International Journal of Innovative
Computing, Information and Control
Volume 8, Number 5(A), May 2012

 ISSN 1349-4198
pp. 3135–3148

MULTI-VIEW VIDEO GENERATION FROM 2-DIMENSIONAL VIDEO

Yun-Ki Baek[1], Young-Ho Seo[2], Dong-Wook Kim[3] and Ji-Sang Yoo[1]

[1]Department of Electronic Engineering
[2]College of Liberal Arts
[3]Department of Electronic Materials Engineering
Kwangwoon University
447-1, Welgye-Dong, Nowon-Gu, Seoul 139-701, Korea
{jsyoo; yhseo; dwkim}@kw.ac.kr

Received August 2010; revised December 2010

Abstract. *This paper proposes a region-based method to generate a multi-view video from a conventional 2-dimensional video. It first segments an image according to its color information and our scheme is based on the resulting segmented regions. The color information is used to extract the boundary of an object, which is very hard if the motion information is used. To classify the homogeneous regions, both the luminance and chrominance components are used. A pixel-based motion estimation is also performed to obtain accurate motion information. Then, we convert the motion values to corresponding depth values for each segmented region. Finally, we generate multi-view video by applying a rotational transformation method to 2-dimensional input images using the obtained depth information of each region. The ability and the resulting image quality are tested by a subjective assessment, called Double Stimulus Continuous Scale, and compared with an existing method. The experimental results showed that the image quality of the proposed algorithm was much better than the precious one.*
Keywords: Multi-view video, Ross phenomenon, Segmentation, Depth information, Pixel-based motion estimation

1. **Introduction.** HDTV and DMB have enhanced video by the aid of better image display apparatuses. These technologies have tried to provide the best service in the limited provisions for 2-dimensional (2D) views. However, human desire has not stopped there, and has been requiring more realistic video which can make the viewer feel as if they are inside the scene. Such a desire naturally leads to increasing the interest in 3D view, which results in active progress in the research of 3D video. In Europe, the 3DTV projects of COST230 (1992∼1994), PANORAMA (1995∼1998), ATTEST (2002∼2004) and 3DTV (2004∼2009) were performed [1-3]. In Korea, the Electronics and Telecommunications Research Institute (ETRI) succeeded in configuring a stereo image broadcasting system and demonstrating its performance by broadcasting the 2002 FIFA Korea-Japan World Cup in a real time. In Japan, a number of studies on 3D video, such as the 3D HDTV project by NHK, are making steady progress. Also, the 3DAV Group of MPEG has standardized the multi-view video encoding (MVC) method [4].

Multi-view video is typically obtained by capturing video images from multiple viewpoints with multiple cameras. It can overcome the shortcoming of stereo video which is that the viewpoint should be fixed, which is the most important feature of multi-view video. However, using the video images acquired by multiple cameras has a serious disadvantage in that the number of data increases linearly to the number of views, and it becomes massive for a large number of viewpoints. In order to efficiently acquire stereo or multi-view video and alleviate this problem, the methods to acquire multi-view video

images using a depth camera have been studied. A depth camera usually acquires color texture data and 8-bit depth information. In order to obtain stereo or multi-view video from the data by a depth camera, it is necessary to convert depth information into corresponding parallax information and then compose a stereo video appropriate for a given view. Therefore, the algorithms to process the depth information to compose stereo video have been researched [5-7]. However, still, multi-view cameras or depth cameras have not been sufficiently provided to the public to be used as a tool to acquire multi-view video.

2D-3D conversion is a technique to generate stereo video from 2D video, which is based on the Ross phenomenon [8]. The key point is that a binocular time delay may cause a 3D effect. If it is possible to use a 2D-3D conversion technique to get a 3D video, not only can the conventional 2D camera or equipment be used to get a 3D video, but also many existing 2D content can be converted to 3D content with low cost and high efficiency. Thus, many studies have been focused on 2D-3D conversion techniques. However, most of them generate only a stereo image, that is, a pair of left and right images, which limits the view field. Moreover, the 3D effects by the binocular time delay were not good enough to feel realistic.

In this paper, a method to convert 2D video into multi-view video is proposed to overcome the problems in the existing 2D-3D conversion methods and enhance better 3D effects. Our method basically uses regional depth and motion information. To do this, it segments a given image into homogeneous regions or objects by using color information. Those segments are the units in estimating motion. However, the real motion is estimated pixel by pixel and the results are aggregated to form the motion of a region or object. Then, the motion of each region is converted into corresponding depth information, which is used to render the multi-view image. For depth information, we perform post-processing to prevent the depth discontinuity in the succeeding frames. The proposed method is tested and compared with an existing method to qualify the 3D effects.

In the next section, the conventional 2D-3D conversion method is described. Our algorithm to generate multi-view video from 2D video is explained in Section 3 and it is experimented and analyzed in Section 4. Finally, this paper is concluded in Section 5, on the basis of the experimental results.

2. Conventional Method.

Generating a 3D video consisting of left and right images from a 2D video has used the modified time difference (MTD) technique which is based on the Ross phenomenon [9-11]. Figure 1 shows an example of the Ross phenomenon, where the mountain that is regarded as a background is fixed, while the airplane that is regarded as an object moves to the right. Thus the current image can be used as the left image and the delayed image as the right image. In this case, the mountain retains zero parallax at which the convergence point is located on the screen and the airplane has a negative parallax and its convergence point is created in front of the screen. There are some other techniques to separate an object from background such as [12,13].

The MTD technique generates a 3D image by using the delayed image according to the Ross phenomenon. Therefore, many of the studies for conventional 2D-3D conversion have focused on the technique to find the best delayed image to be used as a right image. Other techniques such as the ones in [14] can be used to find a proper delayed image. The basic conversion process for the MTD technique is illustrated in Figure 2. From a conventional 2D video, motion vectors and the type of the image are extracted and examined to determine if the delayed image can be used as the right image. If it is possible, the most appropriate delayed image is selected as the right image by referring to the horizontal components of the motion vectors and the image type. Of course a post process could be performed, if necessary. But in such a case when the scene changes or

