Real-time Human Motion Analysis and IK-based Human Figure Control

Satoshi Yonemoto, Daisaku Arita and Rin-ichiro Taniguchi
Division of Intelligent Systems, Kyushu University
6-1 Kasuga-koen Kasuga Fukuoka, 816-8580 Japan
{yonemoto, arita, rin}@limu.is.kyushu-u.ac.jp

Abstract

This paper presents real-time human motion analysis based on real-time inverse kinematics. Our purpose is to realize a mechanism of human-machine interaction via human gestures, and, as a first step, we have developed a computer-vision-based human motion analysis system. In general, man-machine 'smart' interaction requires real-time human full-body motion capturing system without special devices or markers. However, since such vision-based human motion capturing system is essentially unstable and can only acquire partial information because of self-occlusion, we have to introduce a robust pose estimation strategy, or an appropriate human motion synthesis based on motion filtering. To solve this problem, we have developed a method based on inverse kinematics, which can estimate human postures with limited perceptual cues such as positions of a head, hands and feet. In this paper, we outline a real-time and on-line human motion capture system and demonstrate a simple interaction system based on the motion capture system.

1 Introduction

Man-machine seamless 3-D interaction is an important tool for various interactive systems such as virtual reality systems, video game consoles, etc. To realize such interaction, the system has to estimate motion parameters of human bodies in real-time. Up to the present, as a method for human motion sensing, many motion capture devices with special markers or magnetic sensor attachments have been employed. Since they need special marker-sensors, they often impose physical restrictions on the object. On the other hand, recently, fully image-feature-based motion capturing systems which do not impose such restrictions have been developed as a computer vision application[1]. Although the vision-based approach still has problems to be solved, it is a very smart approach which can achieve seamless human-machine interaction. Moreover, it has a merit that it can acquire shape properties and surface textures, which can not be measured by the former approach. Therefore, we are undertaking to develop an image-feature-based motion capturing system, giving consideration to alleviating scene constraints and physical constraints imposed on the system as little as possible.

To analyze human motion, image features such as blobs (coherent region)[1][2][3] or silhouette contours are usually employed. Since the contour-based image features essentially depend on human postures, they are appropriate only for the estimation of typical postures. Therefore, many researchers have developed skin-color region tracking and stereo reconstruction methods using general region clustering. In particular, *Pfinder*[1] has shown that blob tracking is applicable for many real-time applications although the idea is not very new. Recently, *purposeful human motion*[4] employing the above blob tracking has been proposed. It is based on an analysis and synthesis framework with fast dynamics engine[5], and with an HMM based multiple behaviour model. The method is applied to upper body motion estimation, and temporary occlusion in the intersection between blobs of both hands or head-and-hand is handled, although the method does not solve the corresponding problem. In gesture recognition systems, as well as in human motion tracking, human motion primitives are also dealt with. However, in these systems, gesture representation is symbolically defined and such symbol-based approaches are not appropriate for our purpose, which has to generate actual 3-D body postures, or 3-D positions of head, arms, feet, etc.

In this paper, we present vision-based human motion capture system based on inverse kinematics, which can estimate human postures with limited perceptual cues such as positions of a head, hands and feet. We also focus on on-line implementation of the motion capture system using a PC-cluster (multiple PCs connected via high-speed network).

2 System Overview

The flow of our algorithm of real-time motion capturing is as follows:

1. Detection of cues (*perception*)
   - 2-D color blob tracking for each view
   - Calculation of 3-D color blob position using multi-view fusion
2. Generation of human figure full-body motion and rendering in the virtual space and calculation of the interaction (*motion synthesis*)

0-7695-0939-8/00 $10.00 © 2000 IEEE

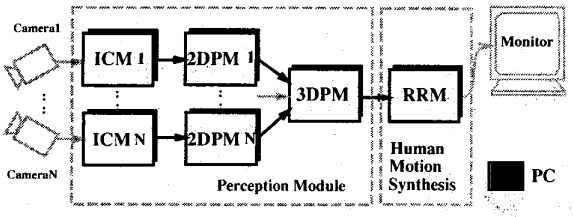


Figure 1: Image processing modules on PC cluster.

A prototypical system developed here is a real-time visually-guided-animation system, and make it real-time and on-line, ew have implemented the system on a PC cluster, which consists of multiple PCs connected via a high-speed network, *myrinet*[7]. Each pipeline step is controlled by a synchronization mechanism[8]. Fig.1 shows the system flow and allocation of processing modules to PCs.

