Coupled State-Space Models for Monitoring Nursing Home Resident Activity

Yuan Shi
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
yuanshi@cmu.edu

Caikou Chen
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
cck.yzu@gmail.com

Fernando De la Torre
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA
ftorre@cs.cmu.edu

Howard Wactlar
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
wactlar@cmu.edu

***Abstract*—Emerging new behavioral monitoring and tracking technologies that use camera networks offer unprecedented capabilities for health-care monitoring. A challenging problem is to track people through very sparse sensor measurements to reduce the cost of expensive sensors and be robust to sensor failure. In this paper we propose a Coupled State Space Model (CSSM) to track people across camera networks using a very sparse set of measurements. CSSM simultaneously models the geometry of the camera network as well as the dynamics of the resident being monitored. We apply CSSM to the problem of tracking elderly people in a nursing home setting. Experiments on synthetic and real data show that CSSM can predict the states of cameras and reconstruct the trajectory of the walking person, using very sparse labeling information.**

I. Introduction

Intelligent video monitoring has become a growing need for elderly health-care, due to the shortage of geriatric care professionals, the growth of the elderly population, as well as the societal benefits of improving quality-of-life (QoL) and quality-of-care (QoC) in skilled nursing facilities [1]. Through camera networks, a continuous, voluminous video record can be captured and used to analyze behaviors of the elder. Since many aspects of behavioral disturbances are visually measurable [2], automatic video analysis becomes a crucial application in nursing home where the medical record documentation is often poor and the medical staff is sometimes undertrained or overburdened [3]. With the help of intelligent monitoring technologies, the geriatrician can assess clinical situations more accurately, comprehensively and objectively [4].

The *CareMedia* project [5] developed a test bed in a dementia unit of a community nursing home in suburban Pittsburgh, PA, USA. It built a 23-camera network which covers the single main hallway, the dinning and living rooms, as shown in Figure 1, in which the cameras seeing the person are illustrated by red circles.

Given this gigantic collection of video, an important problem to solve is the detection, identification and tracking of people through the 23-camera network. Computer vision techniques have been developed for this purpose; however, it still remains unclear how to track and identify people across days due to the low quality video, low resolution and strong view point changes. On the other hand, the video data has been partially manually labeled with information about the identity of the subjects and the camera location. The labels provide a sparse set of measures that in many cases is sufficient to reconstruct the smooth trajectory of the people across the camera network and constraint the video tracking. Figure 1 illustrates the problem. One subject labeled at time t has been seen by several cameras (A, B), and at time $t+20$ seen by cameras (C, D, E), and the problem that we will address in this paper is to predict the camera state at each time frame. In other words, we are interested in figuring out in which cameras (marked by green rectangles in Figure 1) the person has appeared and when he/she appeared.

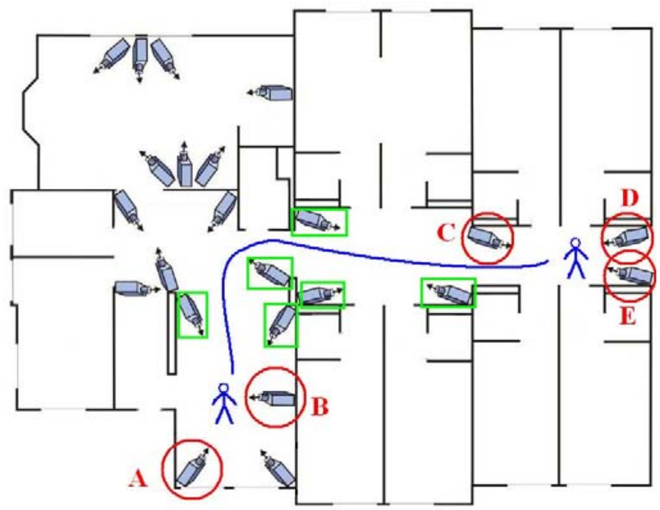

Figure 1. The camera network for dementia unit monitoring (red circles illustrate those cameras capturing the person of interest)

A simple way to solve the above problem is by interpolating the labeled camera states to get the unlabeled ones. However, the interpolation method captures neither the dynamics of the movement of people nor the geometric structure of the camera network. Instead, we propose a Coupled State-Space Model (CSSM) that couples the inference on the *state* of the camera and the *state* of the person, and jointly optimizes all unlabeled states based on labeled ones. CSSM meets two requirements: First, the change of camera states should be smooth with respect to the time and geometry of the camera network. Second, the camera states should be consistent with the trajectory of the walking person. The second requirement motivates us to introduce the state of the person into our model.

