International Journal of Innovative Computing, Information and Control
Volume **9**, Number **4**, April **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 1477–1492

REAL-TIME PROCESSING TECHNIQUE FOR CONTENT-AWARE IMAGE RESIZING OF VIDEO

Daehyun Park, Kanghee Lee and Yoon Kim*

Department of Computer and Communications Engineering
Kangwon National University
192-1, Hyoja2dong, Chuncheon-si, Gangwon-do 200-701, Republic of Korea
{ dhpark509; kanghee }@kangwon.ac.kr; *Corresponding author: yooni@kangwon.ac.kr

Received February 2012; revised August 2012

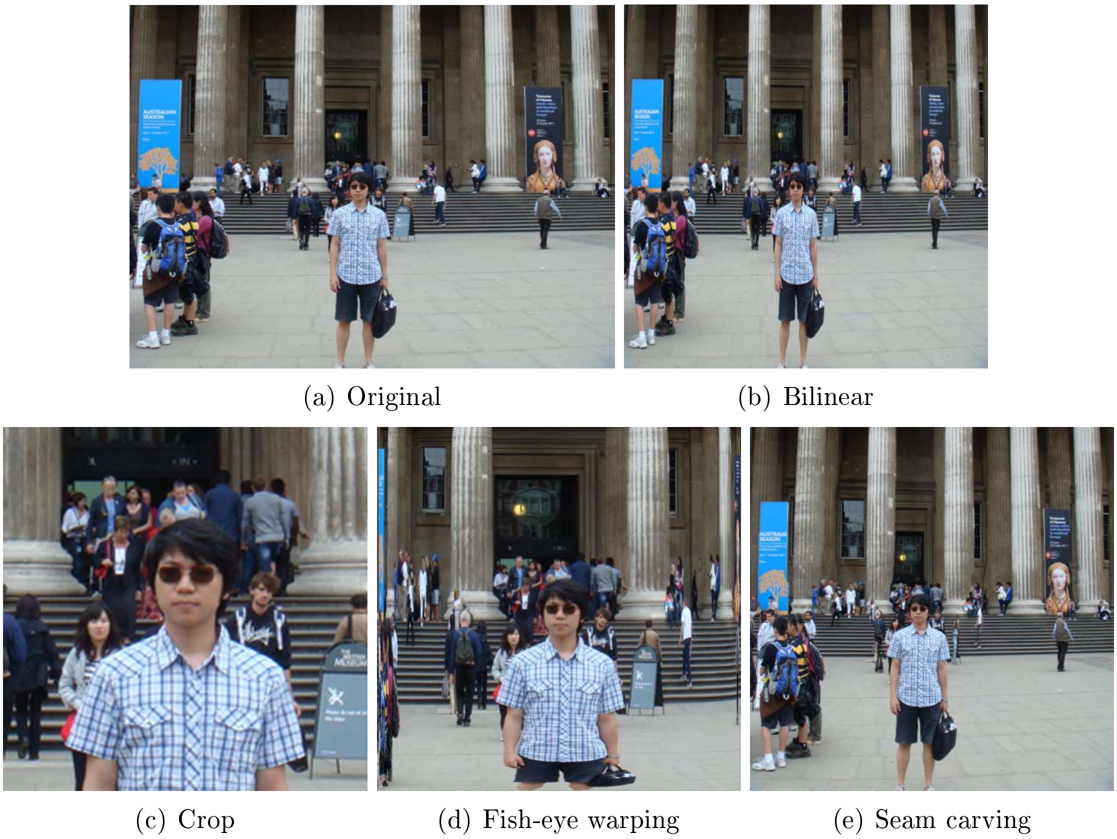
Abstract. *In this paper, a new real-time video scaling technique that preserves the dominant contents of video is proposed. Because a correlation exists between consecutive frames in video, determining the seam of the current frame with reference to the seam of the previous frame allows us to present an effective real-time video scaling technique without additional visual artifacts and analyzing the entire video. For this purpose, frames that have similar features in video are classified into a shot, and the first frame of a shot is resized using conventional seam carving on static images to preserve the important contents of the image as much as possible. At this time, the information about the seam extracted to convert the image size is saved, and the size of the next frame is controlled with reference to the seam information stored in the previous frame. And then, above process is repeated. The proposed algorithm has a fast processing speed similar to the bilinear method, while preserving the main content of an image to the greatest extent possible. In addition, because the memory usage is remarkably small compared with the existing seam carving method, the proposed algorithm is usable in mobile devices, which have many memory restrictions. Computer simulation results indicate that the proposed technique provides a better objective performance, subjective image quality, and content conservation than conventional algorithms.*

Keywords: Real-time processing, Content-aware image resizing, Seam carving

1. Introduction.

Because of the development of wireless communication and mobile electronic equipment, the availability of images has increased, and images have been already considered to be important information media. In order to use an image more effectively, image resizing is important, and thumbnail generation has become an indispensable technology, which produces small images to enable effective searching through a large quantity of images. However, because the simple existing resizing technique does not take into consideration the contents of an image but changes all of its parts equally, the contents become transformed and distorted. Therefore, in order to change the image size effectively, it is necessary to develop a new content-based image resizing method that preserves the contents of an image while changing its size.

In the conventional interpolation method, also known as the representative scaling method, if the scaling ratio is different from the aspect ratio of the source image, the contents of the source image are transformed (Figure 1(b)). Therefore, content-based geometric transformations have been studied in order to prevent this phenomenon. There is the simple crop technique [1], which seeks out and shows the main content within an image (Figure 1(c)). However, this technique has the disadvantage of deleting video information, with the exception of the preserved contents themselves. To compensate for this shortcoming, Liu and Gleicher [2] proposed the fish-eye warping technique, which enlarges the main content and diminishes the rest of the video information except for the preserved contents (Figure 1(d)). Fish-eye warping preserves the main content of



(a) Original (b) Bilinear

(c) Crop (d) Fish-eye warping (e) Seam carving

FIGURE 1. Comparison of various methods to resize images

an image as much as possible but has the disadvantage of severely distorting the rest of the video information. Recently, Avidan and Shamir [3] introduced the seam carving technique, which is acknowledged to have high image scaling performance. This method deletes or adds pixels that have low importance and influence on image distortion (Figure 1(e)). In addition, studies on additional methods to preserve the contents and change the size of an image are in progress [4-6,13-21].

Video carving [9,10], which is the application of seam carving to a video, uses a 3D cube to connect the frames to the time axis. In a video, the contents of an image change locationally and geometrically with time. Therefore, in order to attain effective image scaling, the entire 3D cube has to be detected while considering the energy of one spatial axis and the time axis. Because the dynamic programming technique [22,23] that is used for the seam carving of static images requires a large amount of memory and many operations, the graph cut [24,25] technique is instead used in video carving.

The 3D cube is a set of frames connected to the time axis, and the red surface inside the cube is the best seam set that is obtained by searching the whole cube (Figure 2). The image size in a video can be changed by one pixel by adding or removing one seam set. Therefore, for the conversion to various sizes, the 3D cube must be searched separately for each of the seam sets required. Because the video carving has to generate a 3D cube, it uses a large amount of memory. In addition, because many operations are needed in order to search and analyze the 3D cube, it is impossible to achieve real-time processing on a system with limited resources, such as a mobile terminal.

