Milestone: Automatic Camera Network Topology Recognition

Christian Elder, Judson Wilson, Tony Vivoli
Mentor: Alexandre Alahi

I. Introduction

As the price of digital cameras has dropped, systems of networked cameras have become more prevalent. Furthermore, the cost of computation has dropped low enough that video processing is a viable option for many applications of camera networks. Such applications include analysis of consumer patterns in marketplaces, or use in crime prevention.

While the hardware is now mature and available on the market, there are still many problems to solve and areas for improvement. Manual calibration of such systems still remains an exhausting and expensive task. Cameras that are precalibrated and RGB-D (color and depth) equipped will soon be the norm, but this only solves half of the problem. To make real use of a network of cameras, the position and orientation of each cameras in the world must be known, at least relative to a common frame of reference. Traditionally this requires many many hours of manual work, and does not scale well as more and more cameras are added.

In theory, this calibration information can be inferred from the spatial and temporal information of the scene. Existing techniques can be applied to identify moving objects within the scene from RGB-D data. In the course of this project, we endeavour to create an algorithm which will use the estimated trajectories, or tracklets of the objects in the scene, combined with statistical inference techniques and geometry, to estimate the relative locations and orientations of the cameras from which the data originated.

II. Related Work

The problem of determining relationship between cameras with non-overlapping views is not new, and there are several approaches to the problem. Other work has been done to determining the network topology of the camera system [2], [3]. These algorithms typically estimate connectivity between cameras, in the sense of estimating transition probabilities of objects traversing a network and are observed at the camera nodes. They give little insight into the geometric relationship between the cameras. However, these network topologies may be useful in determining corresponding objects between cameras.

[4] gives an algorithm to determine the relative position and rotation of cameras within a network, however it requires that a single object move through the network during a manual calibration phase. One of the goals of our algorithm is that it will be able to be calibrated with normal daily traffic in the system, and not require any manual calibration at all.

To our knowledge, no paper has been published which thoroughly outlines an algorithm which can estimate the external calibration of a network of cameras without prior information or manual calibration steps.

III. Algorithm

A. Algorithm Hypothesis

The underlying idea behind the algorithm is that, for the planar world case, relative camera orientations can be determined by identifying the image of a line in both cameras, which constrains the relative positions of two cameras to a line parallel to the imaged line. The position of the camera on this line is then further constrained to a point by identifying a point correspondence on the line image.

While there are many ways that such a line and point could be identified; we chose to use the paths of moving objects. If an object is moving at constant speed and direction, the path will be a perfect line. If the object produces a linear tracklet in the first camera, then a speed and direction estimate can be determined. Afterwards, any observation by any camera at a later time can be extrapolated to calculate a corresponce to a specific point on this line in the first cameras coordinate system. If a second camera observes this object, a point correspondence and a line correspondence will be known. This is enough information to determine the relative position and angles of the two cameras in the 2D floorplan.

Given a sufficient number of correspondences between many cameras, the external calibration can be determined.

IV. Dataset

Our dataset comes from a camera system installed in the Lausanne railway station, provided by Alexandre Alahi. The system uses RGB-D cameras to detect moving objects through the station, and a software pipeline is used to create tracklet data which describes the motion of objects through the cameras fields of view. The data is given in planar (x,y) coordinates for each timestep, in the coordinate system of the observing camera, as the object moves along a track. Object identity is not tracked between cameras.

V. Results

We ran our current algorithm on simulated data with varying path curvature and additive Gaussian noise (see Figures 1,2), generating plots of the estimated camera relation for each correspondence (see Figure 3). When we injected no noise or false correspondences, the resulting camera relations mapped

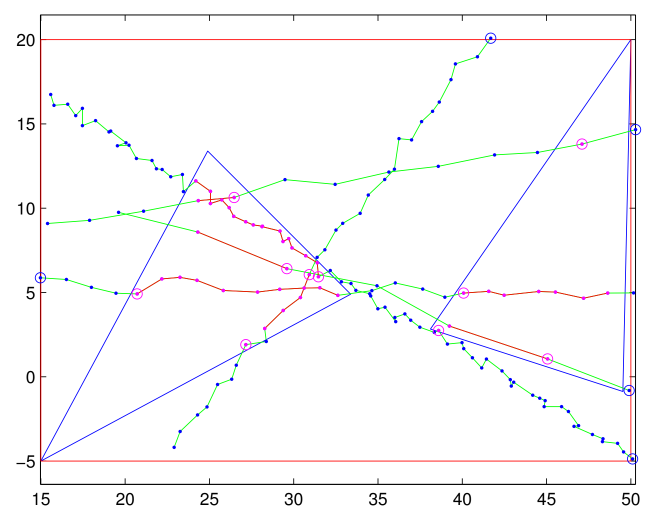

Fig. 1. **Noisy Simulation** Simulation introduces Gaussian noise in both speed and direction of movement across the world

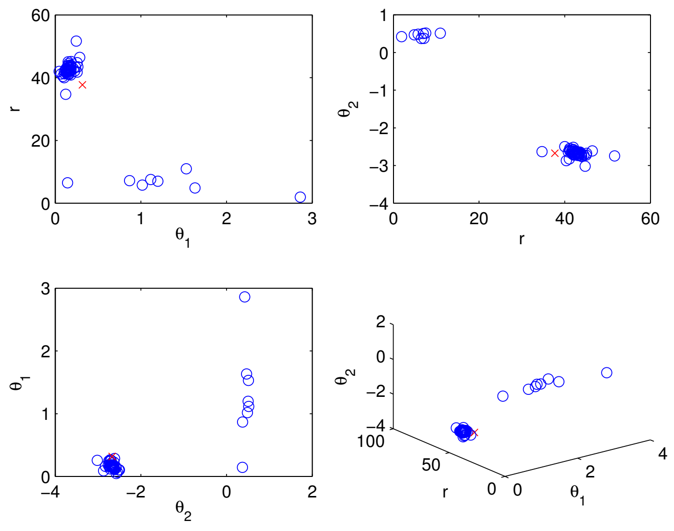

Fig. 3. **Estimated Camera Relation** The algorithm produces an estimate of the camera relation between two cameras for each correspondence. The above plots show the estimates for the camera relation from Figure 2.

