Relative Camera Localisation in Non-overlapping Camera Networks Using Multiple Trajectories

Vijay John, Gwenn Englebienne, and Ben Krose

Intelligent Autonomous Systems Group,
University of Amsterdam, Amsterdam, Netherlands
{v.c.k.john,g.englebienne,b.j.a.krose}@uva.nl

Abstract. In this article we present an automatic camera calibration algorithm using multiple trajectories in a multiple camera network with non-overlapping field-of-views (FOV). Visible trajectories within a camera FOV are assumed to be measured with respect to the camera local co-ordinate system. Calibration is performed by aligning each camera local co-ordinate system with a pre-defined global co-ordinate system using three steps. Firstly, extrinsic pair-wise calibration parameters are estimated using particle swarm optimisation and Kalman filtering. The resulting pair-wise calibration estimates are used to generate an initial estimate of network calibration parameters, which are corrected to account for accumulation errors using particle swarm optimisation-based local search. Finally, a Bayesian framework with Metropolis algorithm is adopted and the posterior distribution over the network calibration parameters are estimated. We validate our algorithm using studio and synthetic datasets and compare our approach with existing state-of-the-art algorithms.

1 Introduction

Estimation of camera parameters (intrinsic and extrinsic) are essential to perform metric world reconstruction and measure real world distances based on their image projections. The vision problems using camera calibration information include tracking, activity recognition, visual reconstruction etc with applications in security, surveillance, bio-medical analysis, character animation etc [1]. In large environment applications, typically, a network of cameras are deployed with non-overlapping field-of-views (FOV) [2]. Calibration in such a network is difficult without the aid of external calibration objects, usually requiring manual intervention.

In this article, we present an algorithm for extrinsic calibration of non-overlapping FOV camera network. Multiple trajectories of objects defined with respect to a pre-defined global co-ordinate system are termed as global trajectories. Global trajectories visible in a particular camera FOV are measured with respect to the camera co-ordinate system and are termed as local trajectories. Calibration can be performed by aligning local co-ordinate systems with

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part III, LNCS 7585, pp. 141–150, 2012.

the global co-ordinate system. However, in order to perform the alignment, the global trajectories need to be derived from the measured local trajectories, but this problem is complicated owing to the lack of overlap between network cameras.

Overview. In our proposed calibration approach, firstly, we consider pair-wise cameras in the network and estimate their extrinsic pair-wise calibration parameters using particle swarm optimisation(PSO)[3] and linear Kalman filter. The estimated pair-wise calibration parameters are used to generate an initial estimate of network calibration parameters, using a weighted directed graph. To account for pair-wise calibration errors, estimated network calibration parameters are corrected using PSO-based local search with forward-backward Kalman filter and K-L divergence based cost function. Finally, a Bayesian framework is formulated to obtain a posterior distribution over the extrinsic network calibration parameter, with priors defined on the corrected network calibration estimate. The likelihood term in our posterior distribution incorporates the trajectory uncertainty obtained using Kalman smoothing. The expected value of calibration parameters are calculated from sequence of samples generated from the posterior distribution using Metropolis algorithm [4], a Markov Chain Monte-Carlo method.

To the best of our knowledge, our work contributes to the state-of-the-art in at least three ways. First, the use of multiple trajectories for extrinsic camera pose estimation in non-overlapping FOV has not been reported. Secondly, we incorporate the uncertainty associated with each trajectory in the Bayesian framework. Thirdly, we automatically initialise the parameter estimation.

The rest of this paper is organised as follows. Section 2 summarizes some of the relevant literature in relation to our work. Section 3 presents our algorithm. Section 4 presents experimental results of our proposed algorithm on our studio dataset and synthetic datasets. Section 5 summarizes our work and suggests future development.

2 Related Work

There are two different classes of camera calibration literature in non-overlapping FOV networks, which we briefly review next. In the first class of literature, external objects are used to aid in calibration, for example, assistive sensors like mirrors [5]. Recently, Micusik et al. [6] use an inertial sensor in their work to obtain a closed-form solution for intrinsic and extrinsic camera parameters.

