芸術科学会論文誌 Vol. 13, No. 3, pp. 154-168

Datamoshing Technique for Video Art Production

Yuichi Ito*　Carl Stone　Masashi Yamada**　Shinya Miyazaki****

*Graduate School of Computer and Cognitive Sciences, Chukyo University
**School of Engineering, Chukyo University

yuichi.8445@gmail.com, {carl, myamada, miyazaki}@sist.chukyo-u.ac.jp

Abstract

In recent years, a number of pieces of art that were created through noise and error for artistic representation have been exhibited. Many of these are classified as glitch art, and they have formed a new trend in media art. In the field of glitch art, there is a technique called datamoshing that uses noise or errors in video imagery. One can produce a video that can serve as a unique representation of datamoshing; however, there is no clear principle for this technique, nor any citable literature to date. In this paper, we clarify the principle of datamoshing and the type of effects that can be generated. In addition, we propose software with which to apply datamoshing to videos and show some pieces of art that have been produced with the software. Furthermore, we discuss the significance of these works from the perspective of media art.

1. Introduction

Recent new media art has utilized a type of expression technique called a glitch. A glitch is a problem or defect in a machine or software program, or an abnormal sudden failure of equipment that uses electric power. For example, when a video game is started on a machine where the game cartridge has been inserted incorrectly, a blocky image error as the one shown in Figure 1-1 might be displayed. Glitch art pieces are made of noise and errors that would normally be eliminated during the production or post-production of a piece of art. A specific glitch technique, datamoshing, is able to intentionally generate materials using motion compensation inter-frame prediction errors that are sometimes the by-product of video compression techniques [1]. The term datamoshing is a compound term that is created from the words "data" and "moshing," where moshing is a slang term that originated from the action of individuals pushing and slamming each other aggressively, usually at live concerts.

Currently, the only book that has been published on the topic of glitches is *Glitch: Designing Imperfection*, written by Moradi et al. [2]. In this book, the authors indicate that:

> *"Visual glitches are rare in standard occurrence and yet paradoxically, they are quite easy to provoke, which makes them ideal for being appropriated, used as a medium for artwork or even elevated to the status of a genre. Imperfections in diamonds are affectionately called "inclusions." Outside of art and design, however, glitches don't get off the hook so easily - they are branded as bad reception, undesirable fuzz, static or interference.*
>
> *As a complete contrast, however, to some, visual glitches have become the rare diamonds and objects of fetish and desire."*

The Japanese programmer and filmmaker ucnv has noted that artists who understand the inner workings of computer file formats have the ability to create errors and re-express them as glitch art [3]. Therefore, we can make a glitch (rare diamonds) positively using information technologies. When authors wish to create works that represent the artist's intention, we want to be able to control even the expressions made of errors, while minimizing chance. Moreover, we would like as artists to reveal how to reproduce errors and noise. In this way, we agree with the position of ucnv regarding glitch art.

Glitch techniques are not structured, and there is no guidance in the literature. In addition, it is conceivably important for glitches to be controllable to embody the intention of artists. This paper, which focuses on video-targeted datamoshing in the glitch area, introduces a general approach. The feasibility of the expression of datamoshing is also shown through the authors' productions.

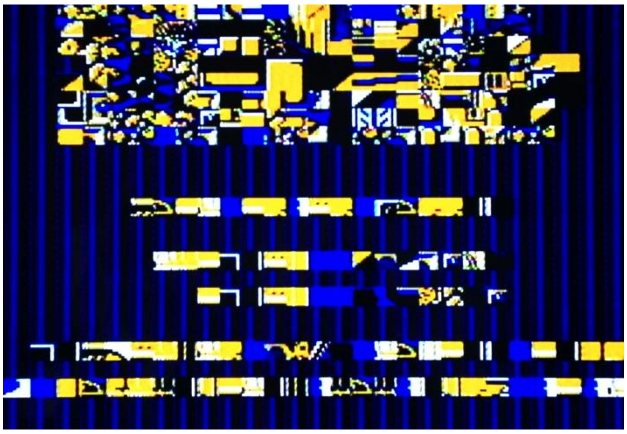


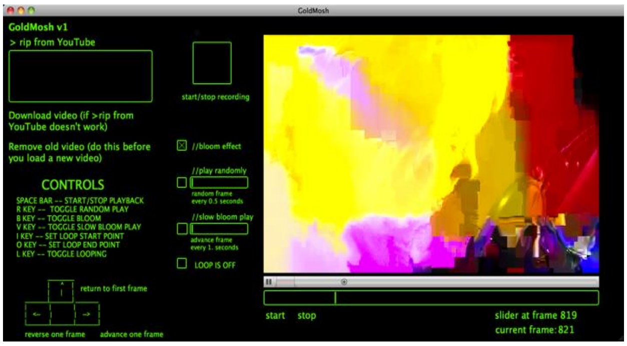
Figure 1-1. Video game glitch

2. Precedents of datamoshing

In 1965, Nam June Paik published a piece of work called *Magnet TV* that placed a magnet on top of a television set to distort the image [4], thus taking an analog approach to generate errors; in a sense, this can be considered proto-glitch art. A filmmaker, Stan Brakhage, has created pieces where images change extremely quickly, without respect to conventional continuity. It is impossible to see each of the images, and the functional original meaning of the images is lost. Thus, this is a type of glitch. Although each of the images is displayed for just one moment, the entirety of the film is extremely impressive. Around the same time, another filmmaker, Jonas Mekas, created image pieces by applying noise frequently, such as camera shake or shooting mistakes, which are usually removed from normal cinema grammar. These are examples of proto-glitch with analog film. Later, a video artist, Scott Arford, experimented with various ways of playing back video abnormally, including shifting video data into the audible spectrum and presenting such data as music. All the artists mentioned in this paragraph reversed the conventional artistic style and explored options of new aesthetic expressions in the visual domain [5].

As mentioned in Section 1, the book *Glitch: Designing Imperfection* was published in 2009, featuring several still works derived from video imagery that belong to post-Nam June Paik video art. Through events such as the conference *GLI.TC/H* held at Roxaboxen Exhibitions and the SAIC Flaxman Theater in Chicago, USA, in 2010, glitch art has been recognized as one of the many artistic fields of study.

In recent years, several datamoshing pieces have been published on the Internet, including *Compression Reel* [6] by David OReilly , and *"Hip Hop no Keinen Henka" no Keinen Henka* [7] by ucnv. It is not difficult to find videos that use



GoldMosh

GoldMosh v1

> rip from YouTube

Download video (if >rip from YouTube doesn't work)

Remove old video (do this before you load a new video)

CONTROLS

SPACE BAR -- START/STOP PLAYBACK
R KEY -- TOGGLE RANDOM PLAY
B KEY -- TOGGLE BLOOM
V KEY -- TOGGLE SLOW BLOOM PLAY
I KEY -- SET LOOP START POINT
O KEY -- SET LOOP END POINT
L KEY -- TOGGLE LOOPING

☒ //bloom effect
□ //play randomly
random frame every 0.5 seconds
□ //slow bloom play
advance frame every 1. seconds
□ LOOP IS OFF

slider at frame 819
current frame: 821

Figure 2-1. *GoldMosh* interface

datamoshing techniques[1]. The software *GoldMosh*[2] published by Sam Goldstein in January 2013 (Figure 2-1) enables users to attempt datamoshing easily and without knowledge of video compression formats.

3. Principle of datamoshing

Datamoshing is a technique for producing videos that contain one or more unique expressions from applied errors in the motion compensated inter-frame prediction used in video encodings, such as MPEG-4. In this paper, the term *datamoshing effect* means the unique expression created as a result of applying datamoshing to videos. Datamoshing has two main steps. The first step disables the reference frames to create errors in the motion-compensated inter-frame prediction, where the reference frames are the frames retained as still images in the process of video encoding. The second step controls the frame playback of a video.

3.1 Motion compensated inter-frame prediction

First, we describe the principle of motion-compensated inter-frame prediction. Motion in video is created by changing images (frames), usually at a set frame rate. In a typical movie, images change at a rate of 24 frames per second. Successive images tend to have high similarity, barring unexpected developments, such as scene changes. Therefore, it is possible to predict the images that will be displayed from the differences between the information in a given frame and the reference frame, specifying the reference frame at arbitrary intervals, unless a frame changes significantly. By storing only the differences, as shown in Figure 3-1, it is possible to greatly reduce data amounts. This technique is called inter-frame prediction, which is further combined with the technology of motion compensation. In motion compensation, the movement of image components between frames in a video, such as from a camera panning from left to right, is expressed by encoding the motion vectors of said image components [8]. Figure 3-2 is an example of motion vectors.

