

# AUTOMATED CAMERAWORK FOR CAPTURING DESKTOP PRESENTATIONS — CAMERAWORK DESIGN AND EVALUATION IN VIRTUAL AND REAL SCENES

M. Ozeki, Y. Nakamura, and Y. Ohta

University of Tsukuba, Japan

## ABSTRACT

This paper introduces a novel automated camera control method for capturing desktop presentations. For this purpose, we first discuss typical features of shots and their cameraworks that frequently appear in TV programs. To realize those features in our automated video capturing system, we classify the purpose of a camerawork from two points of view: *target* and *aspect-of-target*. Then, we consider the correspondence between the classification and typical shots and cameraworks. We propose the virtual-frame control algorithm based on this idea, and the implementation in our video production system. We then show experimental results that verified our method through two kinds of experiments: virtual video capturing using CG animations and real video capturing of real presentations.

## 1. INTRODUCTION

One of the most important topics on multimedia is contents production. There are great demands for automating contents production of video-based multimedia, since video production is a costly task that requires both considerable skills and time. One typical application is lecture archiving and tagging, for which a number of works are reported and some of them are actually used for real scenes. Capturing, recording, and indexing lectures gives good materials for distant learning and e-Learning.

This paper focuses on *cameraworks* for recording video-based multimedia contents for science classrooms or instruction manuals. Camera control is an essential basis of those videos, since poor cameraworks force the audience's unreasonable pains to keep watching or they can easily make the audience give up watching.

For this purpose, we propose our camerawork by the *virtual-frame control* for capturing desktop presentations. The features of our research are as follows:

- The camerawork is designed based on the observation of TV programs. From the observation, we classified the purpose of a camerawork from two points of view, *target* and *aspect-of-target*.
- The virtual-frame control is adjustable to typical purposes of shots, and is easy-to-use for the user who do

not have special knowledge for taking videos.

- We had experiments in both virtual scenes by CG animation and real scenes. We found that sophisticated camera control is really necessary even with an ideal setting in CG animation, and our camerawork obtained good results compared to simple or straightforward ones.

In the following sections, we first discuss typical features of shots that frequently appear in TV programs. We next classify the purposes of shots and cameraworks, and propose a novel camera control algorithm adjustable for each purpose. Then, we present our experiments through virtual video capturing and real video capturing of real presentations.

## 2. CAPTURING DESKTOP PRESENTATIONS

### 2.1. Features of Desktop Presentations





TV programs of *desktop presentations* such as cooking shows or handicraft classes have typical patterns of shots and cameraworks. TABLE 1 shows typical shots that frequently appeared in our video samples, that is, four cooking programs and two handicraft programs<sup>1</sup>. TABLE 2 shows the proportion that each category of shots occupies each video. We can see the importance of shot C that is a close shot of a workspace (hereafter we call the shot a *workspace shot*) since this type of shot occupies from 50 % to 80 % of the whole video.

From this observation, we found that camerawork is not directly determined by the target *i.e.*, which person or object is mainly shot by a camera. Camerawork is rather tightly related to the focused state of a target *i.e.*, which state of the target must be captured and be focused. Even the same target can be taken with different cameraworks: if the *appearance* of an object is focused, the objects should be always kept at the center of the screen<sup>2</sup>; if the *movement* of hands is important, it is preferable to fix or slowly move a camera for capturing the workspace in which the hands are moving around. We can see this feature frequently in the workspace shot (the shot C).

<sup>1</sup>Cooking videos A(15 minutes), B(12 minutes), C(20 minutes), D(7 minutes), and handicraft videos A(10 minutes), B(15minutes).

<sup>2</sup>In this paper, we use term "screen" as it represents an image or a frame of a video.

TABLE 1 - Typical shots for desktop manipulations.

Shot examples	Shot properties
 <p style="text-align: center;"><b>Shot A</b></p>	<p>This type of shot contains a wide-angled view of a speaker, (an) assistant(s), and a workspace on a desk. Usually, this type of shot is used when a speaker gives explanations before actual operations or when no other appropriate shots are appropriate. This shot is sometimes/often used as an alternative for a shot C when a speaker moves around the studio space.</p>
 <p style="text-align: center;"><b>Shot B</b></p>	<p>This type of shot captures a speaker's face or upper body except a workspace. A shot of type A is often used as an alternative of this type of shot if only one person is speaking. This shot is often used when a speaker talks after the operation is almost finished.</p>
 <p style="text-align: center;"><b>Shot C</b></p>	<p>This type of shot captures a workspace during operations. A cameraman shoots at (a) hand(s), a place, or an operated object, and he/she frequently moves cameras to capture these targets at the center of the screen. This shot often continues to the end of an operation.</p>
 <p style="text-align: center;"><b>Shot D</b></p>	<p>This type of shot, that is, a telop or a flip shot, gives information for the names, the quantity, or other important points. We do not need a sophisticated camerawork, and a fixed camera or a still shot is usually enough for this purpose.</p>

A speaker: a person mainly explains a manipulation, *e.g.*, a chef or a teacher.