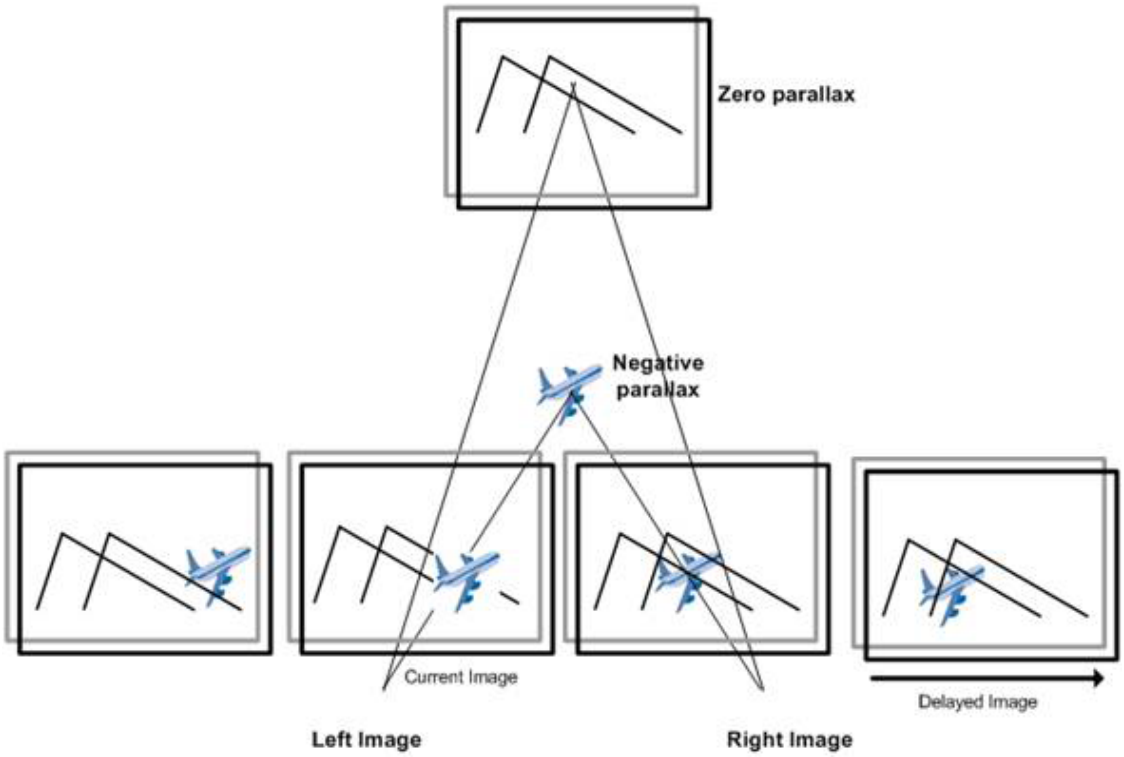


FIGURE 1. An example of the Ross phenomenon

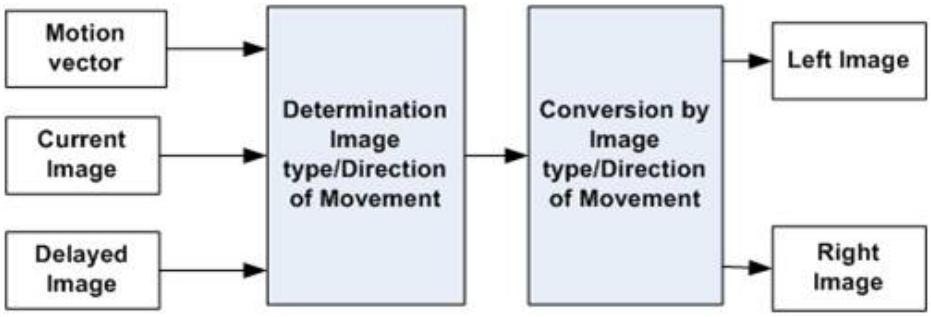


FIGURE 2. Brief block diagram of 2D-3D conversion process

objects move more vertically, it is impossible to apply the MTD technique. Thus other techniques are required for those cases and we intend to solve that problem in this paper.

3. **Proposed Method.** If a near object (the large airplane in Figure 3, for example) and a far one (the small airplane in Figure 3) move at the same real speed in a 2D video, the near one moves faster than the far one. If the moving speeds of the two are the same, the real speed of the far one is higher than the near one. Such a fact indicates that the motion information may reflect the corresponding depth information, which this paper uses to develop a technique to generate a multi-view video from a 2D video. For this, we first separate each object from a background to get its depth information.

The proposed method consists of three phases as shown in Figure 4. Phase 1 is region segmentation and motion estimation. Phase 2 is extraction of depth information. And Phase 3 is multi-view video generation. In Phase 1, the given 2D input image is segmented into regions using color information. Also, the motion intensity is calculated by estimating the motion between the current image and the previous image. In Phase 2, the depth

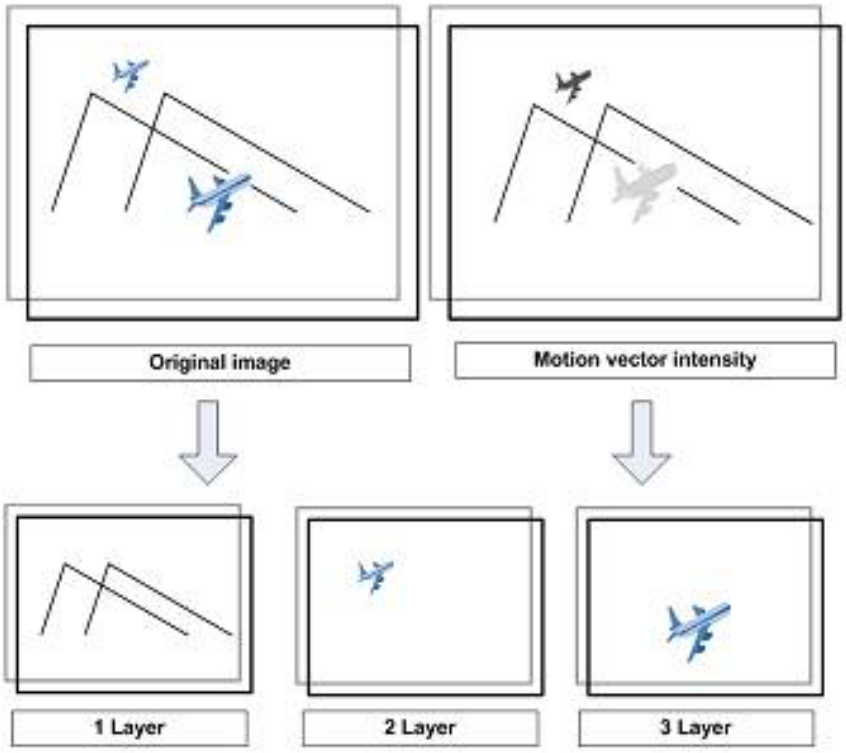


FIGURE 3. Transforming motion vector into depth

information for each segmented region is extracted from the motion information. Finally in Phase 3, the corresponding multi-view images are generated by transforming the 2D input image using depth information. The detailed processes are described as follows.

