A 3D Animation System with Superimposing CG on a Physical Armature

Yuichi Ando, Shin Takahashi, Etsuya Shibayama

Graduate School of Information Science and Engineering,
Tokyo Institute of Technology

ABSTRACT

This paper proposes a new technique to support making 3D character animations with DID (Dinosaur or Direct Input Device). The key idea is to superimpose the image of a CG model on a physical armature using video see-through Head-Mounted Display (HMD). Using this technique, animators can easily recreate the posture of the physical armature. In addition, the system shows candidates of postures for animators who are not familiar with target motions. We have implemented a prototype system for making character animations using this technique.

Keywords: 3D animation, Keyframe, Head-Mounted Display, Physical Armature

1 INTRODUCTION

We aim to design the animation system which is easy to use for beginners of animator. There is a technique that uses an armature for input device [9]. This technique enables the user to pose characters in a keyframe using a physical armature which is covered with sensors or markers that monitor its posture. The user can intuitively specify the posture of characters as compared to common keyframe animation systems. In addition, comparing to motion-capture systems, it does not need a large-scale system, and it can be applied for recording motions of animals and virtual creatures.

However, this technique has the following problems:

1. When the user manipulates a character in a keyframe with an armature, he/she refers the positions of other objects (the background, other characters, etc.) in a virtual world, which he/she cannot see in the real world. Therefore, the user has to move his/her gaze between the armature and the screen that displays the virtual world.

2. It is difficult to recreate already specified postures of characters with an armature, because they can be viewed only in the virtual world. In order to view and modify already specified animation data, users must use keyframe animation tools.

To cope with this problem, we propose a new technique to support specifying character animations where the image of the target character is superimposed on a physical armature using video see-through Head-Mounted Display (HMD). The user wearing HMD can see the armature and the image of a CG model at the same position. Therefore, the user does not need to move his gaze between the armature in the real environment and the screen which displays virtual worlds. It is a kind of Mixed Reality (MR) which is a technique that seamlessly mixes the virtual CG images made by computer and the real environments. Recreating postures is achieved by posing the physical armature so that it is superimposed on the corresponding character model displayed on the HMD screen. Moreover, our technique makes it possible to show various candidates of postures for users who are not familiar with the target motions.

Figure 1 illustrates the overview of our system. The user poses an armature to make a keyframe with looking at the posture candidates displayed on HMD. When the user finished posing the armature, the system captures its 3D posture and position which are used for making keyframes. After the input of all keyframes, the system interpolates intermediate frames to make the entire animation. When the user wants to edit already specified keyframes, the user must firstly manipulate the armature to recreate the posture. This is also achieved by manipulating the armature to superimpose it on the CG model in the user's view. Then, the user modifies its pose and makes the system to capture it again. Note that it is also possible to change only the part of the posture of the model, e.g. change only the posture of the right arm.

We have implemented a prototype system for making character animations with this technique. In our system, the user wears an HMD where two CCD cameras are attached. With looking the screen of the HMD, the user manipulates an armature to express its posture in each keyframe (Figure 2). We also evaluated the effectiveness of our approach with simple user tests. The preliminary results have shown that our technique is effective for creating animations.

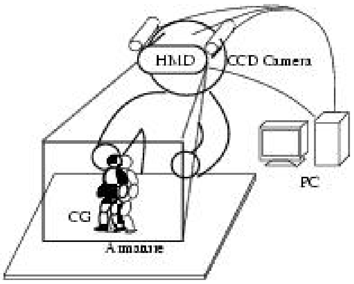
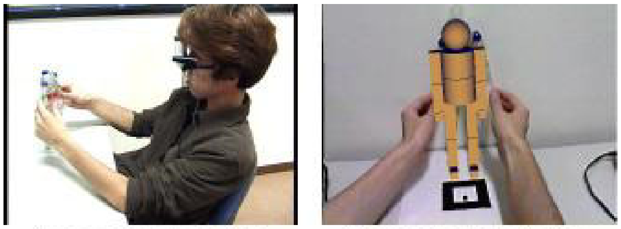


Figure 1: Overview of the System


A user using our system

The image of user's view

Figure 2: Superimposing CG on Armature

2 Background

2.1 Character Animation

In this section, we survey major techniques of creating character animations. The key issue of this research topic is to find easy and intuitive ways of expressing the posture of multi-articulated objects.

