Ohio State University, Dept. of Computer Science and Engineering Technical Report OSU-CISRC-6/10-TR13

Geo-registration and Interactive Control for Distributed Camera Networks

Matthew Nedrich Karthik Sankaranarayanan James W. Davis
Dept. of Computer Science and Engineering
Ohio State University
Columbus, OH 43210 USA
{nedrich,sankaran,jwdavis}@cse.ohio-state.edu

Abstract

Distributed camera networks typically consist of a very large number of cameras. Often, it is difficult to manage and control these cameras in an efficient and intuitive manner. In this paper we present an application study of a camera registration technique used to create an interactive control system. We argue that such a system improves efficiency for controlling a large network of cameras. We first describe the PTZ camera registration technique. Next, we present an interactive map-based system to effectively manage surveillance tasks. We demonstrate that the framework improves efficiency in a distributed camera network by allowing users to concentrate on the environment itself rather than placement and view of individual cameras.

1. Introduction

Distributed camera networks are widely used in surveillance applications for physical security (airports, train stations, military installations, etc.) and transportation systems (highway traffic, pedestrian traffic, etc.). They typically consist of hundreds (if not thousands) of cameras spread across large areas and employ both fixed-view and pan-tilt-zoom (PTZ) cameras.

Viewing a particular location in the world via a camera network can be a difficult task, especially for an inexperienced human operator with a network containing many cameras. Effective use requires significant knowledge of camera placement in the environment in order to choose the correct camera. It also requires experience with accessing and controlling cameras in the network. Furthermore, the complexity associated with this task increases with the size of the network. This makes scalability a highly desired feature in such networks. The best way to achieve scalability is to design the camera network in such a way that the operator is agnostic to the camera topology. This way the operator is only concerned with a location of interest, relying on the system to handle all other issues (e.g. choosing the closest camera, operating the camera at the location, etc.).

While distributed camera networks may employ a few static-view cameras, most cameras are PTZ, each having a large field-of-coverage (viewspace) due to pan and tilt motor controls (typically 360×90 degrees). A control system therefore needs to know the mapping between the field-of-coverage of each camera and the environment. To achieve this, we use an approach to register each camera's pan-tilt space with a common frame-of-reference.

To accomplish the camera-scene registration, we leverage the registration model presented in [10]. This work provides a method to register each camera's complete field-of-coverage (all possible views of a PTZ camera) to a base reference frame. Each camera's complete field-of-coverage is represented using a spherical panorama (see Fig. 1). Since this 360×90 panoramic image simulates a fisheye lens view, a "defishing" operation is performed to warp the panorama onto a base reference frame. We employ an aerial orthophoto (which contains geographic metadata) as the base reference frame (see Fig. 2). The registration between the panorama and the orthophoto is performed by projecting rays of the pan-tilt orientations onto the orthophoto ground plane and then mapping corresponding feature points using a transformation matrix. After registering each of the cameras in the network, we then know the pan-tilt orientations of each camera required to view the same ground location (for all visible ground locations). Furthermore, we also know the true geographic coordinate of each registered ground point, given the geographic metadata in the orthophoto.

We leverage this registration framework to build an interactive map-based system that allows users to effectively manage surveillance networks and tasks. It does so by encouraging the user to focus on the environment of the camera network rather than the placement and control of individual cameras. Our system allows users to interact directly with a map of the environment and control the camera network by using the map. We provide experimental results for the registration component and the use of our resulting system for camera network management.

In Sect. 2, we discuss previous approaches to the problems addressed. In Sect. 3 we describe the registration model, and in Sect. 4 we describe the interactive control system. Section 5 includes experimental results for each of the proposed techniques, and a user interaction experiment demonstrating the advantage of the proposed framework.

2. Related Work

We require a simple, yet scalable and reliable registration model to register PTZ cameras to a base reference frame. Most existing registration work for establishing relationships between cameras in a distributed system propose different methods for finding homographies between each pair of cameras. In [8], tracking data between pairs of cameras are employed and centroids of moving objects are used to solve for correspondences. Another way to obtain these correspondences between objects detected in the cameras is by using feature matching techniques [7] or geometric methods [6, 1]. However, these techniques establish relationships between pairs of cameras and do not share a common coordinate system. These techniques also rely upon recovering the camera locations. In [4], a specialized catadioptric omnicamera is used to establish a mapping from a limited resolution omnidirectional view to the ground plane. However this mapping is employed by means of a lookup table and the inverse mapping from the ground plane back the camera view cannot be expressed analytically, thus hindering analysis sourced from ground plane information.

