THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Master of Engineering (Computer Engineering)
DEGREE

Computer Engineering
FIELD

Computer Engineering
DEPARTMENT

TITLE: Region Substitution in Video Using Graph Cuts Technique

NAME: Mr. Ukrid Kuldiloke

THIS THESIS HAS BEEN ACCEPTED BY

_____ **THESIS ADVISOR**
(Associate Professor Punpiti Piamsa-nga, D.Sc.)

_____ **THESIS CO-ADVISOR**
(Associate Professor Yuen Poovarawan, M.Eng.)

_____ **DEPARTMENT HEAD**
(Assistant Professor Kemathat Vibhatavanij, Ph.D.)

APPROVED BY THE GRADUATE SCHOOL ON _____

_____ **DEAN**
(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

REGION SUBSTITUTION IN VIDEO USING GRAPH CUTS TECHNIQUE

UKRID KULDILOKE

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Engineering (Computer Engineering)
Graduate School, Kasetsart University
2009

Ukrid Kuldiloke 2009: Region Substitution in Video Using Graph Cuts Technique. Master of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Associate Professor Punpiti Piamsa-nga, D.Sc. 39 pages.

Frequently, a screen editor have to edit a video shot by cutting parts of a video clip and patching them onto a targeted video. Even though the cut-and-paste method is efficient for editing still images, it is not appropriate for the same purpose on editing video clips since the assigned patched area is fixed while contents continuously change. In this research, we propose an automatic video patching method that preserves continuousness at the borders of the patched area by determining appropriate cut masks of every frame using a graph cut algorithm. In our algorithm, it does the same thing as the cut-and-paste does except that the patched mask is automatically adapted to contents on new frames. We evaluate the quality of our results by generate five experimental samples and let twenty people mark the area which they believe the composition has occurred then we summarize the results such as recall, false positive and undetectable. The percentages are thirty-one, forty-two and twenty-seven respectively. We also compare our results with the same video generated from Sony Vegas, a video editing application from Sony. The percentages of recall, false positive and undetectable of Sony Vegas results are fifty-seven, thirty and thirteen respectively.

_____ _____ ___ / ___ / ___

Student's signature Thesis Advisor's signature

TABLE OF CONTENT

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

REGION SUBSTITUTION IN VIDEO USING GRAPH CUTS TECHNIQUE

INTRODUCTION

Have you ever wanted to replace some boring things in your video clip with a more interesting stuff from another clip? For example, in Fig. 1 we replace a soothing sky of the city with a stormy cloud from the ocean. Video composition is a technique to composite multiple video clips together. This technique can be divided into three parts, scene matching, video composition and preservation of continuousness.

The first process, scene matching, is a technique to find similarity between two video clips depending on selected features such as color and edge of the image. The second process is video composition. We have to composite two different video clips so that the color and illumination differences along the seam of the composited region should not be detected. Finally, we have to preserve the continuousness of the result video clip.

In this paper we propose an automatic method for video composition. Inputs of our system are two video clips (base and patch) and a mask image for specifying patched area and output is a patched video clip. The result of our algorithm has minimum discontinuities both along the border of composited region and between adjacent frames of the final video clip which we evaluate by let ten people specify the composited region in our result video clip and compute the ratio between correct and incorrect answer they provided.

The structure of this thesis is as follows. Literature review Section is about overview on related works. Three parts of our algorithm: scene matching, image composition and preservation of continuousness are in Methods Section. Experiment and its results are on Results Section. Finally, conclusion and discussion on limitation of our technique are in Conclusion and Discussion Section.

(a) Based clip: City clip (b) Patch clip: Cloudy sky (c) Result: Cloudy city

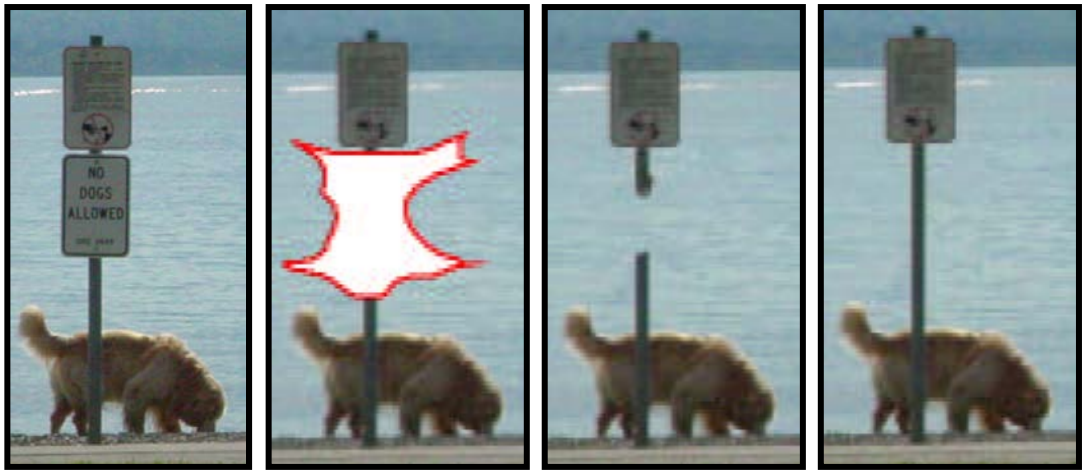
Figure 1 Bring a cloudy sky from (b) into a city in (a) and final result is shown in (c)

OBJECTIVES

To replace region in video clips with the least discontinuity along the seam of the composited region where also reserved continuousness of the result video clip after composition process

LITERATURE REVIEW

The fundamental process for replacing a region in video clip is image composition. There are many algorithms designed to accomplish image composition task. In 2003, Criminisi *et al*. proposed an algorithm to complete an image with masked out region by gradually growing the surrounding area of that region until the hole is filled. Fig. 1 shows Criminisi's method by removing a lower sign out and growing the surrounding region until the hole disappears. This method can perform really well if the masked out region is small because the patch that Criminisi's algorithm created is incomprehensible, based on image gradient, when the masked out region is large, an incomprehensible patch can lead to an incomprehensible result. Another algorithm was proposed by Wilczkowiak *et al*. (2005). They proposed an algorithm to complete image by finding a patch from the same image to fill in the hole. Fig. 2 shows Wilczkowiak's methods by masked out a walking man and then find an appropriate patch from within the same image to complete the missing region. This method has many benefits such as there are small differences in color and illumination along the seam of the masked region because the patch is obtained from within the same image. However, the main problem of this approach is that if there is no compatible patch within the same image or the masked out region is very large so that the remaining region cannot completely fill up the mission region.



(a) Original image (b) Masked out image (c) Grow surrounding region (d) Final image

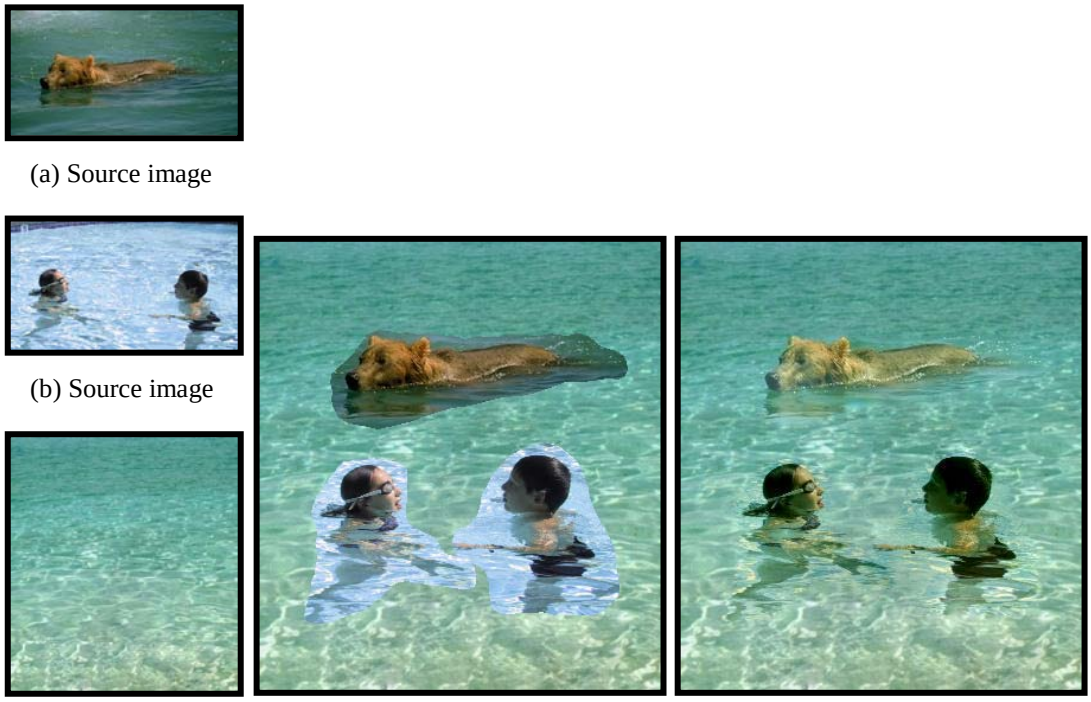
Figure 2 Result of Criminisi *et al*. Original image is in (a). The image after remove an unwanted region is in (b). In (c) shows the growing process of Criminisi. And the complete image is shown in (d).

