Robot Motion Tracking System with Multiple Views

Hiroshi Yamano and Hideo Saito[†]

Department of Information and Computer Science, Keio University

3-14-1 Hiyoshi Kouhoku-ku Yokohama 223-8522, Japan

ABSTRACT

In such a space where human workers and industrial robots work together, it has become necessary to monitor a robot motion for the safety. For such robot surveillance, we propose a robot tracking system from multiple view images. In this system, we treat tracking robot movement problem as an estimation problem of each pose parameter through all frames. This tracking algorithm consists of four stages, image generating stage, estimation stage, parameter searching stage, and prediction stage. At the first stage, robot area of real image is extracted by background subtraction. Here, Yuv color system is used because of reducing the change of lighting condition. By calibrating extrinsic and intrinsic parameters of all cameras with Tsai's method, we can project 3D model of the robot onto each camera. In the next stage, correlation of the input image and projected model image is calculated, which is defined by the area of robots in real and 3D images. At third stage, the pose parameters of the robot are estimated by maximizing the correlation. For computational efficiency, a high dimensional pose parameter space is divided into many low dimensional subspaces in accordance with the predicted pose parameters in the previous flame. We apply the proposed system for pose estimation of 5-axis robot manipulator. The estimated pose parameters are successfully matched with the actual pose of the robots.

Keywords: Multiple views, Model matching, Motion analysis, Industrial robot

1. INTRODUCTION

Recently, many factories are automated to increase the efficiency of work operation. Industrial robots play an important role for such factory automation. Recent advances in robotics technology make the robots widely used, e.g. for exact operation such as semiconductor production, or for the work at dangerous zone.

When workers engage in a routine operation, they gradually get tired and lose concentration and accuracy on their operation. Whereas, industrial robot can work constantly with computer program which defines the behaviour of this robot. It is impossible, however, to say definitely that accident does not happen, for example motor trouble or breakdown in electric system. So it has become necessary to monitor a robot motion for safety. For such robot surveillance, we need to construct a robot motion tracking system.

In this work, we aim at building a robot motion tracking system. There are many researches about tracking moving objects, especially human movements.[1,2] Tracking methods proposed before can be divided into two ways. One uses passive or active markers,[3] the other uses 3D human model.[4] The marker-based method tracks some markers on a target human body, fits tracking result data to human skeleton model, and recognizes human posture in each frame. The model-based method projects 3D model which approximate human body on real image, and decide each pose parameters (i.e. joint angle) to match the model best with real human in the image. When those techniques are applied to robot tracking system, it is likely to choose the model-based technique. If we use the marker-based method, we need to attach markers to the robot. To avoid occlusion for good tracking, it is more desirable that the markers are solid shape than plane shape. However, such markers can obstruct the movement of the robot. Therefore, the model-based method is adequate to our purpose. Moreover, it is easy to construct robot 3D model because it mostly consists of plural rectangular solids and cylinders and detailed size is described in the operating manual.

E-mail address

Hiroshi Yamano: yamano@ozawa.ics.keio.ac.jp

Hideo Saito: saito@ozawa.ics.keio.ac.jp

†PRESTO, Japan Science and Technology Corporation(JST)

Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision,
David P. Casasent, Ernest L. Hall, Editors, Proceedings of SPIE Vol. 4572 (2001)

There is another question whether we choose single camera system[5] or multiple camera system.[6] The advantage of single camera system is its easiness of construction and fast performance. However, occlusion problem makes the result unfavorable. Multiple camera system reduces the invisible part from any cameras and lead to better result.

According to such consideration, we develop the system for tracking a robot motion by multiple images. We take model-based strategy for the pose estimation. The robot shape model is initially constructed in accordance with the physical size of the robot. The pose parameters are estimated by finding the maximum correlation between the silhouette images and the model projected images.

2. MULTIPLE VIEW SYSTEM

We develop the multiple view system that has the advantage of less occlusion occurring and getting good tracking result. The system consists of four cameras to track the movements of the robot. When the numbers of camera increase, the problem of processing cost occurs.

Fig. 1 shows our multiple view system. Four cameras get the images of the robot's posture, then each PC for camera (C_1, C_2, C_3, C_4) processes this image (described in Sec. 3.1, 3.2). Silhouette extracted via these processes are sent to a PC for integration (S) through Ethernet. PC S performs parameter estimation of all the posture parameter values from the silhouette images as described in Sec. 3.3, and provides predicted pose for the next frame processing as described in Sec. 3.4.

