Real-time Human Motion Analysis for Human-Machine Interface

Taniguchi, Rin-ichiro
Department of Intelligent Systems, Kyushu University

Yonemoto, Satoshi
Department of Intelligent Systems, Kyushu University

Arita, Daisaku
Department of Intelligent Systems, Kyushu University

Hoshino, Ryuya
Department of Intelligent Systems, Kyushu University

出版情報：Proceedings of Working Conference on Advanced Visual Interfaces, pp.195-202, 2002-05
バージョン：
権利関係：

Real-time Human Motion Analysis for Human-Machine Interface

Rin-ichiro Taniguchi*
Dept. of Intelligent Systems
Kyushu University
6-1, Kasuga-koen, Kasuga
Fukuoka 816-8580 Japan
rin@limu.is.kyushu-u.ac.jp

Satoshi Yonemoto
Dept. of Intelligent Systems
Kyushu Sangyo University
2-3-1, Matsukadai
Fukuoka 813-8503 Japan
yonemoto@limu.is.kyushu-u.ac.jp

Daisaku Arita
Dept. of Intelligent Systems
Kyushu University
6-1, Kasuga-koen, Kasuga
Fukuoka 816-8580 Japan
arita@limu.is.kyushu-u.ac.jp

ABSTRACT

This paper presents real-time human motion analysis for human-machine interface. In general, man-machine 'smart' interface requires real-time human motion capturing systems without special devices or markers. Although vision-based human motion capturing systems do not use such special devices and markers, they are essentially unstable and can only acquire partial information because of self-occlusion. When we analyze full-body motion, the problem becomes more severer. Therefore, we have to introduce a robust pose estimation strategy to deal with relatively poor results of image analysis. To solve this problem, we have developed a method to estimate full-body human postures, where an initial estimation is acquired by real-time inverse kinematics and, based on the estimation, more accurate estimation is searched for referring to the processed image. The key point is that our system can estimate full-body human postures from limited perceptual cues such as positions of a head, hands and feet, which can be stably acquired by silhouette contour analysis.

Categories and Subject Descriptors

I.4 [**Computing Methodologies**]: Image Processing and Computer Vision; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*motion, tracking, object recognition*; H.5 [**Information Systems**]: Information Interfaces and Presentation; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*interaction styles*

General Terms

Experimentation, Human Factors

*Web page of the authors' laboratory is `http://limu.is.kyushu-u.ac.jp`

AVI 2002, Trento, Italy.

Keywords

Human motion analysis, multiview image analysis, real-time vision, vision-based interaction

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, which often impose physical restrictions on the object, and which are not comfortable for their users. On the other hand, recently, fully image-feature-based motion capturing systems which do not impose such restrictions have been developed as computer vision applications[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 potential 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 and to apply it human-friendly man-machine interface, giving consideration to alleviating scene constraints and physical constraints imposed on the system as little as possible.

There are many researches of human posture/motion analysis, almost all of which employ model-based approaches to analyze complex-shaped objects. Precise analysis of human posture/motion requires sophisticated human models[2, 3, 4], which cause huge computation, and, as a result, their real-time processing becomes almost impossible. To achieve real-time human motion analysis, we have to employ more simplified models and algorithms. In such directions, there are two kinds of approaches: one is blob-based approach[1, 5, 6] and the other is silhouette-based approach[7, 8]. The former approach employs blobs, or coherent color regions, which generally appear in a face and hands in an image, as image features, and the latter employs a silhouette contour of human region. Since color-based image features are not very robust, or they require a color parameter learning phase[9], which is often dependent on target individuals, we employ a silhouette-based approach here. However,

even with a silhouette-based approach, features which can be detected robustly are limited and view dependent, and, thus, we have to develop a mechanism to estimate human full-body posture from the limited number of cues. In addition, to deal with the view dependency and the self-occlusion problem when a human makes various poses, we have employed an approach of multi-view image analysis and have developed a view selection mechanism.

In this paper, we present multi-view-based real-time human motion analysis system and its application to real-time human-machine interface. The key point of our system is that it can estimate human postures from limited perceptual cues such as positions of a head, hands and feet which are stably detected by image analysis. In the viewpoint of human-machine interface, real-time feature is quite important, and, therefore, to realize a real-time system with multiple cameras which produce enormous amount of information, we use a PC-cluster, a set of PCs connected via high-speed network, and acquire quite high performance of image processing.