In the second class of literature, where our proposed algorithm lies, moving objects in camera FOV are used to estimate the parameters. Javed et al. [7] use velocity extrapolation to project FOV of one camera to the other and estimate camera parameters. Later, Rahimi et al. [8] adopted a maximum a posterior (MAP) framework to perform simultaneous tracking and calibration using single estimated trajectories. A linear motion model is used as motion prior and MAP estimates are obtained using Newton-Raphson's method. Nadeem et al. [2] use forward-backward regression to generate trajectories in non-overlapping FOV region, which are used to estimate the parameters. A common feature of the

second class of literature is the assumption of linear trajectories with additive noises in the non-overlap region, with additive trajectory noise estimating non-linearity.

3 Camera Calibration Algorithm

In our work, we make the following assumptions, given a network of N non-overlap FOV cameras $\mathbf{c} = \{c^n\}_{n=1}^{N}$, we assume that each c^n can map image coordinates of k visible trajectory in its FOV to its 2-D local co-ordinate system as $X_k^n = \{\mathbf{x}_k^n(t)\}_{t=1}^{T}$. Similar to [2,8], we also assume that each camera provides a top-down view of the scene with its optical axis being perpendicular to the ground plane. Next, we assume that data association for multiple trajectories both within a camera FOV and across different camera FOV are available.

The extrinsic network calibration parameters ($\mathbf{M_g}$) that need to be estimated are given as $\mathbf{M_g} = \{\mathbf{m}_g^n\}_{n=1}^{N}$ where $\mathbf{m}_g^n = [\mathbf{q}_g^n, \theta_g^n]$ contains 2-D translation $\mathbf{q}_g^n = [qx_g^n, qy_g^n]$ and 2-D rotation θ_g^n. An overview of our algorithm is shown in Figure 1(a) and detailed description of steps involved are given below. Additionally, a layout of network and parameters are shown in Figure 1(b).

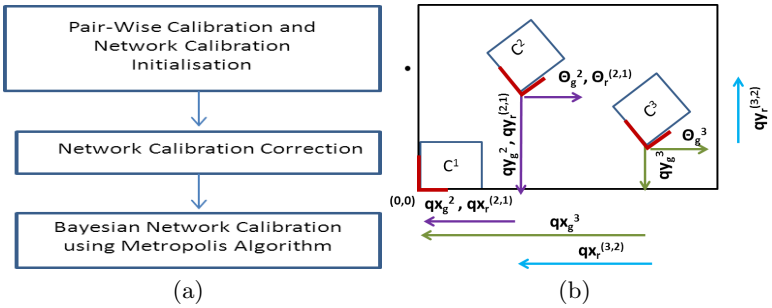

Fig. 1. (a)The overview of our algorithm. (b) A visualisation of camera network layout along with the calibration parameters (network and pair-wise) to be estimated are given.

3.1 Pair-Wise Calibration Parameter Estimation

To estimate pair-wise calibration parameters, represented as $\mathbf{m}_r^{(i,j)}$ for pair (i, j), firstly, camera pairs in the network are selected using FOV exit-entrance time instants. Specifically, cameras c^i and c^j are considered a pair, if a measured local trajectory X_k^i exits c^i at exit time instant t_0 and moves in non-overlap region before entering c^j at entrance time instant t_1, with $j \neq i$, before entering other cameras. The trajectory segments corresponding to camera pair (i, j) and defined in c^i co-ordinate is given as $\overline{X}_k^i = [X_k^i = \{\mathbf{x}_k^i(t)\}_{t=1}^{t_0}, X_k^{(u,i)} = \{\mathbf{x}_k^i(t)\}_{t=t_0+1}^{t_1-1}, \hat{X}_k^i = \{\mathbf{x}_k^i(t)\}_{t=t_1}^{T}]$, where the segments $X_k^{(u,i)}$ in non-overlap FOV and $\hat{X}_k^i$ need to be estimated. Given the estimated trajectory segments, $\mathbf{m}_r^{(i,j)}$

can then be estimated by aligning $\hat{X}_k^i$ with its corresponding trajectory segment defined in c^j co-ordinate system ($\hat{X}_k^j$), where correspondence is established using time stamps.

