Actuation-Assisted External Calibration of Distributed Camera Sensor Networks

Jeffrey Mascia, Juo-Yu Lee, Nima Nikzad,
Mohammad Rahimi, and Mani Srivastava
{jmascia@,juoyul@,nnikzad@,mhr@cens.,mbs@}ucla.edu
UCLA

Abstract

While cameras have the potential to enable many applications in sensor networks, to be effective they first must be externally calibrated. In prior systems, cameras, identified by controllable light sources, utilized angular measurements amongst themselves to determine their relative positions and orientations. However, the typical camera's narrow field of view makes such systems susceptible to failures in the presence of occlusions or non-ideal configurations. Actuation-assistance helps to overcome such issues by essentially broadening each camera's view. In this paper we discuss and implement a prototype system that uses actuation to aid in the external calibration of camera networks. We evaluate our system using simulations and a testbed of MicaZ nodes, equipped with Cyclops camera modules mounted on custom pan-tilt platforms. Our results show that actuation-assistance can dramatically reduce node density requirements for localization convergence.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems; I.4.1 [**Image Processing and Computer Vision**]: Digitization and Image Capture—*Camera calibration*

General Terms

algorithms, experimentation, performance

Keywords

sensor networks, actuation, localization

1 Introduction

With the continuous fall in the price and complexity of the imaging technology, distributed image sensing using networked wireless cameras is increasingly becoming feasible for many applications such as smart environment, surveillance and tracking. A key problem when aggregating information across spatially distributed imagers is determination of their poses (i.e. 3D location and 3D orientation) which is essential to relating the physical world information observed by the different sensors in a common coordinate system. In computer vision the problem of determining 3D pose of a camera is referred to as the external calibration problem, to contrast it with the internal calibration problem of determining internal parameters of the optical system that may vary from camera to camera.

While there is significant prior research on the related problem of node localization in sensor networks, particularly using non-vision sensing modalities such as RF, acoustic, ultrasound, and their combinations, those techniques reveal only the location whereas external calibration requires estimation of the entire six-dimensional pose for all the nodes. More recently, a new family of approaches that exploit vision for localizing the nodes in the network have emerged. These techniques are appealing in networks of cameras for two reasons. Firstly, they use the same modality that is used for sensing, thereby minimizing the cost and complexity of the sensor nodes. Secondly, they not only recover the location (often up to a scale as will be discussed in section 2) but also the orientation of the nodes in the network. At the sensing level, these approaches are either passive relying on two or more cameras observing common feature points, or active using a number of optical beacons that can be identified by the other nodes in the network. In the passive approach, establishing correspondence among common feature points between pairs of the cameras is a significant challenge particularly as the baseline between cameras increases in realistic deployment scenarios. In the active approach, a main challenge is that the limited field of view of cameras poses stringent constraints on the sensor node poses in order to guarantee enough common beacon observations for the external calibration procedure to succeed for the entire network.

We present an actuation assisted vision-based external calibration approach for network of cameras. In our approach, the network consists of actuated camera devices which use pan and tilt capabilities to adapt their field of view for sensing and for external calibration of the network. To calibrate, the nodes, which in addition to having an image sensor are also equipped with an optical beacon, continuously reconfigure their poses to intercept other nodes optical signal and extract their identities. Furthermore, each node transmits the identity and associated pose of its neighbors to a fusion point, where the information is then processed to

recover each node's reference pose (i.e. zero pan and zero tilt). Such processing could conceivable also be done in a decentralized fashion, although our current implementation does not.

Recent research such as [1, 2] has shown that image sensors with actuation capability have advantages during the operation of the system the sensing quality is significantly enhanced, particularly for environments with poorly modeled occlusions and target objects with frequent motion. We argue that the larger effective coverage provided by actuation capability offers several advantages over static imagers in deployment and configuration of the network as well. First, actuation reduces the network density required for the external calibration procedure to converge and establish a single unified coordinate frame for the entire system. This permits the deployment of the network to be driven primarily by sensing considerations, and not constrain deployments to those static configuration for which external calibration will converge. In any case, finding such static configurations where the external calibration will converge in the presence of environmental occlusions is a non-trivial problem. A second benefit of actuation is that it can potentially provide more accurate external calibration results compared to an static network with an equivalent number of nodes due to additional number of observations enabled by actuation.

Clearly actuation capability is not without a price. Actuation is costly in terms of node power consumption, size and node cost. On the other hand, since external calibration and re-calibration of a network is infrequent, one can design the actuation platform with very low static power (vs. active power) to minimize overall power consumption and thus the impact of actuation on the network lifetime. We also note that while the cost of individual actuation-capable image sensor node is higher than that of a static node, the significant reduction in node density enabled by actuation translates into a large overall system cost reduction.

The primary contribution of this paper is a technique for external calibration (i.e. finding location and orientation) of spatially distributed networked image sensors that exploits pan and tilt actuation capability. Through simulation as well as experimentation using the commercially available Cyclops imagers[1] [3] equipped with Crossbow Technology Inc.'s MicaZ mote and a custom pan-tilt mechanisms designed by us for very low static power, we illustrate that by exploiting actuation our technique can enable external calibration of far sparser networks than is possible with techniques that do not exploit actuation. A secondary contribution of our work is the pan-tilt platform itself which brings actuation capability, previously considered to be an exclusive realm of higher-end power-hungry cameras, to low-end power and resource constrained image sensors such as Cyclops.

The rest of this paper is organized as the following. Section 2 discusses the related work on non-vision and vision based localization and external calibration techniques. Section 3 discusses the advantages offered by actuation. Section 4 describes the fundamental concepts underlying our methodology of exploiting actuation during external calibration. Sections 5 and 6 describe the platform and the specific technique for actuation-assisted external calibration. Section 7 presents results of our simulation and experimental evaluation. Finally, section 8 discusses the limitations and capabilities of our work, and directions for future research.

[1] http://cyclopscamera.com/

2 Related Work

The desire for self-configuration as well as various operational and environmental constraints have motivated significant research in sensor network localization. Typically, the non-vision based approaches to sensor network localization use techniques such as time-of-flight (TOF), direction-of-arrival (DOA), signal strength, signal-pattern matching etc. Time-of-flight gives the distance between two uncalibrated nodes or between an uncalibrated node and a geo-calibrated beacon, which can then be used as part of a multilateration procedure to estimate node locations. For the short inter-node distances encountered in sensor networks, the time-of-flight is measured either by a receiver node measuring the difference between arrival time of a radio signal and a slower acoustic or ultrasonic signal that were sent by a sender with a known time gap [4, 5], or by using some separate time synchronization mechanism between the two nodes, and calculating the difference between sender and receiver timestamps for an acoustic or ultrasonic signal [6]. Direction-of-arrival schemes usually involve acoustic source with beamforming or RF sources with directional or sectored receive antennas to measure the relative direction between two uncalibrated nodes or between am uncalibrated node and a geo-calibrated beacon. The positions can then be obtained by using triangulation methods [7, 8, 9]. Signal strength can be combined with radio propagation fading models to relate the distance between sensors and beacons [10, 11]. In the signal-pattern scheme, the multipath becomes effective for generating the location-related signal-signature database [10]. More recently, researchers have explored radio interferometric localization [12] where the beat frequency resulting from interference among two radio signals is used to establish constraints among inter-node distances in groups of nodes, and then the constraints are combined to find relative positions of the nodes. Finally, range-free scenarios eliminate specialized hardware with coarse error granularity [13].

Almost all the non-vision based techniques typically focus on recovering only the location of the nodes, and not their orientation. Some researchers have explored the estimation of orientation either by using combinations of magnetometer acting as compass and accelerometers giving orientation with respect to the vertical [14], or by attaching two localizable tags to a rigid body and then using location estimated of the two tags to find the orientation of the rigid body [15]. Unfortunately, these techniques suffer from poor estimation of orientation and thus not used in practice.