The most important point is that certain *trade-off* (hereafter we call “camera control trade-off”) exists between “capturing a target at the center of the screen in order to intensively show the target” and “fixing the field of view in order to show the locus of movements or to show the relation between the target and the background”. In this sense, adjusting the trade-off in capturing hands and an object that can rapidly move is a characteristic problem in capturing a desktop presentation. This characteristic has not been well discussed in previous works for lecture capturing.

The objective of this research is to automate this process by giving appropriate camera control algorithm. To make this subject really tractable, we use a multi-camera sys-

TABLE 2 - Occurrence ratio of each shot.

Programs	Shot A	Shot B	Shot C	Shot D
Cooking A	21.7 %	11.5 %	48.5 %	18.3 %
Cooking B	29.8 %	0.0 %	70.2 %	0.0 %
Cooking C	25.8 %	2.4 %	69.9 %	1.9 %
Cooking D	13.8 %	2.9 %	74.0 %	9.2 %
Handicraft A	13.6 %	1.5 %	84.9 %	0.0 %
Handicraft B	16.5 %	4.8 %	78.7 %	0.0 %

Meanings of each shot are explained in TABLE 1.

tem. The system simulates a human cameraman by using multiple cameras each of which shoots its own target with its own purpose, while a human cameraman that usually uses one camera and shoots one or more targets by changing cameraworks. The actual camerawork is described in section 4.

## 2.2. System Outline

FIGURE 1 shows an outline of our system. In our system, we assume a speaker gives a talk and a demonstration in a fixed space, *e.g.*, 2 m × 2 m square. Each pan/tilt camera is assigned its own target and its own camerawork appropriate for the purpose. For shooting at a target by tracking the target, the system needs to get accurate positions of targets with small latency. For this purpose, the system measures the 3D position of a speaker or an object by magnetic sensors with small errors<sup>3</sup>, and controls multiple pan/tilt cameras with small latency<sup>4</sup>. A target position of on a screen can be calculated based on the field angle and 3D positions of the camera and the target.

Videos taken by those cameras are recorded in MPEG1/2 format. A speech transcript is obtained by a speech recognition software, and is recorded with the target position data and camera control parameters synchronized with videos. Recorded videos are automatically edited based on those data.

This system is designed for automated video production, whose outputs are provided as video manuals (1) or contents for distant learning. In this paper, however, we concentrate on a camerawork and its evaluation.

## 3. RELATED WORKS

Not a few works have been proposed for capturing lectures or meetings (2)–(5). For example, in TIDE project, videos for a lecturer, students, and a blackboard are captured and utilized for an international lecture exchange

<sup>3</sup>In our calibration, the average positional error is less than 10 mm.

<sup>4</sup>In our experiments, the actual delay before a camera begins to start moving is around 400 msec.

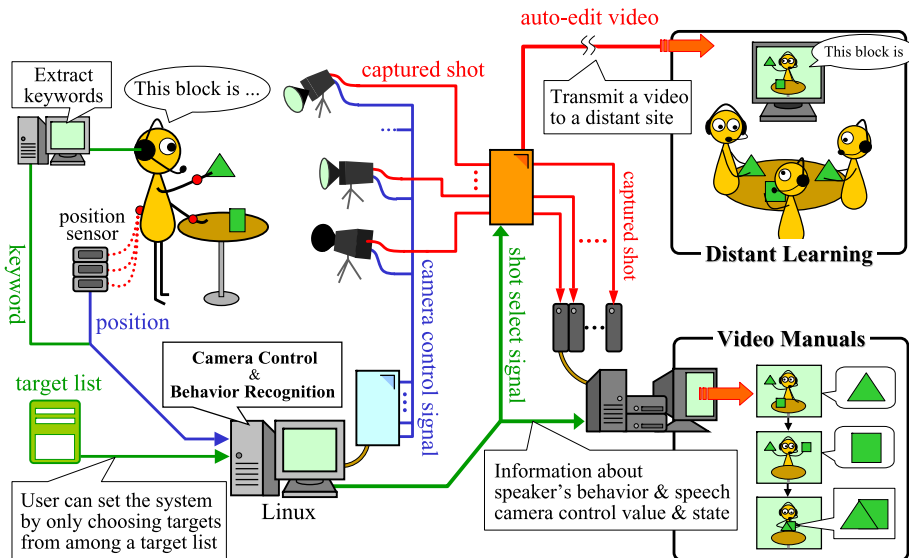


FIGURE 1 - Video capturing system overview.

program. Some of those works use multiple cameras that simultaneously shoot different targets such as a lecturer, a blackboard, or students. However, the camera control techniques used in the above works are mainly for capturing a middle shot or even longer<sup>5</sup> shots, and they are not directly applicable for our purpose, since our target moves much faster in a screen. We need a novel method for taking desktop manipulations.

