Copyright ⓒ 2022 KSII

Video-based Stained Glass

Dongwann Kang[1], Taemin Lee[2], Yong-Hyeon Shin[1] and Sanghyun Seo[3*]
[1] Department of Computer Science and Engineering, Seoul National University of Science and Technology
232 Gongneung-ro, Nowon-gu, 01811 Seoul, Korea
[e-mail: dongwann@seoultech.ac.kr, yshin@seoultech.ac.kr]
[2] Department of Artificial Intelligience and Software, Kangwon National University
346 Chungang-ro, Samcheok-si, 25913 Kangwon-do, Korea
[e-mail: kevinlee@kangwon.ac.kr]
[3] School of Art and Technology, Chung-Ang University
4726, Seodong-daero, Daedeok-myeon, 17546 Anseong-si, Korea
[e-mail: sanghyun@cau.ac.kr]
[*]Corresponding author: Sanghyun Seo

Received April 15, 2022; revised June 5, 2022; accepted June 13, 2022;
published July 31, 2022

Abstract

This paper presents a method to generate stained-glass animation from video inputs. The method initially segments an input video volume into several regions considered as fragments of glass by mean-shift segmentation. However, the segmentation predominantly results in over-segmentation, causing several tiny segments in a highly textured area. In practice, assembling significantly tiny or large glass fragments is avoided to ensure architectural stability in stained glass manufacturing. Therefore, we use low-frequency components in the segmentation to prevent over-segmentation and subdivide segmented regions that are oversized. The subdividing must be coherent between adjacent frames to prevent temporal artefacts, such as flickering and the shower door effect. To temporally subdivide regions coherently, we obtain a panoramic image from the segmented regions in input frames, subdivide it using a weighted Voronoi diagram, and thereafter project the subdivided regions onto the input frames. To render stained glass fragment for each coherent region, we determine the optimal match glass fragment for the region from a dataset consisting of real stained-glass fragment images and transfer its color and texture to the region. Finally, applying lead came at the boundary of the regions in each frame yields temporally coherent stained-glass animation.

Keywords: Non-Photorealistic Rendering and Animation, Stylization, Temporal Coherence, Video Segmentation, Panoramic Image.

A preliminary version of this paper appeared in The International Conference on E-learning and Games (Edutainment) 2017, June 26-28, Bournemouth, UK. This version includes technical details and more results with discussions. This study was supported by the Research Program funded by the SeoulTech (Seoul National University of Science and Technology).

http://doi.org/10.3837/tiis.2022.07.012 **ISSN : 1976-7277**

1. Introduction

Stained glass is a decorative art created from colored glass and is predominantly applied in windows of churches. Although this type of art has been used for religious purposes throughout its thousand-year history, currently, it is widely used for interior decoration.

Stained glass consists of small fragments of glass that have been colored using metallic salts. The fragments are arranged to form some images or patterns, as shown in **Fig. 1**. Lead came is typically used to assemble the fragments. Because stained glass was traditionally used for large windows, its architectural stability for resisting wind and supporting weight is a major consideration in manufacturing. Consequently, neither significantly large nor tiny fragments of glass are used in stained glass artworks.

In computer graphics, several studies have focused on imitating or simulating several styles in arts including stained glass [1]. Several works have proposed methods for generating stained glass-like images from given input images [2, 3, 4, 5]. However, rendering stained glass animation methods for video input have not been proposed yet. In this study, we aim to generate an animation with stained glass style from video input.

Fig. 1. An example of real stained glass artwork.

To obtain video frames with stained glass style, we can employ previous studies for generating stained glass-like images. However, using these studies directly to obtain individual animation frames is not appropriate, because they focus on generating single images without considering the temporal coherence, which is the most significant problem in the stylized animation field. Therefore, in this study, we propose a method for converting given video frames to stained glass while maintaining the temporal coherence between rendered frames.

To address the problems of generating stained glass animation that maintain the temporal coherence between stained glass fragments in adjacent frames, we propose a method described as follows: a) We initially segment input video coherently by employing mean-shift video segmentation. To prevent over-segmentation, we apply video segmentation to low frequency components obtained by decomposing the input video. b) We subdivide segmented regions whose sizes are greater than the user-defined size to prevent the generation of significantly large glass fragments. The subdivided regions must be coherent. To achieve this, we generate a panoramic image from the segmented regions of the input video frames. Thereafter, we apply the weighted Voronoi diagram to the panoramic image to obtain the subdivisions. Projecting the regions to original frames yields coherently subdivided regions in each frame. c) Finally, we render stained glass fragment for each subdivided region. To transfer the visual characteristics of real stained glass examples to the regions, we employ example-based color and texture transfer methods used in previous studies.

The remainder of this paper is organized as follows. In Section 2, we present an overview of relevant studies. Thereafter, we present the details of our method for generating temporally coherent stained glass animation in Section 3. The implementation and results are discussed in Section 4. Finally, we conclude with a summary of our ideas and discuss future work in Section 5.

2. Related Work

In computer graphics, the first research that aimed at generating stained glass was presented by Mould [2]. He proposed a method for creating an image that consists of stained glass fragments by employing several image processing techniques. Mould initially subdivided an input image into several regions using image segmentation. Thereafter, the regions were smoothened using morphological operations, such as erosion and dilation, to obtain smooth region boundaries. Although small regions are eliminated in this process, further progressive erosion is repeatedly performed to subdivide large regions. Furthermore, to mimic the color of medieval stained glass, the average color of regions was modified by employing a limited palette that consisted of the color of tinctures traditionally used in stained glass. Finally, the lead came and irregular glass surfaces were generated using a displacement mapping. The idea of performing image segmentation to obtain regions considered as glass fragments has since been employed in several studies, including our present research.

