Human action sensing for proactive human interface: Computer vision approach

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

Arita, Daisaku
Department of Intelligent Systems, Kyushu University

Uchida, Seiichi
Department of Intelligent Systems, Kyushu University

Kurazume, Ryo
Department of Intelligent Systems, Kyushu University

他

出版情報：Proceedings of Workshop on Processing Sensory Information for Proactive Systems, pp.23-29, 2004-06
バージョン：
権利関係：

HUMAN ACTION SENSING FOR PROACTIVE HUMAN INTERFACE: COMPUTER VISION APPROACH

Rin-ichiro Taniguchi, Daisaku Arita, Seiichi Uchida, Ryo Kurazume and Tsutomu Hasegawa

Department of Intelligent Systems, Kyushu University, Japan
rin@limu.is.kyushu-u.ac.jp

ABSTRACT

In this paper, we discuss human action sensing for proactive human interface. Proactive human interface is an idea of advanced interface based on perceptual user interface, and provides easy and natural interaction between humans and systems based on observing human activity through many modalities including visual, audio, tactile information etc. "Proactive" here means that a system understands a human's intention and that reduces tiresome interactions by planning the interactions based on the acquired intention. In this paper, as a first step, we present human action sensing based on a computer vision approach. Vision-based approaches have a strong merit that any physical restrictions are not imposed on humans, and provide a natural way of measuring human activity.

1. INTRODUCTION

Seamless interaction between humans and systems is of importance in many aspects. Current human interface still have several problems to be solved. We think the following two points are the most important.

- Since communication modalities between humans and the systems are rather restricted, many human activities through computer systems are virtualized, which leads to the dissociation of the real world, where the humans perform their various activities, and the virtual world, where demanded tasks are executed. The larger the dissociation becomes, the larger the difficulty of interaction becomes. To solve this problem, we have to augment the communication modalities: both of virtualization of activities in the real world and presentation of virtualized activities through a variety of modalities are quite important.

- In current computer-based systems, we have to provide a detailed list of commands or demands to the computers either interactively or non-interactively. However, when the dissociation of virtual and real worlds mentioned above is large, providing the detailed list of demands becomes very difficult. If the systems understand human's intention of actions, the detailed list of demands can be planned and optimized by the systems, which drastically decreases the overhead of interaction between humans and the systems.

Considering these problems, we have just started a project "Embodied Proactive Human Interface," sponsored by SCOPE[1]. In this project, we have established the following three research topics.

Presentation of virtualized activities to humans
Especially, a method to interact through embodied agents such as robots should be investigated. Embodied agents can provide more natural interaction modalities.

Sensing of human activities
Human sensing with non-contact device should be investigated, which provides a natural way of human action sensing. In addition, we also investigate human activity sensing through interaction with the embodied agents.

Understanding of human intentions
It is quite important in order to realize proactive human interface. The key issue here is a leaning mechanism of intentions from a set of observed actions in the real world.

In this paper, we pick up the second topic, as a first step of our proactive human interface project.

2. HUMAN ACTION SENSING

A vision-based approach for human action sensing is smart and natural since it does not impose any physical restrictions on a user. However, it has several problems to be solved.

From the viewpoint of human interface, a real-time feature is quite important and, therefore, computation intensive

[1]Strategic Information and Communications R&D Promotion Programme by the Ministry of Public Management, Home Affairs, Posts and Telecommunications in Japan.

approaches[1, 2] is not realistic even though they provide a general framework. Real-time here means that images are processed at the speed of TV camera signal, .i.e., $20 \sim 30$ frames/sec. To realize such real-time systems, the key issues are as follows:

- robust image features, which are easy to extract
- fast human posture estimation from the image features

Usually, as image features, blobs (coherent region)[3][4] or silhouette contours[5] are employed. However, image features which can be robustly detected are limited, and, therefore, the estimation of 3D human postures from the limited cues are quite essential. To solve this problem, we have introduced vision-based inverse kinematics. 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.

3. SYSTEM OVERVIEW

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

1. Detection of visual cues
   - Silhouette detection, skin color blob detection, face direction detection.
   - 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 including calculation of the interaction.

