Smart Camera Network Localization Using a 3D Target

John Kassebaum, Nirupama Bulusu, Wu-Chi Feng
Portland State University
{kassebaj, nbulusu, wuchi}@cs.pdx.edu

ABSTRACT

We propose a new method to localize in three dimensions the camera-equipped nodes in a smart camera network. Our method has both lower costs and fewer deployment constraints than a commonly used computer vision-based approach, which is to opportunistically determine feature points in the overlapping view of pairs of cameras, compute the essential matrix for all such pairs, then perform a bundle adjustment to both refine all camera positions and orientations a determine a common scale. Our method utilizes a feature point filled 3D localization target with efficient detection algorithm to determine the projection matrix for a camera viewing the target. Because the projection matrix gives the position and orientation of the camera in the external coordinate frame of the localization target, two or more nodes simultaneously localizing themselves to the target are automatically localized in the same coordinate frame. This technique can be used to localize a smart camera network with connected views because as the target moves through the network each node will localize itself to at least two target positions that are related by an easily determined rotation and translation and which can be used to globally align all node positions and orientations to any single network-viewable target position. We present results from a real indoor network and suitably designed localization target, and show that our method can accurately localize the network when the target's feature points fill less 5% of the frame. Because the target can be relatively small in frame, pairwise camera overlap can also be small.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Computer-Communication Networks – *Network management, network monitoring, public networks.*

General Terms

Algorithms, Measurement, Design.

Keywords: Localization, smart camera networks.

1. INTRODUCTION

Distributed camera sensor networks can be used for many applications such as unobtrusive monitoring and tracking of wildlife and eco-habitats, 3D surveillance of people and vehicles in urban spaces, next generation network games and virtual reality. To establish spatial context in network deployments for such applications, one could manually measure camera positions and orientations, yet this is neither efficient nor scalable and is subject to errors. Automatic, computer vision-based localization approaches exist, including [5,6,7,8,9] which rely on determining the epipolar geometry between pairs of cameras with overlapping views; but, while accurate, the difficulty inherent to computer vision-based localization techniques is the requirement of detecting and correlating a large number of world feature points common in the views of multiple cameras. Determining these point correspondences opportunistically requires extensive data categorization [12] and message passing, beyond the capabilities of resource constrained smart camera sensor platforms [14]. Our solution directly addresses the point correspondence problem with a feature point filled and efficiently detectable 3D localization target.

Another key advantage to using a 3D localization target is that using its detected feature points to determine a camera's projection matrix gives the camera's position and orientation in the 3D coordinate frame defined by the target's geometry. Not only does this allow for a meaningful and common metric to be applied while localizing the network, but it also simplifies alignment of all cameras to the same coordinate frame. This is because any two cameras that localize to the same target position are automatically localized to the target‘s geometry.

Localizing an entire view-connected smart camera network requires moving the target through the overlapping views of all pairs of cameras. Because accuracy is maintained when the target appears small in frame, the necessary degree of overlap is small. We evaluate our solution in a real network using the Panoptes embedded video sensors [14], consisting of low cost webcams and the PDA-class Stargate processing platform. Our results show that our solution has the same level of accuracy as epipolar geometry-based methods, but requires both less computation and message passing.

2. RELATED WORK

Automated methods to localize sensor networks typically require a source of range information in order to triangulate node positions and orientations. Non-camera-equipped networks, often consisting of resource constrained scalar sensors, can measure ranges from ultrasound, radio, or acoustic signals [10]. Camera equipped networks, to which our localization solution applies, can infer ranges from visually gathered information. Visual information can be of two types: 1) motion which can be tracked to infer an object's trajectory and thereby probabilistically identify and correlate targets in different camera views, or 2) detectable static world feature points observed in overlapping fields of views of cameras and gathered either opportunistically or from deliberately placed identifiable markers. Two solutions that utilize motion tracking are [4,10]. Both maintain a joint distribution over trajectories and camera positions but only in 2D. While the results presented in [4] are restricted to 2D, [10] produces a 3D result by pre-computing camera

translations and orientations to a common ground plane in which all motion is identified and tracked. Solutions utilizing static feature point detection can localize in 3D with no prior knowledge of camera deployment. Table 1 provides a comparitive overview of 5 previously proposed solutions [5,6,7,8,9] to our own solution.