The previous two methods use the data residing on the same image to complete the missing region. However we want not only to fill in the missing region but replace it with something new. Pérez *et al*. (2003) proposed a mathematical model based on solving Poisson equation to edit various types of images. One of them is

seamless cloning which can cut object from source image and paste to destination image with unnoticeable seam. Fig. 3 shows that Pérez's method can blend both color and illumination of pasted region to be like base image even though the pasted regions have come from many other images with different color and illumination. Jia *et al*. (2006) implements a user friendly system called "Drag-and-drop pasting" which based on Pérez's Poisson equation and they add Graph cuts technique (Boykov *et al*, 2001) to find the most appropriate region to perform Poisson blending. Fig. 4 demonstrates the need of adding Graph cuts technique to improve the quality of seamless cloning based on Poisson blending operation. If only Poisson blending is used, the color and illumination of the whole pasted patch will be blended into the background. So it will leave a blurring effect around the edge of the pasted patch. As shown in Fig. 4(c), the rock on the right and the water on the top of the log is blurred out. After computing Graph cuts to find optimal boundary (Fig. 4(d)) the result of Poisson blending is improved as shown in Fig. 4(e) the rock and the water is no longer blurred. In 2007, Hays and Efros proposed a system which can fill in the missing region by finding an appropriate patch from millions of photographs. They used the concept of scene descriptor from Oliva and Torralba (2006) to find twenty images with the most similar scene match and perform the composition by using Graph cuts and Poisson blending. They used Graph cuts (Boykov *et al*, 2001) to find the most suitable region for the composition process then using Poisson blending to reduce the discontinuities along the seam of the composited region. Fig. 5 illustrates the overview of Hays and Efros system. Fig. 5(a) is the original image, Fig. 5(b) is the image after remove an unwanted region, Fig. 5(c) is the top twenty most similar scene matches and Fig. 5(d) is the output image after compute Graph cuts along with Poisson blending.

(a) Matched region to complete image (b) Image after completion process

Figure 3 Result of Wilczkowiak *et al*. (a) shows the masked out region, a black region with a shape of a walking man on the lower right, and a boundary to be used as a completion regions are shown in red and green. The result image is shown in (b).



(a) Source image

(b) Source image

(c) Destination image

(d) Simple cut-and-paste

(e) Image after Poisson blending

Figure 4 Result of Pérez *et al*. (a) and (b) are source images which contain region to be cut. (c) is destination image, the image in which the region from source image will be pasted. The result of simple cut-and-paste is in (d) and the refinement of (d) with Poisson blending is in (e).

Our algorithm adapts Hays and Efros's technique (2007) to apply on video clip, as a sequence of images. Our primary goal is to make the patched video clip have minimal visible discontinuities along the seam of the composited region as well as preserve continuousness of the video clip. We divide our technique into three parts: scene matching to find the most suitable position in patch video clip to composite each frame together, image composition to composite two frames from base and patch video clip together with least visible seam along the composited region and preservation of continuousness to insure the smoothness transition between adjacent frames.

(a) Source image

(b) User boundary (c) Poisson blending with user boundary (d) Optimum boundary (e) Poisson blending with optimum boundary

Figure 5 Result of Jia *et al*. (a) shows source image which contain desire region to cut. (b) shows user drawn boundary for cut-and-paste. (c) shows the result of Poisson blending with user boundary. In (c), the region around the log is blurred by the effect of Poisson blending. After compute graph cuts, the optimum seam is shown in (d). And the result of Poisson blending with optimum seam from graph cuts is shown in (e). The blurring artifact is gone.

(a) Original image (b) Input image (c) Scene matches (d) Output image

Figure 6 Result of Hays and Efros. (a) shows original image. The image with masked out region and will be the input to the system is shown in (b). (c) shows the twenty most similar scene matches. The result of the system is shown in (d).

MATERIALS AND METHODS

Materials

1. Computer notebook
    1) CPU Dual Core 1.6 GHz
    2) RAM 2 GB
    3) Harddisk SATA 80 GB, 5400 rpm
2. Software
    1) Microsoft Windows XP
    2) Microsoft Visual Studio 2005
    3) Microsoft Word
    4) Intel OpenCV Library
    5) Taucs, A Library of Sparse Linear Solvers
    6) Markov Random Field Library
    7) Poisson Image Editing Library (based on Taucs)
    8) Sony Vegas

Methods

1. Scene Matching

We compute scene descriptors on every frame in both video clips to find the most appropriate position in the patch video to be composited. Our descriptor is composed of two main features: color and edge of unmasked areas. We divide an image into several small pieces in grid style as illustrated in Fig. 7. This method makes our descriptors to better describe the image in local context not just global one. Each small image is converted into a Laplacian image and an L*a*b* color image.

We used L*a*b* color space because this color space is designed to approximate human vision (Anonymous, n.d.).

First we compute an image descriptor of every frame in both video clips by divide an image into several parts (Fig. 7). Each part is separated into two images, Laplacian image and L*a*b* color image (Fig. 8), and then compute an average value of these two images separately.

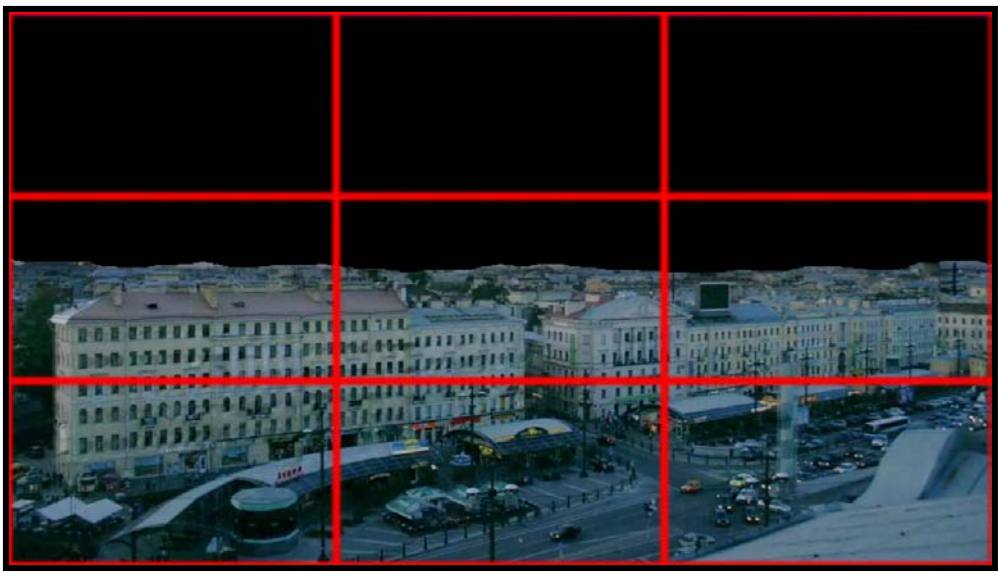
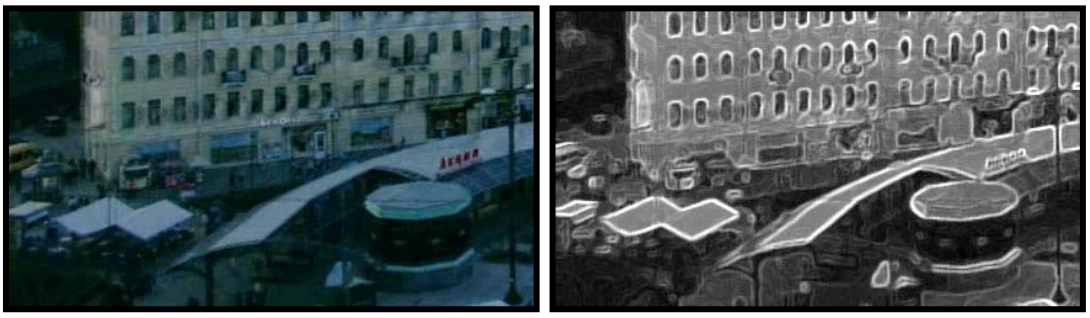


Figure 7 Divided image to compute descriptor.



(a) L*a*b* color image. (b) Laplacian image

Figure 8 Images for compute scene descriptor. Color component and Edge component of scene descriptor can be computed from (a) and (b) respectively.

