MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Multi Human Trajectory Estimation using Stochastic Sampling and its Application to Meeting Recognition

Yosuke Matsusaka Hideki Asoh Futoshi Asano
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{yosuke.matsusaka, h.asoh, f.asano}@aist.go.jp

Abstract

In this paper we present a stochastic sampling approach to estimate multiple human trajectory in the meeting. The algorithm is formalized as a energy minimization problem based on stochastic sampling of deterministic trajectory, and has some effectiveness to the low frame data with jumps and switchings and it can estimate a near optimal result in 9 times faster then the real-time by using Gibbs sampling. Also experiment is shown using meeting data of real environment.

1 Introduction

We have been developing a system called VTMOffice, which is an application system for business user. The system records business meetings in office environment in both visual and auditory ways, and automatically generates an archive of human activity by means of visual and auditory recognition methods. The system is aimed to be a meeting archival system with automatic recognizer which generates searchable indexes.

Precise tracking of humans is essential for this system. Trajectory information can be used to create an archive of human activity (e.g. recognize who is setting at where, detect change of scenes, count number of members in the meeting, etc...) and also it could be used to help improve auditory signal processing, because in such algorithms (e.g. beam forming) requires precise positions of the speakers to get high quality results. This also make easy the following auditory process such as speech recognition.

In this paper, we propose a multi human trajectory estimation algorithm which has efficiency for low frame rate data with many noises. In order to accomplish efficiency in above conditions, we have developed a trajectory estimation algorithm based on stochastic sampling. From following sections, the proposed algorithm is described.

2 Related Works

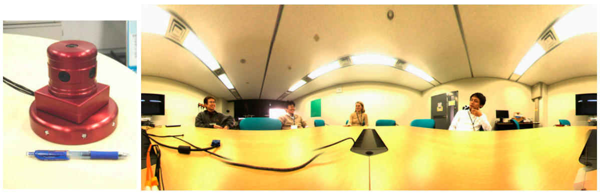
In recent years, there are eager researches on developing algorithms for recognizing meeting scenes. Smith etal. [1] has introduced particle filter based tracking and Kato etal. [2] has introduced dynamic Bayesian network based tracking. For the estimation of multiple trajectory of humans, some of the related work has also been done in sports scene tracking (e.g. [4]). In most of these algorithms, it estimates human positions frame by frame, and then trajectories are calculated as a result of those frame based estimations.

Different from these approaches, our method tries to improve trajectory estimation by stochastic sampling of the trajectory itself. Works related to this approach, Unal etal. [5] has shown a efficient sampling strategy based on the curvature of the trajectory to fit elastic contour model for image segmentation. And also there are numbers of research which have applied snakes energy minimization for open ended contours which could be used to trajectory estimation. Taking part of these research contexts, we try to apply these stochastic sampling based trajectory estimation technique to multi human trajectory estimation problem. Detail of the approach is described in Section 4.

3 The Application

Before get into the main discussion about the algorithm, we first introduce our application system and discuss about the problems occur in such applications to make clear the problems we are facing with.

Figure 1 shows the overview of our system and the image captured. The system is designed as a compact set of sensors which records 360 degree meeting scene. It is placed on the center of the table so that it can capture the frontal face of all the participants. It works with a processor which stores and processes the recorded data.



a)The system b)Example of a image captured from the camera

Figure 1: System overview and the captured image.

It has 6 cameras which covers 360 degree views and 8 microphones which captures audio signals [1]. Resolution

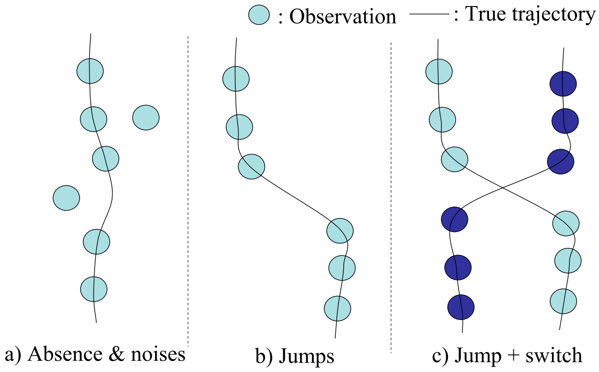
[1] We do not describe about audio signal processing part in this pa-

of the camera is 1024x846 in each and the frame rate is 4.2 fps in average.