Details of the processing modules are as follows:

a) Perception Module:

Image Capturing Module(ICM) These modules work as image-capturing and resizing modules (320×240). Each ICM_v ($v = 1, \cdots, V$; V is the number of cameras) sends image data to 2DPM_v.

2-D Processing Module(2DPM) These modules work as 2-D image processing modules (2-D blob tracking). Each 2DPM receives the image data from ICM_v, and sends 2-D extracted image feature data (positions of the center of gravity of the 2-D blobs) to 3DPM.

3-D Processing Module(3DPM) This module works as a 3-D vision processing module. It receives and integrates the 2-D image feature data from 2DPM_v ($v = 1, \cdots, N$), and estimates 3-D model parameters (3-D positions of blobs). The estimated parameters are sent to the RRM.

b) Human Motion Synthesis:

Real-time Rendering Module(RRM) This module works as a real-time renderer of the virtual space. It receives the 3-D blob positions from 3DPM and estimates 3-D pose and motion of the human body based on the received data.

In the following sections, we will show details of the algorithms and some experimental results.

3 Perception

3.1 Color Identification

In this system, skin color regions observed in an input image are interpreted as hand and face(head) blobs, regions with pre-acquired shoe or sock color as feet, and a region with shirt color as a torso blob(Fig.2). We assume the colors can be represented in a simple parametric form which is relatively robust for illumination changes[6]. In other words, we assume the color features (r,g,b) of each pixel are represented in the following quadratic equations of intensity i:

$$\begin{aligned} \hat{r} &= R_2 i^2 + R_1 i, \\ \hat{g} &= G_2 i^2 + G_1 i, \\ \hat{b} &= B_2 i^2 + B_1 i \end{aligned} \tag{1}$$

For each blob color to be identified, six model parameters, or coefficients, $R_1, \cdots, B_2$ are estimated in advance from a training data set, or real blob images. In color identification, the system computes the following error between observed color features (r, g, b) of a pixel and the model color features $(\hat{r}, \hat{g}, \hat{b})$ calculated, according to the above equation, from the intensity i of the pixel.

$$error = (r - \hat{r})^2 + (g - \hat{g})^2 + (b - \hat{b})^2 \tag{2}$$

The system identifies the color of a pixel as a color giving the minimum error that is less than a certain threshold.

3.2 2-D Blob Tracking

Blob tracking is accomplished according to the following steps:

1. A rectangle containing a human body is detected after background image subtraction and thresholding are applied. Then, regions with skin color, shoe/sock color or shirt color are identified by the above method. At the same time, the torso position (the center of gravity) is estimated.
2. In the rectangle, the color-identified pixels are classified into blobs based on the similarity of their colors and positions to those of blobs detected in the previous frame.

Here, we only assume that the result of background image subtraction is stable, and that cloth color is not similar to skin color. These assumptions can be easily met, particularly, in indoor or studio-like situations.

Initial correspondence of color-identified regions to specific 2-D blobs, i.e., a face, hands and feet, is decided when the system starts up, based on simple heuristics of the natural standing position. The heuristics employed here are as follows:

- The head is a skin color region upper-most in the detection rectangle.
- The left(right) hand is a skin color region which is on the left(right) in the detection rectangle.
- The left(right) foot is a pre-acquired color region which is on the left(right) in the detection rectangle.

This correspondence is also examined when the system fails to track the blobs. The error recovery process is quite important for online algorithms, and the decision process should be carefully designed.

3.3 Estimation of 3-D Blob Position

When a 2-D blob is detected in two views, the 3-D position of the blob can be calculated by a stereo method. However, since self-occlusion often occurs, with only two views it is almost impossible to estimate all parts of the moving body for a long period. Therefore, multi-view fusion is indispensable. In the blob tracking, precise estimation is not required and, therefore, we have employed a simple but fast multi-view fusion strategy. The algorithm of 3-D blob position calculation adopted here is as follows:

Selection of views According to the visibility of views, reliable views, or views whose visibility is higher than a certain threshold, are selected for each blob. The visibility is defined as the number of observed pixels in each blob, and it can indicate whether occlusion is occurring or not.

Calculation of line of sight According to camera calibration information, for each of the selected views, a line of sight, or a vector from the origin of the camera coordinate system to the center of gravity of the blob, is calculated.

Integration of multi-view information Referring to the acquired lines of sight, the 3-D position of each blob is calculated.

