10—1

3-Dimensional Tracking of Multiple Object Motions from Multi-view Images by Mixed-state CONDENSATION Algorithm

Koichiro Deguchi* Koji Hamasaki Taira Nakajima
Takayuki Okatani
Graduate School of Information Sciences, Tohoku University

Abstract

In this paper, we propose an efficient method for tracking multiple objects, which deals with occlusion situations in an image. We use Shape from silhouettes method to reconstruct objects 3D position from images taken from different directions. Then we track the objects' reconstructed 3D positions by using mixed-state CONDENSATION algorithm. The experimental results show the robust tracking of multiple objects.

1 Introduction

In this paper, we treat the situation shown as Fig.1, where multiple cameras observe a scene and track moving multiple objects in the scene. We achieve robust identification of multiple moving objects by fusing multiple images, and track their 3D positions by using CONDENSATION based algorithm[2].

We take multiple camera images from various directions, and reconstruct the objects' 3D-shape, then track them. To track moving objects,first, we must extract some feature of the objects from the image. In a case of tracking multiple objects, the probability of the positions of the extracted features usually have non-Gaussian distribution, so the class of techniques employing the Kalman filter is not available to track their motions.

Our method use the CONDENSATION algorithm combined with a mixed-state model which is extended from the conventional CONDENSATION algorithm developed for visual tracking in clutter. The mixed-state CONDENSATION algorithm has some prediction models which correspond to the dynamics of moving object. By using this algorithm, we track moving object which have several possible states (for example, bouncing ball has two state, constant acceleration state and bounce state). To deal with the occlusions in an image, we extended 2D mixed-state CONDENSATION algorithm to 3D, and realized the robust tracking coping with occlusions.

2 Reconstruction of object's 3D-shape

First, we reconstruct the object's 3D-shape, then recover their 3D positions. We employ the shape from silhouettes method (SS-method) which reconstructs 3D-shape from silhouettes in multiple images[4]. We obtain the silhouettes' images by the background subtraction techniques. Then we reconstruct 3D-shape as the product of cones which are the back projection of each cameras' silhouette images.

*Address: Aramaki-aza Aoba01, Aoba-ku, Sendai 980-8579, Japan. E-mail: kodeh@fractal.is.tohoku.ac.jp

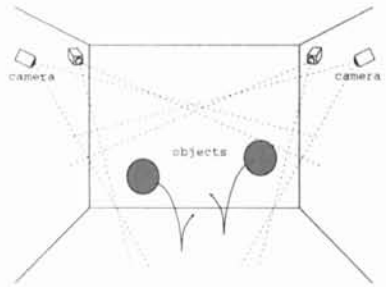


Figure 1: Understanding 3D dynamic environment

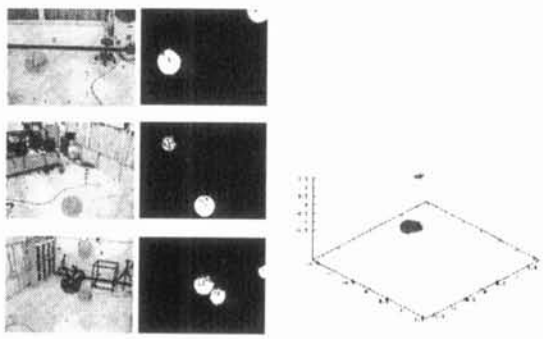


Figure 2: Background subtraction and 3D-shape from silhouettes of balls. Left: 3 view images including moving balls. Middle: Background subtraction images. Right: Reconstructed 3D positions of balls.

3 Mixed-state CONDENSATION algorithm

The CONDENSATION algorithm is a time sequence filter for tracking objects. We consider the object position with a probability, that is, at every time step, every place has some probability with which the object exists at the place. In this method, we simulate the object motion emploting large number of samples, each of which has different random component. The filter's output at each time step is an approximation of a profile of the probability distribution of objects' positions and represented as a weighted sample set $\{s_t^{(n)}, n = 1, ..., N\}$ with weights $\pi_t^{(n)}$, where t is the time and N is the number of the samples.