Localization methods that use static feature point detection and correlation require a minimum of pairwise overlapping views and a fully view-connected network. [5,7] require a minimum of triples of cameras with shared views. Also, the previous proposed solutions in Table 1 rely on essential matrix estimation to determine the epipolar geometry between pairs of cameras, and thereby deduce their relative poses ([5] estimates projection matrices from determining epipolar geometry). But because essential matrix estimation requires correlated sets of image points of commonly detected features whose 3D world coordinates are unknown, the scale of the geometric relationship between the cameras cannot be determined. [7,8] each propose the use of a calibration target consisting of a bar of known length with LEDs at each end; the known 3D length between the detected lights serves as a constraint in determining scale. Lacking a target, opportunistic feature point detection using SIFT [12] must be employed which requires extensive image processing and messaging to correlate image points across views. Also, because scale is unknown in each pairwise essential matrix estimation, each must be realigned to a common, but still unknown, scale using a centrally processed bundle adjustment over all camera parameters and triangulated world feature points.

Our solution, utilizing a 3D localization target, seeks to minimize the cost of feature point detection and inherently provide scale. The image points of the target's feature points are easily and robustly determined by a simple and efficient detection algorithm. Because the geometry of the target is known and used for projection matrix estimation, camera position and orientation is always given in the target's coordinate frame.

3. ALGORITHM

3.1 The Projection Matrix

The essential matrix expresses epipolar geometry between two cameras and can be estimated from correlated sets of image points of world features common in both camera's views. The projection matrix expresses how one camera *projects* world coordinates in an independent 3D coordinate frame to 2D pixel points. This projection transforms between four different coordinate frames, shown in Figure 1.

3D world coordinates in an arbitrary world coordinate frame (WCF) are translated and rotated to a camera-centric

Table 1. Comparison of feature point correlation-based smart camera network localization methods

	Devarajan *et al.*	**Lymberopoulos *et al.***	**Mantzel *et al.***	**Kurillo *et al.***	**Medeiros *et al.***	**Our solution**
complexity	structure from motion estimates projection matrices for triples of cameras	epipolar estimation; refinement via pre-known constraints	iterative epipolar and projection matrix (re)estimation	epipolar with centralized sparse bundle adjustment	iterative epipolar estimation, distributed refinement	pairwise projection matrix estimation
algorithm	nodes correlate common inlier feature points to form a vision graph; graph guides clustering (min=3) where each node in cluster estimates all nodes' projection matrices using SFM and refines with bundle adjustment. A second BA refines camera parameters and feature points including reconstructed feature points not in the original set of inliers.	pairwise cameras compute epipoles and orientation from direct epipole observation or fundamental matrix estimation; known lengths between some nodes provide scale and constrain refinement of unknown lengths	pairwise localized nodes triangulate 3D world points to provide to unlocalized neighbors who then localize from projection matrix estimation; more localized cameras causes more pairwise localizations causes re-triangulated 3D world points -- which causes re-localization of individual cameras; iterates until convergence of localizations	target provides image point correspondences for pairwise essential matrix estimation; scale rectified by target known dimension; shortest paths on vision graph guides realignment of pairwise localizations to global coordinate frame; centralized refined with sparse bundle adjustment	target provides image point correspondences for pairwise essential matrix estimation; scale rectified by target known dimension; reference index algorithm guides realignment of pairwise localizations to global coordinate frame; distributed weighted recursive least squares technique updates localizations when either new target point acquisitions allow re-pairwise calibration or when neighbor provides updated localization estimate	easily detectable 3D target of known geometry provides image to world point correspondences for pairwise projection matrix estimation and common metric and scale for all pairwise localizations; target path through network guides realignment of pairwise localizations to global coordinate frame
assumes	nothing	known lengths between nodes	some cam positions and orientations known	precalibrated cameras	precalibrated cameras	precalibrated cameras
deployment constraints	multiple triples (or more) of cameras with shared fields of view	views overlap, some cameras see each other	multiple triples of cameras with shared fields of view	pairwise overlapping views, time synchronized nodes	pairwise overlapping views, time synchronized nodes	pairwise overlapping views, time synchronized nodes
message passing	SIFT categorized feature points to neighbors	estimated distances between nodes to neigh.	initial and revised localization estimates	initial and refined coordinates of observed 3D points	detected 3D points; neighbors' relative position when updated	transformation to global frame to pairwise neighbor
feature point detection	opportunistic using SIFT and RANSAC to detect highly reliable inliers	LEDs on camera and non-camera nodes	unspecified, suggests opportunistic via opportunistic motion tracking	opportunistic detection of known length bar with LED on each end	opportunistic detection of known length bar with LED on each end	opportunistic detection of 3D target
accuracy	simulated 45cm error per camera at 220m scene width at 1 pixel noise, orientation error not clear; accuracy of actual shown only as accurate wireframe reconstruction	actual 60cm error at 297cm avg node-to-node length; 20cm if all epipoles observed; no orientation error given	no actual deployment evaluated; simulated error of .25% of deployment area diameter; not stated how error applies to orientation	simulated 0.2% position error (% of RMSE between est. and actual) at noise < .6 pixel; small reprojection error stated for actual test	simulated positional error ≈ 1" with no noise; very accurate orientations; no actual deployment evaluated	actual pos. and orient. error < .01% for pairwise localizations; max global error < . 01n% n hops from first camera localized
issues	requires multiple camera overlaps; actual test contains set of 12 overlapping cameras	best when camera nodes see and are able to detect each other	particular method suited to 3D target, but none tested	accuracy greatly improved if target spans 1/3 of frame width	benefits from multiple target passes as more 3D points triggers update to localizations which triggers more refinement	propagates single camera localization errors to later pair-wise localizations

