JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 24, 1241-1255 (2008)

Short Paper

Multilayer Inpainting on Digitalized Artworks*

RONG-CHI CHANG AND TIMOTHY K. SHIH[+]
Department of Digital Media Design
Asia University
Taichung, 413 Taiwan
E-mail: roger@asia.edu.tw
[+]*Department of Computer Science*
National Taipei University of Education
Taipei, 106 Taiwan
E-mail: timothykshih@gmail.com

Image inpainting automatically restores damaged images and partially removed pictures. Since most inpainting mechanisms inpaint damaged portions in a single layer, this study proposes a multilayer inpainting mechanism by examining how Chinese paintings are drawn in layers. The proposed multilayer inpainting mechanism employs a layer fusion strategy to detect the optimal inpaint combination among layers to restore paintings. Because this multilayer multi-resolution strategy considers damages in each layer from a multi-resolution perspective, it is superior to several existing techniques for restoring Chinese and Western paintings. In this study, the proposed algorithm is tested on more than 1,500 still images, with evaluations showing the effectiveness of image inpainting. The results indicate that the proposed algorithm achieves high PSNR values as well as high user satisfactions, including inpainting in some extreme cases where more than ninety percent of a painting are destroyed.

Keywords: image inpainting, image restoration, image completion, multi-resolution inpainting, multilayer inpainting, image processing

1. INTRODUCTION

Automatic digital inpainting is a technique which restores damaged image or video by means of spatial/temporal interpolation and other mechanisms. 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 are based on extrapolation or interpolation of neighboring pixels [4], recovery of edges, curvature-driven diffusions (according to the connectivity principle in vision psychology) [2], and inpainting from multiple view points (*i.e.*, image from movie, or image from different time and view point).

An approach to non-texture inpainting is based on a Partial Differential Equations

Received October 2, 2006; revised January 5 & April 4, 2007; accepted May 2, 2007.
Communicated by Ja-Ling Wu.
* This word was supported in part by the research project "Classic Video Regeneration and Doctoring" and the National Science Council of Taiwan, R.O.C. under contract No. NSC 95-2221-E-468-007.

(PDE) method and a calculus of variations [3]. Oliveira *et al*. [3] introduced a simple and faster mechanism to filling a damaged area. Efficiency of the proposed method [3] is two to three orders of magnitude faster than those using partial differential equations. Chan and Shen developed inpainting schemes from the viewpoint of variational principles and image prior model [5]. The method explains successfully some aspects of the human disocclusion process in vision psychology [5]. Compared with all other variational inpainting schemes, the Total Variation (TV) model has the lowest complexity and easiest implementation. It works remarkably well for local inpainting such as digital zoom-in and text removal [5, 6]. However, for large area 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 [2]. 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 [5]. 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 this paper, a set of new image inpainting algorithms is proposed based on different layers and different levels of details using characteristics of paintings. First, the *multilayer and multi-resolution inpainting* method is presented to achieve a better human perception of inpainted images as well as reasonably good PSNR values. Built upon these research purposes, the characteristics of damaged pictures have been carefully considered. The algorithms are implemented and tested on more than 1,500 pictures, including Chinese and western paintings, photos, and cartoon drawings. As the results indicate, this multilayer and multi-resolution inpainting method has better performances and compare favorably to those obtained by existing techniques.

The rest of this paper is organized as follows. In section 2, an image inpainting method is described based on the multi-resolution characteristic of image. A general principle of Chinese painting is then discussed to elicit the concept behind the proposed algorithms. The strength and effectiveness with examples are provided to show restoration of painting in section 4. Finally, in section 5, the conclusion is presented along with a discussion of detailed results and analysis.

2. THE MULTI-RESOLUTION INPAINTING ALGORITHM

The base inpainting method that is introduced in the proposed algorithm takes in to account different levels of details of still images [8]. In general, 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. Additionally, if the variance of pixel colors is large in an image block, it is possible that the block contains 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	An image block subdivided from a **DIB**
PB	A pixel block subdivided 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 $\boldsymbol{\beta}_1 < \boldsymbol{\beta}_2$

