FUSION-BASED LOCALIZATION FOR A HETEROGENEOUS CAMERA NETWORK

Marci Meingast[†*], Manish Kushwaha[‡], Songhwai Oh [§], Xenofon Koutsoukos[‡]
Akos Ledeczi[‡], Shankar Sastry[†]

[†] Electrical Engineering and Computer Sciences, University of California, Berkeley; CA 94720, USA
[‡] Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37203, USA
[§] Electrical Engineering and Computer Sciences, University of California, Merced; CA 95344, USA

ABSTRACT

Heterogeneous Sensor Networks (HSNs) are becoming more commonly used for purposes such as monitoring and surveillance, as they offer richer sources of data for situational awareness. An important aspect of HSNs is localization. In this paper, we describe a novel method for localizing a network of cameras equipped with wireless radios. Our method fuses both the image data and radio interferometry data in order to determine the position of the sensors and the orientation of each camera's field of view. While existing methods that rely solely on image data alone are often limited in that they can only recover position up to scale factors, by fusing the image data and radio interferometry data, we are able to recover the position and orientation with no scale factor ambiguity. In contrast, localization of sensor nodes using radio alone only recovers the position of the sensors and often relies on computationally expensive methods. The method discussed in this paper exploits both the image and radio data for a more computationally efficient process of localization. We discuss both a linear and nonlinear approach to fusing the data which depend on different constraints on the network. We demonstrate our approach on a real network of camera and radio nodes.

Index Terms— localization, cameras, sensor networks, heterogeneous, data fusion.

1. INTRODUCTION

Advances in camera technology have led to the concept of camera networks. These networks consist of a number of cameras, which have sensing, communication and processing capabilities. Cameras offer a rich source of data and are being used in applications such as surveillance [1, 2], intelligent environments [3, 4], and traffic monitoring [5, 6, 7].

In order to fully take advantage of the measured data, it is important to perform localization and compute the position of the cameras in the network as well as the orientation of their fields of view. By localizing the cameras, the image data becomes more useful as the relation of data from one sensor to data from another is known and tasks, such as fusion, power hand-off, and tracking, can be better performed. Localization aids in data analysis and helps the network operate more efficiently. The use of localization information aids in understanding what portions of the scene are observed by multiple cameras. As cameras are power consumptive and require a high-bandwidth communication medium, bandwidth and energy constraints can be eased by exploiting this overlap and selectively choosing a scene from one camera for data transmission when multiple cameras observe similar portion of the same scene. Manually measuring the pose (location and orientation) of all cameras in the network is a very tedious and time consuming task and sometimes requires special environmental conditions that may not be present, such as special lighting conditions [8]. Thus, a generalizable automatic method to localize the cameras in the network is of paramount importance for the success of HSNs.

The use of automatic feature correspondences between overlapping cameras combined with prior knowledge about the 3D distance between these features, or 3D locations of these features, would allow us to fully localize the network. However, as camera networks are used in many uncontrolled environments, it is often difficult to obtain this 3D information beforehand without the manual intervention. With just the image data, and no prior knowledge on the 3D scene, we can still detect features and use feature correspondences between images to determine orientation, but the position can now only be determined up to a scale factor. By using the unknown internal parameters of the camera and enough feature correspondences, the epipolar geometry allows the essential matrix of the camera to be determined [9]. The orientation of the cameras can be determined from the essential matrix, but the position can only be determined up to a scale factor. Due to the fact that there is no prior knowledge of the 3D scene or the geometry of the cameras, the scale factor ambiguity is intrinsic. It cannot be disambiguated whether the cameras are located at twice the distance, for example, from one another looking at a scene twice as large and two times further away or if the cameras are located at half the distance, looking at a scene half as large and half as far away.

*Corresponding author. Email: marci@eecs.berkeley.edu.

978-1-4244-2665-2/08/$25.00

While there are automatic methods for feature point correspondences in a camera network, many need prior knowledge of the environment or certain camera parameters or cannot handle wide-baseline cameras. These artificially imposed requirements limit the application domain of camera networks. For example, in [10], a common set of static scene feature points is assume to be seen by each set of three cameras. In [11], the height of the cameras is already known as well as two orientation parameters. In [12, 13] a single object is tracked in a common ground plane and it is already known how each camera locally maps to the ground plane. This ground plane concept is extended in [14, 15], where the relation of the local ground plane to the camera is not known, but the localization problem is solved for based on a single tracked objects and then these local ground planes are aligned to a global ground plane based on homographies.