coordinate frame (CCF) whose origin is the camera's point of perspectivity. Using homogenous coordinates for 3D points and where R is a 3D rotation matrix and C a 3D translation vector:

$$\begin{pmatrix} \mathbf{X}_{CCF} \\ 1 \end{pmatrix} = \begin{pmatrix} R & -R\mathbf{C} \\ \mathbf{O}_3^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{X}_{WCF} \\ 1 \end{pmatrix}$$

Next, 3D CCF points are *projected* to 2D points in the image coordinate frame (ICF). The projection is a scaling of 3D CCF points by *f/Z* along their ray through the origin, where *f* is the camera's focal length; the image plane is at a distance *f* from the CCF's origin.

$$\begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{CCF} = (fX_{CCF}, fY_{CCF}, Z_{CCF})^T \equiv \left(\frac{fX_{CCF}}{Z_{CCF}}, \frac{fY_{CCF}}{Z_{CCF}} \right)^T$$

The 3D scaled point is considered a 2D homogenous point and converted to inhomogeneous coordinates.

Finally, 2D ICF points are translated to 2D pixel coordinate frame (PCF) points. This is a transformation from a right handed coordinate system to the traditional left handed PCF which has its origin in the top left corner of a frame. If (x_0, y_0) are the coordinates of the ICF origin in the PCF:

$$u = x + x_0 \text{ and } v = y + y_0$$

The projection matrix combines all transformations into a 3x4 matrix:

$$P = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & -R\mathbf{C} \\ \mathbf{O}_3^T & 1 \end{pmatrix}$$
$$= KR(I_{3x3} | -\mathbf{C})$$

K is referred to as the camera calibration matrix and contains the camera's 5 *intrinisic* parameters. R and $\mathbf{C}$ are the camera's 6 *extrinsic* parameters, which yield the camera's orientation and position in the WCF.

3.2 Camera Position and Orientation

P can be estimated from a correlated set of known 3D world points and their 2D pixel points. Reportedly, 28 point correlations are sufficient, but due to noise in real-world feature point detections, using more correlations gives better results. The point coordinate values are used in an over-determined system of linear equations that is solved with the singular value decomposition. The left 3x3 submatrix of the estimate of P can be decomposed into KR using the RQ decomposition [3]. Because $[I_{3x3} \mid -\mathbf{C}]\mathbf{C} = 0$, $\mathbf{C}$ is determined from the null space of P.

We use Levenberg-Marquadt to minimize *projection* error. Projection error is computed as the distance between a detected feature point and its projection by the estimate of P. We both reduce the cost of minimization and achieve more accurate results by using pre-computed intrinsic and lens distortion parameters [3] in the evaluation of the error function, rather than the parameters obtained from the decomposition of P.

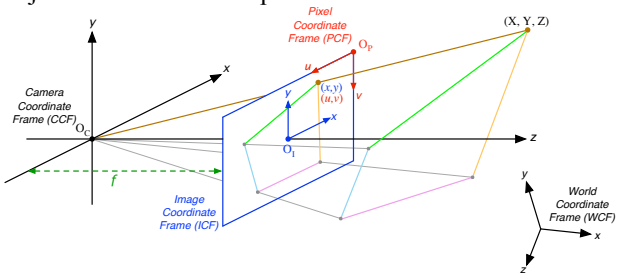


Figure 1. Projection of 3D points to 2D pixel points

3.3 Network Localization

The rotation matrix and translation vector decomposed from an estimate of P give the camera's position and orientation in the 3D world coordinate frame defined by the geometry of the localization target. Thus, any 2 or more cameras that localize to the same target position are automatically localized in the same coordinate frame. The 2 (or more) cameras' relative positions and poses are determined without having computed an essential matrix.