Motion capture is a technique that captures the movement of human body with markers or position sensors attached on his/her joints. It can record a real live performance, but the system is very costly, and needs a large space. In addition, it is difficult to record the motion of subjects to whom it is hard to attach sensors. For example, it is difficult to capture the movement of animals, virtual creatures, or people swimming in water.

Keyframe animation is the most common technique for making 3D animations. To make a character animation, the animator specifies model's postures and positions in keyframes at intervals of a few frames, and the system interpolates middle frames automatically. Commercial tools are widely available for this purpose, such as LightWave[*], ANIMATION MASTER[†]. In addition, there are several techniques that support animators to easily control the model's posture, such as Inverse Kinematics. The work by Laszlo et al. [10] designed mapping from input DOF to output DOF. Using this technique, animators can control motion effectively and intuitively. Nevertheless, keyframe animation requires animator's significant learning and efforts. It is hard for beginners to create animations with keyframe animation tools.

PhysiAnime[‡] and Character studio[§] automatically generate a motion, such as walking and running, which satisfies some constraints. For example, the animator can specify the positions of character’ s footprints to make an animation of a person walking on the footprints. Using this system, animators can create walking or running motions easily. However, it is difficult to specify variety of motions with these systems, and the result motions are often alike each other.

Using motion libraries, animators can apply existing animation data to other characters. However the motions are also alike each other, and, of course, it is impossible to use the motion which is not in the library. Moreover, the library data depend on the target models. To solve this problem, various techniques [6,12,11] are proposed where original motions are transformed to fit to the target models by using the spacetime constraints.

The paper by Knep et al. [9] introduced a technique which uses an armature for input device. A motion of an armature can be input like the motion capture system. The animator manipulates an armature with position sensors to specify postures of a character in keyframes. Monkey 2[**] and MOFU[††] are commercial tools that employ this technique. It is also possible to express the motion of a character in real time with an armature. The advantage of this technique is that it does not need an expensive and large-scale system as compared to motion capture. Armatures have position sensors at their joints so that their postures can be easily obtained. It is also possible to apply this technique for capturing the motion of animals and virtual creatures.

[*] http://www.dstorm.co.jp/

[†] http://www.artware.co.jp/

[‡] http://www.ex-tools.co.jp/products/physiA2/index.html

[§] http://www.discreet.jp/products/cs/

[**] http://www.didi.com/monkey2

[††] http://www.arwn.co.jp/mofu

2.2 Mixed Reality

Mixed Reality (MR) is the technique that seamlessly combines virtual CG images made by computer and the real environments in real-time. This technique comes under the spotlight at the various fields of education, amusement, city plan, contingency plan, medical care, welfare, and so on [2][4] [13][14][1]. The research by Mizumori et al. [5] supports the interaction among users using the overlay of a human and an avatar in an immersive virtual space called CAVE^{TM}. In this system, a user wears stereoscopic glasses and optical motion capture devices, and can control avatar's posture as if he/she is in the virtual space. The CAVE system used in this research is a very large system, so it is difficult to use this system with an armature on a table.

3 SUPERIMPOSING CG ON ARMATURE

The key idea of our technique is to superimpose the image of a CG character on an armature. The user wears an HMD and manipulates the armature. A physical armature and a virtual CG model are superimposed and displayed on the screen of the HMD. The user can feel as if he/she manipulates the virtual model by his/her hands directly. When he/she creates an animation, he/she can concentrate his gaze to the armature so that the user can input animations efficiently. In addition, there are more advantages as follows:

1.It is easy to recreate already created postures with a physical armature.

