International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878, Impact Factor (2015): 3.791

Exemplar Method Based Image Inpainting

Arsha Haridas[1], Shubham Agrawal[2]

Institute of Engineering, MET's Bhujbal Knowledge City, Savitribai Phule Pune University

Abstract: ***Image Inpainting, (generally known as image completion), is the technique to fill holes in an image. The challenge of image inpainting is how to fill the hole in a visually plausible way. The success of structure propagation is highly dependent on the order in which the filling proceeds. Hence, the authors have proposed a computationally efficient algorithm by a patch-based sampling process. Although the algorithm propagates the structure well and produces some amazing results, it still has difficulties when inpainting images where complex salient structures exist in the missing regions. Therefore, the user is allowed to manually specify the important missing structure information by extending a few curves or line segments from the known to the unknown regions. The curves or line segments would be treated as the constraints, and then the structure propagation is formulated as a global optimization problem.***

Keywords: Exemplar based, Image Inpainting, Structure Propagation, Onion Peeling Method

1. Introduction

Removing objects and repairing damaged regions are somewhat a tedious task. Image inpainting is a technique for removing undesired objects in images and reconstructing the missing regions in a visually appealing way. There have been many research works for the same and these works are classified into two major categories. One is non-exemplar based method and the other is exemplar based method. We are focusing on the exemplar based inpainting algorithm.

The exemplar based inpainting approach propagates the image information from the known region into the missing region at the patch level. Exemplar-based methods, which have been successful in problems such as de-noising and in super resolution, have also yielded good results for texture synthesis and inpainting. The usual approach to exemplar based inpainting is to progressively fill in blocks on the boundary of the inpainting region using matching blocks in the known region of the same image.

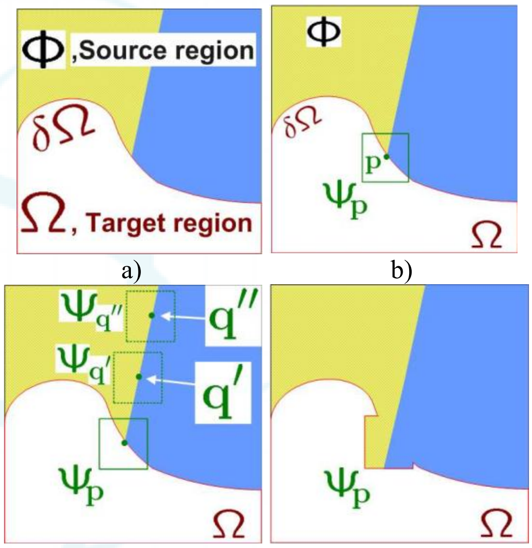
Exemplar based image inpainting algorithms are able to inpaint even for large regions and as well as natural scene images which have complex textures and structures. In this paper we have proposed an efficient exemplar based image inpainting algorithm with an improved priority term that defines the filling order of patches in the image. The analysis of both theoretical and experimental results of exemplar based algorithms provides a good framework for us to extend our contribution to this category. This idea stems from the texture synthesis technique proposed in which the texture is synthesized by sampling the best match patch from the known region.

2. Exemplar based inpainting

There are two key ideas in exemplar-based inpainting, the first is the exemplar-based synthesis and the second is the automatically determined filling order.

2.1 Exemplar Based Synthesis

Figure 1 illustrates the steps in exemplar-based synthesis. Ω is the target region or the region that should be inpainted. Φ is the source region, which means the region which should not be inpainted or be filled already. $\delta\Omega$ is the contour between the source region and the target region.



a) b) c) d)

Figure 1: Inpainting result from user's hints of structure information

In each step, there is a point position along $\delta\Phi$ chosen as the patch centre by some method. And the corresponding patch (target patch, Figure 1 b)) of this patch, centre will be inpainted.

The machine searches for all of the patch candidates from the source patch as shown in Figure 1c), and finds the one with the minimum SSD in the overlapping area (the intersection of the target patch and the source region). Finally, the machine paste the found patch to the target patch as shown in Figure 1(d).

There are two advantages of this synthesis method. First, synthesizing patch by patch performs much faster than synthesizing point by point. Second, it often shows the texture can be propagated better by patch-based filling.