3.1 Problems

Compact panoramic hi-resolution cameras used in such systems usually captures images in low frame rates. Also in normal office environments, we cannot assume backgrounds or lightings are uniform. In such situations, there exists some amount of miss detections generated at the pattern analysis steps. These conditions creates some problems in trajectory estimation.

Figure 2 illustrates some of the major problems need to be considered in such conditions.



a) Absence & noises

b) Jumps

c) Jump + switch

Figure 2: Major problems.

As shown in Figure 2-a), there is some noises and absence in observation data. These data can cause the trajectory estimator to be lured away from the correct track.

During the meetings, humans are mostly sitting on the seat, so the observation data is motionless in most of the period, but they suddenly moves when people changes their seat at some timing. Because the frame rate is not high, it is sometimes observed as a jump as shown in Figure b). The trajectory estimator is required to catch up to those sudden changes.

Because the application is for the business meetings which is consisted by multiple humans, we have to consider multiple trajectories. As shown in Figure c), estimation of multiple trajectory becomes more difficult, especially when people switch positions each other.

Since most of the previous frame based algorithms use time differential information such as optical flows, especially in case when there are jumps in the data, it is less likely to get accurate result if we use those algorithms straightforward.

Given these problems, our proposed algorithm is described from next section.

4 Proposed Algorithm

In most of the conventional frame based trajectory estimation algorithms, those are carried out as follows,

1. Detect position of humans from each frames using pattern detector, and extract identification features (e.g. patterns, colors, etc...) from those detected regions.

2. Based on the position information of the previous frames and the similarity of identification features extracted in 1, estimate the most likely connection of positions from previous to current.

These algorithms try to use previous frame estimations to improve the current estimation. Taking estimation of each frames as a "state", it could be said that the algorithm tries to improve its estimate by modeling transition between those states. The state transition model can be either in deterministic or probabilistic form.

Different from these algorithms, we try to model whole the trajectory as a state and try to improve the trajectory using stochastic sampling. The algorithm is abstracted as follows,

1. Detect position of humans from each frames using pattern detector, and extract identification features from those detected regions.

2. Lay initial trajectory estimation on the positions estimated in 1. At this point there are still many mistakes on the trajectory.

3. Refine trajectory estimation by iterations of stochastic trajectory generations and selections.

Sizable difference between these two approach is in the latter case, we use whole the observation from the very beginning. Note due to this reason, the current algorithm cannot carried out on-line.

The formalization we present here consist of deterministic and probabilistic part. We first present deterministic part.

The input vectors is the set of time t, position p, and identification features f.

$$x_i = < t_i, p_i, f_i > (i = 1 \ldots n) \qquad (1)$$

Here, we model the trajectory using per frame parameters $\mathrm{o} = o_1 \ldots o_t$.

Given this trajectory parameter o, we define trajectory energy as follows.

$$\begin{aligned} E_{\text{total}}(\mathrm{o}, x) &= E_{\text{internal}} + E_{\text{external}} & (2) \\ &= \sum_{i=0}^{t-1} (o_{i+1} - o_i)^2 + \alpha \sum_{i=0}^{n} (p_i - o_{t_i})^2 & (3) \end{aligned}$$

For estimating multiple trajectories, we add clustering steps to above equation,

$$E_{\text{system}}(\mathrm{o}, x) = \sum_c E_{\text{total}}(\mathrm{o}_c, x)|_{x \in X_c} + \beta E_{\text{noise}}(x)|_{x \in X_e} \quad (4)$$

We denote X_c is a membership function and E_{noise} is a penalty of background noises defined as $E_{\text{noise}}(x) =$

per. Those are described in [3]

n/N. Given fixed parameter β, the algorithm is able to estimate optimal number of the trajectories while preserving some robustness to the background noises.

In addition to above deterministic equations, we assume observations follow following probabilistic distribution $X_n \sim Categorical(P)$, $p_{X_n} \sim Normal(o_n, \tau_p)$, $p_{X_e} \sim Uniform(\tau_e)$, $f_i \sim Normal(\lambda_{X_n}, \tau_f)$.

From above equations, the algorithm is now defined as a stochastic system which has a system energy defined by deterministic trajectory model and is driven by some prior probabilistic distributions calculated from observations.

Above description still does not gives concrete process of estimating optimal parameters. Detailed implementation of the algorithm is described in next section.

5 Implementation of the Algorithm

As we can easily imagine, finding optimal parameters for the equations is computationally too expensive. In order to find near optimal parameters in limited computational cost, we use Gibbs sampling with following sampling strategy.

