An Evaluation of Image Inpainting Algorithms

Sheng Li
January 2013

Bachelor Thesis, 15 credits, C
Computer Science

Computer Science
Examiner: Peter Jenke
Supervisor: Anders Jackson
An Evaluation of Image Inpainting Algorithms

by

Sheng Li

Faculty of Engineering and Sustainable Development
University of Gävle

S-801 76 Gävle, Sweden

Email: tfk10hi@student.hig.se

Abstract

Image inpainting is a technique to repair damaged images or to remove/replace select regions. It was used to repair old artwork. It is also a part of movie special effects. With the extended application of electronic products, a lot of work done on mobile devices. But many algorithms may be unsuitable for mobile devices. The purpose of this work is to present an evaluation of several algorithms of image inpainting created in the past. Two classes of algorithms: Partial Differential Equations (PDEs) based algorithms and Exemplar-Based algorithms. The results show the advantages and disadvantages.

Key words: Image inpainting, PDEs-based algorithm, Exemplar-based algorithm.
1 Introduction

Inpainting is a technique of repair and reconstruction damaged parts of images. In the art area, it aims to fill some cracks or holes to make the image restored. But it is a manual operation, expensive and difficult. In digital world an image may lose some information because of the scanning problem, compression and decompression.

For restitute the image, image inpainting come out as a solution. The purpose is to restore damaged portions of image, e.g. an old photography with scratches, or remove unexpected elements present in the image. Meanwhile, this technique has been applied on film post-production and digital image processing for users to create artistic effects, e.g. a car appearing in the background of the movie “The lord of the rings: the return of the king.” [1]

1.1 Background

The image inpainting was introduced by M. Bertalmio et al. [2]. They invented an algorithm for digital inpainting. At first, the target regions to be restored must be marked. Then propagated the structure of boundary of target regions, drawn the contour lines inside target regions. Filled the gaps with color that matching of boundary. Finally, the region are painted.

During the past decade, inspired by inpainting technique, a number of algorithms have been invented. And it has been implemented on different fields. Some of them are good at larger regions inpainting or small regions restoring. The algorithm based on PDEs [3] (Partial Differential Equations), which propagates the information along the isophotes (contour line) at the boundary. Different from the PDEs algorithms, Exemplar-based [9] algorithm uses sample textures to fill larger regions. Copy the sample textures to target region.

The inpainting technique is more mature, but this technique in the past has only been implemented on powerful devices. With the development of technology, camera is more and more common. When capture a picture, extra elements may be very disturbing in the picture. For removing those parts, a powerful computer and complex software are necessary. There are several image tools can do the inpainting such as PhotoShop[4], Inpaint [5] etc. So this work tries to find more efficient algorithm. In the future, this algorithm could be implemented on mobile devices.

1.2 Requirements of implementation

A lot of works are done by mobile devices, so many techniques have been implemented on mobile devices. For image inpainting, the evaluation should take care of several factors. M. Bertalmio et al. [2] had created an algorithm that has an impressive result, but it has a long running time. It is not appropriate to apply it on mobile device, because it’s constraints on battery power.

Due to the limitation of hardware of mobile device, like small screen and slow CPU, a suitable algorithm has to take those factors in concern. The two main factors of the evaluation of algorithms:

- **The quality of result.** After inpainting, the quality of result should be close to a goal that the observer cannot realize that the images have defects or have been repaired.
- **High efficiency.** The CPU and GPU of mobile device are not powerful as computer, so the algorithm must more efficient. The optimization of algorithm may improve the user experience.

In the follow sections, two algorithms that from previous works will be evaluated and tested. The quality of result, processing time and usage of resource are the
criterion of the evaluation. And apply it on images with different domain should be restored (e.g. the image with small holes, scratch, text. Removing a big object that has a simple or complex background.). From those data, find the laws and drawbacks of the algorithm. Finally, discuss the feasibility of implementation and further work. Then conclude this work.

