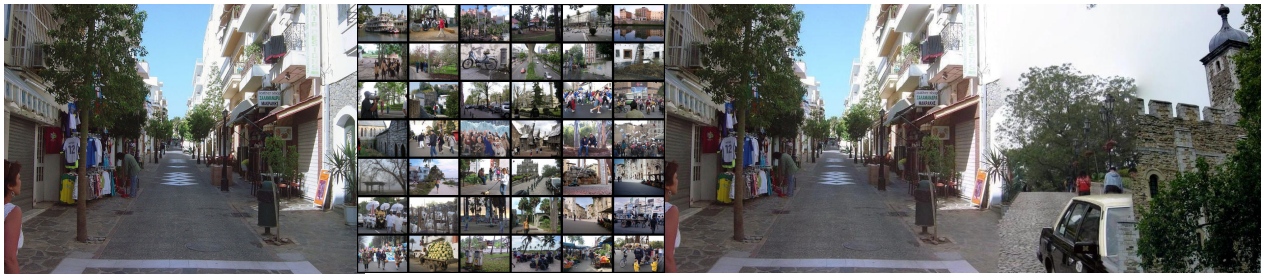
Automatic Generation of An Infinite Panorama

Lisa H. Chan Alexei A. Efros
Carnegie Mellon University



Original Image Scene Matches Output Image

Figure 1: Given an input image, scene matching from a large image database was used to composite an artistic panorama.

Abstract

This paper presents the possibility of using image completion combined with a large image database to create an infinite panorama. The algorithm performs scene matching using a portion of the original input image to find the best matching neighboring scenes and then composites these images in a seamless way. Even though the automatically generated panoramas may not be convincingly realistic, the panoramas are nevertheless aesthetically pleasing and artistic.

Keywords: Image Completion, Image Database, Image Compositing

1 Introduction

With using the traditional methods of creating a panorama, much time and effort are needed to capture the necessary images on location and then to composite the images into a panorama in a photo-editing program. However, another key requirement for compositing a panorama in this way is that the idea of a panorama must have been pre-meditated. If the desire to view a panorama occurs post-image-capture, the only option is to travel back to the original location and capture the necessary images. The problem with this option is that not only is it time and monetarily consuming, but logistically impossible since the mise en scene and plenoptic function will be very different. A possible solution to this whole problem is image completion using a large image database.

The main strategy for image completion in this study involves finding a way to hallucinate the neighboring scene that resides next to a given input image. Existing methods [1-5] in hole-filling employs extending textures found in the input source image into the unknown region. Such methods do not work in this case since the neighboring scene will almost never be a simple extension of the current image. Texture-synthesis will also never allow the creation of an infinite panorama since all the information is not stored in a single source image. The proposed method in creating an infinite panorama therefore lies with searching through a large image database to find a neighboring scene that will fit seamlessly to the original input image.

2 Methods

The scene matching and image compositing methods are largely based on previous work by Hays and Efros [6], and will therefore only be briefly mentioned here. Figure 2 shows the overall schematic of the algorithm used in compositing the images.