After we have computed average value of all the small parts divided earlier, we then pack them up in a single vector which we call “scene descriptor”. A pseudo code for computing scene descriptor is shown in Fig. 9.

Since we have obtained scene descriptor, we then compute the descriptor of all frames in video clip and append them into a single sequence which we call “video descriptor”. A pseudo code for computing video descriptor is shown in Fig. 10.

Fig. 11 is a pseudo code for comparing two scene descriptors. Since scene descriptor compose of two vectors, vector of edge value from Laplacian image and vector of color value from L*a*b* color image, therefore we can compare two scene descriptors by finding a Euclidian distance between edge and color component of

those descriptors then compute an average value of edge and color distance computed earlier.

The computation of comparing two video descriptors is shown in Fig. 12. We compare descriptor of base video clip and descriptor of patch video clip with the same length by using scene comparing algorithm shown in Fig. 9 to compare in frame by frame basis then find an average value from the comparing results of every frame in the comparing sequence. Then sequentially compare base video descriptor with the next frame in patch video descriptor as a new starting point for comparing. We use dynamic programming technique to speed up the sequential comparing process by compute scene descriptor of every frame in both video clips before starting the comparing process. When we have all the compare results then the least value is the most suitable position to start the image compositing process. The illustration of matching base video clips and portion of patch video clips for computing image compositing process is shown in Fig. 13.

```
initialize vector C to zero
initialize vector E to zero
set tuple D to be empty
for i=1 to grid_row do:
    for j=1 to grid_column do:
        convert image in grid(i,j) to L*a*b* color space
        append average value of L*a*b* image into vector C
        compute laplacian image of image in grid(i,j)
        append average value of laplacian image into vector E
set tuple D as tuple(C,E)
return tuple D
```

Figure 9 Pseudo code to compute image descriptor. For each sub image in divided source image, we generate color image in L*a*b* color space and Laplacian image then append average value of them into sequence *C* and *E* respectively. Finally merge those two sequences into a tuple *D* then return tuple *D* as image descriptor.

```
let V be the input video clip
set sequence V_D to be empty
for i=1 to length(V) do:
    compute image descriptor of the i^th frame in V
    append the computed descriptor to sequence V_D
return sequence V_D
```

Figure 10 Pseudo code to compute video descriptor. Compute image descriptor for each frame in video clip *V* and append them to sequence V_D then return sequence V_D as video descriptor.

let D_B be image descriptors of base image
let D_P be image descriptors of patch image
compute distance between vector $D_B[C]$ and vector $D_P[C]$
assign the computed distance to A_C
compute distance between vector $D_B[E]$ and vector $D_P[E]$
assign the computed distance to A_E
set compare result to average value of A_C and A_E
return compare result

Figure 11 Pseudo code to compare image descriptors. Compute distance A_C between $D_B[C]$ and $D_P[C]$. Compute distance A_E between $D_B[E]$ and $D_P[E]$. The compare result is average value of A_C and A_E.

let VD_B be descriptor of based video
let VD_P be descriptor of patched video
set sequence *R* to be empty
for *i*=1 **to** length(VD_P) – length(VD_B) **do**:
    set sequence *CR* to be empty
    for *j*=1 **to** length(VD_B) **do**:
        set D_B as $VD_B[j]$
        set D_P as $VD_P[$length(VD_B)*i+$j]$
        compare descriptor D_B and D_P
        append compare result to sequence *CR*
    append sequence *R* with average value of sequence *CR*
find minimum value in sequence *R*
return position of the minimum value in sequence *R*

Figure 12 Pseudo code to compare video descriptor. Sequentially extract patched video descriptor with the same length as based video descriptor, compare them and find the position of the minimum comparison value.

2. Image Compositing

In compositing process, we divide it into two stages, Graph cuts and Poisson blending. Since image compositing can be mapped into a node labeling problem which maximum label is two (source and destination image) (Szeliski et al, 2006); therefore Markov random fields and Graph cuts can be used to find the seam which has minimum differences between two images. Such computation can be done by minimizing the energy functions:

$$E = E_D + E_S \quad (1)$$

where E, the energy targeted to be minimized. E_D and E_S are the energy derived from image data and smoothness of the result image, respectively.

Figure 13 Scene match in video sequence.

Data energy, E_D, can be computed by Eq. 2, by calculating absolute different color between pixels from source and destination image. We used RGB color space in this equation because this color space affects the computer display.

$$E_D = Abs\left(C\left(p^s_{(i,j)}\right) - C\left(p^t_{(i,j)}\right)\right) \tag{2}$$

The computation of smoothness energy depends on which label, *S* or *T*, the pixel *p* be labeled. If pixel *p* be labeled as *S* (source image) that means *p* will be in the result image, the energy (E_S^S) is an absolute value of the color difference between pixel *p(i,j)* from source and destination image multiply by the inverted value of pixel *p(i,j)* in Laplacian image as in Eq. 3. On the other hand if the pixel *p* was labeled as *T* (destination image) which means pixel *p* will not be included in the result image, the energy (E_S^T) is the inverse value of E_S^S as shown in Eq. 4. In this equation we use L*a*b* color space because we want to make the resulting image look smooth in term of human vision. We multiply the absolute different value of color with inverse of the Laplacian to lower the energy along the edge. We then pass these energy functions to Markov random fields and use Graph cuts to minimize them.

$$E_S^S = Abs\left(C_l\left(p^s_{(i,j)}\right) - C_l\left(p^t_{(i,j)}\right)\right) * \left(-L\left(p^l_{(i,j)}\right)\right) \tag{3}$$

$$E_S^T = -E_S^S \tag{4}$$

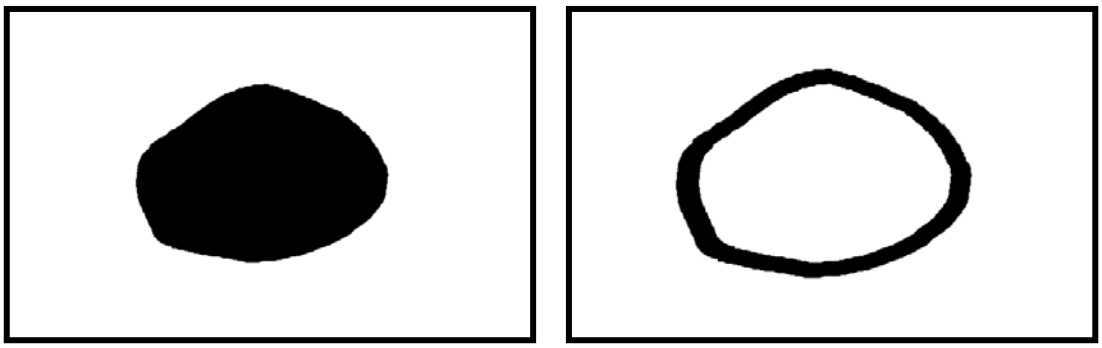
The second stage of our compositing process is Poisson blending which can be used to blend the patch into the source image in order that the discontinuities along the seam after performing Markov random fields and Graph cuts disappear as much

as possible. In addition to remove discontinuities along the seam, Poisson blending also has other benefit that it can blend the color of the pasted patch to match the surrounding area of compositing region as you can see in Fig. 1. The cloud of the resultant video clip (Fig. 1(c)) has imitated the color and illumination from the based video clip (Fig. 1(a)).

3. Preservation of Continuousness

We described our two-stage composition techniques: the energy minimization process, using Markov random fields and Graph cuts, and Poisson blending. These methods can minimize the discontinuities along the seam of the composited region but they cannot insure the continuousness of the whole video. We assume that adjacent frames in video should not be changed abruptly. Therefore we restrict these changes by controlling the masked region to be gradually changing. Since the energy along the seam of the composited region is needed to be minimized and inner region can be ignored, we generate new mask for energy minimization process by focusing only on external edge of the masked region as shown in Fig. 14(b).

In order to preserve continuousness of the video, a mask from the previous frame is used as the initial value to compute energy minimization of the current frame. Since the initial mask used for computation is taken from the previous frame so the previous mask and the new mask will be very much alike. This will comply with our assumption that the adjacent frame in video should not change so much.



(a) Original mask.

(b) Mask for computing energy minimization

Figure 14 Generating mask image for computing energy minimization.

RESULTS AND DISCUSSION

Results

Since we assume that the adjacent frames in video should not change abruptly, so we restrict our test data to be time-lapsed video clips. The time-lapsed video has the property that matches our assumption. The transition from frame to frame in time-lapsed video does not change so much. We downloaded our test data from www.vimeo.com, a web site that has many kind of time-lapsed video.

Because our primary goal is to composite a region not an object in video with minimum discontinuities so we can avoid tracking object. One of our results is shown in Fig. 15, the first, second and third rows are base, patch and result video, respectively. We mask the road in the middle of the scene in the base video and replace it with the one in the patch video. The result in the third row of Fig. 15 shows that our algorithm can cut along strong edge, the ramp on the edge of the composition border. Our algorithm also adjusts color and illumination of the pasted patch so that it can smoothly blend into the base video clip. The additional results can be founded in Appendix B.