2.Candidates of postures can be presented for users who are not familiar with target motions.

In this section, we describe our technique that uses an MR environment for making character animations with a physical armature.

3.1 Recreating a Posture

It is difficult to recreate postures of a CG model with an armature. In existing systems, the user has to manipulate the armature on a physical table with looking at the CG model on the virtual desktop, or has to edit animation data directly using keyframe animation tools with a mouse and a keyboard.

In our system, in order to recreate the posture of a character, the user firstly manipulates the armature so that it is superimposed on a CG model at the screen of HMD. Then, the user edits a posture of the model by moving the armature. The system also suggests to the user which parts of the armature should be moved to which direction, which is displayed based on the comparison between the current posture and the target posture.

3.2 Presentation of Candidates

In the case of motion capture, usually, expert performers act the motions to be recorded. They usually do not lose how should they move. However, in the case of input with an armature, the manipulator of an armature is not necessarily a professional of the target motions. In such case, he/she occasionally lose how he/she move the armature at the next frame. It is difficult to express motions whose details he/she doesn't know well without books or documents.

The problem can be alleviated by presenting candidates of postures to the user using the MR technique. For example, when the system finds that the dynamic equation is applicable to the model from the created frames, the system estimates posture candidates at the next keyframe. The system also searches motion libraries and finds motion data similar to the input frames. The inferred candidates are displayed on the user's view to support manipulating an armature. The user can use the presented candidates as it is or use them by slightly changing them.

3.3 Application Control with Markers

Because our system employs an MR environment where the user wears HMD, it is difficult to use ordinary input devices such as a mouse or a keyboard in order to control the system. Therefore, instead of these devices, we use *markers* to control the system. Figure 3 shows examples of such markers. A picture on each marker represents a command of our system.

To recognize these markers, we employed ARToolkit [7]. Using this library, by registering the command patterns inside the markers, the system can identify which marker is shown by the user.

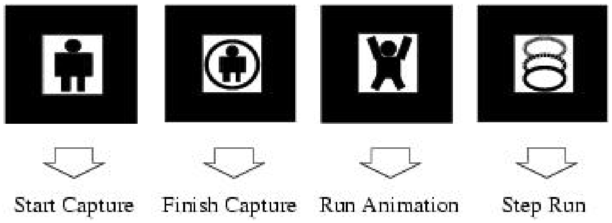


Figure 3: Examples of Markers

4 IMPLEMENTATION

We have implemented a prototype system to evaluate effectiveness of our technique --- superimposing CG models on armatures to support specifying keyframes. The system is written in C++ and works on Linux running on the PC with Pentium4 1.8GHz. The current prototype system is implemented for the purpose of evaluation, and does not aim at practical use. Several functions such as presenting posture candidates for the user described in Section 3 are not yet implemented.

4.1 System Device

We have created two devices for the system: a see-through HMD and an armature for input postures.

Stereo See-through HMD

We have constructed a stereoscopic HMD for our system, in order that the user can get 3D feeling much more. We have put two non-stereoscopic HMD into one so that two inputs from the left and the right camera can be displayed for each eye. The user can see the stereoscopic image, because he/she looks the parallactic images on HMD's left and right eyes. We used EYETREK (OLYMPUS) as the non-stereo HMD, and used Millenium G450 DualHead video card as the device which can output two images.

Because we employed MR, our system needs see-through HMD. There are two types of see-through HMD: *Video See-through* HMD and *Optical See-through* HMD. We adopted video see-through system, because it can be constructed cheaper and it can realize a natural fusion between the real and virtual worlds. In order to implement a video see-through HMD, we attached two CCD cameras to the HMD. The HMD displays the right CCD camera's image on the right eye, and displays the left camera's image on the left eye. As a result, the user can see the stereo and see-through images on the HMD (Figure 4).

Armature with color-codes

In our system, the armature whose joints are color-coded is the device which control posture of the CG model. The system captures this armature's posture by two cameras attached to the HMD.