By contrast, general vision based localization schemes enable recovery of both relative rotation and translation between cameras. The constraints of vision-based localization typically result from the requirement of the number of common feature points and the pose of cameras, particularly as cameras are highly directional with relatively narrow field

of views. Mantzel et. al. [16] considered distributed vision network calibration based on an iterative localization approaches in which six or more feature points are required. Devarajan et. al. [17] extended techniques of factorizing measurement matrix and used at least twelve common image measurements shared with three neighbors. Lee et. al. [18, 19] proposed techniques to address with localization problems with parameters defined in 2D plane are estimated with a priori assumption of image plane orientation. The work of Barton-Sweeney et. al. [20, 21] established vision network localization based on epipolar geometry. The study showed that epipoles could be directly observed or indirectly estimated. Accuracy varies depending on the applied methods. Generally, direct epipole measurements result in better precision. Taylor et. al. [22] specifically applied Bundle Adjustment to minimize the discrepancy between ground truth measurements and estimation based on iteratively presumed pose configuration. Due to the numerical requirements of Bundle Adjustment, the vision network itself is not allowed to exhibit significant imbalance between the number of observations and the number of unknown parameters representing relative rotation and translation. Much of this prior work has relied on passive approaches only, although some such as [20, 21] use active beacons. As we show later in the paper, approaches based on static cameras either require very dense networks or require the cameras to be carefully deployed so that the calibration procedure will succeed. At low densities, very few of the large number of combinations of possible static camera locations and poses lead to successful calibration. This results in high deployment cost (either in the form of more nodes or more placement effort) and makes the deployment constrained by not just sensing but also calibration. By exploiting actuation, our approach significantly decouples the two and enables successful external calibration of much sparser networks and without careful manual placement.

Unlike the above prior work which is for purely static cameras and static environments, [23] describes a distributed approach where cameras collaborate to track an LED-equipped beacon object that moves through the environment and reason probabilistically about which camera poses are consistent with the observed images. Like our work, this also exploits exploitation, except it is the beacon which is actuated and not the cameras. From practical perspective, the two approaches offer different benefits. While [23] does not require actuated cameras, it does require more manual effort which becomes an issue for re-calibration. While one might imagine the use of robotic actuated beacons, the mechanical complexity required for real world operation will be substantially more than that of simple pan-tilt actuation in the cameras. The computation involved is also significantly more complex and the time to calibrate longer with the actuated object approach. Moreover, as shown by [1, 24, 25, 2], actuation of cameras also offers significant sensing performance advantages, and if actuation is present for reasons of sensing then it would be better to use it for external calibration as well instead of introducing additional complexity of actuated beacon. Overall, our approach and that of [23] reside at different points in the design space with different regimes of applicability.

Finally, we note that actuated vision network not only provides alternative localization remedies, but also inherently has superior advantages on surveillance and monitoring. The work of Kansal et. al. [24, 25] has shown that low-complexity primitives such as pan and tilt can be exploited to improve resolution for required spatial coverage and reduce sensing uncertainty with decreasing actuation overhead. Chu et. al [1] demonstrated the actuated vision network facilitates detection and identification of anomalous activities in the environment. From the application perspective, the combination of actuation and camera localization potentially has indispensable capability to be embedded in vision networks. Our work serves to mitigate some constraints, such as density and likelihood of localizability, on previous vision network localization problems while achieving comparable or improving accuracy.

3 Advantages of Actuation

In this section we briefly overview the advantages of actuated vision networks. One of the main benefits of actuated vision networks is network density reduction. Later, in Section 7 we show that there is a significant network density reduction in an actuated vision network while still achieving calibration. Other advantages of actuated vision networks are localization accuracy improvement, network deployment cost reduction and sensing performance enhancement.

3.1 Network Density

For a given node density, an actuated vision network is significantly more likely to achieve self-calibration of location and orientation than a network of static imagers. Furthermore, an actuated vision network can localize target events in a much greater volume of space than a static vision network of the same density can. On average, each object can thus be observed by more cameras in actuated vision network, since a actuated camera is essentially equivalent to a static camera with a much wider field of view, albeit one with observation latency. If the goal is to observe certain fixed regions of space, it is easy to deduce qualitatively that actuation translates into a density gain from both calibration and sensing performance perspective. In particular, relative to a network of static imagers, one can reduce the number of sensors placed solely to achieve full self-calibration but which are redundant from sensing perspective.

3.2 Localization Accuracy

For a vision network, a minimum set of observations is required to estimate the poses and positions of nodes relative to a common frame of reference. Since the effectiveness of individual observations may be offset by noise from the imager or the actuator, network-wide localization estimates can be inaccurate. Additional observations provide extra constraints which can be used to refine these network-wide localization estimates as described in Section 4.3. For any quantity or geometrical configuration of nodes, an actuated vision network yields significantly more observations than does a static vision network. Consequently, one would expect actuated vision networks to yield more accurate self-localization estimates.

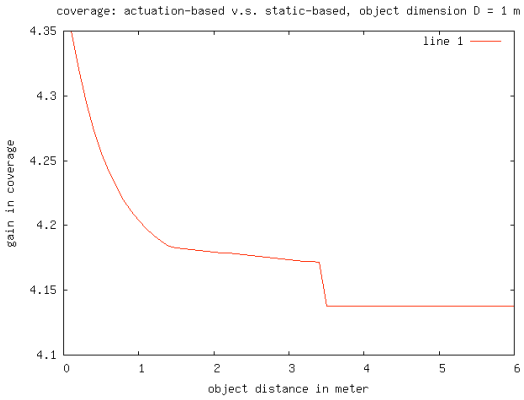

Figure 1. Coverage ratio of actuation-based and static-based schemes.

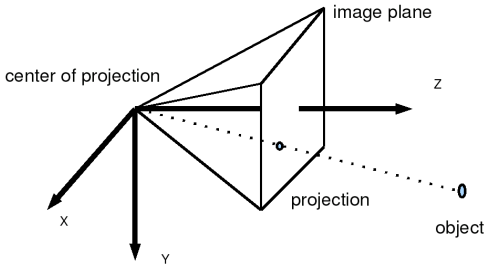

Figure 2. The pin-hole model.

3.3 Network Deployment

In certain environments, the particular orientations of cameras can dramatically affect the network's likelihood of achieving self-localization. Degradation of sensing quality due to inability of pose reconfiguration can easily occur in highly occluded environments. In a network of static cameras, the deployer must take extra effort when installing nodes to maximize potential for localization while meeting the sensing coverage goals. Consequently, the deployment is constrained by two different goals - achieving self-calibration and achieving sensing coverage - and makes the task harder. Moreover, a change in the environment obstacles may render the initial choice of camera locations and poses unworkable both for sensing coverage and future re-calibration. By contrast, in a network of actuated cameras, the deployer can afford to be relatively agnostic to the initial orientations of cameras because they can physically reconfigure at post-deployment stage.

3.4 Sensing Performance

Actuation modality enhances the coverage by providing wider field of view. The enhanced coverage is particularly beneficial if occlusion is significant in the environment. One can quantitatively demonstrate the sensing gain in two dimensional space using analysis similar to that in [24]. For the coverage comparison, we consider: $\theta_{static} = 43.5^\circ, \theta_{actuate} = 180^\circ, w = 128$ pixels. The gain in coverage enhancement is shown in Figure 1. As a rule of thumb, the gain factor is approximately the ratio of field of view between actuation-based and static-based schemes. The abrupt decrease is due to finite-pixel precision and blurring effect.

4 Methodology

In this section, we describe our methodology in fusing information from multiple actuated cameras for external calibration of the network. In section 4.1, we introduce a basic scenario which includes a network of three static cameras and further review the required modification to incorporate the impact of actuation. In section 4.2, we extend our approach to a large camera network by merging multiple calibrated triangles. We describe the challenges of merging calibrated triangles and illustrate the advantages of our actuation-assisted approach in terms of higher likelihood of triangle fusion. To improve the calibration accuracy by taking into account additional observations that is available due to actuation, we introduce our numeric optimization approach in section 4.3 where after fusion of basic triangles, the calibration accuracy can be continuously refined with each new observation. Throughout this paper we assume a simple pin-hole model as shown in Figure 2 where a projection of the target on the camera's image plane is described as a point in two dimensional space. In section 4.4 we study the effect of misalignment in the center of actuation and the center of projection of the camera which results in translational effect among images taken from a single camera(Figure 3).

4.1 Simplest-Case Network

Here we describe the method for external calibration of cameras within a simple three node network, and highlight the difficulties of meeting its constraints when using static cameras. We then introduce actuation-assisted schemes to overcome these problems.