Most work relating to visualizing and controlling camera networks has been focused on passively reporting the state of the environment or visualizing an active process applied to the environment (e.g., tracking). In [2] a semi-autonomous multi-camera surveillance system is described which allows an operator to monitor the environment by automating tasks such as detection, tracking, etc. In [3] a model to visualize inter-camera tracking using a 3D world model is described. In [5] gesture recognition is incorporated into a perceptual user interface paradigm for video surveillance. However, these models do not address the problem of high-level distributed camera network control that we wish to address.

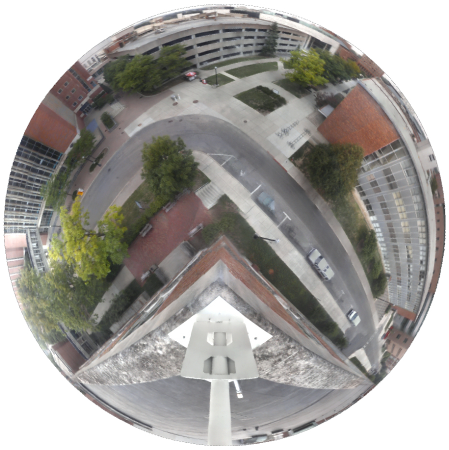

Figure 1. Spherical panorama representing the entire viewspace of a PTZ camera in a single image.

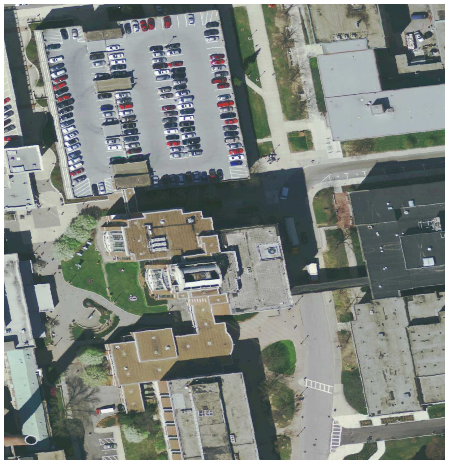

Figure 2. Orthophoto of the local region of our camera network.

3. Registration Approach

Wide-area surveillance cameras are typically equipped with pan-tilt motor controls to see across a large area, but at any given time the camera only views a small portion of this viewspace. Therefore the first step in the integration process of our distributed camera network is to model the entire viewspace for each of the PTZ cameras so it may be registered with a base reference frame. To achieve this we use the camera model described in [9] to build spherical panoramas. Such panoramas provide a compact and unified model for representing the entire viewspace of each camera via a single image, as seen in Fig. 1. We employ the following two equations from [9] to map (x, y) pixel values to their corresponding (θ, ϕ) pan-tilt orientations in order to build the panorama

$$\delta\theta = \tan^{-1}\left(\frac{x}{y \cdot \sin\phi + f \cdot \cos\phi}\right) \tag{1}$$

$$\delta\phi = \tan^{-1}\left(\frac{y + a}{f \cdot \cos\left(\tan^{-1}(\frac{a}{b} \cdot \frac{x}{y+a})\right)} - \frac{a}{f}\right) \tag{2}$$

where f is the focal length of the camera, $a = \frac{f}{\tan\phi}$, and $b = \frac{a}{\sin\phi}$. Here $\delta\theta$ and $\delta\phi$ represent the changes in pan and tilt between the $(pan, tilt)$ of the (x, y) location and the

$(pan, tilt)$ of the center of the camera image. The panorama models a linear fisheye view where, for each (x, y) location in the panorama image, the pan can be obtained by calculating the angle subtended with the X-axis, and the $tilt$ varies linearly as the radius from the center of the panorama image.