A recursive algorithm in the study is designed to take image files of a conventional format and inpaint damaged portions. The algorithm works as the following. Let DIB be a damaged image block. The DIB is subdivided into n by n image blocks (*i.e.*, IBs, see Fig. 1). The default value of n is set to 16. There are three adjustable thresholds used: α is a threshold of variance of pixel colors, and $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are the thresholds of percentages. The selection of these thresholds are discussed in section 5. We assume that, an image block has $i \times j$ pixels, and a color variance, *var*, can be calculated as

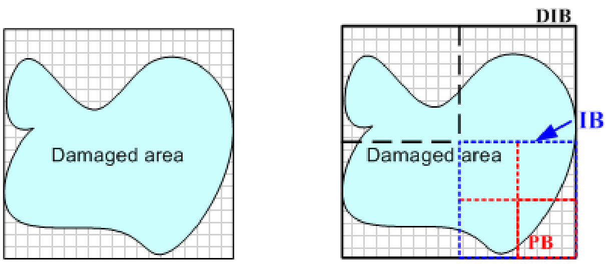


Fig. 1. An illustration of (a) damaged area, (b) damage image block (DIB), image block (IB) and pixel block (PB).

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

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 $\boldsymbol{\alpha}$ is a percentage in the range between 0 and 100 (the maximum *var*). Another criterion is the percentage of potential damaged pixels. 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 (variation and percentage) 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 $\boldsymbol{\alpha}$, the variation of pixels in IB is low. No recursive subdivision is required (*i.e.*, no need of looking at the next level of details). But, 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 and the block size of PB are 4×4 pixles. 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 [11].

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. GENERAL PRINCIPLE OF PAINTING

We study fundamental techniques of Chinese landscape painting [9]. An artist paints trees in dark ink before distant rocks in tan colors are added. Finally, light colors are used in the background. In general, the use of color is from dark to light. We argue that it is important to use dark colors to inpaint missing potion of trees, to use tan colors for rocks, and to use light colors for the background. But, color segmentation is difficult. It is not easy to separate trees from rocks and background. However, an approximation approach should be considered since trees and rocks could be interleaved. 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. More layers can be used. Layer separation on western paintings is not easy but possible.

4. THE NEW INPAINTING STRATEGY

Most inpainting algorithms treat a damaged picture as a single layer and computes inpaint data based on the single layer. However, single layer approach may use irrelevant information on restoring an object, without considering the separation of the object and its background. In our earlier approach [11], we separate a Chinese painting into several layers. Each layer contains different objects (*e.g.*, far mountain, near mountain, trees, and people) and is processed separately. In the end, a fusion algorithm is used to combine the results.

The proposed algorithm, *multilayer inpainting* scheme, has three main building blocks: layer separation, inpainting, and layer fusion. These techniques are discussed in the following subsections.

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 percent since information is lost. With this in mind, an approximation approach is employed to separate objects into different layers. Additionally, a naive layer separation algorithm is used 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.

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 occur in a picture for at least 5 percent. And, other pixels differ from 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 $\boldsymbol{K}$ layers**.**

The results are stored in **DIBLayer[*K*]** , which is used by an inpainting algorithm discussed in the next section. Note that, we use the 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 reasonably good separation.

Algorithm 2 Layer Separation
Let **DIB** be a damaged image block
Let **DIBLayer[*K*]** be a set of damaged image layers
Let ρ be a threshold of percentage
Let δ_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 ρ and δ_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.
}

4.2 The Inpainting 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 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 as the original mask.

```
Algorithm 3    Multi-resolution inpainting algorithm for each layer
Let DIBk be a damaged image block
Let α be a threshold of variance
Let β1, β2 be a threshold of percentage, β1 < β2

Multi_Resolution_Inpainting(block DIBk) {
    divide DIBk into n × n image blocks
    for each image block IB {
        let var be the color variance of IB
        if var < α then {
            Let PB be an x × y pixel block in IB
            for each PB in the image block {
                if the percentage of damaged pixels in PB > β2
                    inpaint style = Mean of IB
                else if percentage of damaged pixels in PB > β1
                        inpaint style = Mean of PB
                    else
                        inpaint style = Neighboring
                inpaint PB use its inpaint style
            }
        }
        else
                call Multi_Resolution_Inpainting(IB)
    }
}
```

4.3 The Multilayer 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 foreground. 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 $\mu[P]$ 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.

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.

