A Multi-Scale Gradient based Method for Image Completion

LU Rui XU De LI Bing
Institute of Computer Science and Engineering
Beijing Jiaotong University
Beijing,100044
CHINA

Abstract:- Image completion is a challenging problem in computer graphics and computer vision. It refers to the problem of automatically filling holes in images left by the unwanted objects removal in a plausible way. Unlike previous methods we incorporate gradient components to get a global effect by performing local operations. At the same time in order to speed up convergence and further enforce global consistency, we use multiple-scale solution through a gauss pyramid. In the end Experiment results show promising performances compared with most of the existing algorithms.
Key Word:-Image Completion Image Inpainting Multi-Scale Solution Texture Synthesis Gradient Domain

1 Introduction

Image completion is a challenging problem in computer graphics and computer vision. It refers to the problem of automatically filling holes (missing parts) in images left by unwanted objects removal in a plausible way. It can be described as follows: Given an input image I with an unknown region Ω, the task of image completion is to propagate structure and texture information from the available parts $I-\Omega$. Obviously, image completion is inherently an under-constrained problem [1].

The goal of image completion is close to texture synthesis and image inpainting. Texture synthesis is used to extend and fill regular fronto-parallel image textures. There already exist many texture synthesis methods, some of which are based on MGRF model, while some others are based on nonparametric sampling, such as [2][3][4]. Although these methods can remain small texture structures in the image, their drawback lies in that they can't remain linear structure.

Image inpainting is similar to image completion. The only difference lies in their goals. The goal of image inpainting is to restore small, smooth and non-textured regions while the goal of image completion is much wider. In general the work in this area can be classified into two groups. One kind is PDE based methods such as [5] [6]. This kind of methods fills the missing region pixel by pixel. PDE based methods can preserve linear structure. When the missing region is small it can obtain smooth result, or the result will be blurred. The other is texture synthesis based methods. The most representative method is proposed by Bertalmio et al.[7].

Many image completion methods have been proposed at present. Criminisi et al. [8] proposed an exemplar-based image completion algorithm, in which the filling order was determined by the angle between the isophote direction and the normal direction of the local filling front, so that the missing region with stronger structures could be filled in higher priority. This paper is just inspired by their work. Sun et al. [1] introduced a novel structure propagation approach to image completion. In their algorithm, the user manually specifies important missing structures by extending a few curves or line segments from a known region to the unknown. But there is a limitation in their algorithm, that is, it works well only when the missing salient structures can be represented by a set of simple curves. More recently, J.b. Shen et al [10] reconstructed the image by solving Poisson Equation. Since there is a equation for every pixel in the hole, this method requires large storage and it is time consuming.

The remainder of this paper is organized as follows: Section 2 describes our method in detail. Section 3 presents some examples to demonstrate that our method is reasonable. The last section makes conclusions and discusses our future work.

2 Image Completion

2.1 Algorithm Overview

The input of our algorithm is the image *I*, original image with the object to be removed. To make it simple, in preprocessing step, we manually remove the object to get the hole *Ω* in the image *I*. The output is the image *I`* with the hole filled in a plausible way.

In order to make it easy to understand, we follow the symbols as used in other literatures. The region to be filled, i.e., the target region is indicated by *Ω*, and its contour is denoted as $\partial\Omega$. The contour evolves inward as the algorithm progresses, here so we also refer to it as the “fill front”. The source region *Φ* which remains fixed throughout the algorithm, provides samples used in the filling process. In this algorithm, we make full use of source region to fill patches in the target region.

See Table 1 for the completion process in detail.

Table1 Image completion process

Input: image *I*
Output: completed image $I^{'}$
Algorithm:
Compute gauss pyramid
For each level of the pyramid from coarse to fine
Repeat until the hole is filled
Identify filling front $\partial\Omega$
Select a target patch φ_t with its center $p \in \partial\Omega$, which is with the highest filling priority $P(p)$;
Search *Φ* to determine a source patch φ_s, which shows the highest similarity between φ_s and φ_t;
Copy image data from φ_s to φ_t;
Update confidence map of the pixels in the hole;
Update Ω and $\partial\Omega$;
Propagate solution to next level;
End For

2.2 Similarity Measurement

At the heart of the algorithm is a well suited similarity measure between target patch and source patches. A good measure needs to agree perceptually with a human observer. The Sum of Squared Differences (SSD) of color information, which is widely used for image completion, does not suffice (regardless of the choice of color pace) [11]. The main reason lies in that the human eye is very sensitive to variation. Thus maintaining variation continuity is much more important.