NHK Science & Technical Research Laboratory has developed an intelligent robot camera system for TV program production, and they are trying to simulate cameraworks of professional cameramen (6). For example, pan/tilt speed is adjusted to that of human cameramen, and some of their experiments showed good results. So far, they have not reported a method applicable to a desktop presentation nor a method applicable to ordinary and inexpensive pan/tilt cameras.

As an automatic camera control system for cooking scenes, Bobick et al., reported the Intelligent Studios (7). Their work may be most closely related to our scheme. Their system, however, did not deal with details of a camerawork. They have not reported subsequent research concerning the realization of a practical system. Our approach is different in the sense that we developed our multi-camera system that works for real presentations through a number of experiments and evaluations in real scenes.

Moreover, we systematically verified our method by virtual video capturing using CG animations. Not a few studies have been reported on cameraworks in computer graphics field. For example, He et al. proposed the Virtual Cinematographer that adaptively switches views

<sup>5</sup>We use term “long” for a shot in which a person or an object appears smaller in a screen, and “close” for a shot in which a person or an object appears bigger.

TABLE 3 - Classification of targets.

Target	
<speaker>	speaker's upper body or face
<hand>	a right(left) hand, both hands
<object>	an important object to be paid attention
<place>	an important static place to be paid attention

from virtual cameras by using a finite automaton (8). Drucker et al. proposed the camera control description language DCCL and applied it to the system CamDroid (9). Those works simulate video capturing on ideal conditions, e.g., no latency for acquiring the position of target<sup>6</sup>, no delay for camera movement, no inertia, and so on. Therefore, while those works are partially good stuff for designing a camerawork in a real scene, they are not directly applicable for our purpose. On the other hand, we evaluate our methods in both a real scene and a virtual scene, and verified that our method is sufficient for both situations. Moreover, we found that we need sophisticated camera control even in a virtual scene with ideal conditions.

#### 4. CAMERA CONTROL FOR DESKTOP PRESENTATIONS

From the observation in section 2.1, we consider cameraworks from two points of view: which *target* we want to shoot, and which *aspect-of-target* we want to focus.

TABLE 3 shows the category of *targets* that we prepared

<sup>6</sup>Even the position in the future is known.

TABLE 4 - Classification of aspect-of-targets and appropriate cameraworks.

Aspect-of-target	Focus	Requirements for camerawork
<appearance>	Appearance of targets such as shape, color, pose, or their changes.	This requires a camera to track a target as quickly as possible with keeping the target at the center of the screen.
<movement>	Movements of target that may include frequent or shaky small motions such as hand movements or dynamic gestures.	This requires target tracking with suppressing small camera movements.
<circumstance>	Target's circumstances or relationship to other objects.	This requires to fix a camera angle as long as possible so that a viewer could easily observe the relation between the target and other objects/persons in the scene.

for this purpose. For example, <speaker> means that the target is speaker's upper body or a face, the <hand> category has three sub-categories, "a right hand", "a left hand", and "both hands". When the target is set to "both hands", a camera tracks a middle point of both hands with the view field around 1 m × 1 m.

TABLE 4 shows the category of *aspect-of-target*. The camerawork suitable for each aspect-of-target is shown in also TABLE 4. This is based on the observation in section 2.1, that is, the camerawork has the trade-off between the following two requirements:

1. Track a target and keep it at the center of the screen as far as possible so that viewers can easily look at the appearance of the target.
2. Fix a camera angle and view field as far as possible to suppress shaky and irritating view changes so that the viewers can easily understand the target's motion and relations to the background.

In our system, the target determines the target to be captured, the view point of a camera, and its view field<sup>7</sup>. The aspect-of-target determines the parameters of our camera control algorithm. Thus, we can determine camerawork characteristics including the above trade-off adjustments by specifying a target and an aspect-of-target. Rough correspondence exists among the above *target* categories, the *aspect-of-target* categories, and the shots shown in TABLE 1:

**shot A:** The target is the <speaker>, and the aspect-of-target can be considered as <circumstance>, since the camera is not often moved even if the speaker moves around.

**shot B:** The target is <speaker>, and the aspect-of-target is <appearance>.

**shot C:** The target is <hand>, <object>, or <place>. The aspect-of-target is <appearance>

or <movement>: it is <appearance> if we want to focus the target's appearance; it is <movement> if we want to focus on motions or loci of the target movement.

**shot D:** The target is <place>. Since this type of shot usually requires a fixed shot, our system do not need to prepare its own camerawork.

The above settings are examples of typical cases, and we can think of other combinations of the target and the aspect-of-target though they might not be important.