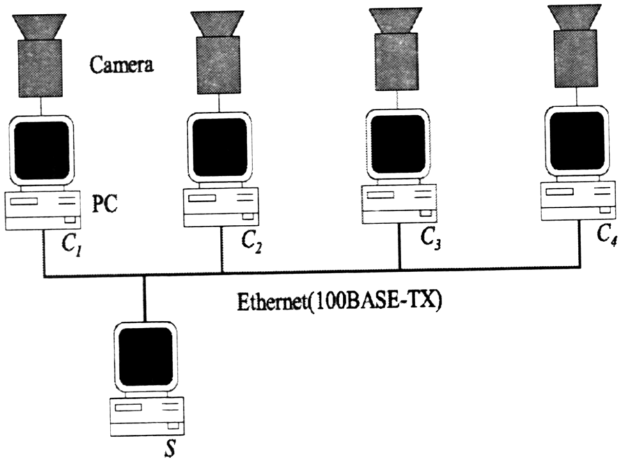


Figure 1. multiple view system

In this work, we use an industrial robot MOVEMASTER EX RV-M1 (Mitsubishi Electric Corp.) (Fig. 2). This robot is composed of one translational axis and four rotational axes. This composition is similar to human arms`, so we call the translational axis as "translation", and four rotational axes as "waist", "shoulder", "elbow" and "wrist" in the order starting from the bottom. Then we aim at finding the most appropriate values of these five pose parameters.

3. ALGORITHM

A general framework for model-based tracking is mentioned by the early work of O'Rourke and Badler.[7] In the work, four main components are involved; synthesis, image analysis, state estimation and prediction. Our proposed algorithm is composed of four stages based on this idea. Image generating stage includes the synthesis components. In estimation stage, the image analysis is processed. Searching stage corresponds to the state estimation stage, and prediction stage is provided. See details in the following sections.

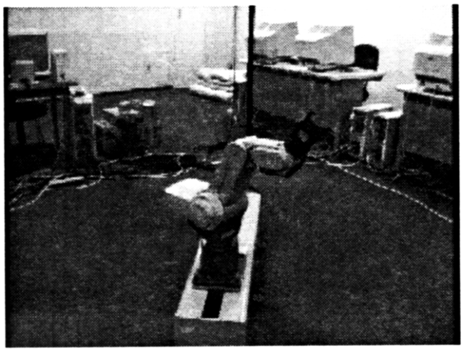


Figure 2. MOVEMASTER EX RV-M1

3.1. Image Generating Stage

3.1.1. Extraction of the Robot Region

To extract the robot region from the image taken by the camera, the original image is subtracted by the background image that is previously acquired. At this time, the fluctuation of intensity between the background image and the input images must be considered. For adapting the fluctuation, we first correct the intensity of the original image by the use of the background image. The sum of the intensity value at the four corner areas in both the original and the background image is calculated. Each area is 16×16 in our experiment. The ratio of the two sum totals is multiplied by each pixel of the original image. The reason four corner areas are selected is to avoid the use of the robot region. However it is difficult to say that the robot is never seen in the four corners, then we set a threshold to the ratio, the correction is not carried out when the ratio is over the threshold.

After the correction of the intensity, the background subtraction is executed. Though correcting the intensity, a proof method against the fluctuation of the intensity is still desirable. Besides, when the intensity is corrected with the method mentioned above, the influence of a shadow cannot be removed. We converse the color model of all images from RGB to Yuv to handle this problem. In the Yuv color model, Y represent a brightness, u is a difference between B and Y, and v is a difference between R and v. This conversion is defined by Eq. 1, 2, 3.

$$Y = 0.3R + 0.6G + 0.1G \quad (1)$$

$$u = B - Y \quad (2)$$

$$v = R - Y \quad (3)$$

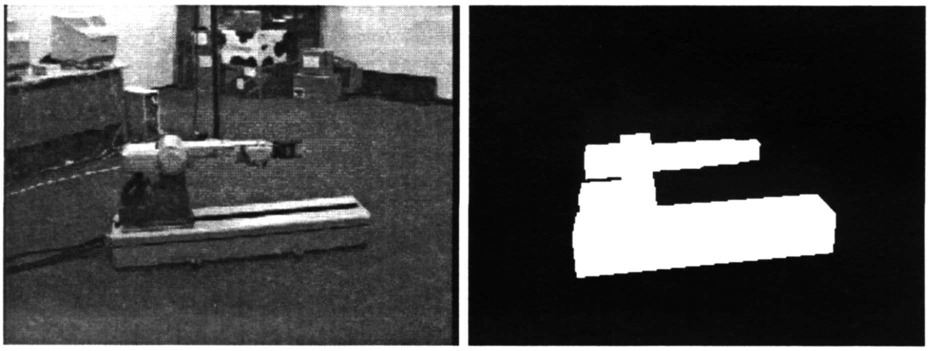
We set thresholds to u and v, a pixel which u or v value is smaller than the thresholds is represented by white, the other pixel is represented by black.