Consider one simple triangle (Figure 4) of two cameras A and B and one beacon C. The goal of general camera network calibration problems is to determine a relative rotation matrix $R \in^{3\times3}$ between A and B, and translation vectors among A, B and C. In a static camera network, the localizability is directly achieved when each camera views the other camera and the beacon [21]. If only the projections of C on the image plan of A and B, respectively, are given, the triangulation cannot be accomplished due to ambiguous angular information of angle $\angle ACB$. If the mutual projection between A and B is also provided, the configuration can be determined up to scale. The mutual projection on the image planes of two cameras facing each other is called epipole. The corresponding triangulation based on epipoles is called epipolar geometry [26]. The simple triangle example also shows that a localizable triangle network forms the fundamental component in camera network calibration. When two cameras view each other, they obtain direct epipole measurements within their field of view. Consider the following visual measurements represented as unit vectors (Figure 4).

$$w_{ab}\ ,\ w_{ac}\ ,\ w_{ba}\ ,\ w_{bc}$$

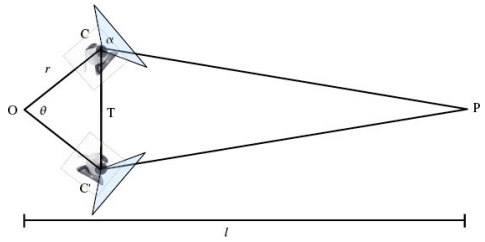

Figure 3. General actuation with offset.

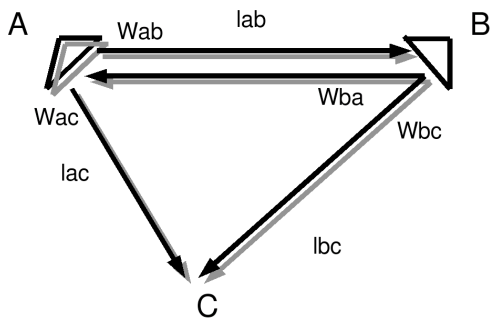

Figure 4. A simple triangle of two cameras and one beacon.

We aim to solve the length of three segments

$$l_{ab}\ ,\ l_{bc}\ ,\ l_{ca}$$

up to scale. By epipolar geometry, the direct epipole measurements can be used to derive the relative rotation R_{ab} between camera A and B [20, 21]. With R_{ab}, the observation in camera B's frame can be translated into camera A's coordinate, or vice versa. Specifically, from camera A's point of view, a close-loop equation can be formed by considering the sum of three vectors.

$$l_{ab}w_{ab}\ ,\ -l_{ac}w_{ac}\ ,\ l_{bc}w_{bc}^{(A)}$$

in which the third term represents the vector $\vec{BC}$ defined in camera A's frame. $l_{bc}w_{bc}^{(A)}$ is not directly observed by camera A but can be obtained from the transformation of w_{bc} via R_{ab}. An equation is thus formulated as follows from which the length of three segments l_{ab}, l_{bc}, and l_{ca} can be calculated.

$$l_{ab}v_{ab} - l_{bc}R_{ab}v_{bc} - l_{ca}v_{ac} = 0 \tag{1}$$

The limitations behind the simple network composed of static cameras are the confined field of view and pose rigidity, which further constraint the probability of forming a localizable triangle. Since the deployment is generally nondeterministic, in a static camera network, it is more difficult to guarantee a pair of cameras possess common feature points and the corresponding localizability.

In computer vision, a standard approach to the above problems is epipole estimation given that multiple common feature points are available. The required number of common feature points may vary depending on the applied methods [21, 16, 17] in which, generally, six or eight points are necessary to estimate epipoles. For example, based on the observations from multiple camera views, we can obtain the fundamental matrix from which epipole estimation is extracted [20], at the cost of more deployed cameras and/or beacons and more particular pose configuration setup. On the contrary, by exploiting actuation modality, we can transform a triangle with critical localizability to another easier case, as shown in Figure 5. Originally, two cameras share one common feature point C. (The shaded areas represent field of views.) The observations are defined in a reference frame (e.g. zero-PAN and zero-TILT). However, A (or B) fails to obtain direct epipole measurement (as shown in the heavily shaded area in Figure 5) because they do not face each other. In a static camera network, such configuration leads to a triangle with non-localizability. In an actuated camera network, each camera is potentially able to reconfigure its pose such that they view each other (as shown in the lightly shaded area in Figure 5) and obtain an epipole measurement **e** (defined in the new frame) regarding its neighbor. Then **e** can be transformed to another epipole **e'** (defined in the reference frame). Both epipoles can be related through a rotation matrix **R** and a translation vector **T**.

$$e' = \frac{R(e+T)}{\|R(e+T)\|_2} \tag{2}$$

To construct correc rotation matrices for PAN and TILT, we can assign two orthogonal coordinates for each pose respectively. If again considering Figure 2, we can, for instance, relate TILT to rotation about X-axis, and PAN to rotation about Y-axis. Any composite rotation consisting of PAN and TILT operations can be decomposed or synthesized as follows.

$$R(\phi,\theta) = R_y(\phi)R_x(\theta) \tag{3}$$

$$R_y(\phi) = \begin{bmatrix} cos\phi & 0 & -sin\phi \\ 0 & 1 & 0 \\ sin\phi & 0 & cos\phi \end{bmatrix}$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & sin\theta \\ 0 & -sin\theta & cos\theta \end{bmatrix}$$

Thus, relative pose can be equivalently recovered by performing direct epipole measurement method. In practice, due to the short rotation arm length, **T** might be minimal and only **R** has to be considered. Furthermore, actuation modality leads to improvements in the reduction of required number of cameras/beacons and the enhancement of field of view.

4.2 Large-Scale Network

To achieve external calibration across a network of greater than three cameras, triangles of locally calibrated nodes must merged into universal reference frame. If two adjacent localizable triangles share the same side and possess at least one camera whose orientation is known to both triangles, they

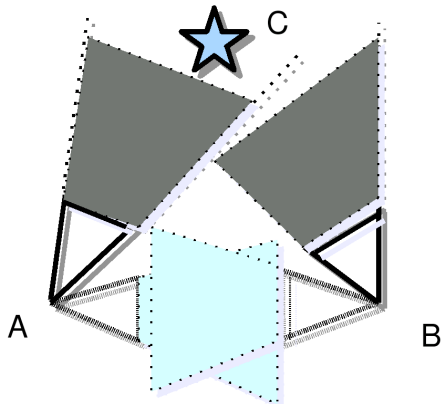

Figure 5. Transformation of a critical triangle by exploiting actuation modality. The heavily/lightly shaded area represents field of views established before/after camera poses are reconfigured.

can be merged into a common frame. Fusion of triangles can be demonstrated in Figure 6. Here, for simplicity, we individually consider observations defined in the reference frame of each camera. Each arrow represents the directional visibility between the observing camera and the target. As shown in Figure 6, two triangles ABC and BCD share one side formed by two cameras B and C. In this case, cameras B and C share two common feature points, cameras A and D. Before ABC and BCD are merged, the epipolar calibration method described in Section 4.1 is applied to each triangle. Ideally, there exists an unique relative rotation matrice R_{bc} which allows us to correlate observations in both triangles and determine the translations among all four cameras. In practice R_{bc} obtained within triangle ABC may not equal its counterpart in BCD, due to measurement error. By exploiting multiple observations, we can merge triangles and then apply Bundle Adjustment to recover a more consistent pose estimation, as described in Section 4.3.

Other circumstances in which triangles can be merged are shown in Figure 7 and Figure 8. When the triangles in Figure 7 are merged to form a cluster, the poses of cameras A, C, and D, and the positions of all cameras can be related within the same reference frame. Similarly, when the triangles in Figure 8 are merged, the poses of cameras A, B, and C, and the positions of all cameras can be related within the same reference frame. However, the network with visibility conditions shown in Figure 9 does not meet the constraints required to uniquely merge the local external calibration results of triangles ABC and BDC into a shared frame of reference. Notice that any geometrical configuration produced by folding the network along the edge BC will satisfy the given set of observations.

Intuitively, an actuated camera network is more likely than its static counterpart to satisfy the rules for merging triangles and hence form larger clusters of commonly calibrated cameras. We support this claim with simulation results in section 7. localizability.

