

Pace-the-Scene Movie: A Scene Representation Method for Walkthrough in Natural Scenery

Katsuyuki Kamei

Kazuo Seo

Industrial Electronics & Systems Laboratory,
Mitsubishi Electric Corporation

8-1-1 Tsukaguchi-honmachi, Amagasaki, Hyogo 661-8661 JAPAN

kamei@sys.crl.melco.co.jp

seo@sys.sdl.melco.co.jp

Abstract

In this paper, we propose a new scene representation method called *Pace-the-Scene Movie* which reproduces a natural, wide scenery angle in 3D software using video sequences. The fundamental functions of our method are as follows: (1) The software selects and shows source frames sequentially, which keeps the natural motions recorded in the source sequence. (2) The selected frames are transformed according to the user's viewpoint, in order to make them consistent with the actual views. (3) When the user's viewpoint moves away from that of the selected frame, the software searches for a new frame that has a nearby viewpoint and changes the source frames (skipping the source sequence), in order to reduce the effects of parallax. Our method can replay natural motions in a scene such as the swaying of branches and allow the user to walk randomly such as changing step or stepping to the side. We have examined the distortion of the generated view, and obtained the skip condition. An extension of our method is also described which uses multiple source video sequence streams. Our method is simple and useful for constructing virtual environments.

Key words: Virtual Reality, Computer Graphics, Walkthrough, Scene Modeling, Image-Based Rendering

1. Introduction

As studies of image-based rendering progress [1], many methods utilizing image sequences in order to represent a natural, wide scenery angle in 3D software have been tried. However, some of them have difficulty in showing objects in motion, because they reproduce the scene simply by selecting the frame only according to the user's viewpoint [2 - 4]. These methods regard an image sequence as a set of individual frames. On the other hand, others handle the image sequence as a movie and reproduce the scene based on a simple playback [5]. However, they have their own viewpoint-changing limits, that is, it is difficult for these methods to perform a random walkthrough operation in the re-

produced scene. Hence, the development of a method of representing natural moving scenes, such as a forest with swaying branches, is eagerly required for constructing natural environments.

This has been attempted by various methods. *Videoplex* [6] gives individual moving objects such as a walker to a scene by putting their video clips on it. *Step-wise motion-pictures* [7] successfully represents moving scenes of a forest by using multiple video streams shot from various viewpoints. Nevertheless, it is still difficult to perform a random walk through the scene. The *Peloton* system [8] represents natural scenes by employing *Pyramidic Planes* onto which video images are projected. By changing their shapes according to the user's viewpoint, this system actualizes an arbitrary walk through a scene shot by a video sequence, although it requires the calculation of the vanishing point in every image and a 3-D computer graphics system.

In this paper, we propose a practical scene representation method called *Pace-the-Scene Movie (PAS-Movie)*. This method can reproduce natural moving scenes using one or more streams of video sequences. Our method is simple and does not require the detailed geometrical information of the scene or complicated image processing. We regard the source video both as a movie and as a set of frames. Ordinarily, our method acts basically on the playback of the video sequence. When it fails to follow the user's path, it skips the source sequence by regarding it as a set of individual frames.

2. Pace-the-Scene Movie

2.1 Outline of the Method

The fundamental functions of our method using one video sequence (source movie) are as follows:

- (1) The software selects and shows source frames sequentially, which keeps the natural motions recorded in the movie.
- (2) The selected frames are transformed according to the user's viewpoint, in order to make them consistent with

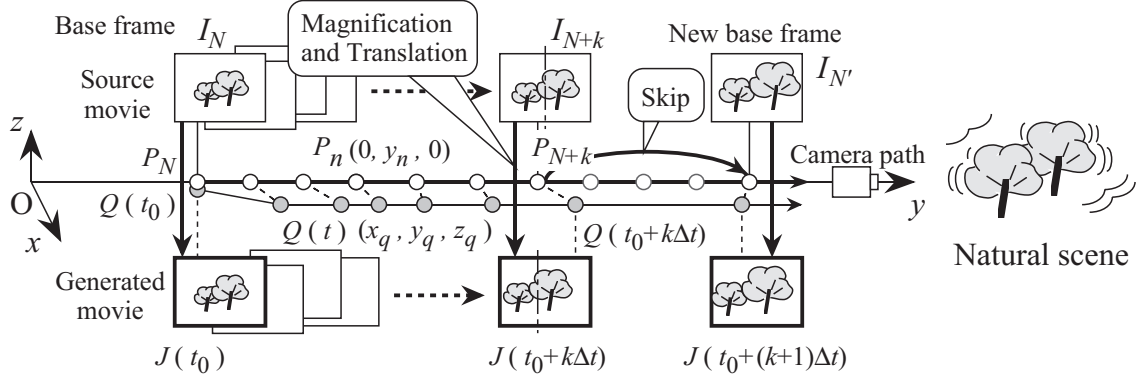


Fig. 1 The outline of our method.

the actual views.

- (3) When the user's viewpoint moves away from that of the selected frame, the software searches for a new frame that has a nearby viewpoint and changes the source frames (skipping the source sequence), in order to reduce the effects of parallax.