2. SYSTEM OVERVIEW

2.1 Outline of the Algorithm

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

1. Perception (Detection of cues)
   - Silhouette detection and 2-D feature extraction
   - Calculation of 3-D positions of features using multi-view fusion
2. Human Motion Synthesis
   - Generation of human figure full-body motion and rendering in the virtual space and calculation of the interaction.

In order to analyze various human postures, we arrange multiple cameras so as to capture horizontal views and vertical views of a human body. As we mentioned, to make the system real-time and on-line, we have implemented the system on the PC cluster. Figure 1 shows the configuration of processing modules on the PC cluster, which is a parallel-pipeline structure. Each rectangle in the figure indicates one PC. At first, each view image is processed, or applied to 2D image processing, in a pipelined combination of ICM and HPM (or TPM). This means that the degree of parallelism at this stage is equal to the number of views, or the number of cameras. Then the processed results of all views are integrated and processed at 3DPM (3D image processing) and reconstructed at RRM (real-time rendering) sequentially. Therefore, when we use N views, or N cameras, we need $N+1$ PCs.

Details of the processing modules are as follows:

a) Perception Module:

Image Capturing Module (ICM)
These modules work as image-capturing modules. Each ICM_v ($v = 1, \cdots, N$; N is the number of cameras) captures images with 320×240 pixel (YUV:422 pixel format) and sends them to TPM and HPMs.

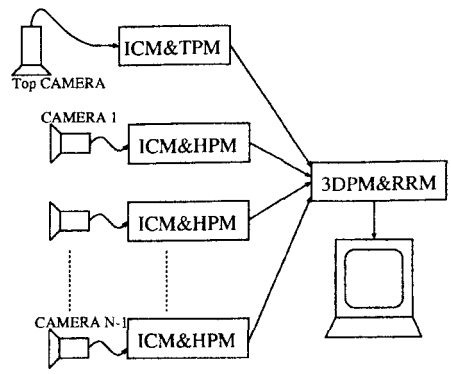


Figure 1: Image processing modules on PC cluster.

Horizontal view Processing Module (HPM)
These modules work as 2-D image processing modules for horizontal-view-cameras. The image processing algorithm contains silhouette detection and 2D feature extraction). Each HPM receives the image data from ICM, and sends 2-D extracted image feature data (positions of detected cues) to 3DPM.

Top view Processing Module (TPM)
This module works as 2-D image processing modules for a vertical-view-camera (or top-view-camera). Since usually the rotation of body around the body axis is not easy to estimate accurately, we have introduced the top view. The role of this module is almost the same as that of HPM, except that image processed here is captured from top view.

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 HPM_v ($v = 1, \cdots, N-1$) and TPM, and estimates 3-D model parameters (3-D positions of cues). 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 cue positions from 3DPM and estimates 3-D pose and motion of the human body based on the received data.

3. 2D IMAGE ANALYSIS

3.1 Feature Extraction Based on Contour Analysis

Here, we use a silhouette contour of a human image for human body posture analysis, because, in general, important features for the human body posture analysis, which are hands, feet, and a head usually appear in the silhouette contour. In addition, silhouette detection can be achieved by relatively simple methods, especially in case of in-house situation, where we assume our system works. An algorithm flow of our feature extraction is as follows:

1. We pre-acquire a background image for each view, in which a human body is not taken.

2. To each frame of an input image sequence, background subtraction is applied, and a human body region is extracted.

3. Coordinates of the centroid (center of gravity) of the extracted human region are calculated.

4. From the calculated centroid, scanning image upward, the system searches for a contour pixel on the extracted region. Then, from the detected contour pixel (we call it "starting point"), the silhouette contour is traced and the 2D coordinates of contour pixels are listed.

The above processing is common to images taken by horizontal-view-cameras and one taken by a vertical-view-camera.

3.2 Analysis for Horizontal-view-camera

Feature points to detect in horizontal-view-camera images are a head top, right/left hands (fingertips), right/left foot points. Except for unusual human body postures, these feature points appear as relatively sharp convex points (hereafter we call these points as "convex points") on the silhouette contour. In general, only from the points with large curvatures, we cannot recognize the correct correspondence between the large curvature points and individual feature points of a human silhouette. Therefore, we have introduced the assumption that in successive image frames shapes of the silhouettes do not change very largely, and the system searches for the feature points in the neighborhood of its corresponding feature point detected in the previous frame. At the initial frame, we assume a standard initial posture, which is a standing posture with arms and legs open like Figure 2, and the order of the feature points appeared on the silhouette contour is given to the system in advance.