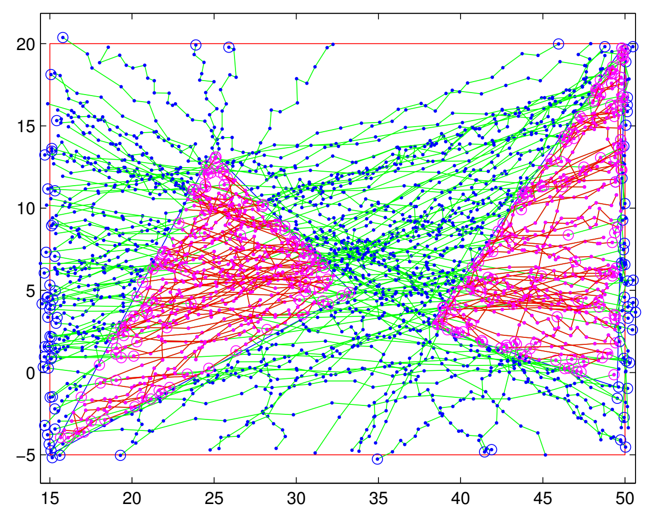

Fig. 2. **Simulated Track Map** Simulation produces tracks across world space (green), which intersect cameras field of view (pink triangle). Data input to localization algorithm is shown in pink

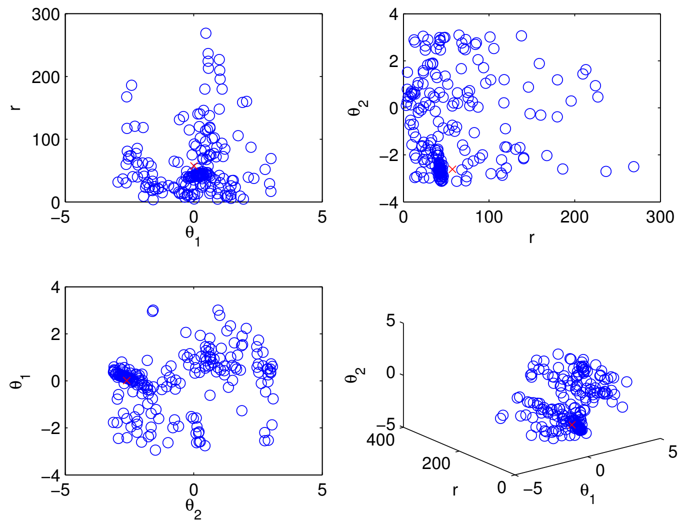

Fig. 4. **Estimated Camera Relation with More False Correspondences** The estimates above are produced from tracks which produce more false correspondences due to multiple tracks entering and exiting the fields of view at a given timestep.

to exactly the correct values. As we introduced noise, curved paths, varying velocity, and false correspondences, the resulting estimates became more diffuse. However, the data points are still clustered around the correct values.

Figure 3 depicts estimates computed from tracks with little noise and few tracks per time interval, resulting in very dense clusters about the actual relative orientations of the cameras. Figure 4 shows camera relation estimates produced when multiple tracks occur per time interval, resulting in more false correspondences. There is, however, still a distinct cluster around the actual relative orientations of the cameras.

VI. Milestone Goals

See Table I and II for achieved and remaining milestones.

Week	Milestone
1/26-2/1	- Determine Project Topic
	- Write Project Proposal
	Thurs 1/30 - Proposal Due
2/2-2/8	- Literature Survey
	- Familiarize with Data Set
	- Begin Algorithm Dev. - Toy Problem
2/9-2/15	- Finish Toy Problem
	- Algorithm Dev. - Handpicked Data
	- Write Milestone Progress Report
	Thurs 2/13 - Milestone Progress Due

TABLE I: Milestones Achieved

Week	Milestone
2/16-2/22	- Algorithm Dev. - Continue with Handpicked Data
	- Expand Dataset, Test, Develop
2/23-3/1	- Expand Dataset, Test, Develop
3/2-3/8	- Measure Performance with Fewer Cameras
3/9-3/15	- Produce Graphical Demo of Results
	Tues 3/11 - Project Presentations (I)
	Tues 3/11 - Project Presentations (I)
3/16-3/19	- Write Final Report
	Wed 3/19 - Final Report Due

TABLE II: Remaining Milestones

VII. Conclusion

The conclusion goes here.

References

[1] O. Javed, Z. Rasheed, O. Alatas, and M. Shah, *KNIGHT: a real time surveillance system for multiple and non-overlapping cameras*, in Proceedings of 2003 International Conference on Multimedia and Expo, 2003, pp. I-649.

[2] T. Ellis, D. Makris, J. Black, *Learning a multi-camera topology*, in Joint IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), 2003, pp. 165-171.

[3] D. Makris, T. Ellis, J. Black, *Bridging the gaps between cameras*, in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, pp. II-205.

[4] A. Rahimi, B. Dunagan, and T. Darrell, *Simultaneous calibration and tracking with a network of non-overlapping sensors*, in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, pp. I-187.