Algorithm 4 Multilayer 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 $\boldsymbol{\mu}$**[*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 μ[P] of layer > μ[P] of layer + 1
         inpaint P in PIC using the pixel in layer
      else
         inpaint P in PIC using the pixel in layer + 1
      }
   }
}
```

5. EXPERIMENTAL RESULTS AND ANALYSIS

Although the mechanisms discussed in [4, 10, 12] are related to image inpainting, however, the approaches are different from ours. We briefly point out these mechanisms before we discuss our experiments and compare our results with others using similar mechanisms. The work discussed in [10] is based on searching for patches (*i.e.*, block of pixels) with two considerations. Firstly, priorities of target patches at the boundary of the inpainted area are computed in order to obtain the order of copying source patches. Secondly, each copied patch is associated with a confidence value which indicates the degree of believe of the copied information. The approach may produce image from a discontinuous space. That is, patches at non-consecutive locations can be combined to fill an area. This is a quite different process as compared to ours. Our algorithm maintains the spatial continuity. The work discussed in [12] is based on texture synthesis with a color segmentation approach, which is not directly related to our methods. The mechanism can be extended to inpaint objects in a 3D space. The approach discussed in [4] uses third-order partial differential equations (PDEs). Similarly, this is a quite different approach as compared to ours.

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. We also compare the Fast Image Inpainting model [3], TV inpainting scheme [5], CDD inpainting [2] and our inpainting method based on the same images. The comparison and analysis are presented in sections 5.1, 5.2, and 5.3. Secondly, we test some extreme cases in section 5.4, with damaged area up to more than 90% of the entire image. Computation time is also summarized.

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.

A set of photos shown in Fig. 2 demonstrate the comparison between *single* and *multiple* resolution inpainting algorithms. The photo was vandalized with a red mask Fig. 2 (b) covering 47.8% of its original. The restored images shown in Figs. 2 (c) and (d), essentially recovers all details of the original picture. Fig. 3 illustrates the result of removing the microphone from the original picture Fig. 3 (a). From Figs. 3 (b) and (c), the results show that the proposed multi-resolution inpainting method has a smoother perception than Bertalmio's *et al*. method [1]. Fig. 4 is the enlargement of images from Fig. 3 (see the differences in the ovals).

(a) (b) (c) (d)

