Video-based Stained Glass

Dongwann Kang

Seoul National University of Science and Technology

Taemin Lee

Chung-Ang University

Yong-Hyeon Shin

Seoul National University of Science and Technology

Sanghyun Seo (✉ sanghyun@cau.ac.kr)

Sungkyul University <https://orcid.org/0000-0002-4824-3517>

Research

Keywords: Non-Photorealistic Rendering, Stylization, Panoramic Image, Optical Flow

Posted Date: June 9th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-17285/v2>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

EURASIP Journal on Image and Video Processing

A Study on Generating Stained Glass Animation from Video Clip

--Manuscript Draft--

Manuscript Number:	JIVP-D-20-00052	
Full Title:	A Study on Generating Stained Glass Animation from Video Clip	
Article Type:	Research	
Funding Information:	National Research Foundation (NRF-2019R1F1A1064205)	Prof Dongwann Kang
Abstract:	<p>This paper presents a way to generate a stained glass animation from a given video input. We first obtain low-frequency components from input video frames to get rid of textures which cause over-segmentation in image segmentation. Then we segment input video volume by employing mean-shift video segmentation. The segmented regions are too large to be architecturally stable, subdivision is required. To sub-divide regions temporally coherent, we obtain the panoramic image from the segmented regions, and sub-divide them by using weighted Voronoi diagram. To render these sub-divided regions as stained glass pieces, we find the best match glass piece in the real stained glass piece image database and</p> <p>transfer its color to that region. Finally, we generate lead came at the boundary of regions, which results in a temporally coherent stained glass animation.</p>	
Corresponding Author:	Sanghyun Seo, Ph.D. Chung Ang University - Anseong Campus KOREA, REPUBLIC OF	
Corresponding Author E-Mail:	sanghyun@cau.ac.kr	
Corresponding Author Secondary Information:		
Corresponding Author's Institution:	Chung Ang University - Anseong Campus	
Corresponding Author's Secondary Institution:		
First Author:	Dongwann Kang	
First Author Secondary Information:		
Order of Authors:	Dongwann Kang	
	Taemin Lee	
	Yong-Hyeon Shin	
	Sanghyun Seo, Ph.D.	
Order of Authors Secondary Information:		
Suggested Reviewers:	Sangoh Park sopark@cau.ac.kr	
Opposed Reviewers:		
Additional Information:		
Question	Response	

RESEARCH

A Study on Generating Stained Glass Animation from Video Clip

Dongwann Kang¹, Taemin Lee², Yong-Hyeon Shin¹ and Sanghyun Seo^{3*}

*Correspondence:
sanghyun@cau.ac.kr
³College of Art and Technology,
Chung-Ang University, 4726,
Seodong-daero, Daedeok-myeon,
17546 Anseong-si, Korea
Full list of author information is
available at the end of the article

Abstract

This paper presents a way to generate a stained glass animation from a given video input. We first obtain low-frequency components from input video frames to get rid of textures which cause over-segmentation in image segmentation. Then we segment input video volume by employing mean-shift video segmentation. The segmented regions are too large to be architecturally stable, subdivision is required. To sub-divide regions temporally coherent, we obtain the panoramic image from the segmented regions, and sub-divide them by using weighted Voronoi diagram. To render these sub-divided regions as stained glass pieces, we find the best match glass piece in the real stained glass piece image database and transfer its color to that region. Finally, we generate lead came at the boundary of regions, which results in a temporally coherent stained glass animation.

Keywords: Non-Photorealistic Rendering; Stylization; Panoramic Image; Optical Flow

1 Introduction

Stained glass, which means that glasses are colored, has been produced since ancient times. Initially, stained glass was crafted for the need of building the high windows of the Gothic church. Through periods of changing architectural trends, stained glass not only became a prominent feature of the large church and the royal palace but also was the trend of interior decoration of the aristocracy and gradually more popular so far.

Stained glass is to connect all the pieces of glass with several shapes, sizes, colors, different nature to form a unified whole for a design idea. The product features art shows lively spirits of light and color.

Stained glass techniques were widespread around the world, especially Europe, America. However, the development of technology has changed somewhat traditional stained glass techniques. A few tools or tricks allow the shape to form products such as stained glass, but actually just a simulation. Besides the trend of industrialization of handcrafted items of high artistic, traditional stained glass techniques were extracted. The cause of the discrepancy caused in understanding of stained glass in some places where stained glass is still fresh.