Without making assumptions on the scene or the objects seen by the cameras, the method proposed in [16] uses tracks of objects in the image plane as (spatio-temporal) features to do correspondence between cameras. This method is well suited for localizing a camera network in an uncontrolled environment due to realistic and fewer requirements and its ability to handle wide-baseline cameras. But the method still cannot determine the scale factor for the position of the camera nodes due to the fundamental limitations in image geometry.

Image data alone is not sufficient to do automatic localization of a camera network in an unknown environment. It is necessary to incorporate another type of data in order to fully localize the network. For low-bandwidth sensor networks, a number of automatic localization techniques has been developed, using acoustic information or radio frequency intensities and exploiting received signal strength indicators, time of arrival, time difference of arrival, or angle of arrival to determine positions of nodes [17]. These methods cannot provide all the localization parameters necessary for cameras, as there is no sufficient data to determine the orientation of the field of view.

In this paper, we present a novel automatic multi-modality localization method which leverages both the high-level image data and low-level radio information. Feature correspondence information is found using an extension of [16] so that the orientation and position, up to scale, are found for a pair of cameras with overlapping view. The algorithm automatically determines whether a pair of cameras has an overlapping view. We then use the position vectors up to scale from the image data and combine it with the radio interferometry based approach described in [18]. We demonstrate both a linear and a non-linear approach to fuse the image data and the radio interferometry data in order to fully localize the camera network. This Fused-Based Localization (FBL) can recover both the orientation of camera's fields of view as well as their complete positions.

This paper organization is as follow: In Section 2, we formally state the localization problem and give an overview of the Fused-Based Localization (FBL) method. The components of FBL method are based on the single modality camera localization method from [16] and the single modality radio interferometry-based method from [18] and they are described in Section 3. In Section 4, we describe both linear and nonlinear fusion methods for combining image data and radio interferometry data. We demonstrate and evaluate the proposed method in simulation and on data collected from outdoor experiments in Section 5.

2. AN OVERVIEW OF FUSION-BASED LOCALIZATION

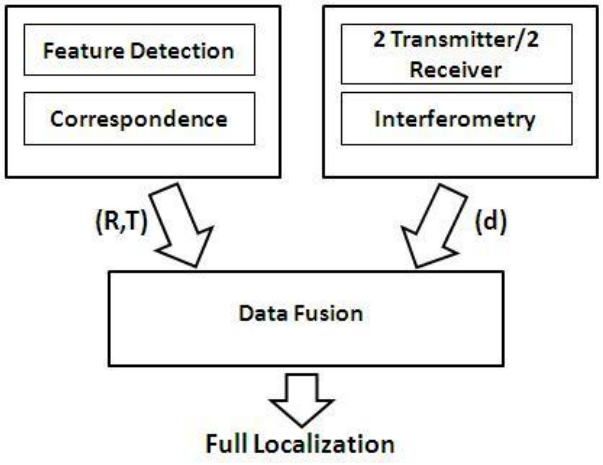


Fig. 1. The structure of the data fusion localization method.

Suppose that we have N cameras in the network. We assume that all the cameras are time-synchronized and each camera is equipped with a radio for wireless communication. Each camera is assumed to have the capability of detecting features, F, in the scene as well as having some overlap in what it sees with another camera in the network for correspondence. While feature correspondence in image data can be used to determine the orientation of a camera relative to another camera that overlaps, the position can only be determined up to scale due to the geometry constraints on the epipoles. Due to this, multiple set of possible positions of cameras will results from the same set of images.

The omnidirectional radios on a wireless node can transmit and receive data but they do not provide directional information. By communicating with one another, the radio nodes can use signal interference to determine linear combination of distances between groups of nodes. Note that the orientations of cameras can not be solved using radio data alone.

We propose a Fusion-Based Localization (FBL) method that can solve for the complete localization of a camera network, finding both the orientation of the fields of view of each camera as well as the positions of cameras including the scale factor, by using q-ranges, $\{d_{ijkl}\}$, computed from the radio

interferometry; and the orientations, $\{R_{ij}\}$, and positions up to scale, $\{T_{ij}\}$, from the image data feature correspondences. The structure of the FBL method is shown in Figure 1.

From the image data from each camera, tracks of moving objects in the scene are built in the image plane using the multi-target tracking algorithm called Markov Chain Monte Carlo Data Association [19]. Segments of computed tracks are then used as (spatio-temporal) features. Between pairs of cameras, the tracks are used as features to decide if there is correspondence between the cameras. If correspondence exists, then the relative orientation and position up to a scale factor between pairs of cameras is done using an extension of [16]. This extension uses the minimization of the reprojection error between the features in each pair of cameras to determine if they overlap and what the relative orientation and and position, up to a scale factor, is for the pair. In conjunction with this, the radio nodes are transmitting information to one another in order to determine q-ranges between groups of 4 radio sensors using radio inferometry [18].

