Image Inpainting Software

Yesheng Ma, Hu Hu, Yikai Zou
Shanghai Jiao Tong University
{kimi.ysma, mihawkhu}@gmail.com, zouyikai1014@163.com

Abstract

Image inpainting is the technique to modifiy an image where the inpainted area is unpredictable. There are already numerous methods introduced by former researchers to handle image inpainting including partial differential equations and Markov random fields. These methods all root in smoothing and have a immediate drawback of blurring.

In this paper, we look deep into another kind of inpainting technique: examplar-based inpainting algorithm and further propose a method inspired by the idea of priority queue. When we inpaint an image, we give different priority to the patches around the area to be inpainted and in this way we inpaint patches that capture the structure of the image better.

Our experiments show that the idea of priority queue works well and is suitable for inpainting images with complex structure.

***Index terms* —** Image Inpainting, Priority, Patch-Based Method

1. Introduction

Image inpainting is the technique of modifying an image in an undetectable form. Some of the common cases image inpainting are: (i) the restoration of old photographs and damaged films (ii) the removal of unwanted text from an image (iii) the removal of an object in purpose. The existence of the image inpainting technique dates back to the beginning of human art and we can see that the application of image inpainting by computer saves a lot of time compared with hand-carfted image inpaiting in museums.

There are alreay several researches on this topic. One kind of technique is based on smoothing, *i.e.* we inpaint that image based on the color of nearby pixels. This method has an intrinsic problem of blurring since the color chosed to inpaint the missing area are all based on the smoothing of nearby colors. Another approach is kind of patch-based image inpainting, where we choose a small patch around the edge of target area and propagate it into target area. In this method, in contrast to the smoothing-based algorithm, can better capture the texture of the image and will not lead to the problem of blurring.

With regard to the patch-based method, every patch around the target region are assigned a priority. When we propagate the patch into target region, the patch with highest priority is chosed. In this paper, we propose an efficient priority assginment strategy that can better capture the structure of the whole image. Kimi: TODO: add illustration of our proposal

2. Related work and our contribution

First we should make it clear that image inpainting and image denoising are actually different tasks. In image inpainting, we aim to remove part of an image and a person needs to explicitly specify the area to be inpainted. In constrast, as regard to image denoising, the task is to remove the "random" noise of an image, which often does not need human supervison.

2.1. Related work

There are mainly two kinds of works on image inpainting: smoothing-based and exemplar-based.

The research on image inpainting emerges on 2000, Marcelo Bertalmio *et al.* [1] first deals with the problem of image inpainting. Their method is mainly based on intuition: propagate color information in the direction of normal. However the limitation of the algorithm is that the texture of its environment is unlikely to be reproduced.

Roth and Black [6] later come up with a more general framework called *Field of Experts* which is a Markov random field model. This work relies on the result of Hinton [5], where Hinton discovers that a factor in MRF can be modeled by a field of "expert" distributions. In this work, they first learn the model and then apply the learned model to Bertalmio's propagation method. Although this work in some sense captures image structure, the blurring problem is also serious after our experiment.

Another different way to tackle this problem is proposed by Criminisi *et al.* [2, 3]. They note that exemplar-based texture synthesis contains the essential process requred to replicate the structure of image. They introduce a priority for each exemplar and propagate according to the priority of each exemplar. This technique does not have the problem of blurring and can restore texture well, but may still fail for large object removal.

2.2. Our contribution

add our contribution

3. Proposed methodology

According to the discussion above, although there are existed work about the exemplar-based image inpainting algorithm *et al*. [2, 3]. But previous attach a little too much calculation work to do when inpainting, it will result in long time to wait. Based on the previous work about exemplar-based image inpainting algorithm, we proposed a new evaluation criteria about the priority of each patch, which costs much less time than previous work and get good inpainted result.

3.1. Two key observations

A. Basic process unit size is important

When select the exemplar region to inpaint the target region, the basic process unit size is important. The size of basic will effect the final result. Previous work has applied pixel as the basic unit[4], the result will result in smoothing and blurring. It's easy to understand because small unit size ignore the connection between the each part of the image. Hence, we should not choose too small size of basic process unit.

However, when the size of basic unit is too large, the target region inpainting will lead in to much difference. The large region of the mask region will get too much influence by the known region. That is to say, the target region will result in strange margin and line. So we should not choose too large basic process unit.

In our experiment, we test different unit size and compare them. Due to the difference of the number of calculation, process speed of each unit size is much different. Consider several factors, this algorithm present good results when the basic process size is $3 * 3$ or $5 * 5$. Smaller size will result in smoothing and blurring, and larger size will result in unexpected disruption of target region. And in this paper, we will use patch as the name of the basic process unit.