The biggest problem of this approach, in which large curvature points are detected from the silhouette contour, is that it is not easy to detect knees and elbows, which are very important features to estimate human postures. This is because they often overlap with other body parts and do not appear in the silhouette contour. To solve this problem, we have introduced multi-view approach and 3D pose estimation based on inverse kinematics and search by reverse projection. Details will be presented in Section 5.

We show some results of 2D feature point detection (Figure 3), where a silhouette contour of human body is presented. Circles in this figure indicate a head top, the centroid of a torso, right/left fingertips, and right/left foot ends. The system can correctly detect feature points even in a crouching pose like (b) and even when the order of feature points on the silhouette contour changes like (c). Of course, when a hand overlaps with a torso, it fails to detect the hand, but, when the hand re-appears, it can re-detect and track the hand again within about 5 frames. It is satisfactory result for single view image analysis of human motion.

3.3 Analysis for Vertical-view-camera

In TPM, the 2D positions of right and left fingertips, those of both shoulders, and the rotation angle of torso around the body axis are estimated. The fingertip positions are estimated by a method similar to 3.2. For estimation of the rotation around the body axis, the orientation of the major axis of the silhouette region is a good cue, and it is calculated from the second order moment and moment of inertia.

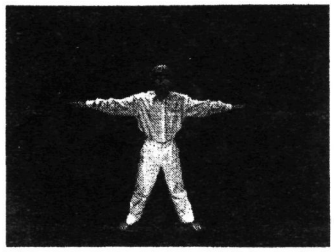


Figure 2: Assumed initial standing position.

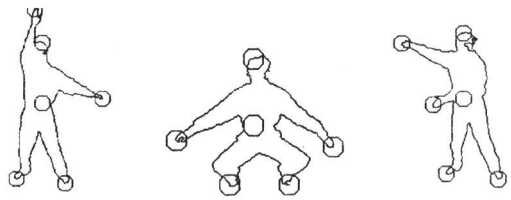


Figure 3: Results of feature point detection.

However, the angle of the major axis of a silhouette region in the top view is often severely affected by configurations of both arms, and, as a result, it is sometimes different from the torso angle. To diminish the influence of the arm configurations, we erase arm parts from the silhouette region by applying a morphology operation, opening, and extract the torso part. The structuring element in this case is very simple, $\{(0,0),(1,0),(-1,0),(0,1),(0,-1)\}$, and we iterate the opening operation seven times. The number of iteration is decided according to preliminary experiments[1].

As for the 2D positions of both shoulders, referring to a pre-defined value of the shoulder width, we simply estimate them from the position of the centroid of the silhouette region and the rotation angle of torso. Figure 4 shows examples of the image analysis results in TPM. Gray regions are extracted torso parts, and a line in each region indicates the rotation angle of torso. Bright dots in the region are the centroid and estimated 2D positions of the shoulders. It shows our image analysis works correctly.

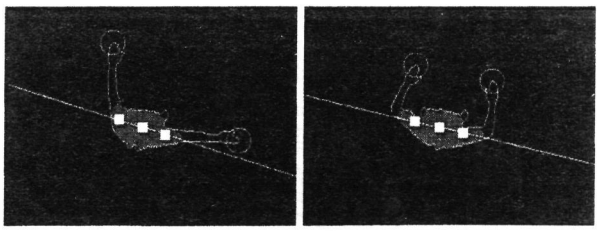


Figure 4: Image analysis in TPM.

[1]The number of iteration should be further considered. Probably, it is better to change the iteration number depending on input images

4. 3D FEATURE ANALYSIS

In principle, when at least two views of an object point are available, we can calculate its 3D position based on the binocular stereo mechanism. However, only two views are not enough to analyze various body posture because of self occlusion. Therefore, in our system, we use multiple, or redundant, views including a top view. In 3DPM, the system dynamically select good views for stereo calculation referring to the result of TPM (described in 3.3). Outline of the algorithm is as follows:

1. Referring to the body rotation acquired in TPM, the system selects two cameras (*front cameras*) facing to the front of the body and one camera (*side camera*) viewing the side of the body. The selection is done referring to inner products of a vector of the body direction and vectors from the centroid of the silhouette region in the top view to cameras (Figure 5). Two cameras giving two maximum inner products are selected as *front cameras* and a camera giving inner product nearest to zero is selected as *side camera*.