After the start of studies related to Non-Photorealistic Rendering, this field now has conducted extensive research. However, there are few studies involving stained glass. Also, no study regarding stained glass animation has not been published. Therefore, in this study, we proposes a method involving generating stained glass animation in order to solve the difficulty in creating stained glass animation.

Before generating stained glass effect on each glass pieces, we need to obtain glass pieces from input video. Obtaining glass pieces in each frame/image are easy, but in video, we need to maintain the temporal coherence of glass pieces between all frames/images. In addition, generating stained glass effect also require to be temporally coherent animation. Both of them make a difficulty in creating stained glass animation and no study solving this issue has not been published yet.

In order to solve the difficulty in creating stained glass animation which maintain the temporal coherence of stained glass pieces between all frames/images, we propose a method described as follows: a) We first segment input video by proposing a video mean-shift segmentation using optical flow which estimates the motion of points in video. b) After getting segmented video from video segmentation step, we continue dividing all segments having big regions in frames into segments with smaller regions in frames. In order to do that, for each segment we need to divide, we first divide panoramic scene which computed from all layers in which represent this segment in all frames, we then obtain the divided region of each layers from the divided panoramic scene. c) Finally, we generate stained glass effect on each segment obtained from previous step. To do that, we extend image-stained glass rendering method to animation. Video decomposition is first applied to input video. And optical component is selectively applied to match the appearance of real stained glass.

We note that a preliminary version of this study was presented in [1].

2 Related Work

Many techniques model non-photorealistic modes, but there are few studies related to stained glass windows, especially no study involved stained glass animation is published.

The first model of stained glass technique using non-photorealistic rendering is a study for creating stained glass from 2D images by David Mould [2]. In this study, creating a glass pieces by using a segmentation technique based on the color of each area of input image. Then all regions are smoothed by performing the erosion and dilation operators from mathematical morphology. Also, big regions are subdivided and too small regions are eliminated. Next, the color of stained glass for each region is equal to the heraldic color which has the nearest distance from the average color of this region in input image. Finally, an image based stained glass is achieved after computing displacement map, representing the leading and imperfections in the glass surface.

Stephen Brooks [3] created image based stained glass from a real stained glass images database by proposing a method to generate stained glass. In this study, segmentation technique is also applied to input image and stained glass image. For each region in segmented image, find the best matching region in the stained glass image, then colorize this region to match the best matching region.

Stained glass filter [4] which has been used in Photoshop is based on Voronoi diagram to create glass pieces. Voronoi diagram is used to dividing a space into a number of regions. Voronoi sites are first created by random, then voronoi cell groups a set of points which have the shortest distance from the same voronoi site. This method is easily used, but glass pieces are created without considering the image content, so these glass pieces are not appropriate.

Figure 1 Decomposing input video frames for getting low-frequency components.

Figure 2 Temporally coherent video segmenting using low-frequency components of video frames.

3 Methods

The process that proposed to generate stained glass animation in this paper mainly consists of two parts: First, it creates the layout of glass pieces which are composed of stained glass from input video (Section 3.1), then it renders them by using real stained glass example images (Section 3.2).

3.1 Generating temporally coherent stained glass layout

The basic unit of stained glass is glass pieces fixed by lead cames. These glass pieces are cut to better represent the shape of the subject to be represented. As this study aims to produce stained glass animations that maintain temporal coherence from the input video, we should let the glass pieces, which are the basic units of stained glass, be cut and combined temporally coherently to form a stained glass layout. To achieve this, we describe the method to create a temporally coherent stained glass layout in this section.

3.1.1 Video segmentation that avoids over-segmentation

Previous studies [2, 3, 5] that produce stained glass from images use image segmentation methods to create the glass pieces. However, many tiny glass pieces that are generated due to over-segmentation may be produced in texture area, if regions obtained by segmenting input image is used as glass pieces. To avoid this, we propose a method to segment the input video using low frequency components only.

We separate the high-frequency components from the input video by using the image decomposition method [6]. Image decomposition is a technique that finds local extremas from signals in the input image, creates the envelopes by using the extremas, and captures the mean signal and oscillation of the envelopes, separating the texture from the individual edges. We employ it to decompose the frame $I_i \in I$ of the input video I into multiple scales as follows:

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

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 Figure 1, the texture which causes over-segmentation is eliminated in a low-frequency component image M . We then apply mean-shift segmentation [7] on M , and obtain large regions which correspond to glass pieces.