For this prototype system, we used a commercially available figure model (VOLKS: NEO-GUY) as an armature. We put marks at the figure's joints by different colored tapes in order to recognize the armature's posture by cameras (Figure 5). The system can easily find the identical points from the captured left and right images by means of color thresholding.

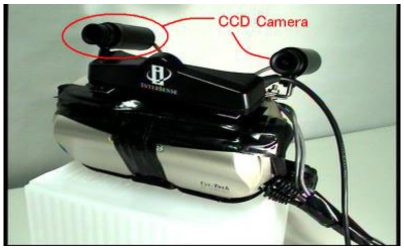


Figure 4: HMD

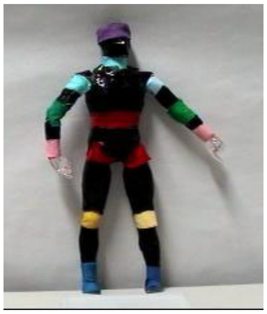
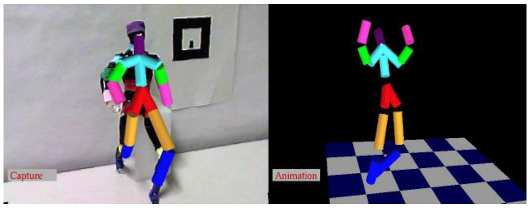


Figure 5: Armature

4.2 The Software System



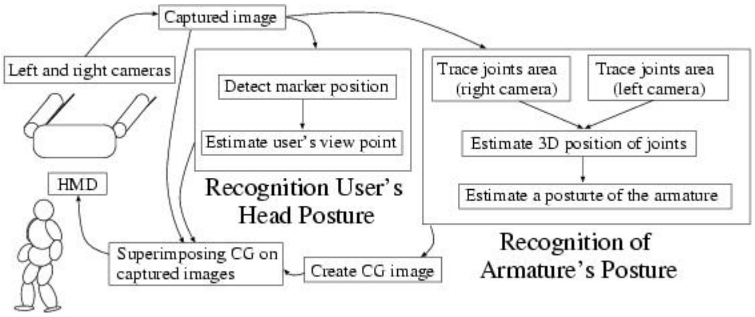

Captured image
Left and right cameras
Detect marker position
Estimate user's view point
Recognition User's Head Posture
Trace joints area (right camera)
Trace joints area (left camera)
Estimate 3D position of joints
Estimate a posture of the armature
Recognition of Armature's Posture
HMD
Superimposing CG on captured images
Create CG image

Figure 6: Snapshots (L: Superimpose R: Run Animation)

Figure 7: Process flow of superimposing

Figure 6 and Figure 7 are snapshots of our system. The user can create a simple walk motion for few minutes. Our system has two modes: *Capture armature's posture (Create/Edit data)* and *Run (step) animation*. Figure 7 illustrates the flows of processing captured images of an armature to superimposing the CG model on HMD screen. We used OpenCV library (Open source computer vision library)[‡‡] for processing images, ARToolkit library[§§] [7] for recognizing head posture, and OpenInventor library [8] and GLUT library for rendering 3D models.

4.2.1 Image Processing

In order to determine the 3D positions of joints of the armature, the system firstly calculates their 2D positions in the image captured by the cameras. The system determines the position of each joints in the following way:

1. Color thresholding
2. Elimination of noise (moving-average method)
3. Dilation
4. Separation of color area by means of labeling

The color thresholding is achieved by two steps: (1) RGB values of image pixels are converted into HSB(Hue, Saturation, Brightness) values, and (2)when the difference between the HSB value and the average of HSB values calculated during the camera calibration is less than the threshold, the pixel is considered as the color area. After the color thresholding, noise elimination and dilation is executed to avoid picking up incorrect pixels and discarding correct pixels in the area of joints. Labeling is necessary to separate the detected color areas, because we used only seven colors and the same colors are marked on different joints. If all joints are colored by different colors, labeling is not necessary. Finally, bigger areas are chosen from the labeled ones, which are considered as the area of joints.

