Direct Manipulation of 3D Virtual Objects by Actors for Recording Live Video Content

Michihiko Minoh, Hideto Obara, Takuya Funtatomi, Masahiro Toyoura, Koh Kakusho
Academic Center for Computing and Media Studies
Kyoto University
Kyoto 606-8501, Japan
{minoh, obara.h, funatomi, toyo, kakusho}@mm.media.kyoto-u.ac.jp

Abstract

In this article, we will describe our work aiming to realize direct manipulation of 3D virtual objects used as virtual props by a human playing a role of an actor for recording a live video content. For direct manipulation of the 3D virtual props, the actor wears a data glove, which is replaced by a realistic virtual hand in the same 3D shape as the actor’s hand using chroma key before the video is delivered to viewers. When the virtual hand replaces the data glove on the actor’s hand, its posture is adjusted so that it grasps the virtual props without misalignment. For modeling the virtual hand, the actor’s hand is measured by the light stripe triangulation while moving with various postures. The 3D point set obtained as the result is segmented into the points corresponding to the parts of the actor’s hand based on the difference in motion among those parts so that the segmented points are integrated into a 3D description of an articulated object. The 3D virtual objects used for virtual props are modeled by recovering the 3D shapes of the real objects by the volume intersection method. The 3D shape recovered by the method is improved using motion of the object and random pattern backgrounds.

1. Introduction

With the recent development of 3D computer graphics (CG) technology, CG animations are often employed together with live-actions in movie production. Although combinations of live-actions and CG animations in movies are very sophisticated and elaborate, they are in most cases realized in the result of manual postproductions by many specialists, after recording of live-actions and creation of CG animations are completed.

In TV production, on the other hand, the live video recording system called a virtual studio (VS) [1] is often used for recording the live videos in which live-actions by real human actors with the virtual set created by 3D CG, in order to check the resultant videos obtained by combining the live-actions and the virtual set in real-time without post-production. However, human actors cannot interact with the virtual set directly in VS, because it does not detect motion of the actors. The only way to realize interaction between an actor and the virtual set, if possible, is that an operator monitoring the actor in the control room produces visual responses of the virtual set manually when the actor shows a motion for interacting with the virtual set.

The goal of our work described in this article is to realize the interactive virtual studio (IVS) where actors can manipulate virtual props given as 3D virtual objects created by CG, directly by interaction with the props with no assistance. IVS is useful especially for recording the live video content in which an actor gives a presentation using 3D virtual props.

Typical examples of the live video content of this kind are videos of lectures with 3D teaching materials. Although lectures in educational institutions are often recorded for the purpose of e-learning recently, all the teaching materials used in the lectures are based on 2D representations including texts, images, figures and so on, because they are presented by slides or blackboards in conventional lecture rooms. By realizing IVS, it becomes possible to produce live video content of lectures with teaching materials based on 3D representations by virtual objects.

In the remainder of this article, we will describe our work for realizing IVS. Since direct manipulations of a virtual 3D object has been realized in the field of virtual reality (VR), it is expected that IVS can be realized if VS and VR are integrated successfully. Thus we will first describe how IVS is realized by introducing VR devices into VS in section 2. In the IVS, the virtual object is manipulated by a data glove. For avoiding that the data glove appears in the video content presented to viewer as well as for correcting the incon-
sistency between the posture of the actor’s hand and the 3D virtual props, we propose to replace the actor’s hand with the virtual hand. This virtual hand needs to have the same shape as the actor’s hand for the reality of the video content. In section 3, we will describe a new method for measuring the 3D shape of an articulated object, which is represented by a human hand, based on the light stripe triangulation. In section 4, we will propose a new method for improving the 3D shapes of real objects recovered by the volume intersection method, assuming to introduce the method for modeling 3D virtual objects to be used as virtual props by observing real objects.