2. Two cameras for stereo calculation are selected among two *front cameras*, *side camera* and *top camera*.

   - For positions of a head, a torso and both feet ends, two *front cameras* are selected. The torso position here is calculated from the position of the centroid of the human region.
   - For positions of both fingertips, or fingertips, the system checks whether fingertips appear on silhouette contours in *front camera* views. This is done by calculating, in a top view image, the distance between a fingertip and a line which is parallel to body direction and which is passing through the centroid of a human region. When the distance is larger than the shoulder width, which means that arms are wide open, two *front cameras* are selected. Otherwise, it is judged that a fingertip does not appear on the silhouette contour in *front camera* views, and *top camera* and *side camera* are selected for stereo calculation.

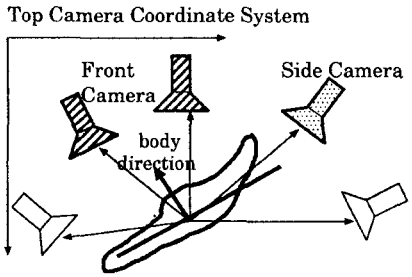


Figure 5: View selection mechanism.

5. 3D POSE ESTIMATION

5.1 Basic concept

Since information acquired in the perception process is just 3-D positions of a torso, a head, hands and feet of a human body, we have to estimate the body posture from these cues, the number of which is less than the degree of freedom of the body. Actually, we have to estimate 3D positions of elbows and knees. In our system, we have adopted two approaches: real-time inverse kinematics (IK) and search by reverse projection (SRP). The former is a general and elegant (and fast!) approach but it can only calculate a suboptimal solution. On the other hand, the latter is rather brute force and searches for the solution referring to original silhouette regions, not to the 3D feature points. The system, at first, estimates the elbow and knee positions using inverse kinematics, and after once the estimation is acquired the system searches for optimal solution referring to the estimated result. In analyzing an image sequence, those positions in a frame are usually searched by the latter approach based on the results in the previous frame. An important point is that when something wrong happens, or the elbow and knee positions can not be detected, because of noise and other reasons, the inverse kinematics is invoked as an error recovery process.

5.2 Inverse Kinematics

In general, estimation of elbow and knee positions is represented in a framework of inverse kinematics[12]. In our case, a human body is represented as a multi-part articulated object, or as 14 parts with 23 degrees of freedom (see Figure 6), and the 3D feature positions (fingertips and foot) are given as the goal positions, or the end effectors. The inverse kinematics which we have designed can be summarized as follows:

- Each arm is represented in a two-link part, and the position of an elbow is acquired by solving inverse kinematics where a fingertip is the goal and a shoulder is the root. The positions of knees can be calculated in a similar way.
- Since rotation angle around the goal direction (see Figure 7), which is called *characteristic angle* hereafter, can not be solved directly, we use a pre-defined value, which is acquired in a learning process. Thus, the inverse kinematics can be reduced into a very simple form, or 2D triangle calculation, and it is calculated very fast, or in real-time,
- The solution gives us continuous and natural-looking motion of the human body.

In our system, a goal position, or a goal vector, $\mathbf{g}^w = (g_x^w, g_y^w, g_z^w, 1)^T$ is represented in the following formula:

$$\begin{aligned}\mathbf{g}_i^w = \; & \mathbf{T}^b\mathbf{R}^b(0, {R_y}^b, {R_x}^b)\mathbf{R}^{b'}({R_z}^{b'}, 0, 0) \\ & \mathbf{T}^{l_1}\mathbf{R}^{l_1}({R_z}^{l_1}, {R_y}^{l_1}, 0)\mathbf{R}^{l'_1}({R_z}^{l'_1}, 0, {R_x}^{l'_1}) \\ & \mathbf{T}^{l_2}\mathbf{R}^{l_2}(0, 0, {R_x}^{l_2})\ \mathbf{t}^e\end{aligned} \quad (1)$$

where $\mathbf{T}^b$, $\mathbf{R}^b$ 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 Figure 6)and link 2 (L2) respectively; $\mathbf{t}^e$ is a translation vector related to the end-effector position of L2. Because it is not very difficult at all, detailed description of analytical solution of the inverse kinematics is omitted here.