The next step is to register the camera's viewspace (represented by the panorama) with a base reference frame for the distributed camera network. Here, we employ an aerial orthophoto as the base reference frame and the registration technique described in [10] to perform the registration. Figure 2 shows a cropped orthophoto for our region of interest. Such imagery is publicly available and we attained this image from the state geographic information office.

The registration is composed of a two step process, 1) panorama "defishing", followed by 2) transformation between the "defished" panorama and the orthophoto.

The registration process works as follows. Let θ and ϕ represent the pan and tilt angle of an (x, y) point in the panorama and x_g and y_g be the corresponding point on the base reference frame. The registration framework provides a means to register the (θ, ϕ) from a camera's viewspace to a rectilinear (x_g, y_g) ground point. We chose the rectilinear Universal Transverse Mercator (UTM) world coordinate system as our ground plane. Thus, each camera's viewspace was registered directly to UTM coordinates (which may be easily converted to Latitude-Longitude coordinates if desired). Combining the defishing with an affine transformation into one step produces a single registration matrix that provides the relationship between pan-tilt (θ, ϕ) locations obtained from the panorama and (x_g, y_g) ground plane locations

$$\begin{bmatrix} x_g \\ y_g \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & t_x \\ a_3 & a_4 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tan\phi \cdot \cos\theta \\ \tan\phi \cdot \sin\theta \\ 1 \end{bmatrix} \quad (3)$$

where the six parameters (a_1, a_2, a_3, a_4, t_x, and t_y) govern all the necessary degrees of freedom (defish, scale, rotation, and translation).

By matching (x, y) points on the panorama to the orthophoto ground plane points (x_g, y_g), we obtain a set of correspondences. For each point on the spherical panorama, we can recover its pan-tilt (θ, ϕ), hence we can calculate $(\tan\phi \cdot \cos\theta)$ and $(\tan\phi \cdot \sin\theta)$ for each point. These values can then be employed in a least squares formulation for Eqn. (3) to learn the transformation.

We performed this registration using cameras from our camera network to obtain a pan-tilt to UTM coordinate mapping. Similarly, the inverse transformation matrix provides a reverse mapping from UTM to pan-tilt. By performing this registration for each of our cameras, we obtain a unified map-based representation for the entire distributed camera network that enables both intuitive and efficient camera control and coverage. We now describe how this registration may be leveraged to improve the management and use of a distributed camera network.

4. Interactive Camera Control

Our criteria for a useful interactive control system are as follows:

- The system should allow for easy camera control across a distributed camera network.
- The user should be able to perform tasks in an intuitive and efficient manner.
- The system should re-focus the attention of the user *from* individual cameras *to* the environment of the camera network. That is, the user should not worry about where cameras exist in the environment or what the viewing range of a specific camera is, but rather what areas are visible and accessible through the camera network as a whole.
- The system should also be scalable and able to incorporate new cameras.

To develop our system we used the Java version of NASA Worldwind (WW), an open source alternative to Google Earth. The Worldwind Software Development Kit (WW SDK) provides the developer with basic functionality to interact with and customize WW's virtual model of the Earth. It also provides capability to place satellite/aerial imagery onto a virtual model of the Earth. Additionally it provides basic controls to navigate around the world (e.g., move to different locations and zoom in/out). When interacting with the world model, WW provides geo-registered information, including the world coordinates (e.g., Lat-Lon, UTM) of the interaction. This functionality is easily extendable to develop custom applications.

To integrate a camera into our system, the user must have access to the camera and have the ability to control the camera. Required camera control functionality includes the ability to send relative (for joystick control, if needed) and absolute (pan, tilt, zoom) position commands to the camera, as well as query the camera for its current (pan, tilt, zoom) orientation. Using a subset of these controls, a spherical panorama can be generated (as described in [9]) and registered to the world using the registration process described in Sect. 3. The registration provides the camera's location in world coordinates (UTM). The camera is then added to a rendering layer where other camera-specific information, such as camera name or camera model, may be included.