4.2.2 Determining the Armature's Posture

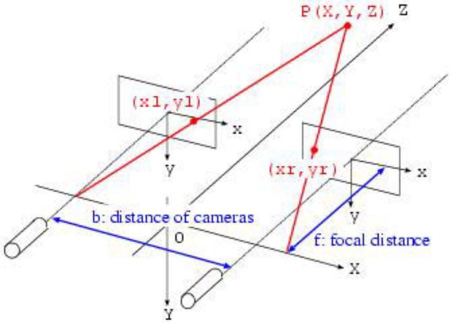


Figure 8: triangulation

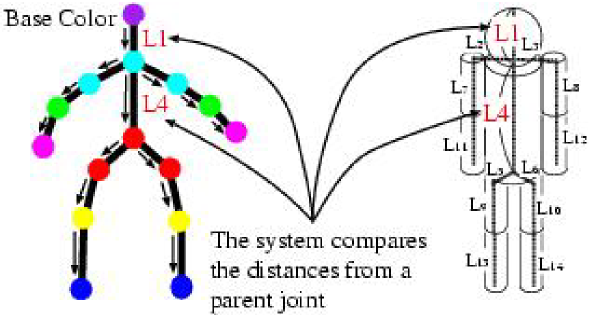


Figure 9: Estimate link structure

[‡‡] Intel Co. (http://intel.com/research/mrl/research/opencv)

[§§] ATR Media Integration Communications Research Laboratories (http://www.mic.atr.co.jp/ poup/research/ar)

The system calculates the 3D positions of all joints from their 2D positions on the left and right images, and determines the armature's posture. In order to simplify the calculation of positions, we set the left and the right cameras so that they are parallel in the horizontal direction. The distance between two cameras and their focal distance are known in advance.

In this case, the position of the point P in 3D space (X,Y,Z) is calculated from the position of P in 2D images $P_l(x_l, y_l)$ $P_r(x_r, y_r)$ using the following equations[3](Figure 8): (The original point in 3D space is the middle point between two cameras. The original point of 2D image plane is the center point of the image.)

$$X = \frac{b(x_l + x_r)}{2(x_l - x_r)}, Y = \frac{b(y_l + y_r)}{2(x_l - x_r)}, Z = \frac{bf}{x_l - x_r}$$

(b: distance between cameras f: focal distance of cameras)

Then, the system estimates the correspondences between the joints and the colored areas. This process uses the constraint that the distance between two neighboring joints is always constant. For example, if a CG model has the link structure as Figure 9, the system estimates the link structure using the following algorithm:

1. The purple area is the base joint, and considered as the top of the head.
2. The distance between the top of the head and the neighboring joint (neck) is L_1, so the sky blue area whose distance from the purple is nearest to L_1 is considered as the neck.
3. In the same way, the red area whose distance from the neck is near L_4 is considered as the hip.
4. The system searches other joints in the same way.

Finally, the system corrects the position of each joint so that the distances from the neighboring joints are equal to the predefined constant value. It is because if the system incorrectly recognizes the link structure of joints, the distance between neighboring joints will have a wrong value. When the system cannot recognize all positions of joints, the user can capture the armature from other multiple angles.

4.2.3 Recognizing the position of user's head

In the previous section, armature's relative position to the cameras is calculated. Superimposing can be possible with only these results. However, the absolute position in the world coordinate system is necessary in order to use armature's position and posture as key-frame data of animation. Because user's head position is equal to camera position, the absolute position in world coordinate system (P_w) is calculated using conversion matrix(M) from world one to camera one(P_c is the position of joints in camera coordinate system).

$$P_w = MP_c$$

Therefore, the system has to estimate user's head position in world coordinate system (Figure 10).In this system, the user sets a square marker which is the base position in the world coordinate system. The system estimates marker's relative position and orientation from the camera image, and estimates user's head position.

