

Photo Cleaning: An automatic Redundant Object Removal Tool

HU Jianwei^a, Fang Wei^{b,*}

School of mathematics and statistics, Huangshan University, Tunxi, Anhui, 245041, China

^ajianwei@hsu.edu.cn, ^bsxxfw@hsu.edu.cn

*Corresponding Author

Keywords: Image Registration; Color Harmonization; Image Completion; Redundant Object Removal; Image Tool.

Abstract.

This paper proposes a novel technique which removes redundant objects from digital photos. The technique is fully automatic without user interaction. The proposed method is composed of three steps: image registration, color harmonization and redundant object removal. Redundant object removal algorithm is composed of two sub-components: the redundant object detection and the hole filling. Experimental results show that our method produces clean images with visual pleasure.

Introduction

Sometimes we wish we could remove some objects from our photographs since the photographs we took are not clean enough. For example, when we take a photo in some scenic area which is full of people, the photo may be filled with redundant objects. The related work is called image completion which is a popular tool for image editing such as object removal or digital photograph restoration.

In the last decade, scientists developed a variety of image completion algorithms which is now a useful feature in some commercial photo editing software such as Adobe Photoshop. In most previous work, image completion is used to fill holes after unwanted objects are removed. The same algorithms, however, can also be used to extend an image beyond its original boundaries.

There are two main different techniques for image completion. The first technique reconstructs the image as accurately as possible, and tries to restore some occluded objects or parts in the image. This accurate reconstruction technique usually has to use some other source of data in addition as the inputs. Irani et al.[1] use various background stabilization techniques to complete video scenes. Some researchers[2,3] use multiple photographs of the same physical scene as input and attempt to reconstruct an accurate image.

The second technique is to try to fill in the missing parts of the image directly in a plausible way. Based on human visual perception observation, this technique has a much less easily processing steps. Some existing methods[4-7] complete images by extending adjacent textures and contours into the missing area. These methods are inspired by the classical example-based texture synthesis techniques[8-11]. Criminisi et al.[12] proposed an automatic algorithm to complete images with additional constraints to preserve structural cues and Sun et al.[13] achieved the goal by user interaction.

The most popular algorithms published recently focus on removing redundant objects such as people, signs, animals or vehicles from backgrounds in images. Most of these algorithms are only capable of dealing with images with simple backgrounds. Some algorithms can fill holes in more complex backgrounds such as structural background[13], but user interactions are necessary. The most typical techniques are example based image completion methods which are in fact a combination of image texture synthesis and inpainting. Although these algorithms can get some visual pleasure results, they usually are time-consuming work.

Recently, some researchers focus on speeding up the procedure of image completion. Fang et al.[14] use multi-resolution patch-based directional and non-directional approach to complete images. The PatchMatch method proposed by Barnes et al.[15] is a randomized correspondence method for structural image completion. And this approach is really fast, but failed in being applied on the structural regions in some cases. Kwok et al.[16] propose an image completion approach based on a

search-array data structure. This approach can be conducted more efficiently but still time consuming. For an image with 538×403 , this algorithm fills the removed region in 11.56 seconds(Figure 1). The left is input image. The middle is the result image in which the people has been removed. The right is the inpainting result in reference [16].



Fig. 1 The example-based inpainting method proposed by Kwok et al.[16].

In this paper, we propose an accurate and fast approach called photo cleaning which is an image editing tool and can automatic remove the redundant objects without any user interaction. Our approach uses two or more photos of the same scene as input, and produces a visual pleasure resulting photo as output. See Figure 2, the left and the middle photos are taken by camera and are inputted into our system. The right image is produced by our photo cleaning tool. It can be seen that our tool can automatically remove the redundant objects and produce a visual pleasure result.

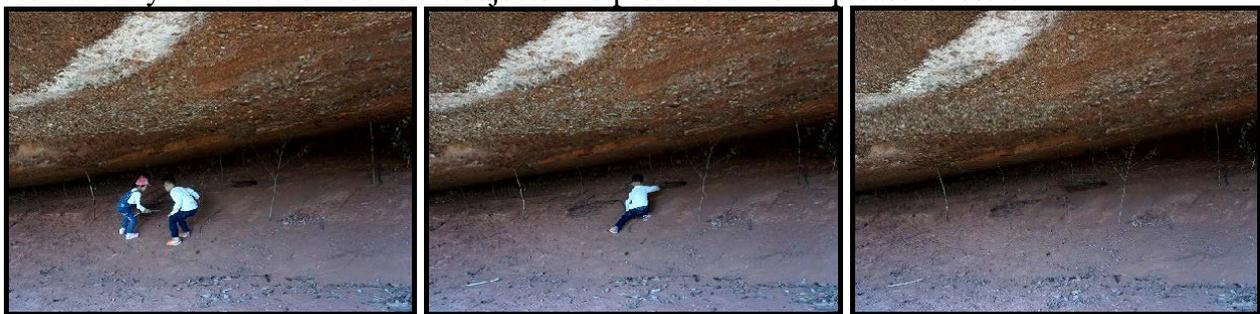


Fig. 2 The photo cleaning example.