Figure 1 shows an outline of our method. The camera path of the source movie is located on the y -axis, and a source movie is shot with the camera moving forward. Here, I_n and $J(t)$ indicate the source frames ($n = 1, 2, \dots$) and the generated view, respectively. Here, t is the time parameter, and I_n is shot with the constant time step Δt . Also P_n and $Q(t)$ are the viewpoints from where I_n was shot and where the user is, respectively. Let $(0, y_n, 0)$ and (x_q, y_q, z_q) be the coordinates of P_n and $Q(t)$, respectively.

Suppose $Q(t_0)$ is located very near P_N at $t = t_0$, then I_N is selected to represent $J(t_0)$. We call this I_N the "base frame." Then, we put its succeeding frames to use sequentially:

$$J(t_0 + k \Delta t) \leftarrow F(I_{N+k}), \quad k = 0, 1, 2, \dots \quad (1)$$

where F indicates the transformation of I_{N+k} . This F makes I_{N+k} consistent with the actual view at $Q(t_0 + k \Delta t)$. For F , we employ only magnification or reduction (for changing step), and translation (for stepping to the side) of the frame because they are simple and do not cause serious distortions of the contents. Of course they are sufficient for the required modification, as well. When $Q(t)$ moves away from the selected P_n , we reset the base frame and skip the sequence. In this way, natural scenes capable of allowing changes in the path walked and the pace can be reproduced.

2.2 Transformation of the Selected Frames

The transformation is carried out as follows. As shown in Figs. 2 and 3, the selected frame is magnified (or reduced) and translated according to the user's position and the frame's viewpoint. In the strict sense, generating an ideal view at $Q(t)$ from I_n requires complete geometrical information regarding the scene, which is quite impossible. We therefore, simply approximate the view with the magni-

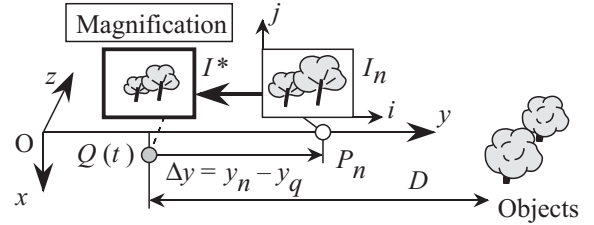


Fig. 2 Magnification of the selected frame.

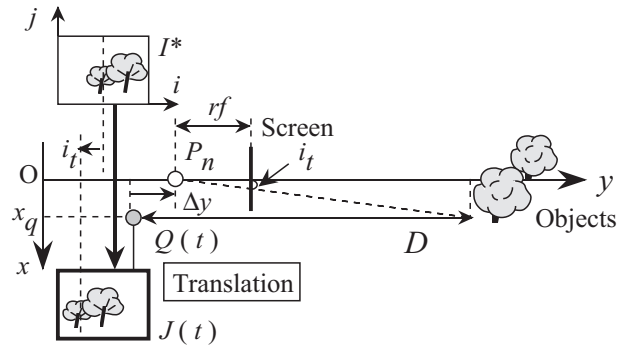


Fig. 3 Translation of the magnified frame.

fied and translated image, which doesn't spoil the fine touch depicted in the source frame.

When the user walks faster than the motion of the camera, $Q(t)$ gets ahead of the corresponding P_n . In this case, most objects in $J(t)$ must be displayed larger than their sizes in I_n , because $Q(t)$ is nearer to each object. On the other hand, when user walks slower, $Q(t)$ cannot keep up with the corresponding P_n . Thus objects in $J(t)$ must be smaller. Therefore, the selected I_n must be magnified or reduced. In order to determine the magnification power, a main but imaginary target, which is located at distance D from $Q(t)$, is assumed. The power is decided so that the target will be drawn the correct size on $J(t)$. An object's

size on an image is inversely proportional to its distance from the viewpoint. Therefore, the magnification power r is set as follows:

$$r = \frac{D - \Delta y}{D} \quad (2)$$

where $\Delta y = y_n - y_q$. This magnified image is represented as I^* , below.

Next, we translate the magnified image (Fig. 3). When the user steps to the side, their path runs off the y -axis, which makes the user's view different from I^* . In order to correct this, we generate the new image by translating I^* according to its distance to the y -axis. Here, we use the distance D , again. The translation value is determined so that the imaginary target will be depicted at the correct position on $J(t)$, as follows:

$$i_t = -\frac{x_q f}{D}, \quad j_t = -\frac{z_q f}{D} \quad (3)$$

where i_t and j_t are the translation values of the i -axis (rightward on images) and the j -axis (upward on images), respectively, and where f is the focal length.

Here, we have examined the transformation focusing on the imaginary target. Generally, most objects are not located in the same place as the target, which may cause the distortion of the generated view. We examine this distortion in the next section.