The q-ranges, $\{d_{ijkl}\}$ are then fused with $\{R_{ij}, T_{ij}\}$ from the image data. A non-linear fusion method can be used then to find the resulting scale factor so the position of the cameras is fully known. A computationally efficient linear fusion method can be used for special cases and it is described below along with conditions under which the method can be applied. The building blocks of the fused localization method are discussed in the following sections in more detail.

3. SINGLE MODALITY RADIO INFEROMETRY AND IMAGE METHODS

3.1. Localization Using Spatio-Temporal Feature Correspondence in Images

For the image analysis block of the fusion-based localization method, feature detection and then correspondence of features between cameras is needed. Since camera network cannot be limited to small baseline, one cannot necessarily use brightness or proximity constraints and traditional methods of static features detection and correspondence, such as SIFT features [20] or Salient Region [21], will not work in the majority of cases. Thus other visual information must be leveraged.

A different approach is to detect moving foreground objects in the scene. If two cameras overlap in their field of view and a moving foreground object appears in the overlap, then both cameras will see the moving object. As the shape and appearance of the object might be different between these cameras, we do not want to use the whole object or a visual descriptor of the object as the feature. Instead we just choose a point on the object in the image. Since we do not know how the cameras are oriented, we choose the centroid of the object as the point. Once we have enough points, that are detected in the cameras at the same time, we can use the epipolar constraint to estimate the rotation R and translation T between the cameras.

Using centroid points alone is not sufficient for a camera network though. First, for generalized systems where we know no information about the object or the scene, detecting foreground objects can be done using adaptive background subtraction techniques. However, background subtraction reacts to noise in the scene, so shadows and slight motion of the background, i.e., swaying tree due to wind, would create an foreground object detection and lead to a centroid point for non-foreground objects. Second, if there multiple foreground objects moving through the scene it becomes more difficult to do correspondence between points in a pair of cameras.

To overcome these difficulties, we use the timing information on the detection of foreground objects and MCMCDA [19] to form tracks of the foreground objects in the image plane as they move through the scene. This cuts down on the number of possible correspondences and noisy detections, while providing robust foreground points to be used for solving the epipolar constraint.

We then extend [16] which uses object image tracks as features for correspondences. We define the problem similarly, as for a given time period $[t_0, t_0 + 1, \ldots, t_n]$: where

(C_i, C_j): pair of cameras i and j, where $i \neq j$
p_i: the total number of tracks in C_i over the time span
Θ_i: set of tracks in C_i
$t_s(\theta_i^m)$: starting time of a track $\theta_i^m \in \Theta_i$ where $m \in \{1, 2, ..., p_i\}$
$t_e(\theta_i^m)$: ending time of a track $\theta_i^m \in \Theta_i$,

$t_0 \leq t_s(\theta_i) < t_e(\theta_i) \leq t_n, \forall \theta_i \in \Theta_i$.

First, we look to see if there are enough tracks that overlap in time between the camera pair (C_i, C_j). A track $\theta_i \in \Theta_i$ can be matched to a track $\theta_j \in \Theta_j$ if θ_i and θ_j overlaps in time. Now we let Γ_{ij} be a set of all matchings between tracks in Θ_i and Θ_j. A matching $\gamma_{ij} \in \Gamma_{ij}$ is a subset of matches such that a track in Θ_i is matched to at most one track in Θ_j. This is similar to a matching in a bipartite graph, but in our case each vertex is a track.

For each matching γ_{ij}, we solve for the essential matrix $E(\gamma_{ij})$ using the standard algorithm [9] . From $E(\gamma_{ij})$, the translation matrix T_{ij} and rotation matrix R_{ij} between camera i and j can be computed. However, if there is no overlapping field of view between camera i and camera j, this leads to an incorrect solution. We can determine whether a pair of cameras has an overlapping field of view by thresholding the the average reprojection error, d, which is defined as:

$$d = \frac{(x_j^T \hat{T} R x_i)^2}{||\hat{e}_3 \hat{T}_{ij} R_{ij} x_i||}$$

where x_i and x_j are ordered feature points from matched tracks in γ_{ij} and $e_3 = [0, 0, 0, 1]^T \in R^3$. The pseudo code for the whole process is as shown in Algorithm 3.1. When the

Algorithm 3.1: LOCALIZATION($rawvideo$)

for $t \leftarrow 1$ **to** t_n
for each C_i, build object tracks
find candidate matches using Γ
if $\Gamma \neq \varnothing$
do { $\forall \gamma \in \Gamma$
Find $\hat{\gamma} = \arg\min d$
if $d(\hat{\gamma}) \leq$ threshold
do { $\hat{E} = E(\hat{\gamma})$
Stop
else $\hat{E} =$ NULL
else increase $t_n \leq$ threshold and run again
return $(\hat{E})$

algorithm returns NULL, the pair of cameras do not have an overlapping field of view.