2. Interactive Virtual Studio (IVS)

2.1. Virtual Studio

VS is an environment for recording videos of human actors with the virtual set created by CG. In VS, the video of an actor is recorded in a studio with the background colored in key color, which is in most cases blue or green, so that the region corresponding to the actor in the video image can be easily extracted by removing the region in the key color. The studio camera is equipped with sensors for measuring the camera work including panning, tilting, zooming and dollying, which are used for creating the CG image of the virtual set with the same camera work as the studio camera. The synthetic video image delivered to viewers is composed by superimposing the region of the actor extracted from the studio camera image onto the CG image of the virtual set.

2.2. Direct Manipulation of Virtual Props

Although VS usually employs 3D virtual objects for the set, it is also possible to use virtual objects as virtual props. However, as described in section 1, neither virtual props nor the virtual set can be manipulated directly by actors in VS, because motion of their hand used for virtual object manipulation is not measured, whereas the camera work of the studio camera is measured by sensors. In order to make virtual objects manipulatable for an actor, it is necessary to measure the motion of the actor’s hand used for virtual object manipulation.

One of the problems for realizing IVS is how to measure the motion the actor’s hand. In the field of VR, a data glove is often used for measuring motion of the user’s hand manipulating virtual objects. However, if we use a data glove in VS, it appears in the video image delivered to viewers. Although it could be considered to use computer vision technique for estimating the posture of the actor’s hand without using data glove, it is not easy to realize real-time posture estimation with accuracy sufficient for virtual object manipulation.

As another problem for realizing IVS, it often happens that the posture of the actor’s hand manipulating a virtual object is misaligned with the object. Since virtual objects cannot be seen by the actor in VS, he/she needs to check whether the posture of the hand is consistent with the shape, position and orientation of the virtual object to be manipulated by looking at the synthetic video image with the monitor placed on the floor of the studio. In this situation, it is not easy for the actor to move his/her hand so that the 3D virtual object is grasped properly by the hand without misalignment. If the actor’s hand and the virtual object under manipulation do not contact with each other appropriately, the resultant video content is visually undesirable.

2.3. Introducing Virtual Hand

In order to solve both of the two problems described above, we measure the motion of the actor’s hand manipulating a 3D virtual object with a data glove, and replace it with a virtual hand created by 3D CG in the live video delivered to viewers. By replacing the data glove with the virtual hand in the shape identical to the actor’s hand, we can measure the motion of the actor’s hand, while avoiding the problem that the data glove appears in the resultant video.

We can also correct misalignment of the actor’s hand with the virtual object to be manipulated, by adjusting the posture of the virtual hand before the video is delivered to viewers. If we do not replace the actor’s hand by a virtual hand, it is only possible for us to adjust the pose of the virtual object. However, if the posture of the actor’s hand is not consistent with the shape of the virtual object to be manipulated, the object never fits into the actor’s hand by rigid transformation, which does not change the 3D shape of the virtual object. By replacing the actor’s hand with a virtual hand, we can determine the posture of the virtual hand so that the misalignment is corrected, independently from the posture of the actor’s hand.

When we replace the actor’s hand by a virtual hand, we need to align the virtual hand grasping the virtual object with the actor’s arm at the wrist. This alignment can be realized by rigid transformation of the virtual hand grasping the virtual object, unlike the alignment of a virtual object with the actor’s hand.

2.4. Constraints for Virtual Hand

In our IVS, we extract the region corresponding to the data glove in the image obtained by the studio camera by painting the data glove in the color different from the key color used for the background of the studio. The corrected posture of the virtual hand is chosen so that the cost functions representing the three kinds of constraints described
below are minimized all together.

The first constraint requires that each fingertip is placed on the surface of the virtual object and each finger cushion faces to the object surface, in order to make the virtual hand grasp the virtual object without misalignment. This constraint is represented by the cost function that evaluates the distance between each fingertip and the surface of the virtual object as well as the difference in direction of surface normal between each finger cushion and the virtual object (see Fig. 1(a)).

The second constraint requires that the whole region corresponding to the data glove in the image of the studio camera is completely covered by the CG image of the virtual hand. After the region corresponding to the data glove is extracted and removed from the image of the studio camera, a blank area is left in the image. The second constraint is introduced to fill up this area with the CG image of the virtual hand. This constraint is represented by the cost function that evaluates the difference between the regions occupied by the data glove and the virtual hand respectively on the image plane of the studio camera (see Fig. 1(b)).