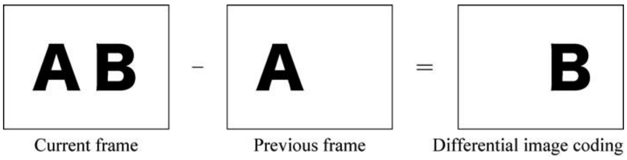


Figure 3-1. Differential image coding

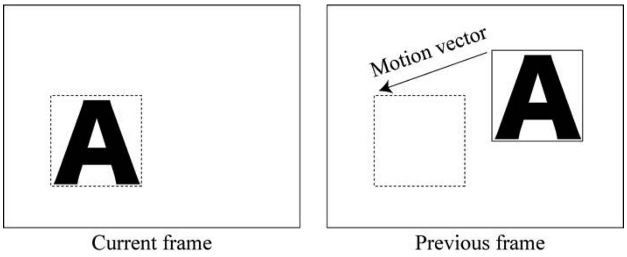


Figure 3-2. Example of motion vectors

3.2 Reference frame disabling

When applying data compression using motion-compensated inter-frame prediction to a video, the video frames are classified into one of three types: Intra-frame (I-frame), Predicted frame (P-frame), and Bidirectional predicted frame (B-frame) [9]. The I-frame is also sometimes called the reference frame, as described in the previous section. Encoding formats such as MPEG-4 generally retain image information as still images. The P-frame predicts images in reference to an I-frame that occurred earlier; however, it does not retain all image information. The B-frame predicts forward and backward images, or both directions, on the basis of either the I-frame or the P-frame. Figure 3-3 shows the relationship between the frames.

For this study, we performed a datamoshing experiment as follows. A video file is opened in a binary editor, and the code that defines an I-frame as reference to B-frames and P-frames is removed. If the I-frame cannot be recognized when playing the video, the playback software (e.g., QuickTime) is unable to refer to the I-frame (see Figure 3-4). Although the software cannot predict correct images because of the absence of reference frames, it attempts to continue playing back the video. The calculated video playback will be substantially different from the original. A simple technique for disabling the I-frame reference is to delete the I-frame code values in the encoded video file using a binary editor. However, if instead of deleting, a different value is substituted for the I-frame reference, the playback software

[1] datamosher: HOW TO DATAMOSH: PART 1 <http://youtu.be/tYytVzbPky8> (accessed 2013-10-30)

[2] GoldMosh!: <http://jonesypop.com/goldmosh/> (accessed 2013-10-30)

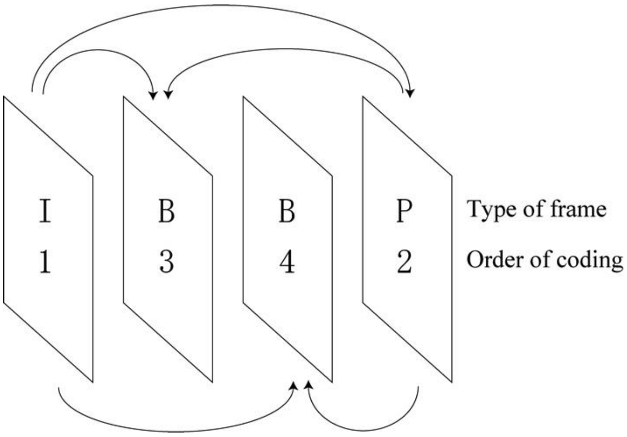


Figure 3-3. Relationship between I-frame, P-frame, and B-frame

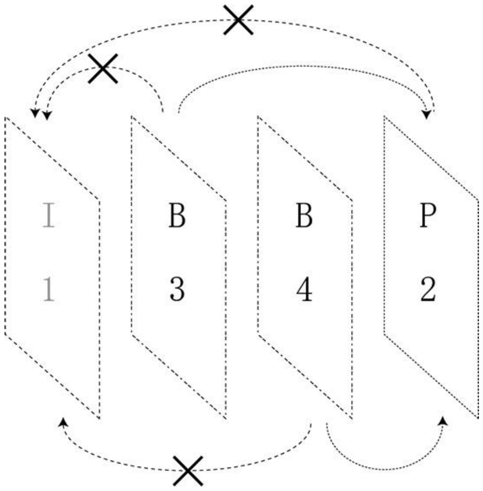


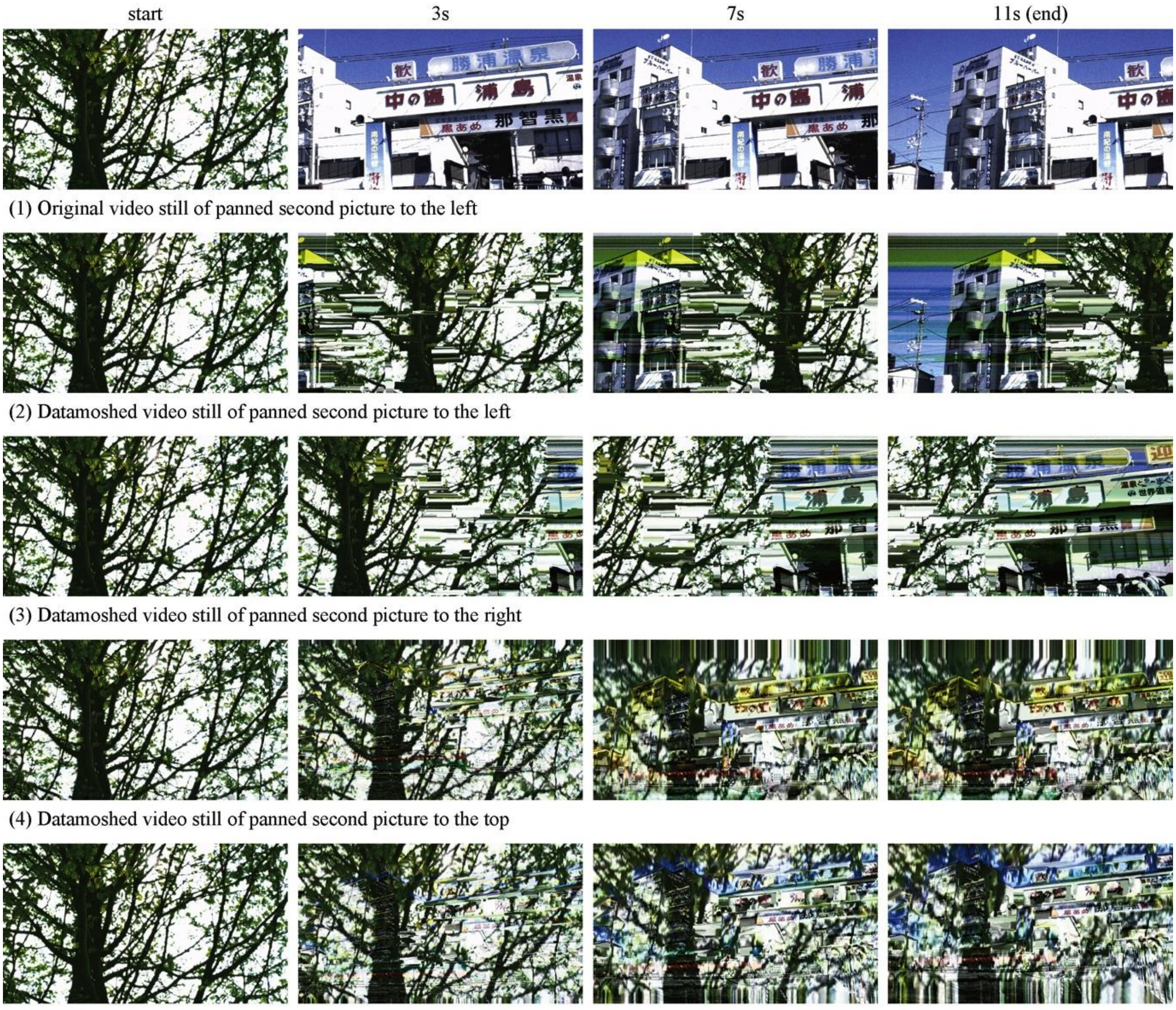
Figure 3-4. No reference to I-frame for P-frame and B-frame

might not recognize the file format correctly and playback might become impossible. With respect to inserting new I-frame information, we verified through multiple experiments a method for achieving a higher probability of playing back the video file to the end while creating errors. In general, inserting a minimal number of I-frames in the setting of the video encoder maintains a high consistency for each video[1]. Therefore, we confirmed that we are able to avoid disabling playback through production experiments.