Overview

The main goal of this paper is to provide user an automatic image editing tool which can remove redundant objects(Figure 2). The system has two features: a) the processing is automatic without user interaction, b) the output resulting images are clean enough.

Our main contributions are two-fold. For redundant object detection, we develop a simple but effective algorithm. Also, we propose a novel redundant object removal algorithm. The pipeline of our system is illustrated as follows (Figure 3).

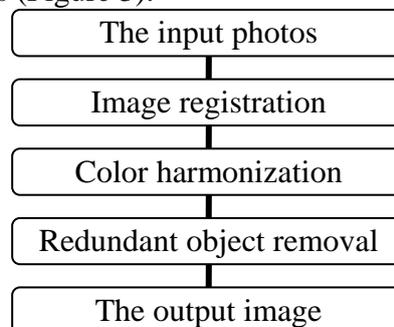


Fig.3 The pipeline of our system.

Image Registration

The input photos are not always taken from the exact angle since the camera shake. It is necessary to match the images before redundant object removal. In this paper, we adopt the image registration method proposed by Kaneko et al.[17].

Image registration is to search most similar parts of the input images. In this step, similarities between the input images are checked by scanning over the scenes. For instance, the input photo G and another input photo G' taken from the scene by user are encoded into corresponding binary images by the equation below.

$$b(x, y) = \begin{cases} 1 & (g(x+1, y) \geq g(x, y)) \\ 0 & (\text{otherwise}) \end{cases}$$

The increment signs are defined by horizontal comparison between pixels. Increment signs can be defined in many different ways through the combinations of the current pixel and its neighbors since any pixel in an image has eight neighbor pixels around it. The main algorithm named Increment Sign Correlation (ISC) is designed and proved to be very robust for brightness change and occlusion[17]. The effect of image registration relies on the information computed by ISC. Experiments show that the horizontal and vertical variations in brightness can be more effective for matching images in case of much noise.

Figure 4 is a registration example. The left and the middle are two photos taken by user. As you can see, these two photos are slightly different in angle. The right image is the registration result.



Fig. 4 An image registration example.

Color Harmonization

The goal of color harmonization is to adjust the color and brightness of the input photos so that they have the same hue. Because of the different shooting time, photos are usually with different color and brightness. The main idea of this color harmonization step is inspired by reference [18].

The method has three components: the input component, the processing component and the output component.

The main purpose of the input component is to convert the input photos into its corresponding binary patterns. This can be done by operating on the $\alpha\beta$ associative memories. The processing component is the core of the method and is consisted of two sub-components: an auto associative $\alpha\beta$ memory of type max and an auto associative $\alpha\beta$ memory of type min. The output component is a reverse procedure which converts the binary patterns resulting from the processing component into the corresponding images with specific color and brightness.

These three components act in two steps: the learning step and the pattern recall step.

Learning step.

- All input images are input into the system and the input component converts them into binary patterns.
- The associative $\alpha\beta$ memory of type max and the associative $\alpha\beta$ memory of type min do their respective learning step.

Pattern recall step.

- The associative $\alpha\beta$ memory of type max and the associative $\alpha\beta$ memory of type min produce two binary patterns.
- The output component converts these two binary patterns into the corresponding images.

Figure 5 illustrates an example of color harmonization. The left and the middle photo are taken from the same scene with different time and camera setting. Our color harmonization algorithm change the color and brightness of the middle photo into the style of the left one. The right image is the result. It shows that the right image has the same hue as the left photo.



Fig. 5 An example of color harmonization.

Redundant Object Removal

Our redundant object removal process consists of two steps. The redundant object detection and the hole filling steps.

Redundant object detection. Since the input photos are two or more, our efficient detection algorithm is designed by using a voting system. For describing convenience, we assume there are N input photos. First, all the N input photos are overlapping together. For each pixel's position $P(x,y)$, there's N pixels overlapped together and N RGB values are in this position. Then, for each pixel's position, the system reads each RGB value and counts the repetitive time of the same RGB value. Afterwards, each RGB value of each pixel's position has its own score. We consider the RGB value with the highest score as the background pixel's color and the other RGB values as the color of redundant objects. The system repeats this for all pixels' positions. This strategy is based on the following observation. When we take a photo of some scenes with many tourists which are not supposed to be there, they are moving all the time. The background objects are stable and still.