The third constraint keeps the posture of the virtual hand as close to as that of the actor’s hand, which is obtained as sensory data of the data glove. In virtual object manipulation, the actor intends to move his/her hand so that its postures become appropriate for accomplishing the planned manipulation, although each posture obtained as the result includes some errors due to the reason described in section 2.2. It means that each posture taken by the actor’s hand does not need a major correction but a minor one to grasp the virtual object properly. The third constraint realizes correction of each posture of the actor’s hand using the virtual hand with a minor degree. By introducing the third constraint, the postures of the virtual hand does not differ widely from those of the actor’s hand, and as the result, the actor can feel that the motion of his/her hand is reflected appropriately in the motion of the virtual hand. This constraint is represented by the cost function that evaluates the difference in posture between the actor’s hand and the virtual hand.

Figure 2 shows the result of replacing the data glove with a virtual hand. For an image of the studio camera in (a), we replace the region of the data glove with the virtual hand as shown in (b), and correct its posture as shown in (c). In this experiment, the virtual hand is not in the same shape as the actor’s hand.

3. Modeling the Virtual Hand Used in IVS

3.1. Light Stripe Triangulation

In order to realize IVS described in section 2, we need to prepare the virtual hand in the 3D shape of the actor’s hand. In order to obtain the virtual hand that really looks like the actor’s hand, we need to measure its 3D shape with sufficiently high degree of accuracy. We employ the light stripe triangulation for this purpose, because it is known as the method that can measure the 3D shapes of objects with the highest accuracy among all the possible methods. Actually, this method is frequently used to measure the 3D shape of a human body [2][3].

The light stripe triangulation measures the 3D shape of an object by observing the object illuminated by the laser
sheet with a camera. Since only the 3D shape of the object surface illuminated by the laser sheet can be obtained at a single moment by this method, the object needs to be swept by the laser sheet by changing its projection angle for obtaining the whole shape of the object. The data obtained by the measurement at each moment consist of a set of 3D points.

3.2. Measuring Moving Objects

In the light stripe triangulation, the object to be measured has to keep still while its whole surface is swept by a laser sheet. If the object moves before the sweeping is completed, the obtained 3D shape is distorted as illustrated in Fig.3. When we measure the body of a human subject, the subject has to keep his/her posture unchanged during the sweep of laser sheet. Although the sweep is usually completed just in a few seconds, it is not easy for our humans to keep their posture unchanged during the sweep of the laser sheet. If the object moves before the sweeping is completed, the obtained 3D shape can be distorted as illustrated in Fig.3.

3.2. Measuring Moving Objects

In the light stripe triangulation, the object to be measured has to keep still while its whole surface is swept by a laser sheet. If the object moves before the sweeping is completed, the obtained 3D shape is distorted as illustrated in Fig.3. When we measure the body of a human subject, the subject has to keep his/her posture unchanged during the sweep of laser sheet. Although the sweep is usually completed just in a few seconds, it is not easy for our humans to keep our body absolutely still like a doll, even if we make a lot of efforts. This is also the case even for measuring a hand.

![Laser scan](image)

**Figure 3. Distortion due to object motion.**

Let us denote the 3D position of a point in the data obtained at moment $t$ by $p_i$. When the whole object is in the same motion represented by rigid transformation $W(t_1)$ from $t_1$ to $t_2$, we can adjust $p_{i2}$ to $p_{i1}$ by removing the portion changed after $t_1$ until $t_2$ by $W(t_1)$ from $p_{i2}$. By making this adjustment to the 3D positions of all the points in the data obtained at each moment, we can integrate those data with each other to obtain the whole 3D shape of the object with no distortion due to its motion.

Based on this idea, FastSCAN[4] realizes measurement of the 3D shape of an object by moving a hand-held light source of a laser sheet equipped with a electro-magnetic 3D position sensor for measuring the motion of the light source. In this case, the measuring device rather than the object is in motion, which is measured by the position sensor in order to cancel the effect of the motion to the data obtained at each moment before they are integrated with each other.