Fig. 3 shows an example of robot region extraction. Fig. 3(c) is an extraction image with RGB color model, Fig. 3(d) is an extraction image with Yuv color model. It can be seen that a shadow region is removed in the Yuv image.

3.1.2. generation of 3D model image

Here we project a 3D model of the robot to an image plane, generate a 3D model image. When we compare the 3D model image with the robot extraction image, it is important to match the position and size of both images precisely. For that purpose, we need to calculate extrinsic and intrinsic parameters of all the cameras we use. In this work, Tsai's calibration method[8] is used to get these parameters. This method calculates all camera parameters from many 3D points in the world coordinate and 2D points correspond to the 3D points in the image coordinate. Among these parameters, translational components and rotation angles for the transform between the world and camera coordinate frames, focal length, 2D coordinates of center of radial lens distortion in the image coordinate, and actual length of one pixel is required. These parameters are applied to OpenGL technology, and the 3D model image is generated. The pixels in the robot region are represented by white, the others black.

As to 3D model representation, Gavrila and Davis[9] use tapered super-quadrics and Wachter and Nagel[5] use right-elliptical cones. Since the shape of the industrial robot is not so much complex as human, we represent the



(a) actual image

(b) 3D model image

Figure 4. an example of 3D model image

s is the sum of these calculation values at each coordinates. The smaller s is, the more similar the two images are.

With Eq. 4, the estimation value is influenced by the percentage of the robot region in the image. If the robot region accounts for large percentage of the image, a slight gap between the two robot regions causes a large estimation result and vice versa. Then s is normalized by the robot region r_d in the extraction image (Eq. 6).

$$s' = \frac{s}{r_d} \tag{6}$$

Now, we intend to give weight depending on each camera to the estimation value. There are some motions of the robot that can easily be recognized from one camera and cannot from other camera. For example, we assume that two cameras are located around the robot, one is at the front of the traveling frame, and the other is at the side. When the robot moves along its rail, it is easier to recognize this motion with the side camera than the front one. In this case it is effective that we give priority to the estimation value of the side camera. Then we give large weight to the estimation value of the side camera and small to the values of the front camera. The problem is how to calculate a suitable weight value. We consider the change of the image correspond to a slight moving as recognition degree, and calculate the weight based on this definition. Details in next section, when we try to find one pose parameter we generate a certain number of 3D model images in which the robot moves along only one axis slightly. If there are much changes among these images we judge the motion easily recognizable from the camera at a position correspond to these model images. We define the weight w as Eq. 7

$$w = \sum_{i=1}^{n-1} \sum_{u} \sum_{v} g_i(u, v) \oplus g_{i+1}(u, v) \tag{7}$$

where n is the number of the 3D model images, $g_1(x, y)$, $g_2(x, y)$, $\cdots$, $g_n(x, y)$ are the model image sequences in which one pose parameter increase or decrease with definite amount. As with the calculation of s, we need to take account of the influence that a large robot region leads w to a large value. Then we improve the weight as

$$w' = \frac{w}{\sum_{j=1}^{n} r_j} \tag{8}$$

where r_1, r_2, $\cdots$, r_n are the robot regions in the 3D model images. s' is divided by w', a final estimation value is calculated.

3.3. Parameter Searching Stage

The estimation values from all the PCs for camera are integrated and most appropriate parameter values are selected on the basis of the integrated values. In this work, one parameter is changed n times (n is the same number as Eq. 7)