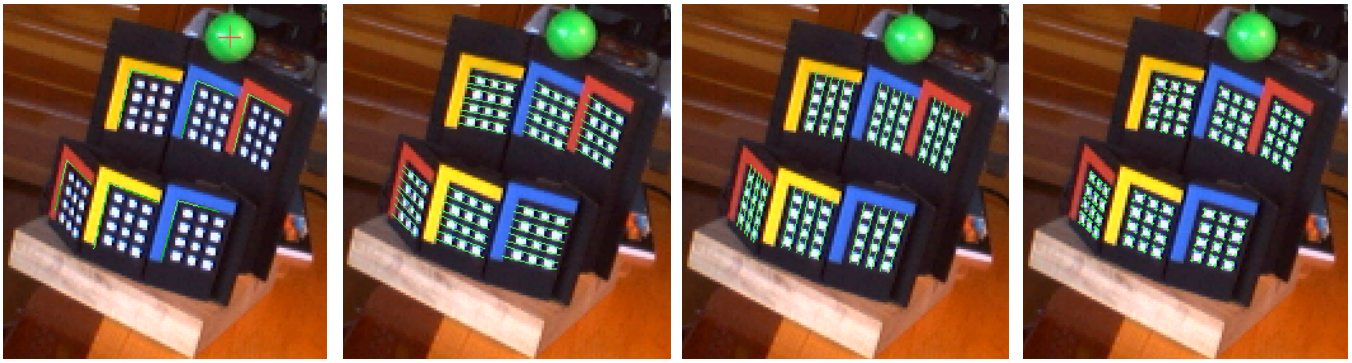
Globally localizing an entire view-connected network in the same 3D coordinate frame requires subsequently positioning the localization target in the view of all pairs of cameras. This movement can be automated. When the target appears simultaneously to two unlocalized cameras, each localizes to the target's current coordinate frame by estimating P after detecting the target's feature points. When the target later appears simultaneously in the view of an already localized camera and an as-yet unlocalized camera, again each localizes to the target's current position but then the camera with 2 different localizations computes and passes to the other a rotation and translation that realigns the current target coordinate frame to the prior. This has the effect of bringing the newly localized camera into the same coordinate frame as that shared by the other camera and its previous pairwise partner. The realignment of cameras to previous coordinate frames can occur either in a linear fashion as the target moves through the network, or it can be done in a more strategic way, such as after computing all pairwise localizations, realignment begins from the camera pair that has the shortest path to the leaves in a vision graph of the network [5,8]. Because error from single camera localizations propagates with the realignments, which will be discussed in Section 4, the latter approach is highly advised.

3.4 The Localization Target

Due to the widely varying environmental and lighting conditions in possible smart camera network deployments, as well as variations in camera quality, subject size, and baselines between cameras, it is unreasonable to expect that any one 3D localization target will be suitable for all networks. Rather, a target should be designed specific to the deployment environment and purpose. To demonstrate the practicality of using a 3D localization target, we have designed and created a small target with 288 feature points set across 6 differently angled grids. We have also designed and implemented a simple and efficient detection algorithm.

Detection of our target (in a 640x480 image) begins by stepping to find a green pixel. Then: find all contiguous green pixels on the row; from the line's midpoint, find all contiguous green pixels on the column; consider this vertical line to be the vertical diameter of the sphere atop the target;

Figure 2. Detecting the 3D localization target's feature points



use the midpoint as a starting reference for finding all grid-side edges of the colored areas beside each grid. These edges define *target-relative* horizontal and vertical lines that bound a grid and define scan line length and orientation for finding edge fits to all sides of squares in the grid. Intersecting edge fits gives corners of squares, which are the feature points of the target, shown in Figure 2. Our results are generated with the target upright for ground truth measurement purposes, but the detection algorithm does not require it to be so. We have also verified detection functions well under various lighting conditions.

4. EVALUATION

4.1 Single Camera Localization

Our algorithm's accuracy is dependent upon the accuracy of the projection matrix estimation by individual cameras upon observing the target. This is because each camera computes and passes to at least one neighbor a transformation between two target coordinate frames it has localized to. Any error in these localizations is propagated through the network via the passed transformations.