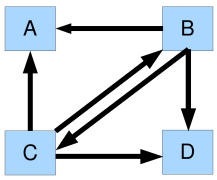

Figure 6. Actuation modality enables fusion of triangles.

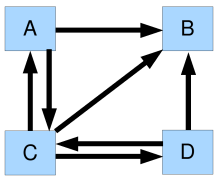

Figure 7. A second example illustrating fusion of triangles.

4.3 Bundle Adjustment Refinement

After triangles are merged, there might exist discrepency among observations from multiple camera views and pose estimates. We can consider the entire merged cluster and jointly refine pose configurations by utilizing Bundle Adjustment [22]. Multiple views within the cluster are related through certain set of rotation matrices and translation vectors which might have been estimated individually by applying the techniques in Section 4.1. To understand the principles of refinement steps, we first consider Figure 6 again and suppose the projection of A on camera B and C are $O_b^{(A)}$ and $O_c^{(A)}$, respectively. Before triangle ABC and triangle BCD are merged, within ABC, $O_b^{(A)}$ and $O_c^{(A)}$ are related through deterministic parameters $R_{bc}^{(A)}$ (rotation) and $T_{bc}^{(A)}$ (translation).

$$O_b^{(A)} = \frac{R_{bc}^{(A)}(O_c^{(A)} + T_{bc}^{(A)})}{\|R_{bc}^{(A)}(O_c^{(A)} + T_{bc}^{(A)})\|_2} \tag{4}$$

Similarly, suppose the projection of feature point D projection on camera B and C are $O_b^{(D)}$ and $O_c^{(D)}$, respectively. Within triangle BCD another relation can be described by another deterministic parameters $R_{bc}^{(D)}$ and $T_{bc}^{(D)}$.

$$O_b^{(D)} = \frac{R_{bc}^{(D)}(O_c^{(D)} + T_{bc}^{(D)})}{\|R_{bc}^{(D)}(O_c^{D} + T_{bc}^{(D)}")\|_2} \tag{5}$$

Ideally, the relative rotation $R_{bc}^{(A)}$ and $R_{bc}^{(D)}$ are equal. Due to calibration error, there is mismatch between the relative rotation $R_{bc}^{(A)}$ and $R_{bc}^{(D)}$. $R_{bc}^{(A)}$ and $R_{bc}^{(D)}$, as well as $T_{bc}^{(A)}$ and $T_{bc}^{(D)}$, thus need to be refined. We aim to find optimal R_{bc}^{optm} and T_{bc}^{optm} to minimize the discrepancy in the above equations. At the optimal point, R_{bc}^{optm} and T_{bc}^{optm} achieve balance between two sides of Equations 4 and 5, respectively, and the minimum inconsistency between Equations 4 and 5.

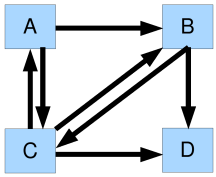

Figure 8. A third example illustrating fusion of triangles.

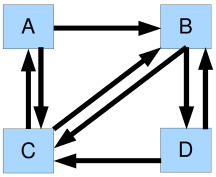

Figure 9. An example in which the configuration cannot be uniquely recovered in a global frame.

Before the refinement is applied, all observations, represented in unit vectors, should be well defined in each camera's reference frame. Triangles are emerged and the observations from multiple camera views are compared. The residual function f_R can be defined in the non-linear least square form as follows [22].

$$f_R = \sum_{i,j,k(i,j)} \| e_i^{(k)} - \frac{R_{ij}(e_j^{(k)} + T_{ij})}{\|R_{ij}(e_j^{(k)} + T_{ij})\|_2} \|_2^2$$

The subscripts i and j represent indices of cameras that can form a pair observing at least one common feature point k. The unit vectors $e_i^{(k)}$ and $e_j^{(k)}$ are the projections of point k onto the image plane of cameras i and j, respectively. The matrix R_{ij} is the relative rotation, and the vector T_{ij} is the relative translation. Based on the residual function and the associated optimization schemes (e.g. Newton's method), the pose configuration is refined after a sufficient number of iterations. Actuation modality provides more observations and equations and ensures the optimization problems better computational conditions. We thus claim that the implementation of actuated camera networks calibration, as future work, results in Bundle Adjustment convergence and precision improvement.

4.4 Alternative Single-Camera Approach

Conceptually, a single actuated camera is inherently able to incur self translation and rotation. The self-actuation capability leads to the fact that an actuated camera can externally calibrate itself by viewing a set of six unknown feature points. Unfortunately a mathematical analysis of the scenario shows that the translation is insufficient for providing useful calibration information.

Assume we have a pan/tilt camera which observes an object P from two different positions, **C** and **C'**, such that **P** is equidistant from each viewpoint (Figure 10). **O** represents the center of rotation for the pan-tilt unit, with r representing the pan-tilt arm length. θ represents the pan or tilt between each view, and α represents the camera's field of view. If we

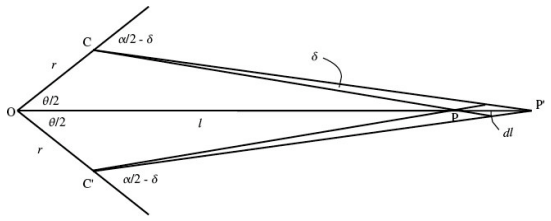

Figure 10. Illustration of pixel quantization error and depth estimation error in the alternative single-camera scenario.

obtain the relationship between the distance l to the target, based on **O** and **P**, and the estimation error, we can determine up to what range this system could be useful. With triangulation, range estimation error has two sources: pixel quantization error and the translation **T** between camera views. If we let δ represent field of view of an individual pixel, i.e. $\delta = \frac{\alpha}{w}$, where w is the image width in pixels, then dl represents the maximum error in the triangulated distance of the object (Figure 10). Qualitatively, we observe that a rotation arm of shorter length (r) results in larger error in depth (dl).

In practice, the rotation arm length is very short, and the translation is minimal. Equivalently, actuation of a single camera does not provide depth information.

5 System Description

Our system follows the general architecture of sensor networks and is composed of three types of wireless devices: actuated camera nodes, optical beacon nodes, and a micro-server (Figure 11). Each camera node is MicaZ mote equipped with a Cyclops image sensor [3] which acquires and analyzes a sequence of images from its neighborhood to identify and localize (in its local coordinate system) its beaconing neighbors. Each camera node also has an optical LED beacon, and external calibration can be achieved with a network consisting only of camera nodes depending upon their quantity and configuration. However, to reduce the overall cost of the system, one may also consider a heterogeneous network of camera nodes with both image sensing and beaconing capabilities and beacon nodes with only beaconing capability. The latter need not even be equipped with radios as long as they optically advertise unique ids, and their sole role is to provide easy to identify commonly viewed feature points between pairs of cameras.

The micro-server has greater computational resources than the actuated camera nodes, and is also responsible for network calibration service. It implements the external calibration service by coordinating progression of the calibration protocol, aggregating the local observation of the camera nodes, and executing the algorithms described in Section 4. The micro-server generates estimates for the external calibration of each camera, and the location of each beacon, in the network. Although, our current approach is centralized, it is quite scalable due to small size of calibration messages which include the identity of the neighbors of each node, their location in its local coordinate system, and the pose of the reporting node. Exploration of a distributed way to organize this computation should be feasible, as prior work on

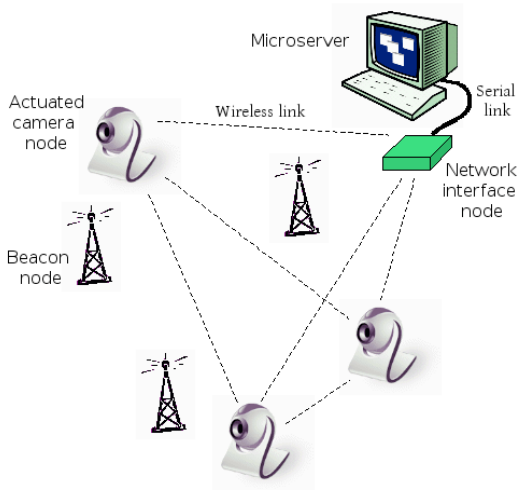

Figure 11. This figure illustrates our system components which consist of camera and optical beacon nodes and a computationally capable node for information fusion and network level external calibration.

localization suggests, but is a topic of future work.