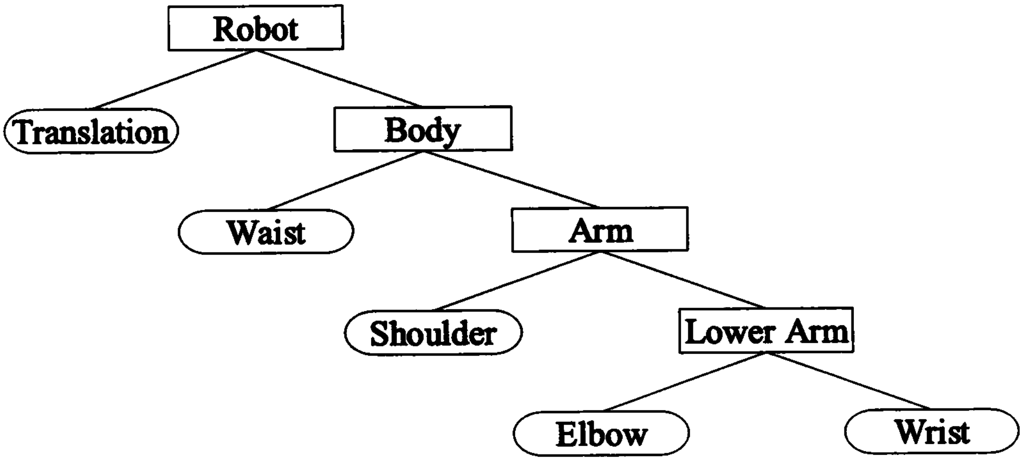


Figure 5. A Searching Tree in Our Experiment

at a constant interval. When changing a parameter, we generate a projected 3D model image corresponding to this parameter. Then this image is compared with the robot extraction image for calculating an estimation value. We decide that a parameter value which leads the sum of the estimation values from each camera to the minimum is the most appropriate parameter value. If we try to find all the parameters at once, the dimensions of the pose parameter space become high, the number is the same as the number of pose parameters (5 in our experiment). This causes a high cost calculation. Then we divide the high dimensional pose parameter space into many low dimensional spaces,[9] increase calculational efficiency.

We define k numbers of pose parameters at frame t as $p_{1,t}$, $p_{2,t}$, $\cdots$, $p_{k,t}$, and pose parameter space Σ as Eq. 9, 10.

$$\Sigma \;=\; \{\{p_{1,t}\} \times \{p_{2,t}\} \times \cdots \times \{p_{k,t}\}\} \tag{9}$$

$$\{p_{i,t}\} \;=\; \{\hat{p}_{i,t} - L_i , \cdots , \hat{p}_{i.t} + L_i\} \text{ , step } \Delta_i \tag{10}$$

In Eq. 10, $\hat{p}_{i,t}$ is a prediction value derived from the parameters up to frame $t - 1$ (detailed in Sec. 3.4). Eq. 10 means the searching space composed of values from $\hat{p}_{i,t} - L_i$ to $\hat{p}_{i,t} + L_i$ with step Δ_i. This space is divided into Σ_1 and Σ_2 with prediction values and already found values as

$$\Sigma \;=\; \{\Sigma_1, \Sigma_2\} \tag{11}$$

$$\Sigma_1 \;=\; \{\{p_{1,t}\} \times \{p_{2,t}\} \times \ldots \times \{p_{l,t}\} \times \{\hat{p}_{l+1,t}\} \times \{\hat{p}_{l+2,t}\} \times \ldots \times \{\hat{p}_{k,t}\}\} \tag{12}$$

$$\Sigma_2 \;=\; \{\{\tilde{p}_{1,t}\} \times \{\tilde{p}_{2,t}\} \times \ldots \times \{\tilde{p}_{l,t}\} \times \{p_{l+1,t}\} \times \{p_{l+2,t}\} \times \ldots \times \{p_{k,t}\}\} \tag{13}$$

where $\tilde{p}_{1,t}$, $\tilde{p}_{2,t}$, $\cdots$, $\tilde{p}_{l,t}$ are derived from the best solution to searching Σ_1. Eq. 11 , 12 , 13 denote that when finding a set of the most appropriate parameters we set prediction values to unknown parameters and found values to already searched parameters. The searching space decomposition is applied recursively and finally all the space is one-dimensional.

The scheme of dividing the searching space depends on the structure of a tracking target. In our experiment, the decomposition is represented as a tree in Fig. 5. In the tree, four non-leaf nodes (illustrated with rectangle) are spaces which can be divided and five leaf nodes (illustrated with ellipsoid) are spaces to be processed. If these processed spaces are all composed of n values the number of calculation times is $5n$ when this searching method is used, while n^5 when a quite simple method is applied.