Brooks proposed a method of stylizing input images using the visual appearance of stained glass work examples [3]. Herein, the input image is initially segmented. Thereafter, the segmented regions are transformed into stained glass fragments using color and texture transfer techniques. In the transformation process, for each region, the optimal match glass fragment is queried from a database of real stained glass images to be used to transfer its appearance, including color and texture, to the region. In a similar manner, Setlur and Wilkinson proposed a method of automatic stained glass stylization that synthesizes glass fragment texture using example images of real stained glass artworks [4]. An image retrieval based on color and texture was used to determine the optimal match glass fragment image for each segmented region. The optimal match image was enlarged using texture synthesis and used to replace its corresponding region in the input image. Similar to Brooks' research, normal mapping was used to represent the texture variation and lead came. The idea in these studies, utilizing real stained glass images, is also employed in our work to mimic real artwork.

To obtain more regular region sizes, Doyle and Mould proposed a region-based stained glass rendering method using simple linear iterative clustering [5]. They classified the edges in input images into two classes, important edges and unimportant edges. Thereafter, they re-segmented regions wherein the boundaries matched the unimportant edges to create regular regions. The idea of subdividing regions to obtain regular sized regions is used in our research. However, to achieve semi-regular size subdivision, we employ Voronoi diagrams, particularly focusing on temporally coherent subdivision.

The stained glass filter in Photoshop creates stained glass-like images consisting of regular glass fragments [6]. To obtain regularly subdivided regions, a Voronoi diagram is employed. Although this method is a simple method to create stained glass-like images, its region shape, which is predominantly hexagonal, is much more regular compared to the shape of glass fragments in real works. To prevent this, we propose a method for subdividing regions semi-regularly resulting in arbitrary region shape by employing weighted Voronoi diagrams.

Since the convolutional neural networks was employed in stylization, especially for the style transfer [7], many neural approaches have been proposed in recent years [8]. In the neural style transfer, neural network models are trained to transfer the style of one image (style image) into another one (content image). By utilizing this approach, a stained glass style can be transferred into the original image to generate a stained glass-like image. However, to generate an animation, the temporal coherence must be maintained. To achieve this, we focus on maintaining temporally coherent stained glass style in this paper.

We note that a preliminary version of this study was presented in [9]. Compared to [9], this study provides more technical details and several results with discussions.

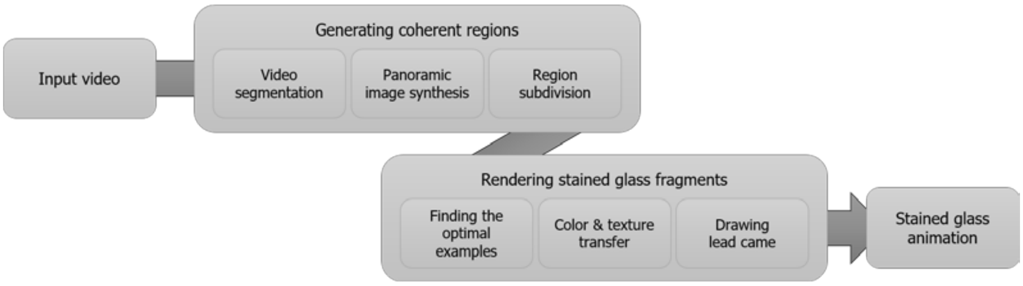

Input video
Generating coherent regions
Video segmentation
Panoramic image synthesis
Region subdivision
Rendering stained glass fragments
Finding the optimal examples
Color & texture transfer
Drawing lead came
Stained glass animation

Fig. 2. An overview of proposed system for generating coherent stained glass animation.

3. Proposed Methods

As shown in **Fig. 2**, the process for generating stained glass animation based on input video frames proposed in this paper mainly consists of two parts: generating regions considered as glass fragments (Section 3.1) and rendering them using real stained glass example images (Section 3.2)

3.1 Generating temporally coherent regions

The key element consisting of stained glass is glass fragment cut to represent the shape of the subject. As this study aims to produce stained glass animation that maintains temporal coherence from the video input, the form of glass fragments in the animation must be temporally coherent as well. To achieve this, we propose a method for generating coherent regions considered as glass fragments from video input in this section.

3.1.1 Segmenting video without over-segmentation

Similar to previous studies for rendering stained glass based on image [2, 3, 4, 5], we employ an image segmentation approach to obtain regions for glass fragments. However, the image segmentation predominantly results in over-segmentation, causing several tiny segments in a highly textured area. In practice, neither significantly tiny nor large glass fragments are used in stained glass manufacturing to ensure architectural stability. Therefore, we use low-frequency components in the segmentation to prevent over-segmentation.

Before performing segmentation, we separate the high-frequency components from the input video by using the image decomposition method [10]. The image decomposition finds local extrema from signals in the input image. Thereafter, it creates the envelopes using the extrema and captures the mean signal and oscillation of the envelopes, separating the texture from the individual edges. We decompose the frame $I_i \in I$ of the input video I into multiple scales as follows:

$$I_i(p) = \sum_{j=0}^{m} D_{i,j}(p) + M_{i,m}(p), \quad \forall p \in I_i \tag{1}$$

here, D_j is the j-th finest local oscillations, M_j is the mean, and p denotes each pixel of frame I_i. We obtain $M = \{M_{i,m} | i \in n\}$ by performing frame-wise image decomposition for the every input frames. As shown in **Fig. 3**, the texture potentially causing over-segmentation is eliminated in a low-frequency component image M. We then apply mean-shift segmentation [11] on M and obtain relatively large regions corresponding to glass fragments.