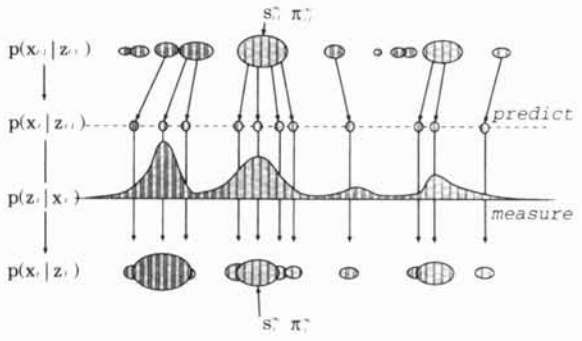


Figure 3: Probability transition at one timestep in the CONDENSATION algorithm.

Fig.3 shows the iterative prediction applied to the sample-set. The top of the figure shows the weighted sample-set $\{(\boldsymbol{s}_{t-1}^{(n)}, \pi_{t-1}^{(n)}), n = 1, ..., N\}$ which is the output at the time $t-1$. The first operation is to choose j^{th} sample $\boldsymbol{s}_t'^{(j)}$ from previous sample-set $\boldsymbol{s}_{t-1}^{(n)}$ according to the previous weight $\pi_{t-1}^{(n)}$. Some samples with high weight may be chosen several times, while others with low weight may not be chosen at all.

Every sample chosen from the weighted sample-set now proceeds to a predictive step where new states of those samples are determined according to the prediction model $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1} = \boldsymbol{s}_{t-1}^{(i)})$ (where $\boldsymbol{X}_t$ is a parameter vector which describes the object's state). The final operation is the observation step where we calculate the weights by evaluating the observation model $p(\boldsymbol{Z}_t|\boldsymbol{X}_t)$ (where $\boldsymbol{Z}$ is the observation. In our method, $\boldsymbol{Z}$ is the 3D positions of reconstructed objects), and obtain the sample-set $(\boldsymbol{s}_t^{(n)}, \pi_t^{(n)})$ of time t.

3.1 A mixed-state model

For example, a bouncing ball has two states, constant acceleration state and bounce state. In other words, it needs two prediction models $p_i(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})$ (where i is the number of states, $i = 1$ or 2 for this example) to support the CONDENSATION algorithm for tracking of moving object. We call the supported algorithm as "mixed-state model", which automatically switches between multiple dynamical model. The extended samples' state is defined to be

$$\boldsymbol{X} = (\boldsymbol{x}, y), \quad y \in 1, ..., N_S \tag{1}$$

where y is a label for each variable of the current model, $\boldsymbol{x}$ is a vector in parameter space which describes the object's position and velocity, and N_s is the number of state. The prediction model $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})$ can then be decomposed as follows:

$$p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}) = p(\boldsymbol{x}_t|y_t, \boldsymbol{X}_{t-1})P(y_t|\boldsymbol{X}_{t-1}) \tag{2}$$

$$P(y_t|\boldsymbol{X}_{t-1}) = P(y_t = j|\boldsymbol{x}_{t-1}, y_{t-1} = i) = T_{ij}(\boldsymbol{x}_{t-1}) \tag{3}$$

where the T_{ij} are the state transition probabilities. Equ.(2) describes the prediction model. Given the old parameter vector $\boldsymbol{X}_{t-1}$ then we obtain the state y_t (by equ.(3)), and we obtain the objects' state of the next time step by the dynamical model $p(\boldsymbol{x}_t|y_t, \boldsymbol{X}_{t-1})$, which is described as equ.(4).

$$p(\boldsymbol{x}_t|y_t, \boldsymbol{X}_{t-1}) = P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, y_{t-1} = i, y_t = j) \tag{4}$$

We obtain the new sample state from the old sample state with equ.(2), equ.(3), and equ.(4).

This mixed-state CONDENSATION algorithm is summarized as follows:

First, we construct a "new" sample-set $\{\boldsymbol{s}_t^{(n)}, \pi_t^{(n)}, n = 1, ...N\}$ for time t from the "old" sample-set $\{\boldsymbol{s}_{t-1}^{(n)}, \pi_{t-1}^{(n)}, n = 1, ...N\}$ at the time $t-1$. The n^{th} sample of N new samples is obtained by:

1. **Selection** of a sample $\boldsymbol{s}_t'^{(n)} = (\boldsymbol{x}'^{(n)}, i)$:
   (a) Generate a random number j with a probability proportional to the weight $\pi_{t-1}^{(j)}$.
   (b) set $\boldsymbol{s}_t'^{(n)} = \boldsymbol{s}_{t-1}^{(j)}$
2. **Prediction:** sample's new state $\boldsymbol{s}_t^{(n)}$ is predicted by prediction model $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1} = \boldsymbol{s}_t'^{(n)})$.
   (a) We obtain the sample's state $y_t^{(n)}$ by $P(y_t^{(n)} = j|\boldsymbol{X}_{t-1} = s_t'^{(n)}) = T_{ij}(\boldsymbol{x}_t'^{(n)})$
   (b) Then we obtain the $\boldsymbol{x}_t^{(n)}$ from $p(\boldsymbol{x}_t^{(n)}|\boldsymbol{X}_{t-1} = \boldsymbol{s}_t'^{(n)}, y_t^{(n)} = j)$
3. **Measurement:** the new position was weighted in terms of the observation data $\boldsymbol{Z}_t$:

$$\pi_t^{(n)} = p(\boldsymbol{Z}_t|\boldsymbol{X}_t = s_t^{(n)})$$

Finally normalize so that $\Sigma_n \pi_t^{(n)} = 1$. Then, let $t = t-1$ and go back to the first step.

When objects' motions are tracked with this algorithm, the state transitions T_{ij} are the most important and to be determined. But, in this paper we set the state transition probability by hand (details are described in the next section).

As preliminary experiments, we tried a mixed-state CONDENSATION algorithm to bouncing ball in single image sequence. The result is shown in Fig.4, where a tracker tracks the bouncing ball by using the

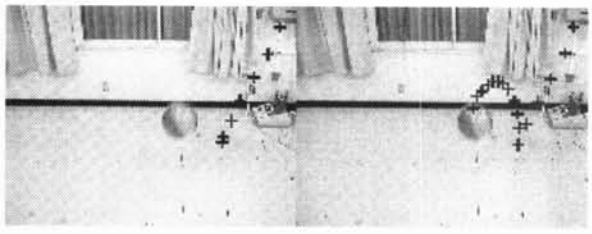
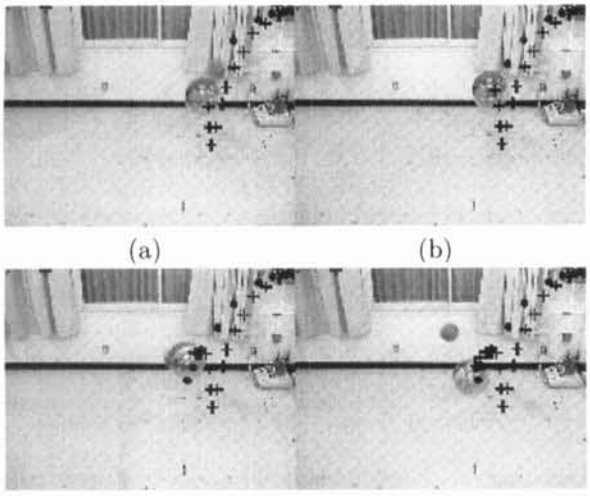


Figure 4: The results of a single state CONDENSATION algorithm(left) and the mixed-state CONDENSATION algorithm(right): left figure shows the tracker didn't track after bounce state, while in the right figure it tracks after bounce state successfully.



(a) (b)

(c) (d)

Figure 5: The failure result of tracking two balls in frames (a)26, (b)27, (c)30, and (d)32

mixed-state CONDENSATION algorithm and the standard CONDENSATION algorithm. In Fig.4 the symbol "+" shows the center of tracker at each time t. The result demonstrates that the mixed-state CONDENSATION algorithm improves tracking performance in the case of plural motion states exist. When there are several motion objects, and occlusions between them occurs. However, this method fails to track multiple objects with a single image sequence; Fig.5 shows an example where "+" and "•" also show the trajectories of centers of trackers. Initially, trackers tracked two balls separately (a), but after occlusion occurred (b) and (c), trackers didn't distinguish two balls (d). In this paper we introduce the method to overcome the effects of the occlusion by using multiple images.

4 Implementation details

Our purpose is understanding 3D dynamic environment, so that we need to extend the mixed state model from 2D to 3D. We assign one tracker to one object and one state $\boldsymbol{x}_t$ to one object. The state has six parameters, three position parameters $x_t = (u_t, v_t, h_t)^\top$ (where u and v are the horizontal positions, and h is the vertical position) and three velocity parameters $\dot{x}_t = (\dot{u}_t, \dot{v}_t, \dot{h}_t)^\top$. The state $\boldsymbol{x}_t$ is described as follows:

$$\boldsymbol{x}_t = \begin{pmatrix} x_t \\ \dot{x}_t \end{pmatrix} \tag{5}$$

The sample's state dynamical model is given by

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{B}\omega_t \tag{6}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are the matrices of dynamical model, and ω_t is Gaussian system noise $N(0, \sigma_w)$. We assume the transition matrix T_{ij} as constant and it depends only on the previous state, so we set $T_{ij}(\boldsymbol{x}_t) = T_{ij}$. In the experiments, the matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{T}$ were given by hand to distinguish the model states.