Unlike the measurement by FastSCAN, when the object is in motion, it is required to obtain the motion of the object. 3D position sensors cannot be used for this purpose because the 3D shape of the object is not preserved if those sensors are attached on the surface of the object. However, it is still possible to estimate the motion of the object by attaching markers to it. Since the object is observed by cameras in the light stripe triangulation, the object motion can be estimated by tracking the markers in the image of each camera. If we employ the markers in flat shapes, they do not interfere with the measurement of the object shape.

3.3. Articulated Objects in Motion

For articulated objects including a human hand, each part of the object can move differently. Let us denote the rigid transformation that represents the motion of $i$-th part $l_i$ of the object from $t_1$ to $t_2$ by $W_i(t_1)$ ($i = 1, \cdots, L$; $L$: the number of the parts of the articulated object). It is possible to estimate $W_i(t_1)$ by attaching markers to $l_i$. It could also be possible to know the jointed structure of the object, which determines the connections between the parts of the object.

However, in order to adjust the 3D position $p_{i2}$ of a point in the data at $t_2$ to $p_{i1}$ as in the case described in section 3.2, the motion of the point needs to be assigned from among $W_1(t_1), \cdots, W_L(t_1)$. It means that the points in the data need to be segmented into the sets of points corresponding to object parts $l_1, \cdots, l_L$, respectively. Since the 3D shape of an object is described as a set of points in light stripe triangulation, finding out the set of points corresponding to each part of the object from among the data by segmentation is equivalent to recovering the 3D shape of each part of the object. Thus, segmentation of the data and shape recovery of the object cause a chicken-and-egg problem.

In order to cope with the problem, we use silhouettes of the object, which are extracted from the images of the cameras observing the object for the light stripe triangulation. Let us consider the case in which the object consists of two parts denoted by $l_a$ and $l_b$, as the simplest case (see Fig.4). The motions of $l_a$ and $l_b$ from $t_0$ to $t_2$ is represented by rigid motions $W_a(t)$ and $W_b(t)$, respectively. These motions are estimated by tracking the markers attached to $l_a$ and $l_b$ from $t_0$ to $t_2$ ($t_0$ and $t_1$ is shown as examples of $t$ in the figure).

First, we assume all the points in the data obtained at $t$ belongs to $l_a$ and move in motion $W_a(t)$. If this assumption is correct, each points in the data at moment $t$ should be projected onto the inside of the silhouette of the object obtained at moment $t_2$ after the 3D position of the point is adjusted to the position at $t_2$ using $W_a(t)$. If this is not satisfied for a point in the data, we call the point a “ghost”. If a point becomes a ghost by assuming the point to move in motion $W_a(t)$, it is found that the point belongs to the part different from $l_a$, moving in the motion different from $W_a(t)$. Since the object consists of only two parts in this
case, we can easily determine that the point belongs to \( l_b \) and moves in motion \( W_b(t) \).

By extending the process above to the case of the object that consists of more than two parts, we can segment all the points in the data obtained all the moments. Once we can obtain segmentation of the points in the data, they can be integrated into the shape of each part by the process of adjusting the position of each point in the data described in the beginning of this section.

Assuming that the human hand consists of 18 rigid parts with the jointed structure illustrated in Fig.5(a), we measure the 3D shape of a human hand changing its posture using our method, where the motion of each part of the hand is estimated by tracking the three markers attached on each part of the hand. The shape obtained by conventional light stripe triangulation is shown in (b). Due to the motion of the hand, the shape includes distortion.

Using our method, we recover the 3D shape of the hand from the data obtained with the motion of the hand taking 9 different postures and the silhouettes of the hand for those postures. The obtained result is shown in (c). This shape is obtained based on the result of segmentation of the data shown in (d), in which different segments are shown in different gray scales. Once the 3D shape in (c) is obtained with correct segmentation of the data, we can reproduce the 3D shape of the hand at any posture. Figures (e) and (f) are examples of the 3D shapes reproduced by changing the posture of the hand in (c) to reproduce the hand in different postures. These results reproduce the real shapes of the hand in the corresponding postures, which are shown in (g) and (h) with sufficient degree of accuracy.