We now describe an overview of the issues involved in the design of the actuation platform for the camera nodes and the implementation of the optical communication protocol.

5.1 Actuation Platform

There are various parameters that we have taken into account in the design of our actuation system including actuation precision, actuation delay, static and active power consumption. The actuation precision should be carefully chosen to minimize the error on the overall calibration performance. In particular, the precision of the angular repositioning should at least be better than worst case angular precision of each node's pose measurement (i.e. at maximum peer distance). In addition, to guarantee minimal accumulation of pose error over time, our actuation platform performs reference pose calibration by periodically revisiting the zero pan and tilt pose at mechanically-enforced hard limit points.

Since the actuation model consists of infrequent short actuation and long inactive episodes, we have designed an actuation platform with minimal power consumption during the inactive periods. This guarantees an overall minimum power consumption with little impact on the lifetime of the camera nodes since they are predominantly in a state of inactive actuation. To achieve this, we have picked an actuation system with a miniature gearbox which holds its pose against gravity (for the current mechanical load of the sensor node) without consuming active power. The price we pay for this is an extra power consumption during the active actuation state due to additional gearbox mechanical load. However, this is acceptable since we seek to optimize the platform energy consumption for the infrequent pose reconfiguration case.

Figure 12 shows a photograph of our prototype actuation platform that carries a pair of Cyclops/Mote. It is capable of 180 degrees rotation on both pan and tilt axis and has 1° angular repositioning error. The actuation platform consumes 800mA and 0mA during active and inactive mode respectively and actuates at an angular speed of 315 degrees/second in both axis. We have also included an accelerometer and a magnetometer to provide supplementary orientation information with the accelerometer giving the orientation relative to earth's gravitational field and the magnetometer giving the orientation relative the earth's magnetic field. This could be used as an initial value by certain iterative estimation algorithms, although the technique described in this paper does not make use of this hardware capability.

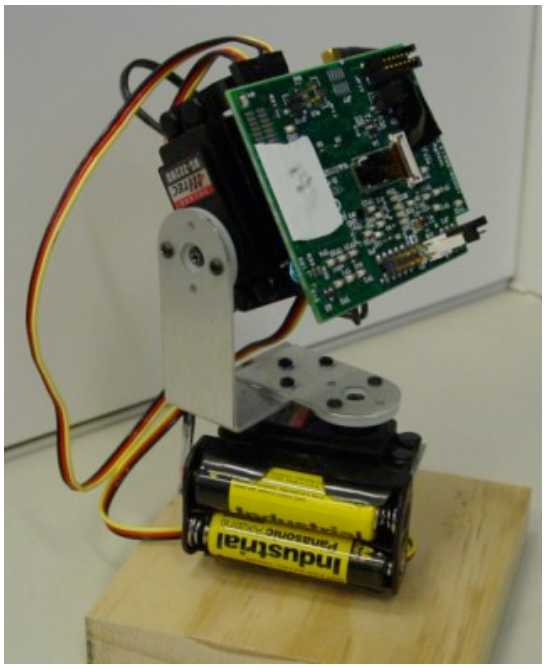

Figure 12. The actuation platform and pair of Mote/Cyclops. The LEDs on Cyclops are used for optical transmission of the node identity.

5.2 Optical Communication for Beaconing

As mentioned before, we use an active beaconing approach to identify commonly observed features between pairs of cameras. Unlike passive approaches, this provides robustness against ambient conditions and simplifies the needed image processing to identify the features. The beacon should be easily observable, and should also be distinguishable (i.e. one should be able to identify the beacon id). We use a pure optical communication approach based on a blinking LED on each beacon, with the neighboring cameras observing the LED over a sequence of images. Beacons send an optical message which consists of a preamble, their unique ID, and a Cyclic Redundancy Check code. The receiving nodes collect images at a frame rate which is reasonably higher than the transmission rate and make successive frame differencing to extract the location of active beacons (in the camera coordinate system) and their raw optical bit stream. They subsequently decode the information to extract the identity of each beaconing node and its location in their local coordinate system. We note that an alternative to purely optical beaconing would be radio assisted optical beaconing whereby radio communication would be used to give trans-

mit token to one of the beacons and to share the identity of that beacon with cameras. With this the optical processing could be made simpler. While our current hardware which uses the LEDs on MicaZ motes is capable of such an approach, we decided to use a purely optical approach because it would permit the use of lower-cost radio-less optical beacons as well.

In the remainder of this paper for analysis and simulation we assume that beaconing nodes offer an omnidirectional view and we only constrain the field of view of the cameras. While this assumption does not hold true in our current platform due to the directivity of the LEDs on the mote, we note that it can easily be implemented by having multiple beaconing LEDs in the platform. However, we briefly mention the implication of beacons with constraint field of view on the overall performance of our external calibration system.

6 External Calibration Architecture

In this section we present our actuation-assisted external calibration system which utilizes the methodologies described in Section 4. Recall that the objective of our system is automatic external calibration of an arbitrarily deployed network of actuated cameras to support location-sensitive network applications. The first step in this process is for cameras to acquire their neighbors coordinates by taking a sequence of images. While this is a simple task in a static system, which can be simply synchronized by the network micro-server, it requires a series of coordinated observations in an actuated system. Additionally, we require rules to indicate when the observation process should be aborted. In this section we provide an overview of these issues and their impact on the latency and power consumption of the calibration services.

6.1 Actuation Strategies

In our system of actuated cameras, each camera can search for additional nodes by changing its orientation to monitor a new region of space. However, the strategy employed by cameras to rotate through their range of motion can affect the quantity of observations acquired, and the latency and energy cost of the system. In the case of a network with omni-directional beacons, cameras in our system will perform a two dimensional *quasi-raster* scan. For example, suppose a camera has a pan range of $\theta_p^{max} \leq 2\pi$ and a tilt range of $\theta_t^{max} \leq \pi$. The camera begins the search by monitoring its view at a reference pose of zero degrees pan and tilt. Following each observation time interval, the camera pans a step $\delta\theta_p$ in the positive direction, until it reaches a pan of θ_p^{max}. Then it tilts a step $\delta\theta_t$ in the positive direction and proceeds to pan back in the negative direction. This process continues until the camera has monitored its view at a pose of θ_p^{max} pan and θ_t^{max} tilt, thus guaranteeing that it will observe all visible nodes in its vicinity. The quasi-raster strategy is time and energy efficient as well. Equation (7) provides an estimate of the time required to perform a full quasi-raster scan where ω represents the angular velocity of the actuator.

$$g(\theta) = \left\lceil \frac{\theta^{max}}{\delta\theta} \right\rceil + 1 \tag{6}$$

$$T = T_{actuation} + T_{observation} \tag{7}$$

$$= \frac{1}{\omega}\left(\theta_t^{max} + g(\theta_t)\cdot\theta_p^{max}\right) + g(\theta_p)g(\theta_t)\tau_{obs} \tag{8}$$

The maximum angular step size equals the camera's field of view θ_{fov}. However, this means some nodes may be observed only near the image's borders where pixel quantization error is greatest. Hence applications with stringent localization accuracy requirements and a tolerance for latency and energy costs may opt for smaller step sizes. The observation time constant τ_{obs} is directly proportional to the bit length of the optical identification packet and inversely proportional to the camera frame rate. Equation (9) provides an estimate of the energy required to perform a full scan where P_{act} represents the power consumed by the actuator while in motion and P_{obs} represents the power consumed by the camera and LEDs while capturing images.

$$E = \frac{P_{act}}{\omega}\left(\theta_t^{max} + g(\theta_t)\cdot\theta_p^{max}\right) + g(\theta_p)g(\theta_t)\tau_{obs}P_{obs} \tag{9}$$

In the case of a network with directional beacons, the quasi-raster scan strategy is not a viable option. Due to the uniform progression of the quasi-raster scan strategy, two actuated cameras located within viewing range of each other may never be able to observe the other regardless of search duration[2]. To avoid this issue, we may use a *random* scan search strategy in networks with directional beacons. When performing a random scan, a camera pans and tilts at randomly selected step sizes after each observation. This will guarantee that cameras ultimately observe each other in a directional beacon system if they are within acceptable distances. However, the problem with the random scan search is that there is no upper bound on the amount of time for the two cameras to observe each other. Furthermore, $\delta\theta_p$ and $\delta\theta_t$ can be greater than θ_{fov}, which yields an inefficient actuation path compared to the quasi-raster scan.