1.3 Research questions

This work aims at an evaluation of the image inpainting technique. The purpose is to find more efficient method to restore the damaged portions of images. Here is some research questions:

- What kind of algorithms can restore an image that has damaged portions?
- What are the advantages/disadvantages of the solution?
- Which algorithm is more efficient? The quality of result or usage of resource?
- Is it possible to work on mobile device with slow CPU, low memory and small screen?
2 Literature and related works

In previous works, many algorithms have been introduced. They are working on pixels. When removing the selected regions, new elements must fill those regions. But the original background of the regions are not known to the algorithm. The only way is to predict it from the boundary of the inpainting region. It can be chosen from neighborhoods as textures to fill the target regions, or combine the nearest pixels information. This can be repeated until inpainting regions have filled.

A number of surveys show that the inpainting algorithms have been divided into two categories [6]: PDEs-based algorithm is used to repair small size region of digital image, exemplar-based inpainting algorithm is used to fill large region.

2.1 PDEs algorithms

For small regions, a mathematical method has been introduced by M. Bertalmio et al [2]. The method is propagating the geometric structure of the image. It’s based on partial differential equations (PDEs). This method translated the manual inpainting to a mathematical and algorithmic language. Just 4 steps:

1. Let the digital image be a discrete 2D gray level image.
2. The surrounded area of select region determined the inpainting region. When propagate information into inpainting region, the propagation direction must be defined. Propagate information along the contour line direction, until reach the boundary.
3. Computing the change information along the propagation direction, then fill the change information into inpainting region.
4. The small details are restored. The algorithm runs a few diffusion iterations of step 2, 3.

There are the main steps of the algorithm. The algorithm ends when all target region be filled.

Based on the work of M. Bertalmio et al [2], Chan and Shen have introduced two image inpainting algorithms [7], [8]. The Total Variational [7] (TV) model uses an Euler-LaGrange equation coupled with anisotropic diffusion to preserve the direction of isophotes. The algorithm can inpainting the small regions and removing noise. But the drawback is broken edges connected. The Curvature-Driven Diffusion (CDD) [8] model extended the TV algorithm to take into consideration geometric information of isophotes by the strength of the diffusion process, so it can repair larger regions than TV [7].

The TV [7] and CDD [8] translate the inpainting into a calculus of variation problem. The partial differential equations and calculus of variation can be equivalent by mathematical derivation.
In figure 1, with target region, source region and vectors. For propagate information from source region to target region, compute its change along the $\vec{N}$ directions. The $\vec{N}$ is the direction of smallest spatial change. Keep doing this, until reach the boundary of target region.

2.2 Exemplar-based algorithms

If the inpainting regions are big, then the mathematical method may not be suitable. So exemplar-based inpainting is the better choice here. An algorithm based on texture-exemplar inpainting was introduced by Criminisi et al [9]. Extension of texture synthesis [10] methods as a way to fill in large regions with “pure” textures – repetitive two-dimensional textural patterns with some stochastically. There are some other algorithms for texture synthesis, such as, stochastic texture synthesis, pixel-based texture synthesis and patch-based texture synthesis [10]. The damaged region is filled by copying color values from the neighborhood.

Patch-based (exemplar-based) technique is a part of texture synthesis. It is more cheaply and effectively than others. Criminisi et al [9] presents a novel and efficient algorithm that combines the “texture synthesis” algorithms generating image regions from sample textures and “inpainting” techniques that fill the small image gaps. This algorithm could be described as follows:
1. Determine the parameters of algorithm. A user selects a target region, $\Omega$, to be restored. So the source region $\Phi$ is entire image minus the target region. Then the size of exemplar texture $\psi$ e.g., $9 \times 9$. Those template textures called patches that contain a color value and a confidence value, which determines a pixel has been filled.

2. Compute patch priorities $P(p)$. The algorithm performs a best-first filling algorithm that determines the priority of every pixel. The priority is defined as the product of confidence term $C(p)$ and data term $D(p)$.
   1) $P(p) = C(p)D(p)$.
   2) $C(p) = \frac{\sum_{q \in \psi(p)} C(q)}{|\psi(p)|}$
   3) $D(p) = \frac{|\nabla p \cdot p|}{\alpha}$

3. Propagate texture and structure information. To find the best matches texture, the SSD (the sum of squared differences) of position of these patches is the measure. When the best one is found, propagate the information to the fill region.

