SPECIAL EFFECTS IN FILM MAKING WITH OBJECT BASED TRANSFORMATIONS

Chun-hao Wang[1], *Xiaoming Fan*[1], *Ming Du*[2], *Bruce Elder*[3], *Xiaoou Tang*[4], *Ling Guan*[1]

[1] Ryerson Multimedia Laboratory, Ryerson University, Toronto, Canada
[2] Department of Electrical and Computer Engineering, University of Maryland, MD, USA
[3] Graduate Programme in Communication and Culture, Ryerson University, Toronto, Canada
[4] Microsoft Research Asia

ABSTRACT

The New Media Initiative project of Ryerson University had been successful in applying image and video processing techniques in creating special effects in film making. Our system utilizes a graph cut image segmentation and snakes active contour approach to obtain object cut outs in 3-D with tracking. It uses exemplar-based inpainting for background filling for missing regions uncovered by object transformation. The two methods combined allows for easy video object editing with reduction in user input. Previously implemented shot detection by twin window amplification method and steerable pyramids texture generation is also part of the system. Lastly, a set of transformation was implemented that serves as a basis for testing the system.

1. INTRODUCTION

Computer generated special effects is a staple of the film and tv industries, but it requires a large resource of computational power and talents. Tools for graphic artists and film-makers have improved in a rapid pace in the recent years, and experimental film making demands new and novel ideas for enhancing computer assisted film making. Our New Media Initiative project aims to automate certain tasks in film production using a combination of image & video processing and machine learning techniques.

The New Media Initiative is jointly sponsored by Natural Science and Engineering Research Council of Canada and Canada Council for the Arts. A multidisciplinary research team was formed from the collaboration between The Department of Electrical and Computer Engineering and School of Image Arts at Ryerson University, Canada.

The project is focused on several objectives: to automatically select a particular object of interest and perform object based transformation, and to learn the processes in which film-makers use in special effects via machine learning. In this paper we present a system and tools to perform object based transformation. Our goal is to reduce the effort and time required for complex image and video processing required in filmmaking.

2. RELATED WORK

The research on extracting objects of interest from images and automatic image inpainting has progressed at a quick pace in the last decade. Interactive Video Cutouts [1] presented a system for extracting video objects of interest using a min-cut algorithm with matting and hierarchical decomposition preprocessing. Proscenium [2], a system for spatio-temporal video editing explored the idea of using a video cube to represent spatio-temporal dependencies of video objects. It allowed for the concept of editing video objects. Also related are advanced image segmentation techniques GrabCut [3] which is built upon graph cut (min-cut max-flow algorithm) that can be extended also to videos with minimal amount of user interaction. Our system utilizes a combination of different techniques.

Previous work [4] introduced a special effects system using steerable pyramid background generation, shot detection, and graph cut object segmentation. Our system addresses new challenges in using a more robust background generation algorithm, transformations, and extending segmentation and inpainting to videos.

3. OVERVIEW OF THE SYSTEM

The block diagram of our system is illustrated in Fig. 1. The system first loads a clip or shot of video to be edited. The user needs to specify the range of frames to be edited and the mask color. The first step is to mark the regions into the foreground and background for object segmentation. Second step is to perform object transformation on the extracted object cutout. Third step is to perform automatic background generation on the extracted background with missing object as the region to be inpainted. The final step is just a step to paste the transformed object back onto inpainted background.

There are two types of transformations, one that changes entire frames (blurring, fadeout, etc), or just an object of interest (scale, rotate, twist, etc). Object based transformations cannot be directly applied to a video sequence without a distinct separation of foreground and background. Furthermore,

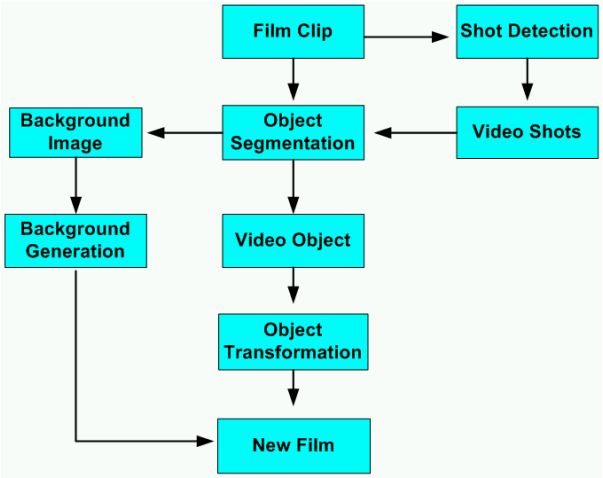


Fig. 1. The special effects system

transformations that alter shape of the object in anyway will cause missing regions in the original frame. Our New Media system overcomes this problem using automatic video object segmentation and video inpainting.