2.3 Skipping the Video Sequence

If $|\Delta y|$ becomes larger, the magnification power grows much larger or smaller than 1, which causes a decrease in the accuracy and quality of the generated view, even though the transformation described above is carried out. In these cases, the base frame should be replaced to a new one ($I_{N'}$) which has the minimum value of $|y_{N'} - y_q|$ for the current user's viewpoint (Fig. 4). Then the source sequence is skipped to the new base. When the user's viewpoint $Q(t)$ goes ahead (that is, N' is larger than the current frame number n), the sequence is skipped forward and some frames are not used. On the other hand, when $Q(t)$ gets behind (that is, $N' < n$), the sequence is skipped backward and some frames are used again.

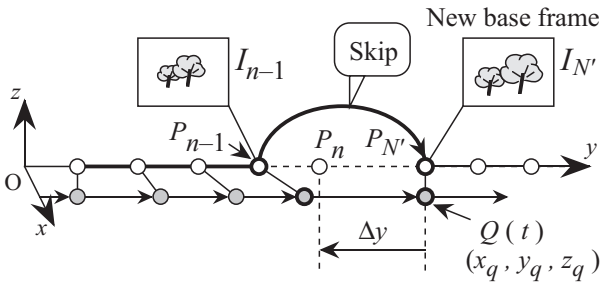


Fig. 4 Skip of source sequence to the new base frame.

3. Examination of Distortion

3.1 Distortion Caused by difference of $Q(t)$ from P_n

By examining the distortion of the generated view $J(t)$, we consider how far the objects are drawn on $J(t)$ from their correct positions. As shown in Fig. 5, let A be an object in the scene and (x, y, z) its coordinates. When the position of $Q(t)$ is not same as P_n , object A may be drawn at the wrong position on $J(t)$. We consider this displacement between the desired position of A and its actual position drawn on $J(t)$. For simplicity, we assume $Q(t)$ is located on the zx -plane ($y_q = 0$), which does not restrict the generality of the discussion.

First, the displacement in the i -direction is considered. Let Δi be the value of subtracting the correct coordinate of A from its the actual drawn coordinate. As $J(t)$ is generated by the magnification and translation described above, the actual i -coordinate is obtained by

$$\frac{r x f}{y - \Delta y} - \frac{x_q f}{D} \quad (4)$$

The correct coordinate is given as follows:

$$\frac{(x - x_q) f}{y} \quad (5)$$

We get the following equation for Δi :

$$\begin{aligned} \Delta i &= \left(\frac{r x f}{y - \Delta y} - \frac{x_q f}{D} \right) - \frac{(x - x_q) f}{y} \\ &= \frac{(D - y) x f \Delta y}{(y - \Delta y) y D} + \frac{(D - y) x_q f}{y D} \end{aligned} \quad (6)$$

If we set

$$\begin{aligned} \Delta i_1 &= \frac{(D - y) x f \Delta y}{(y - \Delta y) y D} \\ \Delta i_2 &= \frac{(D - y) x_q f}{y D} \end{aligned} \quad (7)$$

then we obtain the following inequality:

$$|\Delta i| \leq |\Delta i_1| + |\Delta i_2| \quad (8)$$

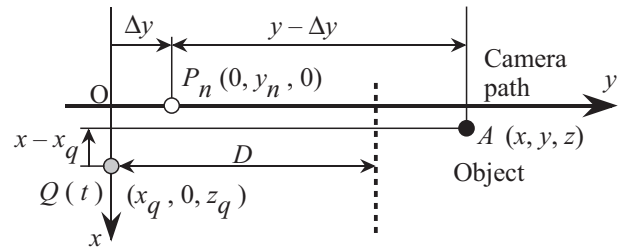


Fig. 5 Configuration of viewpoints and object.

This means that the absolute value of Δi consists of two elements: the first element $|\Delta i_1|$ is caused by the user's pace change (Δy), and the second element $|\Delta i_2|$ is caused by the user's path change (x_q). The maximum value of $|\Delta i|$ is obtained by the sum of each maximum value. We examine each of them in the following sub-sections.

3.2 Distortion Caused by User's Pace Change

Equation 7 means that $|\Delta i_1|$ becomes greater as $|x|$ becomes larger. But, in the actual scene, distant objects are often occluded by nearer ones. Considering this, we assume that all visible objects fall in $-W/2 \leq x \leq W/2$, which often occurs when walking on a road with a width W , as shown in Fig. 6. In this case, $|\Delta i_1|$ reaches a maximum ($\Delta i_{1, x=W/2}$) when $x = \pm W/2$ for each y value. This $\Delta i_{1, x=W/2}$ has a local maximum $\Delta i_{1, \max}$ (Fig. 7):

$$\Delta i_{1, \max} = \frac{(Y-D) W f |\Delta y|}{2(Y-\Delta y) Y D} \quad (9)$$

when y is the following Y :

$$Y = D + \sqrt{D(D-\Delta y)} \quad (10)$$

This $\Delta i_{1, \max}$ is the maximum value of $|\Delta i_1|$ when $y \geq D$. That is, the displacement of objects located farther than D never exceeds $\Delta i_{1, \max}$. Of course, objects nearer than D may cause larger displacements. However, those objects might be located on the roadside, so they may often be out of the field of view of $J(t)$ or drawn at both ends of $J(t)$, which means they do not have much effect on $J(t)$.