Non-overlap FOV Prediction. To estimate $X_k^{(u,i)}$, a linear Kalman filter is initialised at measurement ($\mathbf{x}_k^i(1)$) and filtered state estimates $Z_k^i = \{\mathbf{z}_k(t)\}_{t=1}^{t_1}$ are obtained, where $\mathbf{z}_k(t) = [z_k^h(t), \dot{z}_k^h(t), z_k^v(t), \dot{z}_k^v(t)]'$ with 2-D position and velocity components. During Kalman filtering, the measurements in non-overlap region are considered missing. It has been shown that for missing measurement in Kalman filter, the ideal estimator at update step $z_{t|t}$ can be replaced with predictor, $z_{t|t-1}$, with the proof of error residual convergence shown in [9]. However, the posterior error covariance $\Sigma_{t|t}$ corresponding to missing measurements gradually increases. Given, the filtered state estimates, $X_k^{(u,i)}$ is extracted from $Z_k^{u,i} = \{\mathbf{z}_k(t)\}_{t=t_0+1}^{t_1}$ using matrix H given in Eqn 1. The Kalman filter's state transition matrix A is also given in Eqn

$$A = \begin{bmatrix} 1\,1\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,1 \\ 0\,0\,0\,1 \end{bmatrix} \; H = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,0\,1\,0 \end{bmatrix} \tag{1}$$

Trajectory Shape Estimation. We next estimate $\hat{X}_k^i$ using relative angles and magnitude (shape) information from its corresponding trajectory in c^j ($\hat{X}_k^j$), which can improve the calibration estimation. We derive the shape information using the following steps. Firstly, we consider 3-tuple states in $\hat{X}_k^j$, i.e. $\hat{\mathbf{x}}_k^j(t), \hat{\mathbf{x}}_k^j(t+1)$ and $\hat{\mathbf{x}}_k^j(t+2)$ and derive the difference vector (velocity) magnitudes $\hat{\rho}_k^j(t, t+1), \hat{\rho}_k^j(t+1, t+2)$ and angles $\hat{\phi}_k^j(t, t+1), \hat{\phi}_k^j(t+1, t+2)$. Next, the relative angles $\hat{\psi}_k^j(t, t+2)$ between difference vector angles are calculated. Then from the trajectory set $\overline{X}_k^i$, we consider $[X_k^i, X_k^{(u,i)}]$ and derive only the difference vector angles $\hat{\phi}_k^i$. Finally, using all the calculated information, $\hat{X}_k^i = \{\hat{\mathbf{x}}_k(t)\}_{t=t_1+2}^{T}$ is estimated using Eqn 2 and Eqn 3 in a *for* loop from $t = t_1 \ldots T-2$

$$\hat{\phi}_k^i(t+1, t+2) = \hat{\psi}_k^j(t, t+2) + \hat{\phi}_k^i(t, t+1) \tag{2}$$

$$\hat{\mathbf{x}}_k^i(t+2) = \hat{\mathbf{x}}_k^i(t+1) + \hat{\rho}_k^j(t+1, t+2) \begin{bmatrix} cos(\hat{\phi}_k^i(t+1, t+2)) & 0 \\ 0 & sin(\hat{\phi}_k^i(t+1, t+2)) \end{bmatrix} \tag{3}$$

Note that $\hat{\mathbf{x}}_k^i(t_1+1)$ in Eqn 3 is estimated by extending the Kalman filter used in non-overlap prediction.

Particle Swarm Optimisation. We formulate our pair-wise camera calibration estimation as a non-linear optimization problem, that aligns $\hat{X}_k^j$ with $\hat{X}_k^i$, using particle swarm optimisation (PSO) with a distances-based cost function as shown in Equation 4. PSO a non-linear iterative swarm intelligence algorithm [3] comprises of a swarm of candidate particles or states, here pair-wise calibration pa-

rameters ($\mathbf{m}_r$), that solves a problem by iteratively improving the candidate particle location in the cost function. We use a inertia-varying PSO [3] that performs a global-to-local search. We refer the readers to [3] for further details. Our pair-wise camera calibration problem is formulated as follows for camera pair (i, j)

$$\hat{\mathbf{m}}_r^{(i,j)} = \underset{\mathbf{m}_r^{(i,j)}}{\operatorname{argmin}} \sqrt{(\hat{X}_k^i - F(\hat{X}_k^j, \mathbf{m}_r^{(i,j)}))^2} \tag{4}$$

where F(.) represents the function that performs 2D co-ordinate transformation. Our main motivation for incorporating PSO instead of other optimisation algorithm are results obtained by PSO in complex vision problems like articulated tracking [10]. Specifically, PSO has demonstrated automatic initialisation property in addition to constraining the search within pre-defined search limits [3].