From Fig.1 below, we can clearly find that very different variation along *x* axis can lead to the same SSD score. The function *f(x)* has a noticeable temporal change. Yet, its SSD score relative to a similar looking function *g(x)* of is the same as the SSD score of *f(x)* with a flat function *h(x)*:

$$\int (f-g)^2 dx = \int (f-h)^2 dx \qquad (1)$$

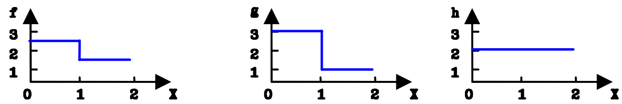


Fig.1. variation along *x* axis

However, perceptually, *f(x)* and *g(x)* are more similar, as they both encode a horizontal change. We would like to incorporate this into our algorithm in order to create a similarity measure that agrees with perceptual similarity. Therefore, we add a measure which is similar to that of normal-flow to obtain a quick and rough approximation of the color variation information. Thus at each point, we compute the spatial derivatives, that is gradient. Thus we obtain a three components representation of each point (c, g_x, g_y), where c, g_x, g_y are vectors with three items individually. Like the other methods we apply a L_2 norm to measure similarity of the patches.

$$\varphi_s = \arg\min_{\varphi_t \in \phi} d\left(\varphi_s, \varphi_t\right) \qquad (2)$$

where φ_s is source patch in *Φ* and φ_t is target patch centered at $p \in \partial\Omega$ .After finding the most similar patch in the source region *Φ*, we use it to fill corresponding pixel in $\varphi_t \cap \Omega$, including color and gradients, at the same time don’t forget to update confidence term and data term.

2.3 Multi-scale Solution

To further enforce global consistency and to speed up convergence, we perform the iterative process in multiple scales using Gauss pyramids. Each pyramid level contains half the resolution of the level above. The filling process starts at the coarsest pyramid level and the solution is propagated to finer levels for further refinement. The computational cost of using a pyramid is almost negligible [11].

In this algorithm we build two gauss pyramids: one is for original image and the other one is for image with object removal. In our experiments we usually build a three level gauss pyramid: fine level is original image; middle level is half the resolution of fine level in each dimension and coarse level is half resolution of middle level in each dimension.

Let p_i denotes a pixel in the hole in level i of the gauss pyramid, α_i denotes corresponding filling confidence, then

$$p_{i+1} = \alpha_i p_i + \alpha_{i+1} p_{i+1} \quad (3)$$

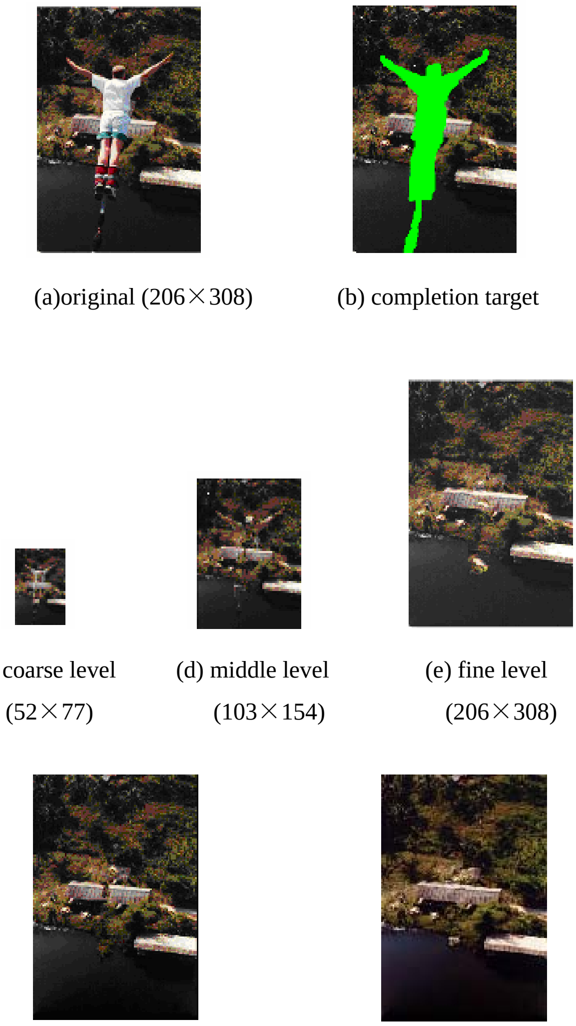
where p_i is the result of level i, p_{i+1} in the right hand of the equation is the result of filling the hole in level $i+1$ of the pyramid, p_{i+1} in the left hand is the final result of level $i+1$, and then propagate it to the next finer level.