This then gives us the orientation R_{ij} and the position up to scale, T_{ij}, for cameras that overlap in the field of view. We still need to get the scale factor for the position element and thus we need to use additional information from the radio nodes.

3.2. Radio Interferometry

The Radio Interferometric Positioning System (RIPS) was proposed in [18] for node localization using the phase measurements of the radio signals with low cost hardware. The basic idea behind RIPS is to utilize two transmitter nodes to create an interference signal. The two nodes transmit sine waves at slightly different frequencies at the same time, creating a composite interference signal with a low frequency envelope. This interference frequency can be measured by cheap and simple hardware available on a wireless sensor node.

Figure 2 shows an example of the interference signal and its low frequency beats at nodes C and D. The model of the radio interference was developed in [18]. The phase offset of the interference signals received at two different receivers can be expressed in terms of a quantity called the *q-range*, which is a linear combination of distances between the two transmitters and two receivers defined as

$$q_{ABCD} = d_{AD} - d_{BD} + d_{BC} - d_{AC}$$

An important theorem on *q-range* presented in [18] states that the relative phase offset of received interference signals at nodes C and D are related to *q-range* as follows

$$\varphi_{CD} = 2\pi \frac{q_{ABCD}}{c/f} \pmod{2\pi}, \qquad (1)$$

and $f = f_A + f_B$, where f_A, f_B are carrier frequencies of transmitters A and B, and c is speed of light.

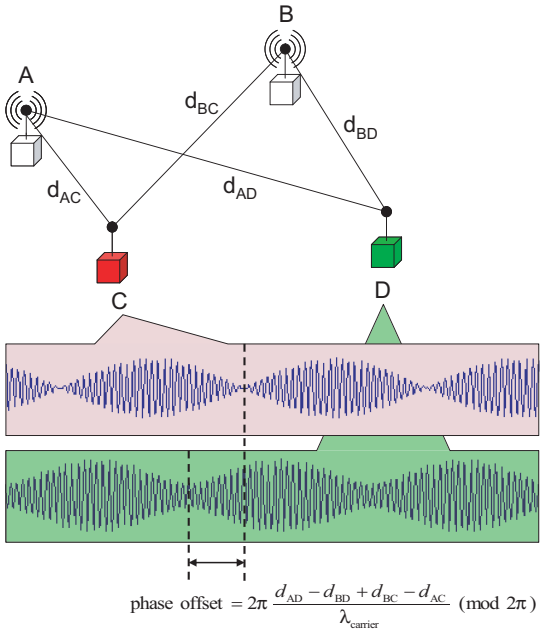


Fig. 2. Two transmitters A and B transmit at the same time at two close frequencies. The interfere signal is observed by receivers C and D. Figure from [18].

Another important result presented in [22] states that in a network of n wireless nodes, there exist a maximum of $n(n-3)/2$ linearly independent *q-ranges*. It was shown that by taking independent *q-range* measurements for different combinations of four nodes, it is possible to reconstruct the relative location of the nodes. An algorithm and implementation for localization were presented in [18].

The main benefit of RIPS lies in the fact that it does not require any additional hardware because common radio transceivers can be utilized for phase measurements.

By using the *q-ranges* we now have distances that can be used in conjunction with the R_{ij}'s and T_{ij}'s we found from the image data. We now need to combine these two sets of information to determine the position of the cameras.

4. RADIO INFEROMETRY AND IMAGE CORRESPONDENCE FUSION

While q-ranges alone or image data alone do not have enough information to completely localize the cameras, fusing the two sets of information gives us complete localization. Here we present both linear and nonlinear methods for solving this data fusion problem. The linear method has a unique solution, but requires certain conditions on the topology of the camera network. If these conditions are not met, the nonlinear method is used for complete localization.