The rest of the paper is structured as follows: Section II reviews the related work. Section III presents the problem statement. Section IV describes the proposed method. Section V reports the experimental results and Section VI finalizes the paper with the conclusions.

II. Previous Work

Tracking people through multiple cameras has received considerable attention due to its importance in many applications. Cai and Aggarwal [6] developed a tracking system using a combination of motion analysis on 3D geometry in different camera coordinates. The system starts with tracking from a single camera view and switches to another camera when it predicts that the active camera will no longer have a good view of the subject of interest. Harville [7] described a new combination of plan-view statistics that better presents the shape of tracked objects and proposed a stereo person tracking technique using adaptive plan-view statistical templates and Kalman prediction. Mittal and Davis [8] described a system that is capable of segmenting, detecting and tracking people in a cluttered scene. The system detects and tracks objects by using evidence collected from many pairs of cameras. Rahimi [9] studied a tracking scenario where the views of multiple cameras are non-overlapping. They showed that if information about the dynamics of the target is available, tracking people using non-overlapping cameras is feasible. Song and Chowdhury [10] presented a stochastic and adaptive method for tracking multiple people in a large camera network. The developed method can adapt the similarities between features at different cameras and find the stochastically optimal path for each person.

Note that most of the previous work requires that the location of the target person can be determined within the view of a camera. In this paper, however, we only know whether the person appears in the view of a camera, since we do not access any image-based information. Our problem setting is also related to tracking in wireless sensor networks where often the connectivity information is used to locate the target person [11] [12] [13]. Yet most work in sensor networks assumes that the connectivity information is available at each time frame, which is a significant difference to our problem. Besides, though state-space model was also used in the above literatures, it only involved the dynamics of the person states.

III. Problem Statement

Let k be the number of cameras. Each camera is associated with a tuple (x_i^c, y_i^c, r_i^c), indicating that the view of the i^{th} camera is modeled as a circle with its center (x_i^c, y_i^c) and radius r_i^c. Let n be the number of time frames. At each time t (i.e. the t^{th} frame), a vector $\mathbf{c}_t = [c_{t1}, c_{t2}, ..., c_{tk}]^T$, namely camera state, denotes the state of all cameras. If the i^{th} camera captures the person of interest at time t, then c_{ti} is labeled as 1, otherwise $c_{ti} = 0$. In this setting, we consider tracking only one person. When there exist multiple people, the tracking can be conducted for each person separately.

Given a few labeled camera states $\{\mathbf{c}_{l1}, \mathbf{c}_{l2}, ..., \mathbf{c}_{ln_l}\}$, the goal is to predict the labels of the remaining camera states $\{\mathbf{c}_{u1}, \mathbf{c}_{u2}, ..., \mathbf{c}_{un_u}\}$. Here $n_l + n_u = n$.

IV. Coupled State-Space Model

The proposed method (CSSM) is based on two assumptions:

1. Camera states $\{\mathbf{c}_t\}$ should be smooth with respect to the time and geometry of the camera network.
2. $\{\mathbf{c}_t\}$ should be consistent with the trajectory of the person.

The first assumption is easy to formulate, and the second assumption employs problem-specific knowledge. The main difficulty is that the specific location of the person is unknown. Therefore, we introduce the positions and velocities of the person $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ to our model. Here $\mathbf{x}_t = [a_t, \dot{a}_t, b_t, \dot{b}_t]^T$ denotes the person state, where a_t and b_t denote the x and y coordinates of the person, whereas $\dot{a}_t$ and $\dot{b}_t$ represent the instantaneous velocity of the person, respectively.