The system is module-based thus it can process a batch of images only for object segmentation, then combine with another software for object transformations. Fig. 2 shows the current prototype user interface of the system. It allows the user to mark pixels during object segmentation and select transformations and its parameters. The program also takes input for selection of video clips and displays the clip information. The program saves the intermediate results in a working directory so external applications can be used to process extracted object, background or the transformations.

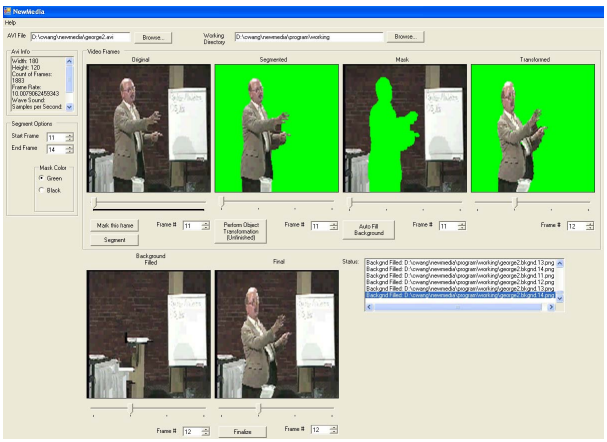


Fig. 2. User Interface

4. TOOL DEVELOPMENT

4.1. Object Segmentation

Mentioned in the previous paper, object segmentation is implemented using two different methods: 1) graph cut [5] segmentation and 2) snakes [6] (active contours). The two methods complement each other with different advantages. Snakes can provide a more accurate contour at the cost of more user marking and processing time. Graph cut requiring less user input, as users only need to roughly mark desired region and background. Graph cut is usually favored as users can provide a quick rough estimate of the object, and if needed, modify the seeding as needed.

A unique strength of graph cut discovered in this work is that, by incorporating temporal information, graph cut can track an segmented object effectively in a video sequence. When a user marks the foreground and background pixels of one frame, it can propagate the information across the entire shot to cut the object over the sequence of frames. The propagation is done by linking each individual image graph together to construct a 3-D graph. Thus each pixel or node on a graph will be a 6-connected. It connects to nearest four neighboring pixels spatially, and also to the previous and next frame temporally using the edge cost $B_{\{p,q\}}$. Assuming that the user wants to track one object only, more emphasis was placed on the edge cost rather than the relative cost $R_p(A_p)$.

It is, therefore, possible to get reasonable cut of an object in a video sequence from marking just one frame, in either the beginning, middle, or end of the shot. An example is shown in Fig. 3, where an object of interest (the person and the sail) is marked in frame 2 only. The system reliably tracked and cut the same object past frame 30. In general, tracking and segmentation continue until a scene change or until the object exhibits a radical change in shape or color. This allows for a cut over a multitude of frames with the same amount of input as for object cut of a single image. Note that if the user wants to also track the board, multiple frames should be marked since it is of similar color to the background and merges in with the water in some frames.

(a) Frame 2 (b) Frame 30 Orig (b) Frame 30 Cut

Fig. 3. Automatic tracking of object

4.2. Background Generation

Background generation was first implemented using Steerable Pyramid texture generation. It is a linear multi-scale multi-orientation image decomposition method. It worked well for homogeneous textures, but does not work well for objects with clearly defined boundaries and produces extra blurring for large regions. Improving upon the previous method, we incorporated another background generation method based on the Exemplar-based inpainting [7]. It combines both advantages of inpainting in extending contours and texture generation in filling in homogeneous patch of pixels. The goal is to fill in target region Ω, one patch at a time $\Psi_{\mathbf{p}}$. The exemplar based algorithm assign each pixel $\mathbf{p}$ with a priority value $P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p})$, where $C(\mathbf{p})$ represents the confidence and $D(\mathbf{p})$ the data term defined in Eq. 1. $\nabla I_{\mathbf{p}}^{\perp}$ is the isophote (direction of the intensity) and $\mathbf{n_p}$ is the unit vector normal to the contour of target region.