3.1. **Region segmentation.** First of all, we segment an image into regions, each of which consists of highly homogeneous pixels. For this process, we use the color information of the image. That is, we use the similarity of colors as an index in segmenting the image [15]. Another purpose of this segmentation is to extract the accurate boundary of each region or object. Because the motion estimation with a measurement window, which is quite usual, may cause bigger motion exceeding the boundary of an object, the estimation based on the segmented region can revise the motion value.

The segmentation process also consists of two steps: region segmentation by initial merging and region merging by small region processing.

3.1.1. *Region segmentation by initial merging.* First, each pixel in a 2D video is initialized to belong to a separate region. In the initial region merging step, regions with similar colors are merged into one region. For this, luminance and chrominance components are used as the indices for color-similarity, which is used as a cost function. The cost value between region R_i and region R_j for each of Y, U and V component is calculated as Equation (1).

$$C_p(R_i, R_j) = \mid P(R_i) - P(R_j) \mid \tag{1}$$

where, $P \in \{Y, U, V\}$ and $P(R_i)$ is the mean value of P component in region R_i and is calculated as Equation (2).

$$P(R_i) = \frac{1}{N_i} \sum_{k=0}^{N_i - 1} P_k \tag{2}$$

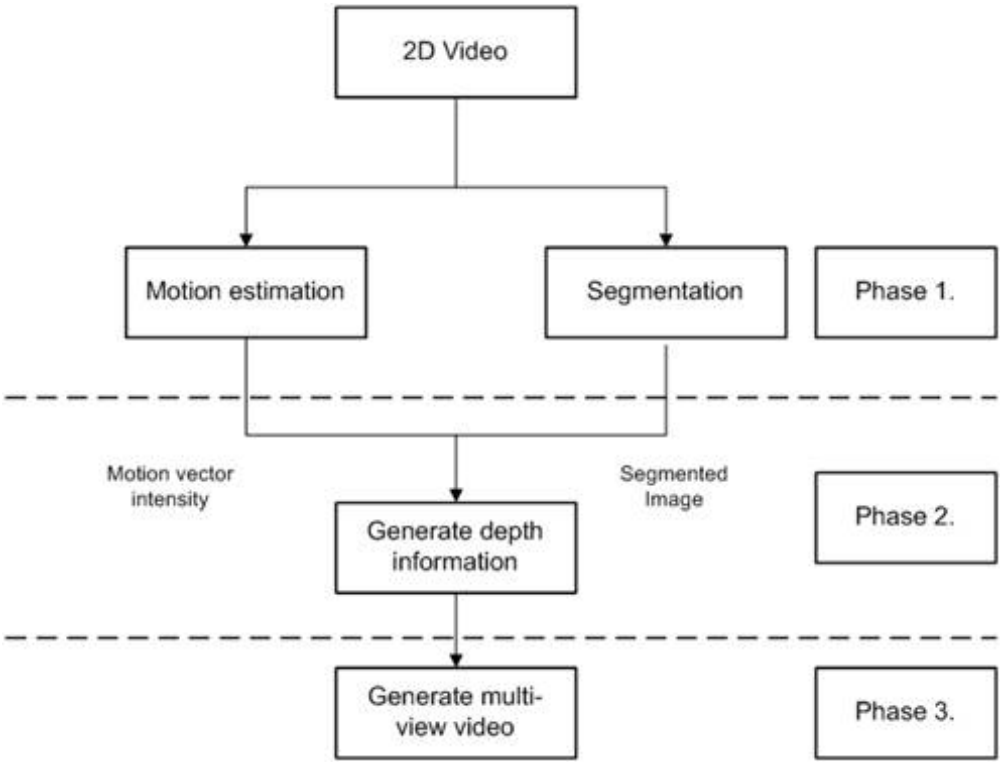


FIGURE 4. Processing flow diagram of the proposed method

where N_i is the number of pixels in region R_i and P_k is the value of P component of k^{th} pixel.

The cost value of Equation (1) should be smaller enough for two regions R_i and R_j to be similar. Thus we set up a criterion for these two regions to be merged as Equation (3), where $T_{p,merge}$ is the threshold value of P component. In this paper, we set these values as $T_{Y,merge} = T_{U,merge} = T_{V,merge} = 5$ so as to prevent incorrect merging caused by assigning too large numbers to them.

$$[C_Y(R_i, R_j) < T_{Y,merge}] \cap [C_U(R_i, R_j) < T_{U,merge}] \cap [C_V(R_i, R_j) < T_{V,merge}] \tag{3}$$

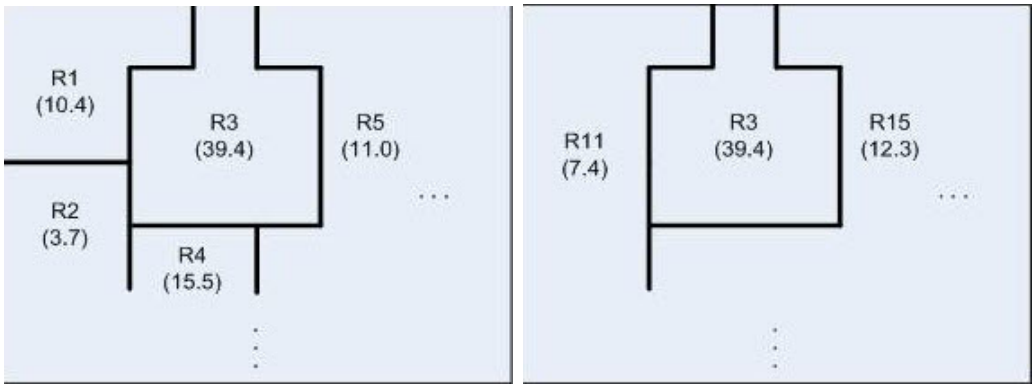
After merging R_i and R_j, the Y, U and V values of the resulting region are reset as Equation (4) assuming that the resulting region is named as R_m, where $P_{(i,k)}$ is the value of P component of the k^{th} pixel in region R_i.

$$P(R_m) = \frac{1}{N_i + N_j} \left(\sum_{k=0}^{N_i - 1} P_{(i,k)} + \sum_{k=0}^{N_j - 1} P_{(j,k)} \right) \tag{4}$$

3.1.2. *Region merging by small region processing.* The segmentation and initial merging process explained above may result in over-segmented regions. This over-segmentation may cause errors in assigning a depth value to each segmented region, which is the next process to be explained afterward.

