RELATIVE POSITION ESTIMATION OF NON-OVERLAPPING CAMERAS

*Nadeem Anjum, Murtaza Taj, Andrea Cavallaro**

Multimedia and Vision Group
Queen Mary, University of London
Mile End Road, E1 4NS London (United Kingdom)
Email: {nadeem.anjum,murtaza.taj,andrea.cavallaro}@elec.qmul.ac.uk

ABSTRACT

We present an algorithm for the estimation of the relative camera position in a network of cameras with non-overlapping fields of view. The algorithm estimates the missing trajectory information in the unobserved areas of the multi-sensor configuration using both parametric and non-parametric algorithms. First, Kalman filtering is used to estimate the trajectories in the unobserved regions. Next, linear regression estimates the position of the target based upon the motion model generated from the measured positions in the field of view of each sensor. Finally, the relative orientation of the sensors is calculated using the observed and estimated target position from adjacent cameras. We demonstrate the algorithm on both synthetic and real data.

Index Terms— Surveillance, distributed tracking, calibration, Kalman filtering, image sensors.

1. INTRODUCTION

The use of multiple cameras in visual surveillance enables the monitoring of wide areas and hence the detection of actions and events of interest on a larger scale, compared to the use of single sensors separately. To exploit multiple cameras, an automatic mechanism is needed that fuses data from the various sensors and generates a global ground plane view of the overall scene. The movements of the targets (trajectories) can then be reproduced in the global view for automated event detection or for visualization.

In many surveillance scenarios the network of cameras does not fully cover the area to be monitored. As a consequence, unobserved areas exist where target information is missing. Moreover, the extrinsic calibration parameters of each camera may be unknown or difficult to obtain in a number of scenarios. Examples of such scenarios are wide area indoor surveillance and mobile ad-hoc networks of low-cost surveillance cameras. In these cases a solution is needed that can estimate the missing trajectories information and the extrinsic cameras parameters to form a global ground-plane view with complete trajectories.

The problem of Simultaneous Localization and Tracking (SLAT) is addressed in [1] using maximum a-posteriori estimation (MAP) combined with Newton Raphson's method. Parzen windows are used in [2] to learn the camera topology and path probabilities. First probabilities are calculated in a training phase, and then camera correspondence is measured using MAP estimation. Approximated Bayesian filtering is used in [3] to provide on-line probabilistic estimates of sensor locations and target tracks, whereas a Bayesian formulation is used in [4] to construct the paths of the moving objects in a non-overlapping multiple camera network. A multi-camera calibration scheme that uses a coordinate measuring machine (CMM) to generate target points for camera calibration and integration is presented in [5]. The CMM identifies the target location points that are then used for camera calibration.

In this paper, we propose an iterative statistical model to recover the position of the cameras and the missing trajectory information. The model estimates missing trajectories across the unobservable regions and uses both parametric and non-parametric algorithms. Kalman filtering estimates the target trajectories in unobserved regions and linear regression estimates the position of the target based upon the motion model generated from the measured positions in the field of view of each sensor. Once the missing trajectory information is estimated, the relative angles between the sensors are calculated based on the observed and estimated target positions.

The paper is organized as follows. In Section 2 we formulate the problem. Section 3 describes the proposed algorithm for the estimation of the relative camera position. In Section 4 we discuss the experimental results on both synthetic and real data. Finally, Section 5 presents the conclusions.

2. PROBLEM FORMULATION

Let the object observations (trajectories) be provided by a network $\Psi = \{C^1, C^2, \ldots, C^N\}$ of N cameras with non-overlapping fields of view. Let a trajectory T be represented as $T = \{(x_j^i, y_j^i) : 0 < j < M_i; i = 1, \ldots, N\}$, where

*The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/D033772/1

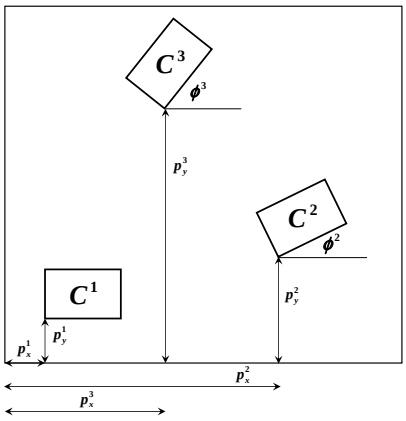

Fig. 1. *Schematic representation of a scene observed with non-overlapping cameras.* $P^i = (p^i_x, p^i_y, \phi^i)$ *represents the unknown camera position and rotation to be estimated*

(x^i_j, y^i_j) is the estimated position of the target in the image plane and M_i is the number of target observations from camera C^i.