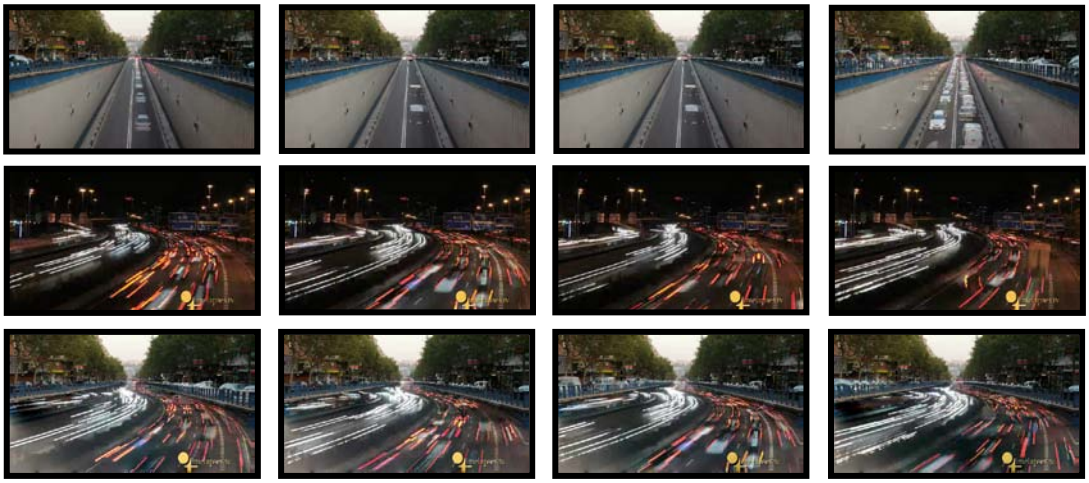


Figure 15 The result of our algorithm, "The busy road".

We generate five result video clips from our system (App. Fig. B1 – B5) and use the same input to create five result video clips from Sony Vegas, the video editing software from Sony (App. Fig. C1 – C5). Then we gather twenty people to observe the clips and let them specify a region which they believe the composition process occurred. The comparison result is shown in Fig. 16. The recall rate of our algorithm is just thirty-one percent meanwhile Sony Vegas has the recall rate up to fifty-seven percent. Our recall rate is about half of Sony Vegas recall rate. False positive and undetectable rate of our algorithm are forty-two and twenty-seven percent respectively which are more than Sony Vegas that has thirty and thirteen percent. The

twenty people we gathered, we mainly divided them into two groups, normal user and expert user. A normal user group, we choose a person who use computer in everyday life and work but not dwell deep into technical stuff. An expert group, we choose a person who has some experiences in image and video processing technique.

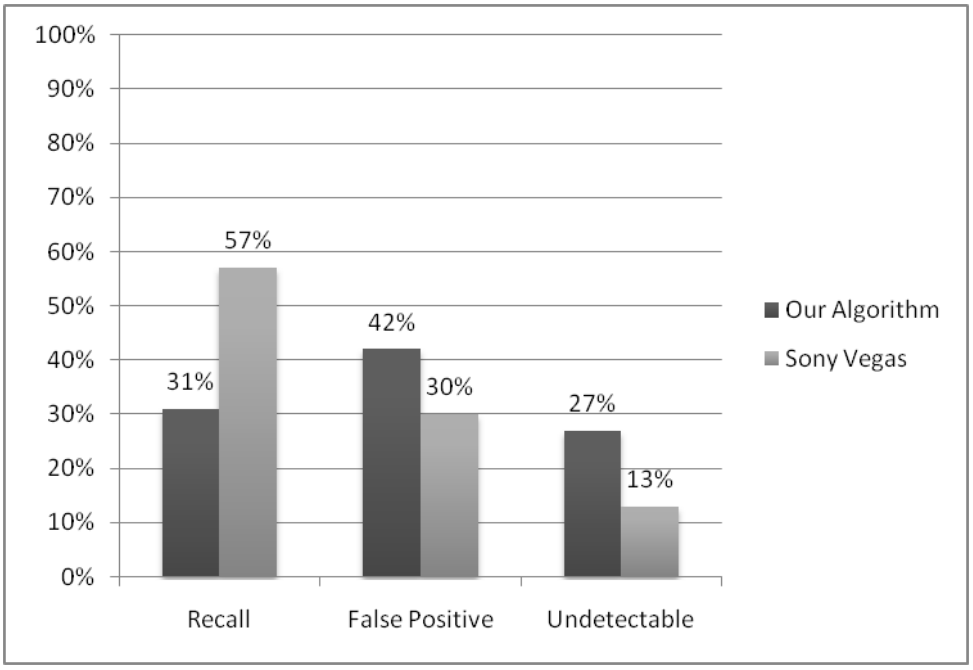


Figure 16 A chart indicates our experimental result.

The comparison chart of only normal user group is shown in Fig. 17. The recall, false positive and undetectable rate of our algorithm is twenty-eight, fifty-four and eighteen percent respectively meanwhile Sony Vegas has fifty-six, thirty and fourteen percent. This shows that a person with no knowledge of image or video processing technique can distinguish a region in video performed composition from Sony Vegas two times greater than a video generated from our technique.

Fig. 18 shows the comparison chart of expert user group. The recall, false positive and undetectable rate of our algorithm is thirty-four, thirty and thirty-six percent respectively meanwhile Sony Vegas has fifty-eight, thirty and twelve percent. Again, the result shows that a video composited from Sony Vegas is easier to detect for about two times more than a video generated from our algorithm.

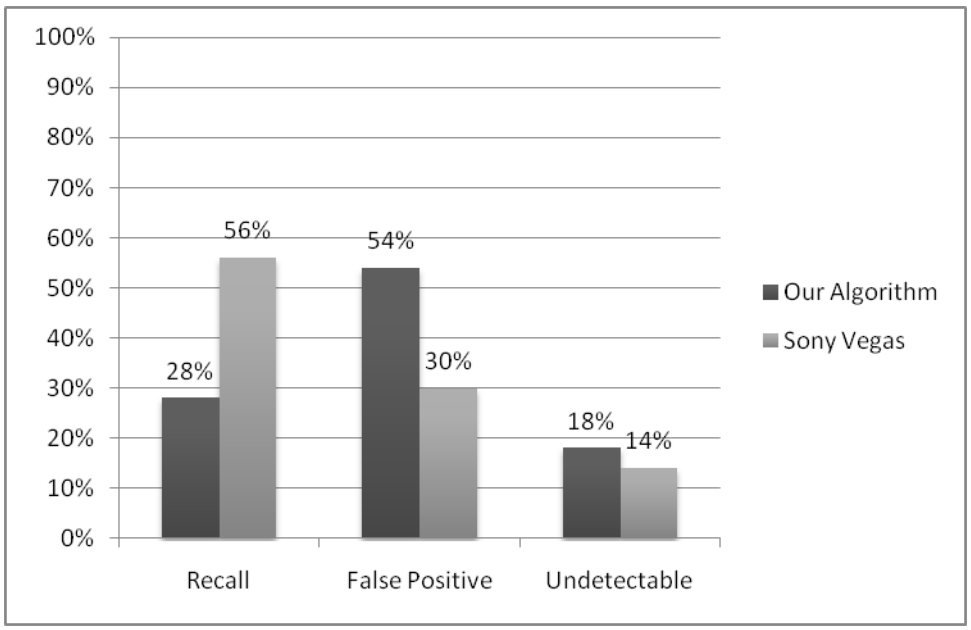


Figure 17 A chart indicates our experimental result of normal user group.

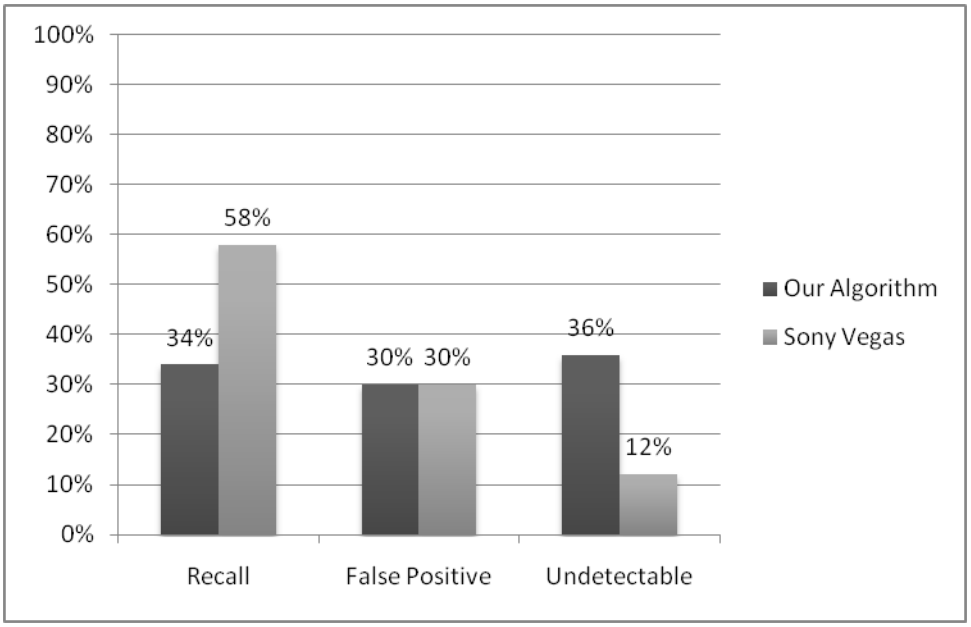


Figure 18 A chart indicates our experimental result of expert user group.