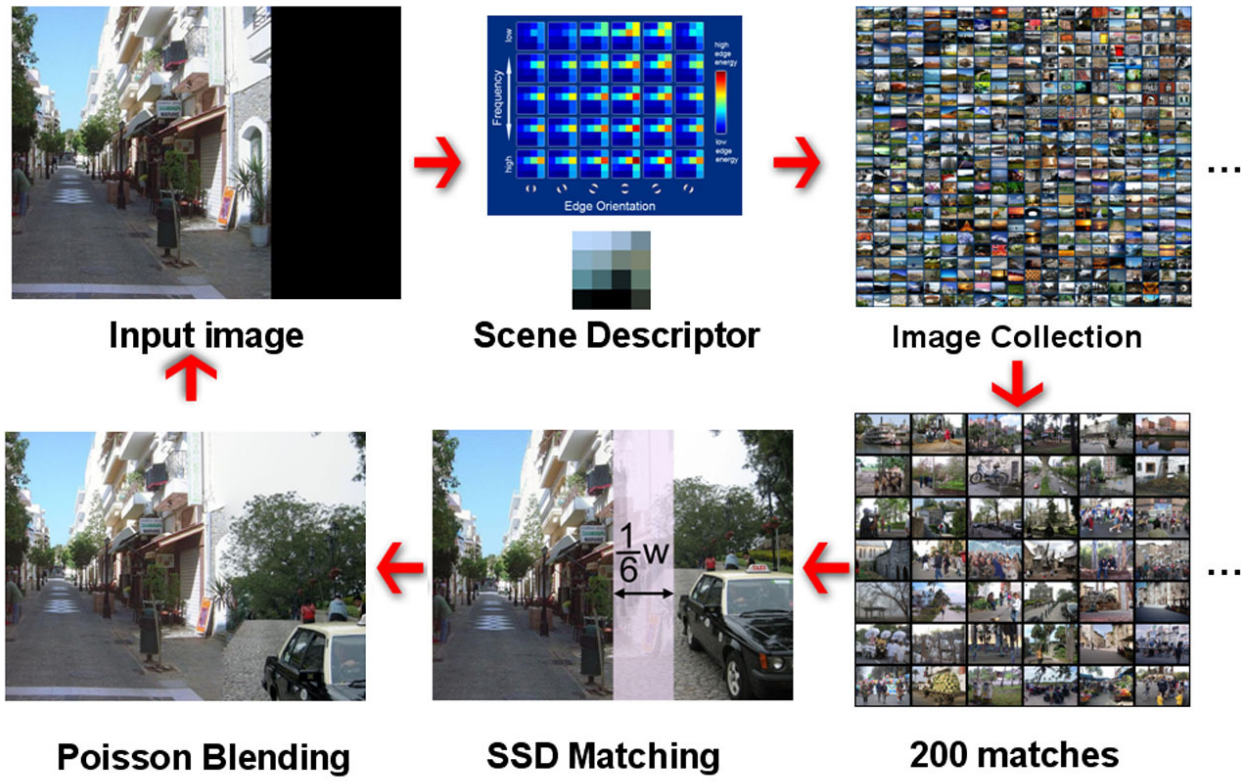


Figure 2: Schematic of the methods used to composite parts of the infinite panorama.

The image database used in this study consists of 2.3 million unique images downloaded from 30 Flickr.com groups that focused on landscape, traveling, or city images. Gist scene descriptors [7, 8] are then used to group semantically similar scenes together for faster searching of nearest scenes. The gist descriptors were built from 6 oriented edge responses at 5 different scales combined to a spatial resolution of 4x4. The scene descriptors also include color information that is a 4x4 down-sampled version of the original images in the database.

Given an input image for compositing, 1/3 of the input image is removed and preserved for later compositing, while a gist descriptor is computed using the remaining 2/3 of the input image. Scene matching is then performed by calculating a gist distance with the sum of the squared differences (SSD) between the gist of the input image and each of the gist descriptors in the database, and a color distance that is computed in the L*a*b color space. The final distance function used to find the top 200 nearest neighbor scenes weights the gist distance about twice as important as the color distance.

With the top 200 nearest neighbor scenes, a final refining search must be performed to decide which image will be used for compositing the input image. Black and white images and images that are smaller than ½ the width of the input image are first eliminated. SSD is then employed to find the best match along the overlap region of the input image and the remaining neighbor

scenes. Since 1/3 of the width of the new database image will be added onto the input image, an overlap region of 1/6 of the width was used to perform the SSD calculation. Other varying overlap regions of 1/8 and ¼ of the width have also been attempted, and the results were consistent with the 1/6 chosen in this study. SSD was calculated separately on the three RGB channels for the neighbor scenes at full resolution. The images were then ranked in three different dimensions based on their SSD distances in each of the color channels. The image that will be chosen is then the image that has the closest match to the original image in all three color channels.

After the neighboring image is selected, graph cut seam finding [9] combined with standard poisson blending [10] are then used to composite the two images. In the seam finding operation, additional pixels from the input image are allowed to be removed to help minimize the gradient of the image differences along the seam. Once the images are blended together, the output image can be reintroduced to the algorithm as a new input image to start a new search for a neighboring scene. The result of the overall algorithm is many smaller pieces of a large panorama. Because poisson blending will actually change the coloring of the images, a simple cut and paste does not work in this case. Instead, a two-level laplacian blending was used to blend all the images together to create the final large panoramic image.