The CSSM loss function is defined as follows:

$$\begin{aligned} \mathcal{L}(\mathbf{X}, \mathbf{C}) &= \sum_{t=2}^{n} \|\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1}\|_2^2 \\ &+ \alpha \sum_{t=m+1}^{n} \|\mathbf{c}_t - \mathbf{S}\mathbf{c}_{t-m}\|_2^2 + \beta \sum_{t=1}^{n} \|\mathbf{B}\mathbf{x}_t - \frac{\mathbf{M}\mathbf{c}_t}{\mathbf{1}^T\mathbf{c}_t}\|_2^2 \end{aligned} \tag{1}$$

The first term, $\sum_{t=2}^{n} \|\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1}\|_2^2$, assumes that the person state $\mathbf{x}_t$ evolves according to linear Gaussian Markov dynamics:

$$\mathbf{x_{t+1}} = \mathbf{A}\mathbf{x}_t + \nu_t \tag{2}$$

where $\mathbf{A}$ is a 4×4 matrix representing the dynamics of the person, and ν_t represents isotropic Gaussian noise. In this paper, we set $\mathbf{A}$ as:

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

The second term in (1), $\sum_{t=m+1}^{n} \|\mathbf{c}_t - \mathbf{S}\mathbf{c}_{t-m}\|_2^2$, enforces the smoothness of the camera states, where $\mathbf{S}$ is a $k \times k$ state-transition matrix which captures the geometry of the camera network (k is the number of cameras). The basic assumption is that the state of each camera is propagating to its neighboring cameras along the time. In particular, we assume the state can be propagated from one camera to its neighbors within m frames. The value of m can be fixed or estimated from the person states. The state propagation presented below is similar to the label propagation in semi-supervised learning [14]. The major difference is that in semi-supervised learning, the propagation repeats in infinite time while in our model, the propagation is conducted in a fixed period. We start by defining a neighborhood graph where the nodes represent cameras. An edge is added between two nodes i and j, if the corresponding cameras are regarded as neighbors. The neighborhood relationship is determined according to the geometry of the camera network. The weight of an edge (i, j) is larger if the Euclidean distance between node i and j is smaller. In this paper, we set the weight according to:

$$w_{ij} = \begin{cases} \frac{1}{\sqrt{(x_i^c - y_j^c)^2 + (y_i^c - y_j^c)^2}} & \text{if edge } (i, j) \text{ exists} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

We let the state of a node propagate to all neighboring nodes through edges and larger edges allow easier propagation. The matrix $\mathbf{S} = [s_{ij}]$ defines the probability of state transition:

$$s_{ij} = P(j \to i) = \frac{w_{ij}}{\sum_{q=1}^{k} w_{qj}} \quad (4)$$

where s_{ij} is the probability for a state to propagate from node j to i.

The third term in (1), $\sum_{t=1}^{n} \|\mathbf{B}\mathbf{x}_t - \frac{\mathbf{M}\mathbf{c}_t}{\mathbf{1}^T\mathbf{c}_t}\|_2^2$, relates the person states with the camera states. This term says that at each time, the position of the person should be close to the center of those cameras capturing that person. Here $\mathbf{B}$ is a matrix for extracting the coordinates of the person, and $\mathbf{M}$ is a matrix containing the coordinates of all cameras. α and β are two parameters balancing the weight of different terms. Specifically, we set the above matrixes to:

$$\mathbf{B}^T = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \ \mathbf{M} = \begin{bmatrix} x_1^c & y_1^c \\ x_2^c & y_2^c \\ \vdots & \vdots \\ x_k^c & y_k^c \end{bmatrix}$$

Directly optimizing (1) is difficult because $\mathbf{c}_t$ is a binary vector. To make the loss function differentiable and easy to optimize, we instead consider an approximate loss function (5). Note in (5), we introduce a set of new variables $\mathbf{f}_t$ to approximate the normalization of $\mathbf{c}_t$. For any labeled camera state, the corresponding $\mathbf{f}_t$ is set to $\frac{\mathbf{c}_t}{\mathbf{1}^T\mathbf{c}_t}$. Once (5) is optimized, the corresponding $\mathbf{c}_t$ can be easily obtained (we will discuss this shortly).