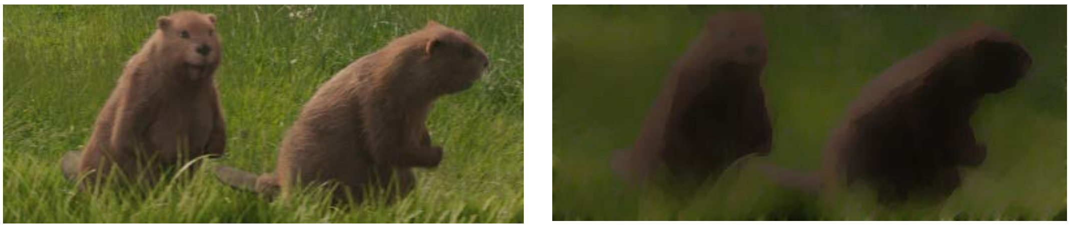

(a) Input video frame (b) Low-frequency components

Fig. 3. Decomposing input video frames to obtain low-frequency components

Mean-shift is a technique to seek local maxima of density in the feature space and widely used in image segmentation. If mean-shift image segmentation is frame-wisely applied to input video frames, incoherent regions are generated due to the lack of connectivity between adjacent frames. To prevent this, mean-shift video segmentation [12] performs the segmentation in the video volume domain yielding temporally coherent segments. In this paper, we obtain coherently segmented regions from M using mean-shift video segmentation, as shown in **Fig. 4**. We note that the segmented regions are temporally smooth, even though we frame-wisely decompose video frames without considering the temporal coherence.

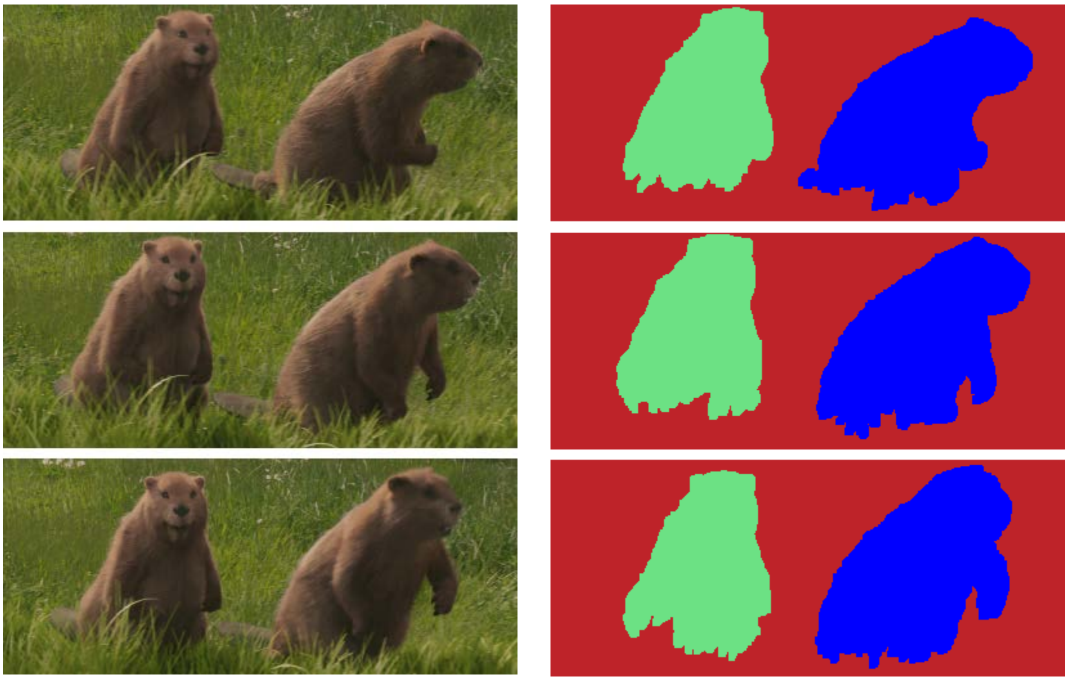

(a) Input video frames (b) Segmented regions

Fig. 4. Temporally coherent video segmentation using low-frequency components of video frames.

3.1.2 Coherently subdividing large regions

Although our segmentation approach prevents the over-segmentation, it could generate relatively large regions. In practice, large glass fragments are not used in stained glass manufacturing due to its heavy weight. Therefore, the large regions generated in the Section 3.1.1 need to be further subdivided. Similar to a recent region-based stained glass rendering study [5] in which large regions are iteratively subdivided to be regular-sized, we subdivide the large regions using Voronoi diagram. However, Voronoi diagram [13] generates a regular hexagonal pattern which is not observed in real stained glass works. To solve this, we employ the weighted Voronoi diagram [14] which yields arbitrary boundaries depending on the weight.

In a Voronoi diagram, the cell for i-th site is defined as

$$C_i = \{p \in I | d(p, s_i) \leq d(p, s_j), \forall j \neq i\} \tag{2}$$

where I, $d()$, and s_i denote an image plane, the Euclidean distance function, and i-th site, respectively. Generating Voronoi diagrams especially on rasterized plane can be accelerated using graphics hardware [15]. In the method, rendering three dimensional cones at Voronoi sites on an image plane rapidly produces an approximation of Voronoi diagram. Here, by using different slope for each cone, we can generate the weighted Voronoi diagram, in which the cell is defined as

$$\mathrm{C}_i' = \{p \in I | w_i d(p, s_i) \leq w_j d(p, s_j), \forall j \neq i\} \tag{3}$$

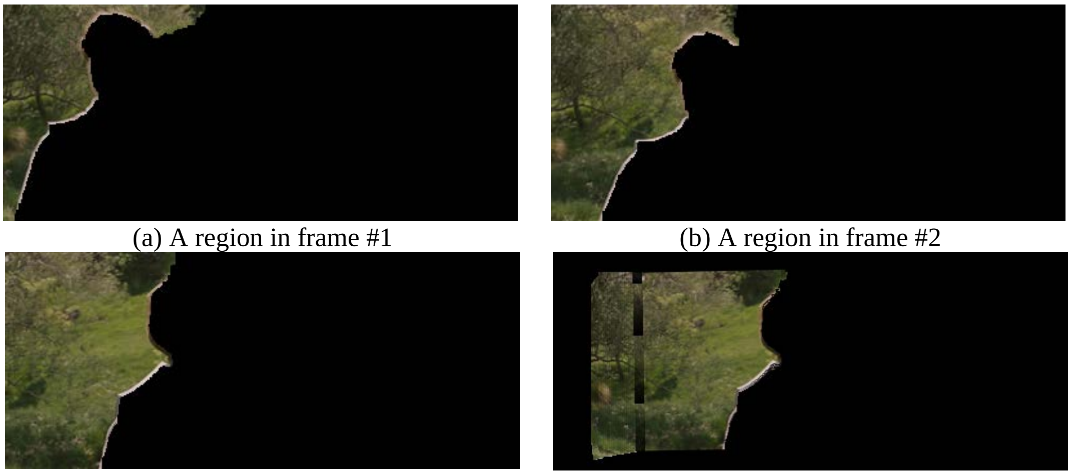
where w_i is the weight value of i-th site. In the weighted Voronoi diagram, its cells form relatively irregular shapes with curved boundaries that are closer to the shape of glass fragments observed in real stained glass works than the standard version of Voronoi diagram, as shown in **Fig. 6a**. In this paper, we assign random values to each site to obtain the weighted Voronoi diagram.