Our system offers an interactive map-based display for the camera network. The user is presented with an orthophoto and may navigate around the orthophoto as well as zoom in and out. Employing our map-based interface the user can open a live camera feed using the camera's icon on the map. The user may then control the camera using simple joystick control. In addition to this standard camera control, and of importance to this work, the user can also use the environment itself to control the camera network by interacting directly with the background orthophoto. When the user selects a ground plane location, the system automatically determines the camera that is closest to the selected point. The system then instantly opens a live camera feed for the closest camera and orients the camera to point at the selected ground point using the registration algorithm described in Sect. 3. This process is accomplished by first obtaining the world coordinate (UTM) that the user selected on the ground (obtained through the WW SDK). Once the selected world coordinate is determined, the system determines the closest camera using the 2D Euclidean distance (or 3D if the camera height is known) between the cameras and the selected location. To resolve camera occlusion problems where the chosen location may be occluded from the closest camera, the user can manually set camera pan-tilt ranges to ignore areas when determining the closest camera (more sophisticated techniques may be used if elevation information, e.g., LIDAR, is available for the area of interest). Once the closest camera is determined, the pan-tilt required to orient the camera to the selected ground location is calculated by the system using the inverse of the transformation matrix described in Sect. 3. The camera is then immediately re-positioned to the selected ground location, showing a live video feed of the desired location.

5. Experiments

We evaluated the registration framework by calculating the error values associated with the mapping technique. We also performed experiments in a distributed camera network to demonstrate the advantages of using the proposed techniques in an interactive control system.

The camera network used for the experiments consists of multiple Pelco Spectra III/IV SE PTZ cameras. These cameras are mounted on buildings at various heights and can view various building entrances, roads, and sidewalks. They connect to our research lab via optic fiber and provide full access to video and PTZ control.

5.1. Registration Evaluation

This set of experiments was performed to measure the accuracy of the camera registration. We accomplished this by registering panorama images from multiple cameras to an orthophoto and calculating the registration error.

We began by manually identifying a set of feature points in each panorama that also appear in the orthophoto. In our experiments, we used 10-12 such feature points (a minimum of 3 points are required for registration). We then computed the pan-tilt orientation of each of those points in that camera's pan-tilt space using the angle subtended by each point with the X-axis (which indicates the pan of that location) and its distance from the center of the panorama (which represents its tilt angle). Next, we marked the points corresponding to each of these locations on the orthophoto. This gave their (x_g, y_g) locations in the rectilinear coordinate frame. Using these (x_g, y_g) values for multiple points, we calculated the registration employing a least squares formulation of Eqn. 3. We used the resulting transformation matrix to register the panorama with the orthophoto. We did this procedure for three different cameras (each on a different building). For each of the camera panoramas, we then picked approximately 20 feature points on the transformed panorama and compared their pixel locations with their corresponding ground truth locations (determined manually) on the orthophoto. The error statistics (in pixels) for the registration technique are shown in Table 1.

Table 1. Error statistics (in feet) for the registration model with different cameras. (1 foot/pixel orthophoto resolution)

	Mean	Std Dev.
Camera 1	2.671	0.829
Camera 2	2.267	0.914
Camera 3	2.884	0.761

These values were obtained using an orthophoto resolution of 1 foot/pixel. Therefore, the results translate to a mean error of less than 3 feet (3 pixels) on the ground with a standard deviation less than 1 foot. We believe that for most common surveillance tasks such as camera control, tracking, camera handoff, etc. these error values are within reasonable limits and could potentially be reduced with higher resolution orthophotos (some public data sets are available in 6 inch/pixel resolution).

5.2. User Evaluation

Users of distributed camera networks often wish to view a specific location in the environment. For example, a security operator may receive an alarm notifying the occurrence of an event at a particular location. Such an event may be caused by a fire alarm, a person entering an unauthorized location, etc. The operator may be provided with a physical location of the alarm (at a bus stop, etc.) or, if GPS equipped, the alarm may be directly displayed on a map. An alarm requires the operator to quickly obtain a visual of the environment surrounding the alarm location. In such a situation response time is very important. A major contributing factor to response time is identifying camera(s) that can view the area of interest and then redirecting the camera(s) manually (with a joystick). This is also a process that re-