Mean-shift is a technique to seek local maxima of density in the feature space and widely used in image segmentation. If image-based mean-shift segmentation is frame-wisely applied to input video frames, incoherent regions are generated due to the lack of connectivity between frames. To prevent this, mean-shift video segmentation [8] performs segmentation in video volume domain, consequently produces temporally coherent segments. In this paper, we obtain segmented regions from M by using mean-shift video segmentation, as shown in Figure 2. At this time, even

though we frame-wisely decompose video frames without considering the connectivity between frames, it does not cause the temporal coherence problem, because video segmentation is applied to the low-frequency components of M .

3.1.2 Temporally coherent subdivision of large region

Although the segmented regions we obtain are not over-segmented, some regions are too large to be used as glass pieces. Generally, the size of each glass piece is not too large to be architecturally stable in real stained glass works. To achieve this, large segments obtained in Section 3.1.1 need to be proper-sized. A recent study for rendering stained glass [5] proposed an approach for re-segmenting such regions to obtain regular-size regions. Similarly, we sub-divide large segments as well. However, this subdivision must be performed while considering the temporal coherence between video frames to create coherent glass pieces in animation. If the regions are sub-divided frame-wisely, the temporal coherence will be insufficient between the boundaries of sub-divided regions between adjacent two frames. Moreover, due to moving objects, the regions that are being occluded and being appeared from occluded part are able to amplify the difference between sub-divided regions in frames. Therefore, to overcome this, we take an approach that sub-divides panoramic image which is generated by stitching regions in frames with considering motions.

We first obtain a panoramic image which is synthesized by stitching each segmented region between frames, we then sub-divide it into several small regions if its size is bigger than user-given predetermined size. To generate panoramic images, we extract the feature points between the regions in two adjacent frames by using the ASIFT algorithm [9] 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} :

$$F_{i,k} = \{(f_{i,k}^l, f_{i+1,k}^l) | 1 \leq i \leq N-1, 1 \leq k \leq R\}. \quad (2)$$

Figure 5 Sample images in our real stained glass image database.

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 $f_{i+1,k}$ is in I_{i+1} .

After obtaining feature point pairs between every adjacent two frames, we deform frame I_{i+1} to make it similar to I_i by employing moving least squares [10], and obtain the deformed frame I'_{i+1} :

$$I'_{i+1} = D(F_{i,k}, I_i) \quad (3)$$

Here, $D()$ is an image deformation function using feature pairs F . In our work, we do not need to determine the exact color of position of object in panoramic image from all relevant position of object in layers because our target is to segment panoramic image from objects having similar color. Figure 3 shows regions in frames and synthesized panoramic image.

We then sub-divide the panoramic regions of which the size is larger than pre-defined threshold value. To achieve this, we employ a multiplicative weighted voronoi diagram[11] which is defined when the distance between points is multiplied by positive weight. First, we randomly generate a number of Voronoi sites. At this time, every sites has its own weight. Then we group a set of points which have the shortest distance from the same Voronoi site. The difference of multiplicative weighted voronoi diagram from classic Voronoi diagram is to create voronoi cells with curve of edges which are similar to the shape of glass pieces in real stained works. Figure 4(a) shows an artistically sub-divided panoramic image generated by using multiplicative weighted Voronoi diagram. From the sub-divided panoramic scene, we obtain artistically sub-divided regions in each frame in which obtained region maintains the temporal coherence in entire video, as shown in Figure 4(b) - 4(d).

3.2 Stained Glass Rendering

Before rendering stained glass pieces obtained in Section 3.1, we find the best matching regions in the database containing real stained glass images (Figure 5).

Brooks [3] has considered all the color and texture of the image to find the best matching regions. In this study, we choose the color from low-frequency component of input video to find the best matching region. In [3], the methods for computing

(a) Color transfer

(b) The glass filter

(c) Lead came

Figure 6 Stained glass rendering.

color distance and texture distance are also used to find the best match, but we find the best match for entire video. Through finding the best match, real stained glass example images are used as a reference stained glass for transferring a region of video into stained glass style.

We then convert the color of each region into the color of the stained glass example which is retrieved as the best match to mimic the color of real stained glass works. To achieve this, we employ Reinhard's color transfer method [12] which is broadly used in transferring colors between images. Figure 6(a) shows the color transfer result.

Even though we applied color transfer from the region of real stained glass into corresponding region in base frames, the result still requires the lighting effect which is one of key features that make stained glass gorgeous. To give this effect to the result frames, we employ the glass filter introduced in [3] with adding Perlin noise [13] and generating many small facets of color in each region. Figure 6(b) shows the result after the lighting effect.

In real stained glass, there are lead came which holds glass pieces together. In this study, we mimic lead came by simply drawing black curves with pre-defined width along to the boundary of glass pieces (Figure 6(c)).