Holes filling. After redundant objects are detected, the system use the RGB value with highest score as the new pixel's color. The result is an image without redundant objects created by our system.

Figure 6 illustrates an example of redundant object removal process. The left and the middle photos are input photos. The right image is created by our system. It is shown that, some people are walking on the beach and some people are swimming in the water. Our system identifies them as redundant objects and removes them. The resulting image is clean enough and matches the human visual pleasure. That's why we call this tool "photo cleaning".

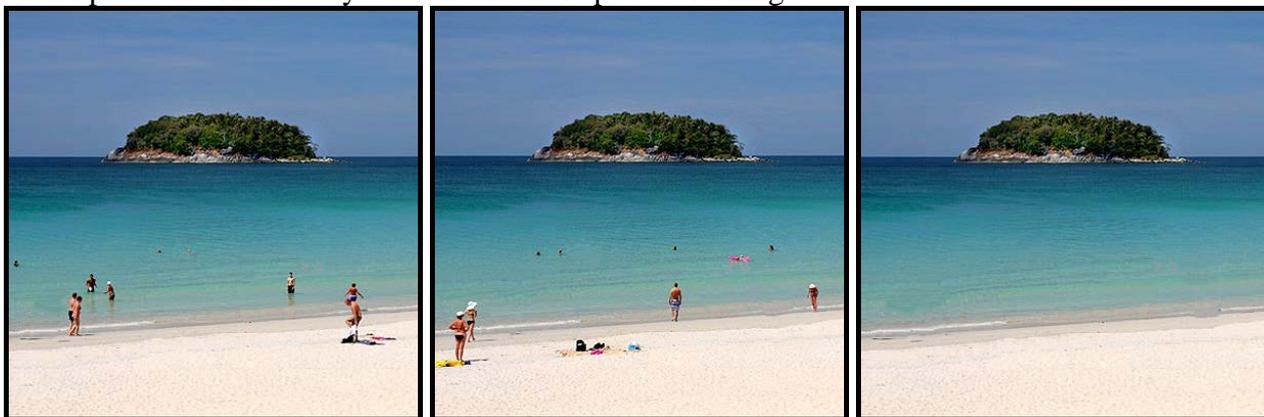


Fig. 6 An example of redundant object removal process.

Experimental Results

A large number of experiments have been done to test our photo cleaning tool. In most cases, our photo cleaning tool can get good results. Figure 7 is an example. The left and the middle photos are the inputs. The right image is the output of our tool. In this example, though the input photos have different camera settings, our system can handle it well. Figure 8 is another example. The first three photos are taken from a video stream. As you can see, someone are walking in the video and our system cleans this moving object. The output is a clean image with human visual pleasure.

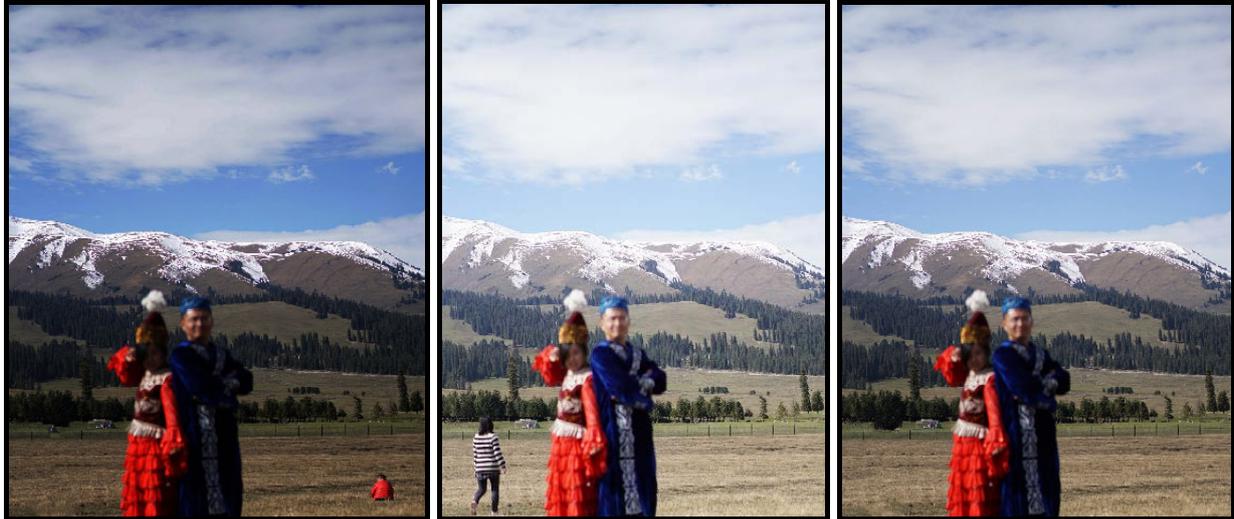


Fig. 7 The photo cleaning example.