3.4. Prediction Stage

In previous section, we search all the pose parameter at frame t. Next, we predict the parameters at $t + 1$ with the values. This prediction value correspond to $\hat{p}_{i,t+1}$ in Eq. 10, and determine the searching space at frame $t + 1$. Wachter and Nagel[5] represent the continuous changes by a state differential equation, predict with extended Karman filter. Gavrila and Davis[9] suppose that human motions are with constant accelerations. The robot motion is usually different from human, with constant velocity. Then we predict the parameters with constant velocity model.

Define the velocity $v_{i,t}$ of the parameter $p_{i,t}$ at frame t as

$$v_{i,t} = \tilde{p}_{i,t} - \tilde{p}_{i,t-1} \tag{14}$$

then the prediction value at $t+1$ can be represented by

$$\hat{p}_{i,t+1} = \tilde{p}_{i,t} + v_{i,t} \tag{15}$$

This prediction does not produce a good result when the estimated parameter value at t is much different from actual value. This is because that a wrong parameter value at only frame t leads a bad prediction value at frame $t+1$ with this. To deal with this problem, we use not only $v_{i,t}$, but other velocities at previous times $t-1$, $\cdots$, $t-3$. Eq. 16 shows a new prediction in which m_1 , $\cdots$, m_4 are weights to the velocities and the sum total is 1.

$$p_{i,t+1} = \tilde{p}_{i,t} + (m_1 v_{i,t} + m_2 v_{i,t-1} + m_3 v_{i,t-2} + m_4 v_{i,t-3}) \tag{16}$$

In addition, we improve the prediction to cope with a stop pose. In many cases the industrial robot moves only a few parts at a time and the other parts stop. Consequently we provide a threshold d for $v_{i,t}$ and when the absolute value of $v_{i,t}$ is less than d, the prediction value is set to $\tilde{p}_{i,t}$.

4. EXPERIMENTAL RESULTS

Fig. 6 shows an arrangement of four cameras in our experiment. Camera1 and 2 are placed at the same height, so camera 3 and 4 are also placed at the same height. 200 frame images of continuous robot motion are captured at 15 frame/sec. After recording, the five pose parameters were estimated by the proposed method. At this time, L_i and δ_i in Eq. 10 were 100mm and 50mm at translation axis and 20 degrees and 10 degrees at rotation axes respectively.

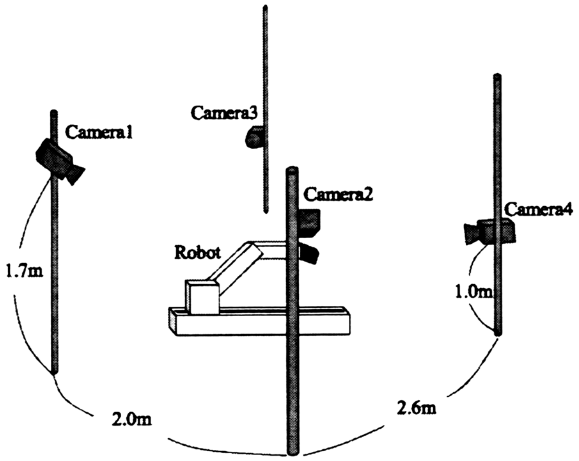
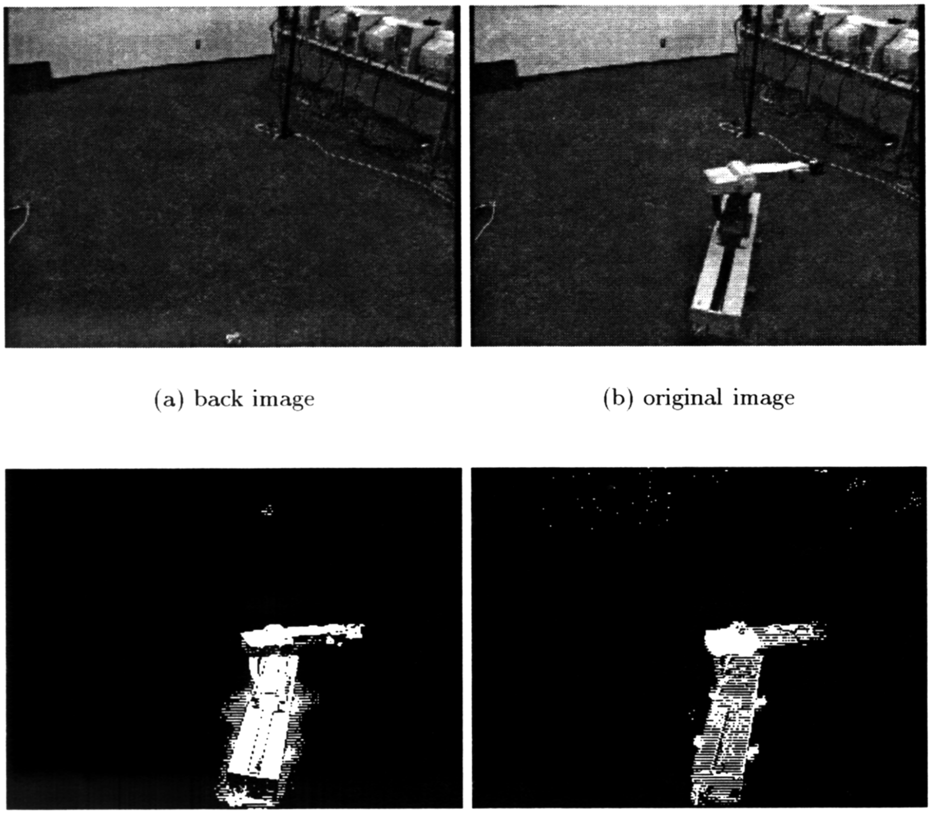