For the experiments described later, the initial state $\boldsymbol{x}_0^{(n)}$ of the sample set was given by hand as

$$\boldsymbol{x}_0^{(n)} = \begin{pmatrix} x_0 \\ \dot{x}_0 \end{pmatrix}, \quad y_0^{(n)} = s_1, \quad \pi_0^{(n)} = \frac{1}{W_N} \tag{7}$$

where s_1 is sample's initial state, n is sample number and W_N is the total weight of initial sample-set.

5 Experiments and Results of Tracking bouncing balls

We construct a model to track balls bouncing on a table. This case requires two states, s_1 is "constant acceleration state" and s_2 is "bouncing state".

The transition matrix T_{ij} is given (manually) as

$$T = \begin{pmatrix} 0.9 & 0.1 \\ 1.0 & 0.0 \end{pmatrix} \tag{8}$$

where the component value mean transition probabilities.

For example, the T_{1i} means that if the previous state was s_1 (constant acceleration state) then at the next time state remains s_1 with probability 0.9 and transits to s_2 with probability 0.1. We set the model as follows:

Constant acceleration model

- position transition is given as

$$\begin{cases} u_t = u_{t-1} + \dot{u}_{t-1} + \omega_t \\ v_t = v_{t-1} + \dot{v}_{t-1} + \omega_t \\ h_t = h_{t-1} + \dot{h}_{t-1} + \gamma_t + a \end{cases} \tag{9}$$

$$\omega_t \in N(0, \sigma_\omega), \quad \gamma_t \in N(0, \sigma_\gamma) \tag{10}$$

- velocity transition is given as

$$\dot{\boldsymbol{x}}_t = \boldsymbol{x}_t - \boldsymbol{x}_{t-1} \tag{11}$$

where a is constant acceleration which depends on the gravity and image frame rate (set manually in our experiment).

Bounce model

In the s_2 ("bounce state"), the transition models about horizontal positions u and v are the same as in the s_1 ("constant acceleration model").

- vertical position transition is given as

$$h_t = h_{t-1} - e\dot{h}_{t-1} + \gamma_t \tag{12}$$

- its velocity transition is given as

$$\dot{h}_t = -e\dot{h}_{t-1} \tag{13}$$

where e is coefficient of restitution of the ball, which depends on the object's characteristics (e.g., shapes, materials ...).

We set six cameras in a room (Fig.6) to track multiple objects moving around in the room.

The image sequence contains two bouncing balls, whose radii are 10cm. They fall into the scene with the initial velocities. The horizontal velocities change at each timestep according to the Gaussian noise $N(0, \sigma_o)$ m/s (where $\sigma_o = 0.01$).

We set the parameters of the model as constant acceleration $a = 0.05$ m/s^2, coefficient of restitution $e = 0.5$, the standard deviations of the noises $\sigma_\omega = 0.02$m and $\sigma_\gamma = 0.05$m respectively, and the number of samples $N = 100$. The observation model $p(\boldsymbol{Z}_t|\boldsymbol{X}_t)$ is defined as follows. Now we obtain one tracker's sample which have the 3D position x^i, then we think all the points in a sphere whose center is x^i

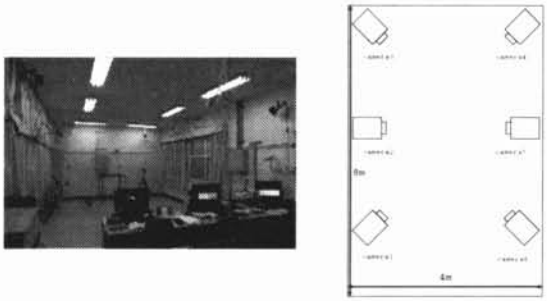


Figure 6: The camera positions: six cameras are set up in this room whose size is 8m×4m×3m

(its radius is optional we set 0.05m), and project all the point in the sphere to all the camera images. If the projected point is observed as objects (projected in a area of silhouettes) at least in three camera images, then the sample is assigned with the weight proportional to the number of projected points in the area of silhouettes. The result is shown in Fig.7 and Fig.8.