Figures 3 and 4 demonstrate the accuracy of single camera localizations using our indoor localization target. Figure 3 shows the position error of a single camera's localization as the target is placed successively further from the camera. Position error is defined as the percentage of the Euclidean error of the camera position estimation to the camera-to-target distance. Percentage of frame area occupied by point matches means the percentage of the area, in square pixels, of the bounding box around all target feature points to the total image area. The graph shows that the localization error when the target is close to the camera is the same at subsequent target positions–thus the decreasing percentage. This suggests that the error is likely from an inaccuracy in manual measurement, because it is consistent. Figure 4 shows that estimated orientation angles fluctuate less than 0.3 of a degree over the same configuration.

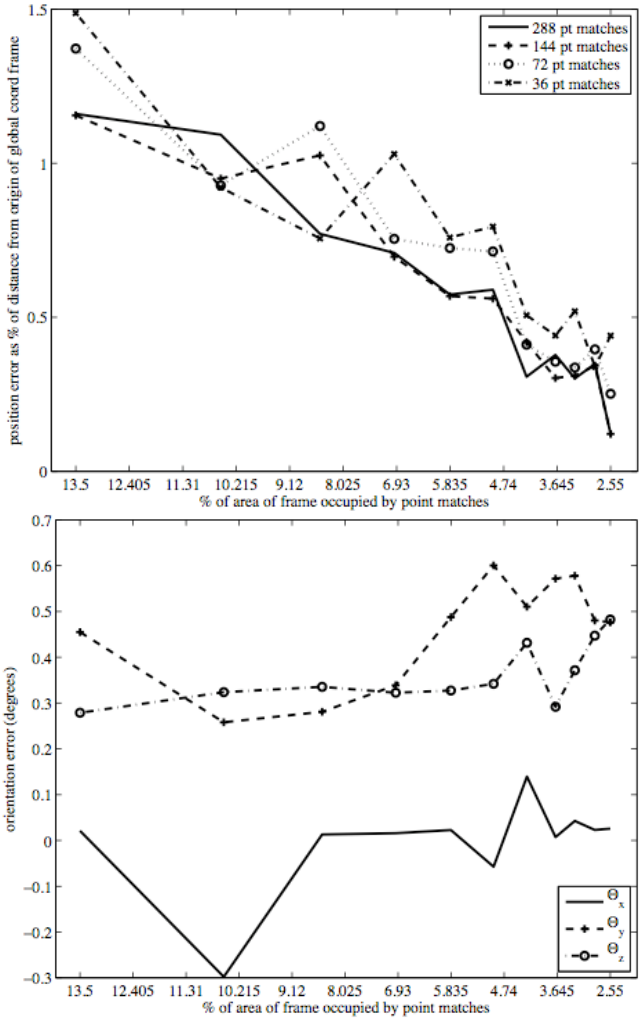
4.2 Network Localization

Figure 5 shows the position error of cameras realigned to the network's global coordinate frame. Due to the propagation of single camera errors in transformations passed to neighbors, the error increases at each *hop* away from the camera chosen as origin of the global coordinate frame. Figure 6 shows the change in the estimate of the *z*-axis orientation at each hop. This error is consistently higher than error in other orientation estimates, and may be due to the fact that in our setup the *z*-coordinate value of target positions was much greater than the other two.