## 5. VIRTUAL-FRAME CONTROL

We propose the *virtual-frame control* for realizing the cameraworks shown in TABLE 4. We can adjust the above *camera control trade-off* by changing the parameters of this algorithm.

FIGURE 2 shows the flow of the algorithm, and TABLE 5 explains the parameters for this algorithm. Roughly speaking, a camera quickly tracks a target by immediate tracking mode, and ignores small or shaky motions by motion suppressing mode. Appropriate switching between two modes suppresses irritating shaky views with keeping the target inside a screen.

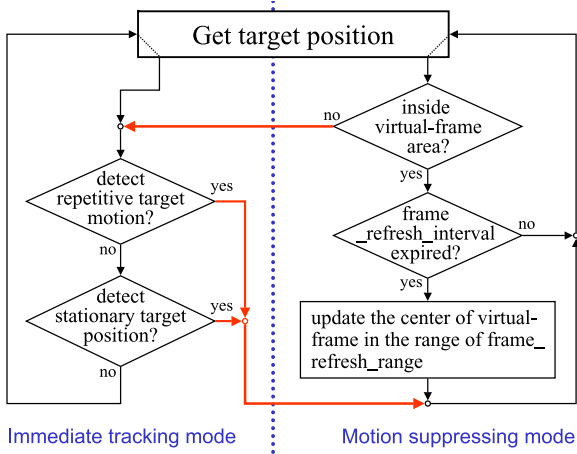
The two modes have the following features:

**motion suppressing mode:** Camera movement is suppressed while the target's apparent position in a screen stays inside the *virtual-frame* assumed in the screen. While the motion suppressing mode is on, the camera pan/tilt angle is re-computed at regular intervals (*frame\_refresh\_interval*) so that the average of the target's apparent position during a certain period (*frame\_refresh\_range*) is located at the center of the screen. The mode switches to the *immediate tracking mode* when the target goes outside the virtual-frame.

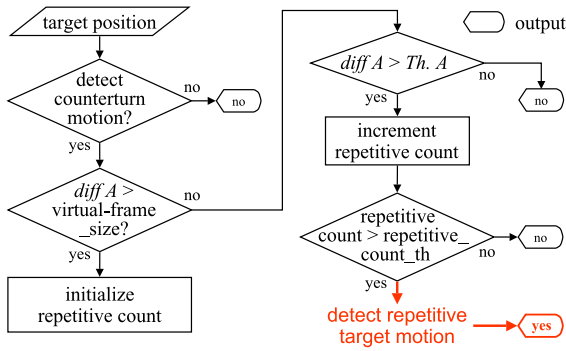
**immediate tracking mode:** A camera quickly and exactly tracks a target. The mode switches to *motion*

<sup>7</sup>From wide-angle to tele-angle, or from long shot to close-up shot.

TABLE 5 - Camerawork parameters.

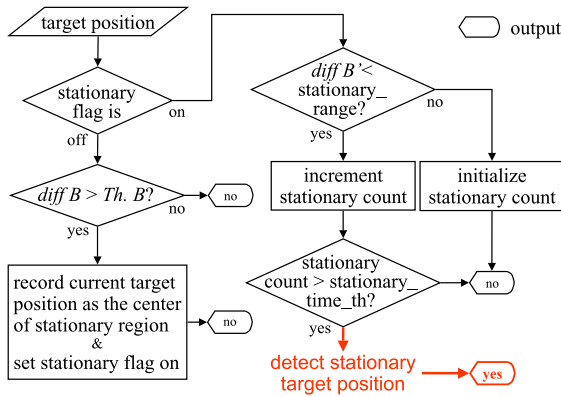


**Detection process for repetitive target motion**



*diff A*: difference from previous counterturn point  
*Th. A*: distance which can be assumed as joggle

**Detection process for stationary target position**



*diff B*: difference from previous target position  
*diff B'*: difference from the center of stationary region  
*Th. B*: distance which can be assumed as stop

FIGURE 2 - Flow of virtual-frame control algorithms (The outline is shown in the upper figure; the repetitive target motion detection shown in middle figure; and the stationary target motion detection is shown in the lower figure).

*suppressing mode* when *repetitive target motion* or *stationary target position* is detected. Repetitive target motion means that counter-changes of target’s movement direction are detected more than a certain

**virtual-frame\_size**

The virtual-frame is a rectangle located at the center of a screen. This parameter represents the ratio of the virtual-frame size to the screen size.

**repetitive\_count\_th**

This parameter represents the threshold for counter-changes(direction changes) of apparent target motion. When the number of counter-turns exceeds this threshold, the system detects “repetitive target motion”.

**stationary\_range**

If the target’s apparent position in the screen stays inside the area specified by this parameter, the system detects “stationary target position”. This parameter is the ratio of the area size to the screen size. The difference from virtual-frame is that this area is floating, and the position is fixed at every time when the target stops.