$$\begin{aligned} \mathcal{L}(\mathbf{X}, \mathbf{F}) = &\sum_{t=2}^{n} \|\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1}\|_2^2 \\ &+ \alpha \sum_{t=m+1}^{n} \|\mathbf{f}_t - \mathbf{S}\mathbf{f}_{t-m}\|_2^2 \\ &+ \beta \sum_{t=1}^{n} \|\mathbf{B}\mathbf{x}_t - \mathbf{M}\mathbf{f}_t\|_2^2 + \gamma \sum_{t=1}^{n} \|\mathbf{1}^T\mathbf{f}_t - 1\|_2^2 \end{aligned} \quad (5)$$

The fourth term in (5), $\sum_{t=1}^{n} \|\mathbf{1}^T\mathbf{f}_t - 1\|_2^2$, enforces that $\mathbf{f}_t$ is a normalization of $\mathbf{c}_t$.

The optimization of (5) is described below. Let $\mathbf{E} = \mathbf{1}\mathbf{1}^T, \mathbf{G} = \mathbf{M}^T\mathbf{M}, \mathbf{H} = \mathbf{M}^T\mathbf{B}$, and $\mathbf{Q} = \mathbf{S}^T\mathbf{S}$, then

$$\frac{1}{2}\frac{\partial \mathcal{L}}{\partial \mathbf{x}_t} = \begin{cases} \beta\mathbf{B}^T\mathbf{B}\mathbf{x}_t - \beta\mathbf{B}^T\mathbf{M}\mathbf{f}_t & t = 1 \\ ((1 + \beta\mathbf{B}^T\mathbf{B})\mathbf{I} + \mathbf{A}^T\mathbf{A})\mathbf{x}_t \\ -\mathbf{A}\mathbf{x}_{t-1} - \mathbf{A}^T\mathbf{x}_{t+1} - \beta\mathbf{B}^T\mathbf{M}\mathbf{f}_t & 1 < t < n \\ (\mathbf{I} + \beta\mathbf{B}^T\mathbf{B})\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1} \\ -\beta\mathbf{B}^T\mathbf{M}\mathbf{f}_t & t = n \end{cases}$$

$$\frac{1}{2}\frac{\partial \mathcal{L}}{\partial \mathbf{f}_t} = \begin{cases} (\beta\mathbf{G} + \gamma\mathbf{E})\mathbf{f}_t - \beta\mathbf{H}\mathbf{x}_t - \gamma\mathbf{1} & t = 1 \\ (\alpha(\mathbf{I} + \mathbf{Q}) + \beta\mathbf{G} + \gamma\mathbf{E})\mathbf{f}_t \\ -\alpha\mathbf{S}\mathbf{f}_{t-m} - \alpha\mathbf{S}^T\mathbf{f}_{t+m} \\ -\beta\mathbf{H}\mathbf{x}_t - \gamma\mathbf{1} & m < t \le n - m \\ (\alpha(\mathbf{I} + \beta\mathbf{G} + \gamma\mathbf{E})\mathbf{f}_t \\ -\alpha\mathbf{S}\mathbf{f}_{t-m} - \beta\mathbf{H}\mathbf{x}_t - \gamma\mathbf{1} & n - m < t \le n \end{cases}$$

By setting $\frac{\partial \mathcal{L}}{\partial \mathbf{x}_t} = \frac{\partial \mathcal{L}}{\partial \mathbf{f}_t} = 0$, we obtain a linear system that can be presented in the following form ($\mathbf{F}_l$ denotes known variables, $\mathbf{V}_u$ denotes unknown variables and $\mathbf{L}$ is the values of $\mathbf{F}_l$):