4.2.4 Superimposing

After the system obtained absolute position and posture of the armature and the user's head, the system superimposes the CG character model on the HMD screen using these information. Firstly, the system displays captured images whose distortion is corrected using the distortion rate of CCD cameras which has been calculated when the camera is calibrated. Then, the CG model is superimposed on the captured images using previously calculated conversion matrix.

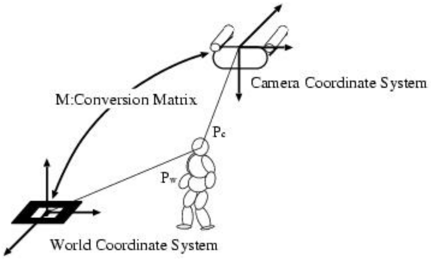


Figure 10: Coordinate System

5 EVALUATION

We have performed simple user tests in order to show an effectiveness of our technique.

5.1 User Test

Subjects were five men in their 20s or 30s, and they all had no experience of creating CG animations. After the trial tasks, we asked some questionnaire. The contents of the test are as follows:

Trial Task

1.Creating an animation which consists of three key-frames using the prepared model

2.Editing all or part of model's posture in a frame

Subjects have done these tasks using the following three methods: (1)superimposing a CG model on an armature using an HMD, (2)only using an armature for control a CG model, and (3)using commercial tools(LightWave3D). The second method is for comparing our technique with the technique using only an armature. It is desirable to use commercial armature devises, but we used our system without superimposing as a devise for the second method. Using this devise, the user can create animations on a virtual desktop.

Questionnaire

After the trial tasks, subjects have answered the following simple questionnaire.

1. Experience of using 3D animation tools (length, scale)
2. Impressions after trial tasks (about each techniques)
   - Complete level of created motion
   - Usability of interfaces (objects manipulation, change camera angle or view point)
   - Usability of our techniques
   - Difficulty, Improvement, Other impressions.

5.2 Results

We have measured trial time of each task. The results of these tasks are Figure11, and the results of questionnaire are Figure 12.

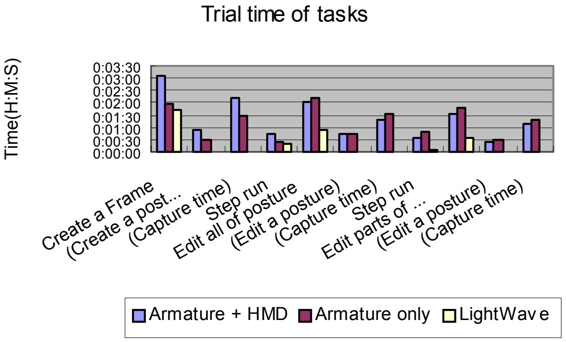


Figure 11: Trial Time of Each Task

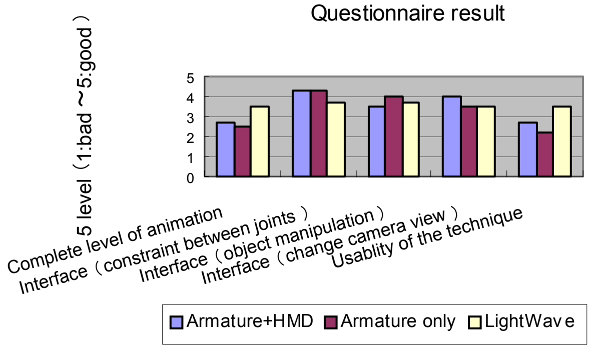


Figure 12: Questions

5.3 Discussion

For editing armature's posture, the result shows the effectiveness of superimposing, that is, the time of user's task trial using superimposing with an HMD is shorter than the time of only using an armature. For creating the posture, the method using superimposing takes longer time. It seems that subjects have acquired some skill of capturing with our system after the test of the first method. Through the test of creating and editing keyframes, our technique took the longest time. This is because the capturing operation in our system is not so accurate, and subjects often failed capturing the armature, which results in long capturing time.