The last step of segmentation is to avoid these over-segmented regions by additional merging, which is carried out only for the regions whose number of pixels is smaller than a pre-defined number. The methodology is to merge the over-segmented region to one of the adjoining regions whose cost value is the nearest to the one to be merged. The cost function for this additional merging is as Equation (5):

$$C(R_i, R_j) = C_Y(R_i, R_j) + C_U(R_i, R_j) + C_V(R_i, R_j) \tag{5}$$



(a) (b)

FIGURE 5. Example of additional merging: (a) before and (b) after

In detail, the additional merging scheme is as follows:

1) For each region resulting from the initial merging step, the number of pixels in the region checked first to see if it is less than the predefined number (we call it the less-pixel region).

2) For each of the less-pixel regions R_i, Equation (5) is calculated for each of the adjoining regions R_j.

3) Among the adjoining regions, the one with a minimum C value is taken and is merged with region R_i to form a new region, R_m and the new mean values for Y, U and V for R_m are calculated in Equation (4).

4) Repeat from 1) to 3) until there are no less-pixel region.

Figure 5 shows an example of this merging process. Here only one of the mean values of Y, U, and V are used for simplicity. Figure 5(a) is the result from initial merging. Among the regions, R1 and R4 are assumed to be the less-pixel regions. For R1, only R2 and R3 are adjoined to R1. Because $C(R_1, R_2)$ is less than $C(R_1, R_3)$, R1 and R2 are merged to R11 and the resulting $P(R11)$ is 7.4. For R4, R2, R3, R5 are adjoined and $C(R_4, R_5)$ is the minimal. Thus R4 is merged to R5 to form R15, whose resulting P value is 12.3.

3.2. **Motion estimation.** As mentioned earlier with Figure 3, the motion of an object may include the depth information of the object. Thus, we estimate the motions to use to convert to depths. For this study, pixel-based motion estimation is performed by assuming that each pixel has a motion, which enables more reliable and elaborate results than the block-based one [16].

Three factors affect the performance of the pixel-based motion estimation: the size of the measurement window, the size of the search range and the cost function. First, the measurement window should be as large as possible to find more accurate motion, which needs more computation cost. The size of the search range is totally dependent of the motion speed. If it is too small, the probability to find the exact motion decreases. But if it is too large, the possibility to find more than one solution increases with more computational cost. One more factor to be considered is the direction of the motion. If the motion is horizontal-dominant (or vertical-dominant), the horizontally-wide (or vertically-wide) search window is more effective. To consider the motion speed, the direction of the motion, and the calculation cost between two consecutive frames, the initial motion is obtained with 16×16 blocks first. Based on this initial motion, the search range is set adaptively from ± 15 to ± 31. Also, the measurement window is established as 10×5, 5×10, or 8×8 depending on the direction of the motion. Rectangular regions in Figure 6

(a) (b) (c)

FIGURE 6. Examples of the adaptive measurement window for: (a) horizontally-dominant motion, (b) vertically-dominant motion and (c) general motion

show examples of the measurement windows: (a) is for a horizontally-dominant motion, (b) is for a vertically-dominant motion and (c) is for a general motion, respectively.

Our pixel-based motion estimation takes more computational time than the block-based one. But it can be compensated somewhat by the horizontally-wide and vertically-wide measurement windows. Also the adaptive size of search range in our method can reduce the computation time.

As the cost function, we use all the three color components, Y, U and V as Equation (6).

$$MC_{i,j}(x,y) = \sum_{P=Y,U,V} \sum_{p=x-a}^{x+a} \sum_{q=y-b}^{y+b} \mid P_R(p-i, q-j) - P_C(p,q) \mid \tag{6}$$

where $MC_{i,j}(x,y)$ is the cost value of the pixel at (x, y) to have motion of (i,j), $a(b)$ is the search range of x (y) direction, and $P_R(x,y)(P_C(x,y))$ is the $P(=$ Y, U or V) value of the pixel at (x, y) of the reference (current) image. With this cost function the estimated motion vector is found as Equation (7).

$$MV(x,y) = \arg_{i,j}[\min(MC_{i,j}(x,y))] \tag{7}$$

3.3. **Depth information generation.** The estimated motion is converted into a corresponding depth value as Equation (8),

$$D(x,y) = Motion_x + Motion_y \tag{8}$$

where $D(x,y)$ is the converted depth value of the pixel at (x, y) and $Motion_x$ ($Motion_y$) is the motion intensity in x (y) direction.

Because rendering the multi-view video from a 2D video in this paper is a region-based process, and it uses the region's depth information, it is necessary to estimate the depth of each region. Because the pixel depths in a region are not the same, in general, we average the pixel depths in a region to use as the depth of the region, which is shown in Equation (9),

$$D_t(R_i) = \frac{1}{N_i} \sum_{k=0}^{N_i-1} D_{t,i}(k) \tag{9}$$

where $D(R_i)$ is the depth of region R_i at time t, N_i is the number of pixels in R_i, and $D_{t,i}(k)$ is the depth of the k^{th} pixel in R_i at time t.

The motion of an object or a region may vary frame by frame, even though the variation is not large. If this variation is not considered, the rendered multi-view images might look as if the scene is interrupted or frozen. To avoid this phenomenon, we use the mean

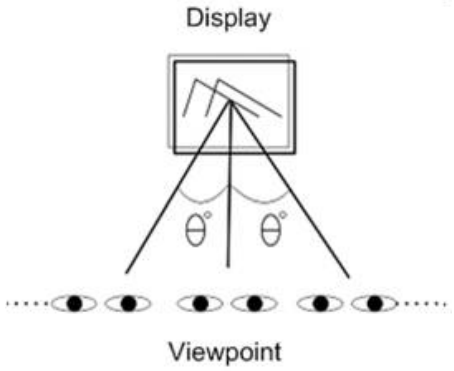


FIGURE 7. Multiple viewpoints

depth of the three consecutive frames of past ($D_{t-1}(R_i)$), current ($D_t(R_i)$) and future ($D_{t+1}(R_i)$), as Equation (10).