Figure 6. arrangement of four cameras

Fig. 7 shows some of the input images from camera2 (Fig. 7(a) to 7(d)) and camera4 (Fig. 7(e) to 7(h)), and the 3D wire frame model images obtained with our method are overlaid. It can be seen that the contours of the actual robot are nearly correspond with the skeletons of wire frame model. To investigate the result further, we show the actual values of translation, shoulder and wrist parameter and the searched values in Fig. 8. See Fig. 8(a), the tracking almost succeeded. About the shoulder parameter, the estimated parameter is almost the same as the actual. However, Fig. 8(c) indicates that the tracking of the wrist parameter includes much error because of the following reasons. One stems from the failure of the extraction. The region of robot hand was very small and black. These two factors might prevent our method extracting the region. The other reason is in searching stage. In this stage we divide a high dimensional pose parameter space into many low space. Though this derives an efficiency of searching, this causes a more error than high calculation cost method. Especially, An error at a parameter space affects a next search and more big error is produced. The wrist parameter is searched last, must be influenced by errors from previous searches.



(a) back image

(b) original image

(c) extraction image with RGB color model

(d) extraction image with RGB color model

Figure 3. extraction of the image

robot with rectangular solids. There is an automatic 3D model acquisition method from its silhouettes,[10] but in this work we model the robot manually because the exact shape of the robot is already known in most cases.

Fig. 4 shows an actual image in which all the parameter values are zero (Fig. 4(a))and correspondent 3D model image (Fig. 4(b)). The shape of the robot region in the 3D image is nearly the same as in the actual image.