In our work, the subdivision must be performed while considering the temporal coherence between video frames to create coherent glass fragments in animation. If the regions are subdivided frame-wisely, the temporal coherence will be insufficient between the boundaries of subdivided regions between adjacent frames. Moreover, due to moving objects in input video, the regions corresponding to the objects that are being occluded or being appeared from other occluded object yield the difference between subdivided regions in adjacent frames. Therefore, to solve this problem, we take an approach that synthesizes a panoramic image by stitching each segmented region between frames and then subdivides it into several small regions if its size is bigger than threshold size.

To synthesize panoramic images, we extract the feature points between the regions in two adjacent frames by using the ASIFT(Affine Scale-Invariant Feature Transform) algorithm [16] which is a fully affine invariant feature detector. $F_{i,k}$, a set of feature point pairs on the k-th region R_k located between frame I_i and I_{i+1}:

$$F_{i,k} = \{(f_{i,k}^l, f_{i+1,k}^l | 1 \leq i \leq N-1, 1 \leq k \leq R\}. \tag{4}$$

Here, N is the number of input frames, R is the number of regions in input video volume, and l is the number of feature points on R_k between frame I_i and I_{i+1}. A feature point $f_{i,k}$ is in the frame I_i, and its corresponding point $f_{i,k+1}$ is in I_{i+1}. After extracting feature point pairs between every adjacent two frames, we deform frame I_{i+1} by employing moving least squares



(a) A region in frame #1

(b) A region in frame #2

(c) A region in frame #3

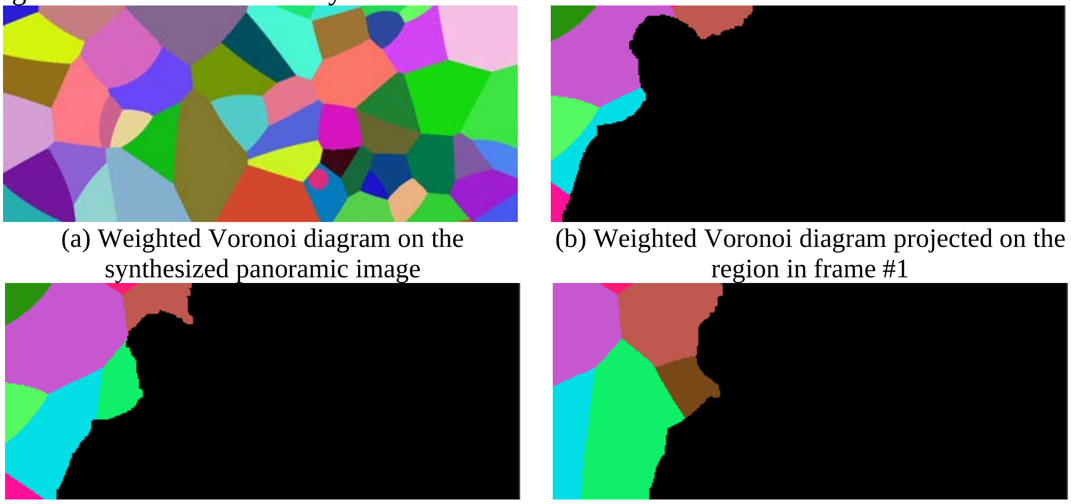
(d) Synthesized panoramic image

Fig. 5. Synthesizing a panoramic image from regions in input video frames.

[17] to obtain the deformed frame I'_{i+1}, which matches I_i:

$$I'_{i+1} = D(F_{i,k}, I_i) \tag{5}$$

Here, $D()$ is an image deformation function using feature pairs F. **Fig. 5** shows regions in several frames and the panoramic image synthesized. We thereafter subdivide the panoramic image's region of which size is bigger than threshold value (**Fig. 6a**). Finally, we project these regions on each frame to obtain coherent subdivision in frames. **Fig. 6b** - **Fig. 6d** show the regions subdivided coherently.



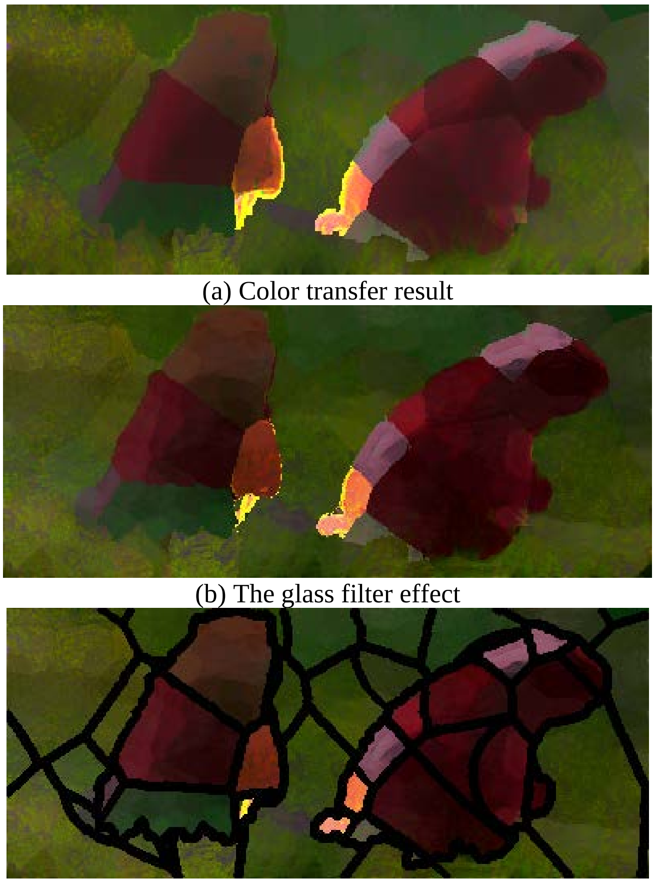
(a) Weighted Voronoi diagram on the synthesized panoramic image