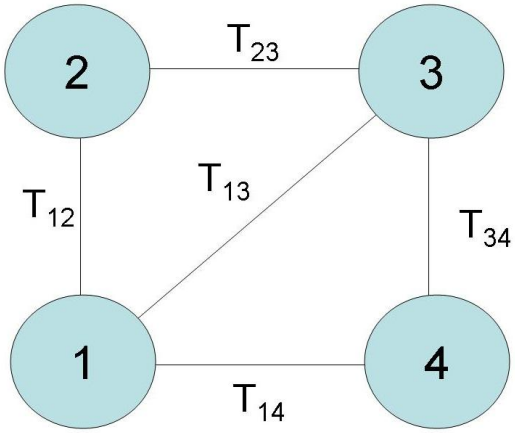


Fig. 3. The overlap in fields of view between the cameras nodes based on the object image tracks

4.1. Linear Method

For a network of N cameras, there exist $N(N-3)/2$ independent q-range equations as stated in [22]. We also know, given the network, that there are $N(N-1)/2$ possible pairings of cameras, thus $N(N-1)/2$ pairwise distances exist. In addition to the $N(N-3)/2$ independent q-range equations, N more independent equations are necessary in order to solve for all the pairwise unknown distances. We look to the camera translations vectors, T_{ij}, where $i \neq j \in 1, ..., k$ and where $k \leq N$ to provide us with these additional equations.

Using the available T_{ij}, we want to find equations such that we can use the vector notation and the unknown scalars to write one T_{ij}, with it's unknown scale λ_{ij}, as a sum of T_{kl} and λ_{kl} and T_{mn} and λ_{mn} such that $ij \neq kl \neq mn$. For example, Figure 3, the unit translation vector T_{13} can be written as:

$$\lambda_{13}T_{13} = \lambda_{12}T_{12} + \lambda_{23}T_{23} \tag{2}$$

If we take this equation and write it in terms of the unknown scales, λ_{ij} we get:

$$[T_{12} \;\; T_{12} \;\; -T_{13}] \begin{bmatrix} \lambda_{12} \\ \lambda_{23} \\ \lambda_{13} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

As can be seen, the matrix $[T_{12} \;\; T_{23} \;\; -T_{13}]$ is not full rank and $[\lambda_{12} \;\; \lambda_{23} \;\; \lambda_{13}]^T$ lies in the null space of the matrix. Thus, we can only get at most, two independent equations from using Equation 3. Additionally, it can be seen that a clique of three cameras all being able to view one another, is needed to write this type of equation, in order to write one T_{ij} in terms of other T_{ij}. Therefore, in addition to the q-range independent equations, we need N equations from the camera network which means $N/2$ cliques of three cameras need to exist in the camera network.

If enough cliques are found in the camera network, then we can write the unknown scales, λ_{ij} in terms of the q-ranges and translations T_{ij} as:

$$Ax = b \tag{4}$$

where

$$b = \begin{bmatrix} d_{1234} & d_{1324} & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5}$$

$$A = \begin{bmatrix} 0 & -1 & 1 & 1 & -1 & 0 \\ -1 & 0 & 1 & 1 & 0 & -1 \\ T_{12} & -T_{13} & \mathbf{0} & T_{23} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & T_{13} & -T_{14} & \mathbf{0} & \mathbf{0} & T_{34} \end{bmatrix} \tag{6}$$

$$x = \begin{bmatrix} \lambda_{12} & \lambda_{13} & \lambda_{14} & \lambda_{23} & \lambda_{24} & \lambda_{34} \end{bmatrix}^T \tag{7}$$

Thus, given enough cliques of three cameras in a network which have overlapping fields of view such that all T_{ij} can be found for that clique, a linear method exists to solve for the scales λ_{ij} as shown in Equation 4.

4.2. Nonlinear

While a unique solution from the linear method can be found if certain conditions exist in the camera network, we have developed a nonlinear method for general cases. From the localization algorithm described in section 3.1 we have a set of pair of camera nodes with overlapping field-of-views (FoV). Lets denote the set as,

$$\mathcal{O} = \{(C_i, C_j) : C_i \text{ has overlapping FoV with } C_j, i \neq j\}$$

For a camera pair $P_k = (C_i, C_j) \in \mathcal{O}$, denote scaling factor between camera nodes C_i and C_j as λ_k. We also know the relative rotation matrix and unit translation vector for the pair. Let the rotation matrix denoted as R_j^i and the unit translation vector as, $\mathbf{T}_{ij}^i$. Assuming the camera network is connected, we can compute the rotation matrices for each of the camera node in a common frame of reference. Without loss of generality, lets consider reference frame of node 1 as the global reference frame. With successive multiplications of the relative rotation matrices we can compute absolute rotation matrices R_i^1 for $i > 1$ as

$$R_i^1 = R_{i_1}^1 \times R_{i_2}^{i_1} \times \cdots \times R_i^{i_n}$$

where $\{1, i_1, i_2, \cdots, i_n, i\}$ is a path from node 1 to node i. For $i = 1$ the rotation matrix is an identity matrix of size 3.