3.3 Frame playback control

When creating a piece of art using datamoshing, it is important to be able to predict and control the datamoshing effects to accurately reflect the artist's intentions. To this end, we have identified two important elements; one is the movement of the original video, and the other is the frame playback control. Movements in the original video greatly influence datamoshing effects. To demonstrate this, we created four slideshow videos with different motion. These videos were created using two photos captured by the authors, a photo of a tree and a photo of a sightseeing pier. In the first video, we see a one-second zoom of the photo of the tree followed by ten seconds of panning the photo of the pier in the directions left, right, up, and down. These serve as source videos. Each of the subsequent videos is generated by disabling I-frames as described in Section 3.2; then, the videos are played back with an ordinary video playback software. Moreover, the videos are short (approximately 11 seconds), and they have different effects, depending on the panning direction in the source videos. The images in Figure 3-5 (1) were exported from the video panning to the left at the starting time, three seconds later, seven seconds later, and eleven seconds later. Figure 3-5 (2) shows the video created from Figure 3-5 (1) through datamoshing. In the second image of Figure 3-5 (2), we can see that the trunk of the tree has displaced to the right, and that the pixels on the left side have changed colors. In addition, Figures 3-5 (3), (4), and (5) are the result of datamoshing the videos that pan to the right, top, and bottom, respectively. These show changes in the colors of the pixels according to the panning direction. From this example, we can see that one method for controlling the datamoshing effects is by selecting and using different motion patterns in videos. The particular relationship between the original video movements and the datamoshing effects can be represented as shown in Figures 3-6 and 3-7. For illustration purposes, we have outlined the P-frames produced by the playback process with dotted lines. The P-frames are not displayed, but they are predictions of subsequent frames. The predictive result, which we call R-frame, is visually apparent and we illustrate it in the diagram with a solid line. Figures 3-6 (1), (2), and (4) are the frames to be displayed on the screen. We call the frames shown in Figures 3-6 (1), (2), and (4) the previous R-frame, the first subsequent R-frame, and the second subsequent R-frame, respectively. The P-frame (Figure 3-6 (3)) contains the differential information between the adjacent R-frames, but is not displayed on the screen. The first subsequent R-frame image serves as the I-frame. The second subsequent R-frame (Figure 3-6 (4)) is composed on the basis of the information of the P-frame (Figure 3-6 (3)); A, B, and C in Figure 3-6 denote the values of the pixel colors, and the arrows shown in Figure 3-6 (3) signify motion vectors. The information of the P-frame (Figure 3-6 (3)) indicates that the colors C and D are assigned to the pixels as signaled by the arrows, whereas the color of the previous frame is assigned if a color value is not specified. When playing back the video for this example, we can perceive the motion from left to right because the previous frame color shifts to the right. Figure 3-7 shows an example of datamoshing effects. In the case where the I-frame has been disabled, the first subsequent R-frame referred from the P-frame is the same as the previous R-frame. Therefore, the second subsequent R-frame differs from the result of the normal frame sequence.

Next, we describe the relationship between the frame playback control and the datamoshing effects. *GoldMosh* has four types of frame playback control. These frame playback control techniques were not common until the release of this software; thus, the type of effects that occur from datamoshing with frame playback control techniques was uncertain.



(1) Original video still of panned second picture to the left

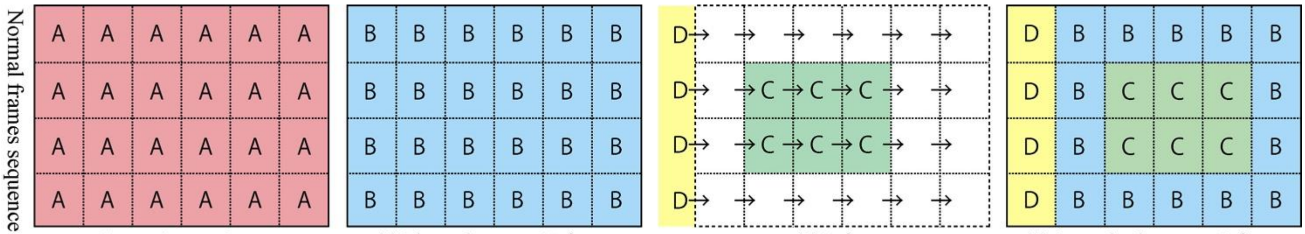
(2) Datamoshed video still of panned second picture to the left

(3) Datamoshed video still of panned second picture to the right

(4) Datamoshed video still of panned second picture to the top

(5) Datamoshed video still of panned second picture to the bottom

Figure 3-5. Comparison between original video and datamoshed videos



(1) Previous R-frame

(2) First subsequent R-frame = I-frame

(3) P-frame

(4) Second subsequent R-frame

Figure 3-6. Normal frames sequence process

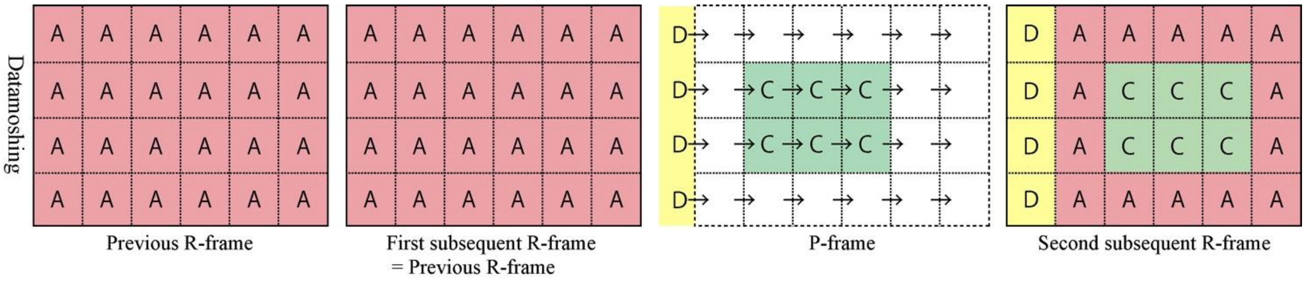


Figure 3-7. Datamoshing process

Next, we describe four frame playback controls and their related effects.

(a) Consecutive frame repeat-playback

When the specification of a P-frame (first P-frame) and its next consecutive P-frame (second P-frame) is repeated, we have a phenomenon similar to video feedback that is called "bloom" in *GoldMosh*. Figure 3-8 shows a simple mechanism of bloom occurrence. The initial frame (Figure 3-8 (1)) is the R-frame of a specified timing. The first subsequent R-frame (Figure 3-8 (3)) is displayed using predictions based on the first P-frame (Figure 3-8 (2)). The second subsequent R-frame (Figure 3-8 (5)) is displayed using predictions based on the second P-frame (Figure 3-8 (4)) and the first subsequent R-frame (Figure 3-8 (3)). Thereafter, repeating the same predictions based on the P-frames (Figures 3-8 (2) and (4)), B and C are stretched to the right as shown in Figure 3-8. This playback control generates an effect similar to a stretching picture, as shown in Figure 3-9 (bottom), on the basis of the differential information between the two frames in the upper row of Figure 3-9.