Fig. 8 The photo cleaning example.

Summary

This paper proposes an automatic image editing tool which can remove the redundant objects and create a clean enough image. For most cases, our photo cleaning tool works well and the output image matches the human visual pleasure.

However, our system fails in some cases. For example, the redundant object which the user wants to remove stay still. Our system treats it as background and fails to clean it. In the future, we attempt to integrate user interaction interface to our system so that the user can remove any object by specifying it.

Acknowledgements

The work in this paper has been supported by Key Project of University Excellent Youth Scholars of Anhui Province (gxyqZD2016303).

References

- [1] Irani M, Anandan P, Hsu S. Mosaic based representations of video sequences and their applications[C]// International Conference on Computer Vision, 1995. Proceedings. 1995:605-611.

- [2] Agarwala A, Dontcheva M, Agarwala M. et al. Interactive digital photomontage[J]. *Acm Transactions on Graphics*, 2004, 23(3):294–302.
- [3] Snavely N, Seitz S M, Szeliski R. Photo tourism: exploring photo collections in 3D[J]. *Acm Transactions on Graphics*, 2006, 25(3):835-846.
- [4] Criminisi A, Prez P, Toyama K. Object Removal by Exemplar-Based Inpainting[C]// *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*. 2003:721-728.
- [5] Wexler Y, Shechtman E, Irani M. Space-time video completion[C]// *Computer Vision and Pattern Recognition*, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. 2004:I-120-I-127 Vol.1.
- [6] Wilczkowiak M, Brostow G J, Tordoff B, et al. Hole Filling Through Photomontage[C]// *British Machine Vision Conference 2005*, Oxford, Uk, September. 2005.
- [7] Lundbæk K, Malmros R, Mogensen E F. Image Completion Using Global Optimization[C]// *Computer Vision and Pattern Recognition*, 2006 IEEE Computer Society Conference on. 2006:442-452.
- [8] Efros A A, Leung T K. Texture syntesis by non-parametric sampling[C]// *The Proceedings of the Seventh IEEE International Conference on Computer Vision*. IEEE, 1999:1033.
- [9] Efros A A, Freeman W T. Image quilting for texture synthesis and transfer[C]// *Conference on Computer Graphics and Interactive Techniques*. ACM, 2010:341-346.
- [10] Kwatra V, Sch&#, Dl A, et al. Graphcut textures: image and video synthesis using graph cuts[J]. *Acm Transactions on Graphics*, 2003, 22(3):277-286.
- [11] Kwatra V, Essa I, Bobick A, et al. Texture optimization for example-based synthesis[J]. *Acm Transactions on Graphics*, 2005, 24(3):795-802.
- [12] Criminisi A, Prez P, Toyama K. Object Removal by Exemplar-Based Inpainting[C]// *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*. 2003:721-728.
- [13] Sun J, Yuan L, Jia J, et al. Image completion with structure propagation[J]. *Acm Transactions on Graphics*, 2005, 24(3):861-868.
- [14] Fang C W, Lien J J. Rapid image completion system using multiresolution patch-based directional and nondirectional approaches.[J]. *IEEE Transactions on Image Processing*, 2009, 18(12):2769-2779.
- [15] Barnes C, Shechtman E, Finkelstein A, et al. PatchMatch: a randomized correspondence algorithm for structural image editing[J]. *Acm Transactions on Graphics*, 2009, 28(3):341-352.
- [16] Kwok T H, Sheung H, Wang C C L. Fast query for exemplar-based image completion[J]. *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, 2010, 19(12):3106-15.
- [17] Kaneko S, Murase I, Igarashi S. Robust image registration by increment sign correlation[J]. *Pattern Recognition*, 2010, 35(10):2223-2234.
- [18] Yáñez C, Felipe-Riveron E, López-Yáñez I, et al. A Novel Approach to Automatic Color Matching[J]. *Lecture Notes in Computer Science*, 2006, 4225:529-538.