The key point of our system is *human motion synthesis* referring to a limited number of visual cues, which are acquired by the perception process. Several real-time vision-based human motion sensing have been developed. However, they are based on a rather simple human model and its generated human motion sensing is not natural. For example, the direction of face is not detected, or the number of articulations is limited. Here, we make the human figure model more complex and develop a vision- based algorithm for human motion sensing based on the model (details are discussed in 5).

When a real-time and on-line system is designed, an error recovery mechanism is quite important. If the system is based on feature tracking, once features fail to be tracked, the posture estimation process may reproduce unrealistic human postures, and it can not escape from the erroneous situation. Therefore, in order that we need not reset the system in such a case, we have introduced a simple error recovery process, which executes concurrently with the main human posture estimation process, and which always checks whether the human silhouette makes predefined shapes which are easy to recognize precisely. When the process finds the silhouette makes such shapes, it notifies the human posture estimation process, and the estimation process adopts the recognition result regardless of the estimation result. According to these considerations, we have designed a vision-based real-time human motion sensing as shown in Fig.1.

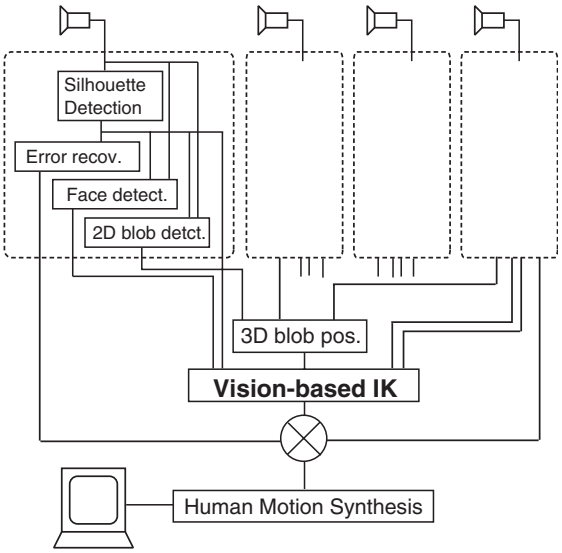


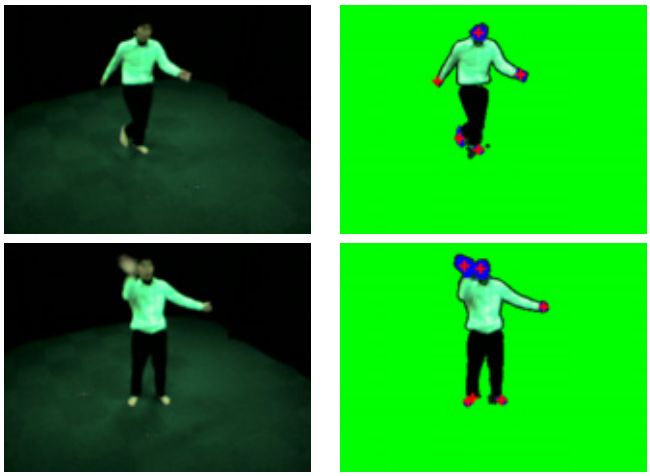
Fig. 1. Software structure of real-time vision-based human motion sensing.

4. ACQUISITION OF PERCEPTUAL CUES

4.1. Color blob detection

We detect color blobs for a head, hands and feet, whose colors are acquired in advance, and calculate 2D positions of the color blobs as the centroids of the blobs. Color blob detection is based on color similarity evaluation, which is done in HUV color space to exclude the influence of lighting condition as much as possible. We also extract a human silhouette region by background subtraction, which is used by an error recovery process and a human posture estimation process.

In principle, the 3D positions of the blobs can be calculated from two views. However, due to self-occlusion, we can not always calculate the 3D positions with only two views, and we have used a certain number of cameras. Therefore, we have to solve a stereo-pair correspondence problem in a multi-view situation. We, first, extract possible stereo-pair candidates of the 2D blobs. When the distance of the