Discussion

We compare our evaluation result between normal user group and expert user group in Fig. 19. The recall rate of expert user group is higher than normal user group as expected. But the result indicates that the expert can recall just six percent more than the normal user. This shows that even the expert who has knowledge of image

and video processing cannot detect the differences between base and patch region in the video generated from our technique more than normal user much.

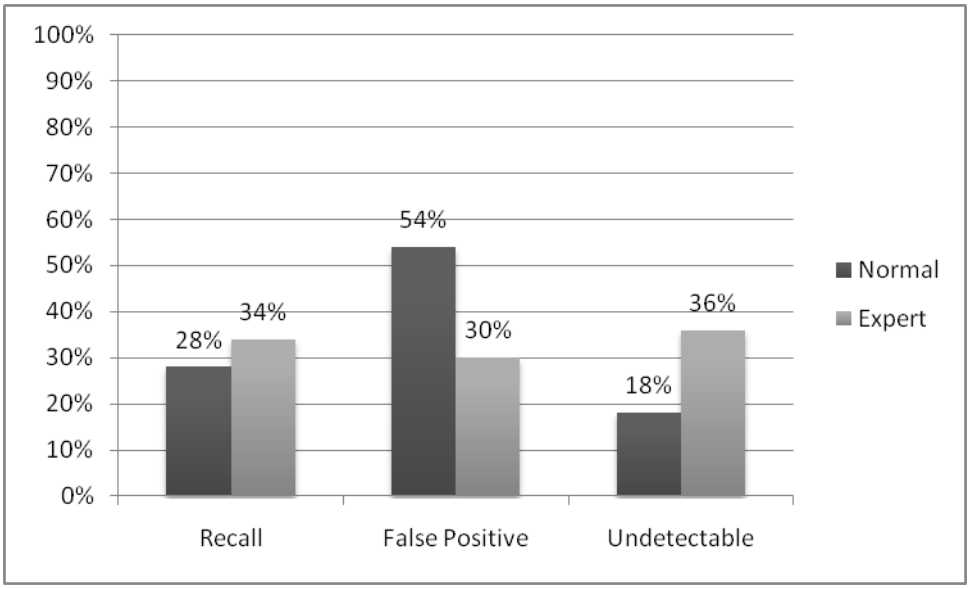


Figure 19 A chart comparing our result between normal user group and expert user group.

Even though, Sony Vegas can perform video composition as well but Sony Vegas can only do simple cut-and-paste operation along with some modification of the mask image for each frame such as decrease brightness of the mask by 0.1 along the compositing range. This will decrease the brightness of the patch image before paste it into base image. This modification is not automatic. It needs scene editor to specify the value and operation to perform. Our algorithm performs operation like this automatically and will help reduce the time needed to composite video clip greatly.

We compare our result with Sony Vegas result in App. Fig. D1 – D5. In order to see the differences more clearly we also indicate the compositing region in our result compare to Sony Vegas result.

CONCLUSION AND RECOMMENDATION

Conclusion

This research introduces an automatic method to composite video clips with the least visible discontinuity. We can minimize the differences of both color and illumination between the pasted patch and the base video. Our technique also preserves the continuousness of the result video clip.

Our algorithm begins with computation of descriptors of the input video clips, which derived from a Laplacian image and a L*a*b* color image for the edge and color energy respectively, then used these descriptors to find the most appropriate position in video sequence to start compositing. The main goal of this step is to find the sequence that minimizes coarse differences between two input videos. Then we compute Markov random fields by using Graph cuts technique to minimize the energy. This process is like the refinement from the previous one. The first step minimizes the energy in whole image scale when the second step minimizes the energy in pixel scale. After that the seam might still be visible, Poisson blending is used as the last step to blend the pasted region into the base image and make the seam smoother along with color modification to match the tone of the base image. Now that we can make the pasted region smooth, we also constrain the changes of mask image between adjacent frames by using the computed mask from the previous frame to be the beginning point in computation of Markov random fields of the current frame. With this constrain, we can prevent the abrupt changes along the seam of the compositing region between adjacent frames and make the transition of the result video clip smooth.

Our algorithm yields good result in both within frame, minimizes discontinuity along the compositing edge, and between frames, prevent the abrupt changes of the compositing edge between adjacent frames and make the transition smooth, with the assumption that the scene of adjacent frames do not change completely. There still have lots of problem that we cannot handle such as an interaction between object in the video or changes which occur by modification of camera properties like zooming.

Recommendation

One of our drawbacks is that we do not track changes of region we are interested in so our algorithm cannot remove moving object in video clip. This problem can be solved by adding region tracking capability into our algorithm. We recommend region tracking instead of object tracking because our algorithm aim to remove or replace region and region tracking can be used with more type of video clip. Implementing region tracking has other benefits since it will solve the problems

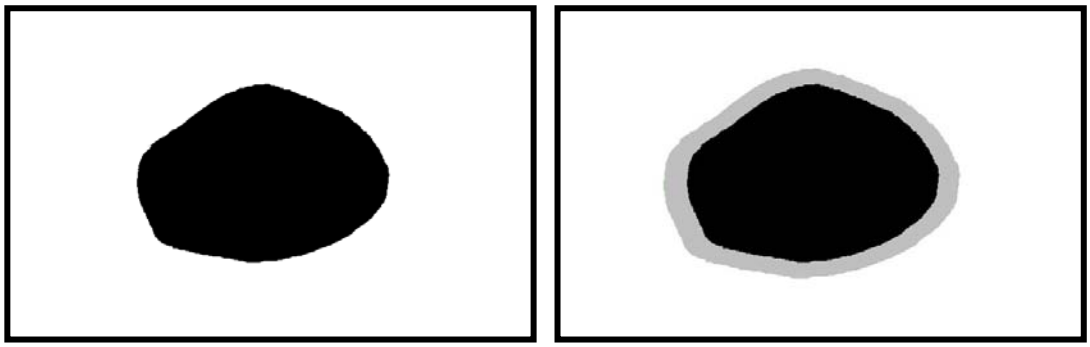
that occur because of modification of camera properties like zooming or panning as well.

Another drawback of our algorithm is an interaction between object, this drawback might be overcome by expanding mask image to cover up the object outside our region of interest which interact with the object inside the region. Of course this method is not ideal so we might need some artificial intelligent techniques to help us determine which and when we should expand the mask image to cover the object outside our region.

This work can be extended in several ways. In the first process “Scene matching”, we use average values of both Laplacian image and L*a*b* color image as our scene and video descriptors. These descriptors are quite rough and can be improved by adding some factor to indicate the priority of energy such as (0.7/0.3) this will make the edge energy more important factor than color energy. Another improvement for scene matching process is changing the method of comparing descriptors. In this research, we use simple algorithm which is computing an angle between vectors. If we change this comparison algorithm to be more complicate such as Mahalanobis distance, the comparison result might be more accurate.

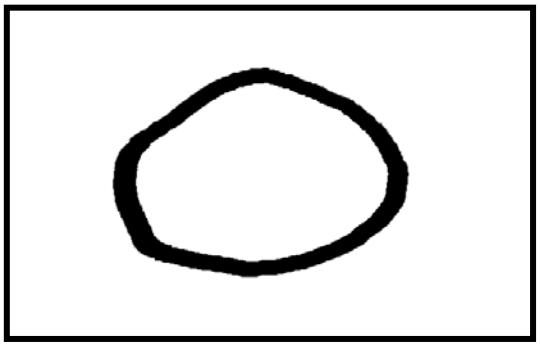
The second process “Image compositing” can be improved by modify Poisson blending operation to take the computed parameters of the Poisson equation from the previous frame into account when compute parameters of the current frame. This will greatly reduce computing time and will also yield smoother result in both image composition and transition between frames.