6.2 Actuation Termination Rules

Each time a camera identifies the modulated light pattern of a node in its field of view, it reports the observation to a micro-server. An observation message consists of the following data: ID of reporting node, ID of target node, image plane coordinates of target node, and pan and tilt angles of reporting node. When the micro-server receives each observation, it incorporates the new information into its database of previously acquired observations. Then it executes the algorithms in Section 4 using the updated information to generate a more complete and accurate external calibration model of the network. The localization protocol continues in this manner until certain stop rules are met.

Our system uses two types of stop rules: *local* rules which are evaluated and followed at each camera, and *global* rules which are evaluated at the micro-server and followed network-wide. For a network with omni-directional beacons, the localization protocol is terminated when any one of the following four rules are met.

[2]To conceptualize this, consider a two dimensional example where two actuated cameras looking at opposite directions with in phase actuation schedule.

Local *Exhaustive*: For this rule to be met, a camera must monitor all space in its viewable domain. For instance, once a camera completes a full quasi-raster scan it will cease searching for new node observations. This guarantees that a camera will collect all possible node observations if and only if we can assume that the camera's surrounding environment is time-invariant. This rule serves as a worst-case scenario for localization costs because once this rule is met by every camera, the network should not be able to acquire any additional observation.

Neighborhood: For this rule to be met, a camera must observe each of its neighboring nodes. Here we define two nodes to be neighbors if they can engage in single-hop radio communication. This rule is only useful under the assumption that our system's radio channel has a greater range than does its optical channel. This is often reasonable given the constraints on image resolution and optical beacon intensity imposed by the typical energy, financial, and form factor limitations of sensor networks. In a sparse network, this rule has the potential to reduce energy and latency costs because it is unlikely that a camera will observe another node at each of its possible poses. Nonetheless given the greater range of radio communications and the fact that occlusions often affect only optical transmissions, oftentimes this stop rule may never be met.

Global *Complete Network Fusion*: For this rule to be met, the micro-server must receive enough observations to localize the entire network into a common frame of reference (i.e. complete fusion of triangles as described in Section 4.2). Notice that this rule may be met before both the exhaustive and neighborhood rule and hence has the potential to reduce energy and latency costs even further. However, depending on deployment characteristics, the network may never generate sufficient observations for complete localization under a common frame of reference and hence this rule may never be met.

Negligible Marginal Change: As described in Section 4.3, many factors of a real-world deployment can cause the external calibration estimates produced by the triangle fusion algorithm to be erroneous. Recall that if the system acquires extra observations, bundle adjustment refinement can be used to converge on a more accurate model. Ideally, we would like our system to halt the localization protocol once the average localization estimation error within the network is below some threshold, but accuracy cannot be determined without knowledge of ground truth positions and poses. The negligible marginal change stop rule represents a close approximation to this ideal. For this rule to be met, a consecutive sequence of newly reported observations must not lead to the external calibration of any new nodes and must not refine any position or pose estimate within the network by greater than some maximum threshold.

The same set of stop rules can not be used for a network with directional beacons. In particular, obeying the local rules may cause the localization protocol to terminate prematurely. For example, consider the exhaustive rule. With directional beacons, even if a camera monitors its entire actuation domain, it may not observe some cameras because they are pointed in the wrong direction. Next consider the neighborhood stop rule. Even if a camera observes all its neighbors, some of its neighbors may not have observed it. If the camera were to stop actuating and emitting its identifying light pattern, it may never be observed by some neighbors. Hence, for networks with directional beacons, our system obeys either of two global stop rules.

7 Evaluation

In this section we evaluate the performance of our actuation-assisted system with respect to the following metrics: required node density, external calibration accuracy, and external calibration latency. To perform these evaluation we use both a simulation environment and a testbed experimentation environment. We developed the simulation environment in Python making heavy use of the Numeric Extensions package (NumPy). The simulator allows us to generate camera networks (static or actuated), acquire observations, execute external calibration algorithms, and visualize the results. This is especially useful for analyzing network scenarios that are hard to realize in practice. Our testbed experimentation environment consists of actuated Crossbow MicaZ motes with attached Cyclops image sensor, and running the SOS operating system on both MicaZ and Cyclops. We use these nodes to capture real-life images, which are processed to make node observations. These observations are imported into our simulation environment to for executing and verifying external calibration algorithms.

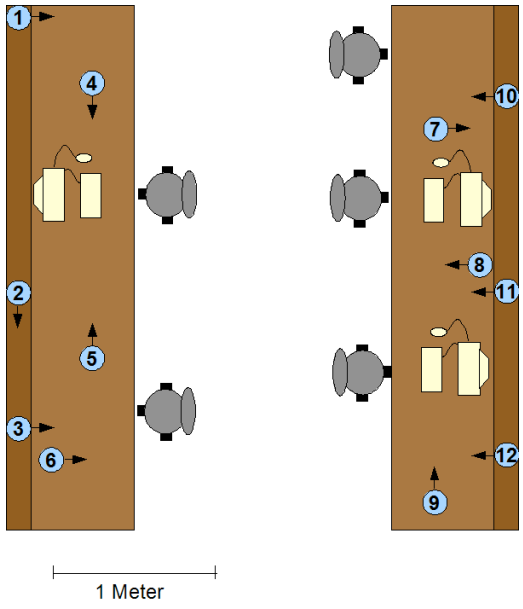

Figure 13. Topology of actuated camera network testbed used in experimentation

7.1 Node Density

The goal of this experiment is to determine the relationship between the spatial density of camera nodes in a network and the likelihood of externally calibrating these cameras in a common frame of reference. Specifically we compare the results obtained from a network of static cameras to those obtained from a network of actuated cameras. Here we refer to node density as the ratio of the number of nodes

contained within a space, to the volume of that space. Due to the enormously large number of nodes and unique network configurations required to produce statistically meaningful results, we implemented this experiment in our simulation environment.

In this experiment, we simulate a network of cameras with random locations and orientations in a fixed space of 3x3x3 meters, the approximate size of a small room. We chose these dimensions because three meters is the range at which a Cyclops camera can distinguish the particular LEDs we use as beacons. We also assume that each camera has a 45° horizontal and vertical field of view, and is equipped with an omnidirectional beacon. To analyze the case in which all cameras are static, the cameras monitor the view from only their reference poses. Then the simulator performs the local epipole and triangle fusion algorithms on the observations to determine the number of cameras externally calibrated under a common frame of reference. Next we use the same network configuration to analyze the case in which all cameras are actuated. We assume these cameras have a range of 180° in both the pan and tilt direction, which are the same capabilities as our custom pan-tilt platform. The cameras employ the quasi-raster scan described in section 6.1 to monitor the environment, and then the acquired observations are used to determine the number of externally calibrated cameras. This experiment begins with a network of three cameras, and repeats this process for networks up to 1000 cameras in size.

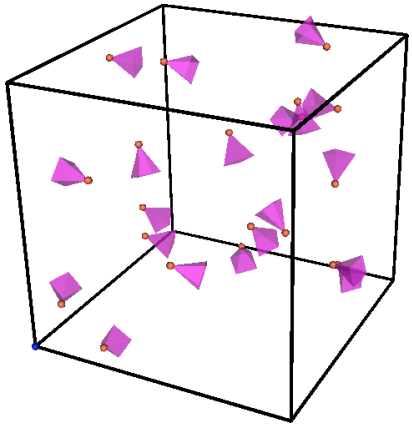

Figure 14. Example topology of simulated network in node density experiments as described in 7.1. Network consists of 20 cameras deployed with random location and pose within 3 meter cube. Cameras have 3 meter viewing range.

It is important to note that the correlation between node density and feasibility of external calibration is not one-to-one. In the case that cameras have directional fields of view and restricted actuation, the feasibility of external calibration also depends upon the initial relative orientations of cameras. For example, one can imagine a situation in which two cameras located in close proximity cannot be calibrated because they are oriented in opposite directions, while two cameras located farther apart can be calibrated because they are facing each other and observe another node in common. To account for the effects of orientation, for each network size that we test, we generate 10 unique network configurations and compute an average for the number of cameras that can be externally calibrated in each.