B. Inpainting order is important

When the image is inpainted by exemplar-based algorithm, the final result depends on the inpainting order selected by the algorithm. Some previous works randomly select the patch to inpaint, the result is worse than some previous works which select specific patch to do inpainting

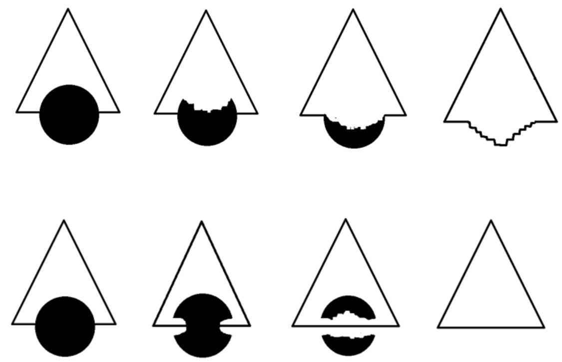

Figure 1. **The importance of the inpainting order**. The first group of images is the process of being inpainted from the up to the bottom, the second group of images is the process of being inpainted from the specific order.

et al. [7]. This section will demonstrate that the quality of the output image synthesis is highly influenced by the order of the image inpainting.

In order to understand the importance of the inpainting order, a comparison between the specific order and fixed order is shown in $Figure.1$. the first group of images is using the up-to-bottom patch selection. Due to the target patch is always influenced by the nearby patch, so the bottom horizontal line of the triangle has not been inpainted. But if we do inpainting from the intersection of the target region and the bottom horizontal line, just as is shown in the second group of images. The horizontal line will be inpainted much better.

Therefore, a better image inpainting algorithm would be one that gives higher priority of synthesis to those regions of the target region which lie on the continuation of image structures, as shown in the second group of $Figure.1$.

3.2. Our proposed method

According to the two key observations mentioned above, the selection of patch that do inpainting first effects the final result much. So we should carefully select the specific patch.

Based on the image itself, in order to restore the original image, we should start from the intersection of the real line in image and the margin of the mask. So we use the dot cross result of the gradient and the unit vector orthogonal to the margin of the target region. Given a patch Φ_p centred at the point p for some $p \in \Omega$ (see $Figure.2.$), we define its priority $P(p)$ as the product of two terms:

$$P(p) = C(p) * D(P) * A(p)$$

Where $C(p)$ is the confidence term of the patch p, $D(p)$ is the data term of the patch p, $E(p)$ is the evaluation term, which records the mask region in the target patch.

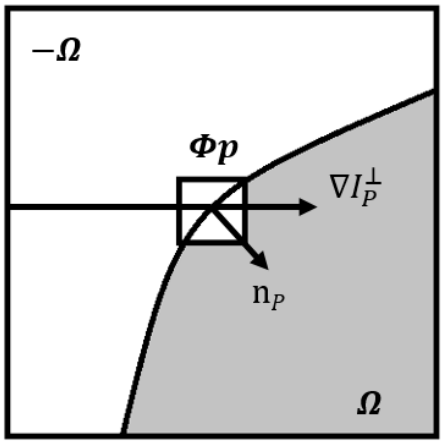

Figure 2. **Notation figure**. The mask region and and the know region. The gradient and the unit vector is shown in the figure

Confidence term represents the amount of reliable information surrounding the pixel p. The intention is to fill first those patches which have more of their pixels already inpainted, with additional preference given to pixels that were inpainted early on (or that were never part of the target region), it is calculated as follow:

$$C(p) = \frac{\sum_{q \in \Phi_p \cap (-\Omega)} *C(q)}{|\Phi_p|}$$

Where Φ_p is the patch region, $|\Phi_p|$ is the area of the patch region. Ω is the target region which is under the mask. $-\Omega$ is the region except for target region, it's the known region. During initialization, the confidence value is set to 1 in known region and 0 in mask region.

Data term represents the real line or margin information in image. This factor is of fundamental importance in our algorithm because it encourages linear structures to be synthesized first. It is the dot cross of the gradient and the unit vector orthogonal to the margin of the target region. It is calculated as follow:

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

Where $\nabla I_p^{\perp}$ is the gradient of image in pix p. n_p is the unit vector orthogonal to the margin of the target region.

Evaluation term represents the weight of each pixel in the target patch, which records the area of mask region in the target patch. It is calculated as follow:

$$E(p) = \frac{1}{\sum_{q \in \Phi_p \cap \Omega} q}$$

Where Φ_p is the patch region, Ω is the target region.

After defined the priority function of each patch, out algorithm work as the following steps.

First, given the input image and the input mask, initialize the whole patch in the margin with confidence term, data term and evaluation term.

Next, compute the whole priority of each patch in mask margin, choose the maximum priority patch.

Then, traverse the patch that is near the target patch, compute the Euclidean distance of each pair of patch, find the exemplar patch minimizing the Euclidean distance.

Finally, copy the exemplar patch to the target patch where the pixel is unknown. And repeat these steps until there is no mask in the input image.

4. Result and analysis

References

[1] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In J. R. Brown and K. Akeley, editors, *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, New Orleans, LA, USA, July 23-28, 2000*, pages 417–424. ACM, 2000.

[2] A. Criminisi, P. Pérez, and K. Toyama. Object removal by exemplar-based inpainting. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pages 721–728. IEEE Computer Society, 2003.

[3] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Processing*, 13(9):1200–1212, 2004.

[4] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038, 1999.

[5] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[6] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 860–867. IEEE Computer Society, 2005.

[7] Y. Wei and S. Liu. Domain-based structure-aware image inpainting. *Signal, Image and Video Processing*, 10(5):911–919, 2016.