**stationary\_time\_th**

This parameter represents the time threshold for detecting “stationary target position”. If the target is apparently inside the area defined by *stationary\_range* longer than this threshold, the system detects “stationary target position”.

**frame\_refresh\_interval**

This represents the temporal interval at which the algorithm adjusts a camera angle in the motion suppressing mode. The camera is controlled so that the center of the target’s trajectory is located at the center of the screen.

**frame\_refresh\_range**

This parameter represent the temporal duration in which the algorithm calculates the center of the target’s trajectory.

**smoothing\_degree**

This parameter is the ratio of the process noise variance to the measurement noise variance in the Kalman filter, and it governs the smoothness of tracking. If the ratio is small, the camera tracks more smoothly.

threshold (*repetitive\_count\_th*). Stationary target position means that the target’s apparent position in the screen stays inside a certain area (*stationary\_range*) for a certain period (*stationary\_time\_th*).

To eliminate sensor noise and small irritating motions such as trembles in immediate tracking mode, we use the Kalman filter with the rigid body motion model as system dynamics. A state variable  $\mathbf{x}_k$  and a state transition matrix  $\mathbf{F}$  are as follows.

$$\mathbf{x}_k = \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \quad \mathbf{F} = \begin{pmatrix} 1 & \Delta \\ 0 & 1 \end{pmatrix}$$

where,  $\Delta$  is a sampling interval of a measurement.  $\mathbf{x}_k$  is a state vector containing the current values of position and velocity.

parameters	aspect-of-target		
	<appearance>	<movement>	<circumstance>
virtual-frame_size	fix camera angle only when target stopped		fix camera angle as much as possible
repetitive_count_th	N/A	fix camera angle as soon as repeated motion	change motion suppressing mode slowly
stationary_range	fix camera angle as soon as stationary		
stationary_time_th	fix camera angle as soon as stationary		
frame_refresh_interval	N/A	capture target at screen center	fix camera angle as much as possible
frame_refresh_range	N/A	adjust camera at center of movement	adjust screen center at latest target position
smoothing_degree	track target precisely		track target smoothly

FIGURE 3 - Overview of camerawork parameters for three aspect-of-targets(the base of a triangle means that a the parameter value is large).

The smoothness of the output depends on the ratio of the process noise variance to the measurement noise variance. We consider this ratio (*noise\_variance\_ratio*) also as a camera control parameter that governs the smoothness of tracking.

Thus, the virtual-frame control algorithm realizes various cameraworks by just changing the above parameters. For example, three typical cameraworks shown in TABLE 4 are realized as follows:

**camerawork for <appearance> (A):**

Virtual-frame\_size and stationary\_time\_th are set small in order to quickly follow the target motion and to capture it at the center of the screen as much as possible.

**camerawork for <movement> (M):**

Repetitive\_count\_th is set small in order to quickly detect repetitive motions. This realizes stable capture of target's movements by preventing shaky view field changes. To follow the target movements with ignoring the small motions, virtual-frame\_size is set small and frame\_refresh\_interval is also set small.

**camerawork for <circumstance> (C):**

We use large virtual\_frame\_size and long stationary\_time\_th so that the camera motion is much suppressed compared to the above two cameraworks. We need to make stationary\_time\_th and repetitive\_count\_th large so that system can safely estimates the area where the target moves.

FIGURE 3 shows camerawork parameters for these cameraworks. Since actual values of the camerawork parameters depend on an actual studio environment or equipments, we introduce them in section 6.

## 6. EXPERIMENTS

We had experiments for the evaluation of our camerawork: whether the categorization of a camerawork is necessary, and whether the virtual-frame control algorithm has advantages over other methods. For actual evaluations, we first used CG animation. The virtual scenes are captured by virtual cameras controlled by our camerawork or other typical cameraworks. Then, we had subjective evaluation of the obtained shots based on Thurstone's method of paired comparison (10). After the evaluations by virtual video capturing, we applied our camerawork to the real scenes.



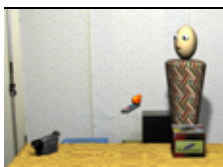
### 6.1. Experiments in Virtual Scenes

In CG animation, we can set various parameters as we want.

- We can place at any location, for example, at exactly the same location, more than one camera that have ideal characteristics.
- We can simulate ideal system conditions, for example, no noise in measuring target position, no camera control delay, and so on.
- We can use every position, velocity, or other information of objects at anytime even in the future.

TABLE 6 shows the scene examples and their explanations: a content, a target, an aspect-of-target, and criteria for evaluation. The aspect-of-target is just for the reference of what should be focused, and this takes no effect on what is actually done in the experiments.

TABLE 6 - Scenes and criteria for evaluation.