4. Update confidence values. After filling the patch, the confidence of the patch is updated.

Figure 2 shows a single process of exemplar-based algorithm. The top-left one is the original image, the top-right one has a patch P, the bottom-left one has three patches P1, P2 and P3. In the bottom-right, the patch P be restored.

![Figure 2: An example of exemplar-based algorithm. (By Sheng Li)]
3 Results

In this chapter, two main methods will be tested. The first one is PDEs inpainting algorithm. Another is exemplar-based inpainting. For PDEs method. When the selected region is small, the result is great. In texture synthesis, a large region, the foreground, or a big object, are easy to fill. In this work, PDEs algorithm for inpainting and “Object Removal by Exemplar-Based Inpainting” [9]. The test based on MATLAB environment.

3.1 PDEs inpainting algorithm

This method is provided by MATLAB. It has a function that named ROIFILL () [11] which is used to fill the target regions. Steve Eddins [12] said that “roifill which implements a simple form of PDE-based inpainting.” It can only work on uint8, uint16, int16, single, or double images. If the input image is a RGB image, it should be transferred to three channels(R, G, and B) or (H, S, V). Then apply this method to each channel.

Figure 3 shows two examples. The original image with RGB three channels and HSV image with Hue, Saturation, and Value channels in MATLAB. Transfer the image to single channel. Then apply the algorithm.

After transferring the image to several single channels, the target area should be classified. The function provides several ways to choose the area, one is the user give the coordinates of vertices, or uses a binary image as mask. The target region corresponding to the nonzero pixels and so on.
In the figure 4, the lighter line disappear. The result is acceptable. The processing time is 9.1105s. The size of this image is 3264*1840. The original image is a flight track of an airplane in the sky. After inpainting, it mostly disappears.

### 3.2 Exemplar-based inpainting algorithm

This test inspired by Sooraj Bhat [13]. The first step of exemplar-based inpainting algorithm is to specify the region that will be restored. Then follow the steps of exemplar-based inpainting algorithm described in section 2.2.

![Figure 4: The result of inpainting. (By Sheng Li)](image)

The figure 5 shows the left one is original image with (400*200 pixels), to be restored image with 1248(16*78) pixels at the center, and the result (right one). In the program, the green part is the inpainting part. The processing time is 2.5 seconds.

From the result, the geometry of the image is restored. From the formulas, the result of exemplar-based texture inpainting is really good. The experimental results of this algorithm are perfect in some cases. But if the background is too complex or the inpainting region is too big, the processing time and quality of results are bad than expected. The size of source was the whole image, but the processing time is too long.
3.3 Comparison

To compare PDEs algorithm and exemplar-based algorithm, the processing time, memory usage and the quality of result are three main aspects.

All these three images, the two algorithms have been applied. For different image with different situation, the result shows the difference between those two algorithms. The A image has two different grayscale parts, the inpainting region is connection part. The B image is a wall with windows. The mission is repaired a half window. The last one C as the same as B, but the objective is to remove the window. The A image tested the inpainting of simple geometry image. The B image show the restoration of image with a complex background. The C image will be removed a big object with simple background.

Table 1: Information of inpainting image resolution and to be restored area.

<table>
<thead>
<tr>
<th></th>
<th>Resolution</th>
<th>Size to be restored</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>200*400(80000) pixels</td>
<td>4505</td>
</tr>
<tr>
<td>B</td>
<td>503*491(246973) pixels</td>
<td>21033</td>
</tr>
<tr>
<td>C</td>
<td>503*491(246973) pixels</td>
<td>37349</td>
</tr>
</tbody>
</table>

Figure 6: To be restored area of each image. (By Sheng Li)
The figure 7 shows the result of each algorithms. The a, b and c are implemented exemplar-based texture algorithm. The d, e and f are implemented PDEs algorithm. From the result, the exemplar-based algorithm can retouch the image. It can partly repair the image. But the c shows a drawback of exemplar-based inpainting. The PDEs algorithm looks like the blur of the image. But the f shows a good result. It is suitable for a narrow or small region filled and the object with a simple background deletion.