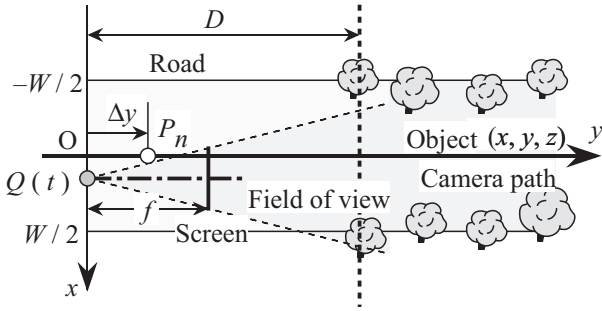


Fig. 6 Assumption of objects' location.

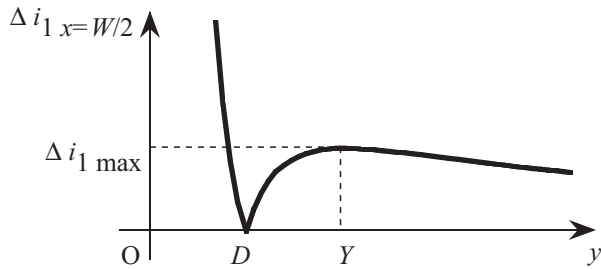


Fig. 7 The maximum value of $|\Delta i_1|$ for each y .

3.3 Distortion Caused by User's Path Change

Next, we consider $|\Delta i_2|$ caused by user's path change. As we assume $Q(t)$ is located on the zx -plane here, the path change is restricted to a change of x_q (x value of $Q(t)$). This $|\Delta i_2|$ becomes 0 when $y = D$, and gets larger and converges to $|x_q| f / D$, when y becomes larger than D . Therefore, the second element $|\Delta i_2|$ has the maximum value of $|x_q| f / D$ for objects more distant than D . Though objects within D may cause larger displacement, they do not influence $J(t)$ so much, as described above.

In this way, we obtain the following inequality when $y \geq D$:

$$|\Delta i| \leq \frac{(Y-D) W f |\Delta y|}{2(Y-\Delta y) Y D} + \frac{|x_q| f}{D} \quad (11)$$

This examination about the distortion of the i -direction (horizontal direction) of images is also good for that of the j -direction (vertical direction), and for when $Q(t)$ moves vertically.

4. Skip Condition

By examining the distortion of $J(t)$, we have determined the condition regarding the skip of the source sequence. Only Δy , which effects the maximum of the first element $\Delta i_{1, \max}$, is related to skipping the sequence. Therefore, we determine the tolerance limit for $\Delta i_{1, \max}$, and obtain a condition by determining the range of Δy which satisfies the tolerance limit. Let E_1 be the tolerance limit (in pixels). From Equation 9 and $\Delta i_{1, \max} \leq E_1$, we get the following condition for Δy :

$$-\Delta y_{\min} \leq \Delta y \leq \Delta y_{\max} \quad (12)$$

where

$$\begin{aligned} \Delta y_{\min} &= \frac{8 W f D^2 E_1}{(W f - 2 D E_1)^2} \geq 0 \\ \Delta y_{\max} &= \frac{8 W f D^2 E_1}{(W f + 2 D E_1)^2} \geq 0 \end{aligned} \quad (13)$$

This is just the skip condition. While Δy satisfies this condition, the distortions on $J(t)$ are generally within E_1 pixels, therefore the source frames are continued to be selected and shown sequentially. Otherwise, the source movie must be skipped forward or backward.

5. Extension to Using Multiple Source Streams

In order to represent various walking patterns, we employ the policy of using multiple streams of source movies. The most suitable source movie is chosen according to the path and the pace of user's viewpoint $Q(t)$, and its frames are selected and shown in the manner described above. We have also examined the path and the pace of the source movies, and obtained the principle of shooting them with a constant path interval and at a constant speed increase for efficient performance, as described below.

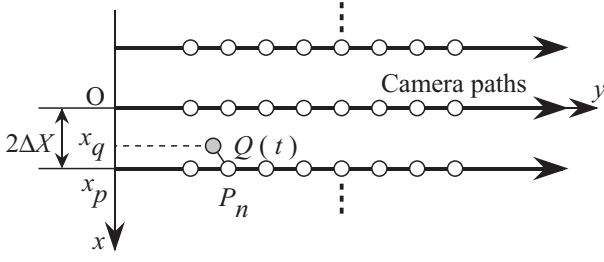


Fig. 8 Extension to employing multiple shooting paths.

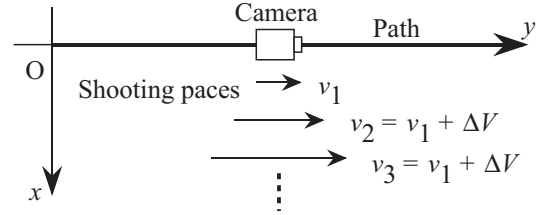


Fig. 9 Extension to shooting movies at various paces.