Figure 4. Segmentation of data.

Figure 5. Measurement of human hand.

4. Modeling 3D Virtual Props Used in IVS

4.1. Volume Intersection Method

In order to record a live video with direct manipulation of 3D virtual objects by IVS, we need to prepare not only the virtual hand as described in section 3, but also the virtual objects used as virtual props. When we record lectures as typical examples of live videos by IVS, virtual props used as teaching materials are in many cases cultural heritages, natural history specimens, and so on, which actually exist as real objects. For adding various kinds of visual effects including marking, highlighting, and so on, it is of course not sufficient just to present those real objects directly in live video, but we need to model the real objects as vir-
tual objects for realizing visual effects with CG. If the 3D shapes of the real objects used for virtual props can be obtained by computer vision techniques in some degree of accuracy, those 3D shapes can be used as good initial models, with which CG illustrators can start manual modeling of the virtual objects used as props by just modifying those 3D shapes slightly.

Since various kinds of real objects can be used as virtual props, the method for recovering their 3D shapes should be applicable to various kinds of objects regardless of the properties of their surface including colors, reflectance of light and so on. Although the method that is currently available for measuring 3D shapes of objects with the highest accuracy is light stripe triangulation as discussed in section 3, the method cannot be applied to the objects that do not reflect laser light. For this reason, we employ the volume intersection method, which can recover the 3D shapes of real objects regardless of their surface properties.

The volume intersection method recovers the 3D shape of an object from silhouettes extracted from 2D images of the object with different viewpoints. Given a silhouette of an object from a viewpoint, it is guaranteed that the object is included in the visual cone with the apex at the viewpoint and the base corresponding the silhouette. The volume intersection method recovers the 3D shape of the object as the visual hull, which is the intersection of the visual cones from all the viewpoints.

### 4.2. Increasing Viewpoints

For obtaining the precise 3D shape of an object, the volume intersection method has some problems. One of them is how to increase the number of viewpoints. When the number of viewpoints is not sufficiently large, the resultant visual hull includes the region that does not actually include the object. It is necessary to increase viewpoints in order to obtain the visual hull close to the object. However, it is not realistic to place many cameras around the object for increasing viewpoints, due to spatial constraints for installing cameras densely. In order to cope with this problem, we propose to increase viewpoints without increasing cameras by observing the object in rigid motion.

When the object whose 3D shape needs to be recovered is in a rigid motion, each camera observing the object changes its relative positions to the object every moment. If the rigid motion of the object can be correctly estimated, images obtained by each camera at different moments serves virtually as images obtained by cameras in different positions. By using these images, we can increase the accuracy of the visual hull without increasing the number of cameras.

In order to estimate the rigid motion of the object correctly, we need to find some feature points on the real surface of the object as the landmarks for tracking its motion. However, it is not easy to find such feature points in the visual hull because it does not coincide with the object completely and its 3D shape changes every time the silhouette of a new viewpoint is obtained through the object motion.

In previous work, the feature points called Frontier Points are used as the landmarks for tracking the motion of the object [5]-[10]. However, the frontier point is not actually guaranteed to be included in the object region. We propose new feature points called Outcrop Points, which are guaranteed to be on the real surface of the object once they are found in the visual hull. We use these points as the landmark points for tracking object motion by extracting them in the visual hull obtained at any moment of the object motion. When a voxel in the visual hull satisfies the following two conditions, we regard the voxel as an Outcrop Point.

1. The voxel is projected onto a pixel on the contour of the silhouette for at least one image.
2. No other voxel is projected on the same pixel in condition (1).

The voxels that satisfy condition (1) correspond to the Frontier Points. As illustrated in Fig.6, voxels $v_b, v_c, v_d$ satisfy condition (1). However, even if at most two of them is removed from the visual hull, the remaining visual hull is still consistent with the silhouette of the object. Unlike these voxels, if voxel $v_a$, which satisfies both of the conditions (1) and (2), is removed from the visual hull, the remaining visual hull causes missing at pixel $P_a$ in the silhouette of the object. It means that $v_a$ is guaranteed to be on the surface of the object.