3 Results

Some results from this study are given in Figure 3. Due to the limitation of the size of this paper and the inability to infinitely search for interesting scenes, the images have been cropped to the size shown. Readers are referred to the project web page for viewing the full panoramic images shown in Figure 3. Overall, the images all show nice seamless transitions between the additional images, even though at certain times the scene changes are rather obvious.

Figure 4 shows some of the failures where scenes are blatantly different from the input images. In most of the failure cases, either the person included in the image is simply not the correct size for the type of scene introduced by the input image, or the images were blurry as in the bottom right image. However, it should be pointed out that these failures did not result in the final algorithm created for this study. These failures were simply created by stitching all 200 top nearest neighbor scenes and manually picking out the failures shown in Figure 4.

Readers are shown a glimpse of the problem that arises with the construction of infinite panoramas using scene completion in the last image of Figure 3. Near the end of the panorama, it can be seen that the scene has transitioned from the original high frequency input image into a region of low frequency image additions. The low frequency images added onto the panorama will cause any further image searching to remain in the low frequency region, never being able to recover back into a scene with high frequencies. Therefore, at this point, infinite panoramas may not be possible, unless a large region of low frequency images is desired. However, there are possible solutions for solving the problem. Some of the possible solutions have already been attempted, while others will be the future work of this study.

The possible solutions in resolving the hole of low frequency images rest with additional variables that can be added in either the scene matching code or the local refining search for choosing the final neighboring image. The first solution attempted was to keep the total energy of the gist of the input image, and use it as part of the distance calculation. The results from that search showed that the general coloring of the input image was kept constant while the semantic

scene matching is lost. Other possible solutions that will be attempted in the future includes stitching all the 200 nearest neighboring scenes into one panorama, stitching pairs of images within the 200 scenes and then further searching for the pieces that will join the pairs of images together into a full panorama, and using a combination of histograms and Fourier transforms to maintain the frequency throughout the entire panorama.

Figure 3: Example results with the leftmost part of the images being the starting input images.

Figure 4: Example failure cases where pronounced differences between the images are seen.

4 Conclusions

The algorithm used in this study is able to search for a relatively close matching neighboring scene and composite the images in a seamless manor. The remaining problem with this study is extending the panorama to an infinite length without having the resulting searched images fall into an infinite range of low frequency images. Some possible solutions to the problem have been proposed, and any updates will be added to the project web page. Even though the panoramas created thus far may not look plausible or realistic, they are nevertheless aesthetically pleasing and artistic.

Acknowledgements: The authors would like to thank James Hays for the scene matching and image compositing codes and Jim McCann for many helpful discussions and suggestions.

Project Web Page:
http://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/f07/proj_final/www/lisachan/

References

[1] A. Criminisi, P. Perez,K. Toyama, CVPR 02 (2003) 721.
[2] I. Drori, D. Cohen-Or,H. Yeshurun, ACM Trans. Graph. 22 (2003) 303-312.
[3] N. Komodakis, CVPR (2006) 442-452.
[4] Y. Wexler, E. Shechtman,M. Irani, CVPR 01 (2004) 120-127.
[5] M. Wilczkowiak, G.J. Brostow, B. Tordoff,R. Cipolla, BMVC (2005) 492-501.
[6] J. Hays,A.A. Efros, SIGGRAPH 2007, Los Angeles, 2007,
[7] A. Oliva,A. Torralba, Visual Perception, Progress in Brain Research 155 (2006)
[8] A. Torralba, K.P. Murphy, W.T. Freeman,M.A. Rubin, ICCV (2003)
[9] V. Kwatra, A. Schodl, I. Essa, G. Turk,A. Bobick, ACM Trans. Graph. 22 (2003) 227-286.
[10] P. Perez, M. Gangnet,A. Blake, ACM Trans. Graph. 22 (2003) 313-318.