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\Psi_{\mathbf{p}}\cap\overline{\Omega}} C(\mathbf{q})}{|\Psi_{\mathbf{p}}|} \tag{1}$$

$$D(\mathbf{p}) = \frac{|\nabla I_{\mathbf{p}}^{\perp} \cdot \mathbf{n_p}|}{\alpha}$$

Confidence is a direct calculation based on the number of known pixels in the patch, thus favoring patches near the boundary. Data term calculates the strength of the isophote is in the boundary, giving preference to extensions of existing edges and contours. After calculation of the highest priority region to be filled in, an exhaustive search is performed. In our system, we also extend the filling information by searching source regions temporally. The search for the best exemplar source patch $\Psi_{\mathbf{q}}$ is

$$\Psi_{\mathbf{q}} = \arg\min_{\Psi_{\mathbf{q}}\in\Phi} d(\Psi_{\mathbf{p}}, \Psi_{\mathbf{q}}), \tag{2}$$

where Φ is the source region in the current image as well as $\pm k$ frames. $d()$ is currently the sum of squared differences. Additional postprocessing is required to prevent different frames with different source patches from producing a stuttering background when each patch is different. To solve this problem, the patch with the best match is copied across the frames, on the assumption that the background does not vary much. So far, for simple shots it can produce smooth background temporally, however it may be possible generate better video inpainting results by using source patches from all frames and smoothing out the target regions.

4.3. Object Transformations

Special effects (SFX) are prevalent in many different fields of video processing. It is used in film, television, and entertainment industries and can range from text overlays on live news broadcasts to completely rendered scenes. In testing our New Media system, several transformations were implemented for the film editors to choose from.

The transformation modules implemented in our system are swirl, color change, nudge, explode, tear, and a set of 3x3 window filters. Previously implemented mesh warping and B-spline mapping were also included. These transformations are all optical and object-based transformations, rather than frame-based. Each transformation can be performed on a video sequence independently or combined together. These transformations are chosen that can represent a set of possible special effects for the film-maker. The set only include basic convolution and geometric warping algorithms, but it is meant to represent a basic set that form the basis for other transforms. Most basic image processing such as blurring, sharpening, translation, and rotation can be formed by these transformation modules. The set of transformations will be used in a machine learning algorithm in future research to automatically learn what transformations an artist would apply to a certain scene.

After image transformation, all the images are recomposed using background or objects have been cut by graph cut module, and put into a new video sequence. User could see the image migrating gradually in the revised video sequence.

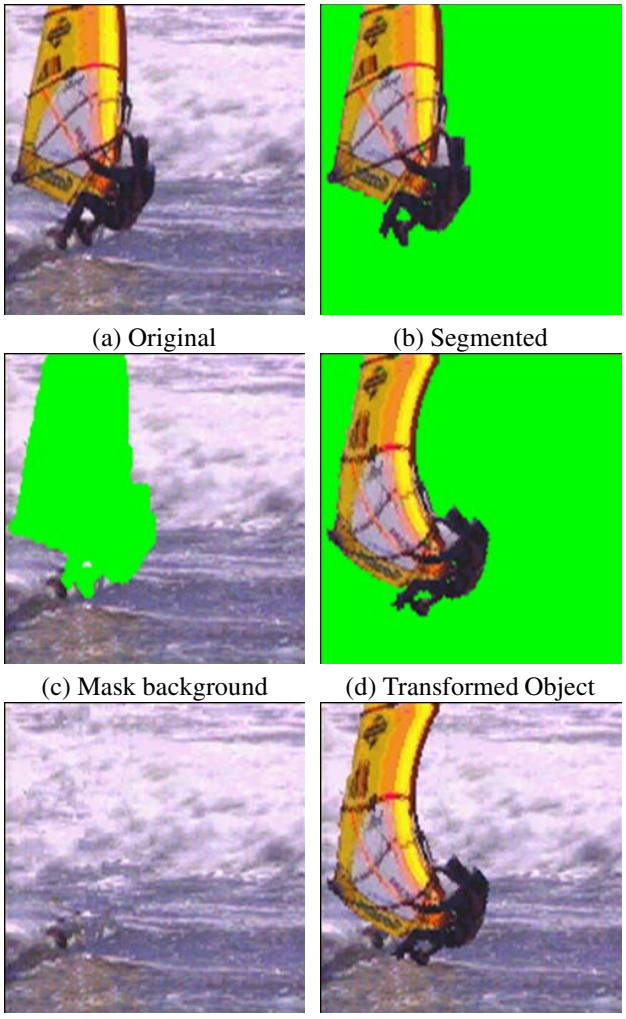
The color change module changes the RGB value of the object. The red, green, and blue values of the object could be changed by sliding the color value bar separately. The explosion modules shrinks or expands the object by selecting a scaling factor. The swirl module twists the object with varying angle corresponding to the distance from object center and the varied angle could also be defined by user. The rotate module is a little similar to the swirl module, the difference is the later module rotates the whole object with a fixed angle without varying the angle by distance from pixel to the center. The nudge module is a combination of swirl and translation that moves the object, but also depends on pixel distance from object center. The tear module splits the object into half and push two parts of object to user defined distance.