(b) Jump playback

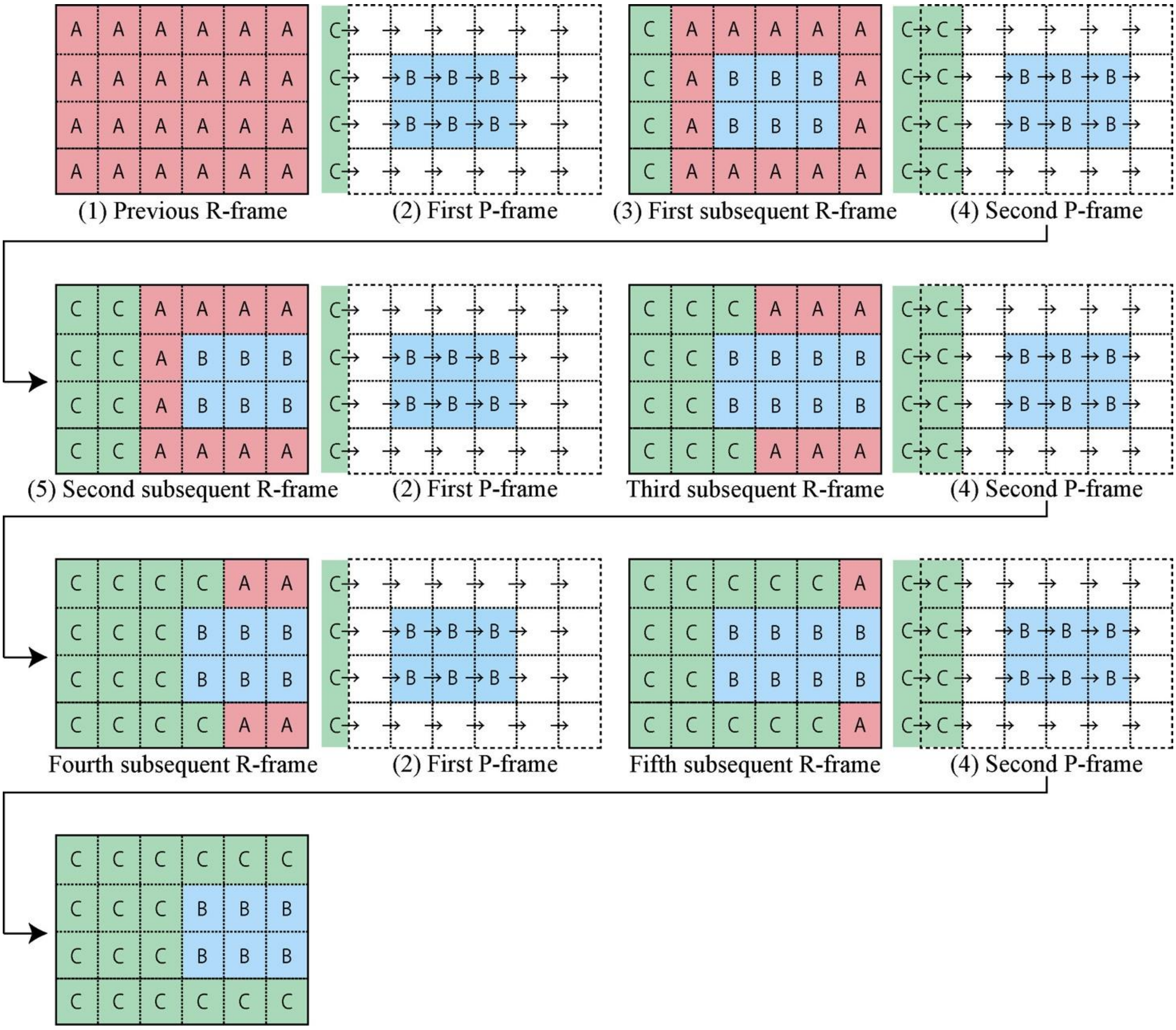
This control "jumps" to a selected P-frame during playback and continues to play back the video from this location. Figure 3-10 shows a simple mechanism for this type of frame playback control technique. If there is a certain continuous motion vector direction for the P-frame as shown in Figures 3-10 (1), (2), and (3), the fourth subsequent R-frame (Figure 3-10 (5)) can be displayed differently if a noncontinuous P-frame such as Figure 3-10 (4) is selected randomly and inserted. Differences vary depending on the original videos applied, but Figure 3-11 shows one of the frame control techniques of datamoshing with large changes in the inter-frame differences, such as those between the third and fourth columns.

(c) Shift consecutive frame repeat-playback

Bloom is a technique for repeating the same two P-frames; however, this frame playback control technique is performed by gradually shifting repeating frames at regular intervals and playback. This control moves to the third and fourth P-frame and plays back these P-frames after repeating the first and second P-frames’ playback. This is called "slow bloom" in *GoldMosh*. Figure 3-12 shows the slow bloom mechanism. First, this control generates the first bloom by repeating the first P-frame (Figure 3-12 (1)) and the second P-frame (Figure 3-12 (2)), and then obtaining the first bloom R-frame (Figure 3-12 (3)). (The process outlined by Figures 3-12 (1) to (3) is shown in more detail in Figure 3-8.) Next, bloom changes from the first bloom R-frame (Figure 3-12 (3)) to the second bloom R-frame (Figure 3-12 (6)), because the bloom process is repeated with the third and fourth P-frames (Figures 3-12 (4) and (5)). By repeating the bloom process, a gradual change effect is obtained through Figures 3-12 (3), (6), and (7). Although the results from the control depend on the original videos, we can see the clear, definite effects of slow bloom by shifting the P-frames every few seconds. Furthermore, this is confirmed through production experiments. Figure 3-13 shows the difference between the original video and the slow bloom video. The bottom row of Figure 3-13 shows the process of change in each loop.

(d) Loop playback

This control is a loop playback between two arbitrarily specified frames. When a loop jumps from the end point to the start point, the frame continuity is broken and the datamoshing effect occurs. This control is similar to the jump playback frame playback control, and it sometimes generates the bloom effect through a specific loop interval.