Finally, the mask image, in Fig. 14(b), which is used to constrain the region to compute Markov random fields and prevent abrupt changes between adjacent frames can be extended by dynamically compute the optimal region instead of a fixed region as in this research we use fixed ratio to expand the mask from the edge of the composited region as illustrated in Fig. 20(b). With this change, the algorithm support range will be wider.



(a) Original mask.

(b) Expanded region from original mask.



(c) Mask for computing energy minimization

Figure 20 Steps of generating mask image for energy minimization.

LITERATURE CITED

Boykov, Y. and Kolmogorov, V. 2004. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. **IEEE Trans. Pattern Anal. Mach. Intell.** 26: 1124-1137.

Boykov, Y., Veksler, O. and Zabih, R. 2001. Fast approximate energy minimization via graph cuts. **Pattern Analysis and Machine Intelligence.** 23: 1222-1239

Criminisi, A., Pérez, P., and Toyama, K. 2003. Object Removal by Exemplar-Based Inpainting. **Computer Vision and Pattern Recog, CVPR'03.** 2: 721-728.

Hays, J. and Efros, A.A. 2007. Scene completion using millions of photographs. **ACM SIGGRAPH 2007.** 4-10.

Jain, Anil K. 1989. Fundamentals of Digital Image Processing. **Prentice Hall.** 68-73

János, S. 2007. Colorimetry. **Wiley-Interscience.** 61

Jia, J., Sun, J., Tang, C. and Shum, H. 2006. Drag-and-drop pasting. **SIGGRAPH2006.** 631-637.

Kolmogorov, V. and Zabin, R. 2004. What energy functions can be minimized via graph cuts? **Pattern Analysis and Machine Intelligence, IEEE Transactions.** 26: 147-159.

Kwatra, V., Schödl, A., Essa, I., Turk, G. and Bobick, A. 2003. Graphcut textures: image and video synthesis using graph cuts. **SIGGRAPH2003.** 277-286.

Leyvand, T. Poisson blending source code: http://www.cs.tau.ac.il/~tommer/extra/adv-graphics/PoissonEditing-src-win32.zip

Oliva, A. and Torralba, A. 2006. Building the gist of a scene: the role of global image features in recognition. **Progress in brain research.** 155: 23-36.

Pérez, P., Gangnet, M., and Blake, A. 2003. Poisson image editing. **ACM Trans. Graph.** 22: 313-318.

Szeliski, S., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M. and Rother, C. 2006. A comparative study of energy minimization methods for markov random fields. **ECCV2006.** 16-29.

Torralba, A., Murphy, K., Freeman, W. and Rubin, M. 2003. Context-based vision system for place and object recognition. **ICCV.** 273-280.

Wilczkowiak, M., Brostow, G., Tordoff, B. and Cipolla, R. 2005. Hole Filling Through Photomontage. **BMVC05.** 492--501.

APPENDICES

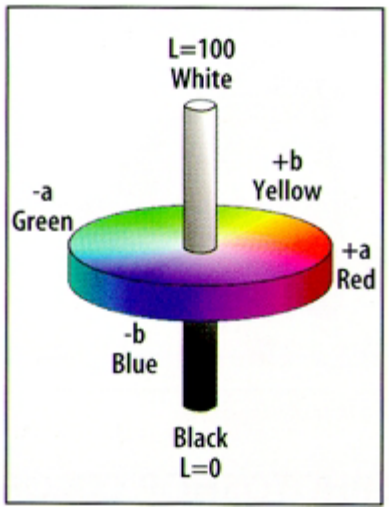
Appendix A

L*a*b* color space

An L*a*b* color space is an abbreviation of CIE 1976 LAB color space which is derived from CIE 1931 XYZ color space. Even though the color space is called CIE LAB, the components are actually L*, a* and b*. L* is the lightness value with value range from 0 (black) to 100 (white). a* is the redness/greenness with positive a indicates red and negative a indicates green. b* is the yellowness/blueness with positive b indicates yellow and negative b indicates blue. App. Fig. A1 shows an illustration of L*a*b* color space in three dimensions.

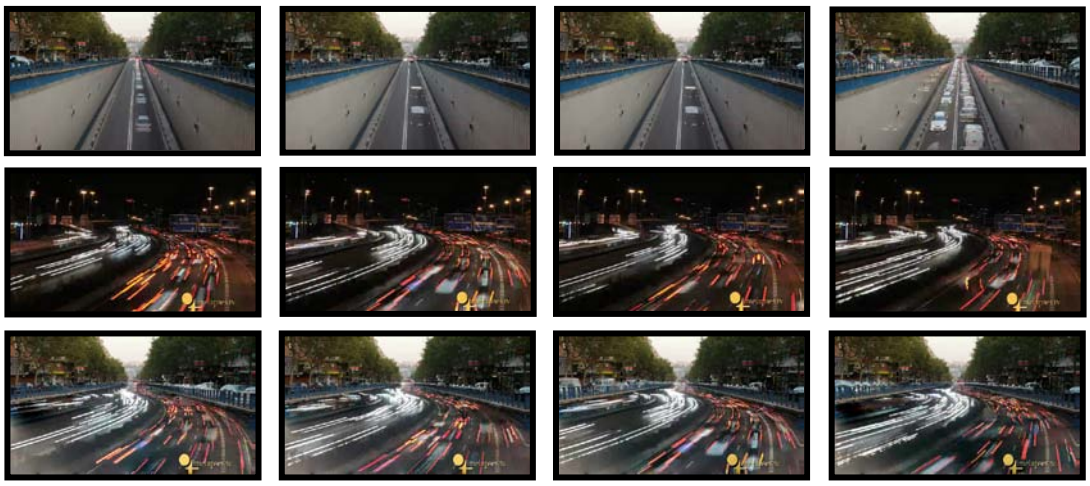
The L*a*b* color space is designed to approximate human vision unlike other color space such as RGB or CMYK. The L* component of L*a*b* is closely matches human perception of lightness so the color balance can be adjust by modify curves of a* and b* component. Since L*a*b* color space is designed to match human vision and the value is measures in a standard lightness (the light from fluorescent), it is used as a reference when dealing with color for human needs such as a color for a furniture in your bedroom.

The disadvantage of L*a*b* color space is that it can represent much more color than RGB and CMYK color space, that designed for device output. So when L*a*b* color image is stored in computer, it needs more spaces than those of RGB and CMYK. And it needs to be converted into RGB or CMYK when display on computer screen.



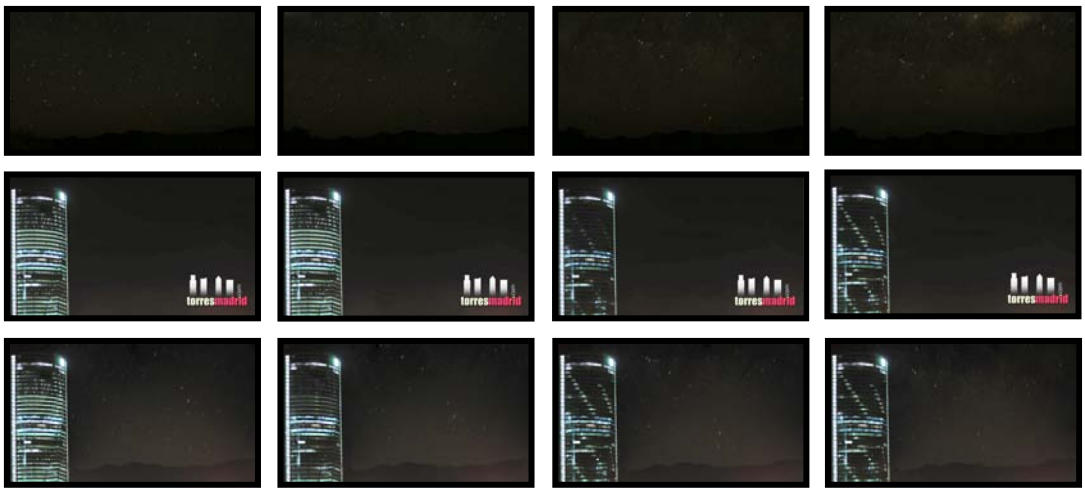
Appendix Figure A1 An illustration of L*a*b* color space in three dimensional space.