Input image Detected blob

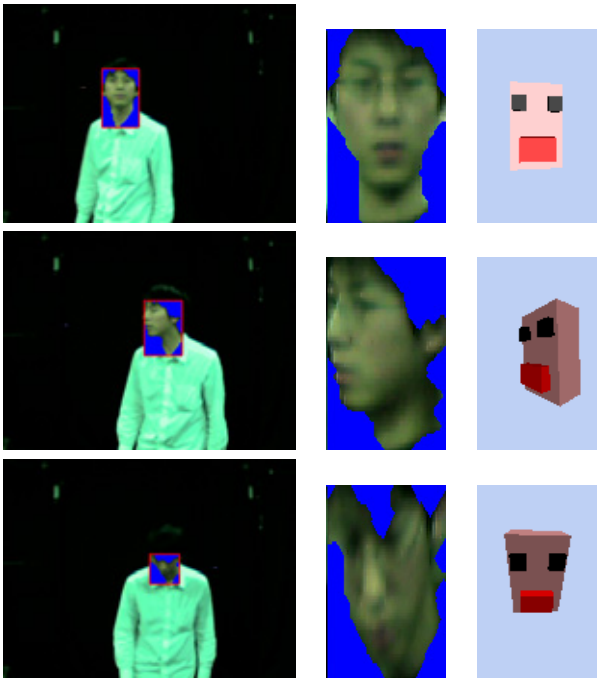
Fig. 2. Color blob direction

two lines of sights passing through the centroids of two 2D blobs is small, the two blobs are a stereo-pair candidate. In this case, a point, which is the nearest to the both lines in a sense of the least square error, is judged to be their 3D point. Then, we classify their 3D positions into 5 clusters of feature points: head, right hand, left hand, right foot, left foot. Classification is done based on the distances from the feature points detected in the previous frame. In each cluster, we estimate the feature point position as the average position of the 3D position candidates after a dense part of the cluster is selected.

4.2. Face direction

Face direction is an important feature in human posture and is indispensable for interactive application. The problems here is the low resolution of face region, because cameras are arranged to capture a full-body and a face becomes small in the field of view. Therefore, face features such as eyes and mouths can not be clearly detected, and, then, feature-based or structure-based techniques of face direction estimation such as [6] can not be applied. We, here, have employed a template matching method preparing face templates with multiple aspects.

Currently, we prepare 300 or more templates for each person in advance. Making templates is quite easy: 3D rotation sensor is attached to top of the user's head and the user makes a variety of face directions in front of a camera before using the motion sensing system. Recording the output of the rotation sensor and the face image frames synchronously, we can get face templates with a variety of face directions. It takes only a couple of minutes. This approach is not sophisticated but quite practical, i.e., it is very simple



Input image Matched template Estimated direction

Fig. 3. Estimation of face direction

but quite easy to use and robust.

To reduce the computation time, we have employed an eigen-space method[7] with several speed-up tactics[8]. The size of templates is 80×130, and the dimension of the eigen-space is 60. After detecting a face region, or skin-color region which corresponds to a face, the face region is normalized to have the same size as that of the template. Fig.3 shows an example of the face direction estimation. The estimation accuracy is not very high because of the low resolution images, but in most of interactive applications, we can get certain feedbacks and can modify the face direction based on them. We think the accuracy acquired by this approach is high enough to use.

4.3. Silhouette shape recognition

As mentioned before, a silhouette shape recognition process is executed in parallel to the human posture estimation process, to make error recovery possible. The important point is that the shape recognition algorithms is frame independent, or it is not history sensitive. It supplies a two kinds of information: "recognized" or "not recognized." When it recognizes a pre-defined shape, the posture parameters associated to the shape are used instead of the result of the posture estimation process. The feature detection is restarted based on the associated posture parameters.

Fig. 4. Error resetting posture

The recognition algorithm is quite simple.

- Boundary pixels of a silhouette region are listed.
- The list of boundary pixels is re-sampled so as that the length of the list becomes the same as that of the template.
- The distance of the detected silhouette and the template is calculated based on the following formula:

$$D = \sum dist(\mathbf{d}_i, \mathbf{d}_i^0)$$

  where $\mathbf{d}_i$ is the 2D position of the i-th boundary pixel of the silhouette region and $\mathbf{d}_i^0$ is that of the template.
- When the distance is small enough, the detected silhouette is recognized to be the same as the template.

Currently, a posture shown in Fig.4 is used for the error recovery operation, whose recognition can be achieved with high precision.