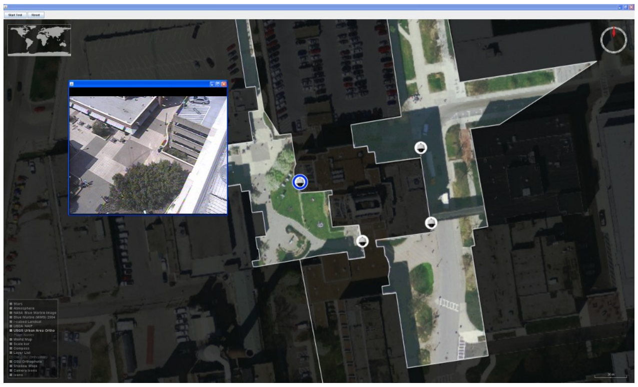

Figure 3. Screen shot of the baseline and proposed system interface. The four camera icons surrounding the center building represent the four cameras used in the experiment. The highlighted region corresponds to the area of the ground plane visible to the camera network.

quires significant training time for large environments. We propose that more fully integrating a distributed camera network into an environmental representation will improve the efficiency of viewing a desired location. To evaluate this hypothesis, we compared our proposed system to a baseline system for a task where the user was asked to view a randomized set of locations in the environment through a distributed camera network.

Baseline System: In the baseline system, the user was provided an orthophoto with four camera icons representing four different cameras in the world (see Fig. 3). A portion of the orthophoto was highlighted, roughly representing the area of the ground plane covered by the camera network. The user could select any of the four camera icons to open a live camera feed, and control the selected camera using a joystick (standard practice).

Proposed System: In our proposed system the user was provided with the same orthophoto, camera icons, and highlighted region on the ground plane as in the baseline system. However, using our proposed system, the user could interact with the environment by clicking on the background orthophoto. Doing so would open a view of the nearest camera to the selected point and automatically orient the camera to the selected location. We kept the camera icons on the background orthophoto for our proposed system to keep the experiment consistent with the baseline, though we could have left them off as the user does not need to know anything about camera placement to use our system.

In the experiment, ten subjects (including a security professional) were asked to orient a live camera feed to various locations in the environment. Ten predetermined locations of interest were chosen in the scene and randomly presented to the subject. Each location was represented by a push-pin icon on the orthophoto with a label describing the location

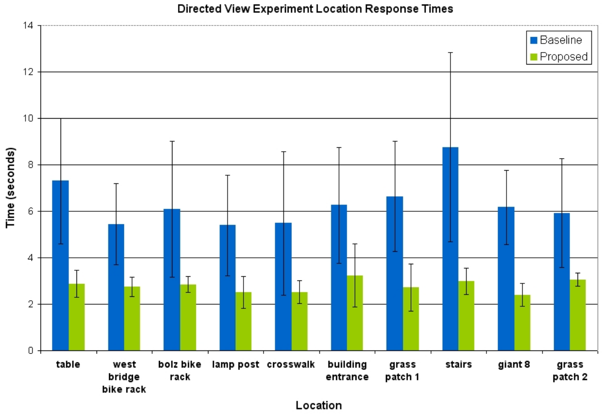

Figure 4. Bar graph summarizing the average time required by subjects to view each of the ten environment locations (shown with one standard deviation).

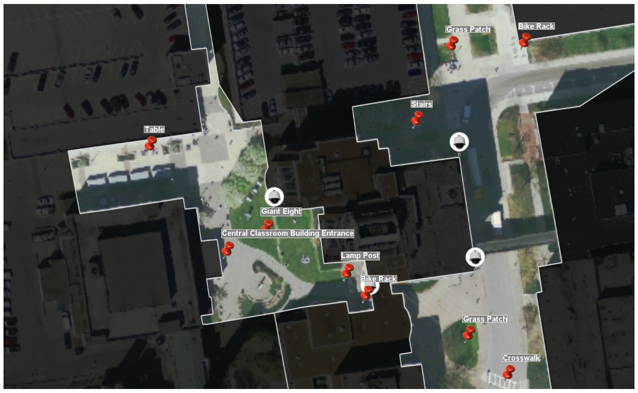

Figure 5. Labels representing real world objects or locations that subjects were asked to view during the directed view experiment.