Appendix B
Experimental Results



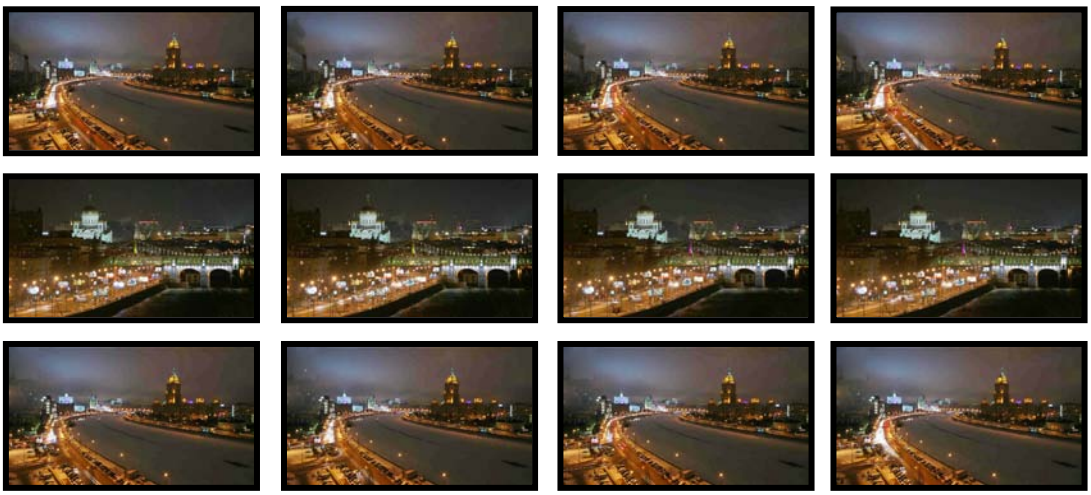
Appendix. Figure B1 The result of our algorithm, "The busy road".

Appendix Figure B2 The result of our algorithm, "The stormy cloud".



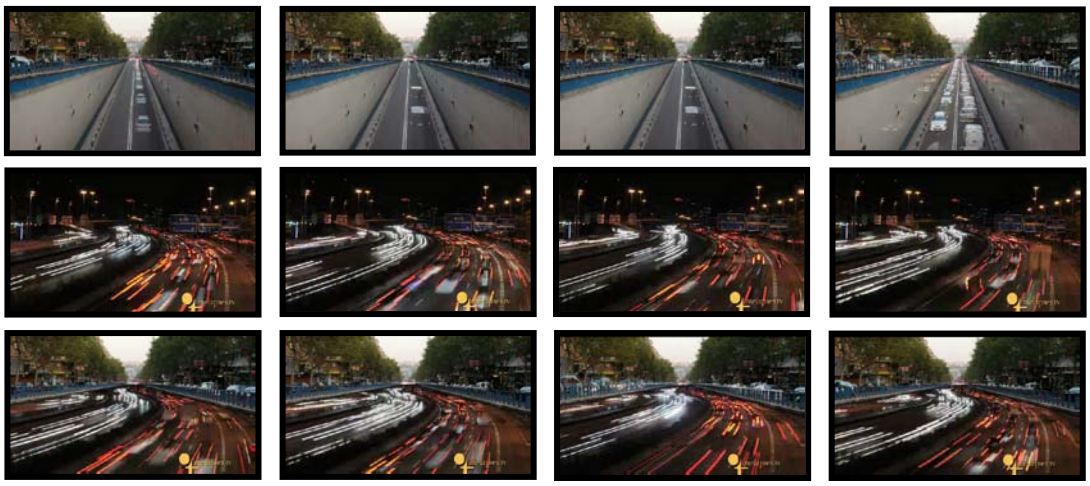
Appendix. Figure B3 The result of our algorithm, "The starry night".

Appendix Figure B4 The result of our algorithm, "The missing dome".



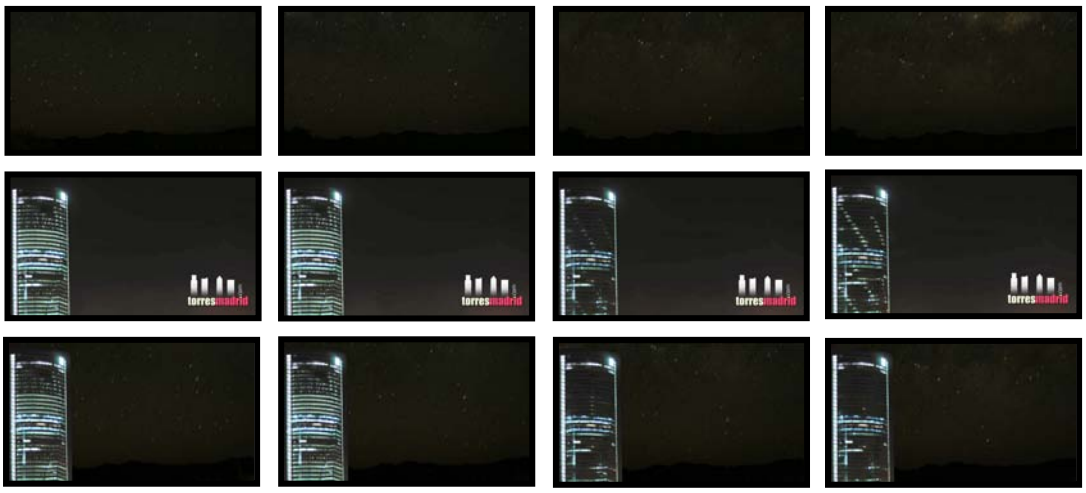
Appendix Figure B5 The result of our algorithm, "Get the factory away".

Appendix C
Sony Vegas Results



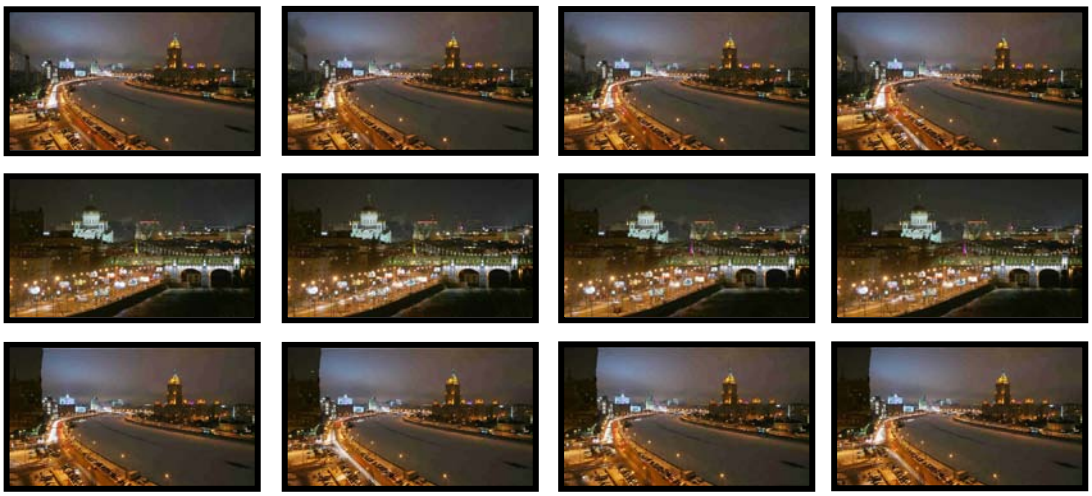
Appendix Figure C1 The result of Sony Vegas, "The busy road".

Appendix Figure C2 The result of Sony Vegas, "The stormy cloud".



Appendix Figure C3 The result of Sony Vegas, "The starry night".

Appendix Figure C4 The result of Sony Vegas, "The missing dome".



Appendix Figure C5 The result of Sony Vegas, "Get the factory away".

Appendix D
Compare Results

(a) A frame from our result video clip “The busy road”.

(b) A frame from Sony Vegas result video clip “The busy road”.

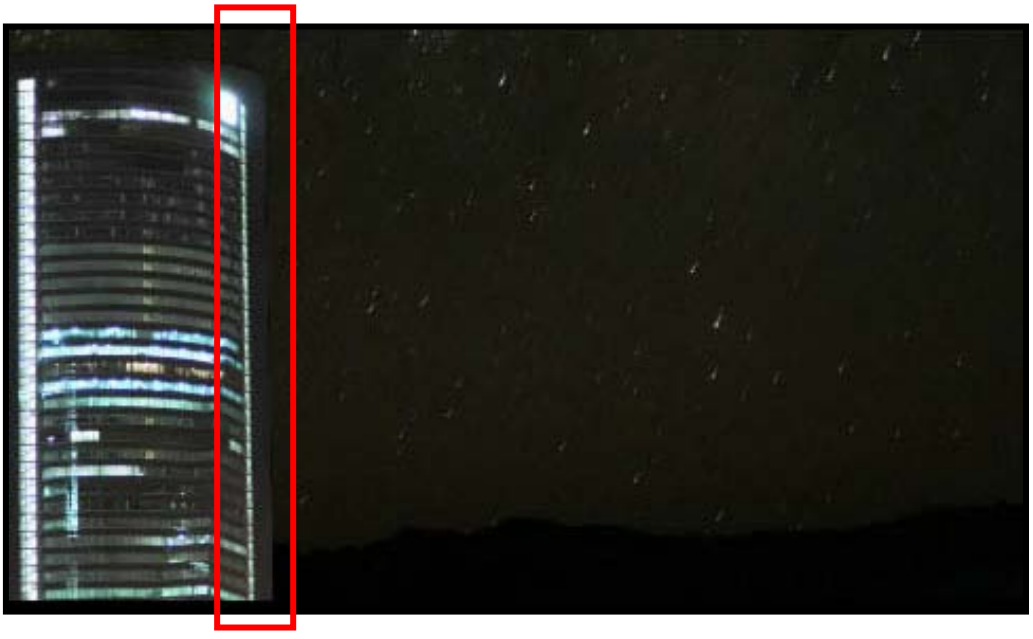
Appendix Figure D1 Compare “The busy road” between our result and Sony Vegas result. The most noticeable difference is indicated in red rectangle.