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{P}_{ul} & \mathbf{P}_{uu} \end{bmatrix} \begin{bmatrix} \mathbf{F}_l \\ \mathbf{V}_u \end{bmatrix} = \begin{bmatrix} \mathbf{L} \\ \mathbf{0} \end{bmatrix} \quad (6)$$

where

$$\mathbf{F}_l = \begin{bmatrix} \mathbf{f}_{l1} \\ \mathbf{f}_{l2} \\ \vdots \\ \mathbf{f}_{ln_l} \end{bmatrix}, \mathbf{V}_u = \begin{bmatrix} \mathbf{F}_u \\ \mathbf{X} \end{bmatrix}$$

$$\mathbf{F}_u = \begin{bmatrix} \mathbf{f}_{u1} \\ \mathbf{f}_{u2} \\ \vdots \\ \mathbf{f}_{un_u} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}$$

It's easy to show

$$\mathbf{V}_u = -\mathbf{P}_{uu}^{-1}\mathbf{P}_{ul}\mathbf{L} \quad (7)$$

Once we obtain $\{\mathbf{x}_t\}$, it is straightforward to calculate $\{\mathbf{c}_t\}$ by:

$$c_{ti} = \begin{cases} 1 & \text{if } (a_t - x_i^c)^2 + (b_t - y_i^c)^2 < (r_i^c)^2 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where $i \in \{1, 2, ..., k\}, t \in \{1, 2, ..., n\}$.

It is interesting to point out that equation (7) has a same structure as the label propagation formulation in graph-based semi-supervised learning [14]. (7) shows that the unlabeled states can be easily inferred from labeled ones. In fact, our model can be regarded as a special case of semi-supervised learning - transductive inference method [15] [16].

V. Experiments

In this section, we test CSSM in both synthetic and real-world data. CSSM is compared with three interpolation methods which treat each camera independently and interpolate the values of labeled camera states using different types of curves. The interpolation methods used are linear interpolation (denoted by Linear), cubic spline interpolation (denoted by Spline) and piecewise cubic Hermite interpolation (denoted by Cubic). CSSM is implemented using the following parameters: $\alpha = 0.5, \beta = 2, \gamma = 0.5, m = 5$.

A. Synthetic Data

We first compare four methods on a synthetic camera network (Figure 2). The camera network consists of 12 cameras and the corresponding neighborhood graph is given in Figure 3. The neighborhood structure is manually determined in our experiments though it can be automatically calculated. We also manually create two synthetic cases (green circles represent the synthetic trajectories) in Figure 3.

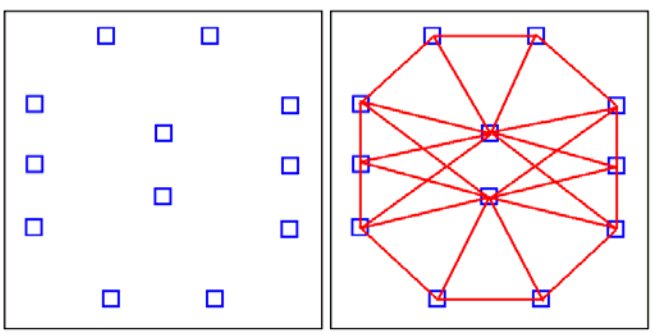

Figure 2. Left: synthetic camera network (blue squares represent cameras) Right: the neighborhood graph of the camera network