(b) Weighted Voronoi diagram projected on the region in frame #1

(c) Weighted Voronoi diagram projected on the region in frame #2

(d) Weighted Voronoi diagram projected on the region in frame #3

Fig. 6. Obtaining temporally coherent subdivided regions using weighted Voronoi diagram.



(a) Color transfer result

(b) The glass filter effect

(c) Lead came effect

Fig. 7. Stained glass rendering results.

3.2 Stained Glass Rendering

To render regions obtained in Section 3.1 as stained glass, we find the optimal match glass fragment for each region from the dataset consisting of real stained glass fragment images (**Fig. 1**). In [3], the color and texture of the input image are used to find the optimal match of real glass fragment image which corresponds to each region. We employ this method for the same purpose, but we use only low-frequency components for the color characteristics. In addition, unlike [3] which treats a single input image, our study finds the optimal match with considering entire video frames.

Once the optimal match glass fragment image is found for each region, we then convert the color of each region into the color of the glass fragment which is found as the optimal match to mimic the appearance of real stained glass works. To achieve this, we employ Reinhard's color transfer method [18] which is broadly used in transferring colors between images. Figure **Fig. 7a** shows the color transfer result.

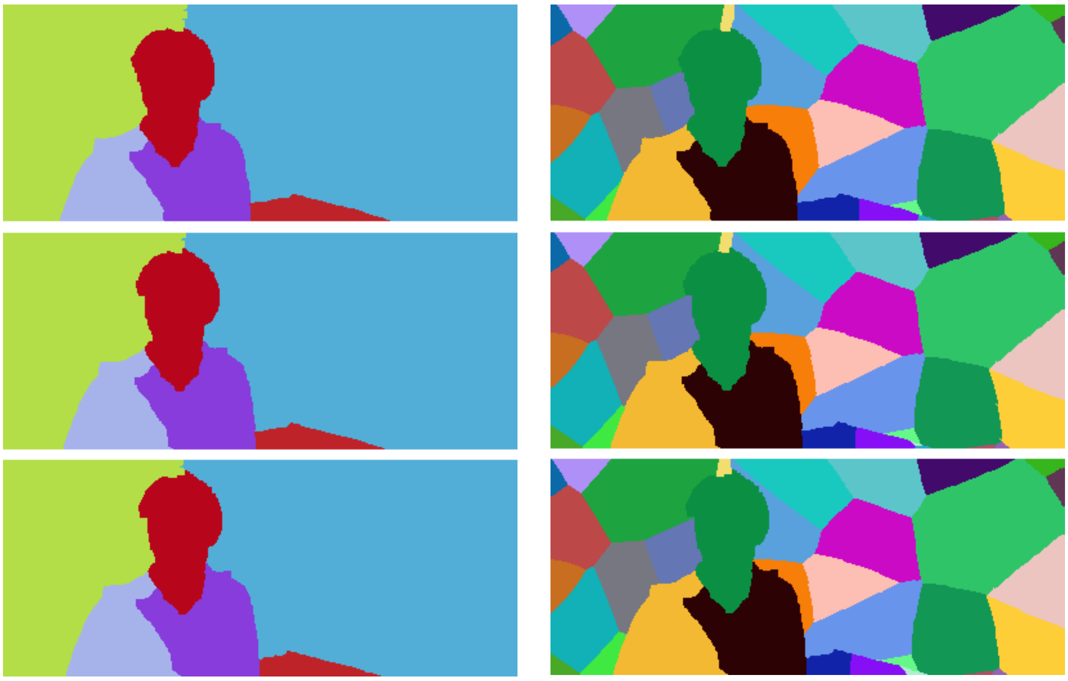
Even though we apply the color transfer, the result still requires the lighting effect which is one of key features that yield stained glass-like impressions. To give this effect to the result frames, we employ the glass filter introduced in [3] with adding Perlin noise [19]. **Fig. 7b** shows that the lighting effect generates many small facets of color in each region.

In real stained glass, lead came holds glass fragments together. To mimic this, we simply draw black borders with predefined width along the boundary of each region (**Fig. 7c**).