	<p><b>Contents:</b> A person holds up a remote controller and a video camera, and explains about them.  <b>Target:</b> &lt;object&gt;  <b>Aspect-of-target:</b> &lt;appearance&gt;  <b>Criterion:</b> Whether it is easy to see the appearance of an object.</p>
	<p><b>Contents:</b> A person takes a small box out of a large box, takes a tea pot out of the small box, and shakes the tea pot.  <b>Target:</b> the right hand  <b>Aspect-of-target:</b> &lt;movement&gt;  <b>Criterion:</b> Whether it is easy to see the hand motion.</p>
	<p><b>Contents:</b> A person controls a video camera by using a remote controller while walking around a desk.  <b>Target:</b> &lt;speaker&gt;  <b>Aspect-of-target:</b> &lt;circumstance&gt;  <b>Criterion:</b> Whether it is easy to see how the person moved in his/her circumstances and what the person did.</p>

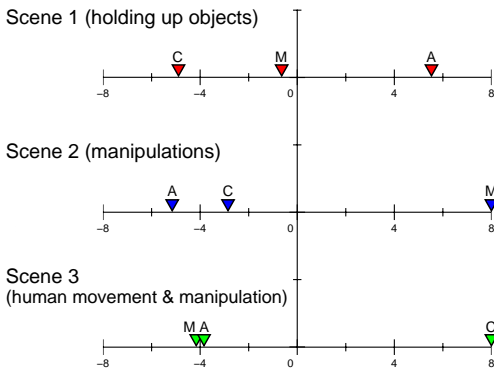


FIGURE 4 - Evaluation results by paired comparisons among shots obtained from cameraworks in virtual scenes. Each rectangle is placed according to the score calculated by Thurston’s method. The values of symbol A, M, and C represent cameraworks for <appearance>, <movement>, and <circumstance>, respectively.

We evaluated the shots obtained by virtually capturing scene1 through scene3 with cameraworks for <appearance>, <movement>, and <circumstance>. TABLE 7 shows the camerawork parameters used in this experiment. We showed 16 subjects several pairs of videos, and asked them to choose which is better based on the criteria as shown in TABLE 6.

**Discussion on categories of camerawork.** FIGURE 4 shows the result. The position of a rectangle shows the score obtained by the subjective evaluation, and greater values mean good shots. As shown in the figure, the cam-

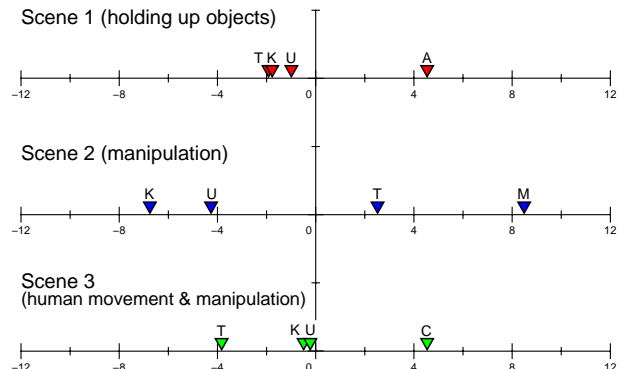


FIGURE 5 - Evaluation results by paired comparisons among shots obtained by our cameraworks and other typical methods in virtual scenes. The values of symbol U, T, and K represent the score of direct and delay-less tracking, the tracking with insensitive-area, and the adaptive smooth tracking, respectively. The values of symbol A, M, and C are explained in FIGURE 4.

erawork designed for each purpose (aspect-of-target) got the best score. The result means that choosing appropriate camerawork is really necessary, and our camerawork setting for each purpose is appropriate for it. This also implies that camerawork categorization contributes to taking good videos.

**Discussion on advantages over other methods.** For comparison, we captured virtual scenes with other typical camera control methods shown in TABLE 8: “direct and delay-less tracking (U)”, “tracking with insensitive-

TABLE 7 - Camerawork parameters for a virtual scene and a real scene.

Parameters	Virtual scene				Real scene		
	A	M	C	T	A	M	C
virtual-frame_size (ratio of a screen)	0.25	0.7	0.95	0.5	0.4	0.7	0.95
repetitive_count_th (times)	$\infty$	2	6	$\infty$	$\infty$	2	6
stationary_range (ratio of a screen)	0.2	0.4	0.5	0.5	0.2	0.4	0.5
stationary_time_th (seconds)	0.5	2.0	5.0	0.5	0.5	2.0	5.0
frame_refresh_interval (seconds)	$\infty$	2	12	$\infty$	$\infty$	2	12
frame_refresh_range (ratio to the frame_refresh_interval)	0	1.0	0.25	0	0	1.0	0.25
smoothing_degree (0-15)	15	10	5	15	12	7	3

The values of symbol A, M, and C represent camerawork parameters for <appearance>, <movement>, and <circumstance>, respectively. The value of symbol T represents camerawork parameters for the tracking with insensitive-area.