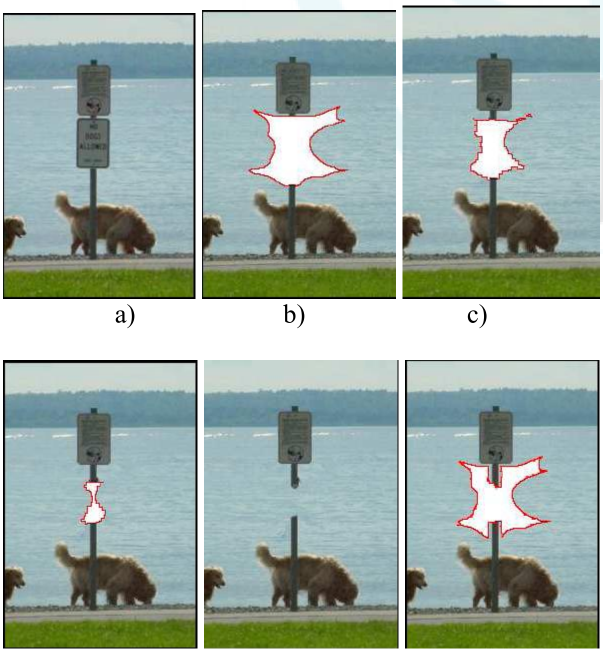
2.2 Filling Order

In [1], the authors found the filling order plays an important role in the inpainting job. Figure 2 shows one of the examples. Figure 2(a) is the original image. As can be seen, there are two signs on the bar. If we want to remove the lower sign and inpaint it as shown in Figure 2(b), we can use either onion peeling or structure-guided method to determine the filling order. Figs. 2(c) - 2(e) shows the results generated by using the onion peeling method, and Figs. 2(f) - 2(i) shows the results generated by using the structure-based method proposed by the authors. From Figs. 2(e) and 2(i), we know that we should inpaint the region which is close to the existing structure first so that the structure can propagate into the target region well.

However, it is not easy for the computer to know where the structure is. Therefore, the authors defined the structure according to the edge information which can be extracted easily from any image. Their idea can be explained by Figure 3. Suppose $\nabla I_p^{\perp}$ is the isophote (direction and intensity) at point p, np is the normal to the contour, and Φp is the patch, they defined the data term $D(p)$ as

$$D(p) = \frac{|\nabla I_p^{\perp} \cdot n_p|}{|\Phi_p|} \tag{1}$$

The data term is used to describe the strength of the isophote hitting the front. By using this term, they can encourage the machine to synthesize the linear structure first and therefore propagate securely to the target region. They also have defined a confidence term *C(p)* to describe the amount of reliable information surrounding pixel *p*. However, we feel it is somewhat heuristic and is not as important as the data term *D(p)*. Therefore, we decided to skip the description of the confidence term here. All we have to know is that they use the confidence term and the data term together to determine the filling order.



a) b) c)

d) e) f)

g) h) i)

Figure 2: The figure shows the importance of the filling order. Figure 2(a) is the original image. Figure 2(b) shows the target region with a red boundary. Figs. 2(c) - 2(e) are the results generated by using the onion peeling order. Figs. 2(f) - 2(i) are the results generated by using the structure-guided method proposed in [1].

2.3 Implementation

The most difficult implementation part is the data term, especially the computation of normal np. Therefore, we spent some effort describing some more details about the computation of normal *n(p)* here. In order to compute n*p*, there are mainly two possible ways. The first is finding the preceding and successive points and calculating the vector orthogonal to the line through these two points. This is also the method recommended by the authors in [1]. However, we felt the computation power of this method is too high, because we must find the contour at each iteration. Therefore, we proposed another method to estimate n*p*. The method is as follows. First, we have a mask that tells whether a pixel belongs to the source region (with value 255) or the target region (with value 0). And then we filter this mask in two directions. From the gradient, we can obtain the contour's normal direction. To make the results more reliable or more robust to the contour shape, we smooth the gradient by a Gaussian filter before we take its value.

3. Structure Propagation Inpainting

3.1 Algorithm

The structure propagation problem is modeled as a Markov Random Field (MRF) energy minimization problem [3]. As shown in figure 4, I is the input image region, Ω is the unknown region, and *C* is a user-specified curve.

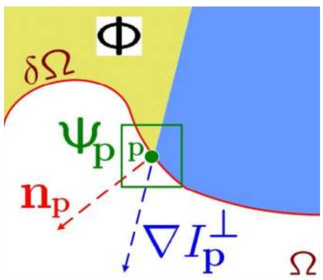


Figure 3: Notation diagram. Given the patch Ψ_p, n_p is the normal to the contour Ω of the target region and $\nabla I_p^{\perp}$ is the isophote at point p. The entire image is denoted with *I* (referenced from [1]).