When a line of sight calculated for the most reliable view is parameterized as $\mathbf{T}_1 = \mathbf{o}_1 + t_1\mathbf{d}_1 (t_1$ is a parameter), and the rest of the lines of sight as $\mathbf{T}_j = \mathbf{o}_j + t_j\mathbf{d}_j (t_j$ is a parameter; $j = 2, \cdots, J)$, the intersection point $\mathbf{T}$ is approximated as a point on the line of sight $\mathbf{T}_1$ whose average distance to the other lines of sight is smallest in the sense of the least squares error.

$$\mathbf{T} = \mathbf{o}_1 - \frac{\sum_{j=1}^{J} (\mathbf{d}_1 \times \mathbf{m}_j, \mathbf{o}_1 \times \mathbf{m}_j - \mathbf{n}_j)}{\sum_{j=1}^{J} ||\mathbf{d}_1 \times \mathbf{m}_j||^2} \mathbf{d}_1, \quad (3)$$

where

$$\mathbf{m}_j = \frac{\mathbf{d}_j}{\sqrt{1+||\mathbf{o}_j \times \mathbf{d}_j||^2}}, \qquad \mathbf{n}_j = \frac{\mathbf{o}_j \times \mathbf{d}_j}{\sqrt{1+||\mathbf{o}_j \times \mathbf{d}_j||^2}}.$$

The calculated point $\mathbf{T}$ corresponds to the 3-D blob position $(T_x, T_y, T_z)^T$.

4 IK-based Motion Synthesis

Information acquired in the perception process is just 3-D positions of blobs, which correspond to a torso, a head, hands and feet of a human body. Therefore, to estimate the body posture from these cues, the number of which is less than the degree of freedom of the body, we have to solve the inverse kinematics[9]. In our case, a human body is represented as a multi-part articulated object, or as 14 parts with 23 degrees of freedom (see Fig.3), and the 3D blob positions are given as the goal positions, or the end effectors. Of course, there are approaches in which knees and elbows are detected based on contour analysis, or silhouette analysis, but they cannot stably detect those positions for many postures.

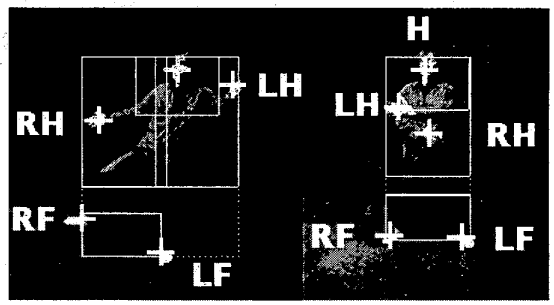


Figure 2: 2-D blob position estimation results.

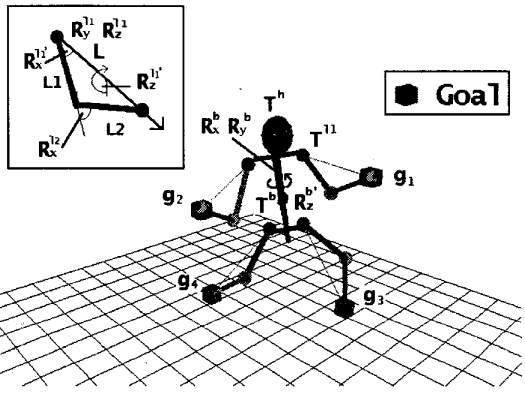


Figure 3: Our human figure model geometry.

The goals of the inverse kinematics which we have designed can be summarized as follows:

- Inverse kinematics of four connecting links, which are two arms and two legs, can be solved in real-time.
- Even when goal positions (3-D blob positions) given by the perception module are not precise, a solution can be derived to some extent.
- The solution gives us continuous and natural-looking motion of the human body.

In our case, as mentioned above, the 3-D blob positions acquired by the perception modules are sometimes imprecise. In other words, the goal positions are sometimes established at positions where physically possible solutions cannot be derived. Therefore,

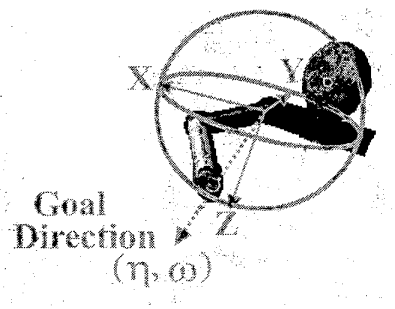


Figure 4: Definition of a goal with arm direction.

we interpret each of the given goals as the combination of the direction of the goal and the distance to the goal. When the goal position is located where a physically possible solution can not be derived, we find a solution in which the direction of the connecting link coincides with the goal direction (see Fig.4).