(1) Previous R-frame

(2) First P-frame

(3) First subsequent R-frame

(4) Second P-frame

(5) Second subsequent R-frame

(2) First P-frame

Third subsequent R-frame

(4) Second P-frame

Fourth subsequent R-frame

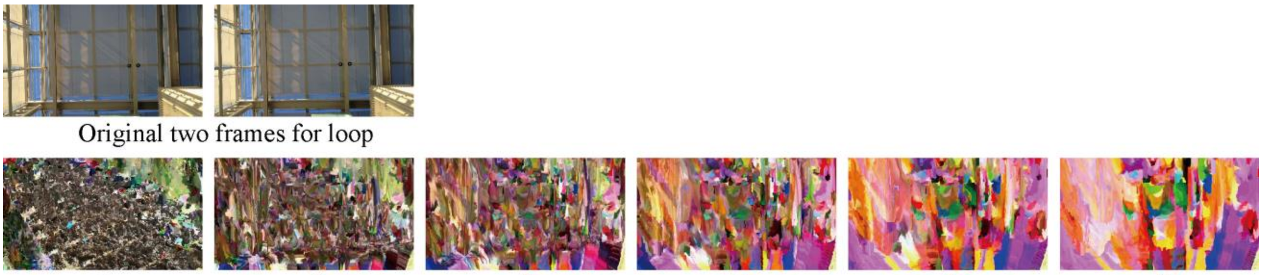
(2) First P-frame

Fifth subsequent R-frame

(4) Second P-frame

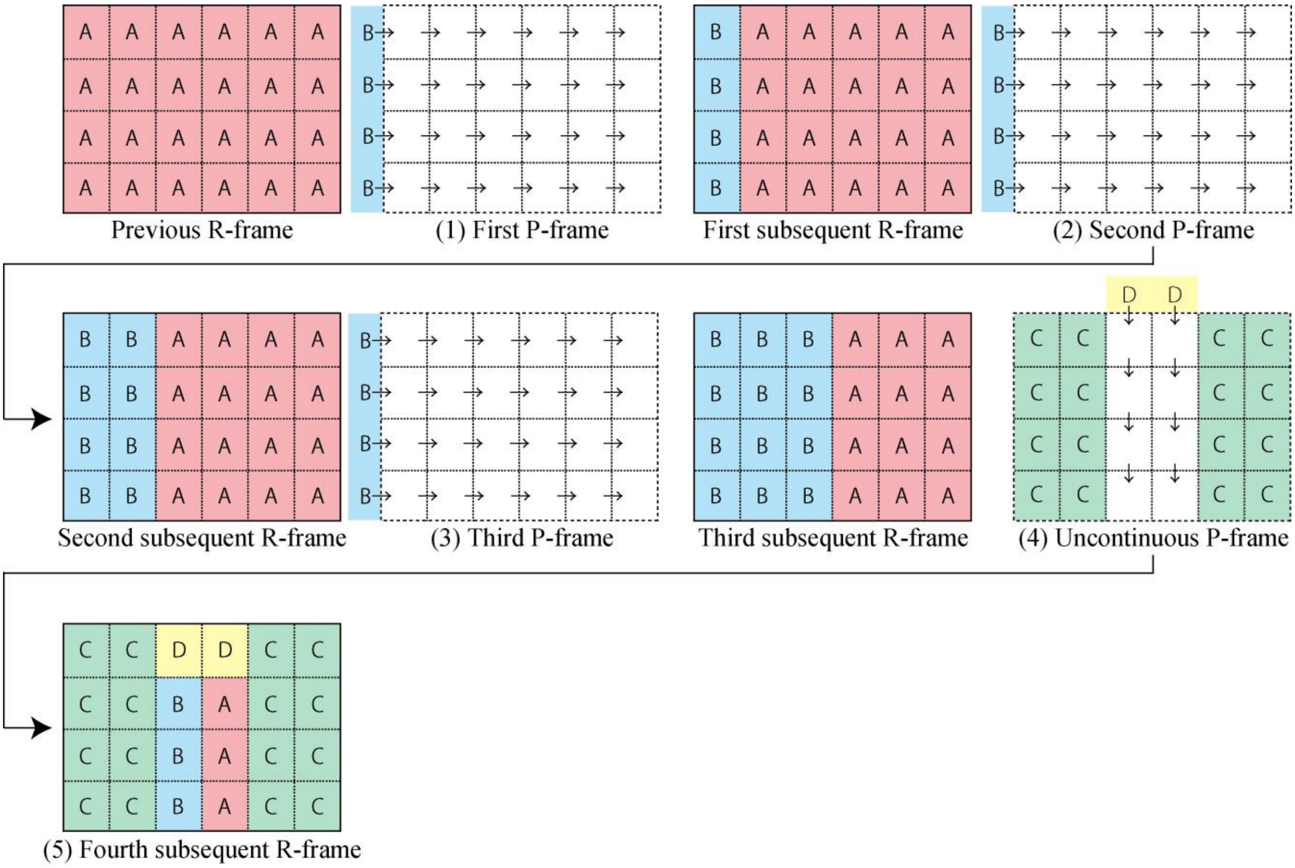
Sixth subsequent R-frame

Figure 3-8. Bloom effect mechanism


Original two frames for loop

1 2 3 4 5 6

Figure 3-9. Original frames (top) and the process of change (bottom)



Previous R-frame

(1) First P-frame

First subsequent R-frame

(2) Second P-frame

Second subsequent R-frame

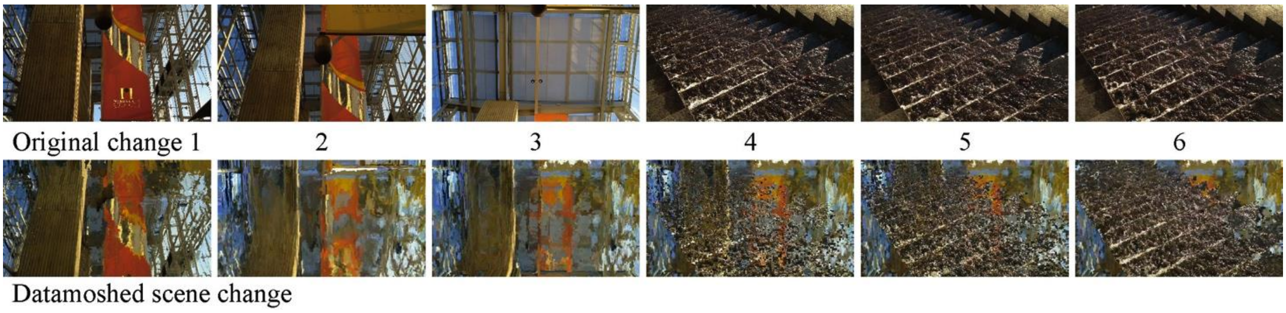
(3) Third P-frame

Third subsequent R-frame

(4) Uncontinuous P-frame

(5) Fourth subsequent R-frame

Figure 3-10. Prediction process in jump playback



Original change 1

2

3

4

5

6

Datamoshed scene change

Figure 3-11. Original video and datamoshed video (jump playback)

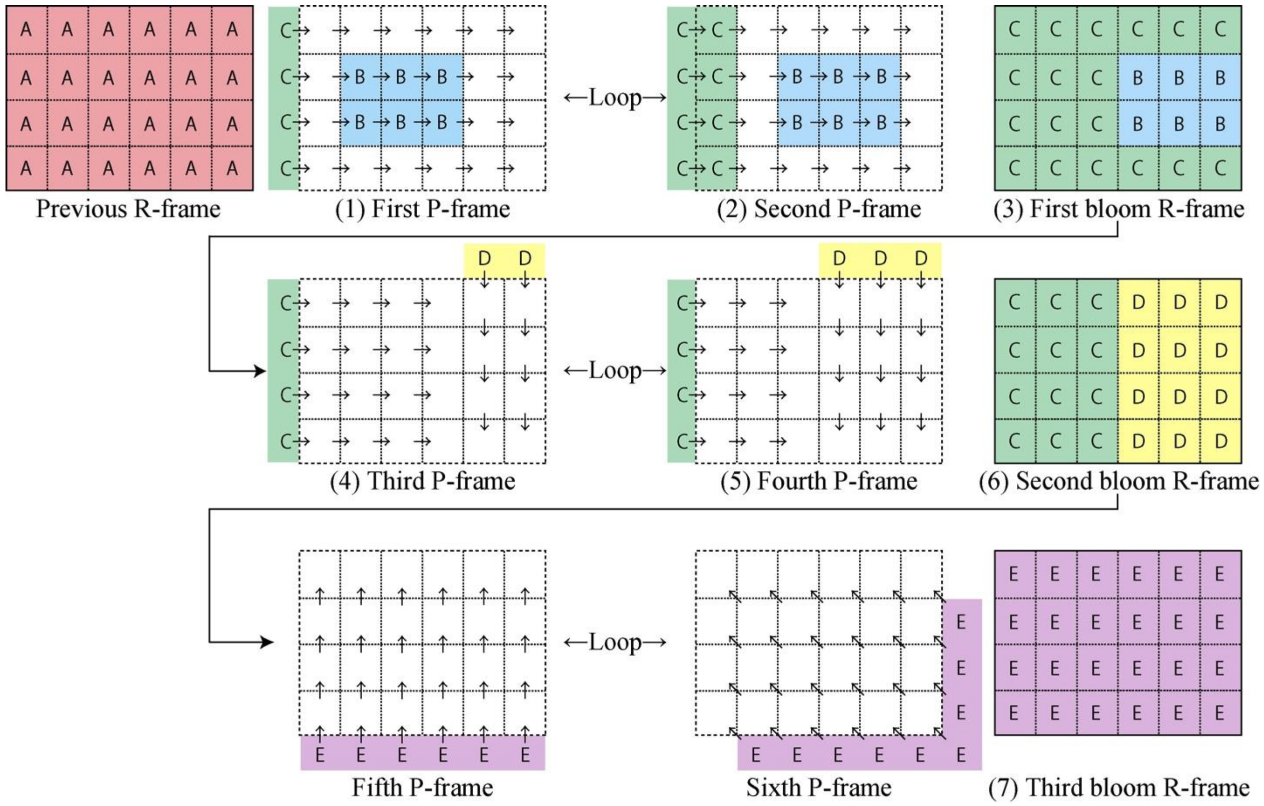


Figure 3-12. Slow bloom mechanism

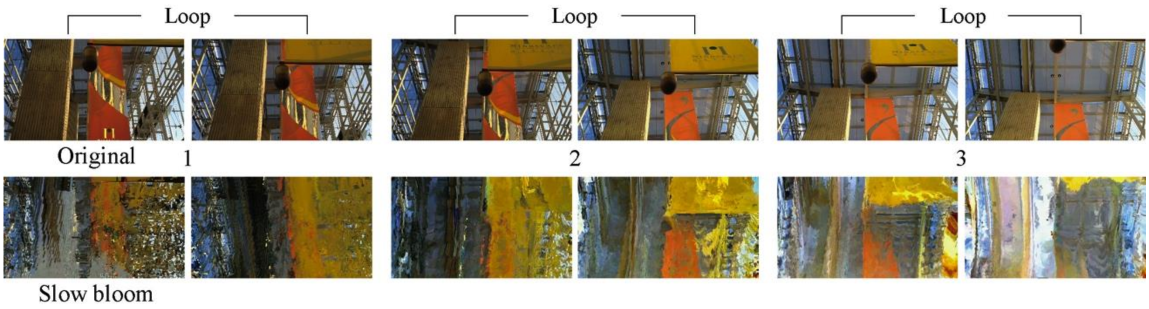


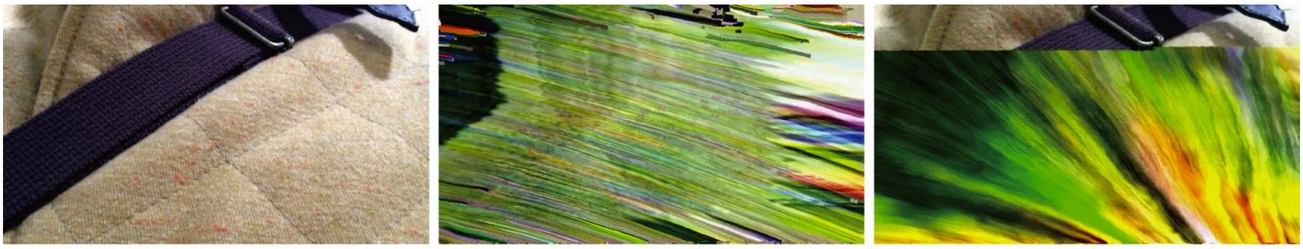
Figure 3-13. Original video (top) and slow bloom video (bottom)

4. Development of software for datamoshing

We developed software to apply datamoshing to videos. Section 4.1 describes the functions of our software and how to execute and control datamoshing using the proposed software. Section 4.2 describes a technique for real-time datamoshing. Section 4.3 compares the proposed software with *GoldMosh*, which is described in Section 2.