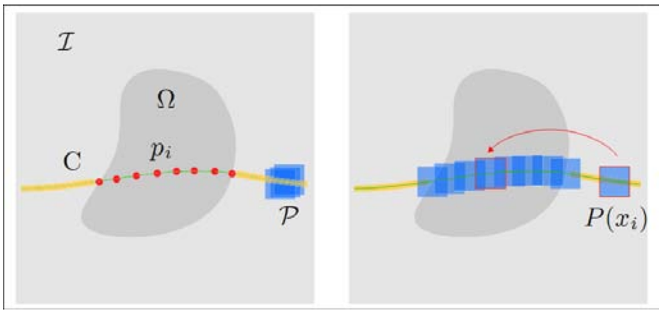


Figure 4: Structure Propagation chain

We first sample some anchor points on curve *C*, the distance between anchor points is quarter of the patch size, which is specified by user. Structure propagation uses sample set *P* to synthesis image patches for these anchor points. *P* consists of all patches whose centers are within a narrow band (we set it as half of the patch size wide here) along curve *C*. We can think of anchor points as nodes and patchs in *P* as labels. Finding the most suitable patch for each anchor point can be modeled as a MRF problem. The energy terms are defined as follows:

$$E(X) = \sum_{i \in V} E_1(x_i) + \sum_{(i,j) \in \varepsilon} E_2(x_i, x_j) \quad (2)$$

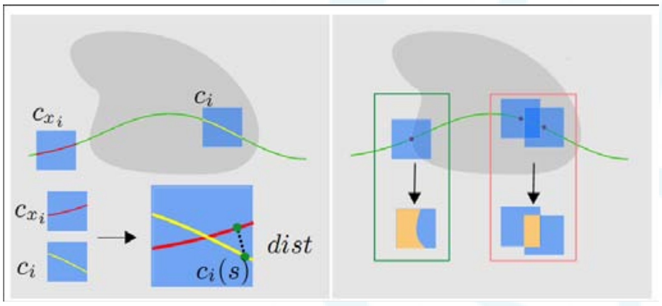


Figure 5: Energy terms for structure propagation

where,

$$E_1(x_i) = k_s \cdot E_s(x_i) + k_i \cdot E_1(x_i) \quad (3)$$

$E_s(x_i)$, $E_1(x_i)$, and $E_2(x_i;x_j)$ are energy terms for structure, completion, and coherence constraints, respectively. k_s, k_i are weighting factors, which are specified by user. As illustrate in figure 5, $E_s(x_i)$ evaluates the structure similarity between the curves in the source and target patch. The structure similarity is defined as:

$$E_s(x_i) = d(c_i, c_{x_i}) + d(c_{x_i}, c_i) \quad (4)$$

where,

$$d(c_i, c_{x_i}) = \sum_s \| dist(c_i(s), c_{x_i}) \|^2 \quad (5)$$

$E_1(x_i)$ is defined as the sum of the squared difference between the overlapping region of the input image and the unknown region. It makes the synthesized structure matches the input image at boundary of the unknown region.

$E_2(x_i;x_j)$ evaluates the coherence between two synthesized patches of adjacent anchor points. It constraints the synthesized structure to be smooth.

3.2 Implementation

The implementation of structure propagation algorithm is straightforward, except the data cost term $E_s(x_i)$.

$E_s(x_i)$ evaluates the distance between two curves in different patches, which is defined in equation 5. To evaluate $d(c_i, c_{x_i})$ in equation 5, we build a block of the same size as the patch size used for inpainting. This block is used to store the distance to c_{xi} at each position in the block. Firstly, we set the value of the positions that curve c_{xi} passes through as zero and leave other positions with empty value. Then we iteratively expand positions with nonempty values by setting the value of their neighbors as their value plus one. After the filling the whole block with values, evaluation of the distance to c_{xi} is completed. $d(c_i, c_{x_i})$ can be obtained by visiting the positions along the curve c_i and sum up all the values.

4. Experimental Results

There is a fence in Figure 8 that blocks the beautiful scene behind it. Therefore, we apply the [1] to inpaint the fence region. We get a very promising result. It is hard to believe from the result image that there was a fence existing.

a) b) c)

Figure 6: a) Original image b) Users' Hints c) Result

5. Conclusions

In this project, we implemented two inpainting algorithms. The first one is exemplar-based inpainting, and the second one is an extension to the first, which allows user to manually specify the structure information of the region to be inpainted. The algorithms are rather robust in the sense that the parameters, such as weighting factors for the data term in structure propagation and patch size, are insensitive. We do not have to spend much effort on tuning parameters but still can get good results.

Generally speaking, current inpainting algorithms work well when region to be inpainted has complex texture, such as grass, cloud, and waves, or when object to be inpainted is natural. However, they may fail when we try to inpaint artificial things, like human bodies and faces. Also, these inpainting algorithms have their limitations.

The most common problem is that when there is no suitable patch in the same image, the algorithms may not be able to produce promising result. One possible solution is to develop an algorithm that can automatically retrieve suitable sample images from the Internet, and complete inpainting with such samples.

References

[1] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, vol. 2, p. 721, 2003.
[2] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," ACM Trans. Graph., vol. 24, pp. 861–868, July.
[3] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, no. 6, pp. 1068–1080, 2008.
[4] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, pp. 1222–1239, November 2001