In this paper, a content-based technique for geometric image transformation on a real-time basis is proposed. Because a correlation exists between consecutive frames in a video, the seams of two consecutive frames are analogous. By determining the seam

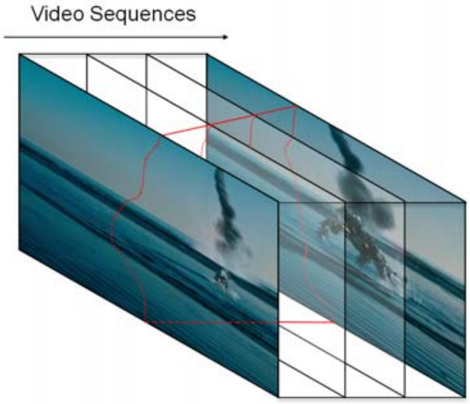


FIGURE 2. Video carving

of the current frame from this correlation, an effective real-time video scaling technique without the shaking phenomenon and analyzing the entire video is proposed.

The proposed technique operates by shot unit, which means that the consecutive images are taken by a single camera, and all of the frames within a shot have similar features. First, in order to separate each shot effectively in a video, a shot change is detected by monitoring the brightness differences and the histogram differences, which are susceptible to movement and color change [11,12], respectively. If a shot change is generated and a new shot begins, the first frame of the shot is resized using the conventional seam carving technique on the static image. At this time, the seam extracted by the seam carving technique and the energy information according to the seam coordinates are stored. The proposed image resizing technique can calculate the new seam of the next frame in real-time using the stored information instead of creating a 3D cube which requires information on all of the video frames.

This paper is organized as follows. The proposed algorithm is presented in Chapter 2. Chapter 3 presents and discusses the experimental results. Finally, our conclusions are given in Chapter 4.

2. **Proposed Image Resizing Algorithm in Video.** As shown in Figure 3, the proposed real-time content-based geometric image transformation system is composed of three parts: shot change detection (SCD), generating seam using the information of the previous frame, and image resizing. The input image is obtained by the frame and its RGB values are recorded. The proposed algorithm determines whether a new shot is initiated by detecting a shot change. In a case where a new shot is initiated, the stored seam information of the previous frame is ignored, and the seams are searched using the seam carving technique on the static images. At this time, after saving the information about the searched seam, the image is converted to the target size. On the other hand, in a case where a shot change is not generated, the seams of the current frame are calculated by using the stored seam information of the previous frame, and then the image resizing is carried out.

As shown in Figure 3, the proposed real-time content-based geometric image transformation system is composed of three parts: shot change detection (SCD), generating seam using the information of the previous frame, and image resizing. The input image is obtained by the frame and its RGB values are recorded. The proposed algorithm determines whether a new shot is initiated by detecting a shot change. In a case where a new shot is initiated, the stored seam information of the previous frame is ignored, and the seams are searched using the seam carving technique on the static images. At this

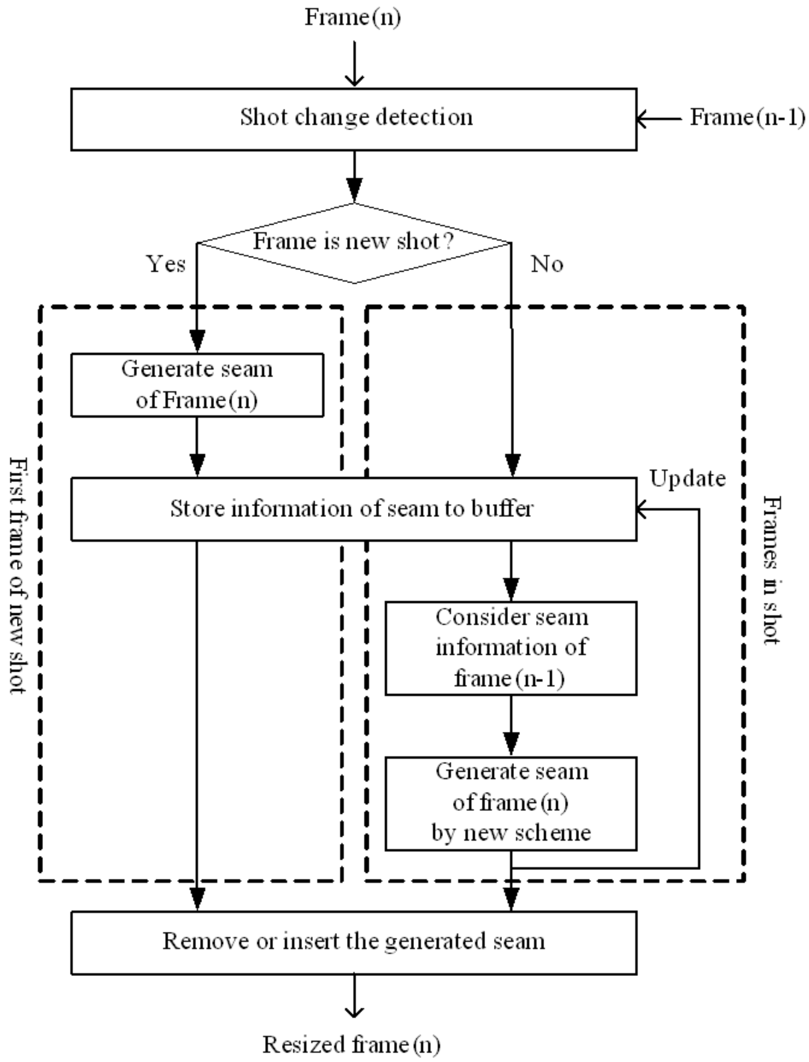

Frame(n)
Shot change detection
Frame(n-1)
Frame is new shot?
Yes
No
Generate seam of Frame(n)
Store information of seam to buffer
Update
First frame of new shot
Frames in shot
Consider seam information of frame(n-1)
Generate seam of frame(n) by new scheme
Remove or insert the generated seam
Resized frame(n)

FIGURE 3. Overall system block diagram

time, after saving the information about the searched seam, the image is converted to the target size. On the other hand, in a case where a shot change is not generated, the seams of the current frame are calculated by using the stored seam information of the previous frame, and then the image resizing is carried out.

2.1. **Detecting shot change.** Because a video viewed on a PC and/or in a mobile environment has generally more than 10 fps, the shot change detection is performed every 10 frames. First, the feature values are extracted between two consecutive frames.

$$\begin{aligned} f_i(n) &= \sum_{j=0}^{height-1} \sum_{i=0}^{width-1} |i_n(i,j) - i_{n-1}(i,j)|, \\ f_h(n) &= \sum_{k=0}^{255} |h_n(k) - h_{n-1}(k)|, \end{aligned} \tag{1}$$

where $i_n(i,j)$ is the (i,j)th pixel value in the nth frame, and f_i represents the brightness change susceptible to movement. In addition, $h_n(k)$ indicates the number of the pixels whose value is k in the nth frame, and the difference between $h(k)$s of neighboring frames is defined as f_h of the histogram change susceptible to color change.

