ABSTRACT

Image inpainting or completion is a technique to restore a damaged image. Recently various approaches have been proposed. In the past, this problem has been addressed by two classes of algorithms: (i) "texture synthesis" algorithms for generating large image regions from sample textures, and (ii) "inpainting" techniques for filling in small image gaps. The former has been demonstrated for "textures" – repeating two-dimensional patterns with some stochasticity; the latter focuses on linear "structures" which can be thought of as one-dimensional patterns, such as lines and object contours. This paper presents a novel and efficient algorithm that combines the advantages of these two approaches. We first note that exemplar-based texture synthesis contains the essential process required to replicate both texture and structure; the success of structure propagation, however, is highly dependent on the order in which the filling proceeds. We propose a best-first algorithm in which the confidence in the synthesized pixel values is propagated in a manner similar to the propagation of information in inpainting. The actual color values are computed using exemplar-based synthesis. In this paper, the simultaneous propagation of texture and structure information is achieved by a single, efficient algorithm. Computational efficiency is achieved by a block-based sampling process. A number of examples on real and synthetic images demonstrate the effectiveness of our algorithm in removing large occluding objects as well as thin scratches.


1. INTRODUCTION

This paper demonstrates a novel algorithm for removing large objects from digital photographs and replacing them with visually plausible backgrounds. Fig. 1 shows an example of this task, where the tower (manually selected as the target region) is automatically replaced by data sampled from the remainder of the image. The filling-in of missing regions in an image is known as Image Inpainting. Inpainting is the art of modifying an image or video in a form that is not easily detectable by an ordinary observer. Image Inpainting has become a fundamental area of research in image processing. The modification of images in a way that is non-detectable for an observer who does not know the original image is a practice as old as artistic creation itself. Medieval artwork started to be restored as early as the renaissance. The motive was simple, to bring medieval pictures, ‘up to date’ as to fill any gaps. This practice is called ‘retouching’ or ‘inpainting’. Also image inpainting has been widely investigated in the applications of digital effect (i.e. object removal, image editing, image resizing), image restoration (e.g. Scratch or text removal in photograph), image coding and transmission (e.g. recovery of missing blocks) etc.

![Image](image_url)

(a) Original photograph  (b) Inpainted photograph

Fig.1. Removing large objects from images.

The conventional schemes that are proposed for image inpainting can be divided into two categories:
- Texture oriented
- Structure oriented

In the previous work, several researchers have considered texture synthesis as a way to fill large image regions with ‘pure’ textures-repetitive two dimensional textural patterns. Example of texture synthesis is the exemplar-based technique. This approach effectively generates new texture by sampling and copying color values from the source. Though the techniques of texture synthesis are effective, they have difficulty in filling holes in photographs of real world scenes consisting of composite textures. On the other hand, the structure-oriented technique we pay special attention to linear structures. But, linear structures in the target region only influence the fill order of is an exemplar based texture synthesis algorithm. The result is an algorithm that has the efficiency and qualitative performances of exemplar based texture synthesis, but which also considers the image constraints imposed by surrounding linear structures.

One of the first attempts to use exemplar based synthesis for object removal was by Harrison [5]. There the order in which a pixel in the target region is filled was decided by the level of ‘texturedness’ of pixel’s neighborhood. But for this algorithm, the strong linear structures were often overruled by nearby noise. Recently Jia and others [6] have presented a technique for filling image regions based on texture-segmentation step and smooth linking of structures across the holes. The algorithm has a clear advantage that it can be used to connect curved structures. On the other had the algorithm requires i) an expensive segmentation step and ii) a hard decision about what is present at the boundary between two textures. Finally Zalesny and others [7] described an algorithm for parallel synthesis of composite textures. The algorithm proposes a special purpose solution for synthesis of interface between to ‘knitted’ textures. All the above mentioned
approaches are traditional approaches. Recent work on exemplar based approach is by Zongben [2]. There structure sparsity is used which enables better discrimination of structure and texture.