The visual hull recovered from object images including the image in Fig.7(a) is shown in (b). By using Outcrop Points extracted from the visual hull as shown in (c) for motion estimation, the accuracy of the 3D shape is improved when the object is in motion as shown in (d). However, the
tail of the recovered shape in (d) is missing due to the failure in extracting silhouettes from object images.

![Image](image1.png)

(a) Real image  (b) 3D Shape without motion

(c) Outcrop points  (d) 3D Shape with motion

Figure 7. Shape recovery of object in motion.

4.3. Improvement for Silhouette Extraction

As illustrated in Fig.7, missing part occurs in the recovered shape in spite of the increase of the viewpoints, if the silhouette extraction fails. It actually illustrates another problem of the volume intersection method; how to extract silhouettes of the objects precisely. The silhouettes of the object are usually extracted from the object images based on the difference in color between the background and the object region. When some part of the object is in the same color as the background, that part is missed in the silhouettes. Those missing part in the silhouettes cause missing part also in the recovered visual hull. In order to guarantee that the color of the background is different from any colors included in the object, we need to take a trouble for choosing a color appropriate for the background every time we are given a new object.

Although it is proposed to switch two backgrounds in different colors for extracting the silhouettes without depending on the color of the objects[11][12], it requires the object to be fixed while the backgrounds are switched. This requirement lose the advantage of the volume intersection method that the 3D shape of the object is recovered in real-time.

In order to cope with this problem, we propose to use random pattern backgrounds, which consists of many small regions in colors chosen randomly in the color space, for silhouette extraction, as shown in Fig.8.

![Image](image2.png)

Figure 8. Example of random pattern.

One of the advantages expected for the random pattern backgrounds is that the color of the object coincides with the background only for small regions scattering in the image. It means that the missing parts of the silhouette, even if they occur, are small and scattering in the silhouette. Such regions can be easily recovered by filling up small holes in the silhouettes. Fig.10(b) illustrates the silhouette extracted from the object image in (a) by white regions. The holes in the silhouette in (b) can be filled up as shown in (c).

Another advantage of the random pattern background is that different colors are observed for the background of each point on the object from different viewpoints, whereas the color of each point on the object is the same regardless of the viewpoint. By taking this advantage, we can correct the error of the visual hull still remained even after filling up the holes in the silhouettes.

As illustrated in Fig.9, by projecting a voxel (denoted by \( v \) in the figure) on the surface of the current visual hull onto each image of the object, a set of pixels (denoted by \( P_m, P_n \) in the figure) corresponding to the voxel is obtained \((m, n = 1, \cdots, N; N: \text{the number of cameras})\). The voxel on the surface of the current visual hull should be around the surface of the object because the current visual hull, which is obtained from the silhouettes extracted with the random pattern background, should describe the object in some degree of approximation. The set of pixels corresponding to \( v \) should have the same color if \( v \) is really included in the object. We estimate the color of the object from the majority of the colors among those pixels. If the estimated color of \( P_m, P_n \) is close to that of the background (denoted by \( B_m, B_n \) in the figure) in a image of the object, the silhouette extracted from the image is restored at pixel \( P_m, P_n \), because that pixel may be missed due to the similarity in color between the object and the background.
Fig. 9. Object color estimation.

Fig. 10(f) is the visual hull obtained by correcting the visual hull in (e) by the process above. By this correction, the corrected visual hull becomes sufficiently close to the shape of the object in (d).

5. Conclusions

We have described our work for realizing IVS together with the methods for modeling the actor’s hand and 3D virtual objects necessary for the IVS. Since the techniques explained in section 2 - 4 in this article have not already been integrated into a single environment in which all the three methods collaborate with each other, we are planning to accomplish the integration for a future step. For another future step, we are also considering for using our IVS for recording videos of lectures given in our university in order to evaluate the system by producing live video content for practical use in educational institution.

References