TABLE 8 - Other typical cameraworks.

#### **Direct and delay-less tracking (U):**

This is a simple and tracking under the condition of no measurement noise, no measurement delay, and no control delay. A camera constantly keeps the target at the center of the screen. This is the reference for checking if simple and direct tracking on ideal conditions takes good shots or not.

#### **Tracking with insensitive-area (T):**

To suppress irritating view field changes, a camera tracks a target with a certain tolerance area. If the apparent position of the target is inside of this area, the camera motion is eliminated. This area is placed at the center of the screen, and the size is set to 50 % of the screen<sup>8</sup>size. This is the reference for checking if repetitive target motion detection and stationary target position detection in our algorithm is necessary.

#### **Adaptive smooth tracking (K):**

The camera control by only adjusting the smoothing\_degree parameter without using any other functions of the virtual-frame control algorithm. This is the reference for checking if smoothing of camera motions by the Kalman filter is enough for taking good shots.

area (T)<sup>9</sup>”, and “adaptive smooth tracking (K)”. For example, “tracking with insensitive-area” is the reference for checking if repetitive target motion detection and sta-

<sup>9</sup>This camerawork simulates the camerawork with tolerance range of the target’s position that is reported by the NHK Science & Technical Research Laboratory as far as we can read (6)



Scene 1'                      Scene 2'                      Scene 3'

FIGURE 6 - Scene samples for an evaluation.

tionary target position detection in our algorithm is necessary. The obtained videos are compared with the videos captured by our virtual-frame control. Thurston’s paired comparison was also used for this evaluation.

FIGURE 5 shows the result. We can see that the videos captured by the virtual-frame control obtained the best scores out of all methods. It is also remarkable that videos taken by “direct and delay-less tracking (U)” did not get good scores even with no measurement errors and no delay. This strongly demonstrates that we need sophisticated cameraworks even under ideal conditions as given in computer graphics. Considering the results by “tracking with insensitive-area (T)” and “adaptive smooth tracking (K)”, a fixed size insensitive area or smoothness adjustment is not enough for capturing desktop presentations.

## 6.2. Experiments in Real Scenes

We implemented the virtual-frame control algorithm in our multi-camera system, and we had experiments of capturing various shots for three scenes. The three



scenes, that are scene1' through scene3' shown in FIGURE 6, have the similar situations as scene1 through scene3 shown in TABLE 6. Each of three scenes was simultaneously captured by four cameras located close to one another. The cameras were controlled by the camerawork for <appearance>, <movement>, <circumstance>, and simple tracking.

TABLE 7 shows the camerawork parameters used in this experiment. Since we have measurement errors and mechanical shakes, smoothing\_degree for all types of cameraworks is set larger than that for virtual scenes, and virtual-frame\_size of camerawork for <appearance> is also set larger. The camera control delay is around 10 frames<sup>10</sup>, and the average measurement error of the magnetic position is around 10 mm.

For this subjective evaluation, we gathered 17 subjects, and used the same method as in the experiment for virtual scenes. FIGURE 7 shows the result. The result is similar to that of the virtual video capturing. We, therefore, can safely say that our method is effective in both a real scene and a virtual scene.

For reference, sample shots taken for a desktop presentation “assembling a toy car” are shown in FIGURE 8. The images in the upper three rows are the shots taken by three cameras, whose targets are the right hand, the speaker, and the both hands, respectively. The aspect-of-targets are set <appearance>, <circumstance>, and <movement>, respectively. We can see that each camera well captured the target when the specified aspect-of-target was important. The bottom row of FIGURE 8 shows an example of an edited video by switching the views<sup>11</sup>. We can see that the shots automatically taken are good stuffs for composing comprehensible videos.

## 7. CONCLUSION

In this paper, we categorized typical types of shot that frequently appear in TV programs, and proposed a novel framework for taking these shots by an automated multi-camera system. We classified the purpose of a camerawork from two points of view: target and aspect-of-target, and considered the correspondence between shot purposes and cameraworks. Then, we proposed the virtual-frame control algorithm that realizes various cameraworks by changing its parameters, and verified the method through two kinds of experiments: virtual video capturing and real video capturing.

For future area for work, our goal is constructing a multimedia contents production system, which automatically

<sup>10</sup>The camera we used in this experiment is Sony EVI-D100. A target sometimes goes out of the screen because the speed of a hand motion turns up around 300 cm/sec in ordinary desktop manipulations.