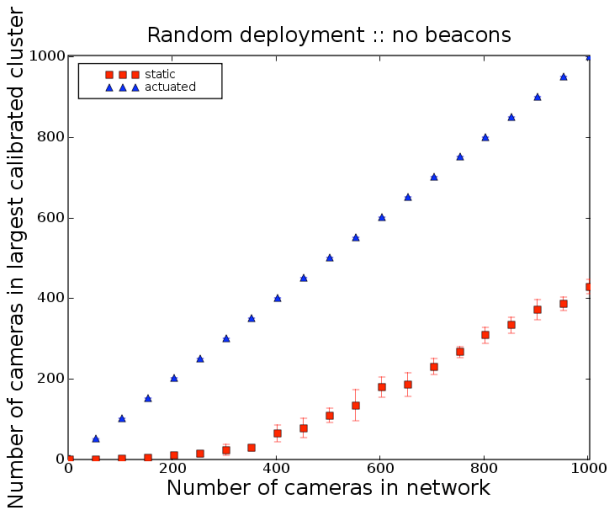

Figure 15. Number of externally calibrated cameras in largest cluster versus total number of cameras in a randomly deployed, simulated network. A cluster refers to a group of cameras externally calibrated relative to the same reference frame.

Figure 15 shows the results from the simulated test runs. The number of cameras in the largest externally calibrated cluster is used as a representative metric for the extent to which external calibration succeeded at a given node density. Clearly, the system of actuated cameras significantly outperforms the system of static cameras. For all node quantities displayed, actuated camera networks are completely calibrated. Conversely, static camera systems never achieve a cluster size greater than 50% of the network size, and this best case scenario only occurs for absurdly large network sizes. The results are even more drastic for smaller, more practical network densities. For systems of 15 actuated cameras or less, at worst, 50% of the cameras in the network are calibrated relative to a common frame of reference. For systems of greater than 15 actuated cameras, at least 90% of the cameras in the network are always calibrated in a unified cluster. Contrast this with static camera systems in which a maximum cluster size of only three cameras is achieved for networks up to 100 cameras in size.

Since previous work on external calibration of static cameras relies on the presence of beacons [21, 22], we performed another version of the experiment in which the simulation space is randomly scattered with 200 beacon nodes. In order to compare the results with those of the previous experiment we do not count a priori localized beacons as part of a cluster. Essentially they exist as extra feature points for the cameras. Figure 16 shows that the additional beacons yield a minimal increase in the largest cluster size of externally calibrated static cameras.

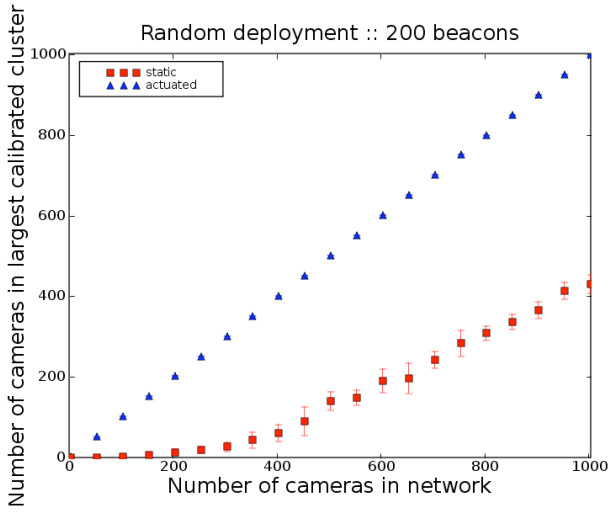

Figure 16. Number of externally calibrated cameras in largest cluster versus total number of cameras in a randomly deployed, simulated network with 200 additional beacon nodes.

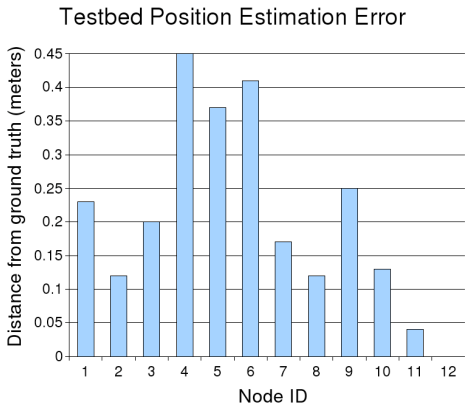

Figure 17. Calibration results driven by images captured in experimentation environment (testbed of 12 actuated cameras). Error represents the distance from ground truth position to estimated position. Note that camera 12 has no error because we arbitrarily chose it as the anchor from the network frame of reference to the global frame of reference.

7.2 External Calibration Accuracy

The goal of this subsection is to evaluate the estimation accuracy of our external calibration system. To accomplish this we performed our external calibration algorithms both on our experimentation testbed and in our simulation environment. A row of laboratory cubicles served as the testbed for our experimentation. 12 actuated cameras were placed along the bottom and top shelves of two desks on opposite sides of the cubicle row. The deployment covered a 9 m^2 area and the shelves are separated by a height of one meter. Note that it is not possible for two cameras located on the same desk but different shelves to view each other. Figure 13 represents an overhead view of our testbed topology. The measured ground truth position of each camera is marked by a numbered circle, and the reference pose of each camera is indicated by an arrow.

Figure 17 shows the external calibration error, represented as the distance from ground truth to the estimated position, for each node. The magnitude of the errors we encountered in our current experiments is larger than originally anticipated, can be attributed to several factors. One likely source is the mechanical imprecision of the actuation platform. Another source of error is the pixel quantization effects introduced by low resolution images sensors used in Cyclops. Furthermore our pin-hole camera model is not a true representation of a Cyclops camera. For this experiment, cameras were externally calibrated using the triangle fusion algorithm. Hence error is accumulated as locally calibrated triangles are stitched together into common frame of reference. Use of refinement optimization should reduce error greatly, and is under implementation. Lastly, we acknowledge the possibility of human error in the measurement of ground truth locations in the 3D space.

In addition, we evaluate the external calibration accuracy of our system on a larger simulated network. In this experiment 30 actuated cameras are scattered with random locations and poses in a 6x6x3 meter space. As in the node density experiments from section 7.1, cameras have a three meter viewing range, 45° field of view, and 180° pan and tilt range. We use both the quasi-raster and random actuation strategies to generate two independent sets of observations. We then use the local epipole and triangle fusion algorithms to perform external calibration. Figure 18 shows the mean estimation error versus the time duration of the calibration routine. Using quasi-raster strategy, the mean position estimation error settles to only 3 cm once all cameras are calibrated. This error occurs because our simulation environment includes quantization noise in observations.

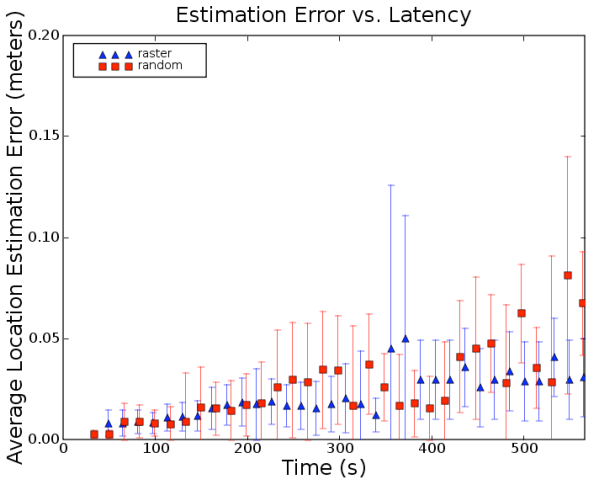

Figure 18. Relationship between duration of calibration protocol and average location estimation error for calibrated cameras.

7.3 Latency

As previously stated, a significant cost of actuation assistance is increased latency. Section 6 discussed how our

choice of actuation strategies and termination rules can affect calibration latency. In this section we evaluate the relationship between the time duration of the external calibration protocol and the effectiveness of external calibration. Figure 19 shows the number of cameras in the largest calibrated cluster as a function of time, for the network simulation described in section 7.2. For the same network topology, we acquired two independent sets of observations using both the quasi-raster and random actuation strategy. At each actuation step in the calibration process, we perform the local epipole and triangle fusion algorithms on the observations recorded up to that point.