$$D(R_i) = \frac{1}{3}[D_{t-1}(R_i) + D_t(R_i) + D_{t+1}(R_i)] \tag{10}$$

3.4. **Multi-view video generation.** In general, a multi-view image displays multi-angled views as shown in Figure 7. In a viewpoint a pair of views (one for the left, and one for the right) makes the viewer feel as if they are looking at a 3D image. Assume that the angle between two adjacent viewpoints is θ. Then, the image projected to the viewers eyes at a new position would be the images rotated by the angle of θ. The position of the image pixel rotated by the angle of θ is determined by Equation (11) [18],

$$\begin{bmatrix} x_\theta \\ y_\theta \\ z_\theta \end{bmatrix} = R(\theta) \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{11}$$

where $(x_\theta, y_\theta, z_\theta)$ is the coordinate of the image rotated by the angle of θ, (x, y, z) is the coordinate of the original image, and $R(\theta)$ is the rotational conversion matrix of Equation (12).

$$R(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{12}$$

Because the original image contains the x and y coordinates of each pixel and the corresponding depth information gives the z coordinate, Equation (11) can be solved easily. However, by comparing it to the original image, the rotated image has an occluded region (resides in the rotated image but not in the original image) and a dis-occluded region (resides in the original image but not in the rotated image). The amount of the occluded pixels and dis-occluded pixels is dependent of the rotation angle, the depth of the object, and the surface morphology of the object. Because it is not easy to consider all the three factors, we only considered the rotation angle and the depth of the object.

In the rotated image, the occluded regions should be created and the dis-occluded regions should be removed. In general, one of the two methods is used to process the occluded region: linear interpolation of foreground and/or background and extrapolation of the background color information. In this paper, the linear interpolation method is used to process the occluded region and the inverse process is applied to treat the dis-occluded region.

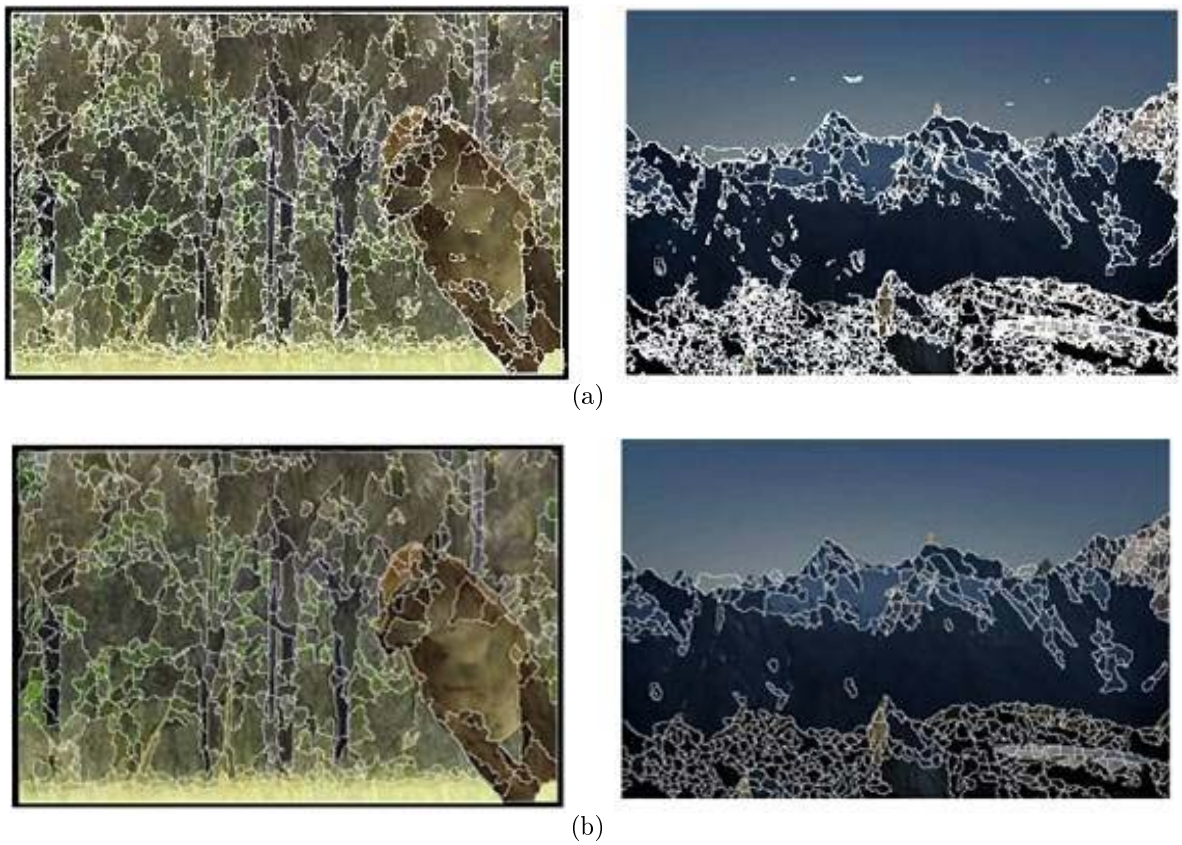


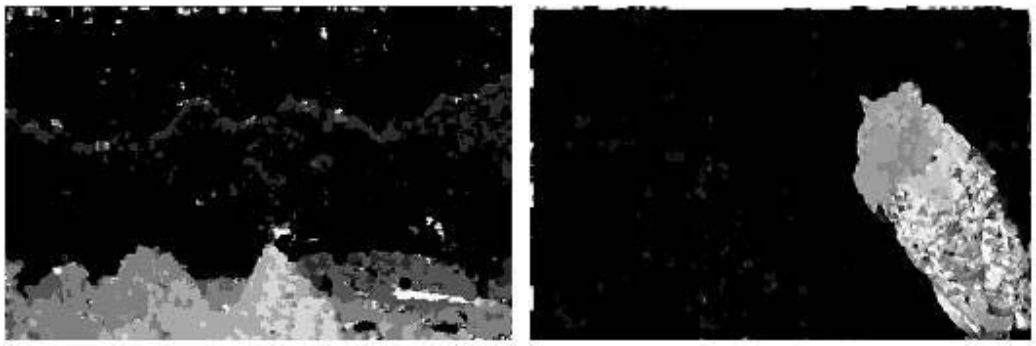
FIGURE 8. Examples of the region segmentation: (a) before and (b) after small region process

4. **Experiment Results and Analyses.** We have experimented with each processing step (region segmentation, motion estimation and depth information extraction, and multi-view video generation) of our scheme with dozens of videos.