5.1 Paths of Source Movies

We obtained the skip condition by examining Δi_1 , as described in the previous section. Here, we examine the camera paths on which the source movies are shot by considering Δi_2 . We assume that the paths are linear and parallel to the y -axis. Let E_2 be the second tolerance limit (in pixels) for the converged value $|x_q| f/D$ of the second element $|\Delta i_2|$. Then, we get

$$|x_q| \leq \frac{D E_2}{f}. \quad (14)$$

Let ΔX be the right side of this inequality. In order to satisfy this inequality every time, the source movies must be shot in parallel to each other with a constant interval of $2\Delta X$, as shown in Fig. 8. With this configuration, $J(t)$ is generated by using the source movie which is shot on the nearest path (whose x value is x_p) to $Q(t)$, then the difference of the x coordinates $|x_q - x_p|$ becomes less than or equal to ΔX , which puts $|\Delta i_2|$ under the tolerance limit E_2 .

5.2 Paces of Source Movies

We now examine the paces of the source movies using the skip condition described above. Let v_q and v_p be user's walking pace and the velocity of the camera shooting the scene, respectively. Let τ be the time between skips. If v_q and v_p are constant, τ is calculated using Equation 12: time from one skip (when Δy is nearly reset to 0) to the moment when Δy goes off the condition, as follows:

$$\tau = \begin{cases} \frac{\Delta y_{\min}}{(v_q - v_p)}, & \text{If } v_q > v_p \\ \frac{\Delta y_{\max}}{(v_p - v_q)}, & \text{Otherwise.} \end{cases} \quad (15)$$

If $v_q > v_p$, that is, the user's pace is faster than the source movie's, a skip forward occurs. On the other hand, if $v_p > v_q$, a skip backward occurs.

A small τ causes frequent skips in the generated movie, which may decrease its quality. Therefore, in order to en-

sure τ is not smaller than a certain time T for every user's pace v_q , we employ a method using several source movies which are shot on the same path but with different paces. The limitations of the difference of both paces which gives adequate skip intervals is obtained by setting $\tau \geq T$ in Equation 15,

$$|v_p - v_q| \leq \begin{cases} \frac{\Delta y_{\min}}{T}, & \text{If } v_q > v_p \\ \frac{\Delta y_{\max}}{T}, & \text{Otherwise.} \end{cases} \quad (16)$$

From this limitations, the pace increase of the source movies ΔV is determined as follows:

$$\begin{aligned} \Delta V &= \frac{\Delta y_{\min}}{T} + \frac{\Delta y_{\max}}{T} \\ &= \frac{8 W f D^2 E_1}{T (W f - 2 D E_1)^2} + \frac{8 W f D^2 E_1}{T (W f + 2 D E_1)^2}. \end{aligned} \quad (17)$$

When source movies are shot with this constant pace increase ΔV (Fig. 9), there will be a movie whose pace v_p should satisfy the condition of Equation 16 for any walking pace v_q . In this way, we can make a movie whose skip interval never becomes less than T for every v_q .

6. Experimental Results

Figure 10 shows examples of movies generated from a video sequence shot on a tree-lined pavement. The walking pace of each generated movie is set at twice (Fig. 10b) and half (Fig. 10c) as fast as that of the source movie (Fig. 10a). The source and generated views at $t = 0, 40\Delta t, 80\Delta t, 120\Delta t$ are shown. Fast walks and slow walks are successfully represented in Figs. 10b and 10c, respectively. From these views, the viewpoint of I_{80} of Fig. 10a is the same as that of $J(40\Delta t)$ of Fig. 10b, and the viewpoint of I_{40} of Fig. 10a is the same as that of $J(80\Delta t)$ of Fig. 10c. In each case, both views are the same and seem to be the same in fact. The magnifying power r of each frame is also shown.