(see Fig. 5). A location chosen at random was presented to the subject simulating an alarm notification. The subject was then asked to view that location using the camera network. Once the subject felt confident that they were viewing the correct location, they were required to verbally notify the experiment administrator. The time between the event notification and the subject's verbal confirmation was measured. The cameras were reset to a home position before each alarm location was presented to the user. Each subject completed this experiment (for ten locations) using first the baseline and then our proposed system. This ordering was done to prevent subjects from learning the true map-world correspondences from the proposed system (instant mapping) before being asked to search through the environment manually using the baseline system.

As shown with the response times for the two systems in Fig. 4, subjects were able to navigate to the randomized locations more quickly using our proposed system. This was expected due to the nature of the experiment, as simple map interaction is all that is required in the proposed system (no joystick). The average time required to view a location was fairly constant in our proposed system as compared to the baseline system. It took subjects an average of 6.35 seconds (with a standard deviation of 1.03 seconds) to view a location using the baseline system, compared to 2.79 seconds on average (with a standard deviation of 0.26 seconds) using our proposed system.

Eight of the ten subjects that participated in this experiment were familiar with the area of the camera network, though they did not perform significantly better than the other two subjects. The the average times for the eight subjects were 5.99 seconds using the baseline system and 2.47 seconds using the proposed system. The average times for the two subjects not familiar with the area were 7.07 seconds and 2.67 seconds respectively. Furthermore, one subject, who was familiar with the environment, was unable to correctly navigate to one location using the baseline system (not included in the data for Fig. 4). This emphasizes the difficulty of having to manually correlate what is seen through a live camera feed with the environment – even if one is familiar with the environment. Other subjects, also familiar with the environment, had issues in the baseline system when creating a correspondence between what they were viewing through the camera and the location it represented in the environment. This was evident from their repeated gaze between the live camera feed and the map to confirm they were viewing the desired location in the baseline system. This ambiguity does not arise with the proposed system as the user works directly with the environment. Thus the problem of having to correlate the live camera feed with the environment is circumvented.

This experiment demonstrates that the time required to view a location using a joystick in the baseline system is more variable and location dependent than viewing a location using our proposed system. Response times are very important when reacting to alarms or events. The time required to view an event in the environment should not be a function of its location. Our system provides a fast and reliable solution to viewing arbitrary locations and accessing the environment in a consistent manner.

6. Conclusion

Users of distributed camera networks are often faced with the problem of managing and controlling a large number of cameras in an efficient and intuitive manner. We described how cameras can be registered and integrated with their environment to produce a useful interactive system that improves efficiency for controlling such large networks of cameras. To accomplish this we utilized a camera-to-map registration technique and demonstrated how users can manage a distributed camera network and perform an important surveillance task more efficiently. The proposed system addresses the problem of scalability among distributed camera networks, as it allows the user to remain agnostic to camera placement and focus on the environment of the camera network. This approach is extendable to different types of cameras and is applicable to large widespread camera networks.

References

[1] J. Black and T. Ellis. Multi camera image tracking. *Image and Vision Comp.*, 24(11):1256–1267, November 2006. 2

[2] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proc. of the IEEE*, 89(10), 2001. 2

[3] S. Fleck, F. Busch, P. Biber, and W. Strasser. 3D surveillance: A distributed network of smart cameras for real-time tracking and its visualization in 3D. In *Workshop on Embedded Computer Vision*, 2006. 2

[4] J. Gaspar and J. Santos-Victor. Visual path following with a catadioptric panoramic camera. In *International Symposium on Intelligent Robotic Systems - SIRS*, pages 139–147, 1999. 2

[5] G. Iannizzotto, C. Costanzo, F. La Rosa, and P. Lanzafame. A multimodal perceptual user interface for video-surveillance environments. In *Int. Conf. on Multimodal Interfaces*, 2005. 2

[6] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 25(10):1355–1360, October 2003. 2

[7] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *International Workshop on Visual Surveillance*, pages 3–10, 2000. 2

[8] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 22(8):758–767, August 2000. 2

[9] K. Sankaranarayanan and J. Davis. An efficient active camera model for video surveillance. In *Proc. Wkshp. Applications of Comp. Vis.*, 2008. 2, 3

[10] K. Sankaranarayanan and J. Davis. A fast linear registration framework for multi-camera gis coordination. In *Advanced Video and Signal Based Surveillance*, 2008. 1, 3