Initial Network Calibration Parameter Estimate. An initial estimate of network calibration parameters can be obtained from pair-wise calibration estimates. As done previously [8], we represent the global co-ordinate system at origin of the camera with maximum network connection in the network, for example c^1, and the corresponding unobserved global trajectories are represented as $Y_k = \{\mathbf{y}_k(t)\}_{t=1}^T$. Given this assumption, network calibration parameter can then be calculated from sum of pair-wise calibration parameters i.e. $\mathbf{m}_g^n = \mathbf{m}_r^{(1,j)} + \mathbf{m}_r^{(j,n)}$, with j intermediate cameras. To select the intermediate j cameras, a weighted directed graph is set-up with each node representing a camera and the weighted edges representing a trajectory-connected camera pair. The edge weights are derived from the average time duration taken by trajectories in the non-overlap region between trajectory-connected camera pairs. The edge weights are inversely proportional to the average time duration. Given the weighted directed graph, Djikstra's algorithm is used to calculate the shortest path (j cameras), from which an initial estimate of network calibration parameters are obtained.

3.2 Network Parameters Correction

An inherent problem with calculating the network calibration parameters from pair-wise parameters is the possibility of error accumulation. To account for accumulative errors, a local search is set-up using inertia-varying PSO [3] with forward-backward Kalman filter and K-L divergence based cost function (Eqn 5). Specifically, we consider the transformed set of initial global trajectories $\mathbf{Y} = \{Y_k\}_{k=1}^K$, where the various trajectory segments in Y_k are given as $Y_k = [Y_k^1, \acute{Y}_k^{(u,1)}, \hat{Y}_k^n]$, where $\hat{Y}_k^n$ is the transformed trajectory segment in c^n and $\acute{Y}_k^{(u,1)}$ is the non-overlap segment. Our cost function is designed to align $\hat{Y}_k^n$ with the complete global trajectory using 2-D co-ordinate transformation parameters (PSO state) $\mathbf{M_C} = \{\mathbf{m}_c^n\}_{n=1}^N$. The corrective PSO swarm is initialised around the initial estimate of network calibration parameters.

Cost Function. During PSO optimisation run, for each initial global trajectory, the PSO states in the swarm generate corrected global trajectories, which

are evaluated using a forward-backward Kalman filter and K-L divergence based cost function. Specifically, for each corrected global trajectory, a forward Kalman filter is initialised at starting time instant $Y_k(1)$ generating posterior state estimates $p_f^k(t)$. Additionally, a backward Kalman filter is initialised at final time instant T ($Y_k(T)$) generating posterior state estimates $p_b^k(t)$ in the reverse direction to forward Kalman filter. Given, the Kalman filter estimates a K-L divergence based cost function is set-up as

$$\hat{\mathbf{M}}_\mathbf{c} = \underset{\mathbf{M_c}}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{t=1}^{T} D_{kl}(p_f^k(t)||p_b^k(t)) \tag{5}$$

Since our corrective cost function is designed for linear non-overlap global trajectory segments, we automatically select such global trajectories using a threshold based on non-overlap-segment length. Note that this selection is performed only for corrective optimisation. Next, we describe our Bayesian formulation, where $\hat{\mathbf{M}}_\mathbf{c}$ is used as priors.