Let each cameras C^i provide a vertical top-down view of the scene, as shown in Figure 2. We assume therefore that the trajectories are preprocessed using a homography transformation [6] or that the cameras are mounted so that their optical axis is perpendicular to the ground plane. Under this assumption, the number of parameters for the localization of each camera C^i is reduced to three, namely the camera position, $P^i = (p^i_x, p^i_y)$, and the rotation angle, ϕ^i, expressed as the relative angle between the camera and the horizontal axis. To summarize, the unknown parameters Θ^i for camera C^i are

$$\Theta^i = [p^i_x, p^i_y, \phi^i]. \tag{1}$$

By fixing one camera as a reference, the objective is to estimate the camera configuration

$$\Theta = [\Theta^1, \Theta^2, \Theta^3, \ldots, \Theta^{N-1}], \tag{2}$$

as discussed in the next section.

3. PROPOSED ALGORITHM

We use the target trajectories observed in each camera C^i to estimate the unknown parameter Θ^i. To locate the unknown camera configuration, the missing trajectory data are estimated after Kalman filtering the observed data. Next forward and backward linear regression are used in the unobserved regions to propagate a motion model consistent with the Kalman estimations.

Let us define the target state X_t at time t as

$$X_t = [x, \dot{x}, y, \dot{y}], \tag{3}$$

where (x, y) is the target position and $(\dot{x}, \dot{y})$ is the target velocity. The state X_{t+1} at time $t+1$ is defined as

$$X_{t+1} = AX_t + V_t \tag{4}$$

where

$$A = \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix A is the observation model which transforms the target state at time t to the next state at time $t+1$. The matrix used here ensures that the target follows a smooth path. V_t is the process additive noise that models small variations in the motion of the target and is assumed to be zero-mean Gaussian noise with covariance Σ_v, defined as

$$\Sigma_v = diag[10^{-10}, 10^{-6}, 10^{-10}, 10^{-6}].$$

The observation model can be expressed as

$$Z^i_t = R^i(LX_t - P^i) + W_t \quad i = 1, \ldots, N \tag{5}$$

where R^i is the rotation matrix

$$R(\theta) = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix},$$

and

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

W_t is the measurement Gaussian noise with variance Σ_w, given by $\Sigma_w = diag([10^{-10}10^{-10}])$.

The missing trajectory points between two consecutive sensors C^i and C^{i+1} are computed in two steps, using forward and backward estimation. Forward estimation computes the missing trajectories from C^i to C^{i+1}. Backward estimation computes the missing trajectories from C^{i+1} to C^i. The final missing trajectories between C^{i+1} and C^i are estimated by fusing the results of the two estimates.

Kalman filtering is applied to the M_i target observations in each camera C^i. The prediction at $t+1$ uses the estimate from t to compute the current target state (Eq. 4). Similarly, the error covariance matrix is estimated with $\Sigma_v(t+1) = A\Sigma_v(t)A^T$. Next, measurement information from time t is used to refine the predictions using the Kalman gain $K_t = \Sigma_v(t)/(\Sigma_v(t) + R)$. The estimates are then updated with the measurement as

$$X_{t+1} = X_t + K_t(Z_t - X_t). \tag{6}$$

The smaller the residual $(Z_t - X_t)$ the higher the agreement between the estimation and the measurement. Finally the error covariance is updated as $\Sigma_v(t) = (I - K_t)\Sigma_v(t)$.

Equations (4)-(6) are used iteratively for all the available noisy measurements. Next, trajectory data in the unobserved regions are derived using forward and backward estimation.

For forward estimation, the filtered C^i data are represented as a second order polynomial ($y = a_0t^2 + a_1t + a_2$) for both (horizontal and vertical) directions. To estimate the target trajectory in the unobserved regions, the coefficients a_k ($k = 0, 1, 2$) of the polynomial are computed using linear regression. The noisy estimations are Kalman filtered to generate the final trajectory data from C^i to C^{i+1} for the forward estimation step.

To obtain a more robust estimation of the trajectory data in the unobserved regions, the same process is applied backwards from C^{i+1} to C^i. Finally, the forward and backward estimation results are averaged to obtain the final missing trajectory estimation. The same process is repeated for all the adjacent sensor pairs to estimate the missing trajectories throughout the network.

The relative angle between C^i and C^{i+1} is computed by calculating the angle between P, the observed target position in sensor C^{i+1}, and $\widehat{P}$, the estimated target position in the same sensor estimated from the adjacent sensor C^i. This relative angle between consecutive cameras, $\phi_{i,i+1}$, is computed as

$$\phi_{i,i+1} = \cos^{-1} \frac{P \cdot \widehat{P}}{|P||\widehat{P}|} \tag{7}$$

for the complete network. Finally, all cameras C^i are rearranged with respect to the reference sensor C^1 to obtain the final configuration of the network. For a network of N cameras, the algorithm estimates the configuration in $2N - 2$ iterations. The results of the algorithm are discussed in the next section.