For the stability of the algorithm, the shot change detection is not performed until 10 feature values are gathered. After 10 feature values are gathered, the largest and the second largest feature values are extracted and the difference between the two values is calculated. The shot change between two consecutive frames is detected through the following equations.

$$\begin{aligned}
&SCD = \begin{cases} 1, & \text{if } m_i > 3s_i \text{ and } m_h > 3s_h \\ 0, & \text{otherwise} \end{cases}, \\
&m_i = \max_{\alpha \in \boldsymbol{F}_i}(\alpha), \quad s_i = \max_{\alpha \in \boldsymbol{F}_i, \alpha \neq m_i}(\alpha), \\
&m_h = \max_{\alpha \in \boldsymbol{F}_h}(\alpha), \quad s_h = \max_{\alpha \in \boldsymbol{F}_h, \alpha \neq m_h}(\alpha), \\
&\boldsymbol{F}_i = \{f_i(n-9), f_i(n-8), \cdots, f_i(n-1), f_i(n)\}, \\
&\boldsymbol{F}_h = \{f_h(n-9), f_h(n-8), \cdots, f_h(n-1), f_h(n)\},
\end{aligned} \tag{2}$$

where $\boldsymbol{F}_i$ and $\boldsymbol{F}_h$ are the sets of the feature values calculated on the previous 10 frames, and m_i and s_i are the largest and the second largest values within the set $\boldsymbol{F}_i$. Also, m_h and s_h are the largest and the second largest values within the set $\boldsymbol{F}_h$. In the case where m_i and m_h are three times greater than s_i and s_h, respectively, it is determined that the shot change has happened. If a shot change is detected, as mentioned above, the shot change detection process is not performed until 10 new feature values are gathered.

2.2. **Deriving seam in the first frame.** After a shot change is generated, the conventional seam carving technique for a static image is applied to the first frame. All the coordinate and energy values of the seams of the first frame are stored in order to use these information when finding seams in the next frame. The following equations show the stored information of the seam in $W \times H$ frame.

$$\begin{aligned}
&\boldsymbol{S} = [\boldsymbol{S}_1,\ \mathrm{S}_2,\ \boldsymbol{S}_3,\ \cdots], \quad \boldsymbol{S}_n = \{\boldsymbol{C},\ \boldsymbol{E},\ e_{ac}\}, \\
&\boldsymbol{C} = [x_0,\ x_1,\ \cdots,\ x_{H-1}] \text{ or } [y_0,\ y_1,\ \cdots,\ y_{W-1}], \\
&\boldsymbol{E} = [e_0,\ e_1,\ \cdots,\ e_{H-1}] \text{ or } [e_0,\ e_1,\ \cdots,\ e_{W-1}],
\end{aligned} \tag{3}$$

where the set $\boldsymbol{S}_n$ includes the information about one seam, and $\boldsymbol{S}$ is the array of $\boldsymbol{S}_n$ found in one frame. The number of seams is determined by the target image size. $\boldsymbol{S}_n$ is comprised of the array $\boldsymbol{C}$ of the seam's coordinates, the array $\boldsymbol{E}$ which shows the energy in each coordinate, and the value e_{ac} accumulating all the values in $\boldsymbol{E}$. At this time, $\boldsymbol{C}$ stores only x coordinates in case of the vertical seam or only y coordinates in case of the horizontal seam. W and H show the width and height of image, respectively. Figure 4 shows an example of storing seam information from the results of seam carving.

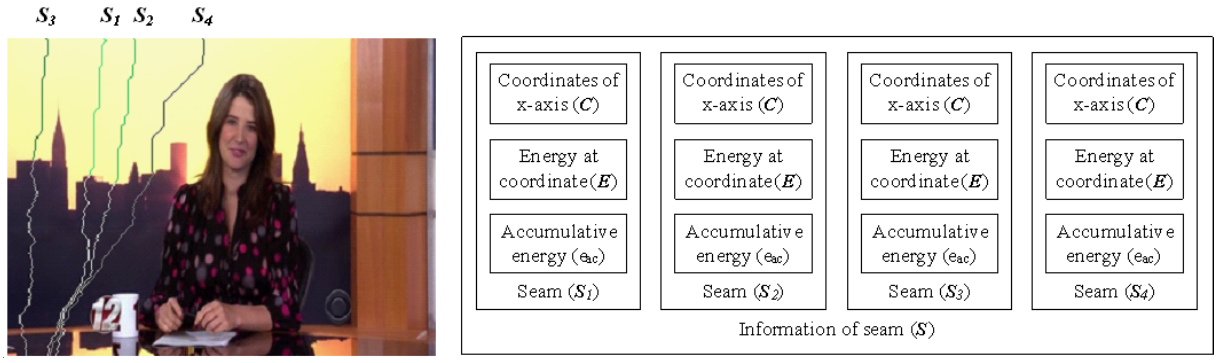
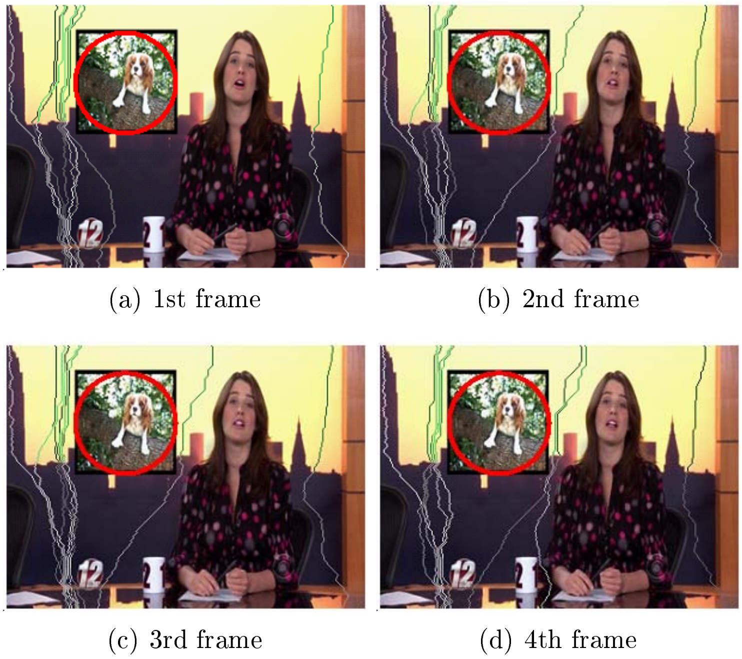

S3
S1
S2
S4
Coordinates of x-axis (C)
Energy at coordinate(E)
Accumulative energy (eac)
Seam (S1)
Coordinates of x-axis (C)
Energy at coordinate(E)
Accumulative energy (eac)
Seam (S2)
Coordinates of x-axis (C)
Energy at coordinate(E)
Accumulative energy (eac)
Seam (S3)
Coordinates of x-axis (C)
Energy at coordinate(E)
Accumulative energy (eac)
Seam (S4)
Information of seam (S)

Figure 4. Seam data storage example

Seams 1, 2, 3, and 4 are numbered in the ascending order of their energy values. The corresponding coordinate sets, energy values, and total cumulative energy values for each seam are stored systematically in the buffer.