5. VISION-BASED INVERSE KINEMATICS

Our problem is to estimate human postures from a limited number of perceptual cues, which are blobs corresponding to hands, feet and head. This problem can be explained in a framework of Inverse Kinematics (IK) in the field of robotics. IK is to determine joint angles $\theta_\mathbf{i}$ of a manipulator so that the position of an end effector, or a final point, $\mathbf{P}_n$, coincides with a given goal point $\mathbf{G}$: $\mathbf{P}_n(\theta_1, ..., \theta_n) = \mathbf{G}$: where the manipulator has n segments. The difficulty here is that even if the goal is attainable[2], there may be multiple solutions and, thus, the inverse problem is generally ill-posed.

In our problem, end effectors are hands, feet and a head, and the goals are the blob positions acquired by the perceptual process. The posture estimation, which is to decide the positions of joints of the human model, is achieved by calculating the joint angles in the frame work of IK. In human posture sensing, each joint position acquired by IK should be coincide with a joint position of a given human posture, and, therefore, we have to find the unique and correct solution.

Our method to solve this problem is divided into two phases: *acquisition of initial solution* and *refinement of initial solution* For simplicity, here, we explain human posture estimation of a upper body.

5.1. Acquisition of initial solution

Inverse Kinematics is solved by an ordinary numerical method[9] and initial candidates of 3D positions of shoulders and elbows are calculated. Here, we assume that the lengths of the bones in Fig.5 are given in advance. At time t, a hand position $(x(t), y(t), z(t))$ is represented as

$$(x, y, z) = \mathbf{P}(T_x(t), T_y(t), T_z(t), \theta_1(t), \theta_2(t), \cdots, \theta_N(t)) \quad (1)$$

where

- $T_x(t), T_y(t), T_z(t)$ indicate the head position in the world coordinate,
- $\theta_1(t), \theta_2(t), \theta_3(t)$ indicate rotation angles between the world coordinate and the local coordinate of the head, which are calculated from the face direction,
- $\theta_j(t)(4 \leq j \leq N(= 8))$ indicate rotation angles among connected parts .

We suppose that, at time $t + 1$, the hand position moves to $(x(t + 1), y(t + 1), z(t + 1))$, the head position moves to $(T_x(t + 1), T_y(t + 1), T_z(t + 1))$, and the head direction changes to $(\theta_1(t+1), \theta_2(t+1), \theta_3(t+1))$. Here, we slightly modify $\theta_j(t+1)$, $(4 \leq j \leq N)$ so as that the hand position, i.e., the position of the end effector, $\mathbf{P}(T_x(t + 1), T_y(t + 1), T_z(t+1), \theta_1(t+1), \cdots, \theta_N(t+1))$ approaches the goal position $(x(t+1), y(t+1), z(t+1))$. Repeating this process until the end effector position coincides with the goal position, we acquire the positions of a shoulder and an elbow. In order to exclude impossible postures, we have imposed a possible range on each angle θ_j.

5.2. Refinement of initial solution

The posture estimated in the previous step is just a solution of inverse kinematics, and it is not guaranteed that it coincides with the actual posture. This is due to ill-poseness of the inverse kinematics. To estimate the posture more accurately, we refine the acquired solution by referring to input image data. The basic idea is simple: if the shoulder and elbow positions acquired by the previous phase are correct, they should be inside of the human region in 3D space. Otherwise, the acquired solutions are not correct and they should be modified so as to be included in the human

[2] If the distance of the goal to the initial point of the manipulator is larger than the sum of the lengths of the segments, the goal is not attainable.

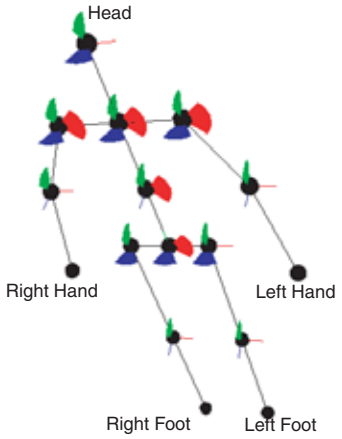


Fig. 5. Human model

region. Here, we empirically assume that the shoulder position is acquired by solving the basic inverse kinematics, and we mainly refine the elbow position. Its basic algorithm is as follows:

- We have the shoulder position by solving the inverse kinematics and the hand position by color-blob analysis.
- When the lengths of its upper arm and forearm are given, the position of its elbow is restricted on a circle in 3D space. The circle is indicated by C.
- When the elbow is searched on the circle, we exclude impossible values, with which the arm get stuck in the torso.
- As shown in Fig.6, an elbow detection rectangle is established in a plane which is constructed by the shoulder, an hypothesized elbow and the hand.
- Then, in each view, the rectangle is reversely projected on the image plane and correlation between the projected rectangle and the human silhouette region is calculated.
- Then, by varying the position of the hypothesized elbow, the correlation R can be parameterized by ϕ, which is the angle around the center of the circle C. We search for ϕ giving the maximum R, which indicates the elbow position.