3.3 Bayesian Framework

The extrinsic network calibration parameters $\mathbf{M_g}$ are estimated by formulating a Bayesian framework. Specifically, a posterior distribution over $\mathbf{M_g}$ given as

$$p(\mathbf{M_g}|\mathbf{X}) \propto p(\mathbf{X}|\mathbf{M_g})p(\mathbf{M_g}) \tag{6}$$

The likelihood distribution for the Bayesian formulation is then set up as

$$p(\mathbf{X}|\mathbf{M_g}) = p(F(\mathbf{X}, \mathbf{M_g})) \tag{7}$$

where F(). performs coordinate transformation. To obtain $P(F(\mathbf{X}, \mathbf{M_g}))$, we use the Kalman smoother. First, we consider $F(\mathbf{X}, \mathbf{M_g})$ to be the measurement for state $\mathbf{Z}$. Specifically, the functional form of Kalman smoother is given as follows

$$\mathbf{z}_k(t+1) = A\mathbf{z}_k(t) + w_t \tag{8}$$

$$f(\mathbf{x}_k(t), m_g) = H\mathbf{z}_k(t) + v_t \tag{9}$$

$$\tag{10}$$

where $\mathbf{z}_k(t) = [z_k^h(t), \dot{z}_k^h(t), z_k^v(t), \dot{z}_k^v(t)]'$ is state with 2-D position and velocity, A and H are state transition and measurement matrix given in Eqn 1, w_t and v_t are normally distributed process and measurement noise with zero mean and learnt covariances [11]. Kalman smoothing obtains the state estimate for all $Z_k = \{\mathbf{z}_k(t)\}_{t=1}^T$, where the state estimate $\mathbf{z}_k(t)$ at t is given as

$$p(\mathbf{z}_k^n(t)|\{f(\mathbf{x}_k^n(t), \mathbf{m_g})\}_{t=1}^T) = \mathcal{N}(\mathbf{z}_k^n(t); \mu_{zk}^n(t), \Sigma_{zk}^n(t)), \tag{11}$$

where $\{f(\mathbf{x}_k^n(t), \mathbf{m_g})\}_{t=1}^T$ is the complete measurement. Kalman smoother's posterior state estimate and error covariance at t is given by $\mu_{zk}^n(t)$ and $\Sigma_{zk}^n(t)$. Given the state estimates the likelihood is then formulated as.

$$p(F(\mathbf{X}, \mathbf{M_g})) \propto \prod_{k,n,t \in D} p(F(\mathbf{x}_k^n(t), \mathbf{m_g}^n); H\mu_{zk}^n(t), H\Sigma_{zk}^n(t)H') \tag{12}$$

where D is set of k independent trajectories visible in c^n camera at t, with $(t, n, k) \in D$. Furthermore, 2-D rotation parameter θ_g^n and translation parameters q_g^n in each $\mathbf{m}_g^n$ are considered independent, as a-priori, resulting in posterior distribution represented as.

$$p(\mathbf{M}|\mathbf{X}) \approx \prod_{k,n,t \in D} p(F(\mathbf{x}_k^n(t), \mathbf{m}_g^n); H\mu_{zk}^n(t), H\Sigma_{zk}^n(t)H')p(\theta_g^n)p(q_g^n) \tag{13}$$

The prior distribution over angles $p(\theta_g^n)$ is a Von-Mises distribution, a circular normal distribution, given in Eqn , where μ and $1 \setminus \kappa$ are analogous to mean and covariance of normal distribution and I_0 is a modified Bessel function of order 0. The translation prior distributions $p(q_g^n)$ are given as Gaussian distributions. The mean of Von-Mises and Gaussian priors are based on corrected network calibration estimates while the covariance is set based on empirical trials.

$$p(\theta|\mu, \kappa) = \frac{e^{\kappa cos(\theta - \mu)}}{2\pi I_0 \kappa} \tag{14}$$

Finally, to derive the expected value of the network calibration parameters $E(\mathbf{M_g})$, we use Metropolis algorithm [4], a Markov chain Monte carlo method (MCMC), to generate a sequence of samples from the posterior distribution in Equation 6. The proposal distributions for our algorithm are also based on corrected network calibration estimates. Specifically, a Von-Mises angle proposal distribution and Gaussian translation proposal distribution. As is the case with Metropolis algorithms, in our generated sequence of samples, while calculating the expected value, we consider samples after a burn-in period b and only take every s^{th} sample, described in the experiments below, from the sequence of samples.