4.1 Datamoshing control with proposed software

When processing MP4 files, it is often the case that the datamoshing effect is not reflected in the entirety of the frame, as shown in Figure 4-1 (3). However, when applying datamoshing to an AVI file, the bloom effect is apparent over the entire video, as shown in Figure 4-1 (2). Although the encoding format MPEG4 Part2 is the same for AVI and MP4, because the file formats are different, the datamoshing effects are likewise different. Practical techniques for disabling the I-frame are different based on the file formats. For AVI files, the binary value "00 00 01 b0" or "01 b0" should be deleted to disable the I-frame. On the other hand, for MP4 files, the binary values "01 b6" should be replaced with "00 00" to achieve the same effect. Incorrectly deleting the binary values in MP4 files results in file corruption; hence, attention must be observed to the technique applied. Thus, even with the same encoding formats, when file formats differ, the datamoshing results are not the same. It must be noted that MP4 processing typically does not affect the entire frame; rather, only a portion of the frame is affected.



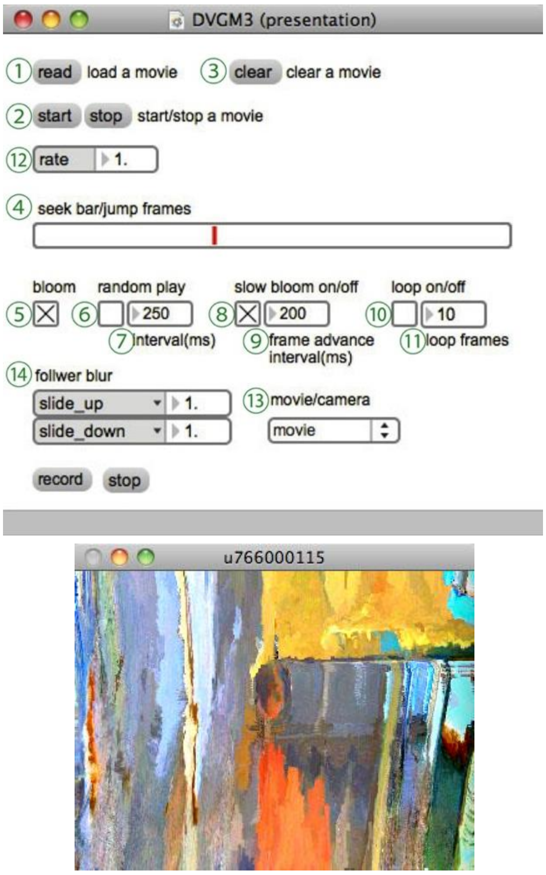
(1) Original (2) AVI (3) MP4

Figure 4-1. Difference between datamoshing effects

Such problem does not exist in AVI processing; thus, this type of processing seems preferable in most cases. The proposed software is developed in the Cycling '74 Max6 programming environment, which allows for easy video processing. Furthermore, Max6 is primarily a frame playback control that can play video with disabled I-frames.

The frame playback controls described in Section 3.3 (a) to (d) are implemented in the proposed software. Figure 4-2 illustrates the software interface. Clicking the "read" button (1) allows the user to select a video file with disabled I-frames and import the video to the software. The "start" and "stop" buttons (2) start and stop video playback, respectively. The "clear" button (3) erases any stored video file from software memory. The "seek bar/jump frames" trackbar (4) jumps to a frame position by dragging the red slider or by clicking the trackbar directly, and plays back a video from the selected position. The "bloom" check box (5) enables the effect described in Section 3.3 (a) by selecting the check box. Users can observe a bloom generated at any frame using the interface elements (4) and (5). The interface element "random play" (6) allows the control described in Section 3.3 (b) by changing the playback position randomly, and produces a datamoshing effect that is difficult to predict. The interface element "interval (ms)" (7) allows users to decrease or increase the interval of the random play for better artistic control. The interface element "slow bloom on/off" (8) allows the control described in Section 3.3 (c), and the interface element "frame advance interval (ms)" (9) specifies the timing required to proceed to the next frame. Users can change the speed of the slow bloom using these functions, thus observing variations in the slow bloom effect. The interface element "loop on/off" (10) enables and disables the loop playback described in Section 3.3 (d), respectively. The interface element "loop frames" (11) designates the number of frames to loop (loop length). It is possible to how the scene repeats by adjusting the loop length.

The functions listed in the previous paragraph enable the frame playback controls described in Section 3.3. Moreover, the proposed software has two other functions, playback speed control and reverse playback control. The proposed software can control the playback speed according to the value set up in "rate" (12). The rate values 1, 2, and 0.5 mean normal, double, and half speed, respectively; negative values indicate reverse playback; for example, -2 means double speed and reverse playback. The datamoshing effect changes according to changes in playback speed. When using reverse playback, the order of the P-frames that are played back is disturbed and correct prediction becomes impossible. Thus, the original content of the video is entirely destroyed; however, reverse playback provides a variety of interesting representations, especially intense noises.



DVGM3 (presentation)

(1) read load a movie (3) clear clear a movie

(2) start stop start/stop a movie

(12) rate 1.

(4) seek bar/jump frames

bloom (5) ☒

random play (6) □ 250

(7) interval(ms)

slow bloom on/off (8) ☒ 200

(9) frame advance interval(ms)

loop on/off (10) □ 10

(11) loop frames

(14) follwer blur

slide_up 1.

slide_down 1.

(13) movie/camera movie

record stop

u766000115

Figure 4-2. Proposed software interface (top) and monitor window (bottom)

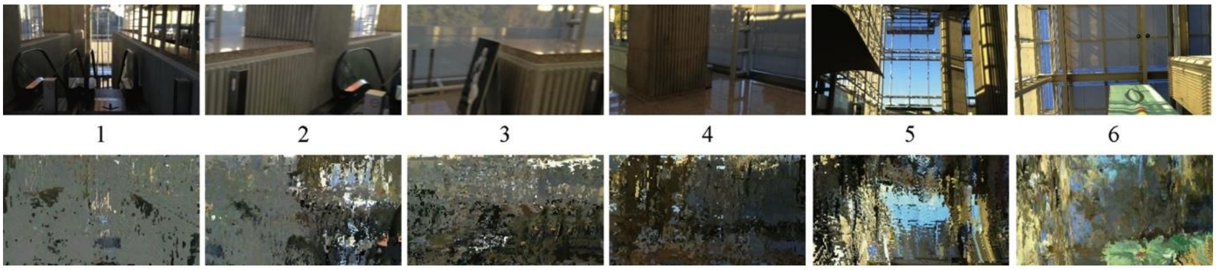


Figure 4-3. Original video (top) and datamoshing video (bottom)

In addition, processing AVI files provides more complicated results when compared to MP4. Figure 4-3 shows two videos. The images in the top row correspond to the video that is played back normally; the images in the bottom row correspond to the video that is datamoshed with reverse playback control. (The frame images in the bottom row have been arranged to be compared easily with the frames images in the top row.)

4.2 Real-time datamoshing

The proposed software can be applied for datamoshing pre-recorded videos, as well as for datamoshing real-time camera signals. We call such application real-time datamoshing (RTD). RTD applies datamoshing to video obtained at random using a camera in real-time.