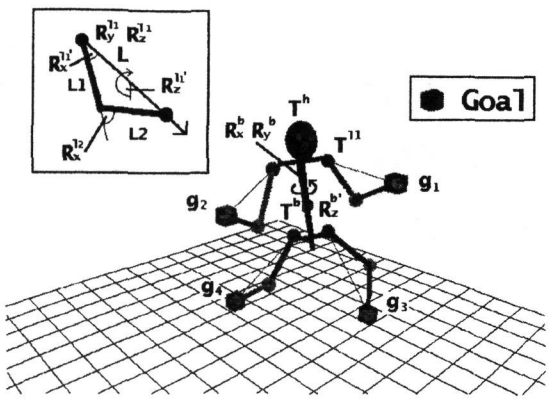


Figure 6: Our human figure model geometry.

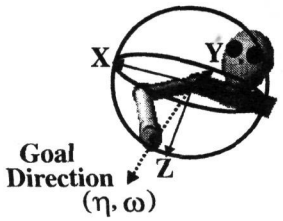


Figure 7: Definition of a goal with an arm direction.

As mentioned above, the 3-D feature 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, 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 Figure 7).

5.3 Search by Reverse Projection

Basic idea of estimation of elbow and knee positions based on *search by reverse projection* is not very difficult. After we estimate the positions of a fingertip and a shoulder for one arm, if we know the lengths of its upper arm and forearm, the position of its elbow is restricted on a circle in 3D space shown in Figure 8, and we only have to search for the elbow position on the circle. When we select a point on the circle and map it onto an image plane of each view, the point should be included in every silhouette region. In other words, points which satisfy the above condition might coincide with the elbow position with high probability.

Suppose that

- the position of the shoulder point is (x_1, y_1, z_1),
- $\vec{A} = (A_x, A_y, A_z)$ is a vector from the shoulder point to the fingertip,
- the length of the upper arm is l_1 and that of the forearm is l_2,

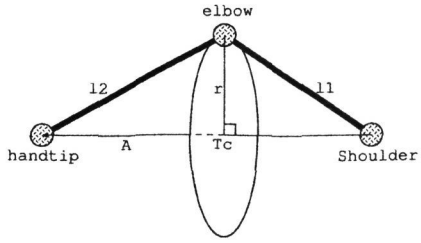


Figure 8: Positional relation among a shoulder, an elbow and a hand.

the radius r and the center T_c of the circle are represented as follows:

$$r = \frac{\sqrt{4|\vec{A}|^2(l_1^2+l_2^2)+4l_1^2l_2^2-(l_1^2+l_2^2+|\vec{A}|^2)^2}}{2|\vec{A}|} \quad (2)$$

$$T_c = (x_1 + A_x t,\ y_1 + A_y t,\ z_1 + A_z t) \quad (3)$$

$$\text{where } t = \frac{\sqrt{l_1^2 - r^2}}{|\vec{A}|^2} .$$

Actual procedure of elbow position estimation is as follows:

1. **Estimation of 3D position of shoulder**
   A z coordinate of the shoulder point can be calculated by adding the height of torso (D in Figure 9), which is a pre-defined value, to the position of the torso (*cf* 4). As for its x and y coordinates, first, a line of sight passing through the shoulder point in the image plane is calculated (dotted line s in Figure 9) based on camera calibration parameters and the 2D position of a shoulder in a top view. Then x, y coordinates of the shoulder position is calculated from its z coordinate and the line of sight.

2. **Setup of local coordinates for elbow position estimation**
   A local coordinate system whose origin is T_c, whose x axis is $\vec{A}$, whose z axis coincides with an axis which is the projection of the z axis of the world coordinate system onto a plane including the search circle. Then, the search circle becomes to lie on the y-z plane of the defined local coordinate system, and, therefore, the calculation of elbow position search can be simplified. The reverse transformation from this local coordinate system into the world coordinate system is very simple as well.

3. **Estimation of elbow position**
   Figure 10 shows the process of elbow position search in the t-th frame. At first, the possible range of the elbow position is established as a neighborhood of the elbow position estimated in the (t-1)-th frame (or by IK) (*cd* in Figure 10). Then, the possible range is reversely projected onto all the image planes, and the possible range which is inside of all of the silhouette regions becomes the final possible range of the elbow position (*ab* in Figure 10). Generally, the final possible range

does not become a single point, and, therefore, the center of the possible range is decided as the estimated elbow position.

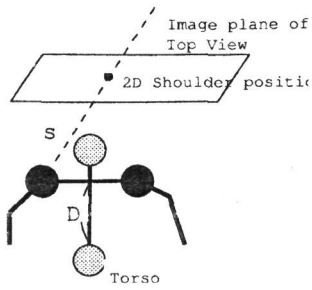


Figure 9: Shoulder position in the world coordinates.