4. EXPERIMENTAL RESULTS

In this section, the proposed algorithm for the estimation of the relative camera position is demonstrated on synthetic data and on real data. Synthetic target data are generated for a network of $N = 4$ and for a network $N = 8$ cameras (Figure 2). The estimated camera configuration is shown in brown, whereas the exact sensor position is indicated in black to visualize the localization error. Based on 10 datasets with $T_p = 10000$ trajectory points, for $N = 4$, the average estimated orientation error per camera is $\epsilon_o = 1°$ and the average estimated position error is $\epsilon_p = 0.04$ units (Table 1). The algorithm converged in 6 iterations. For $N = 8$, the average estimated orientation error for each camera is $\epsilon_o = 0.5°$ and the average position estimation error is $\epsilon_p = 0.03$ units. The algorithm converged in 14 iterations. In a similar configuration, the system proposed in [1] converges to the solution in 65 iterations.

To analyze the expected estimation error depending on the available trajectory data, Figure 3 shows the performance of the proposed algorithm as a function of the number of observations for a network with $N = 8$ sensors. In this case the total trajectory points $T_p = 10000$ and the observed trajectory points are $T_o = \{563, 1126, 2252, 4504\}$. It is possible to notice that when the observed points are increased from

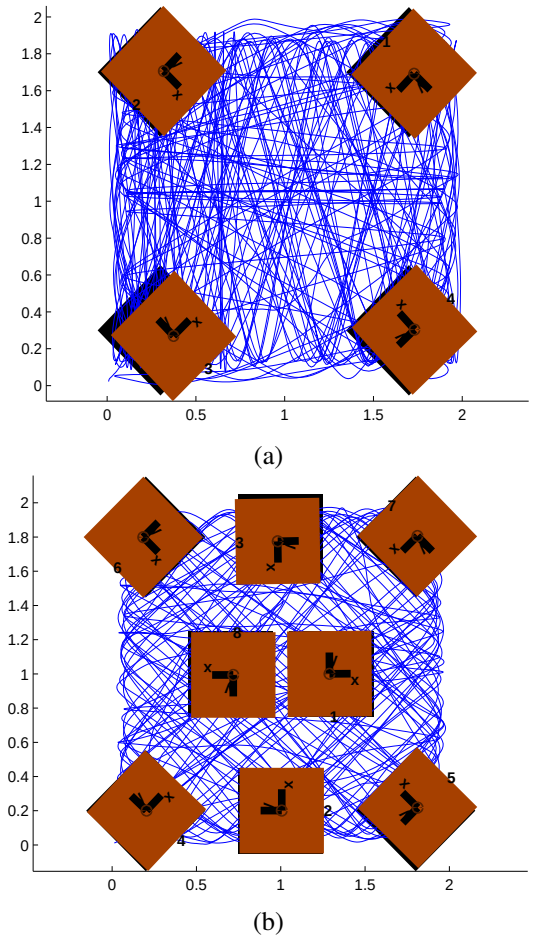

Fig. 2. *Networks of (a) $N = 4$ and (b) $N = 8$ non-overlapping sensors, with $T_p = 10000$ synthetic trajectory data points. The fields of view of the cameras are shown as squares. The estimated sensor configuration (brown) is shown on top of the ground-truth configuration (black)*

N=4			N=8	
ϵ_o	ϵ_p		ϵ_o	ϵ_p
0.94	0.04	C_1	0.49	0.045
1.00	0.02	C_2	1.00	0.003
1.40	0.06	C_3	1.00	0.040
0.10	0.01	C_4	1.40	0.002
-	-	C_5	0.06	0.010
-	-	C_6	0.06	0.050
-	-	C_7	0.04	0.100
-	-	C_8	0.07	0.005

Table 1. *Average orientation error (ϵ_o) and average position error (ϵ_p) over 10 datasets for a network of $N = 4$ and $N = 8$ cameras, with $T_p = 10000$*