(a) A frame from our result video clip “The cloudy city”.

(b) A frame from Sony Vegas result video clip “The cloudy city”.

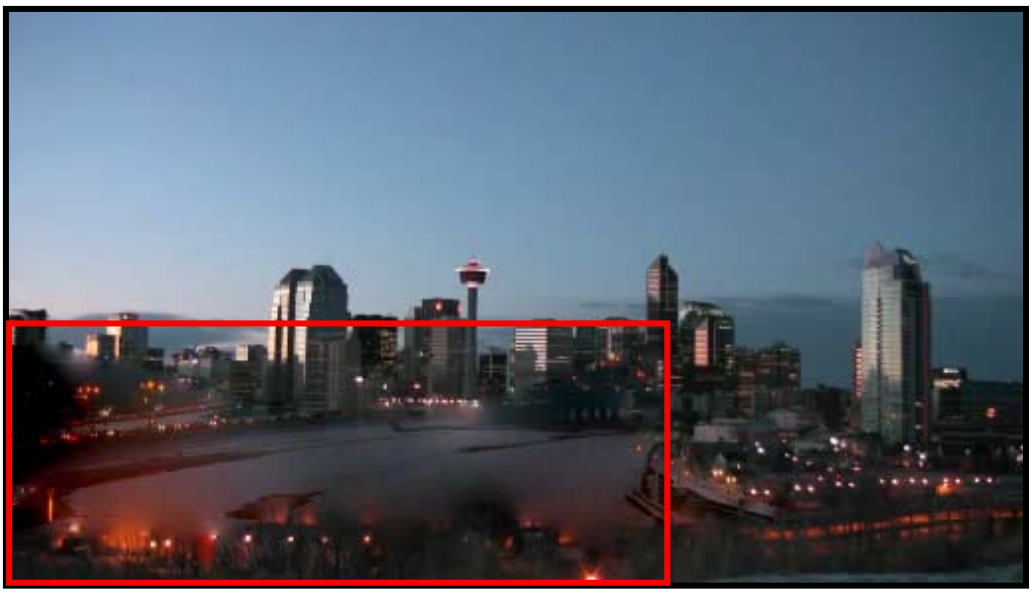
Appendix Figure D2 Compare “The cloudy city” between our result and Sony Vegas result. The seam of the compositing region in Sony Vegas result is clearly visible indicated in red rectangular. Even though the seam in our result has blurring effect from Poisson blending but it is just a minor issue since the seam is hided completely.

(a) A frame from our result video clip “The starry sky”.

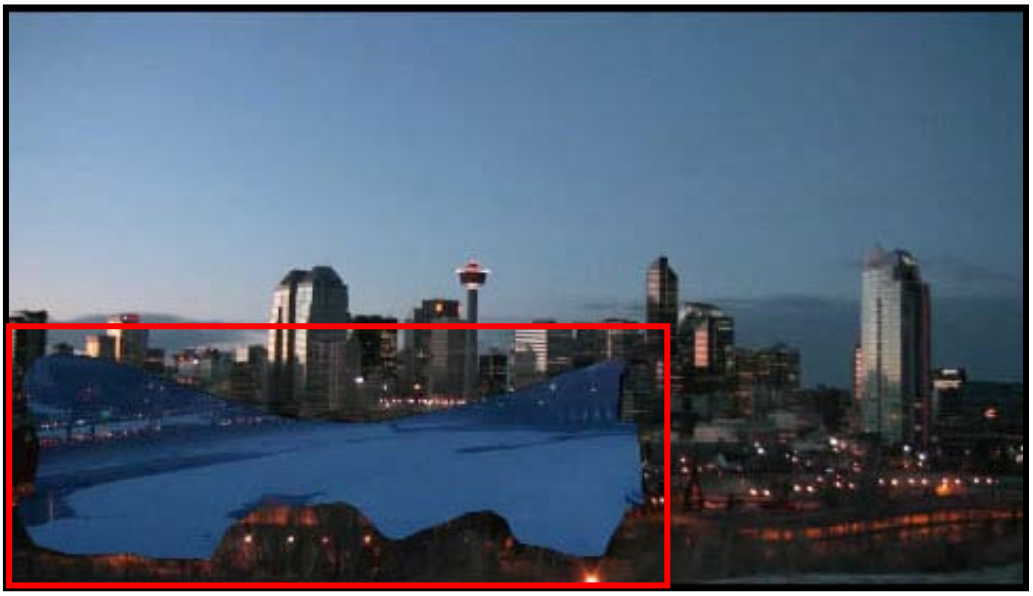


(b) A frame from Sony Vegas result video clip “The starry sky”.

Appendix Figure D3 Compare “The starry sky” between our result and Sony Vegas result. The seam of the compositing region in Sony Vegas is clearly visible indicated in red rectangle.

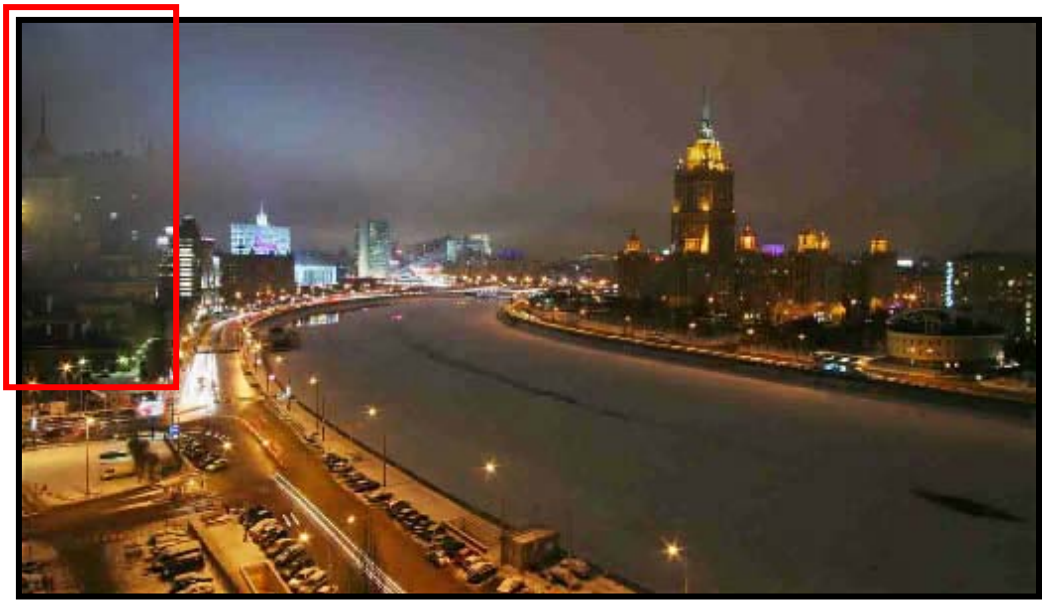


(a) A frame from our result video clip “The missing dome”.



(b) A frame from Sony Vegas result video clip “The missing dome”.

Appendix Figure D4 Compare “The missing dome” between our result and Sony Vegas result. Since Sony Vegas do not has the ability to search within patch video to find the most suitable position to start compositing so the color and illumination differences between frame from base and patch is too much. As a result the compositing region in Sony Vegas result is clearly noticeable which is indicated in red rectangle.



(a) A frame from our result video clip “Get the factory away”.

(b) A frame from Sony Vegas result video clip “Get the factory away”.

Appendix Figure D5 Compare “Get the factory away” between our result and Sony Vegas result. Sony Vegas do not has feature to automatically adjust color and illumination of pasted region to match with the base image so the pasted region in Sony Vegas result is clearly visible in red rectangle. Our result use Poisson blending to adjust both color and illumination of pasted region to match the base image. As you can see the sky from pasted region is blended to match the sky from base image so it looks as if it is the same sky.

CIRRICULUM VITAE

NAME : Mr. Ukrid Kuldiloke

BIRTH DATE : June 6, 1985

BIRTH PLACE : Ratchaburi, Thailand

EDUCATION :

YEAR	**INSTITUTE**	**DEGREE/DIPLOMA**
2006	Kasetsart Univ.	B.Eng. (Computer)

POSITION/TITLE : Computer Scientist

WORK PLACE : Kasetsart University Research & Development Institute