4.1. **Experimental result of each step of the algorithm.** Figure 8 shows two examples (left and right) resulting from the region segmentation. In this figure, the ones in Figure 8(a) are the result from segmentation by initial merging (3.1.1) and the ones in Figure 8(b) are the result after performing small-region processing (3.1.2). Before small region processing, the sky region in the upper part of the right image is merged into almost one region while the complicated regions in the lower part include a lot of small regions. There are a lot of small regions in the left image also because of complicated shapes of leaves. Those two images seem inappropriate as the results of the region segmentation. On the other hand, in the two images in Figure 8(b), the extremely small regions are merged by the small region processing to form the appropriate segmented results.

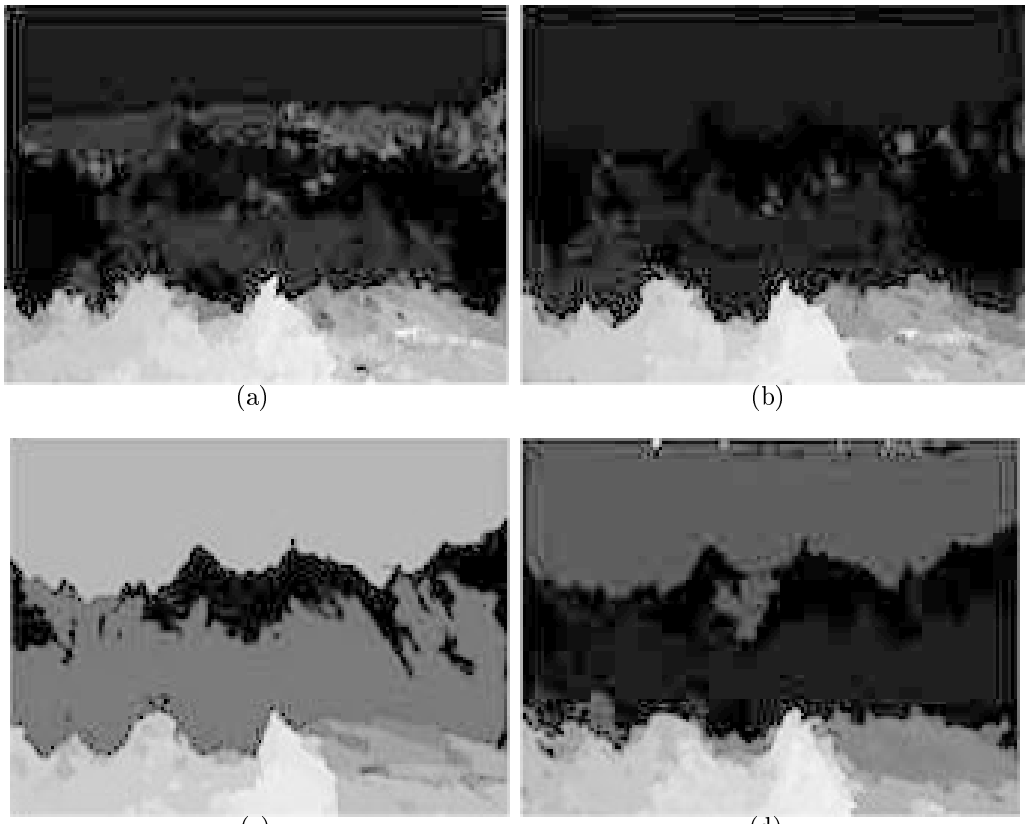
The next process is to extract the pixel-based motion estimation and depth extraction. Figure 9 shows two examples, the one (a) for an image in which the horizontal motion is dominant and the other (b) in which the vertical motion is dominant. Note that the depth values in these figures are adjusted to be more configurable. The measurement window and the search range are varied as explained in 3-2. As can be seen from the depth images in Figure 9, the pixel-based depths are not proper to use to render a view with a different viewing angle.

The pixel-based depths are converted to region-based ones by averaging the pixel depths region by region, which was extracted by region segmentation as Figure 8. The example



(a) (b)

FIGURE 9. The example results of the pixel-based motion estimation and depth extraction: (a) image that horizontal motion is dominant and (b) image that vertical motion is dominant



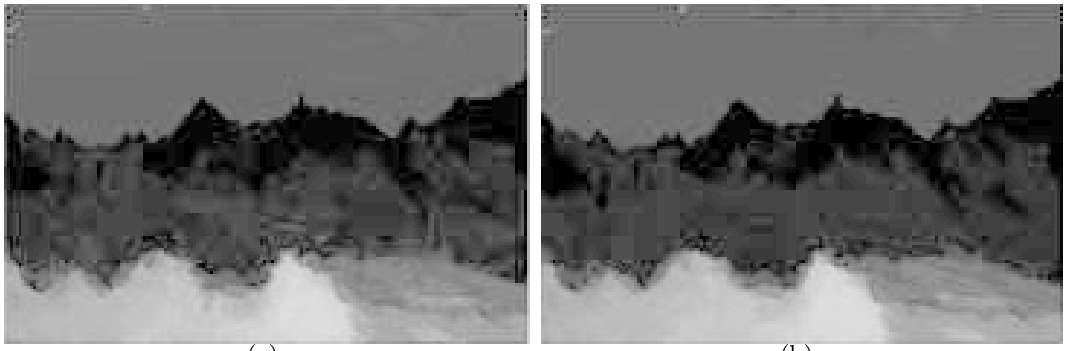
(a) (b)

(c) (d)

FIGURE 10. Examples of region-based depth maps: (a) 1st frame, (b) 2nd frame, (c) 3rd frame and (d) 4th frame