Fig.7 shows the reconstructed 3D trajectries of the two balls. Small angurations were caused by slight missyncronization of the cameras. Besides, the total reconstruction were rather well.

Fig.8 shows the tracking result of the mixed-state CONDENSATION algorithm in 3D environment. Each figure contains six camera images respectively. In the images, the large circles show two balls and the small circles show trackers. In the initial scene (frame 2) there are two balls, and trackers are assigned respectively. Occlusions appears in one image (frame 7 and 9 upper middle image), while another image has no occlusion. Therefore trackers are easily able to distinguish the balls. When the bouncing occurs (in frame 15 the white ball bounce), the system can track the balls by using the mixed-state CONDENSATION algorithm. Even if one camera could not observe the ball (frame 20 upper left and upper middle images), trackers tracks the two balls respectively because other cameras observed the balls.

6 Conclusions and Future Works

The robust tracking of multiple objects in an image sequence is proposed in this paper. We extended the 2D mixed-state CONDENSATION algorithm to 3D to realize the robust tracking. We track multiple objects successfully by using multiple cameras to reduce noises and mutual interferences. If some cameras had occlusions and did not observe objects, other cameras having no occlusion in the same frame tracked completely. For practical use, the total system must work automatically. For example, the automatic initialization remains as an interesting problem.

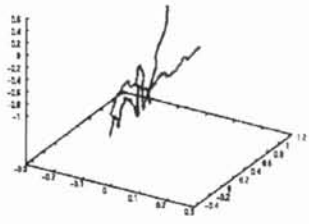


Figure 7: Error distance: the distance between the center of the object and that of the tracker

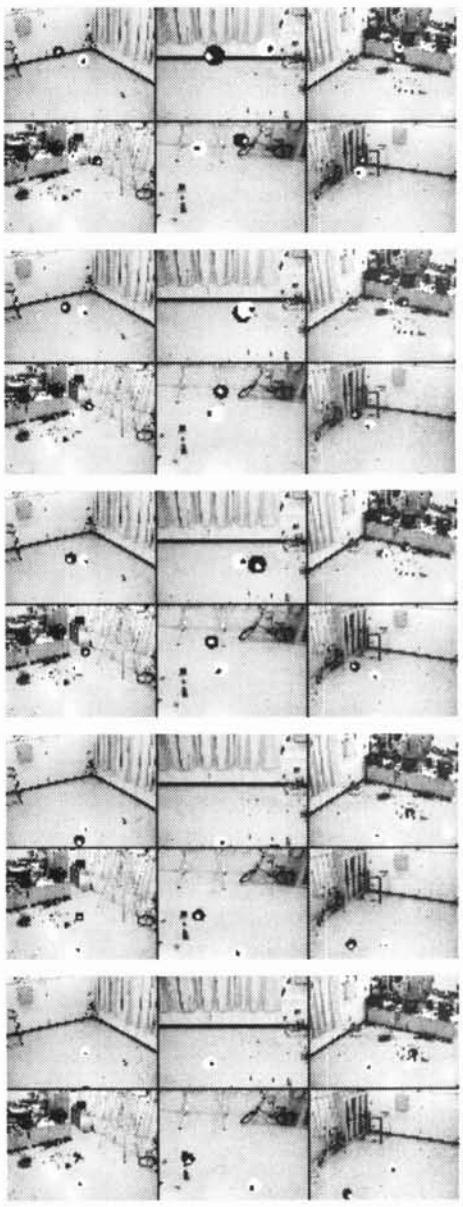


Figure 8: Result of the simulation sequence in frame from top to bottom 2, 7, 9, 15, 20: each images contain six cameras' images

References

[1] T. B. MoesLund and E. Granum, A Survey of Computer Vision-Based Human Motion Capture. Computer Vision and Image Understanding, 81(3), 231-268, 2001.

[2] M. A. Isard and A. Blake, CONDENSATION - conditional density propagation for visual tracking, Internanional Journal on Computer Vision, 29(1), 5-28, 1998.

[3] M. A. Isard and A. Blake, A mixed-state Condensation tracker with automatic model-switching, In Proc. 6th Int. Conf. on Computer Vision, 107-112, 1998.

[4] W. N. Martin and J. K. Aggarwal, Volumetric descriptions of objects from multiple views, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.5, No.2, 150-158, 1983.