2.3. **Generating seam of current frame by new scheme.** When no shot change occurs and the current frame belongs to the same shot as the previous frame, the seams of the current frame are extracted with reference to the stored information about the seams in the previous frame. Because there is a correlation (a similarity) between neighboring frames within an identical shot, the energy distribution of the neighboring frames are also correlated and similar, and the seams of a frame are analogous to those of neighboring frame. Therefore, the seams of the current frame are extracted in the specified range from the coordinates of the seams of the previous frame. That is, when the seams are extracted, the continuity of the time axis has to be considered. If the seams for each frame in video are extracted independently, the shaking phenomenon (jitter of the main content in video) occurs. This shaking phenomenon occurs, in particular, because of a difference in the numbers of the seams around the contents each frame. For example, assume that in the first frame, three seams and five seams were extracted from the left and right of some content, respectively, whereas in the second frame, five seams and three seams were extracted from the left and right of the same content, respectively. When the image size is changed, the two image sizes are identical because eight seams were equally extracted for each frame. However, the relative locations of the contents between the two frames have a difference of 2 pixels. If this process is repeated, a jitter of the contents will occur. Figure 5 shows the results of independently expanding the size of the consecutive frames by the seam carving.

If we give attention to the picture in the red circle each frame in Figure 5, we can observe that 7 seams and 1 seam exist to the left and right of the red circle in the first frame, respectively, whereas 6 seams and 2 seams exist to the left and right of the red circle in the second frame, respectively. In the original video, the picture in the red circle exists in a fixed location. However, in the images expanded independently by the seam



(a) 1st frame (b) 2nd frame

(c) 3rd frame (d) 4th frame

FIGURE 5. Independent seam carving result for each frame

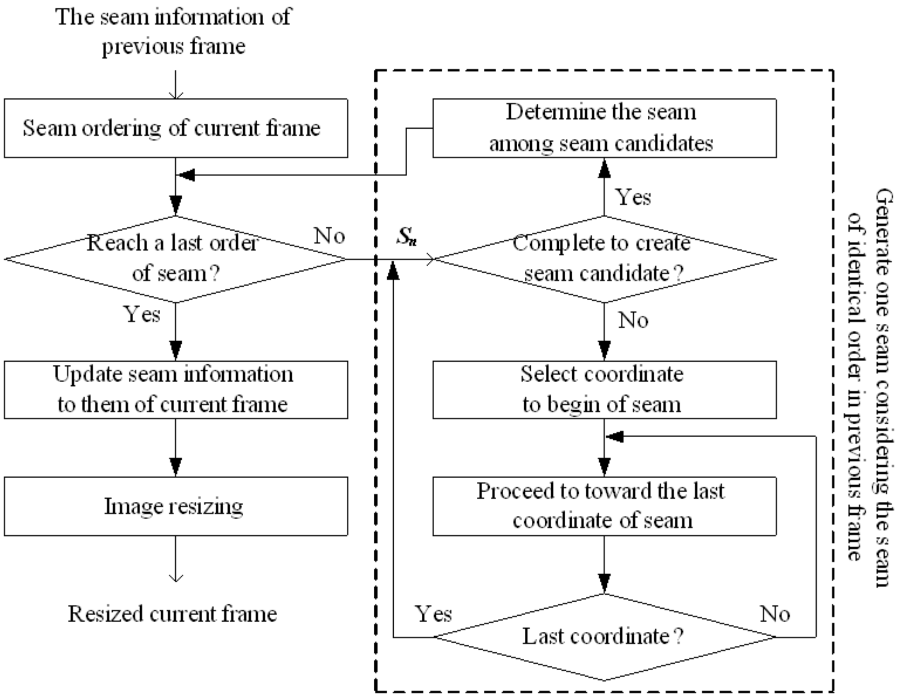

The seam information of previous frame
Seam ordering of current frame
Reach a last order of seam?
No
Yes
Update seam information to them of current frame
Image resizing
Resized current frame
S_n
Determine the seam among seam candidates
Yes
Complete to create seam candidate?
No
Select coordinate to begin of seam
Proceed to toward the last coordinate of seam
Yes
No
Last coordinate?
Generate one seam considering the seam of identical order in previous frame

FIGURE 6. Detailed process of determining seams of the current frame using seams of the previous frame

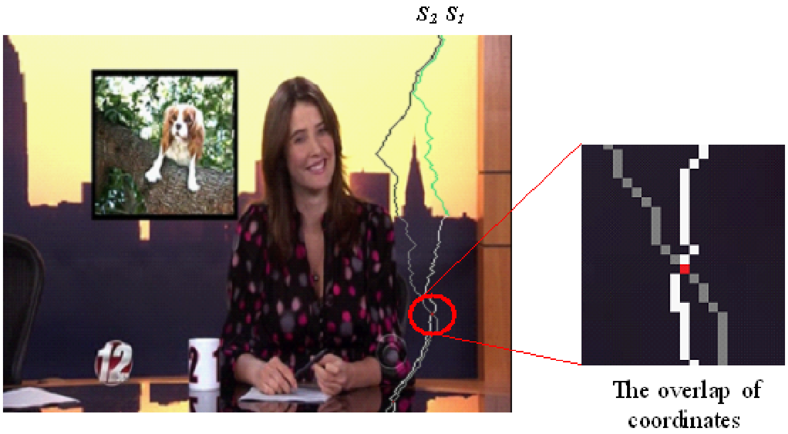


FIGURE 7. Order of seams and example of coordinate overlap

carving, the picture in the second frame moves 1 pixel to the left compared to the first frame. If these processes are repeated, the contents in the red circle shake tremendously.

Therefore, in a video, preventing the shaking phenomenon is more important than finding the optimum seam. This section presents a new process to extract seams that prevents the shaking phenomenon and preserves the form of the dominant content. Figure 6 shows the whole procedure for determining the seam of the current frame by using the seam of the previous frame.

2.3.1. *Seam ordering of current frame.* When the seam carving is performed, after removing the seam that is extracted first in order to avoid overlapping between seams, the next seam is extracted. Figure 7 shows the coordinates overlapping phenomenon in a case where seams overlap.

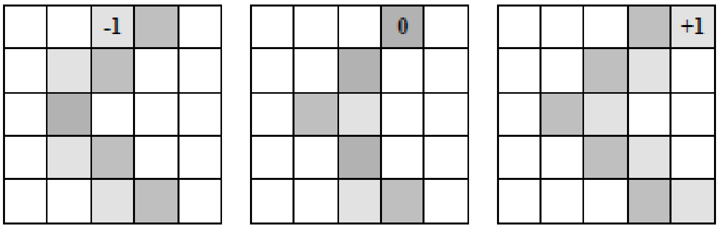


Figure 8. Examples of three kinds of starting coordinate and candidate seam

In Figure 7, overlapped coordinates are generated at the location where the first seam and the second seam meet. If the coordinate of the overlapped part is used when the image size is modified by the seams, it will be incorrect by 1 pixel at the location of the overlap. In conclusion, a distortion of the image occurs. Therefore, a specific order is used for the seams. That is, the seam order of the current frame is identical to that of the previous frame. For example, the information of the 4th seam of the previous frame is stored in order to get the 4th seam of the current frame. Equation (4) indicates that the seam information of the previous frame is referred in order to produce the seam of the current frame.

$$\boldsymbol{S}_{ref} = \boldsymbol{S}_i(n-1), \tag{4}$$

where $\boldsymbol{S}_{ref}$ has the same structure as $\boldsymbol{S}_n$ in Equation (3), and is the reference to produce the new seam. Also, n is the number of the current frame, and i is the number of the current seam.

2.3.2. *Generating candidate seam.* In determining the starting coordinate of a seam to be extracted, the starting coordinates for the reference seam of the previous frame are not used, but the n coordinates connected from side to side are considered. The reason why these connected coordinates are considered is that the locations of the next coordinates are changed according to the starting coordinates, as shown in Figure 8.