4 Experimental Results

Our proposed algorithm was validated using our studio dataset as well as a synthetic dataset. To the best of our knowledge, we are the first work to estimate calibration parameters using multiple trajectories, hence for comparative purposes, we could only compare our algorithm with single trajectory algorithm by Rahimi et al. [8], considered as state-of-art in literature.

Studio Dataset
Our studio dataset consists of 3 sequences captured using 5 non-overlapping FOV stereo cameras, capturing @15 Hz mounted on the ceiling to provide a top-down view of the environment. The studio environment, represented an indoor retail shop with $2-4$ subjects performing actions simulating a shopping environment. Our in-house tracking software generates the multiple trajectories, on which manual data-association is performed. These manually annotated trajectories function as input to our algorithm. The layout and motion characteristics of our studio is shown in Figure 2, where non-overlap trajectory segments are typically non-linear between c^3 and c^4, while being linear in the remaining non-overlap regions.

Table 1. Comparative calibration errors (mean,std.dev) between our algorithm and MAP [8] (b) Detailed calibration errors (mean,std.dev) of our algorithm on the studio dataset

Algorithm	Studio			Synthetic		
	θ(rad)	qx (m)	qy (m)	θ(rad)	qx (m)	qy (m)
Multiple Track	0.23±0.1	0.46±0.2	0.34±0.2	0.12±0.1	0.31±0.1	0.28±0.2
MAP [8]	0.25±0.1	0.56±0.1	0.39±0.2	0.2±0.1	0.43±0.1	0.36±0.1

(a)

Cam 2			Cam 3		
θ(rad)	qx (m)	qy (m)	θ(rad)	qx (m)	qy (m)
0.24±0.1	0.47±0.3	0.27±0.2	0.24±0.2	0.51±0.2	0.32±0.1
Cam 4			**Cam 5**		
θ(rad)	qx (m)	qy (m)	θ(rad)	qx (m)	qy (m)
0.26±0.1	0.42±0.2	0.47±0.4	0.23±0.2	0.6±0.4	0.65±0.5

(b)

Algorithm parameters. The location and orientation of the cameras in the network with respect to the global co-ordinate system (pre-defined camera) was measured manually. In the pair-wise calibration estimation, we used 5 particles for 100 iterations, with starting inertia 2.0 and ending inertia 0.1. Additionally, the same initial state or calibration parameter is used for all camera-pairs reflecting automatic initialisation of PSO. In the corrective optimization step, we used 3 particles with same ending inertia and 0.5 starting inertia to simulate a local search [3]. Finally, in our component-wise Metropolis algorithm, the Von-Mises distribution had κ set to 10, while the Gaussian distribution had covariance of 0.1. Furthermore, the burn-in period b was set to 1000, and we sampled every 10 sample (s). The number of time steps in our MCMC method was set to 2500.

Results. As shown in Table 1 and Figure 2, our proposed algorithm is able to accurately estimate the network extrinsic calibration parameters. The average errors reported are obtained using distances between our algorithm's estimates and the ground truth parameters. Note that the average calibration errors in Table 1a (Studio dataset) are based on manually selected single long trajectories, as a result of MAP algorithm limitation [8]. Specifically, [8] require single long trajectory across all camera FOV.

Synthetic Dataset

We have generated sequences from three layouts of non-overlap cameras, *zig-zag*, *diamond* and *square*, with 3 trajectories each. For the algorithm parameters, we utilised the same parameters as used for our studio dataset. The layout and motion characteristics of the synthetic dataset is shown in Figure 2.

Comparison with Rahimi et al.[8]. As shown in Table 1a, our algorithm performs marginally better or similarly to Rahimi et al. [8]. Additionally, our algorithm has demonstrated automatic initialisation. In case of the work by Rahimi et al. [8], based on Newton-Raphson's method, it can be seen that to obtain an optimal calibration estimate a good initialisation of parameters are needed, as wrong initialisation could result in wrong convergence [12]. Finally, on comparing