During the process, we just use the coarser level result to refine the result of current level. Thus we also need to fill the hole of the current level. What's more, current level result plays an important role in final result of current level. As a result you can make a rule: if α_{i+1} is smaller than a given value, we use the coarser level to complement current level result, or just current level is used.

3 Experimental Results

Our algorithm has been applied to a variety of colorful photographs with complex background structures. As the purpose of image completion is to fill damaged areas while satisfying visual perception, it is commonly accepted that the quality of results is detected by the human perception of the appearance in completed images. The experimental results demonstrate that our approach can generate satisfactory results. All the examples shown in this section are tested on a PC with Pentium IV 3.0GHz CPU + 1GB RAM. Similar to other patch-filling based image completion approaches (e.g., [8]), we employ patches with 9×9 pixels in all our examples.

In the first example demonstrated here, the image size is 206×308, while there are about 7997 pixels in the hole. Our method needs about 52.558seconds while [8] needs about 48.372seconds.But from the figure, we can see our method could maintain the linear structure well while there is a big gap in the result of [8]. So it's worthy to use a little more time to get a better result.The result of [12] is directly taken from the paper. In the second example we remove the lady and then fill the hole. Our result is much more plausible compared with the result of [8], as is shown in the upper part of Fig.3 below. In the lower part of Fig3, another example is demonstrated, in which nearly 80,000 pixels are to be filled. The result is also given in Fig.3 below.



(a)original (206×308) (b) completion target

(c) coarse level (52×77) (d) middle level (103×154) (e) fine level (206×308)

(f) result of [8] (g) result of [12]

Fig.2. Example of completion processing

4 Conclusions

This paper presents a novel algorithm for filling holes which is generated by the removal of unwanted objects or missing information from digital photographs. In our algorithm we can replace the hole by a visually plausible background that mimics the appearance of the source region. During our implementation process, we try to find a patch in the source region that is similar with target patch not only in color but also in color variation. Thus our method can produce a more plausible result even with complex background. Although the final result may be a bit slower than [8], due to the multi-scale solution, it indeed can maintain image structure feature very well. In the future we will try to extend our method to image completion of several discrete images or video completion.

(a) original (b) completion target

(c) our result (d) result of [8]

(e) original (by Jianbing Shen) (f) completion target

(g) our result

Fig.3. Other examples of completion processing

References:

[1] J. Sun, L. Yuan, J. Jia, HY. Shum, Image completion with structure propagation, ACM Transactions on Graphics, Vol.24, No.3, 2005, pp.861-868.

[2] A.A. Efros, T.K. Leung, Texture synthesis by non-parametric sampling, Proceedings of International Conference on Computer Vision, Vol.2, 1999, pp. 633-648

[3] A.A. Efros，T.K. Leung, Image quilting for texture synthesis and transfer, Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001, pp.341-346

[4] N. Kornodakis, G. Tziritas, Image completion using global optimization, IEEE Conference on Computer vision and pattern recognition, Vol. 1, 17-22, 2006, pp.442-452

[5] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, Proc. ACM Conf. Comp. Graphics (SIGGRAPH), 2000, pp: 417-424.

[6] T.F. Chan, S. H. Kang, J. Shen, Euler's elastica and curvature based inpainting, SIAM Journal of Appl. Math,vol. 2, no. 63, 2002, pp. 564-592.

[7] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, Simultaneous structure and texture image inpainting, IEEE Transactions on Image Processing, Vol.2, NO.8, 2003, pp.882-889

[8] A. Criminisi, P. Perez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Transactions on Image Processing, Vol.13, No.9, 2004, pp.1200-1212

[9] J. Jia, C. K. Tang, Image repairing: robust image synthesis by adaptive ND tensor voting, IEEE Conference on Computer Vision and Pattern Recognition, Vol.1, 2003, pp.1643-1650

[10] J.b. Shen, X.g. Jin, Ch. Zhou, Charlie. Wang, Gradient based image completion by solving the poisson equation, Elsevier Science on Computers & Graphics, Vol.31, No.1,2007, pp.119-126

[11] Y. Wexler, E. Shechtman, M. Irani, Space-time completion of video, IEEE Transaction on Pattern analysis and machine intelligence, Vol.29, No.3, 2007, pp.463-476.