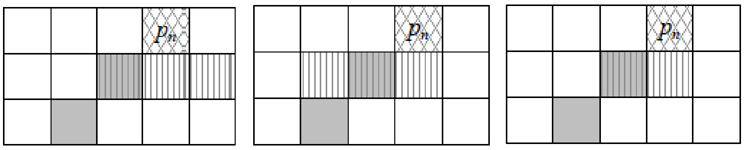
The coordinates of the reference seam $\boldsymbol{S}_{ref}$ computed in the previous frame are indicated in gray color boxes, and the striped pattern boxes indicate the coordinates of the seams computed in the current frame. The following equation indicates the process of calculating the starting coordinate from $\boldsymbol{S}_{ref}$.

$$\begin{aligned} &\boldsymbol{C} \subset \boldsymbol{S}_{ref}, \quad \boldsymbol{c} = \boldsymbol{C}(0), \\ &\boldsymbol{SP} = [\boldsymbol{c} - k,\ \boldsymbol{c} - (k-1),\ \cdots,\ \boldsymbol{c} + k], \end{aligned} \tag{5}$$

where the array $\boldsymbol{SP}$ of the starting coordinates is the set of the pixels whose distance is k or less from the first coordinates $\boldsymbol{c}$, and the seam which starts with the each element of $\boldsymbol{SP}$ is the candidate which can be the final seam. If the parameter k is too high value, temporal connection cannot be maintained, so shaking phenomenon occurs. Figure 8 shows each seam candidate if $k = 1$, that is, three possible starting coordinates are taken into consideration. The next coordinate of a seam is obtained with reference to the currently selected coordinate and the seam of the previous frame. Thus, the coordinates of candidate seam are sequentially obtained to top-down direction in vertical seam (left-right direction in horizontal seam).

First, the starting coordinate of the candidate seam is the each element of $\boldsymbol{SP}$, which becomes the determined coordinate p_0. The $\boldsymbol{p}$ is set of coordinate of seam candidate. The next coordinate p_{n+1} of p_n is obtained with reference to p_n and the reference seam $\boldsymbol{S}_{ref}$. The condition to find p_{n+1} is given by

1. p_n and p_{n+1} are spatially connected (spatial connection).
2. p_{n+1} and $\boldsymbol{C}$ ($\subset \boldsymbol{S}_{ref}$) are connected to the time axis (temporal connection).



(a) Spatial (b) Temporal (c) The last candidates (Spatial + Temporal)

Figure 9. Example of coordinate candidates

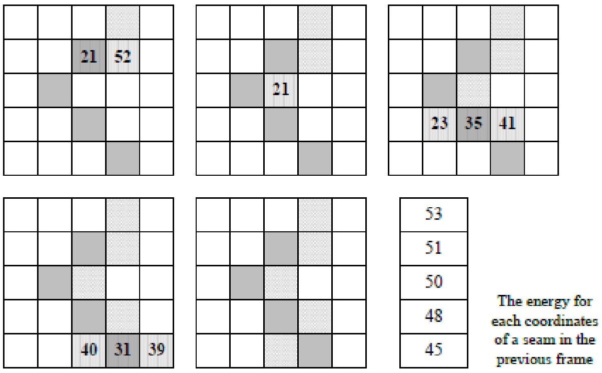

21
52
21
23
35
41
40
31
39
53
51
50
48
45
The energy for each coordinates of a seam in the previous frame

Figure 10. Example of selection and storage of seam coordinates

Equation (6) is the process of finding the candidate pixel ($\boldsymbol{CanPix}$) satisfying the above condition.

$$\begin{aligned}&\boldsymbol{CanPix} = \boldsymbol{SPA} \cap \boldsymbol{TEM},\\ &\boldsymbol{SPA} = \{x \mid p_n - 1 \le x \le p_n + 1\},\\ &\boldsymbol{TEM} = \{x \mid \boldsymbol{C} \subset \boldsymbol{S}_{ref},\ \boldsymbol{C}(n+1) - 1 \le x \le \boldsymbol{C}(n+1) + 1\},\end{aligned} \tag{6}$$

where n is x coordinate (horizontal seam) or y coordinate (vertical seam) of p_n. $\boldsymbol{SPA}$ and $\boldsymbol{TEM}$ is the pixel set satisfying the spatial connection and the temporal connection, respectively. That is, $\boldsymbol{SPA}$ includes the adjacent pixels to p_n, and $\boldsymbol{TEM}$ includes the adjacent pixels to $\boldsymbol{C}$ ($\subset \boldsymbol{S}_{ref}$). Figure 9 shows an example of the spatial connection condition, temporal connection condition, and the set $\boldsymbol{CanPix}$ satisfying two conditions.

By using that the energy distribution between the previous frame and the current frame is similar, among $\boldsymbol{CanPix}$, the pixel whose energy value is the most similar to $\boldsymbol{E}$ ($\subset \boldsymbol{S}_{ref}$) is determined as the next coordinate p_{n+1}. The energy equation is the forward energy used in the seam carving of the static images, and p_{n+1} is obtained by Equation (7).

$$p_{n+1} = \arg \min_{\alpha \in \boldsymbol{CanPix}\ \&\ \boldsymbol{E} \subset \boldsymbol{S}_{ref}} (|FE(\alpha) - \boldsymbol{E}(n+1)|), \tag{7}$$

where $FE(\bullet)$ is the function finding the forward energy.

Figure 10 shows the process where a seam is determined by comparing the energy values for candidate coordinates and the energy value for each seam coordinate of the previous frame.

$\boldsymbol{C}$ ($\subset \boldsymbol{S}_{ref}$) and the determined coordinates of the candidate seam are shown in gray color boxes and nets pattern boxes, respectively. And $\boldsymbol{CanPix}$ are shown in striped pattern boxes each step. After the energy values of $\boldsymbol{CanPix}$ are compared with $\boldsymbol{E}$ ($\subset \boldsymbol{S}_{ref}$) in the same row, the most similar pixel coordinate is selected as the next coordinate p_{n+1}. In the leftmost image in Figure 10, two candidate pixels which can be a seam in the current frame have energy values of 21 and 52. Because the candidate with an energy value that is most similar to 51 among the two candidates is the right one, the candidate with the energy value of 52 becomes the element of candidate seam in the current frame. This process is repeated until the candidate seam for the current frame is finally made.

2.3.3. *Seam determination.* In order to determine the final seam among all candidate seams obtained in above section, e_{ac} ($\subset \boldsymbol{S}_{ref}$) is compared with the total energy value for each candidate seam. The seam of which the total energy value is most similar to e_{ac} ($\subset \boldsymbol{S}_{ref}$) is selected as the final seam. Figure 11 shows the process of selecting the final seam.

$\boldsymbol{C}$ ($\subset \boldsymbol{S}_{ref}$) and the candidate seam for each starting coordinate is shown in gray color boxes and nets pattern boxes, respectively. Comparing the total energy value of each seam with the energy value of the seam in the previous frame, the seam with the smallest difference is determined to be final seam in the current frame. The information about the seam determined in the current frame as the final one are saved, and these information are then used to find the seam in the next frame.