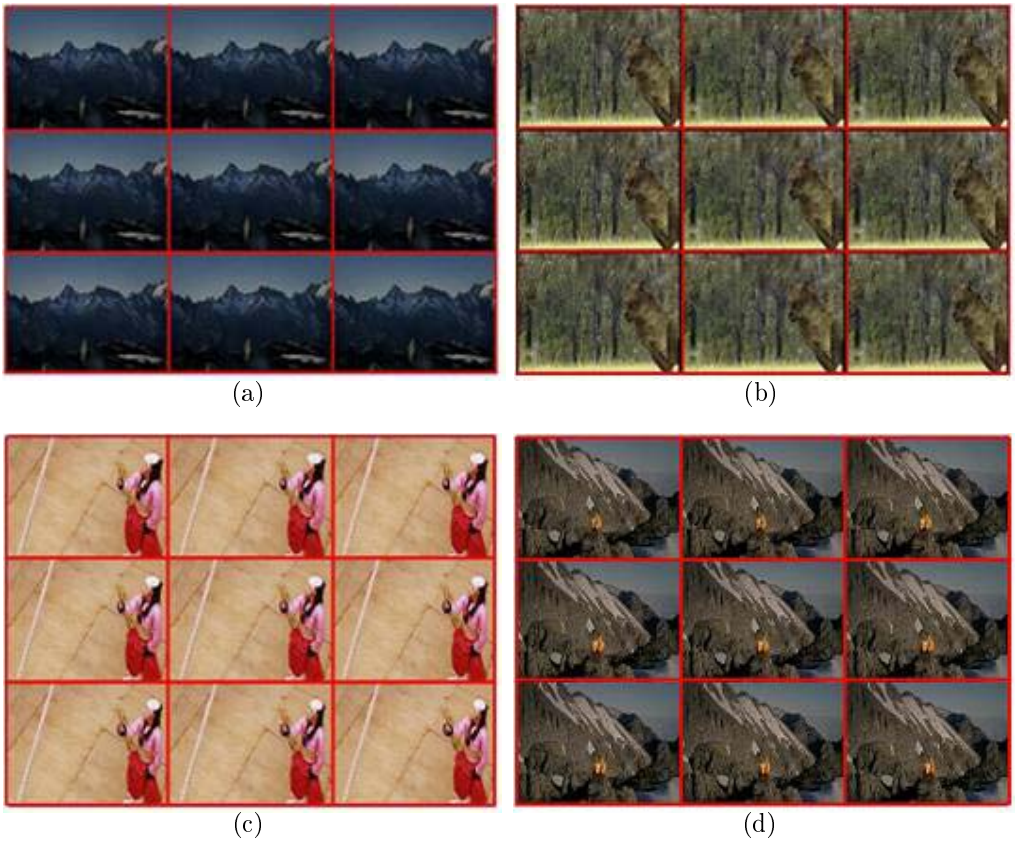
region-based depth maps for 4 frames of the depth image of Figure 9(a) are shown in Figure 10. Note that the depth values in the figures were adjusted for more configurability.

Each depth image in Figure 10 shows some noise information. Especially, Figure 10(c) shows discontinuity due to the change in the motion of the camera or the object, or due to an error in the motion estimation. To avoid this problem, three depth images are averaged to get the final depth map and the resulting examples are shown in Figure 11, where Figure 11(a) is the result from averaging Figures 10(a)-10(c), while Figure 11(b)



(a) (b)

FIGURE 11. Final depth maps: (a) 2nd frame and (b) 3rd frame



(a) (b)

(c) (d)

FIGURE 12. Examples of the generated nine-view images ($\theta = (-4°, 0°, +4°)$) in both horizontal and vertical directions: (a) sky, (b) woods, (c) traditional dancer and (d) rocks

is the average of Figures 10(b)-10(d). As can be seen in the figures, the two consecutive images have quite consistent depth profiles.

Finally, we rendered the multi-view images with corresponding depth map and the original image by calculating the rotated coordinates and the linear interpolation method. The resulting four examples are shown in Figure 12, which are the results of the angles, –4, 0 and +4 degrees in both horizontal and vertical directions.

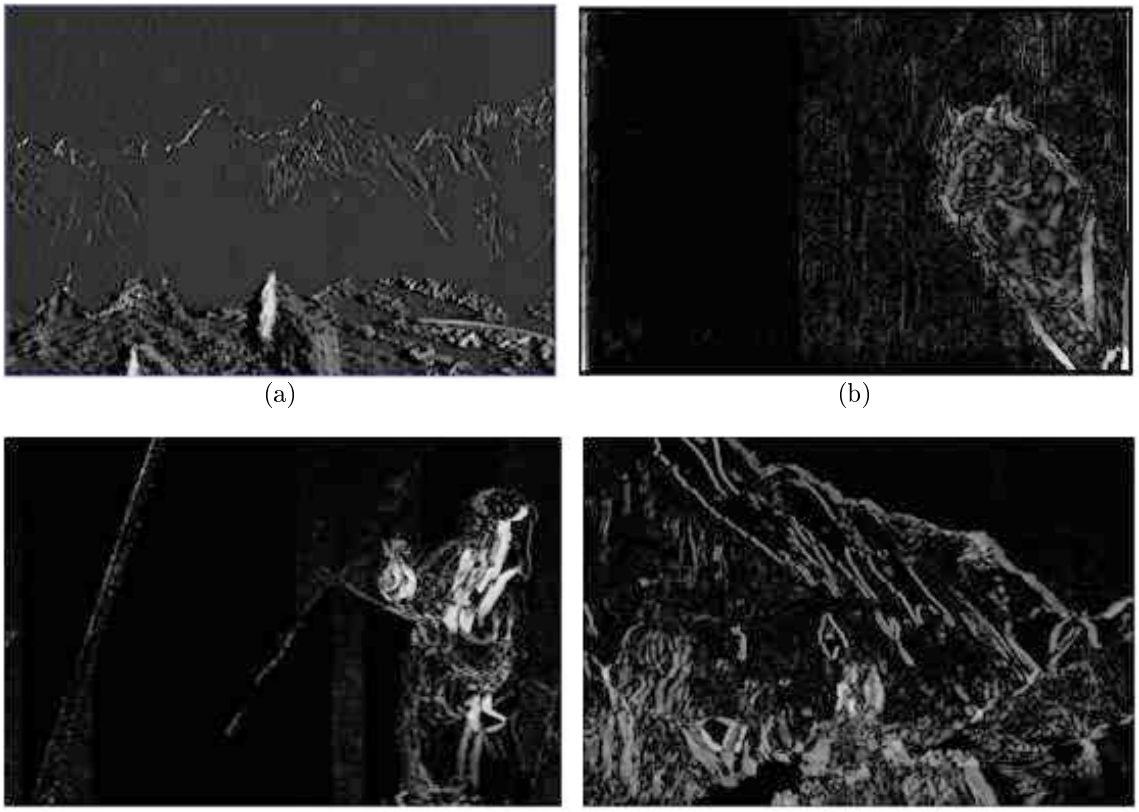


FIGURE 13. Difference image between −4-degree image and +4-degree image; (a) sky, (b) woods, (c) traditional dancer and (d) rocks

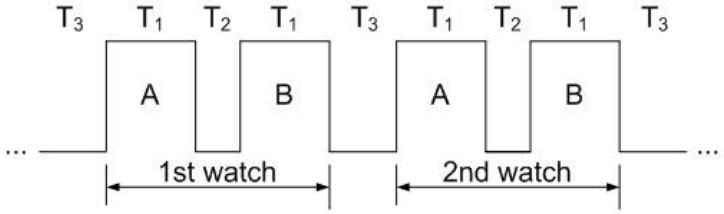


FIGURE 14. Subjective assessment of DSCQS

Figure 13 shows the difference image between −4 degree image and +4 degree image for each of the example images in Figure 12. As can be seen in the figures, it is clear that the amount of displacement is bigger as an object resides nearer, which is the desired result.