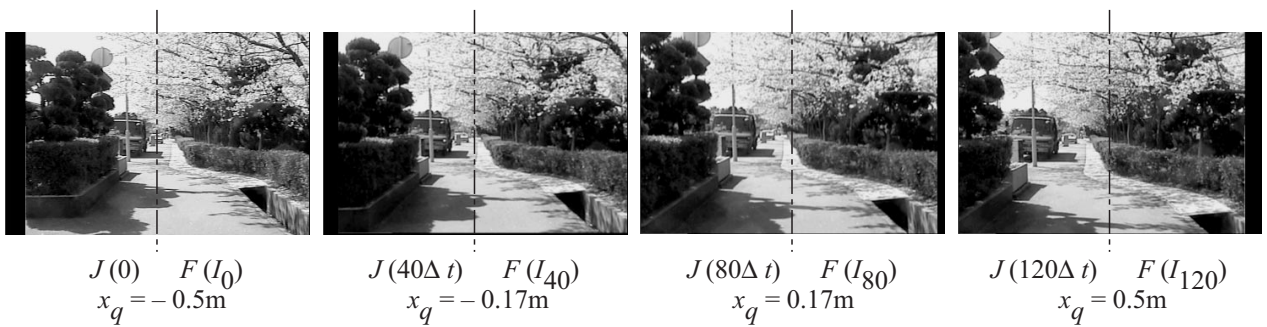
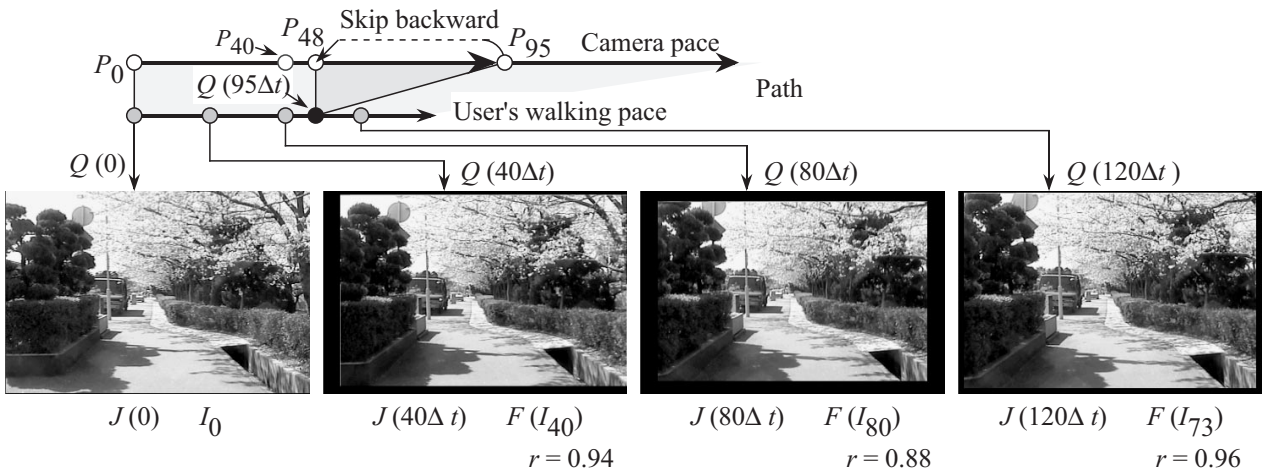
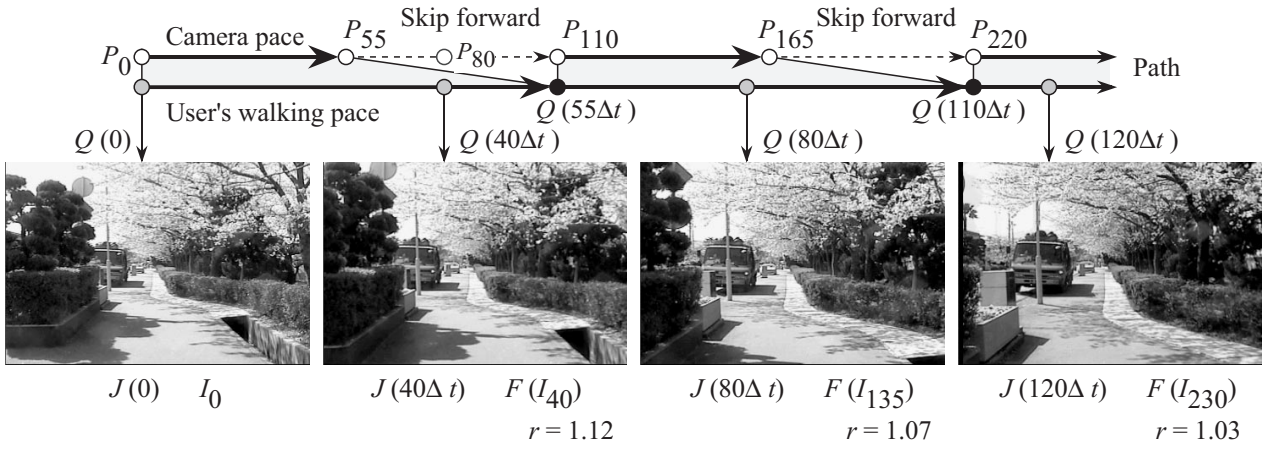
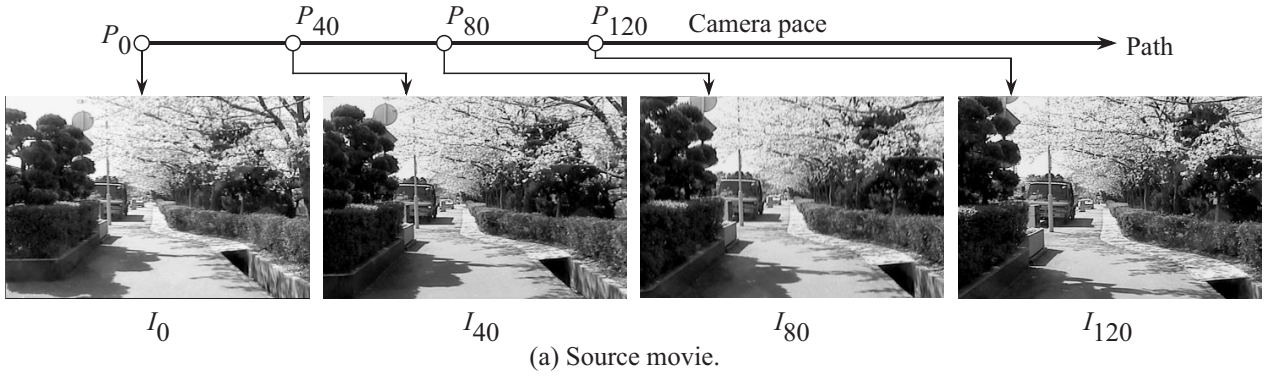


Fig. 10 Experimental results of scene reproduce.