2.4. **Image resizing.** The image size is modified by the coordinates of all the seams that are finally determined in the current frame. When reducing the image size, as many seams as necessary are deleted in the order of the seams, one at a time. On the other hand, pixel values are inserted to the coordinates of the seams in the order of the seams in the case of increasing the image size. At this time, existing image interpolation methods are used to add these pixel values. Figure 12 shows examples of the process to control the image size. As shown in Figure 12(a), a seam map is first produced by the stored seam information. The size of the seam map is identical to that of the original image, and the corresponding seam numbers are stored with the coordinates of the seams. The image size is controlled by the produced seam map. When reducing the image size, as shown in Figure 12(b), the seam map is searched and the pixels with the coordinates of the first seam are removed. After the size of the source image is reduced by one seam, in order to fit the synchronization of the coordinates, the referred seam is removed from the seam map. The image size is reduced by repeating this process for each seam to be removed.

On the other hand, when the image size is enlarged, as shown in Figure 12(c), empty spaces are inserted at the same coordinates as the coordinates of a seam. In addition, the pixel values generated by an interpolation method are filled in the empty spaces, and the image size is expanded. As the image size is increased, after the size of the source image is expanded by one seam, in order to fit the synchronization of the coordinates, the referred seam is inserted in the seam map. The target image is obtained by repeating this process.

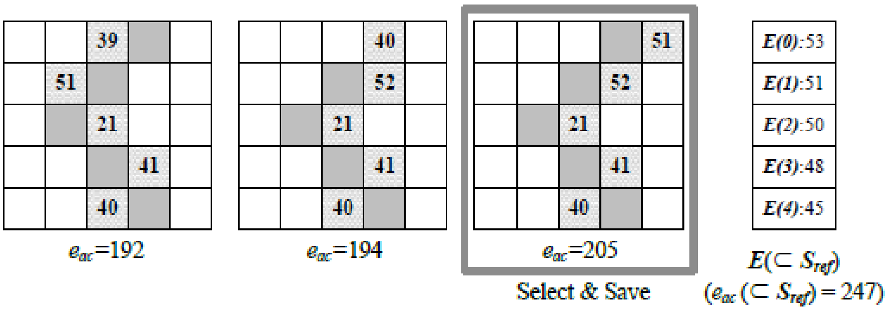
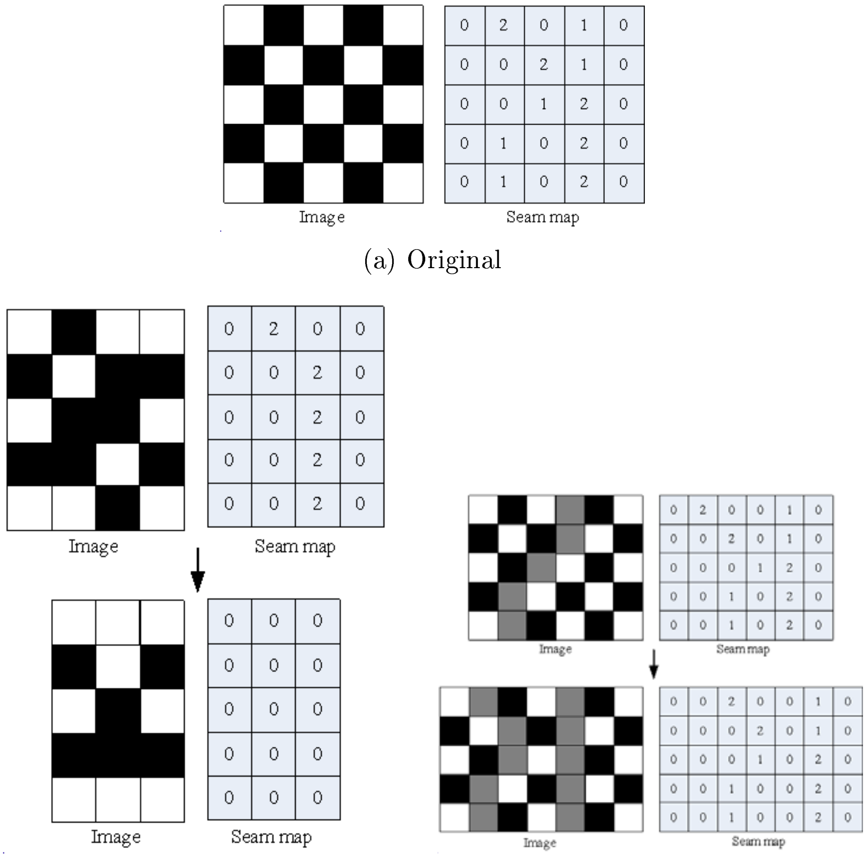


Figure 11. Example of final seam determination



0	2	0	1	0
0	0	2	1	0
0	0	1	2	0
0	1	0	2	0
0	1	0	2	0

(a) Original

0	2	0	0
0	0	2	0
0	0	2	0
0	0	2	0
0	0	2	0

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

(b) Reducing image size

0	2	0	0	1	0
0	0	2	0	1	0
0	0	0	1	2	0
0	0	1	0	2	0
0	0	1	0	2	0

0	0	2	0	0	1	0
0	0	0	2	0	1	0
0	0	0	1	0	2	0
0	0	1	0	0	2	0
0	0	1	0	0	2	0

(c) Enlarging image size

FIGURE 12. Examples of image resizing process

3. **Experimental Result.** In this section, the performance of three image resizing techniques are evaluated, namely, the bilinear method, the technique of applying Avidan's algorithm [3] to a video, and the proposed technique. A 320 × 240 resolution video was used as the test image and horizontally enlarged by 10% and 30%. First, each method was evaluated on the basis of its runtime and the average memory usage, which are the most important factors in real-time processing. The experiments were performed in the 1.86GHz dual core with 2GB memory. In order to enhance the reliability in the measured value, the same process was repeated 10 times, and the averages of the result values were compared.

Table 1 and Table 2 show the runtime and the average memory usage of each algorithm, respectively.

As the Avidan's algorithm needs many operations and the large storage space in order to analyze the entire frames in video, it cannot be performed on a system with limited resource such as a mobile terminal. However, the proposed algorithm runs about 11 $\sim$ 13 times faster than the Avidan's algorithm and achieves the comparable runtime as compared with the bilinear method as shown in Table 1.