To make sure of the performance of our method, we have compared it with an existing method [11] by a subjective test called the Double Stimulus Continuous Quality Scale (DSCQS) [19] as shown in Figure 14. We made each viewer watch each video twice, alternatively watching video A or video B. The assessment was done during the 2^{nd} watch. As can be seen in the figure, each video was viewed during T_1. T_2 was the break time between Video A and video B, while T_3 was the break time between the two watches. T_1, T_2 and T_3 were 6[sec], 2[sec] and 5[sec], respectively. The watching order of the videos in each of the two watches was randomly selected. 15 persons assessed and each video had 120 frames. For each of the four videos in Figure 13, 0 degree and +1 degree videos were used in assessment.

Table 1. Assessment result comparison

Previous method [11]	Proposed method
Pair (58)	Good (71)

Assessment was carried out with 5 levels of excellent, good, fair, poor and bad. The assessment results were converted into the scale of 0 to 100 and averaged as shown in Table 1. The proposed method revealed much higher scores in the assessment, which we think is because method [11] is quite good for horizontal movements but poor for other movements.

5. **Conclusion and Further Research.** Although 3D video acquisition and processing technology has not yet become popular, it has been gathering attractive attentions in its diverse applications in communication, broadcasting, medical service [20], education, military training and gaming. The method proposed in this paper to generate multi-view video from 2D video can be considered a method to acquire 3D video with relatively low cost.

In this study, region segmentation and motion estimation were used to generate the multi-view video by using the fact that motion information could be converted to depth information. In addition, by using the depth values of the temporally neighboring images to the current image, it was possible to solve the problem of severe change in the depth information between two consecutive frames, which might be caused in the process of converting single-view motion information into corresponding depth information.

The proposed method was compared with a previous method by performing a subjective assessment. The results indicated that the proposed shows much better quality in 3D multi-view effects. It also shows that the proposed method appropriately considers motions in all directions.

The 3D video extraction from single-view video is dependent mainly on the estimated depth information. In order to obtain more exact depth information from a single-view video, it is necessary to study some methods to use other clues than the motion information. As the region segmentation method used for this study is sensitive to homogeneity and rigidity of image color, a method that is strong against them still needs to be studied. Also, more studies to process occlusion regions resulting from generating virtual viewpoints should also be done.

Acknowledgment. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0006791).

REFERENCES

[1] R. Franich, R. Lagendijk and R. Horst, Reference model for hardware demonstrator implementation, *RACE DISTIMA Deliverable 45/TUD/IT/DS/B/003/bl*, 1992.

[2] A. Redert et al., ATTEST: Advanced three-dimensional television system technologies, *3D Data Processing Visualization and Transmission, the 1st International Symposium*, pp.313-319, 2002.

[3] *3DTV NoE*, http://www.3dtv-research.org/.

[4] Subjective test results for the CfP on multi-view video coding, *MPEG/ISO/IEC JTC1/SC29/WG11*, Bangkok, Thailand, 2006.

[5] C. Fehn, Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV, *Proc. of SPIE Stereoscopic Displays and Virtual Reality Systems XI*, San Jose, CA, USA, pp.93-104, 2004.

[6] S.-Y. Kim, S.-B. Lee, Y.-K. Kim and Y.-S. Ho, Generation of multi-view images using depth map decomposition and edge smoothing, *The Journal of the Korean Society of Broadcast Engineers*, vol.11, no.4, pp.471-482, 2006.

[7] K.-W. Seo, C.-S. Han and J.-S. Yoo, Pre-processing of depth map for multi-view stereo images synthesis, *The Journal of the Korean Society of Broadcast Engineers*, vol.11, no.1, pp.91-99, 2006.
[8] J. Ross, Stereopsis by binocular delay, *Nature*, vol.248, pp.354-364, 1974.
[9] M.-B. Kim and S.-H. Park, 2D/3D video conversion based on key frame, *HCI Conference*, 2002.
[10] H.-K. Hong, Y.-K. Baek, S.-H. Lee, D.-W. Kim and J.-S. Yoo, 3D conversion of 2D H.264 video, *Korea Institute of Communication and Sciences*, vol.31, no.12(C), pp.1208-1215, 2006.
[11] Y. Matsumoto, H. Terasaki, K. Sugimoto and T. Arakawa, Conversion system of monocular image sequence to stereo using motion parallax, *SPIE Photonic West*, vol.3012, pp.108-115, 1997.
[12] H. Zhu, T. Takeshita and S. Li, 2D pattern vs. surrounding 3D pattern for fisheye camera calibration, *International Journal of Innovative Computing, Information and Control*, vol.4, no.6, pp.1499-1506, 2008.
[13] S.-Y. Lee, J.-H. Heu, C.-S. Kim and S.-U. Lee, Object removal and inpainting in multi-view video sequences, *International Journal of Innovative Computing, Information and Control*, vol.6, no.3(B), pp.1241-1256, 2010.
[14] Z. Musa and J. Watada, Video tracking system: A survey, *ICIC Express Letters*, vol.2, no.1, pp.65-72, 2008.
[15] S. Cooray, N. O'Connor, S. Marlow, N. Murphy and T. Curran, Semi-automatic video object segmentation using recursive shortest spanning tree and binary partition tree, *Workshop on Image Analysis for Multimedia Interactive Services*, pp.16-17, 2001.
[16] Y. Wang, J. Ostermann and Y. Zhang, *Video Processing and Communication*, Prentice Hall, 2002.
[17] C.-H. Kim, H.-K. Lee and Y.-H. Ha, A stereo matching algorithm in pixelbased disparity space image, *Korea Institute of Communication and Sciences*, vol.29, no.6(C), pp.848-856, 2004.
[18] S.-H. Jang, C.-S. Han, J.-W. Bae and J.-S. Yoo, Real-time multiple stereo image synthesis using depth information, *Korea Institute of Communication and Sciences*, vol.30, no.4(C), pp.239-246, 2005.
[19] ITU-T, Methodology for subjective assessment of the quality of television picture, *ITU-T Recommendation BT. 500-11.*
[20] M.-H. Tsai, S.-F. Chiou and M.-S. Hwang, A progressive image transmission method for 2D-GE image based on context feature with different thresholds, *International Journal of Innovative Computing, Information and Control*, vol.5, no.2, pp.379-386, 2009.