$$\begin{aligned}\mathbf{g}_i^w = \; & \mathbf{T}^b\mathbf{R}^b(0, {R_y}^b, {R_x}^b)\mathbf{R}^{b'}({R_z}^{b'}, 0, 0) \\ & \mathbf{T}^{l_1}\mathbf{R}^{l_1}({R_z}^{l_1}, {R_y}^{l_1}, 0)\mathbf{R}^{l'_1}({R_z}^{l'_1}, 0, {R_x}^{l'_1}) \\ & \mathbf{T}^{l_2}\mathbf{R}^{l_2}(0, 0, {R_x}^{l_2})\,\mathbf{t}^e\end{aligned} \quad (4)$$

where $\mathbf{g}_i^w = (g_x^w, g_y^w, g_z^w, 1)^T$ is a goal vector($i = 1, \cdots, 4$); $\mathbf{T}^b$, $\mathbf{R}^b$ and $\mathbf{R}^{b'}$ are matrices representing the body pose; $\mathbf{T}^{l_1}$, $\mathbf{R}^{l_1}$ $|\mathbf{R}^{l'_1}$, $\mathbf{T}^{l_2}$ and $\mathbf{R}^{l_2}$ are pose matrices related to link 1 (L1 in Fig.3)and link 2 (L2) respectively; $\mathbf{t}^e$ is a translation vector related to the end-effector position of L2.

Here, some rotation elements are represented in two matrices—$\mathbf{R}^{l_1}$ and $\mathbf{R}^{l'_1}$ of L1, for example. We have divided the original rotation matrix into two matrices to simplify analytical solution of our inverse kinematics. In $\mathbf{R}$, R_z, R_y, R_x represent roll, pitch, and yaw angle respectively.

4.1 Analytical Solution Using Real-time Inverse Kinematics

Analytical solution of the inverse kinematics previously mentioned is as follows[1]:

$${R_y}^{l_1} = -\arccos\left(\frac{g_z - {T_z}^{l_1}}{\|\mathbf{g} - \mathbf{T}^{l_1}\|}\right) \quad (5)$$

$${R_z}^{l_1} = -\arctan\left(\frac{g_y - {T_y}^{l_1}}{g_x - {T_x}^{l_1}}\right) \quad (6)$$

$${R_x}^{l'_1} = \arccos\left(\frac{{L_1}^2 + L^2 - {L_2}^2}{2L_1L}\right) \quad (7)$$

[1]All the coordinates here are represented in the local coordinate system of the torso.

$${R_x}^{l_2} = \arccos\left(\frac{{L_1}^2 + {L_2}^2 - L^2}{2L_1L_2}\right) - \pi \quad (8)$$
$$(|L_1 - L_2| \le L \le L_1 + L_2)$$

where L_1 L_2 L are the lengths of link 1, link 2, and the distance between the origin of link 1 and the goal position.

4.2 Estimation of Torso Posture

Torso posture consists of two elements, the axis of the torso and the pan angle around the axis. The axis of the torso is an axis connecting the centers of gravity of a head blob and a torso blob and is defined as follows:

$${R_x}^b = -\arcsin\left(\frac{{T_y}^h - {T_y}^b}{\|\mathbf{T}^h - \mathbf{T}^b\|}\right) \quad (9)$$

$${R_y}^b = -\arctan\left(\frac{{T_x}^h - {T_x}^b}{{T_z}^h - {T_z}^b}\right) \quad (10)$$

The pan angle (i.e. human body direction), is difficult to estimate correctly from perception results, or blobs. However, since we use multiple cameras, we can estimate the body pan angle for a variety of body postures. We estimate the pan angle based on the direction that both feet point, assuming that both feet touch the ground plane or that they are very close to the ground plane:

$${R_z}^{b'} = -\arctan\left(\frac{{T_y}^{rf} - {T_y}^{lf}}{{T_x}^{rf} - {T_x}^{lf}}\right) \quad (11)$$

The characteristics of our method can be summarized as follows:

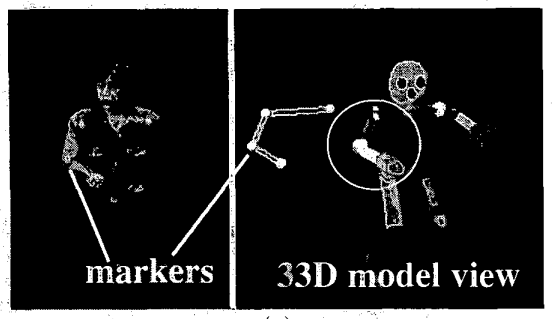
- Since only two-link inverse kinematics is solved, it can be used for real-time pose estimation of human bodies.
- Parameters which are not represented explicitly in the solution of the inverse kinematics, such as the pitch of elbow $({R_z}^{l'_1})$ (we call it the *characteristic angle*), can be used to control precise human body pose if necessary[2]