The process of the RTD first records 20 frames and saves them in an MP4 file format. In this file, the first frame is saved as an I-frame, and the remaining frames as P-frames. The file is called immediately by the software and datamoshing is applied. Using the proposed software, we generate the bloom effect described in Section 3.3 (a) for frame control playback, and we specify the first P-frame and the second P-frame at random time intervals. However, we do not specify a first frame, thus avoiding a declaration of the I-frame; therefore, playback is a state of constant P-frame. As a result, even though the I-frame is present, playback occurs in a manner similar to a disabled I-frame. This operation repeats at intervals of approximately 15 to 30 seconds. In this way, the original video can be processed in a short time and real-time datamoshing can be realized. Figure 4-4 shows an operation flow for this method.

A user can utilize this function by selecting the camera option from the "movie/camera" menu (Figure 4-2 (13)) in the interface of the proposed software. Figure 4-5 shows an example of such execution. The images Face 1 to 6 show the effect of changing by facial expressions. The images Walk 1 to 3 and Clothing 1 to 3 show the effects of walking and of moving clothes. The images Pause 1 to 6 in the bottom row show a static human being recording as affected by the noise of a web camera.

4.3 Comparison with GoldMosh and future consideration of the adaptation of hardware decoding

This section compares the proposed software with *GoldMosh* (described in Section 2). Table 4-1 lists the differences in capability between *GoldMosh* and the proposed software. *GoldMosh* recognizes MP4 video file formats only. On the other hand, the proposed software is able to apply datamoshing to AVI video file formats in addition to MP4 file formats. Moreover, the proposed software supports speed control, reverse frame playback control, RTD, and full high-definition (HD) (1920 x 1080 px) resolution recording. These features help users create diverse artistic representations.

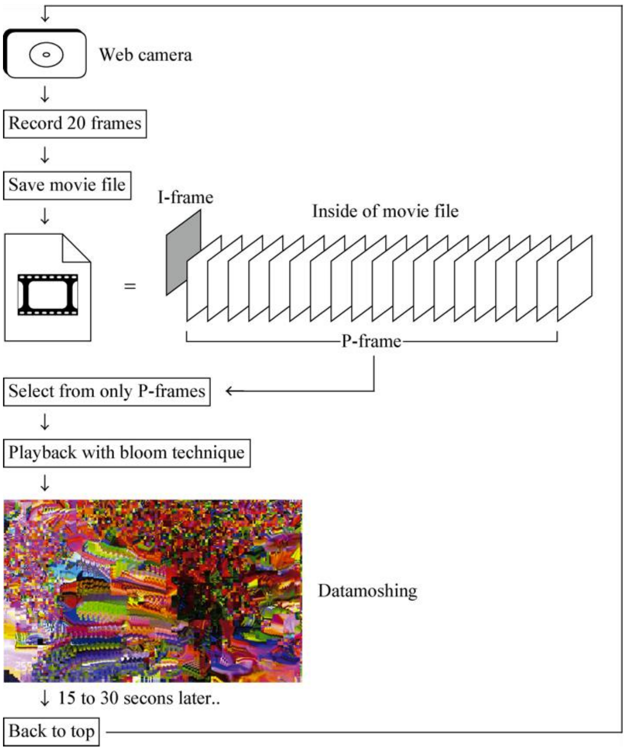


Figure 4-4. RTD operation flow

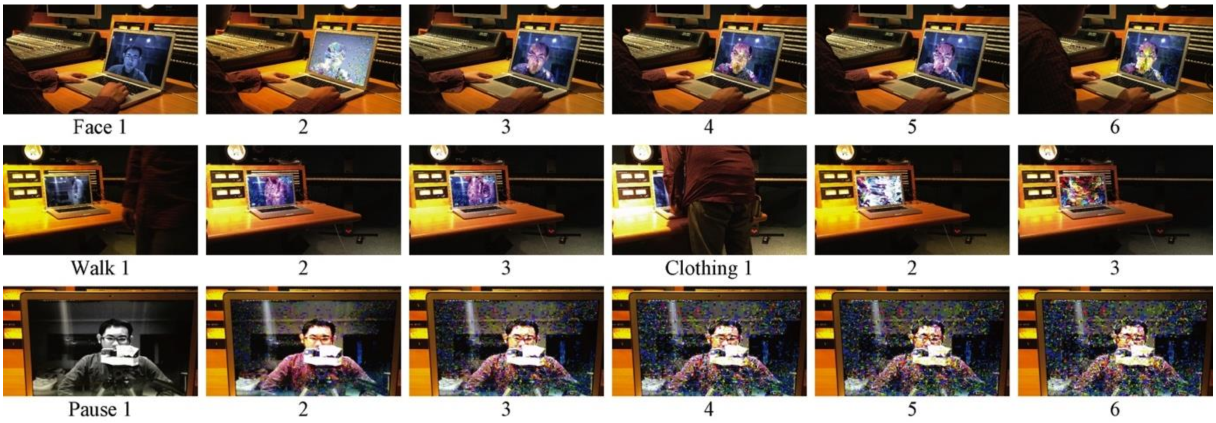


Figure 4-5. Real-time datamoshing videos playback process

Table 4-1. Difference in functionality between proposed software and *GoldMosh*

	Playback						File format		Real-time	Recording
	Bloom	Slow bloom	Random	Loop	Reverse	Speed control	MP4	AVI	datamoshing	resolution
GoldMosh	●	●	●	●	–	–	●	–	–	800 x 600
Proposed software	●	●	●	●	●	●	●	●	●	1920 x 1080

This research deals with datamoshing using software decoding techniques. Further consideration should be given in the future to the differences between these and hardware decoding, which can reduce the latency of datamoshing and adds responsiveness, adaptivity and intuitiveness to the datamoshing.

5. Art of datamoshing

In this section, we discuss creating works using the proposed software described in Section 4. Utilizing datamoshing techniques enables us to engage some of the normally unnoticed artifacts of media communication in modern society. A video is played normally if it does not have disabled I-frames. However, with arbitrarily controlled datamoshing effects, the error essence changes to become intentional, rather than a mistake or a superfluous effect. Given that video sites usually recover quickly from any glitches that result from errors in signal transmission or decoding, viewers usually do not feel uneasy. Rapid recovery is possible thanks to the native function of the I-frame. However, there is an opportunity for thinking about errors as right or wrong in the nature of society by deliberately producing work that is driven by error. Such is the purpose of this creation.

Datamoshing works often have an interesting general appearance and surface appeal and have been utilized by artists usually without any particular thought of deeper aesthetic or social implications. However, to address these issues in artworks, it is important to express the intention of the creator by carefully selecting the original source videos and carefully choosing the datamoshing techniques that should be applied.

To produce the work discussed in this paper, we primarily used the effect of slow bloom and set the playback speed control to slow values to create a gradual slow bloom effect. In addition, we processed AVI files to process the entire image frame. At first glance, the results appear similar to animation using abstract painting techniques. However, in the course of examining the created work, viewers can change their perception by finding that the intention of this creation is the technique of using error from datamoshing, thereby creating an opportunity for changing the essence of error (and of "wrongness"). For this work to experience such change, it should avoid understanding error at a glance. In addition, this work can be used to suggest a notion about society through pictures that transition from images of the shadow of a tree to images of industrial machinery through twists and turns.

By creating art from the environment of video compression technology, artists can find results outside the realm of standard video synthesis that are difficult or impossible to produce using popular or commercial video editing systems.

5.1 Producing a piece using a photo slideshow video

To create the work called *Don’t flatten. Part 2.1*, we prepared 30 photos captured by us and made a photo slideshow video file that transitions every second, adding motion to the pictures by determining start and end points to create zoom in and zoom out effects toward an arbitrarily determined center point. We prepared this original video as an AVI file, encoded it in MPEG4 Part2 format, and disabled the I-frames using the procedure described in Section 3.2. This created an expression of gradually mixing images from the video file by reading and datamoshing with the proposed software described in Section 4. Figure 5-1 shows the playback process of changing the original video (top five rows) and the datamoshing video (bottom five rows); the order is from upper left to lower right, aligned in an almost identical frame timing.