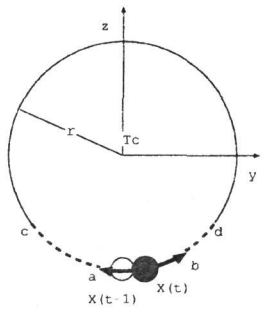


Figure 10: Estimation of an elbow position.

Figure 11 shows difference between analysis result based on IK (b) and on SRP (c). As shown here, although IK gives a result with relatively large error because its characteristic angle is not precise, SRP gives more accurate result.

5.4 Estimation of Torso Posture

Torso posture consists of two elements, the axis of the torso and the pan angle around the axis. The pan angle can be estimated by moment analysis of the silhouette region in top view (3.3). The axis of the torso is an axis connecting the centroids of a head part and a torso part and is defined as follows:

$$R_x{}^b = -\arcsin \frac{T_y{}^h - T_y{}^b}{||\mathbf{T}^h - \mathbf{T}^b||} \quad (4)$$

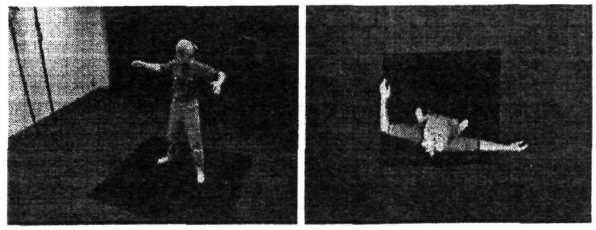
$$R_y{}^b = -\arctan \frac{T_x{}^h - T_x{}^b}{T_z{}^h - T_z{}^b} \quad (5)$$

6. EXPERIMENTS

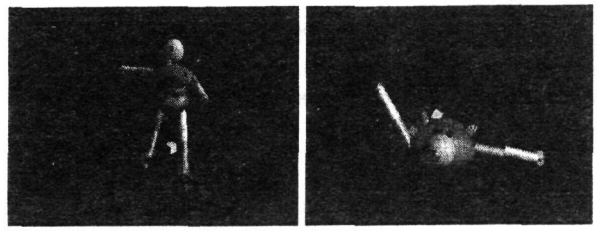
6.1 Basic Experimental Results

In this experiment, we have used 6 IEEE1394-based color cameras (Sony DFW-V500) with f:4mm lenses, all of which are synchronized by a common external trigger. Five of them are horizontal-view-cameras and arranged circularly at intervals of about 30 degrees, and the rest is a vertical-view-camera (Figure 6.1). These cameras are geometrically calibrated in advance[2]. The images are captured with the size of 320 × 240 pixels, and the frame rate is 15 fps[3]. The number of PC's in the system is seven, each of which has dual PentiumIII's(700MHz) running Linux.

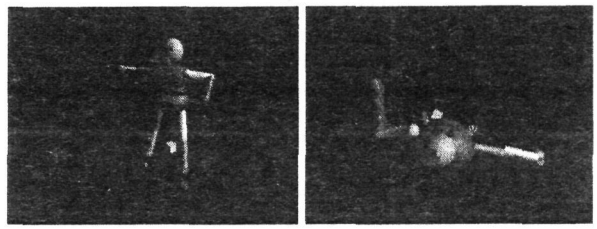
The goal of our method is that, referring to multiple, or redundant, view images, the system can analyze human postures in such situations that a hand overlaps with a torso part, which cannot be analyzed by usual silhouette analysis. To illustrate this effect of multi-view image analysis, we analyzed human motion in which a left hand swings in a arc from the left side of the body to the front of the body.



(a) Input images.



(b) Estimation results by the original IK.



(c) Estimation results by our method.

Figure 11: Difference between the results of the original IK and those of our new method.

[2]Camera calibration is accomplished based on Tsai's method[13].

[3]When an external trigger is given to our cameras, the maximum frame rate becomes 15fps, not 30fps. However, the potential performance of the system can be 30fps, normal video rate.

Figure 13 shows the analysis result, which illustrates the human motion reconstructed by CG technique from view 2. From the top, they are estimated result at frames 0, 23, 42, 68, 92. This shows that our proposed method can analyze the human postures, by selecting adequate views, in such situations that a body part overlaps with other body parts. However, the problem of estimation accuracy still remains because of the error of the body rotation estimation in top view images.