Using the unit translation vectors, absolute rotation matrices and the scaling factor for each pair, we can compute vector for pair P_k as

$$\mathbf{x}_{ij}^1 = R_i^1 \times \mathbf{T}_{ij}^i \times \lambda_k$$

The camera node locations in the global reference frame can be computed as

$$\mathbf{x}_i^1 = \mathbf{x}_1 + \mathbf{x}_{1i_1}^1 + \mathbf{x}_{i_1 i_2}^1 + \cdots + \mathbf{x}_{i_n i}^1$$

where $\{1, i_1, i_2, \cdots, i_n, i\}$ is a path from node 1 to node i, and $\mathbf{x}_1 = [0, 0, 0]^T$.

Both vectors $\mathbf{x}_{ij}^1$ and $\mathbf{x}_i^1$ are functions of the scaling factors λ, i.e. $\mathbf{x}_{ij}^1 = R_i^1 \times \mathbf{T}_{ij}^i \times \lambda_k \triangleq \mathbf{F}_{ij}(\lambda_k)$, and $\mathbf{x}_i^1 \triangleq \mathbf{G}_i(\lambda)$. Hence, the distance between all other camera pairs that are not members of the set $\mathcal{O}$ can also be represented as a function of scaling factors as

$$d_{ij} = ||\mathbf{G}_i(\lambda) - \mathbf{G}_j(\lambda)|| \triangleq H_{ij}(\lambda)$$

The q-ranges defined in section 3.2, which are linear combinations of distances can also be expressed as functions of scaling factors as:

$$\begin{aligned} q_{\text{ABCD}} &= d_{\text{AD}} - d_{\text{AC}} + d_{\text{BC}} - d_{\text{BD}} \\ &= H_{\text{AD}}(\lambda) - H_{\text{AC}}(\lambda) + H_{\text{BC}}(\lambda) - H_{\text{BD}}(\lambda) \\ &\triangleq Q_{\text{ABCD}}(\lambda) \end{aligned}$$

The problem can be expressed as a non-linear least squares problem as

$$\text{minimize } \mathcal{L}(\lambda) = \sum_{\text{ABCD} \in \mathcal{T}} (Q_{\text{ABCD}}(\lambda) - \tilde{q}_{\text{ABCD}})^2$$

where $\mathcal{T}$ is the set of q-range tuples, $\tilde{q}_{\text{ABCD}}$ is the measured q-range for tuple ABCD.

5. EXPERIMENTAL RESULTS

In this section, we apply our fusion method on real data. The algorithm is tested on an outdoor deployment of a network of cameras. Using both the camera images and the radio data we are able to estimate the position and orientation of the cameras. The network consists of Linux PCs equipped with Logitech Quick Cam Pro 4000 cameras, and XSM wireless sensor motes. Six camera nodes and 7 XSM motes are used. The ground truth location of the cameras+XSM motes is show in fig 4. The cameras have a resolution of 240 x 320 pixels and acquired images at 8 frames per second(fps). 12 minutes of data is taken from the cameras for localization. Multiple types of objects moved through the scene during this recording. An example of the different objects is shown in Figure 5

We use the existing TinyOS implementation of RIPS developed at Vanderbilt. The TinyOS implementation running on 7 XSM motes and a Java application running on base station provide us with the required q-range measurements. For more detail of RIPS and its implementation we refer reader to [22].

The steps of the automatic method applied are as follows. Adaptive background subtraction is applied to each image to obtain the foreground objects. Bounding boxes are created around the foreground object and if a bounding boxe is located at the edge of an image, the foreground object is not considered for further processing to build object image track. The reason this check is implemented in the algorithm is because a bounding box at the edge of the image could indicate that part of the foreground objects it getting occluded and thus the centroid would not be unstable.