5 Implementation and Experiments

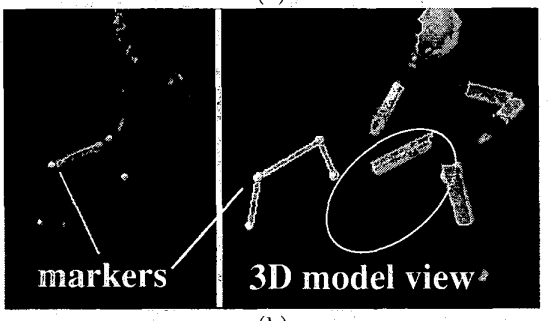
5.1 Characteristic Angle Estimation from Real Motion Capture Data

In order to investigate features of the characteristic angle ${R_z}^{l'_1}$, we measured its real angle for various arm and leg directions using a marker-based motion capture system[10], in which 4 markers are attached to

[2]In this paper, we have used a pre-acquired constant value based on measurements of limb postures using another motion capture device. See5.1.

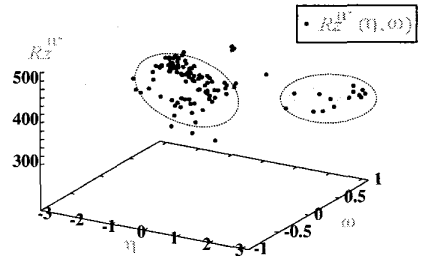


(a)

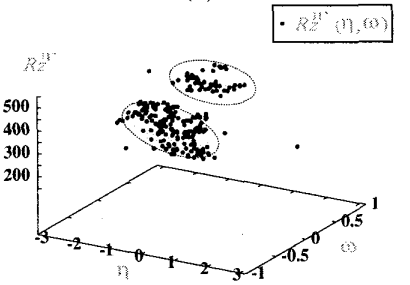


(b)

Figure 5: Samples of input images and estimated 3-D model views:(a)right arm,(b)right leg.



(a)



(b)

Figure 6: plots of characteristic angle: (a)right arm, (b)right leg.

both shoulders(hips), the elbows(knees) and the hands(feet) of the right arm(leg). Fig.5(a)(b) shows samples of input images and of reconstructed 3-D model views, which poses are estimated by our IK method. Fig.7 shows 3-D position trajectories of a right elbow and a right knee. Fig.6 shows plots of the characteristic angle ${R_z}^{l'_1}$ with various arm and leg directions (latitude η and longitude ω, see Fig.4). From this result, we can suppose that the characteristic angles can be approximated by constant values. In fact, for example, the errors between the elbow positions of the human body generated referring to measured angles and ones generated with a constant angle, which is a mean value of the measured angles, 16[deg] in (a), are small enough to reconstruct natural poses of the model. In case of (b), or in case of the right leg, the mean value of the characteristic angles is 236[deg].

5.2 Real-time Interaction between Human and Virtual Environment

Here, the human figure motion generator mentioned above is applied to *Visually Guided 3D Animation*, or a real-time (video-rate) online interaction system in a virtual space. The example shown in Fig.8 shows a user, or a target object, which is visualized as an *avatar* in the virtual space, kicking a virtual soccer ball. Its interaction is simply realized by detecting collision of the ball and the body parts and by simulating rebound of the ball from the body. In this case, a delay of about 0.2 second is inevitable because of the latency of the pipelined implementation. Therefore, in these shots(Fig.8), the avatar posture is slightly different from that of user.

6 Conclusions

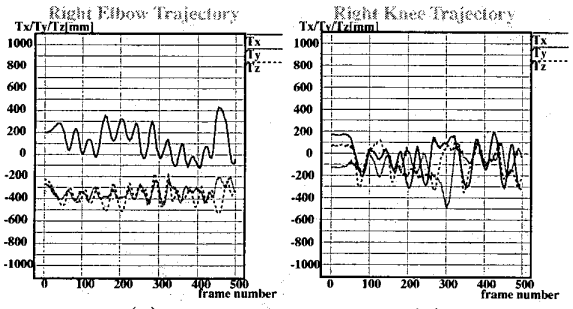
In this paper, we have shown a real-time human motion capturing without special marker-sensors. We have adopted multi-view fusion and inverse kinematics to realize full-body motion analysis from a limited number of perceptual cues. Since the system works in real-time and online, it can be applied to various *real-virtual* applications such as smart man-machine 3-D interaction. In future work, we will achieve more natural motion of the human model by employing emotional and dynamical filtering.