To solve the problem of inaccurate capturing, we are thinking the following three solutions: (1) Using other cameras for capturing an armature will raise accuracy of capturing operation. By using high-resolution wide camera, the capturing will be more accurate. In that case, the CCD cameras on HMD are used only for video see-through for superimposing. (2) Using a head tracker to get user's head position. When the cameras are away from the marker some extent, the system often fails to recognize the marker which is the base position of the absolute world coordinate system. On the other hand, when the head is close to the marker, it is difficult for users to execute capturing operations with bringing the marker and the entire armature into his/her view. Using a head tracker, we can get head positions easily. (3) It is preferable to adjust light conditions so that the colors in the captured images do not change drastically. Therefore, we have to adjust lighting (fluorescent light, spot light etc.) well.

According to the results of questionnaire, many subjects answered that our technique is easy to use as compared to the commercial tool about interfaces. For total usability of tools, the commercial tool got better impression. We think our system will get better results by improving inaccurate capturing and insufficient functions for creating animations.

6 SUMMARY

In this paper, we have focused on the method for creating character animations using an armature, and have proposed a new technique for supporting it by using Mixed Reality environments. In MR environments, the system can enrich the real world and can support the user's work in the real world. Our system supports user's interaction with an armature by superimposing a CG model of the target character on the armature in MR environments. When the user edits a frame, the system displays CG model of editing frames on the HMD screen. When he/she creates a new frame, the system shows candidates of posture in the next frame.

We have constructed an animation tool that implements the above technique, and also evaluated the system through simple user tests. We have got the result from the time of trial tasks and the questionnaire, which using an armature in MR environments raises working efficiency as compared to using it in normal environments. We are to improve the usability of our system by analyzing problems occurred during the user tests.

REFERENCES

1. Rolland Arsenault and Colin Ware. Eye-hand co-ordination with force feedback. *Proceedings of CHI 2000*, 2000.
2. Ronald T. Azuma. A survey of augmented reality. *Teleoperators and Virtual Environments 6,4 (1997)*, 1997.
3. Al Bovik, editor. *Handbook of Image and Video Processing*. Academic Press, 2000.
4. David Drascic, Julius J. Grodski, Paul Milgram, Ken Ruffo, Peter Wong, and Shumin Zhai. Argos: A display system for augmenting reality. *INTERCHI'93: ACM Conference on Human Factors in Computing Systems*, 1993.
5. Masayuki Ihara et al. Human movement instruction system that utilizes avatar overlays using stereoscopic images. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG '2000)*, 2000.
6. Michael Gleicher. Retargetting motion to new characters. *Computer Graphics (SIGGRAPH) 1998*, 1998.
7. H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. *ISAR '2000 (Intelligent RObots and Systems)*, 2000.
8. Open Inventor Architecture Group Josie Wernecke. *The Inventor Mentor*. Addison Wesley, 1994.
9. Brian Knep, Craig Hayes, Rick Sayre, and Tom Williams. Dinosaur input device. *Conference proceedings on Human factors in computing systems (CHI) 1995 Pages 304 - 309*, 1995.
10. Joseph Laszlo, Michiel van de Panne, and Eugene Fiume. Interactive control for physically-based animation. *Computer Graphics (SIGGRAPH) 2000*, 2000.
11. Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. *Computer Graphics (SIGGRAPH) 1999*, 1999.
12. Zoran Popovic and Andy Witkin. Physically based motion transformation. *Computer Graphics (SIGGRAPH) 1999*, 1999.
13. Yanqing Wang and Christine L. MacKenzie. The structure of object transportation and orientation in human-computer interaction. *Proceedings of CHI 1998*, 1998.
14. Yanqing Wang and Christine L. MacKenzie. Object manipulation in virtual environments: Relative size matters. *Proceedings of CHI 1999*, 1999.