Initialization step:

Create clusters of identification feature using GMM model $X_n \sim Categorical(P)$, $f_i \sim Normal(\lambda_{X_n}, \tau_f)$. GMM based clustering is a well used framework which feature cluster is created as each Gaussian distribution using EM iterations. Then based on these clusters, we create initial trajectory as.

$$o_{ci} = p_j \text{ where } j = \text{argmax}_j E(N(\lambda_j, \tau_f), f_i) \tag{5}$$

Convergent step:

In this step, Gibbs sampling is applied. In order to realize quick convergence, we use following local energy based sampling strategy to chose a parameter.

$$i \sim \frac{\sum_{j=i-2}^{i+2}(o_{j+1} - o_j)^2}{\sum_i \sum_{j=i-2}^{i+2}(o_{j+1} - o_j)^2} (i \in 3, \ldots, t-3) \tag{6}$$

and update the parameter as $\hat{o}_{ni} \sim Normal(o_{n(i+1)}, \tau_p)$.

Annealing:

After the convergent step, the system energy is calculated, and then, the updated trajectory is either selected or rejected using following simulated annealing criteria.

$$A = \frac{\exp(-E_{\text{system}}(\hat{o}, x))}{\exp(-E_{\text{system}}(o, x))} \tag{7}$$

Select $\hat{o}$ as a trajectory for the next iterations when $A > 1$.

End step:

Above iteration is stopped after the energy went under certain level or the iteration has reached to certain numbers.

Trajectory number estimation:

For estimating optimal trajectory numbers, here, we simply increase the number after each annealing loops while the final energy decrease continuously

Figure 3 illustrates the process described above. The energy function and the convergence process introduced here resembles to those of snakes models[6]. But in this case we use stochastic sampling. So the trajectory sample can easily make jumps based on prior distribution of the identification features, and thus expected to get up with the sudden changes occurred in the data.

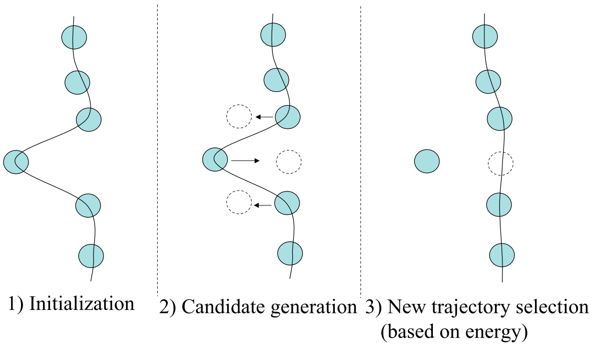


Figure 3: The stochastic relaxation framework.

6 Experiment

6.1 Experiment Condition

The data used in this experiment are 3 set of casual meetings each consisted between 2, 3, 4 participants. Participants are asked to have free talks and asked to change their seat at some specified timings. During the meeting they are allowed to move their faces and bodies freely (thus there happens some occlusion of the faces hided by their hands and bodies). Their clothes and the room backgrounds are not controlled. See Figure 1-b) for the example image extracted from the test data.

For evaluation, we not only evaluate results of our proposed algorithm, but also compare with conventional frame based algorithm to make clear the characteristics of the proposed. For this experiment, a particle filter with following simple formula is used. $p_{X_n} \sim Normal(o_n, \tau_p)$, $o_{x+1} \sim Normal(o_x, \tau_o)$, $f_i \sim Normal(\lambda_{X_n}, \tau_f)$.

6.2 Image Feature Extraction

The images captured from each 6 cameras are undistorted using pre-calibrated camera parameters and stitched into single panoramic image. This process produce 360 degree 5400 x 2250 pixel panoramic image as shown in Figure 1.

Then we extract feature to determine human positions. This process consist of two steps, first we conduct face detection and then we extract identification feature. For the face detection, we use Viola Jones [7] algorithm. For the identification feature extraction, because in our application, the resolution of the panoramic image is not as high to get detailed texture of the face, we use color information instead. Pixels under the face region is extracted as a color information of the clothes that participants are wearing and encoded as a feature $f =< r, g, b >$.

6.3 Result

Figure 4 shows trajectory of each test data estimated with our algorithm. In the figure, left is the true trajectory, center is the result of initial trajectory estimate and the right is the final trajectory estimation respectively. As we can see from the figure, each trajectories are estimated precisely without having confused by jumpings or switching.