Parameters	Notes	Values
D_r	Spatial range thresholds	4
D_s	Color range threshold	8
t	Time threshold	5
M	The minimum number of pixels of a segment in each frame.	80

Figure 7 The value of parameters used in this paper.

Figure 11 shows the results of stained glass animation generated by proposed method. As shown in the figure, stained glass effects are coherently maintained in frames.

4 Results and Discussion

We have experimented on various input videos to examine different results. Input videos are first segmented with different parameters to compare the results for choosing a best result of each video that maintain the temporal coherence of segments. Also, extracting the motion flow by applying optical flows [14] is very important to estimate the movement of objects between frames in video and used to segment input video by applying mean-shift algorithm. The parameters for video segmentation using optical flow and their explanations are listed in Figure 7. In our experiment, we tested with various different parameters to compare the results and we chose D_r equal to 4, D_s equal to 8, t equal 5 and M is equal to 80. In addition, to divide segmented video by dividing panoramic frames, it takes about 2-3 minutes for a video with 5-seconds and the resolution of 400×168 pixels. Regarding to generating stained glass effect, we used various real stained glass sources to obtain the result as well as we can. In our implementation, we took about 3-4 minutes. Also, we have used a CPU of 4.1 GHz core i5-9400 CPU PC with 16GB memory. Moreover, the running time depends on the size of video.

Figure 8(a) shows the result of segmentation which maintain the temporal coherence between regions in continuous frames. The result from video segmentation does not bring about desired glass pieces that are nearly same with both the shape and size of real stained glass pieces. To obtain desired glass pieces, we performed sub-dividing operation to divide big regions and complex region into desired regions as glass pieces. Figure 8(b) shows the result after sub-dividing operation on these regions having the shape and size as original desire. Also, the result satisfies the requirement of temporal coherence of region by region between frame by frame.

Figure 9 shows the result of stained glass rendering process. We first obtained segments from the input video by mean-shift segmentation using optical flow. Then all big segments were divided to match the shape and size of real stained glass. Before generating stained glass effect on each segment, video decomposition was applied to input video to obtain base layers. Next, we performed a color transfer

from real stained glass database to base layers. Finally, we generated lead came after adding stained glass effect.

Figure 11 shows a stained glass animation from this study. The figure shows the movement of several regions (stained glass pieces) of video and the temporal coherence of regions is basically kept.

Figure 10 shows some frames of the resulting videos after generating stained glass animation from given videos.

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 segmentation. We then sub-divide large regions into smaller sub-regions. To obtain temporally coherent regions, we merge each region in frames to a panoramic image, and sub-divide it by using weighted Voronoi diagram. To render these sub-divided regions into stained glass pieces, we find the best matching glass piece from example stained glass piece database, and transfer its color to the region. Finally, we draw lead came on the boundary of regions, consequently obtain temporally coherent stained glass animation of input video.

In this study, we utilize the low-frequency components of input video to omit the detail which disrupts video segmentation and color transfer. However, this also causes the lack of important detail in results. In traditional stained glass, detail was usually painted on the glass directly. In future work, we will enhance the detail by adding detail part through painterly rendering techniques.

Abbreviations

ASIFT: Affine Scale-Invariant Feature Transform; CPU: Central Processing Unit; PC: Personal Computer

Figure 9 Stained glass rendering & animation process proposed in this study.

Acknowledgements

The authors would like to thank the editor and anonymous reviewers for their helpful comments.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1F1A1064205).

Availability of data and materials

The data used and/or analyzed during the current study are available from the authors on reasonable request.

Author's contributions

All authors take part in the discussion of the work described in this paper. Kang proposed the framework of this work and initiated the main algorithm of this work. Shin and Seo carried out the whole experiments and drafted the manuscript.

Authors' information

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

Figure 10 Various stained glass animation results.

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 Skłodowska-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. and M.S. degrees from Chung-Ang University, in 2011 and 2013, respectively. He is currently working toward his Doctor's degree in same 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 and M.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.

Competing interests

The authors declare that they have no competing interests.

Ethical Approval and Consent to participate

Not applicable

Consent for publication

We give our consent for information about our manuscript to be published in EURASIP Journal on Image and Video Processing.

Author details

¹Department of Computer Science and Engineering, Seoul National University of Science and Technology, 232 Gongneung-ro, Nowon-gu, 01811 Seoul, Korea. ²Department of Computer Science and Engineering, Chung-Ang University, 84 Heukseok-ro, Dongjak-gu, 06974 Seoul, Korea. ³College of Art and Technology, Chung-Ang University, 4726, Seodong-daero, Daedeok-myeon, 17546 Anseong-si, Korea.