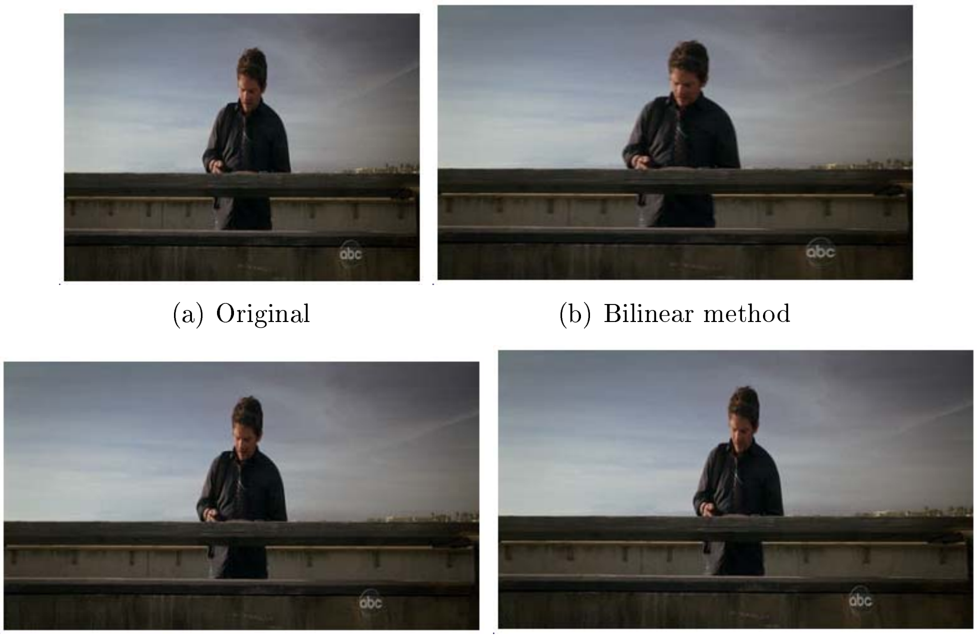
Since the proposed algorithm is designed for mobile terminal, the memory usage is also important. As shown in Table 2, the proposed method requires lower memory about 3 times than the Avidan's algorithm. Because the new seam of the current frame is computed with reference to the seam information of the previous frame, the memory usage of the proposed method is similar to that of the bilinear method which is usually performed to resize image on mobile device. Thus, the proposed method can be practically

Table 1. Run-times for different algorithms (second)

Algorithm	352 × 240 (10%)	416 × 240 (30%)
Bilinear	6.234	6.875
Avidan's	91.890	252.218
Proposed	8.296	17.437

Table 2. Memory usages for different algorithms (KB)

Algorithm	352 × 240 (10%)	416 × 240 (30%)
Bilinear	472.5	517.5
Avidan's	1,973.4	2,018.4
Proposed	565.2	790.4



(a) Original

(b) Bilinear method

(c) Avidan's algorithm

(d) Proposed

Figure 13. Results expanded in width direction by 30%

applied to resize the frames in video on mobile terminal while preventing the shaking phenomenon and preserving the important contents.

Next, whether the main content was maintained and whether the shaking phenomenon exits or not were compared through each result frame. Figure 13 shows the 6th frame from the frame results of each algorithm.

Compared with the source image in Figure 13(a), the result of the bilinear technique in Figure 13(b) indicates that the shape of the human face has been broadened. However, in the images results from Avidan's algorithm and the proposed algorithm, the shapes of the face are similar to that in the original image. Thus, it is seen that the proposed algorithm maintains the main content of the image.

TABLE 3. Error rates for different algorithms

Algorithm	352 × 240 (10%)	416 × 240 (30%)
Avidan's	66.879	77.472
Proposed	3.583	6.204

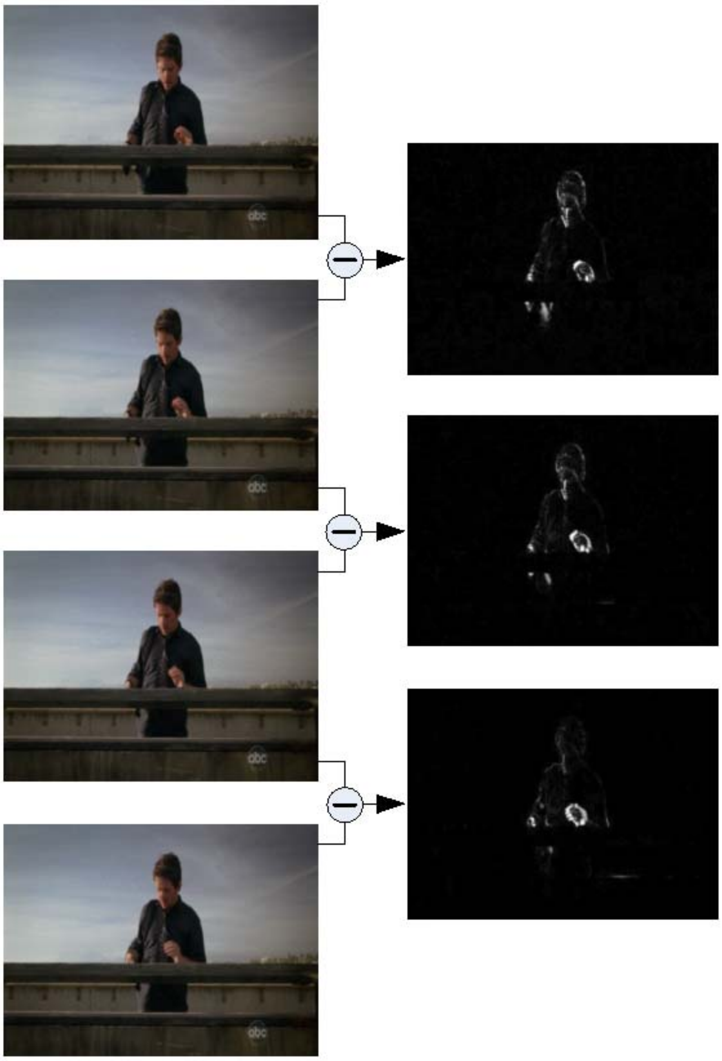
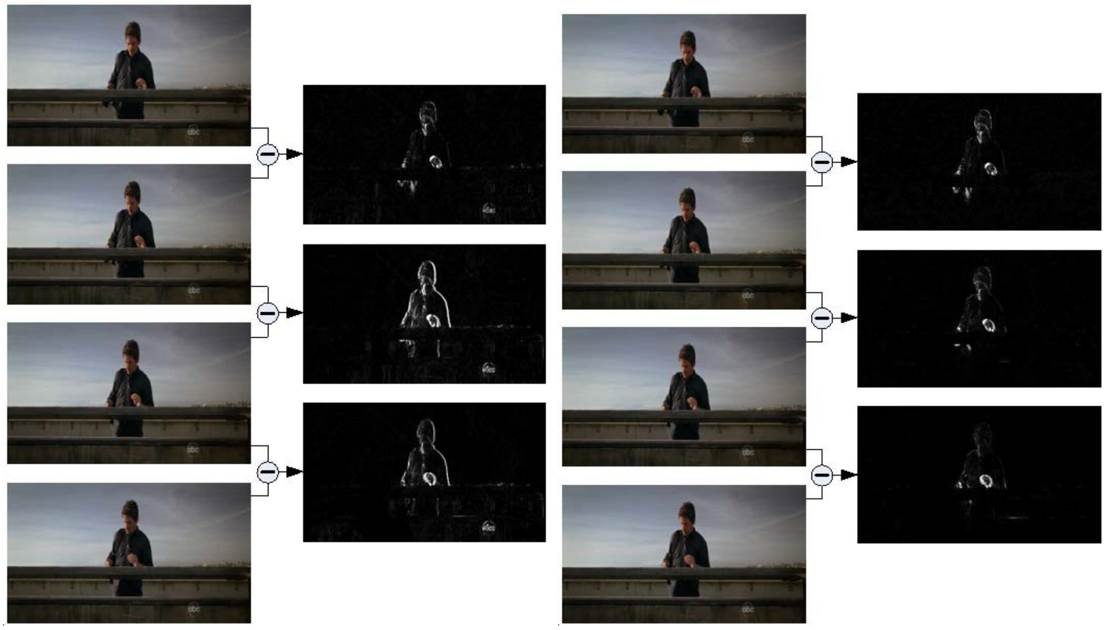


FIGURE 14. Differences between adjacent frames in original video

Finally, the differences between the experimental results and source image are shown as Error Rate given by

$$
\begin{aligned}
D_n &= \sum_{j=0}^{height-1} \sum_{i=0}^{width-1} |f_n(i,j) - f_{n+1}(i,j)|, \\
\zeta &= \frac{1}{height \times width} \sum_{k=1}^{K-1} D_k, \quad \mathit{Error\ Rate} = \left|1 - \frac{\zeta_0}{\zeta}\right| \times 100,
\end{aligned} \tag{8}
$$

where f_n indicates R, G, B values of the nth frame, and D_n shows the error per pixel between nth frame and $(n+1)$th frame. K is the number of total frames, and ζ_0 is the error between frames in the original video. *Error Rate* represents the difference between original video and the result video. Table 3 shows numerically how many differences the



(a) Avidan's algorithm (b) Proposed

Figure 15. Differences between adjacent frames after applying Avidan's and proposed algorithm

result images by the proposed method and the Avidan's method shows with the original video by *Error Rate*.