In the figure, we also shows RMS errors between the estimated and true trajectory. It also shows low values for any numbers of trajectory. To show the difficulty of applying frame based algorithm to this kind of task, we also shows the RMS error of simple particle filter for reference (shown as "PF-RMS" in the figure). The result shows higher errors mainly caused by the confusions occurred at the jumps.

As we can see from the figure of initial trajectory, missdetection of the face makes some amount of spike like noises. Those are also reduced in final estimation.

For the processing time, it took about 2.23 minutes on Penitum4 3.8GHz machine to fully converge 20 minutes meeting data. Although this is about 9 times faster than the realtime, we are planning to make it faster by improving the sampling strategy.

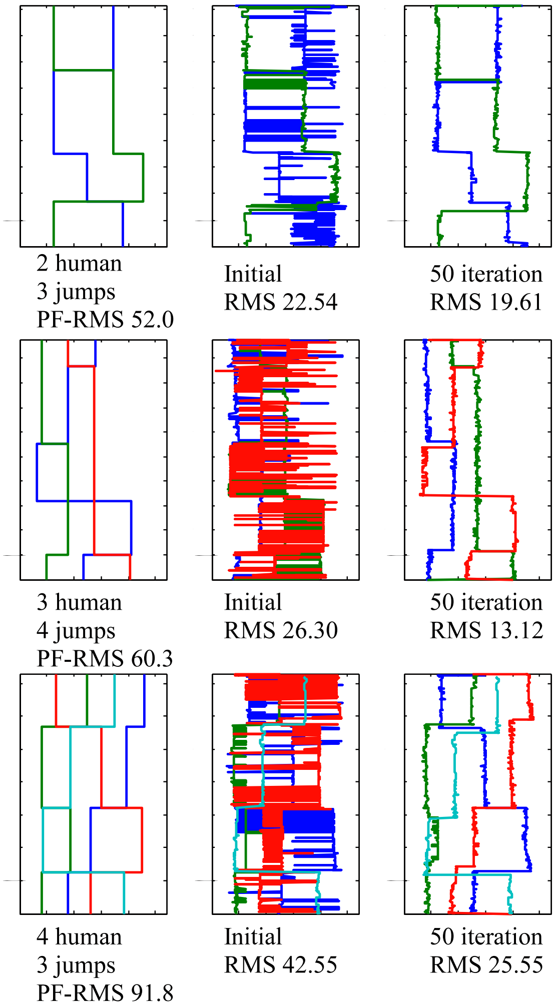


Figure 4: Result of the experiment. Left is the true trajectory, center is the result of initial trajectory estimate and the right is the final trajectory estimate respectively. Rows are the respective data sets.

7 Discussion & Conclusion

In this paper we have presented a stochastic sampling approach to estimate multiple human trajectory in the meeting. The algorithm is formalized as stochastic sampling of deterministicly defined trajectory, and has some effectiveness to the low frame data which has jumps and switchings.

The advantage of using such algorithm is flexible modeling of the trajectory. Since in this algorithm trajectory is connected from the initial to final time frame from the beginning, we can jump time frame back and forth without care. And also it is easy to model complex trajectory, because the trajectory is always evaluated in its whole shape. Time based algorithms are generally difficult to estimate such complex trajectories, because in such case it is required to treat long contextual information in order to make next predictions.

One of the disadvantage of this approach is the algorithm is not on-line and also have to care about the sampling strategy to realize the rapidness of the convergence. For this problem we have introduced a Gibbs sampling strategy with a local energy based sampling strategy and realized a speed 9 times faster then the realtime, but this part may be improved more in the future work.

References

[1] K. Smith, D. Gatica-Perez, and J.-M. Odobez "Using Particles to Track Varying Numbers of Interacting People," in Proc. CVPR, vol.1, pp.962-969, 2005.

[2] M. Kato etal., "State estimation of meetings by information fusion using Bayesian network," In Proc. of Interspeech2005, pp.113-116, 2005.

[3] F. Asano and J. Ogata, "Detection and separation of speech events in meeting recordings," In Proc. of Interspeech2006, 2006.

[4] K. Okuma etal., "A boosted particle filter: Multitarget detection and tracking," European Conference on Computer Vision, pp.28-39, 2004.

[5] G. Unal etal., "Algorithms for stochastic approximations of curvature flows," Proc. of International Conference on Image Processing, vol.2, pp.651-654, 2003.

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes - Active Contour Models," International Journal of Computer Vision, vol.1(4), pp.321-331, 1987.

[7] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Proc. of CVPR, vol.1, pp.511-518, 2001.