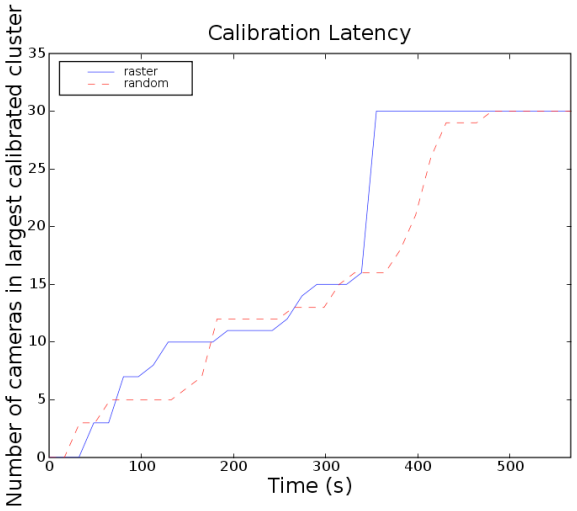

Figure 19. Relationship between duration of external calibration protocol and number of cameras in largest calibrated cluster for quasi-raster and random actuation strategies.

The elapsed time is calculated based on the time spent monitoring each pose and the time spent actuating from pose to pose. For this experiment, we make the following assumptions: an optical identification packet is 2 bytes long (1 byte for preamble and synch code, and 1 byte for node ID), a camera can capture images at a rate of 1 frame per second, and the actuator has an angular velocity of 315° per second. The raster search strategy uses an actuation step size equal to the camera field of view of 45 degrees. For the random search strategy, each time a camera actuates, it selects its step size at random. Since all cameras must transmit and monitor the optical channel at the same time in order to make observation, some sense of actuation step synchronization is required. Hence for the random strategy, each camera must behave as though it actuates the maximum amount, 180°, even though it may not be. As it turns out, this added latency affects overall localization latency by only a small amount because the maximum per step actuation time is still less than the per step observation time by more than an order of magnitude. According to the graph, the 'complete network fusion' stop rule is met after 355 seconds when using raster strategy, and is met after 480 seconds when using the random strategy. When using the raster actuation strategy the 'exhaustive' stop condition is met at each node after 400 seconds. In general, these delays will vary with network topology.

8 Conclusion

In this paper we presented a methodology, system architecture and prototype platform for actuation assisted vision-based external calibration (i.e. determining 3D location and 3D orientation) of a network of spatially distributed cameras. In our system, the network consists of actuated camera devices which are also equipped with an optical beacon. During the calibration period, each node continuously sends a unique optical message and actuates its camera to intercept the optical messages of its neighboring nodes. Furthermore, each node determines the pose of its neighbors in its local coordinate system and sends that information to a fusion point where the information is aggregated to build a unified coordinate system.

We believe that we have made a first attempt toward exploiting actuation of cameras for external calibration of a camera network. Our experimental investigation illustrates a significant reduction in the required network density vs. a static camera network. However our results are far from comprehensive and future works include experimental comparison of calibration accuracy vs. static network of camera. In addition, we plan to further investigate the impact of various actuation strategies on the latency and energy cost of our system. This is particularly important in cases which involve more frequent calibration of the network, for instance, due to sensors mobility. An avenue of further research is investigating a hybrid case which involves a heterogeneous combination of static and actuated cameras on mobile objects and infrastructure. Likewise, another avenue would be to consider a combination of our approach of actuated cameras with the mobile beacon approach of [23].

In this paper, we have taken a centralized computation approach, in which, a more computationally capable node such as a network microserver collects and aggregates sensors information. While this a reasonable approach which has also been advocated by sensor network architectures such as Tenet [27] for reasons of robustness and ease of development, it would nevertheless be interesting and useful for certain scenarios to investigate a distributed version of our algorithm for larger networks. Prior research on the closely related problem of distributed node localization in sensor networks suggests that this should be feasible.

9 References

[1] M. Chu, P. Cheung, and J. Reich. Distributed attention. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 313–313, 2004.

[2] A. Kansal, W. Kaiser, G. Pottie, M. Srivastava, and G. Sukhatme. Virtual high-resolution for sensor networks. *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 43–56, 2006.

[3] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: in situ im-

age sensing and interpretation in wireless sensor networks. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 192–204, New York, NY, USA, 2005. ACM Press.

[4] A. Savvides, CC Han, and MB Srivastava. Dynamic fine-grained localization in ad-hoc wireless sensor networks. *Proc. International Conference on Mobile Computing and Networking*, 2001.

[5] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, 2000.

[6] L. Girod, M. Lukac, V. Trifa, and D. Estrin. The design and implementation of a self-calibrating acoustic sensing platform. *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 71–84, 2006.

[7] J.C. Chen, K. Yao, and R.E. Hudson. Acoustic Source Localization and Beamforming: Theory and Practice. *EURASIP Journal on Applied Signal Processing*, 2003(4):359–370, 2003.

[8] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS) using AoA. *Proceedings of the IEEE Infocom*, 2003.

[9] K. Chintalapudi, A. Dhariwal, R. Govindan, and G. Sukhatme. Localization Using Ranging and Sectoring. *Proceedings of the IEEE Infocom*, 2004.

[10] P. Bahl and VN Padmanabhan. RADAR: an in-building RF-based user location and tracking system. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2, 2000.

[11] T.S. Rappaport. *Wireless Communications: Principles and Practice*. IEEE Press Piscataway, NJ, USA, 1996.

[12] M. Maroti, B. Kusy, G. Balogh, P. Volgyesi, K. Molnar, A. Nadas, S. Dora, and A. Ledeczi. Radio interferometric geolocation. In *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys '05)*, pages 1–12, New York, NY, USA, 2005. ACM Press.

[13] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95, New York, NY, USA, 2003. ACM Press.

[14] S. Park, I. Locher, A. Savvides, M.B. Srivastava, A. Chen, R. Muntz, and S. Yuen. Design of a wearable sensor badge for smart kindergarten. In *Proceedings of the 6th IEEE International Symposium on Wearable Computers (ISWC '02)*, page 231, Washington, DC, USA, 2002. IEEE Computer Society.

[15] N.B. Priyantha, A. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 1–14, New York, NY, USA, 2001. ACM Press.

[16] WE Mantzel, H. Choi, and RG Baraniuk. Distributed camera network localization. *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, 2, 2004.

[17] D. Devarajan, R.J. Radke, and H. Chung. Distributed metric calibration of ad hoc camera networks. *ACM Trans. Sen. Netw.*, 2(3):380–403, 2006.

[18] H. Lee and H. Aghajan. Collaborative self-localization techniques for wireless image sensor networks. *Proc. of Asilomar Conference on Signals, Systems, and Computers*, 2005.

[19] H. Lee and H. Aghajan. Vision-enabled node localization in wireless sensor networks. *COGnitive systems with Interactive Sensors (COGIS 2006)*, 2006.

[20] D. Lymberopoulos, A. Barton-Sweeney, and A. Savvides. Sensor Localization and Camera Calibration using Low Power Cameras. Technical report, Yale ENALAB Technical Report 08050, 2005.

[21] A. Barton-Sweeney, D. Lymberopoulos, and A. Savvides. Sensor localization and camera calibration in distributed camera sensor networks. *Proceedings of IEEE Basenets, October*, 2006.

[22] C.J. Taylor and B. Shirmohammadi. Self Localizing Smart Camera Networks and their Applications to 3D Modeling. *Proceedings Workshop on Distributed Smart Cameras (DSC 2006)*, 2006.

[23] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar. Distributed localization of networked cameras. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 34–42, New York, NY, USA, 2006. ACM Press.

[24] A. Kansal, E. Yuen, W.J. Kaiser, G.J. Pottie, and M.B. Srivastava. Sensing uncertainty reduction using low complexity actuation. *Proceedings of the third international symposium on Information processing in sensor networks*, pages 388–395, 2004.

[25] A. Kansal, J. Carwana, W.J. Kaiser, and M.B. Srivastava. Coordinating camera motion for sensing uncertainty reduction. *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 307–307, 2005.

[26] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[27] O. Gnawali, K-Y Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler. The tenet architecture for tiered sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys '06)*, pages 153–166, New York, NY, USA, 2006. ACM Press.