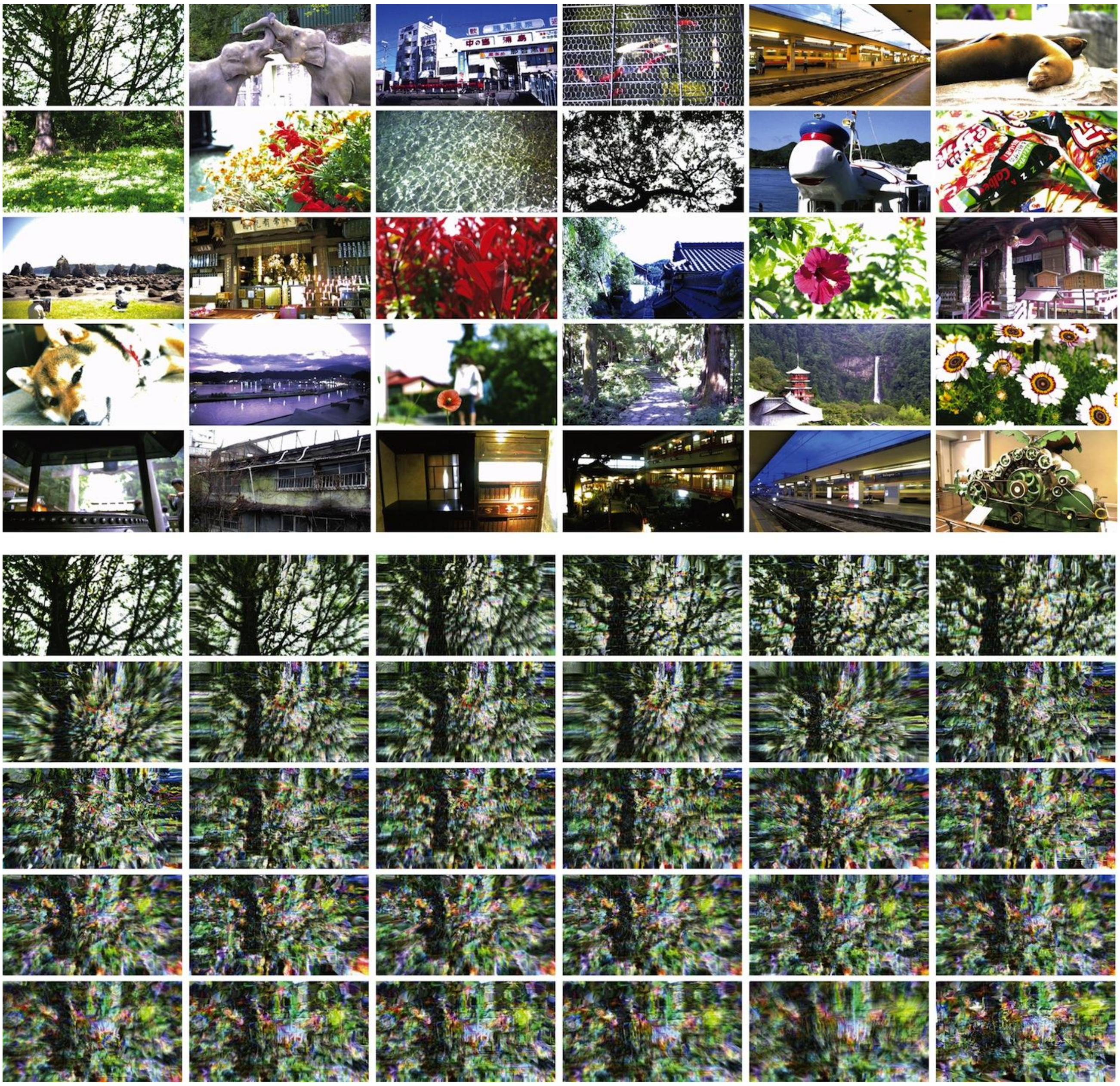


Figure 5-1. Playback processes of original video (upper five lines) and datamoshing video (lower five lines)

5.2 Appreciation

Don't flatten. Part 2.1 was created in September 2012 and was selected for presentation in the *2012 Asia Digital Art Award Exhibition FUKUOKA*[3] in the video art category. In an announcement from the organizer, more than 700 works from 16 countries and regions were submitted, and 101 works were selected from all categories. This work was chosen in the video category A, along with 15 other works. The video category A is intended for professional designers with proficiency in technical and artistic abilities. The judges were prominent writers and scholars; moreover, this competition used as one criterion "high artistic sensibility is based on logical thinking." Therefore, we consider that this work has been appreciated with constant evaluation as video art work. In addition, we received the following comment from another artist who presented at the same venue, "Very interesting! I think this technology has the possibility of various developments." Moreover, we received many same questions from another artists and viewers that is "how to make the author's piece?" We feel from these comments that to the extent that the audience had prior experience viewing datamoshing works, they were intrigued by the unique visual qualities in this piece, which are appreciably different.

Datamoshing (in particular real-time datamoshing) expresses another world with errors. This technique has the possibility of representing a different world, similar to augmented reality. The error used for the authors' representation changes the original nature of an unnecessary error such that the error becomes necessary, rather than a useless or a mistake. This is unique representation of media art using information technologies such as computers, and also is a representation technique that can be implemented in current information technologies. Our goal is to propose to artists a new representation of error, and to viewers a different way of thinking of error, by providing unique techniques in media art that intentionally misuse video compression technology.

6. Conclusion

In this paper, we described new expressions of work based on a general technique of datamoshing. As a video production technique, datamoshing requires the use of computers and generates unique expressions that cannot be performed by hand. Thus, the study of further expressions and techniques of datamoshing are effective enhancements of expression for computer-based video production. Considering artistic expression with the use of datamoshing techniques makes it possible to raise a question of society through our work, and allows for the constant evaluation of video art. In the future, we would like to extend our research beyond the techniques of datamoshing to broaden the range of expression and hope to proceed with further research to perform unique creations and presentations.

Acknowledgments

We thank the judges of the *2012 Asia Digital Art Award Exhibition FUKUOKA* for their high evaluation of our work. In addition, we would like to thank Masahiro Ura (Institute for Advanced Studies in Artificial Intelligence, Chukyo University) whose comments and suggestions were immeasurably valuable throughout the course of our study.

References

[1] Ito, Y., Stone, C., Yamada, M. and Miyazaki, S., Datamoshing Technique for Video Production, The Society for Art and Science, Proc. of NICOGRAPH International 2013, CD-ROM pp. 60-71 (2013)

[2] Moradi, I., Scott, A., Gilmore, J. and Murphy, C., Glitch: Designing Imperfection, Mark Batty Publisher (2009)

[3] Visual Glitch, using Ruby | RubyKaigi 2011 (July 16 - 18), http://rubykaigi.org/2011/en/schedule/details/18S02, (accessed 2013-10-30)

[4] Nam June Paik Art Center, With Nam June Paik, pp. 300, Nam June Paik Art Center (2011)

[5] Hanhardt, J. G., Nam June Paik and the Transformation of Video into Art, Mostly Video Nam June Paik, Tokyo Metropolitan Art Museum (1984)

[6] OReilly, D., *Compression Reel*, https://vimeo.com/2564771, (accessed 2013-10-30)

[7] ucnv, *"Hip Hop no Keinen Henka" no Keinen Henka*, http://kuchiroro.com/special/cd/3/index.html, (accessed 2013-10-30)

[8] ITU, Audiovisual and multimedia systems, http://www.itu.int/rec/T-REC-H/en, (accessed 2014-2-25)

[9] Information Processing Society of Japan, ISO/IEC JTC 1/SC 29 Programme of Work, http://itscj.ipsj.or.jp/sc29/29w42911.htm, (accessed 2014-2-25)

[3] 2012 ASIA DIGITAL ART AWARD <http://adaa.jp/2012/> (accessed 2013-10-30)

Yuichi Ito received his B.S., M.S., Ph.D. degrees in the Media Engineering Department of the School of Engineering at Chukyo University, Japan. He is currently a research assistant in the same department. He has engaged in the research and production work of sound art as well as media art, and he is a member of the Society for Art and Science.

Carl Stone is a graduate of the California Institute of the Arts. He is currently a professor in the Media Engineering Department of the School of Engineering at Chukyo University, Japan. His research interests include audio informatics, digital signal processing, and alternative interfaces for musical performance.

Masashi Yamada received his B.S. and M.S. degrees as well as his Ph.D. in Engineering from Nagoya Institute of Technology in 1992, 1994, and 1999, respectively. He is currently a professor in the Information Engineering Department of the School of Engineering at Chukyo University, Japan. His research interests include artificial intelligence and smart interface.

Shinya Miyazaki received his B.E., M.E., and Ph.D. degrees in Information Engineering from Nagoya University. He is currently a professor in the Media Engineering Department of the School of Engineering at Chukyo University, Japan. His research interests include applications of computer graphics and virtual reality.