<sup>11</sup>Our system automatically edits videos by utilizing speaker's behaviors. For more detail on automated editing, please refer to (11)

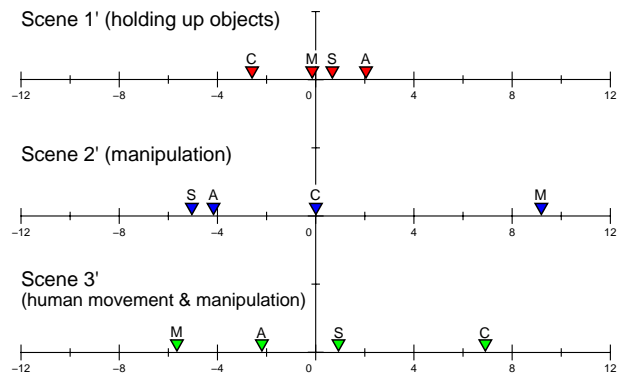


FIGURE 7 - Evaluation results by paired comparisons among shots captured by our system. The value of symbol S represent a simple tracking, and values of symbol A, M, and C are explained in FIGURE 4.

or semi-automatically captures presentations, gives indices to the obtained videos, and edit s and presents the videos to the users. In this sense, we need further research for automatic editing, indexing, and content production for various applications.

## Acknowledgements

In this paper, we used a portion of “Video Database for Evaluating Video Processing (VDB)” developed by VDB-Working Group with the Pattern and Media Understanding (PRMU)-Technical Group in Japan (12).

## REFERENCES

1. Ozeki, M., Itoh, M., Nakamura, Y., and Ohta, Y., 2002, “Tracking hands and objects for an intelligent video production system”, *Proc. ICPR*, 1011–1014
2. He, L., et al., 1999, “Auto-summarization of audio-video presentations”, *Proc.ACM Multimedia*, 489–498
3. Mukhopadhyay, S. and Smith, B., 1999, “Passive capture and structuring of lectures”, *Proc.ACM Multimedia*, 477–487
4. Kameda, Y., Minoh, M., et al., 2000, A study for distance learning service - tide project -, *Proc. ICME*, 1237–1240
5. Liu, Q., Rui, Y., Gupta, A., and Cadiz, J. J., 2001, “Automating Camera Management for Lecture Room Environments”, *Proc. of ACM CHI*, 442–449
6. Katou, D., Katsuura, T., and Koyama, H., 2000, “Automatic control of a robot camera for broadcasting based on cameramen’s techniques and subjective evaluation and analysis of reproduced images”,



FIGURE 8 - Shot samples captured by the cameras and an example of an edited video. Hand(R) means the right hand, and Hand(B) means the left hand. The phrases below the images are the transcribed speech. The upper line shows the actual speech in Japanese, and the lower shows the translation into English. The sample movies (mpeg format) can be obtained at [http://www.image.esys.tsukuba.ac.jp/~ozeki/e\\_movies.html](http://www.image.esys.tsukuba.ac.jp/~ozeki/e_movies.html).

J. Physiological Anthropology and Applied Human Science, 19(2), 61-71

7. Bobick, A. and Pinhanez, C., 1997, "Controlling view-based algorithms using approximate world models and action information", Proc. CVPR, 955-961
8. He, L., Cohen, M. F., and Salesin, D. H., 1996, "The virtual cinematographer: A paradigm for automatic real-time camera control and directing", Proc. SIGGRAPH 96, 217-224
9. Drucker, S. M. and Zeltzer, D., 1995, "Camdroid: a system for implementing intelligent camera control", Proc. Symposium on Interactive 3D Graphics, 139-144
10. Ohkushi, K., Nakayama, T., and Fukuda, T., 1991, "Evaluation Techniques for Image and Tone Quality", chapter 2.5, SHOKODO, Japan (in Japanese)
11. Ozeki, M., Nakamura, Y., and Ohta, Y., 2002, "Human behavior recognition for an intelligent video production system", Proc. PCM, 1153-1160
12. Babaguchi, N., et al, 2002, "Video database for evaluating video processing", Proc. Technical Report of IEICE, PRMU2002-30 (in Japanese)

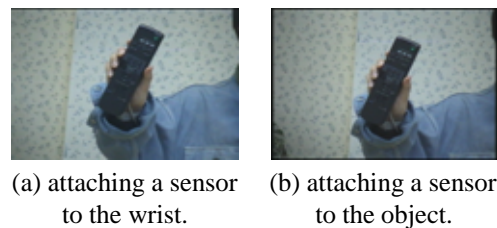


FIGURE 9 - Shot samples: tracking (a) a sensor attached to the wrist, (b) a sensor attached to an object.

**APPENDIX**

**Tracking Objects held by Hands**

Since objects appearing in desktop presentations frequently change their positions, shapes, or structures, it is hard to constantly measure positions of all objects. However, by tracking the hand position, the object can be captured when it is held by a speaker. Since objects do not usually move by themselves in ordinary desktop manipulations, this substitution can be a practical function for capturing a focused object.

We, therefore, let a camera track the hand for capturing a held object. FIGURE 9 shows the comparison between object tracking and hand tracking. In the case as shown in this figure, there is little difference between both schema. Further discussion would be necessary in other cases.