If the refinement process fails to find the correct solutions, the system restarts the modification process by changing the initial values. Since this system is required to work in real-time, this iteration is stopped when the deadline comes, and intermediate result is returned.

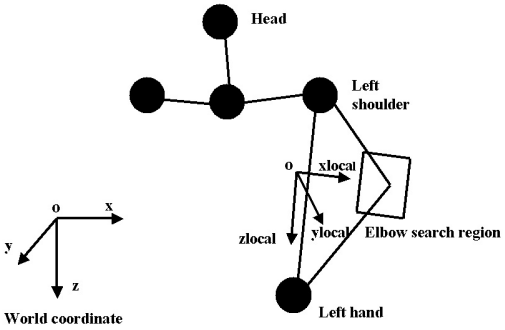
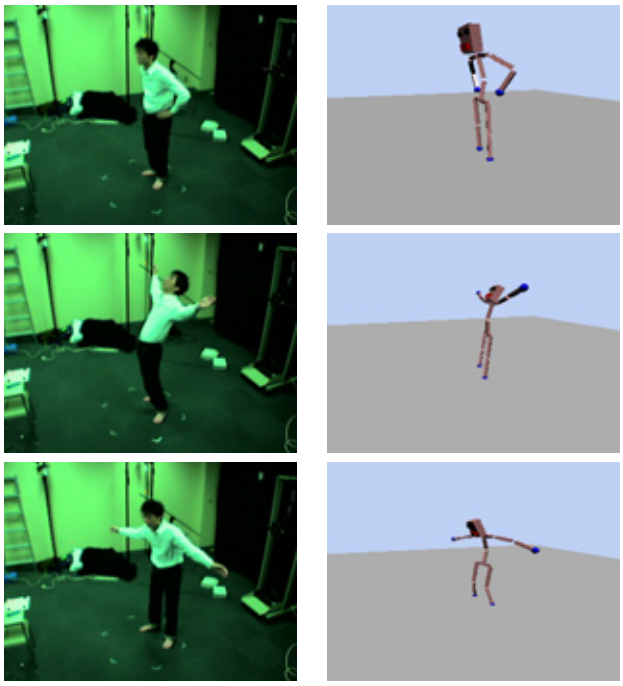


Fig. 6. Elbow position estimation



Actual posture Estimation result

Fig. 7. Result of human posture estimation

6. EXPERIMENTAL STUDY

In this experiment, we have used 9 sets of IEEE1394-based color cameras (Sony DFW-V500) with f:4mm lenses, which are geometrically calibrated in advance. The images are captured with the size of 640×480 pixels, and the frame rate is 15 fps[3].

We have implemented our vision-based human motion analysis on PC-cluster with 3.2 GHz PentiumIVs, where detection of perceptual cues in each view is executed in parallel, and where solving IK and human figure generation are

[3]due to camera specification

Vision algorithm	Time
2D blob detection:	4ms
3D blob calculation:	5ms
face direction estimation:	15ms
IK initial solution:	3ms
IK solution refinement:	27ms
template matching for error recovery:	11ms

Table 1. Computation time

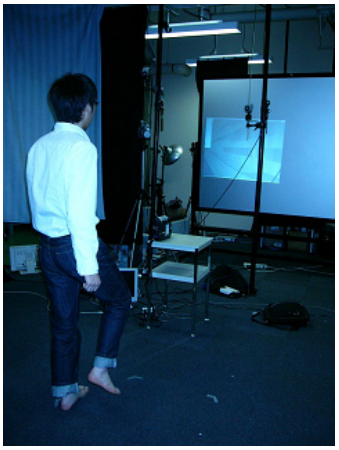


Fig. 8. VR walk-through evnironment

executed in a succeeding PC. Computation times required in major vision algorithms of our human motion analysis are shown in Table 1, which indicates that real-time processing of our vision-based human motion sensing can be achieved on the PC-cluster.