4. Results and Discussion

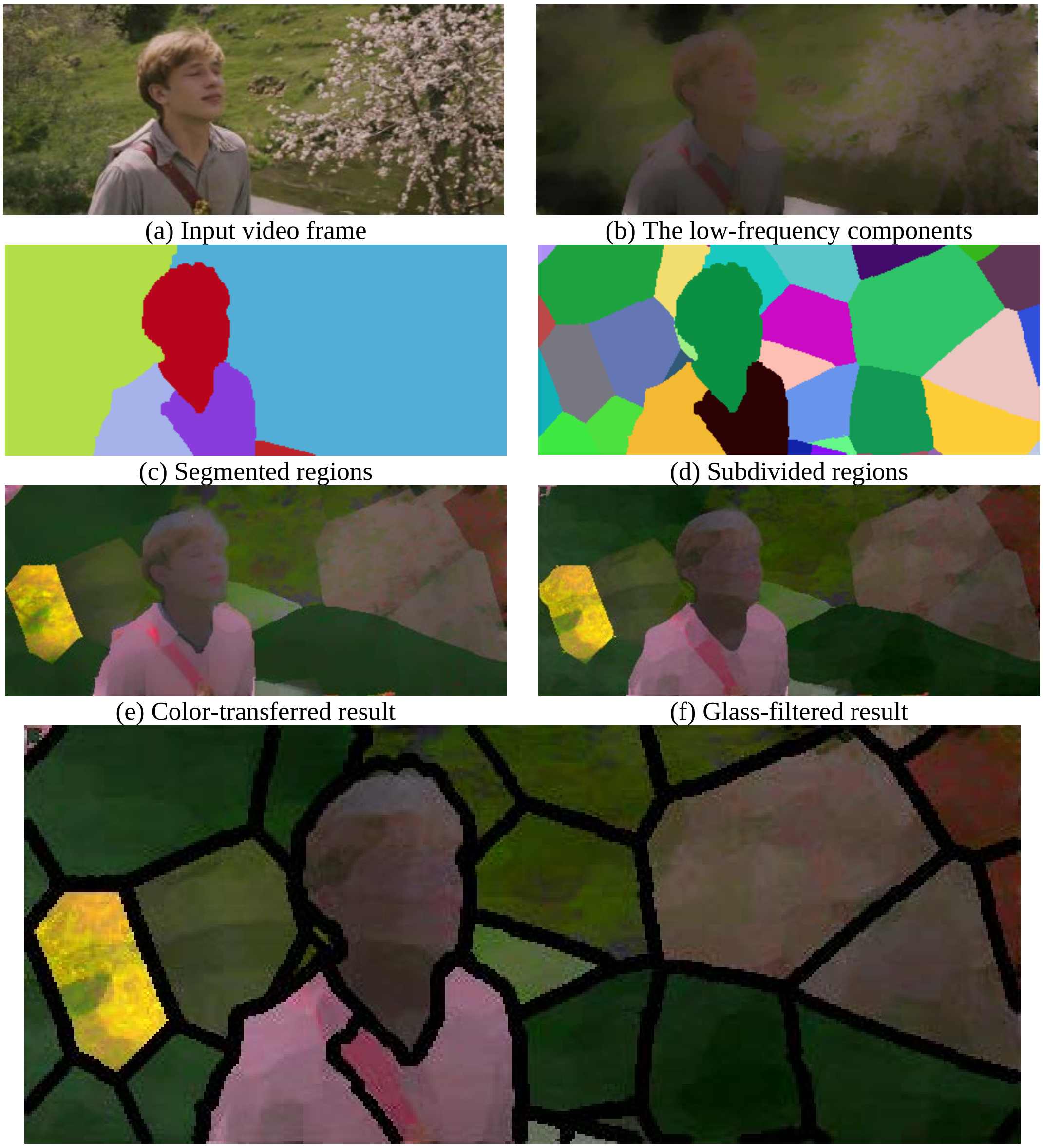
We experimented on various input videos to examine different results. Input videos were segmented with different parameters and the optimal result that is visually pleasing was chosen for our experiments. In our experiments, a system containing an i7-9700 (Central Processing Unit) with 16GB memory was used. In this experimental environment, obtaining subdivided regions described in Section 3.1 and generating stained glass effects explained in Section 3.2 took approximately 2-3 and 3-4 minutes respectively for a given video of which length was about 120 frames and resolution was 400×168 pixels.

Fig. 8a shows the result of segmentation which maintains the temporal coherence between regions in adjacent frames. The video segmentation did not yield desired regions considered as glass fragments of real stained glass work. To obtain desired regions, we performed the subdivision process using a weighted Voronoi diagram. **Fig. 8b** shows the result of the subdivision. In fact, subdivided regions performed on a panoramic image were projected on each frame. As shown in the figure, regarding the shape and location of regions, the temporal coherence between frames were significantly maintained. Moreover, the regions were effectively subdivided in terms of the architectural stability mentioned in Section 1.



(a) Video segmentation results (b) Region subdivision results

Fig. 8. Generating coherent regions considered as stained glass fragments.