Figure 7: The result of different image. (By Sheng Li)
Table 2: The processing time of each image on each inpainting algorithm.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exemplar</td>
<td>8.90s</td>
<td>194.04s</td>
<td>390.09s</td>
</tr>
<tr>
<td>PDEs</td>
<td>0.34s</td>
<td>4.36s</td>
<td>7.42s</td>
</tr>
</tbody>
</table>

From the table 2, implemented different algorithm on the same image, the processing time of PDEs algorithm is faster than the exemplar, but the quality of exemplar is better than PDEs. With the target region increase and more complex, the exemplar based inpainting algorithm can get a better result than PDEs. But if the background of target region is simple, the PDEs get an ideal result.

In my case, the PDEs algorithm is faster than exemplar-based algorithm. But the PDEs is an optimized MATLAB function. It is not fair for comparison. It gives us a future research: How to optimize the algorithm.
4 Discussion

In this work, this project just finds the suitable method. In the future, it should be able to apply it on mobile devices.

Those inpainting algorithms are successfully restore those images. They are all suitable for small region inpainting. And the PDEs are good at remove object that if the background is simple. Otherwise, the result will looks like blurring.

Exemplar-based inpainting algorithms are good at restoration of missing region. But the processing time is a problem. Too much computations on find the best matches patch. If the source region is larger, the time will be longer. If the source region is small, the quality of the result will be bad.

The weakness of my exemplar-based inpainting algorithm is the choice of size of patches. If the size of source patch is too big, the best patch may not suitable. If otherwise, the size of source region is too small, the algorithm couldn’t find the best match.

In the future work, the application may chose the suitable algorithm depended on the shape of inpainting region or the complexity of the boundary that can base on the number of contour lines.
5 Conclusion

Related to research questions in section 1.3 on page 2, I have some answers and ideas.

- In this work, I choose the two categories algorithm to do inpainting. One is PDEs algorithms and the other is exemplar/patch based inpainting. There are several other algorithms like sparse inpainting algorithm, clone stamp tools in PHOTOSHOP and so on.
- From the testing data, they are suitable for different situations, like PDEs is good at restore small holes, stretch or a big region with a simple background. It is strict on the complexity of background. The exemplar based inpainting is good at larger region restoration, like in figure 8(b), the style of window have been restored. But it produce too much computation on match of patches.
- For the small and narrow target region, the PDEs is more efficient. If the target region is larger, the result of PDEs is unacceptable. But the exemplar algorithm is better.
- It is possible that implement the inpainting technique on mobile device. But the algorithm must more efficient.

Though the exemplar-based algorithm is a suitable method, but it still needs to improve. If the image has a big resolution e.g., 3264*1840, the processing time almost 10 minutes. So the algorithm is not yet suitable to implement on mobile device. It has to be suitable for today’s mobile devices.

For the PDEs algorithms, the processing is faster than texture algorithm, but it only suitable for narrow and simple region inpainting. When imply the algorithm on big region, the result looks like the blur of the region. But the algorithm would work on today’s devices.

Recently, some mobile device has this technology. For example, the Nokia Lumia 920 has a new function called Smart Shoot that can remove the extra object that specify by the user or detection by system on mobile device. The user chose this function and capture several images, then the user select one image and remove region.

There is also a software called Retouch came on Android and IOS device in the summer of 2012. The result is good and the processing time is acceptable. It is usable for mobile device.
6 Acknowledgements

Thanks my thesis supervisor Anders Jackson, guiding me to the right way to achieve the goal. I also thanks those who provide the images and the example code of the project.

I really grateful for my supervisor’s patient and help. Thanks to the examiner Peter Jenke.
References


Appendix: Example images.

In this appendix, two ordinary images that capture by mobile device as example for image inpainting. The same area in the images are restored by PDE algorithm and texture patch algorithm.

Figure 8: An original image (left) and the image with green color region that should be restored (right). (By Sheng Li)

Figure 9: After PDE algorithm inpainting (left) and the details (right). (By Sheng Li)
Figure 10: After texture-based algorithm inpainting (left) and the details (right). (By Sheng Li)

Figure 11: An original image (left) and the image with green color region that should be restored (right). (By Sheng Li)
Figure 12: After texture-based algorithm inpainting (left) and the details (right). (By Sheng Li)

Figure 13: After PDE algorithm inpainting (left) and the details (right). (By Sheng Li)