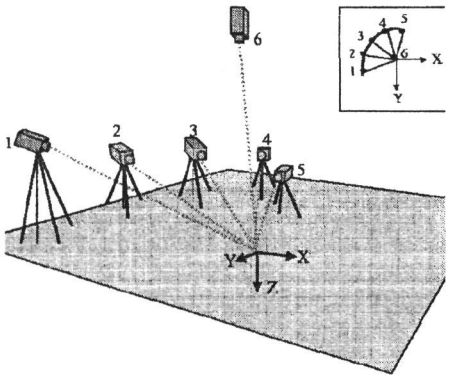


Figure 12: Multiple camera configuration.

In this method the system can not estimate very complex postures, since the human model is very simple:

- Twist of the neck can not be detected. However, it can be detected after we introduce face detection to the system.
- Postures in which hands or feet can not be detected are almost impossible to estimate.

In spite of these problems, we can apply this system to various interactive applications.

6.2 Real-time Interaction between Human and Virtual Environment

Here, the real-time human motion analysis mentioned above is applied to *Visually Guided 3D Animation*, or a real-time online interaction system in a virtual space. Figure 14 shows demo shots of a prototypical system, in which a user visualized as an *avatar* in the virtual space kicks a ball in the virtual space. In this case, reaction in the virtual space is calculated based on simple physical collision detection between the body parts of the avatar and the ball and on simple physical characteristics.

The performance of the system is summarized as follows:

- For each frame, about 20 msec is required for HPM and about 40 msec for TMP. The size of each frame is 320 × 240 pixels.
- 35 msec is required for 3DPM and RRM.
- The delay introduced is about 0.1 second because of the latency of the pipelined implementation.

As shown above, the system can work completely in real-time, i.e., 15fps. Because of the delay introduced to the system, in the demo shots the avatar postures are slightly different from those of the user.

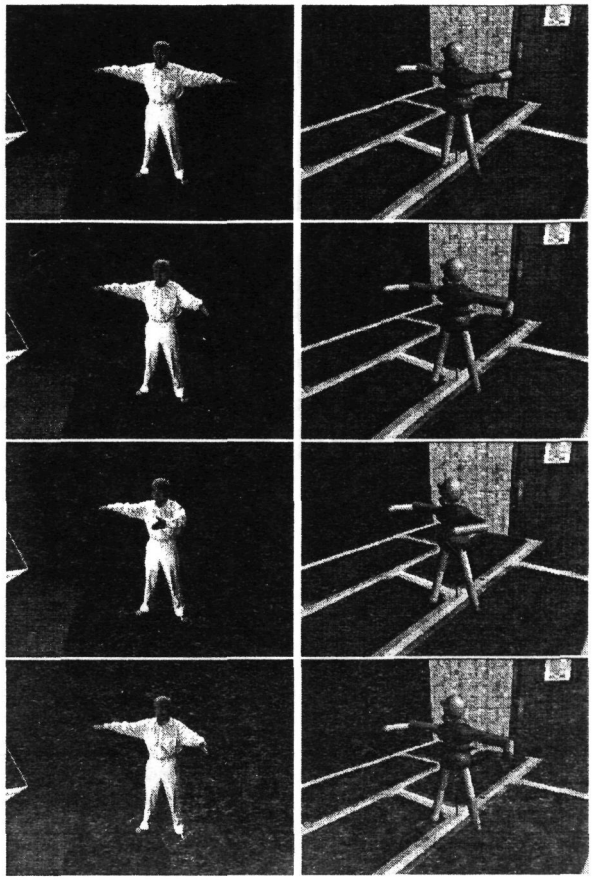


Figure 13: Results when a hand overlaps with a torso.

6.3 Application to Desktop Interface

Our idea can be also applied to desktop interface with little modification. In case of desktop interface, the system has to only analyze upper body motion, and the body posture is rather restricted compared with full-body motion. Therefore, the system can be simpler, i.e., the number of cameras required for analysis can be reduced. In practice, two cameras are usually enough for the analysis. Thanks to the recent speed-up of micro processors and peripherals, we can construct a compact system. Actually, we have constructed a prototypical desktop interface system using one PC which has two PentiumIII's (700MHz) and two IEEE1394-based cameras[14]. The two cameras are connected to the PC via one IEEE-1394 bus, which can transfer video images of two cameras simultaneously. Figure 15 shows a demo shot of the prototypical desktop interface system, where a user indicates his instruction by his gesture.