Fig.7 shows a typical example of human action sensing. In this system, face direction is estimated, the shoulder positions are slightly shifted according to the hand positions, and the upper body bends forward slightly, which can be achieved only with our complex human figure model.

Using this system, we are developing several interactive systems including VR walk through, VR ball playing, etc. Since this system works completely in real-time, we expect it can be applied to many interaction systems.

The biggest problem of this system is the latency of the system. Currently, the latency is about 200 msec and it is a bit large for smooth interaction. To reduce the latency, we have been developing an estimation-based latency reduction mechanism, where we can dynamically control the tradeoff between the latency and the accuracy of the system[10].

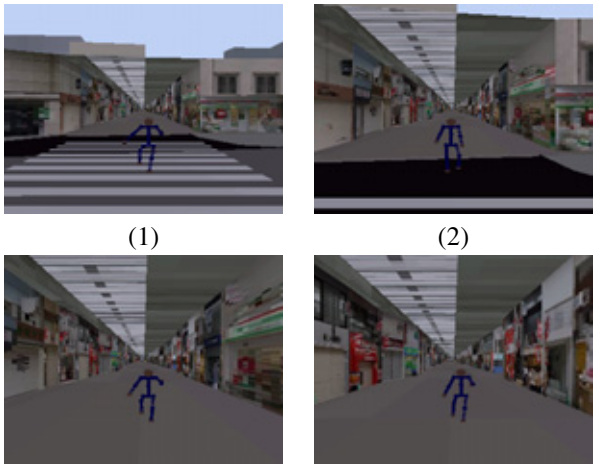


Fig. 9. Generated images of VR walk-through

7. CONCLUSIONS

In this paper, we have shown a vision-based human action sensing for proactive human interface. The vision-based approach has a merit that it does not impose any physical restrictions of a target human and that it provides a natural way of human action sensing. The key point 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 interactive applications. For future works, we are going to apply this system to a sensing system of our proactive human interface. To accomplish this goal, we have to develop a mechanism to understand the user's intention from the result of the human action sensing. Of course, a cooperative mechanism among different communication modalities are very important.

ACKNOWLEDGEMENT

This work was supported in part by "Embodied Proactive Human Interface," the Ministry of Public Management, Home Affairs, Posts and Telecommunications in Japan under Strategic Information and Communications R&D Promotion Programme (SCOPE).

REFERENCES

[1] T.Nunomaki, et al, "Multi-part Non-rigid Object Tracking Based on Time Model-Space Gradients," *Proc. Workshop on Articulated Motion and Deformable Objects*, pp.72-82, 2000.

[2] J.Deutshcher, et al, "Automatic Partitioning of High Dimensional Search Spaces Associated with Articulated Mody Motion Caputure," *Proc. CVPR*, Vol.2, pp.669-676, 2001.

[3] C.Wren, et al, "Pfinder: Real-Time Tracking of the Human Body", *IEEE Trans. PAMI*, Vol.19, No.7, pp.780-785, 1997.

[4] M.Etoh, et al, "Segmentation and 2D Motion Estimation by Region Fragments", *Proc. ICCV*, pp.192–199, 1993.

[5] K.Takahashi, et al, "Remarks on a Real-Time 3D Human Body Posture Estimation Method using Trinocular Images," *Proc. ICPR*, Vol.4, pp.693–697, 2000.

[6] P.Yao, et al, "Face Tracking and Pose Estimation using Affine Motion Parameters", *Proc. SCIA*, pp.531-536, 2001.

[7] H.Murase, et al, "Visual Learning and Recognition of 3-D Objects from Appearance," *International Journal of Computer Vision*, Vol.14, No.1, pp.5-24, 1995.

[8] S.A.Nene, et al, "A Simple Algorithm for Nearest Neighbor Search in High Dimensions," *IEEE Trans. PAMI*, Vol.17, No.9, pp.989-1003, 1997.

[9] L.Wang, et al, "A Combined Optimization Method for Solving the Inverse Kinematics problem of Mechanical Manipulators", *IEEE Trans. Robotics and Automation*, Vol.7, No.4, pp.489-499, 1991.

[10] H.Yoshimoto, et al, "Vision-based Real-time Motion Capture System Using Multiple Cameras", *Proc. IEEE Int. Conf. Multisensor Fusion and Integration for Intelligent Systems*, pp.247-251, 2003.