2. MAIN IDEA

Adopting notations similar to that used in Criminisi’s inpainting algorithm [1], the region to be filled i.e. the target region or masked region is indicated by $\Omega$ and its contour is denoted by $\delta \Omega$. As shown in fig. 3 the contour evolves inward as the algorithm progresses and so it is also called as “fill front”. The source region $\Phi$ which remains inward throughout the algorithm, provides samples used in the filling process. Formally we could express the inpainting problem in this way: Given an image $I$ with a target region $\Omega$, fill in each pixel inside $\Omega$ with a pixel value taken from $\Phi$. Suppose that the square template $\Psi_p \subset \Phi$ centered at point $p$ (fig 2) is to be filled. The best match sample comes from source region patch $\Psi_q \subset \Phi$. In the e.g. in fig 2, we see that if $\Psi_p$ lies on the continuation of an image edge, the most likely best matches will lie along the same edge.

3. REGION FILLING ALGORITHM

We now proceed with the details of our algorithm. First, a user selects a target region, $\Omega$, to be removed and filled. The source region, may be defined as the entire image minus the target region ($\Phi = I_{\Omega}$), as a dilated band around the target region, or it may be manually specified by the user. Next, as with all exemplar-based texture synthesis [4], the size of the template window $\alpha$ must be specified. We provide a default window size of $9 \times 9$ pixels, but in practice require the user to set it to be slightly larger than the largest distinguishable texture element, or “texel”, in the source region. Once these parameters are determined, the remainder of the region-filling process is completely automatic. In our algorithm, each pixel maintains a colour value (or “empty”, if the pixel is unfilled) and a confidence value, which reflects our confidence in the pixel value, and which is frozen once a pixel has been filled. During the course of the algorithm, patches along the fill front are also given a temporary priority value, which determines the order in which they are filled.

In Criminisi’s original algorithm [1], the priority function $P(p)$ for each unfilled pixel $p \in \delta \Omega$ is defined as the product of two terms as follows.

$$P(p) = C(p)D(p),$$

where $C(p)$ is called the confidence term and $D(p)$ is called the data term. The confidence term is used to capture the texture property and the data term represents the structure characteristic. They are respectively defined as follows.

$$C(p) = C(\Psi_p \cap \Phi) * C(q)$$

$$D(p) = |\nabla \Psi_p| \cdot |\nabla q|$$

Following are the steps adopted for the region filling-

a) Original image, with the target region $\Omega$, its contour $\delta \Omega$, and the source region $\Phi$.

b) Synthesizing the area delimitated by the patch $\Psi_P$ centered on the point $p \in \delta \Omega$.

c) The most likely candidate matches for $\Psi_P$ lie along the boundary between the two textures in the source region. e.g. $\Psi_q'$ and $\Psi_q''$.

d) The best-matching patch in the candidates set has been copied into the position occupied by $\Psi_P$, thus achieving partial filling of $\Omega$.

4. EXPERIMENTAL RESULTS AND CONCLUSION

The algorithm is applied to both images as shown in figures and it’s PSNR (db), Total time required (sec) are given as follows. This paper has presented a novel algorithm for removing large objects from digital photographs. The result of object removal is an image in which the selected object has been replaced by a visually plausible background that mimics the appearance of the source region. This is robust algorithm for example based image inpainting, which can be adapted to any image contents of different characteristics. One important parameter of the algorithm is the size of the patch. With bigger patch size, the filling rate is high, thus the program runs faster. However, there’s more important implications on choosing the right patch size.

Results of fig 3:

i) PSNR in db : 2.709051e+001

ii) Total Time taken to inpainting image (in sec) : 5.50000

Results of fig 4:

i) PSNR in db : 2.239232e+001
ii) Total Time taken to inpainting image (in sec) : 134.0160

REFERENCES