMANN, J., ESTRIN, D., AND TRAN, T. 2004. Self-configuring localization systems: Design and experimental evaluation. *Trans. on Embedded Computing Sys. 3,* 1, 24–60.

CARCERONI, R. L., PADUA, F., SANTOS, G., AND KUTULAKOS, K. 2004. Linear sequence-to-sequence alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. Washington D.C., 746–753.

CASPI, Y., SIMAKOV, D., AND IRANI, M. 2006. Feature-based sequence-to-sequence matching. *Int. J. Comput. Vision 68,* 1, 53–64.

CHENG, Z., DEVARAJAN, D., AND RADKE, R. J. 2007. Determining vision graphs for distributed camera networks using feature digests. *EURASIP J. Appl. Signal Process. 2007,* 1, 220–220.

DEVARAJAN, D., CHENG, Z., AND RADKE, R. 2008. Calibrating distributed camera networks. *Proceedings of the IEEE 96,* 10, 1625–1639.

DEVARAJAN, D., RADKE, R. J., AND CHUNG, H. 2006. Distributed metric calibration of ad hoc camera networks. *ACM Transactions on Sensor Networks 2,* 3, 380–403.

FENG, W.-C., KAISER, E., FENG, W. C., AND BAILLIF, M. L. 2005. Panoptes: scalable low-power video sensor networking technologies. *ACM Trans. Multimedia Comput. Commun. Appl. 1,* 2, 151–167.

FUNIAK, S., GUESTRIN, C., PASKIN, M., AND SUKTHANKAR, R. 2006. Distributed localization of networked cameras. In *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, New York, NY, USA, 34–42.

FURUKAWA, Y., CURLESS, B., SEITZ, S. M., AND SZELISKI, R. 2009. Reconstructing building interiors from images. In *International Conference on Computer Vision*. Kyoto, 80–87.

GIBBONS, P., KARP, B., KE, Y., NATH, S., AND SESHAN, S. 2003. Irisnet: An architecture for a world-wide sensor web. Tech. Rep. IRP-TR-04-12, Intel Research. October.

GOLUB, G. H. AND LOAN, C. F. V. 1996. *Matrix Computations*. Johns Hopkins.

HARTLEY, R. E. AND ZISSERMAN, A. 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

HEIKKILA, J. AND SILVEN, O. 1997. A four-step camera calibration procedure with implicit image correction. In IEEE Conference on Computer Vision and Pattern Recognition. *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1106 –1112.

HENGSTLER, S. AND AGHAJAN, H. 2006. A smart camera mote architecture for distributed intelligent surveillance. In *ACM Sensys Workshop on Distributed Smart Cameras (DSC 06)*. Boulder, 23–31.

KULKARNI, P., GANESAN, D., SHENOY, P., AND LU, Q. 2005. Senseye: a multi-tier camera sensor network. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, New York, NY, USA, 229–238.

LIN, C. H., LV, T., AND W. WOLF, I. B. O. 2004. A peer-to-peer architecture for distributed real-time gesture recognition. In *International Conference on Multimedia and Exposition*. Vol. 1. Baltimore, 57–60.

MOORE, D., LEONARD, J., RUS, D., AND TELLER, S. 2004. Robust distributed network localization with noisy range measurements. In *Proceedings of the Second International Conference on Embedded and Networked Sensor Systems, SenSys 04*. Baltimore, 39–50.

NATH, S., DESHPANDE, A., AND GIBBONS, P. 2002. Mining a world of smart sensors. Tech. Rep. IRP-TR-02-05, Intel Research. August.

NATH, S., DESHPANDE, A., KE, Y., GIBBONS, P., KARP, B., AND SESHAN, S. 2002. Irisnet: An architecture for compute-intensive wide-area sensor network services. Tech. Rep. IRP-TR-02-10, Intel Research. December.

NEWMAN, P. AND LEONARD, J. 2003. Pure range-only subsea slam. In *IEEE International Conference on Robotics and Automation*. Vol. 2. Taiwan, 1921–1926.

PIOVAN, G., SHAMES, I., FIDAN, B., BULLO, F., AND ANDERSON, B. D. O. 2008. On frame and orientation localization for relative sensing networks. In *IEEE Conference on Decision and Control* (2009-07-12). IEEE, Cancun, 2326–2331.

RAHIMI, M., BAER, R., IROEZI, O. I., GARCIA, J. C., WARRIOR, J., ESTRIN, D., AND SRIVASTAVA, M. 2005. Cyclops: in situ image sensing and interpretation in wireless sensor networks. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, New York, NY, USA, 192–204.

SHIRMOHAMMADI, B., TAYLOR, C. J., YIM, M., SASTRA, J., PARK, M., AND DUGAN, M. 2007. Using smart cameras to localize self-assembling modular robots. In *First ACM/IEEE International Conference on Distributed Smart Camera*. Vienna, 76–80.

SIMON, G., BALOGH, G., MAROTI, M., KUSY, B., SALLAI, J., LEDECZI, A., NADAS, A., AND FRAMPTON, K. 2004. Sensor network-based countersniper system. In *Proceedings of the Second International Conference on Embedded and Networked Sensor Systems, SenSys 04*. Baltimore, 1–13.

SINHA, S., POLLEFEYS, M., AND MCMILLAN, L. 2004. Camera network calibration from dynamic silhouettes. In *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. Washington D.C., 195–201.

SINHA, S. N. AND POLLEFEYS, M. 2006. Pan-tilt-zoom camera calibration and high-resolution mosaic generation. *Comput. Vis. Image Underst. 103,* 3, 170–183.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph. 25,* 3, 835–846.

TAYLOR, C. J. 2004. A scheme for calibrating smart camera networks using active lights. In *ACM SENSYS 04 - Demonstration Session*. Baltimore.

TAYLOR, C. J. AND CEKANDER, R. 2005. Self localizing smart camera networks. In *IEEE Conference on Computer Vision and Pattern Recognition (Demonstration Session)*. San Diego.

TAYLOR, C. J. AND SHIRMOHAMMADI, B. 2006. Self localizing smart camera networks and their applications to 3d modeling. In *ACM SenSys/First Workshop on Distributed Smart Cameras (DSC 06)*. Boulder.

TRON, R. AND VIDAL, R. 2009. Distributed image-based 3-d localization of camera sensor networks. In *IEEE Conference on Decision and Control*. Shanghai, 901–908.

TUYTELAARS, T. AND GOOL, L. V. 2004. Synchronizing video sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. Washington D.C., 762–768.

WOLF, L. AND ZOMET, A. 2002. Sequence-to-sequence self calibration. In *European Conference on Computer Vision*. Copenhagen, 370–382.

YUE, Z., ZHAO, L., AND CHELLAPPA, R. 2003. View synthesis of articulating humans using visual hull. In *Proceedings of the IEEE International Conference on Multimedia and Exposition*. Vol. 1. Baltimore, 489–492.