Acknowledgments

This work has been partly supported by "Cooperative Distributed Vision for Dynamic Three Dimensional Scene Understanding (CDV)" project (JSPS-RFTF96P00501, Research for the Future Program, the Japan Society for the Promotion of Science).

References

[1] C.Wren, A.Azarbayejani, T.Darrell, A.Pentland, "Pfinder: Real-Time Tracking of the Human Body", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.19, No.7, pp.780-785, 1997.



(a) (b)

Figure 7: 3-D position trajectory: (a)right arm, (b)right leg.

[2] C.Bregler, "Learning and Recognizing Human Dynamics in Video Sequences", *in Computer Vision and Pattern Recognition*, pp.568–574, 1997.

[3] M.Etoh, Y.Shirai, "Segmentation and 2D Motion Estimation by Region Fragments", *in International Conference on Computer Vision*, pp.192–199, 1993.

[4] C.Wren, A.Pentland, "Understanding Purposeful Human Motion", *in Fourth IEEE International conference on Automatic Face and Gesture Recognition*, 2000.

[5] A.Witkin, M.Gleicher, W.Welch, "Interactive Dynamics", *in ACM SIGGRAPH*, Vol.24, no.2, pp.11–21, 1990.

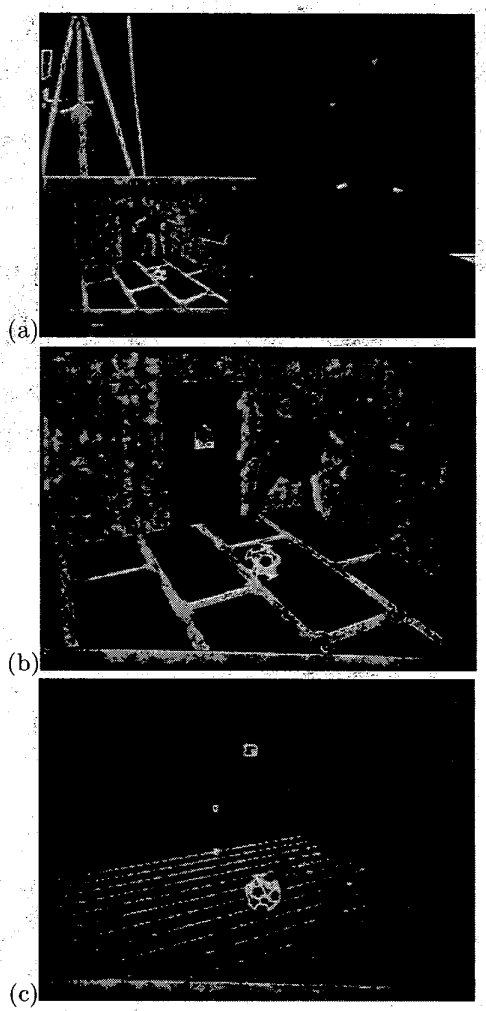
[6] Y.Okamoto and R.Cipolla and H.Kazama and Y.Kuno, "Human Interface System Using Qualitative Visual Motion Interpretation", *IEICE*, Vol.J76-D-II, No.8, pp.1813–1821, 1993.

[7] Myrinet. http://www.myricom.com

[8] D. Arita, N. Tsuruta and R. Taniguchi, "Real-time parallel video image processing on PC-cluster", *Parallel and Distributed Methods for Image Processing II, Proceedings of SPIE*, Vol.3452, pp.23–32, 1998.

[9] J. Zhao and N. Badler: Inverse Kinematics positioning using nonlinear programming for highly articulated figures, Transactions on Computer Graphics, Vol.13, No.4, pp.313–336, 1994.

[10] S.Yonemoto, N.Tsuruta, and R.Taniguchi: "A Real-time Motion Capture System with Multiple Camera Fusion", Proc. ICIAP'99, pp.600-605, 1999.



(a) (b) (c)

Figure 8: Online demo shots: (a)*real-virtual soccer scene*, (b)zoom in, (c)6 blob representation. Note that these shots were taken at different times.