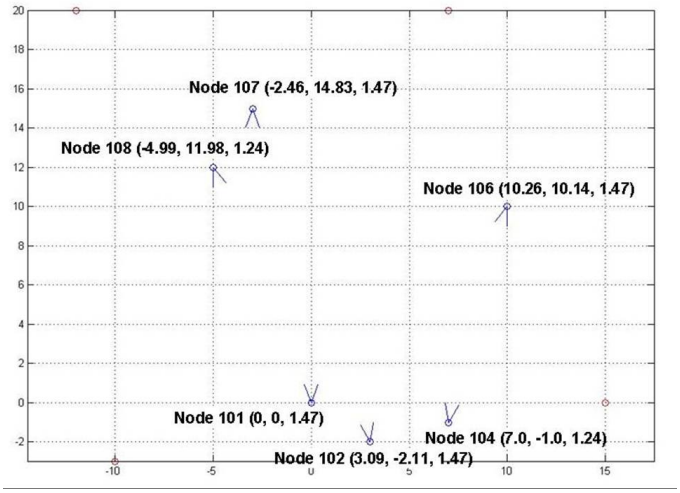


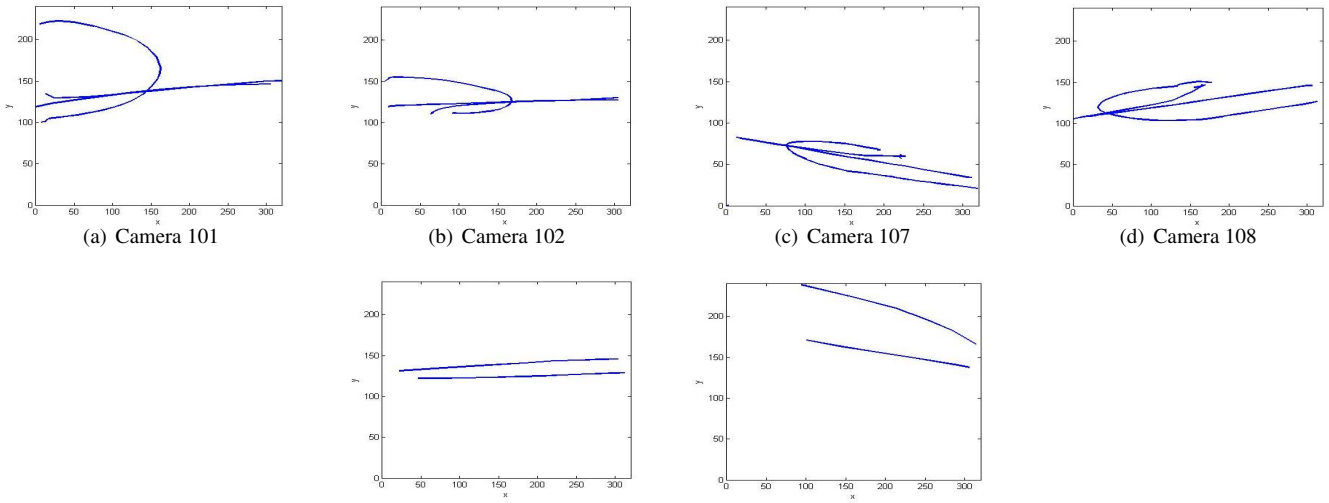
Fig. 4. An overhead view of the layout of the cameras+radio sensors. The units are measured in meters

Using the remaining centroids from the bounding boxes, object image tracks are built using MCMCDA. In our method, the number tracks needed is defined to be at least 4 seconds long to further constrain the paths used and so each track had to have at least 32 points due to the frame rate. This parameter in the algorithm can be adjusted as desired.

The tracks from the centroids are then used for feature correspondence between the cameras using the epipolar constraint and R and T, up to scale, solved using SVD. Figure 7 shows what cameras are visually connected given the correspondence between object image tracks. As can be seen, cameras 104 and 106 are disjoint from the rest of the network. While their fields of view overlap with each other, they are not connected to the rest of the network. We verify this is correct based on the ground truth of the setup. The average reprojection error for all camera pairs is < 3 pixels.

Using the T 's for all overlapping camera pairs, this is fed into both an automatic nonlinear and linear methods along with the q-ranges obtained from the XSM motes. The resulting scale factors are shown in Table 1. It should be noted that linear method solves for all scaling factors even if the camera pair does not have overlapping field-of-view.

Fig. 5. Some of the moving foreground objects in the scene as observed from camera 101



(a) Camera 101 (b) Camera 102 (c) Camera 107 (d) Camera 108

(e) Camera 104 (f) Camera 107

Fig. 6. (Top) Object image tracks for frames 1 through 500 for cameras which had correspondence in their respective tracks (Bottom) Object image tracks for frames 1 through 500 for cameras that did not have correspondence with the top row cameras, but which had correspondence with each other.

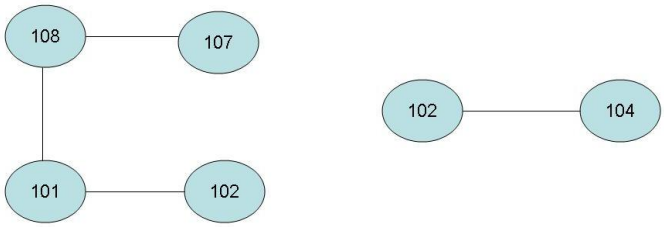


Fig. 7. The overlap in fields of view between the cameras nodes based on the object image tracks

Camera Pair	Ground Truth	Nonlinear Estimate	Linear Estimate
101 to 102	3.7417	3.2376	4.2593
101 to 107	15.0326	15.5074	13.5940
101 to 108	12.9797	13.2370	13.1016
102 to 107	17.8260	15.4361	14.9415
102 to 108	16.2440	–	15.4440
107 to 108	3.8179	3.6743	1.5795

Table 1. The scale factors for the distances between cameras.