References

- Kang, D., Vu, D.Q., Yoon, K.: Generating stained glass animation. In: Tian, F., Gatzidis, C., El Rhalibi, A., Tang, W., Charles, F. (eds.) E-Learning and Games, pp. 228–232. Springer, Cham (2017)
- Mould, D.: A stained glass image filter. In: Proceedings of the 14th Eurographics Workshop on Rendering. EGRW '03, pp. 20–25. Eurographics Association, Goslar, DEU (2003)
- Brooks, S.: Image-based stained glass. IEEE Transactions on Visualization and Computer Graphics **12**(6), 1547–1558 (2006). doi:10.1109/TVCG.2006.97
- McCathran, K.: Adobe Photoshop CC: Learn by Video, 1st edn. Peachpit Press, USA (2013)

5. Doyle, L., Mould, D.: Painted stained glass. In: Proceedings of the Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering. Expresive '16, pp. 1–10. Eurographics Association, Goslar, DEU (2016)
6. Subr, K., Soler, C., Durand, F.: Edge-preserving multiscale image decomposition based on local extrema. *ACM Trans. Graph.* **28**(5), 1–9 (2009). doi:10.1145/1618452.1618493
7. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5), 603–619 (2002). doi:10.1109/34.1000236
8. Wang, J., Thiesson, B., Xu, Y., Cohen, M.: Image and video segmentation by anisotropic kernel mean shift. In: Pajdla, T., Matas, J. (eds.) *Computer Vision - ECCV 2004*, pp. 238–249. Springer, Berlin, Heidelberg (2004)
9. Morel, J.-M., Yu, G.: Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences* **2**(2), 438–469 (2009). doi:10.1137/080732730. <https://doi.org/10.1137/080732730>
10. Schaefer, S., McPhail, T., Warren, J.: Image deformation using moving least squares. In: *ACM SIGGRAPH 2006 Papers. SIGGRAPH '06*, pp. 533–540. Association for Computing Machinery, New York, NY, USA (2006). doi:10.1145/1179352.1141920. <https://doi.org/10.1145/1179352.1141920>
11. Aurenhammer, F.: Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.* **23**(3), 345–405 (1991). doi:10.1145/116873.116880
12. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. *IEEE Computer Graphics and Applications* **21**(5), 34–41 (2001). doi:10.1109/38.946629
13. Perlin, K.: An image synthesizer. *SIGGRAPH Comput. Graph.* **19**(3), 287–296 (1985). doi:10.1145/325165.325247
14. Horn, B.K.P., Schunck, B.G.: Determining Optical Flow. In: Pearson, J.J. (ed.) *Techniques and Applications of Image Understanding*, vol. 0281, pp. 319–331. SPIE, ??? (1981). doi:10.1117/12.965761. International Society for Optics and Photonics. <https://doi.org/10.1117/12.965761>

Figure 11 Stained glass animation results.

Figures

(a) Input video frame

(b) Low-frequency components

Figure 1

Decomposing input video frames for getting low-frequency components.

(a) Input video frames

(b) Coherently segmented regions in adjacent frames

Figure 2

Temporally coherent video segmentating using low-frequency components of video frames.

(a) Region in frame #1

(b) Region in frame #2

(c) Region in frame #3

(d) Synthesized panoramic image

Figure 3

Temporally coherent video segmentating using low-frequency components of video frames.

(a) Weighted Voronoi diagram on the synthesized panoramic image

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

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

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

Figure 4

Temporally coherent regions sub-divided by using weighted Voronoi diagram.

Figure 5

Sample images in our real stained glass image database.

(a) Color transfer

(b) The glass filter

(c) Lead came

Figure 6

Stained glass rendering.

Parameters	Notes	Values
D_r	Spatial range thresholds	4
D_s	Color range threshold	8
t	Time threshold	5
M	The minimum number of pixels of a segment in each frame.	80

Figure 7

The value of parameters used in this paper.

Figure 8

Stained glass layout generation

(a) An input video frame

(b) The low-frequency components of input video frame

(c) Segmented regions

(d) Sub-divided regions

(e) Color-transferred result

(f) Glass-filtered result

(g) Final result with lead came

Figure 9

Stained glass rendering & animation process proposed in this study.

Figure 10

Various stained glass animation results.

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(k)

(l)

Figure 11

Stained glass animation results.