As shown in Table 3, the result images by the proposed method have the smaller error rate and are more similar to the original video than those of the Avidan's method.

Figure 14 shows the differences between adjacent frames in the original video. Because these frames belong to a single shot, any differences between adjacent frames are small.

As shown in Figure 15(a), because the technique applying Avidan's algorithm to video does not consider the relation between adjacent frames, the shaking phenomenon occurs and many differences between neighboring frames are generated. On the other hand, because the proposed algorithm considers the correlation between adjacent frames, there is no shaking phenomenon and the differences between neighboring frames are similar to those in original video as shown in Figure 15(b).

4. **Conclusions.** In this paper, a new real-time video scaling technique that preserves the contents of a video was proposed. Because a correlation exists between consecutive frames in a video, by determining the seam of the current frame with reference to the seam of the previous frame, we were able to propose a real-time video scaling technique without the shaking phenomenon for the contents and analyzing the entire video. The conventional seam carving requires too much complexity and a large amount of memory because the entire frames in video have to be analyzed. Therefore, the conventional seam carving cannot be performed on a system with mobile terminal. The proposed algorithm has a fast processing speed similar to that of the bilinear method, while preserving the main content of an image to the greatest extent possible. In addition, because the memory usage is remarkably small compared with the existing seam carving method, the proposed algorithm is usable in mobile terminals which have limited memory resources. Computer

simulation results indicate that the proposed technique provides better objective performance, subjective image quality, shaking phenomenon removal, and content conservation than conventional algorithms.

Acknowledgment. This study was supported by the Research Grant from Kangwon National University. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] A. Santella, M. Agrawala, D. Decarlo, D. Salesin and M. Cohen, Gaze-based interaction for semi-automatic photo cropping, *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pp.771-780, 2006.

[2] F. Liu and M. Gleicher, Automatic image retargeting with fisheye-view warping, *Proc. of Symposium on User Interface Software and Technology*, pp.153-162, 2005.

[3] S. Avidan and A. Shamir, Seam carving for content-aware image resizing, *ACM Transactions on Graphics, Proc. of ACM SIGGRAPH*, vol.26, no.3, 2007.

[4] V. Setlur, S. Takagi, R. Raskar, M. Gleicher and B. Gooch, Automatic image retargeting, *Proc. of the 4th International Conference on Mobile and Ubiquitous Multimedia*, vol.154, pp.59-68, 2005.

[5] C. Tao, J. Jia and H. Sun, Active window oriented dynamic video retargeting, *ICCV Proc. of Workshop on Dynamical Vision*, pp.1-12, 2007.

[6] S. Cho, Y. Matsushita and S. Lee, Image retargeting with importance diffusion, *Proc. of KIISE Korea Computer Congress*, vol.35, no.1(B), pp.236-239, 2008.

[7] V. Setlur, T. Lechner, M. Nienhaus and B. Gooch, Retargeting images and video for preserving information saliency, *IEEE Computer Graphics and Applications*, vol.27, no.5, pp.80-88, 2007.

[8] F. Liu and M. Gleichar, Video retargeting: Automating pan and scan, *Proc. of ACM International Conference on Multimedia*, pp.241-250, 2006.

[9] M. Rubinstein, A. Shamir and S. Avidan, Improved seam carving for video retargeting, *ACM Transactions on Graphics*, vol.27, no.3, 2008.

[10] B. Chen and P. Sem, Video carving, *Short Papers Proc. of Eurographics*, 2008.

[11] U. Gargi, R. Kasturi and S. H. Strayer, Performance characterization of video-shot-change detection methods, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.10, no.1, pp.1-13, 2000.

[12] Y. Gong, An accurate and robust method for detecting video shot boundaries, *IEEE Multimedia Computing and Systems*, vol.1, pp.850-854, 1999.

[13] L. Chen, X. Xie, X. Fan, W. Ma, H. Zhang and H. Zhou, A visual attention model for adapting images on small displays, *Multimedia Systems*, vol.9, no.4, pp.353-364, 2003.

[14] H. Liu, X. Xie, W. Ma and H. Zhang, Automatic browsing of large pictures on mobile devices, *Proc. of the 11th ACM International Conference on Multimedia*, pp.148-155, 2003.

[15] B. Suh, H. Ling, B. B. Bederson and D. W. Jacobs, Automatic thumbnail cropping and its effectiveness, *Proc. of the 16th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, pp.95-104, 2003.

[16] X. Fan, X. Xie, H. Q. Zhou and W. Y. Ma, Looking into video frames on small displays, *Proc. of the 17th ACM International Conference on Multimedia*, pp.247-250, 2003.

[17] J. Xiao, X. Zou, Z. Liu and X. Guo, A novel adaptive interpolation algorithm for image resizing, *International Journal of Innovative Computing, Information and Control*, vol.3, no.6(A), pp.1335-1345, 2007.

[18] Y. Zhang, S. Gao, C. Zhang and J. Chi, Application of a bivariate rational interpolation in image zooming, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4299-4307, 2009.

[19] P.-H. Lin, H.-K. Chu and T.-Y. Lee, Smooth shape interpolation for 2D polygons, *International Journal of Innovative Computing, Information and Control*, vol.4, no.9, pp.2405-2417, 2009.

[20] Y. Tian, C. Zhang, F. Peng and S. Zheng, An iterative hybrid method for image interpolation, *International Conference on Intelligent Computing*, pp.10-19, 2005.

[21] J. Xiao, X. Zou, Z. Liu and X. Guo, Adaptive interpolation algorithm for real-time image resizing, *Proc. of International Conference on Innovative Computing, Information and Control*, vol.2, Beijing, China, pp.221-224, 2006.

[22] R. Bellman, Some problems in the theory of dynamic programming, *Econometrica*, vol.22, no.1, pp.37-48, 1954.

[23] D. P. Bertsekas, *Dynamic Programming and Optimal Control (2 Vol Set)*, 3rd Edition, Scientific, Athena, 2007.

[24] P. Kohli and P. H. S. Torr, Dynamic graph cuts for efficient inference in Markov random fields, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.29, no.12, pp.2079-2088, 2007.

[25] V. Kwatra, A. Sch'odl, I. Essa, G. Turk and A. Bobick, Graphcut textures: Image and video synthesis using graph cuts, *ACM Transactions on Graphics*, vol.22, no.3, pp.277-286, 2003.