(a) Input video frame

(b) The low-frequency components

(c) Segmented regions

(d) Subdivided regions

(e) Color-transferred result

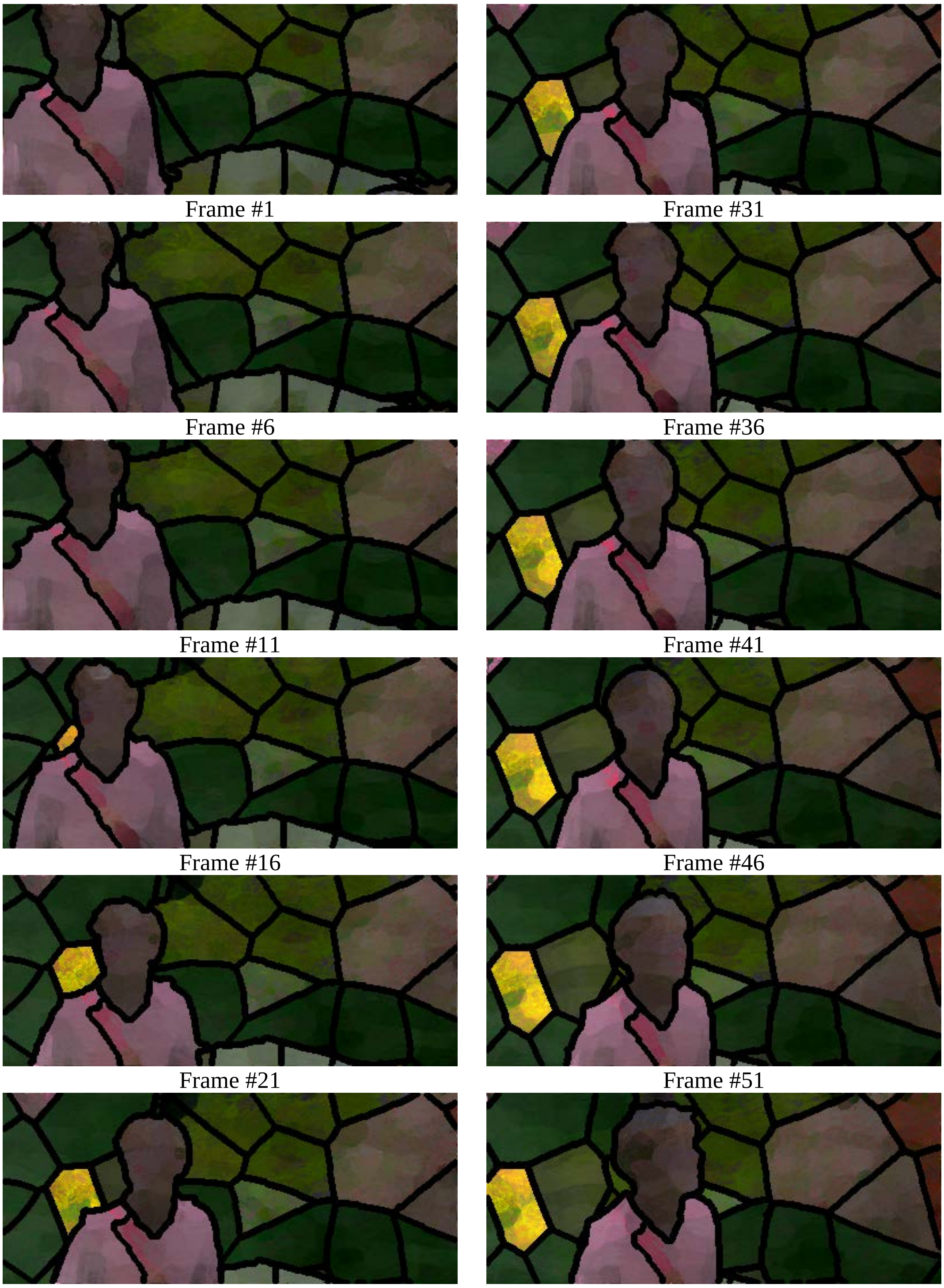
(f) Glass-filtered result

(g) Final result with lead came

Fig. 9. Stained glass rendering and animation process proposed in this study.

Fig. 9 shows the stained glass rendering process. We first obtained regions from the input video by mean-shift video segmentation. Then large regions were divided into smaller regions to yield the regions corresponding glass fragments in real art works. For each region, we then found the optimal stained glass fragment image from the dataset and transferred color and texture from the optimal image to the region. Finally, we generated lead came on the boundaries of each region.

Fig. 10 shows a stained glass animation result generated by the proposed method. As shown in the figure, glass fragments moved coherently along the motion while preserving overall structure. Fig. 11 shows the results on various input videos.



Frame #1

Frame #31

Frame #6

Frame #36

Frame #11

Frame #41

Frame #16

Frame #46

Frame #21

Frame #51

Frame #26

Frame #56

Fig. 10. Stained glass animation results.

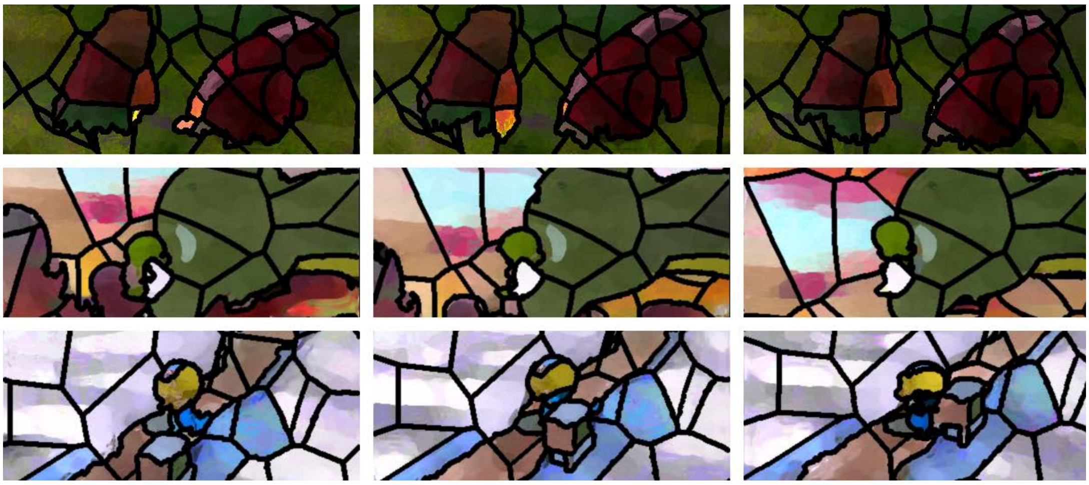


Fig. 11. Various stained glass animation results.

5. Conclusion

This paper presents a method for generating stained glass animation from a given video. To achieve this, we segment video frames into coherent regions by using mean-shift video segmentation. We then subdivide large regions into smaller regions. At this time, to obtain temporally coherent regions, we synthesize a panoramic image from regions in frames and subdivide it by using a weighted Voronoi diagram. To render these subdivided regions into stained glass fragments, we find the optimal match from the glass fragment image dataset and transfer its color and texture to the region. Finally, we obtain a stained glass-like frame by drawing lead came on the boundary of regions. The resulting frames are temporally coherent in terms of its style.

In this study, we utilize the low-frequency components of input video to omit the detail which disrupts video segmentation and color and texture transfer. However, this also causes the lack of important detail in results. In practice, the detail has been traditionally painted on the glass, as addressed in [5]. In future work, we will enhance the detail by painting the detail part through painterly rendering techniques.

Acknowledgement

This study was supported by the Research Program funded by the SeoulTech (Seoul National University of Science and Technology).