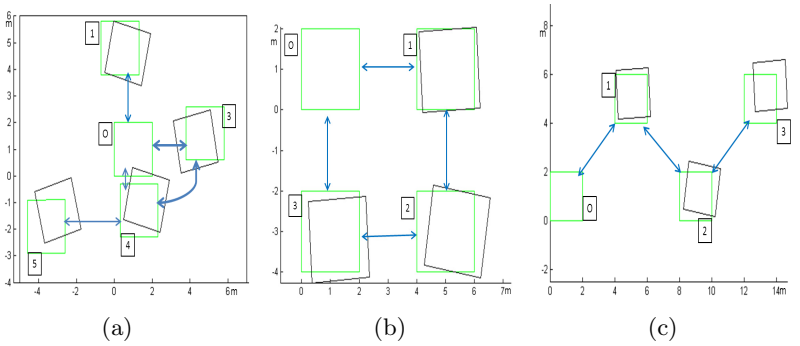

Fig. 2. Visualisation of Network Calibration parameter estimates for (a)Studio dataset, (b) Square dataset and (c) Zig-zag dataset. The green box represents ground truth camera position, while black box represents our algorithm's estimate. The motion observed in the non-overlap region between camera pairs is represented by arrows. The global co-ordinate cam is represented by 'O'.

the Bayesian frameworks, [8] obtain the MAP point estimate of the posterior distribution, while we obtain an approximation of the posterior distribution itself.

5 Conclusion and Future Work

We have presented a extrinsic network calibration parameter estimation algorithm for non-overlapping FOV cameras network using multiple trajectories. First, we estimate pair-wise camera calibration parameters using PSO, which are then used to estimate global trajectories. Next, we perform a local search on the global trajectory and correct it. Finally, a Bayesian formulation with Metropolis algorithm is adopted and posterior over calibration parameters are obtained. We demonstrate good calibration accuracies on our studio and synthetic datasets.

Our proposed algorithm obtains good calibration accuracies on our test datasets as observed in Table 1 and Figure 2. However, it can be seen that as the camera distance from global co-ordinate camera increases there is a relative error increase, for example, Cam 4 in Zig Zag dataset (Figure 2c). Similarly, cameras not directly connected to global co-ordinate camera also report relatively higher errors, for example, Cam 5 in our studio dataset (Figure 2a). In our future work, we plan to address this issue, by either adopting a switching linear dynamic model within the Kalman filter for curved trajectory prediction.

Acknowledgments. This work was supported by a grant from Point One UII project PNU10C18 in a collaboration with Philips Research, Eindhoven and Eagle Vision Systems, The Netherlands (EVS). EVS provided the Eagle-Z stereo cameras and tracking software for our experiments with the studio dataset being captured at Philips Research.

References

1. Moeslund, T.B., Hilton, A., Krger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU 104, 90–126 (2006)
2. Anjum, N.: Camera localization in distributed networks using trajectory estimation. Journal of Electrical and Computer Engineering (2011)
3. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimizer. In: The IEEE International Conference on Evolutionary Computation Proceedings (1998)
4. Beichl, I., Sullivan, F.: The metropolis algorithm. Computing in Science and Engg. (2000)
5. Kumar, R.K., Ilie, A., Frahm, J.M., Pollefeys, M.: Simple calibration of non-overlapping cameras with a mirror. In: CVPR (2008)
6. Micusik, B.: Relative pose problem for non-overlapping surveillance cameras with known gravity vector. In: CVPR (2011)
7. Javed, O., Rasheed, Z., Alatas, O., Shah, M.: Knight: A real time surveillance system for multiple overlapping and non-overlapping cameras. In: ICME (2003)
8. Rahimi, A., Dunagan, B., Darrell, T.: Simultaneous calibration and tracking with a network of non-overlapping sensors. In: CVPR (2004)
9. Chen, G.: A simple treatment for sub-optimal kalman filtering in case of measurement data missing. IEEE Transactions on Aerospace and Electronic Systems (1990)
10. John, V., Trucco, E., Ivekovic, S.: Markerless human articulated tracking using hierarchical particle swarm optimisation. Image Vision Comput. 28, 1530–1547 (2010)
11. Ghahramani, Z., Hinton, G.E.: Parameter estimation for linear dynamical systems. Technical report, University of Toronto (1996)
12. Kelley, C.T.: Solving Nonlinear Equations with Newton's method. Fundamentals of Algorithms (2003)