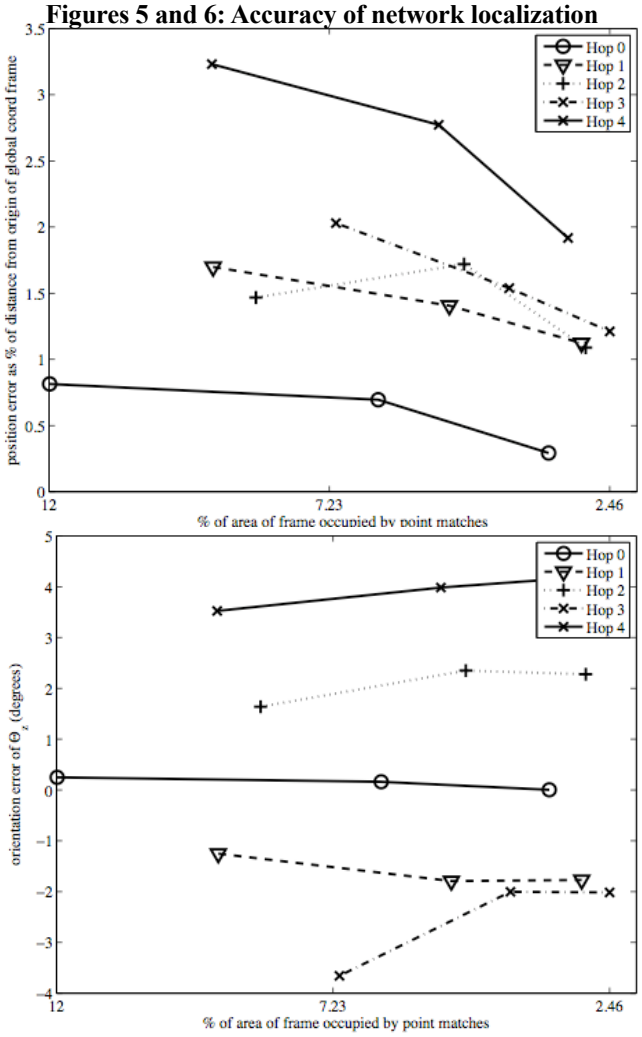
4.3 Message Passing

Message passing in our solution consists only of determining simultaneous target detections between pairs of cameras, and

Figures 3 and 4: Accuracy of single camera localizations



Figures 5 and 6: Accuracy of network localization



the passing of realignment transformations. Because projection matrix estimation occurs between the target and one camera, there is no need to pass or correlate detected feature point sets between pairs of cameras.

5. FUTURE WORK

Due to the uncertainty manual measurement errors cast over our real deployment results, we are implementing a simulator. We will also implement a centralized bundle adjustment for comparison purposes, as well as the use of local pairwise bundle adjustments, although both would increase message passing if adopted into the solution.

6. CONCLUSION

We have presented a new solution for smart camera network localization in 3D that addresses both the point correspondence problem and high amount of processing required in epipolar geometry-based computer-vision localization algorithms. Our solution also addresses the unknown scale issue inherent in using epipolar geometry to determine relative pose between cameras. Recent epipolar geometry-based solutions [8,9] propose the use of a simple 2D calibration target to resolve the scale issue. Our solution takes the next step of a full-featured 3D target that not only resolves scale, but also reduces both message passing and computation.

7. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0722063. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

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

[2] Faugeras, O. The Geometry of Multiple Images. *MIT Press*, 2004.

[3] Zhang, Z. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22, 11, (Nov 2000), 1330-1334.

[4] Funiak, S., Guestrin, C., Paskin, M., and Sukthankar, R. Distributed localization of networked cameras. *Information Processing In Sensor Networks*, (Apr 2006), 34-42.

[5] Devarajan, D., Radke, R., and Chung, H. Distributed Metric Calibration of Ad-Hoc Camera Networks. *ACM Transactions on Sensor Networks*, 2, 3 (Aug. 2006), 380-403.

[6] Lymberopoulos, D., Barton-Sweeny, A., and Savvides, A. Sensor localization and camera calibration using low power cameras. ENALAB Technical Report, 090105, September 2005.

[7] Mantzel, W., Choi, H., Baraniuk, R.G. Distributed camera network localization. *38th Asilomar Conference on Signals, Systems and Computers,* (Nov 2004)

[8] Kurillo, G., Li, Z., Bajcsy, R. Wide-area external multi-camera calibration using vision graphs and virtual calibration target. *Second ACM/IEEE International Conference on Distributed Smart Cameras*. (Sep 2008)

[9] Medeiros, H., Iwaki, H., Park, J. Online distributed calibration of a large network of wireless cameras using dynamic clustering. *Second ACM/IEEE International Conference on Distributed Smart Cameras*. (Sep 2008)

[10] Taylor, C., Rahimi, A., Bachrach, J., Shrobe, H., and Grue. A. Simultaneous localization, calibration, and tracking in an ad hoc sensor network. *Information Processing In Sensor Networks,* (2006), 27-33.

[11] Rahimi, A., Dunagan, B., Darrell, T. Simultaneous calibration and tracking with a network of non-overlapping sensors. *Computer Vision and Pattern Recognition,* 1, (Jul 2004), 187-194.

[12] Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60, 2 (2004), 91-110.

[13] Bindel, D., Demmel, J., and Kahan, W. On computing givens rotations reliably and efficiently. *ACM Transactions on Mathematical Software*, 28, 2, (Jun 2002), 206-238.

[14] Feng, W., Code, B., Kaiser, E., Shea, M., Feng, W., Panoptes: scalable low-power video sensor networking technologies. ACM Multimedai (2003), 90-91.