Table 1. Parameters used in the experiment.

Items	Values
Size of images	360 x 240
f	457
Δt	1 / 30 sec.
W	4.80 m
D	10.0 m
E_1	4

Figure 10d shows an example of the user's stepping from left to right at the same pace as that of the source movie. In all cases, adequate movies are generated in which the key movements such as sway of branches or the blinkers of the truck can be represented naturally. The parameters used in this experiment is shown in Table 1. The skip condition of Equation 12 becomes $-1.57 \text{ m} \leq \Delta y \leq 1.36 \text{ m}$. The skips are not so noticeable and can be smoothed by dissolving. In the example of Fig. 10d, the converged value $|x_q| f / D$ becomes 22.9 at $|x_q| = 0.5 \text{ m}$. We used the IRIS Onyx2 InfiniteReality for this experiment which was enough to process the view generation in the time step Δt .

7. Discussion

Our method is basically based on a sequential playback of the source movies, which guarantees the photo-quality of the views generated. If we have to point out a problem of our method, it must be the discontinuity of view which occurs when skipping or changing source movies. Fortunately, our experiments show they have little effect, but they may have more influence in other cases. In particular, at every end of $J(t)$, large displacements are likely to occur, which may generate remarkable discontinuities. Here, we consider four cases of skipping and changing movies: skip forward, skip backward, change to the left movie and change to the right movie. Of course there are also changes to the movies above or below, but they are similar to the cases of changes to the left or the right.

To begin with, when skipping, the positions of drawn objects may jump incorrectly on $J(t)$. This jump distance in the i -direction (i_{jump}) is described as follows from Equation 6 (we assume that the source movie is shot on the y -axis and that P_n may become the same as $Q(t)$ after skipping):

$$\begin{aligned}
 i_{\text{jump}} &= \left(\frac{x f}{y} - \frac{x_q f}{D} \right) - \left(\frac{r x f}{y - \Delta y} - \frac{x_q f}{D} \right) \\
 &= \frac{(y - D) x f \Delta y}{(y - \Delta y) y D}
 \end{aligned} \tag{18}$$

We focus on noteworthy objects which have small y values ($\Delta y < y < D$ and $y > 0$) and may give large displacements on $J(t)$. Therefore Equation 18 means that if $\Delta y < -y_{\text{min}}$ and is skipping forward, the objects may jump outward, because the sign of i_{jump} becomes the same as that of x . This is the same as the j -direction. When $\Delta y > y_{\text{max}}$ and is skipping backward, the jump of those objects is inward, as the sign of i_{jump} becomes opposite to x .

Next, when changing to the left or right movies, the jumps are represented as follows:

$$\begin{aligned}
 i_{\text{jump}} &= \left(\frac{(x - x_{\text{new}}) f}{y} - \frac{(x_q - x_{\text{new}}) f}{D} \right) \\
 &\quad - \left(\frac{(x - x_{\text{old}}) f}{y} - \frac{(x_q - x_{\text{old}}) f}{D} \right) \\
 &= \frac{(D - y) (x_{\text{old}} - x_{\text{new}}) f}{y D}
 \end{aligned} \tag{19}$$

where x_{old} and x_{new} are x values of paths of the current and the newly chosen movies, respectively. We assume that $\Delta y = 0$. This indicates that the objects jump to the right ($i_{\text{jump}} > 0$) when $x_{\text{old}} > x_{\text{new}}$ (stepping to the left) and to the left ($i_{\text{jump}} < 0$) when $x_{\text{old}} < x_{text{new}}$ (stepping to the right).

These results are arranged in Fig. 11. This figure shows that the jumps when skip forward, change to the left and change to the right, agree with view flow when actually walking forward, to the left and to the right. Therefore,