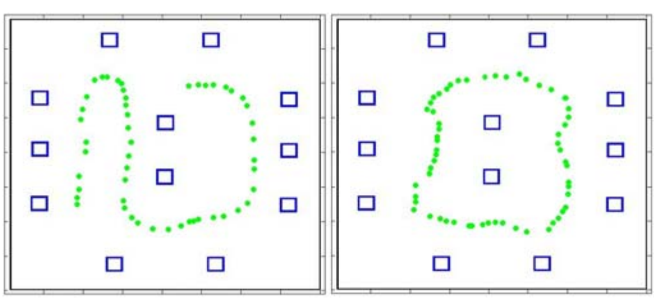

Figure 3. Two synthetic trajectories presented by green circles

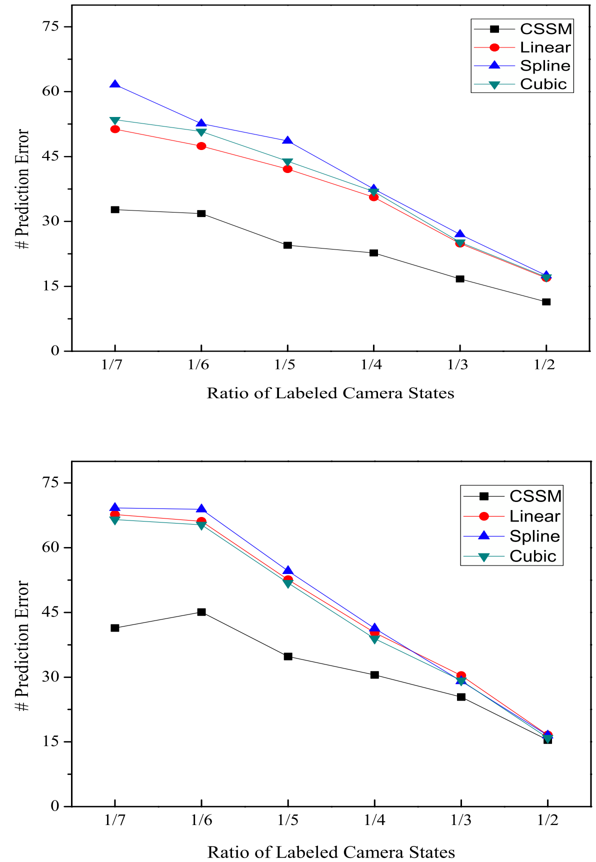

Figure 4. Performance comparisons of four methods

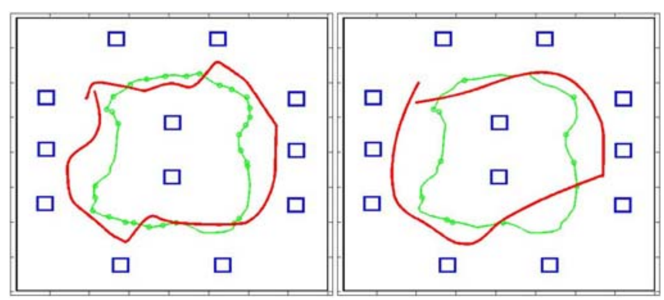

Figure 5. Trajectories estimated by CSSM with 1/2 (left) and 1/6 (right) of labeling ratio (green curve represents the true trajectory, green dots correspond to the labeled frames, and red curve shows the estimated trajectory)

Figure 4 shows the performance comparisons between different methods when the ratio of labeled camera states varies. For each labeling ratio, the experiments are conducted randomly for 10 times and the average performance is recorded. It is easy to see that, CSSM outperforms the other three interpolation methods in each case. When the ratio of labeling is relatively small (1/7), CSSM performs significantly better than interpolation, demonstrating that CSSM better captures the underlying dynamics of the camera states, thus is a promising method to tackle with label sparseness. Also note that when the ratio of labeling becomes larger, all methods perform better, yet CSSM still maintains its advantage. Figure 5 presents the estimated trajectory in the second synthetic case, when the labeling ratio is 1/2 and 1/6. We can see that when the ratio is larger, CSSM is able to estimate the trajectory more accurately.

B. Real Data

The real data is collected from the test bed shown in Figure 1. The corresponding neighborhood graph is presented in Figure 6. In our study, we choose three video sequences in which the person keeps walking. For each video sequence, its frames are further sampled at a time interval of two seconds [1]. The description of the data is given in Table 1. We then generate 12 test cases by varying the labeling ratio. For every test case, we run each algorithm 10 times (since the data is randomly labeled according to the labeling ratio) and measure the average performance. Table 2 lists the experimental results. We can see that CSSM performs best in almost all test cases, compared with the three interpolation methods. Note that when the labeling is sparse (i.e. the ratio is 1/10 or 1/8), CSSM significantly outperforms other methods. When the labeling ratio is relatively high (i.e. 1/4), CSSM still gives comparative accuracy. The estimated trajectories by CSSM in two test cases are visualized in Figure 7.