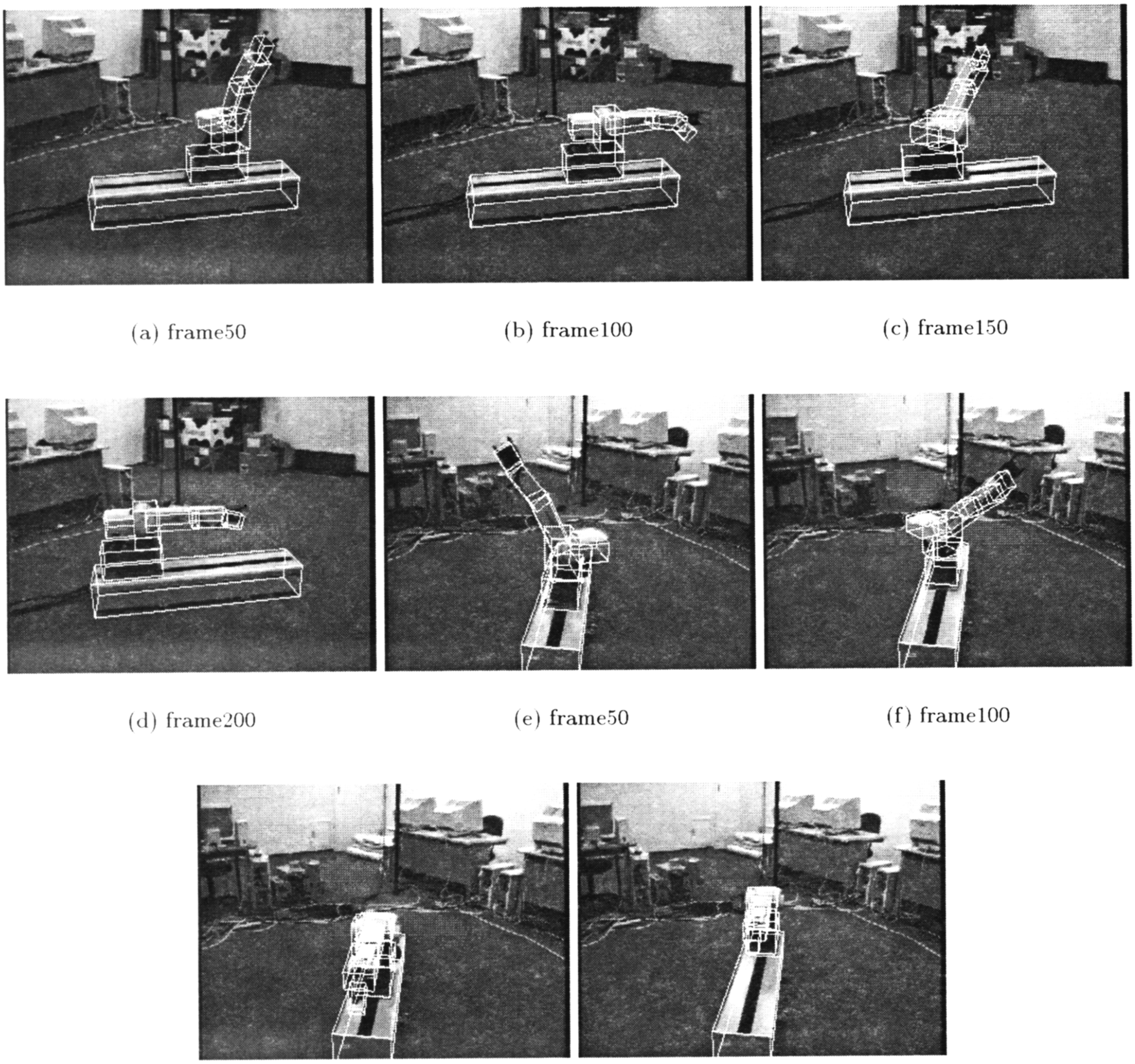
3.2. Estimation Stage

In this stage, we compare the robot extraction image and 3D model image, and calculate the correlation between the two images. Gavrila and Davis[9] extract edge contours of both images and calculate the distance of the two contours when comparing two images. This method derives good correlations, but costs a large amount of calculations. We calculate the correlation based on region of robots. Estimation value s which denote the similarity between the extraction image $f(x , y)$ and the model image $g(x , y)$ is defined by Eq. 4.

$$s = \sum_u \sum_v f(u , v) \oplus g(u , v) \tag{4}$$

In this equation, $\oplus$ represents an exclusive-OR between left and right terms. In both the two images, the robot region is represented by white and the other region is black, so Eq. 4 means that at a certain XY-coordinates s is a constant value (255 when 8bit image) if one image is white and the other is black, zero if both images are same color. That is,

$$(u , v) \oplus g(u , v) = \begin{cases} 0 , & \text{for } (u , v) = g(u , v) \\ const. , & \text{for } (u , v) \neq g(u , v) \end{cases} \tag{5}$$



(a) frame50 (b) frame100 (c) frame150

(d) frame200 (e) frame50 (f) frame100

(g) frame150 (h) frame200

Figure 7. The input and result images

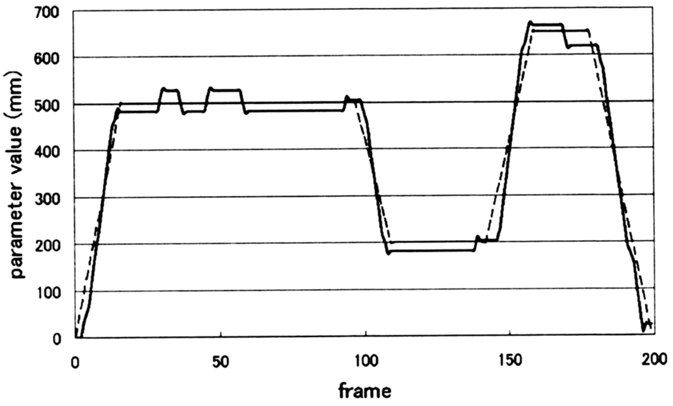
5. CONCLUSION

In this paper, we presented a system for tracking the robot arm motion from multiple view images. In the proposed method, the pose parameters of robot model that provide the best matching with silhouette images of input multiple view images are estimated by the use of hierarchical search of the parameters. We demonstrated that our proposed system is able to track the robot motion successfully. Though there leave something to be desired which is described in Sec. 4, the drawbacks may be improved not so difficultly. This result indicates that the model-based method is useful for robot motion tracking.