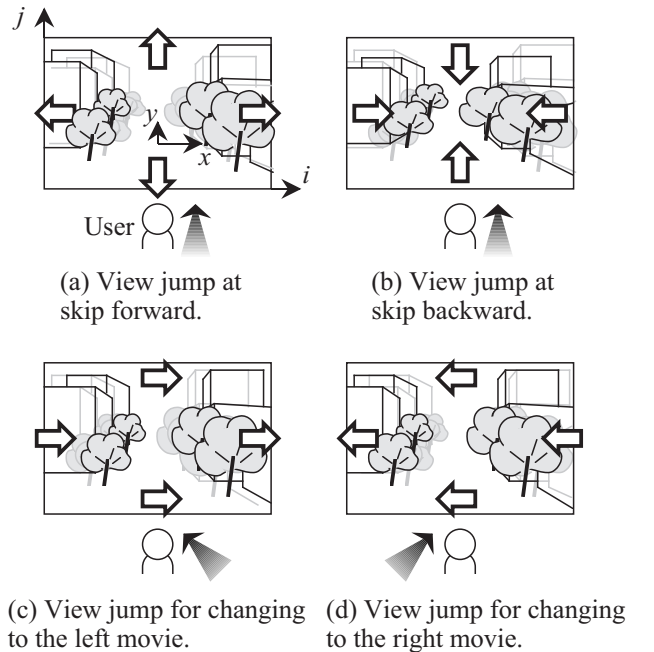


Fig. 11 View jump when skipping sequence and changing movies.

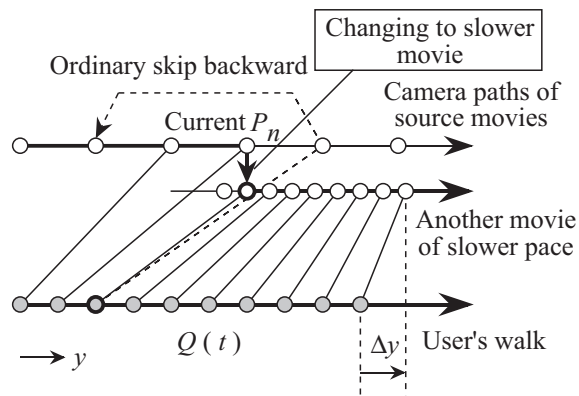


Fig. 12 Alternative method for skipping backward which uses a slower movie.

they give no feeling of unease to users. But skipping backward may give users the feeling of stepping back, although they are walking forward.

There are some ways of dealing with the problem of skipping backward. The first solution is to make the skip condition more rigid when skipping backward. That is, y_{\max} is given a smaller value. Another solution is to change movies to one shot more slowly, if movies of various paces are available. Figure 12 shows this change. When skipping backward, we select a new base frame in the slower movie on the path, whose viewpoint is nearby not to $Q(t)$ but to the current selected P_n . In this way, the generated view $J(t)$ does not have jumping objects because the viewpoints before and after the change are near to each other, and Δy decreases because the pace of the new source movie is slower than the user's. This technique can also be applied to skipping forward.

8. Conclusion

We have developed a new scene representation method called *Pace-the-Scene Movie*. It represent natural moving scenes using one or more streams of video sequences. A view from an arbitrary user's position is generated by selecting the best frame among them and transforming it. We have also made the skip condition and the style of shooting source movies clear. Because of the peculiarities described above, our *Pace-the-Scene Movie* easily reproduces natural scenes in which a walkthrough operation along a path can be performed. Our method is simple, practical and useful for constructing virtual environments.

Here, we have described our method with assuming that the paths of the user and the source movies are parallel to the y -axis, and that the user walks forward on a road. We plan to extend this method to more arbitrary directions including the user stopping, turning and crossing a road. Panning and tilting may also be easily performed by using wide

or panoramic source movies. For a freer walk, however, a more advanced principle must be introduced. In the future, we intend to attack this point and apply our *Pace-the-Scene Movie* to practical systems.

References

- [1] S. E. Chen: "QuickTime VR - An Image-Based Approach to Virtual Environment Navigation," *proceedings of ACM SIGGRAPH'95*, pp. 29-38 (1995).
- [2] A. Lippman: "Movie Maps: An application of the optical videodisc to computer graphics," *proceedings of ACM SIGGRAPH'80*, pp. 32-43 (1980).
- [3] M. Hirose, K. Takahashi, T. Morinobu, T. Koshizuka, and Y. Watanabe: "An Alternative Way to Generate Virtual World: A Study of Image Processing Technology for Synthetic Sensations," *Presence*, Vol. 5, No. 1, pp. 61-71 (1996).
- [4] M. Hirose, Y. Watanabe, and T. Endo: "Generation of Wide-Range Virtual Spaces Using Photographic Images," *proceedings of VRAIS'98*, pp. 234-241 (1998).
- [5] M. Hirose, M. Taniguchi, Y. Nakagaki, and K. Nihei: "Virtual Playground and Communication Environment for Children," *NETWORKED REALITY'94* (1994).
- [6] G. Nishimura, T. Kihara, and R. Kataoka: "Videoplex: A New System Framework for Constructing Video-Based Three-Dimensional Space," *Lecture Notes in Computer Science*, Vol. 1554, pp. 289-300 (1999).
- [7] M. Ohsuga, Y. Tatsuno, F. Shimono, K. Hirasawa, H. Oyama, and H. Okamura: "Development of a Bedside Wellness System," *CyberPsychology & Behavior*, Vol. 1, No. 2, pp. 105-112 (1998).
- [8] G. U. Carraro, J. T. Edmark, and J. R. Ensor: "Techniques for Handling Video in Virtual Environments," *proceedings of ACM SIGGRAPH'98*, pp. 353-360 (1998).