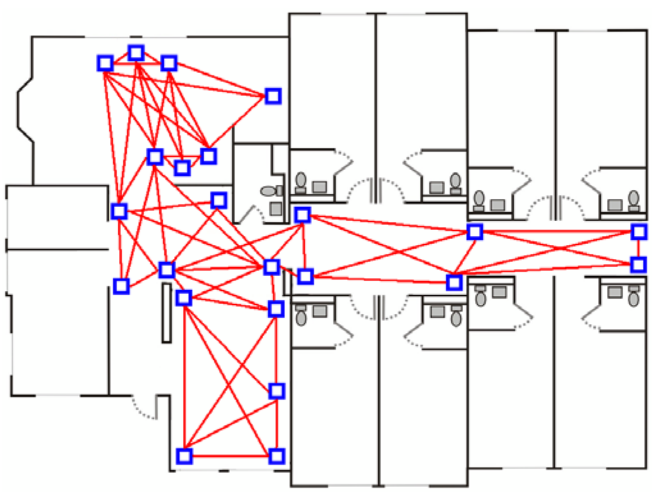

Figure 6. The neighborhood graph of real camera network

Table I
DESCRIPTION OF REAL DATA

Video ID	Duration	#Sampled Frame	#State Variables
1	148 Sec.	74	1702
2	162 Sec.	81	1863
3	136 Sec.	78	1564

[1]The reason for sampling is to reduce the number of variables in the model. If a different sampling ratio is used, one needs to choose a different value for the parameter m.

Table II
EXPERIMENTAL RESULTS ON REAL DATA

Test Case		#Prediction Error			
Video Id	Labeling Ratio	CSSM	Linear	Spline	Cubic
1	1/10	**66.4**	99.2	109.6	100.3
1	1/8	**63.1**	71.1	85.1	74.3
1	1/6	**58.7**	68.9	73.6	69.5
1	1/4	**43.0**	44.8	50.5	45.1
2	1/10	**86.7**	97.0	111.6	98.9
2	1/8	**73.6**	81.9	91.9	84.8
2	1/6	**48.1**	51.3	57.0	52.7
2	1/4	39.4	**39.2**	46.8	40.1
3	1/10	**56.2**	63.2	72.8	64.8
3	1/8	**42.0**	49.4	52.8	49.6
3	1/6	**33.6**	33.9	35.0	34.0
3	1/4	27.3	**25.9**	30.8	26.1

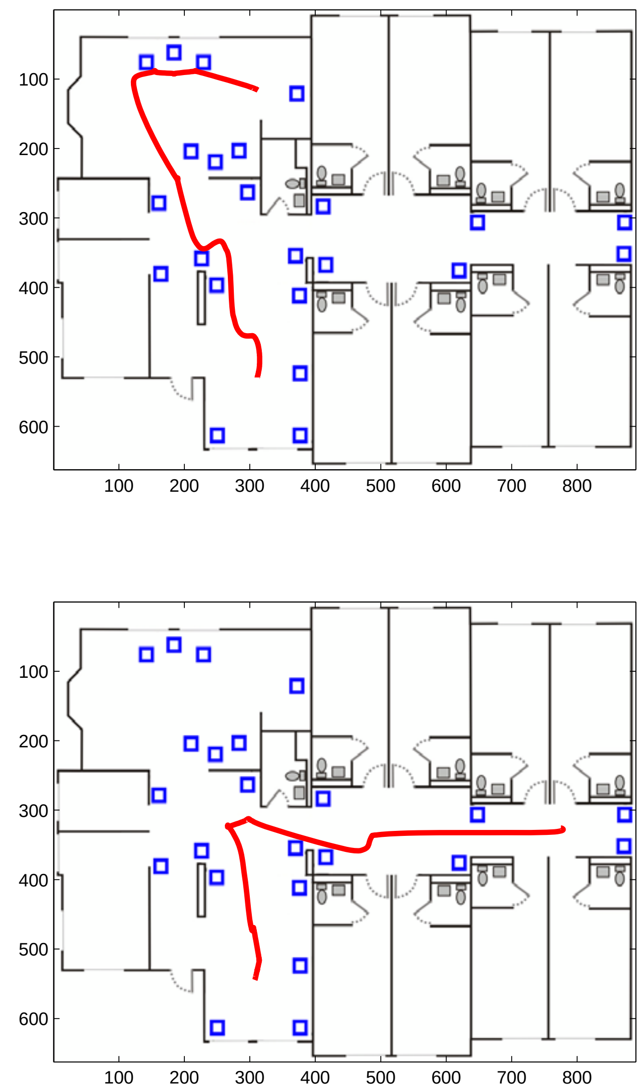

Figure 7. Trajectories estimated by CSSM (Top: on Video 1 with 1/4 labeling ratio, Bottom: on Video 2 with 1/4 labeling ratio)