References

[1] J. E. Kyprianidis, J. Collomosse, T. Wang and T. Isenberg, "State of the "art" : A taxonomy of artistic stylization techniques for images and video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 5, pp. 866–885, 2013. Article (CrossRef Link)

[2] D. Mould, "A stained glass image filter," in *Proc. of the 14th Eurographics Workshop on Rendering (EGRW 03)*, Leuven, Belgium, pp. 20–25, June. 2003. Article (CrossRef Link)

[3] S. Brooks, “Image-based stained glass,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1547–1558, Sep. 2006. Article (CrossRef Link)

[4] V. Setlur and S. Wilkinson, “Automatic stained glass rendering,” in *Proc. of the 24th international conference on Advances in Computer Graphics (CGI 06)*, Hangzhou, China, pp. 682–691, June. 2006. Article (CrossRef Link)

[5] L. Doyle and D. Mould, “Painted stained glass,” in *Proc. of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering (Expressive 16)*, Lisbon, Portugal, pp. 1–10, May. 2016. Article (CrossRef Link)

[6] K. McCathran, *Adobe Photoshop CC: Learn by Video 1st edition*, SF,USA: Peachpit Press, 2013.

[7] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 2414-2423, 2016. Article (CrossRef Link)

[8] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, “Neural style transfer: A review,” *IEEE transactions on visualization and computer graphics*, vol. 26, no. 11, pp. 3365-3385, 2020. Article (CrossRef Link)

[9] D. Kang, D. Q. Vu and K. Yoon, “Generating stained glass animation,” *E-Learning and Games, LNCS*, Springer, vol.10345, pp. 228-232, Oct. 2017. Article (CrossRef Link)

[10] K. Subr, C.Soler and F. Durand, “Edge-preserving multiscale image decomposition based on local extrema,”*ACM Transaction on Graphics*, vol. 28, no. 5, pp. 1–9, Dec. 2009. Article (CrossRef Link)

[11] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002. Article (CrossRef Link)

[12] J. Wang, B. Thiesson, Y. Xu and M. Cohen, “Image and video segmentation by anisotropic kernel mean shift,” *Computer Vision (ECCV 2004), LNCS*, vol. 3002, pp. 238-249, 2004. Article (CrossRef Link)

[13] F. Aurenhammer, “Voronoi diagrams—a survey of a fundamental geometric data structure,” *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, Sep. 1991. Article (CrossRef Link)

[14] A. Secord, “Weighted voronoi stippling,” in *Proc. of the 2nd International Symposium on Non-Photorealistic Animation and Rendering (NPAR 02)*, Annecy, France, pp. 37–43, June. 2002. Article (CrossRef Link)

[15] K. E. Hoff, J. Keyser, M. Lin, D. Manocha and T. Culver, “Fast computation of generalized voronoi diagrams using graphics hardware,” in *Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 99)*, pp. 277–286, July. 1999. Article (CrossRef Link)

[16] J. -M. Morel and G. Yu, “Asift: A new framework for fully affine invariant image comparison,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, Jan. 2009. Article (CrossRef Link)

[17] S. Schaefer, T. McPhail and J. Warren, “Image deformation using moving least squares,” *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 533-540, July. 2006. Article (CrossRef Link)

[18] E. Reinhard, M. Adhikhmin, B. Gooch and P. Shirley, “Color transfer between images,” *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, Aug. 2001. Article (CrossRef Link)

[19] K. Perlin, “An image synthesizer,” *SIGGRAPH Computer Graphics*, vol. 19, no. 3, pp. 287–296, July. 1985. Article (CrossRef Link)

Dongwann Kang is an Assistant Professor in the Department of Computer Science and Engineering at Seoul National University of Science and Technology. He received his Ph.D. from Chung-Ang University in Korea in 2013, where he was a research fellow until Jun 2015. He was a lecturer of Undergraduate Interdisciplinary Program in Computational Sciences, Seoul National University, Korea (from Mar 2014 to Jun 2015); a lecturer at the Department of Multimedia, Sookmyung Women's University, Korea (from Mar 2014 to Dec 2014); and a visiting researcher (from Jul 2015 to Jan 2018) and a Marie Sklodowska-Curie fellow (from Feb 2018 to Aug 2018) at the Faculty of Science and Technology, Bournemouth University, UK. His research interests include non-photorealistic rendering and animation, emotional computing, image manipulation, and GPU processing.

Taemin Lee received his B.S., M.S. and Ph.D. degrees from Chung-Ang University, in 2011, 2013, and 2019, respectively. He is an Assistant Professor in the Department of Artificial Intelligence and Software at Kangwon National University. His research interests are in the area of painterly rendering, non-photorealistic rendering and emotion with paintings.

Yong-Hyeon Shin is an Associate Professor in the Department of Computer Science and Engineering at Seoul National University of Science and Technology. He received his Ph.D. degrees from Seoul National University. His research interests are in the area of web system, embedded operating system, storage system management.

Sanghyun Seo received his B.S. degrees in Computer Science and Engineering from Chung-Ang University, Seoul, Korea, in 1998 andM.S. and Ph.D. degrees in GSAIM Dep. at Chung-Ang University, Seoul, Korea, in 2000 and 2010. He was a senior researcher at G-Inno System from 2002 to 2005. He was the postdoctoral researcher at Chung-Ang University, in 2010, and the postdoctoral researcher at LIRIS Lab, Lyon 1 University from February 2011 to February 2013. He had worked at the ETRI (Electronics and Telecommunications Research Institute), Daejeon, Korea, May 2013 to February 2016 and at the Department of Media Software at Sungkyul University, Anyang-si, Korea, March 2016 to February 2019. Now, he is currently an associate professor in the College of Art & Science, Chung-Ang University. He has been a reviewer in Multimedia Tools and Applications (MTAP), Computer and Graphics UK (Elsevier), Journal of Supercomputing (JOS), Visual Computer (Springer), and Program Committee member in many international conferences and workshops and has edited a number of international journal special issues as a guest editor, such as Journal of Real-Time Image Processing and Journal of Internet Technology, and Multimedia Tools and Applications. He has been appointed as an Associate-Editor of the Journal of Real-Time Image Processing since 2017. His research interests are in the area of computer graphics, non-photorealistic rendering and animation, 3D GIS system, and real-time rendering using GPU, VR/AR, and game technology.