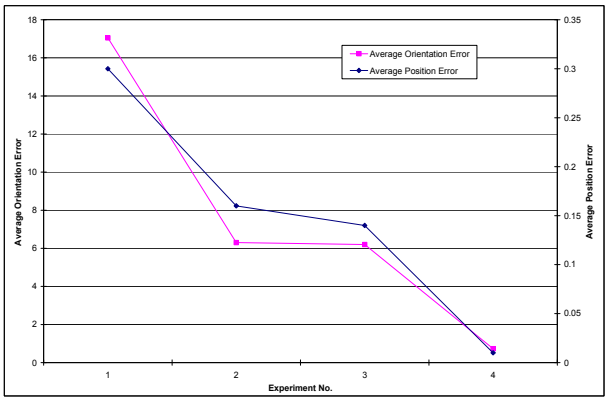

Fig. 3. *Estimation error as a function of the number of observations for a network of $N = 8$ sensors, with $T_p = 10000$ and a varying number of observed trajectory points. Key: Experiment 1: $T_o = 563$; Experiment 2: $T_o = 1126$; Experiment 3: $T_o = 2252$; Experiment 4: $T_o = 4504$*

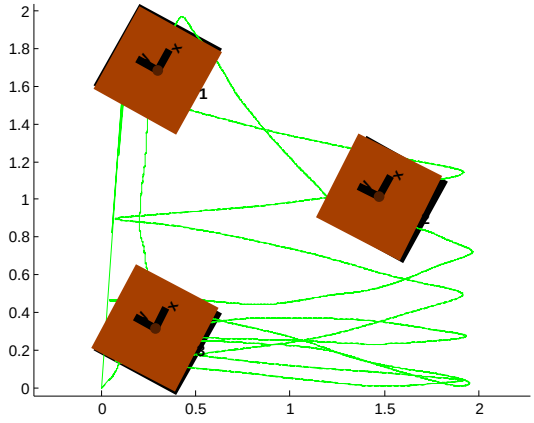

Fig. 4. *Network of $N = 3$ cameras with real trajectory data. The estimated sensor configuration (brown) is shown on top of the ground-truth configuration (black). The estimated configuration on the top of ground truth to visualize the estimation error (black areas)*

$T_o = 563$ to $T_o = 4504$ the average orientation error is reduced from $\epsilon_o = 17.05°$ to $\epsilon_o = 0.73°$. Similarly, the position error is reduced from $\epsilon_p = 15$ to $\epsilon_p = 0.5$ units.

To test the algorithm with real data, we used a network of $N = 3$ cameras, without changing the parameters used for the synthetic data experiments. The distance d between the cameras was $d(C_1, C_2) = 110$ cm, $d(C_1, C_3) = 127$ cm, and $d(C_2, C_3) = 116$ cm. The observed track data T_o were generated from a ball moving across the cameras. Figure 4 shows the estimated network configuration (brown) on top of the hand-made ground-truth camera position (black). The average orientation error is $\epsilon_o = 1°$ and the average position error is $\epsilon_p = 2.5$ cm. Raw data and additional results are available at the following URL:
http://www.elec.qmul.ac.uk/staffinfo/andrea/multi-sensor.html

5. CONCLUSIONS

We proposed an algorithm for the recovery of the missing trajectory points to estimate the relative position between cameras with non-overlapping fields of view. We used Kalman filtering and linear regression to model the trajectories in unobserved regions. Forward and backward estimations are used to increase the reliability of the results. The performance of the algorithm was demonstrated on a set of real and synthetic data in networks of $N = 3$, $N = 4$ and $N = 8$ cameras. The results showed that for a network of $N = 4$ non-overlapping sensors the average location estimation error is approximately 1% of the size of the scene and this error is further reduced to 0.75% for a $N = 8$ sensors network.

Future work includes the use of the homography transformation [6] on the trajectory data to relax the assumption on the sensor positioning (vertical top-down view of the scene) and the testing of the algorithm using trajectory data generated with audio-visual sensors.

6. ACKNOWLEDGEMENTS

We would like to thank Ali Rahimi for providing us with the visualization and the synthetic data generation software.

7. REFERENCES

[1] A. Rahimi, B. Dunagan, and T. Darrell, "Simultaneous calibration and tracking with a network of non-overlapping sensors," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, June 2004.

[2] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *Int. Conf. on Computer Vision*, Nice, France, 2003.

[3] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue, "Simultaneous localization, calibration, and tracking in an ad hoc sensor network," in *Proceedings of the fifth Int. Conf. on Information processing in sensor networks*, New York, NY, USA, 2006, pp. 27–33.

[4] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, June 1999, vol. 2, pp. –259.

[5] T. S. Shen, J. Huang, and C. H. Menq, "Multiple-sensor integration for rapid and high-precision coordinate metrology," in *IEEE/ASME Transactions on Mechatronics*, June 2000, vol. 5, pp. 110–121.

[6] K. Kanatani, N. Ohta, and Y. Kanazawa, "Optimal homography computation with a reliability measure," in *IEICE Transactions on Information and Systems*, June 2000, pp. 1369–1374.