The outstanding performance of CSSM can be explained in the following way. CSSM connects both camera and person states in a coupled model, thus jointly optimizes the unlabeled states given the labeling information. The joint optimization not only ensures the camera states to be smooth along the time and the geometry of the camera network, but also constrains them using the dynamics of the walking person.

VI. Conclusion

In this paper, we have addressed a problem of predicting camera states when tracking people in camera networks. We have proposed a coupled state space model (CSSM) that simultaneously models the geometry of the camera networks as well as the dynamics of the humans being tracked. CSSM can well predict the camera states and at the same time reconstruct the trajectory using very few labels. Synthetic and real data has shown the validity of our approach.

One interesting aspect of CSSM is that it connects the camera states, the structure of the camera network, and the trajectory of the walking person together. Hence it's promising to extend CSSM to solve more challenging problems such as determining proper frames to label and optimizing the camera placement. We will address these problems in our future work.

References

[1] V. Mor, "Defining and measuring quality outcomes in long-term care," *Journal of the American Medical Directors Association*, vol. 7, no. 8, pp. 532–540, 2006.

[2] A. Bharucha, J. Rosen, B. Mulsant, and B. Pollock, "Assessment of behavioral and psychological symptoms of dementia," *CNS Spectr*, vol. 7, no. 11, pp. 797–802, 2002.

[3] J. Schenelle, B. Bates-Jensen, L. Chu, and S. Simmons, "Accuracy of nursing home medical record information about care-process delivery: implications for staff management and improvement," *Journal of the American Geriatrics Society*, vol. 52, no. 8, pp. 1378–1383, 2004.

[4] D. Chen, H. Wactlar, and A. Bharucha, "People identification through ambient camera networks," *International Conference on Multimedia Ambient Intelligence, Media and Sensing (AIMS)*, 2007.

[5] A. Bharucha, "Caremedia: Automated video and sensor analysis for geriatric care," *Annual Meeting of the American Medical Directors Association (AMDA)*, 2006.

[6] Q. Cai and J. Aggarwal, "Tracking human motion in structured environments using a distributed-camera system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1241–1247, 1999.

[7] M. Harville, "Stereo person tracking with adaptive plan-view statistical templates," *Image and Vision Computing*, vol. 22, pp. 127–142, 2002.

[8] A. Mittal and L. Davis, "M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo," *International Journal of Computer Vision*, pp. 189–203, 2002.

[9] A. Rahimi, B. Dunagan, and T. Darrell, "Tracking people with a sparse network of bearing sensors," *European Conference on Computer Vision (ECCV)*, 2004.

[10] B. Song and A. Chowdhury, "Stochastic adaptive tracking in a camera network," *IEEE 11th International Conference on Computer Vision (ICCV)*, 2007.

[11] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Ordinal mds-based localization for wireless sensor networks," *ACM International Symposium on Mobile Ad-hoc Networking and Computing*, 2003.

[12] V. Vivekanandan and V. Wong, "Ordinal mds-based localization for wireless sensor networks," *IEEE Vehicular Technology Conference (VTC)*, 2006.

[13] J. Cabero, F. D. la Torre, G. Unibaso, and A. Sanchez, "Tracking algorithms based on dynamics of individuals and multidimensional scaling," *IEEE International Symposium on Wireless Pervasive Computing*, 2008.

[14] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," *Technical Report CMU-CALD-02-107, Carnegie Mellon University*, 2002.

[15] V. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[16] X. Zhu, "Semi-supervised learning literature survey," *Computer Sciences TR 1530, University of Wisconsin C Madison*, 2008.