6. CONCLUSION

Localization of a camera network is important for full efficiency of the network. In this paper, we have shown that using image information alone will not be enough to fully localize a camera network, but using radio information in conjunction with image information, will allow a camera network to be fully localized. We have presented both a linear method to fuse the data which can be used to find a unique solution if there are certain conditions on the camera network such that enough camera cliques of size three exist. We have also presented a nonlinear method for fusing the data which will work on any network.

7. ACKNOWLEDGEMENTS

This work is partially supported by ARO MURI W911NF-06-1-0076.

8. REFERENCES

[1] M. McCahill and C. Norris, "From cameras to control rooms: the mediation of the image by cctv operatives," *CCTV and Social Control:The politics and practice of video surviellance-European and global perspectives*, 2004.

[2] M. T. Moore, "Cities opening more video surveillance eyes," *USA Today*, July 18 2005.

[3] Robin Wolff, Dave J. Roberts, Anthony Steed, and Oliver Otto, "A review of telecollaboration technologies with respect to closely coupled collaboration," *International Journal of Computer Applications in Technology*, vol. 29, no. 1, 2007.

[4] Z. Yang, W. Wu, K. Nahrstedt, G. Kurillo, and R. Bajcsy, "Viewcast: View dissemination and management for multi-party 3d tele-immersive environments," *ACM Multimedia*, 2007.

[5] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. Intelligent Transportation Systems*, vol. 1, no. 2, pp. 108–118, June 2000.

[6] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Trans. Intelligent Transportation Systems*, vol. 1, no. 2, pp. 119–130, June 2000.

[7] O. Masoud, N. P. Papanikolopoulos, and E. Kwon, "The use of computer vision in monitoring weaving sections," *IEEE Trans. Intelligent Transportation Systems*, vol. 2, no. 1, pp. 18–25, Marci 2001.

[8] "http://www.cis.upenn.edu/teleimmersion/research/downloads/EasyCal/," .

[9] Y. Ma, S. Soatto, Jana Kosecka, and S. Shankar Sastry, *An Invitation to 3D Vision*, Spinger-Verlag, 2004.

[10] W.E. Mantzel, Choi Hyeokho, and R.G. Baraniuk, "Distributed camera network localization," *Asilomar Conference on Signals, Systems and Computers*, Nov. 2004.

[11] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar, "Distributed localization of networked cameras," *The Fifth International Conference on Information Processing in Sensor Networks*, April 2006.

[12] A. Rahimi, B. Dunagan, and T. Darrell, "Simultaneous calibration and tracking with a network of non-overlapping sensors," *Computer Vision and Pattern Recognition*, vol. 1, July 2004.

[13] R. Cucchiara, A. Prati, and R. Vezzani, "A system for automatic face obscuration for privacy purposes," *in Pattern Recognition Letters*, 2006.

[14] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," *Proceedings of Workshop on Motion and Video Computing*, 2002.

[15] L. Lee, Raquel Romano, and Gideon Stein, "Monitoring activities from multiple video streams: Establishing a common coordinate frame," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[16] M. Meingast, S. Oh, and S. Sastry, "Automatic camera network localization using object image tracks," in *IEEE International Conference on Computer Vision*, 2007.

[17] Koen Langendoen and Niels Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Comput. Networks*, vol. 43, no. 4, pp. 499–518, 2003.

[18] Miklos Maroti, Branislav Kusy, Gyorgy Balogh, Peter Volgyesi, Andras Nadas, Karoly Molnar, Sebestyen Dora, and Akos Ledeczi, "Radio interferometric geolocation," in *ACM SenSys 2005*, November 2005.

[19] Songhwai Oh, Stuart Russell, and Shankar Sastry, "Markov chain Monte Carlo data association for multi-target tracking," *IEEE Trans. Automatic Control (submitted)*, 2007.

[20] D. Lowe, "Distinctive image features from scale invariant keypoints," *In IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[21] A. Zisserman T. Kadir and M. Brady, "An affine invariant salient region detector," *In ECCV*, 2004.

[22] Branislav Kusy, Akos Ledeczi, Miklos Maroti, and Lambert Meertens, "Node density independent localization," in *International Conference on Information Processing in Sensor Networks (IPSN 2006)*. 2006, ACM.