4.4. Implementation Results

The system is implemented using tools and libraries in C#, C, and Matlab. Some algorithms such as modified exemplar-based inpainting source code is implemented in Matlab and modified into C. Graph cut segmentation is compiled in C for efficiency and both are linked from the main C# user interface. The system was tested on a Pentium 4 2.8Ghz, 1GB Ram machine.

Using a reference video of 320x240 pixels, user marking and seeding for graph cut takes about 2-3 seconds per frame. Just one marked frame is enough to obtain a fair approximate segmentation of the video object. Further marking of additional frames and altering the foreground/background seeding is 2-3 seconds per frame, if pinpoint pixel boundary needs to be defined. Graph cut drastically reduces the amount of effort

and time required for user input. For background generation using exemplar-based inpainting, each frame takes about 3-5 seconds to generate a missing background with object taking approximately 10% of the video frames.



(a) Original (b) Segmented

(c) Mask background (d) Transformed Object

(e) Generated background (f) Final Result

Fig. 4. Step by step results of our system

Figure 4 shows step by step processing of our prototype system. The original video screen capture is shown in a), after the user marked foreground and background for a frame, the automatic video object segmentation is shown in b), and the inverse background mask in c). Next step the segmented object is transformed to create a special effect as seen in d), commercial programs can also be used to produce the transformation. Background generation is then applied to the masked background image to fill in missing regions created by the object cutout, shown in e). The final step combines the transformed object back onto the generated background f). As it can be seen, the results for background generation in e) produces good general background inpainting using the exemplar-based method. For indoor scenes or scenes with complex and cluttered objects, exemplar-based inpainting will not perform well. It is best suited for missing regions with homogeneous texture, and it can regenerate basic contour and boundary using isophote calculations. For purposes of filling in the gap created by transformations, it is sufficient as during the final step, the transformed object will cover most of the missing region.

5. CONCLUSION AND FUTURE RESEARCH

This paper presented an enhanced new media system capable of producing object based special effects using video object segmentation, inpainting, and implemented transformations. It is currently being used by Media Arts department in Ryerson University in their experimental film-making. The system provides intelligent and automatic methods for video object selection, cutout, performing object transformation, and background generation. Our system is designed to reduce the complexity and repetitiveness, while introducing several new powerful tools to manipulate the video sequence.

Different methods for improving the new media system is possible by researching in the area of combining video cutouts with video inpainting. Advanced compositing using alpha channels can be used in the final step to combine the object and background. Lastly, further research is directed towards machine learning to automatically apply special effects based on expert knowledge of the film-makers.

6. REFERENCES

[1] J. Wang, P. Bhat, R. Colburn, M. Agrawala, and M. Cohen, "Interactive video cutout," *Proceedings of ACM SIGGRAPH*, pp. 585–594, 2005.

[2] E. Bennett and L. McMillan, "Proscenium: A framework for spatio-temporal video editing," *Proceedings of ACM Multimedia*, pp. 177–183, 2003.

[3] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut - interactive foreground extraction using iterated graph cuts," *Proceedings of ACM SIGGRAPH*, pp. 309–314, 2004.

[4] C. Wang, Y. Wang, M. Lian, B. Elder, X. Tang, and L. Guan, "Special effects in film/video making: A new media initiative project," *International Conference on Multimedia and Expo*, pp. 1049–1052, 2006.

[5] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," *International Conference on Computer Vision*, vol. I, pp. 105–112, 2001.

[6] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1987.

[7] A. Criminisi and P. Perez K. Toyama, "Object removal by exemplar-based inpainting," *Proc. IEEE Computer Vision and Pattern Recognition*, vol. 2, 2003.