7. CONCLUSIONS

In this paper, we have shown a real-time human motion capturing without special marker-sensors and its application to real-time human-machine interaction. For real-time motion analysis, we have adopted silhouette-based multi-view fusion to realize full-body motion analysis. The key point

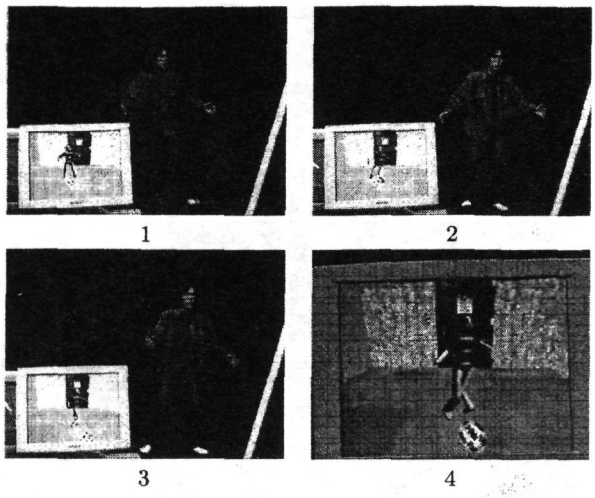


Figure 14: Snapshot of an online demonstration.

Figure 15: Snapshot of a desktop interface.

is that we have established a framework of estimation of full-body motion from a limited number of perceptual cues, which can be stably extracted from input images. Since the system implemented on PC-cluster works in real-time and online, it can be applied to various *real-virtual* applications. In our system, we refer to pre-defined values with relation to the sizes of several body parameters, such as shoulder width. We are constructing a system calculating those pre-defined values from images, where a certain kind of posture is made. This, as an initial setup stage, will be integrated into the system. We will also improve each image analysis algorithm to make more robust human motion analysis possible, which leads our real-time human motion analysis more applicable. In future work, we will investigate more practical applications of human-machine interface based on real-time human motion analysis.

8. ACKNOWLEDGMENTS

This work has been partly supported by "Intelligent Media Technology for Supporting Natural Communication between People" project (13GS0003, Grant-in-Aid for Creative Scientific Research, the Japan Society for the Promotion of Science).

9. ADDITIONAL AUTHORS

Additional authors: Ryuya Hoshino (Department of Intelligent Systems, Kyushu University, email: hoshino@limu.is.kyushu-u.ac.jp).

10. REFERENCES

[1] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (19)7:780–785, 1997.

[2] S. Yonemoto, N. Tsuruta, and R. Taniguchi. Tracking of 3D multi-part objects using multiple viewpoint time-varying sequences. In *Proc. Int. Conf. Pattern Recognition*, pages 490–494, 1998.

[3] T. Nunomaki, S. Yonemoto, D. Arita, R. Taniguchi, and N. Tsuruta. Multi-part non-rigid object tracking based on time model-space gradients. In *Proc. Int. Workshop Articulated Motion and Deformable Objects*, pages 720–82, 2002.

[4] L. Herda, P. Fua, R. Plankers, R. Boulic, and D. Thalmann. Skeleton-Based Motion Capture for Robust Reconstruction of Human Motion. In *Proc. Computer Animation*, pages 77–83, 2000.

[5] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. Computer Vision and Pattern Recognition*, pages 568–574, 1997.

[6] M. Etoh and Y. Shirai. Segmentation and 2D motion estimation by region fragments. In *Proc. Int. Conf. Computer Vision*, pages 192–199, 1993.

[7] M. K. Leung and Y-H. Yang. First sight: a human body outline labeling system. *IEEE Trans. Pattern Analysis and Machine Intelligence*, (17)4:359–377, 1995.

[8] K. Takahashi, T. Sakaguchi, and J. Ohya. Remarks on a real-time 3D human body posture estimation method using trinocular images. In *Proc. Int. Conf. Pattern Recognition*, Vol.4, pages 693–697, 2000.

[9] Y. Okamoto, R. Cipolla, H. Kazama, and Y. Kuno. Human interface system using qualitative visual motion interpretation. *Trans. Institute of Electronics, Information and Communication Engineers*, (J76-D-II)8:1813–1821, 1993 (in Japanese).

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

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

[12] J. Zhao and N. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *Trans. Computer Graphics*, (13)4:313–336, 1994.

[13] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Trans. on Robotics and Automation*, (3)4:323–344, 1987.

[14] S. Yonemoto and R. Taniguchi. High-level human figure action control for vision-based real-time interaction. In *Proc. Asian Conf. Computer Vision*, pages 400–405, 2002.