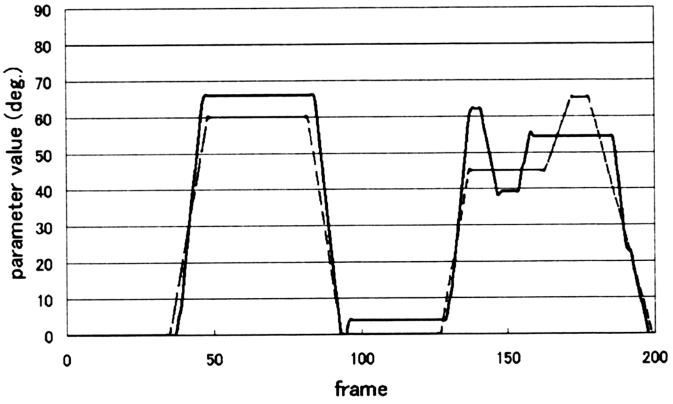
For a realization of robot surveillance, we need to improve the method more. There are two main improvement points. One is an automation of 3D robot modeling. There are many industrial robots of various shapes. To cope with all the robots the automation is expected. The other is more high speed processing. In our experiment the tracking took 2.6 second per frame. If this system is used to survey a robot real-time processing must be demanded.

REFERENCES

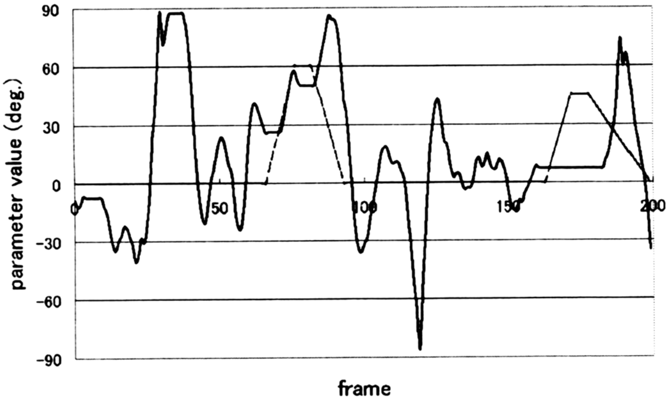
1. D. M. Gavrila, "The Visual Analysis of Human Movement: A Survey," *Computer Vision and Image Understanding* **73-1**, pp. 82–98, 1999.
2. A. Pentland, "Looking at People: Sensing for Ubiquitous and Wearable Computing," *IEEE Transaction on Pattern Analysis and Machine Intelligence* **22-1**, pp. 107–119, 2000.
3. M. Silaghi, R. Plänkers, R. Boulic, P. Fua, and D. Thalmann, "Local and Global Skeleton Fitting Techinques for Optical Motion Capture," *IFIP CapTech '98* , 1998.
4. M. Yamamoto, Y. Ohta, T. Yamagiwa, K. Yagishita, H. Yamanaka, and N. Ohkubo, "Human Action Tracking Guided by Key-Frames," *Proc. of the Forth IEEE Int. Conference on Automatic Face and Gesture Recognition* , pp. 354–361, 2000.
5. S. Wachter and H. H. Nagel, "Tracking Persons in Monocular Image Sequences," *Computer Vision and Image Understanding* **74-3**, pp. 174-192, 1999.
6. M. Yamamoto, A. Sato, S. Kawada, T. Kondo, and Y. Osaki, "Incremental Tracking of Human Actions from Multiple Views," *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , pp. 2–7, 1998.
7. J. O'Rourke and N. Badler, "Model-based Image Analysis of Human Motion Using Constraint Propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2-6**, pp. 522–536, 1980.
8. R. Y. Tsai, "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision," *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , 1986.
9. D. M. Gavrila and L. S.Davis, "3-D Model-based Tracking of Humans in Action: a Multi-view Approach," *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , pp. 73–80, 1996.
10. I. A. Kakadiaris and D. Metaxas, "3d Human Body Model Acquisition from Multiple Views," *Proc. of the Fifth Int. Conference on Computer Vision* , pp. 618–623, 1995.



(a) translation



(b) waist



(c) wrist

Figure 8. actual (dotted) and searched (solid) parameter value