Fig. 2. Experiment results by single and multiple resolution inpainting algorithms. (a) Original picture; (b) A damaged picture with 47.8% red mask covered; (c) The result by single-resolution algorithm, PSNR = 24.73 dB; (d) The result by multi-resolution algorithm, PSNR = 27.44 dB.

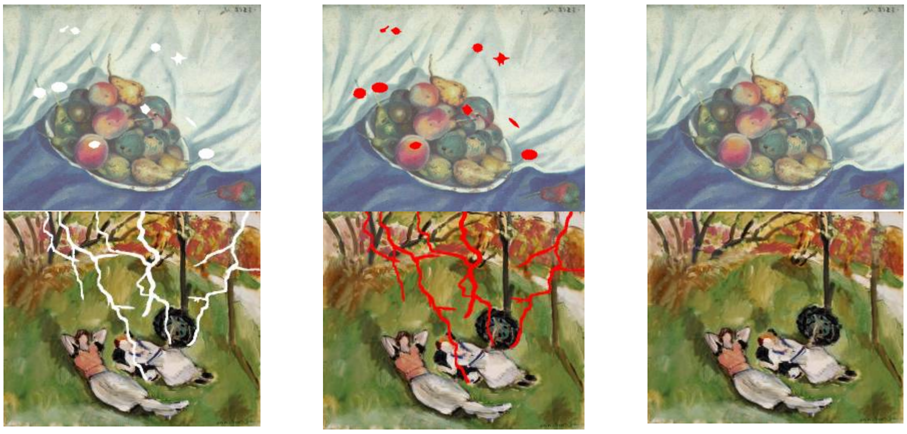
(a) Original picture [1]. (b) By Bertalmio *et al.* [1]. (c) By our propose method.

Fig. 3. The microphone has been removed in the test picture.

Some test results of multilayer multi-resolution inpainting is shown in Fig. 5. Different categories of Chinese painting and western paintings are randomly tested on different damages. In the experiments, each picture is decomposed into 4 layers (*i.e.*, $K = 4$). It is very difficult to choose thresholds in our algorithm. Thus, different combinations are tested to perform a complete analysis on these thresholds.

(a) Original picture [1]. (b) By Bertalmio *et al.* [1]. (c) By our propose method.

Fig. 4. The enlargement of image from Fig. 3 (see the differences in the ovals).



(a) An original picture. (b) Damaged area marked in red. (c) Inpainted picture (our result).

Fig. 5. Damaged and inpainted pictures (western painting).

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

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 (Chinese and western)	18.54	26.95	28.72
Average PSNR	18.54	31.87	34.04

(b) Multilayer stage.

	Damaged	Single Resolution	Multi-Resolution
Photo	19.33	34.73	38.66
Cartoon	17.76	25.25	27.80
Painting (Chinese and western)	18.54	30.86	32.03
Average PSNR	18.54	30.28	32.83

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.

Besides Chinese paintings, we also use photos and cartoon drawings in our experiments. We tested 1,500 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 (including 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. The multilayer approach is better than single layer approach in Chinese and western painting. This result (Table 1) proves that our multilayer inpainting strategy is suitable for painting in general. And, single layer approach on painting obtains a less favorable result. However, the performance of single layer inpainting is better in cartoon and photo images. The average of single layer inpainting is superior to multilayer.

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 [2, 3, 5].

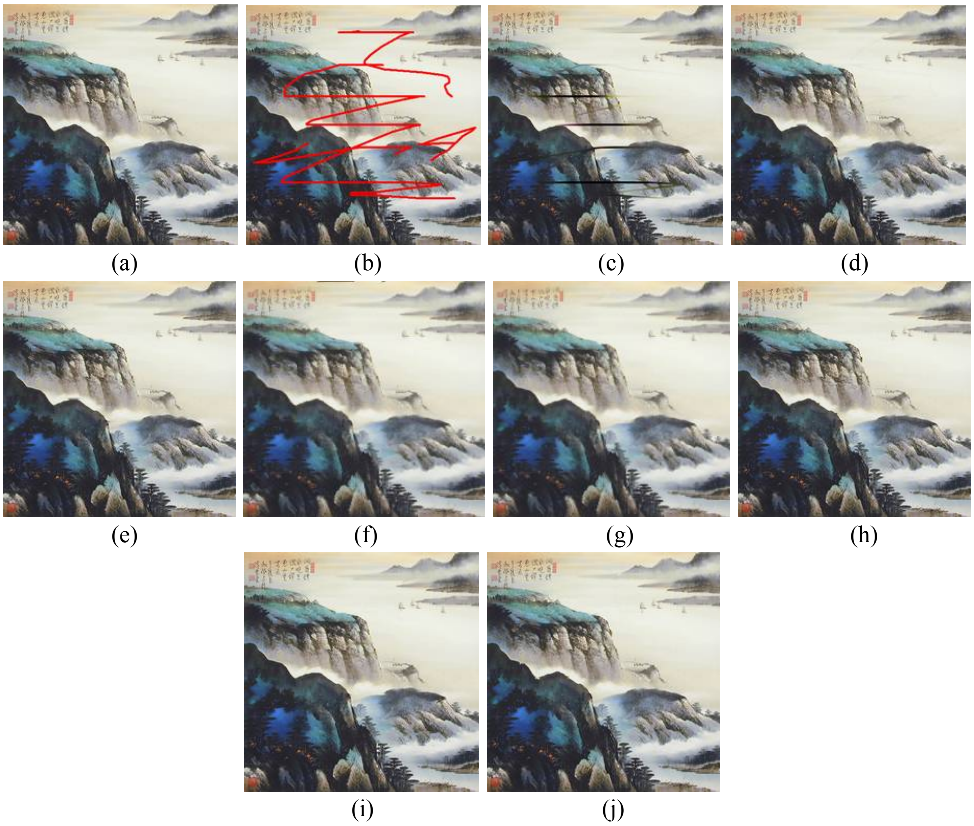


Fig. 6. Results of inpainted Chinese painting by different methods. (a) Original image; (b) Damaged image (7.3% damaged); (c) Result of TV inpainting, PSNR = 24.13 dB; (d) Result of fast image inpainting, PSNR = 32.01 dB; (e) Result of CDD inpainting, PSNR = 34.26 dB; (f) Result of BSCB inpainting [1] PSNR = 32.33 dB; (g) Result of single resolution inpainting, PSNR = 33.56 dB; (h) Result of our multi-resolution inpainting, PSNR = 34.23 dB; (i) Result of our multilayer single resolution inpainting, PSNR = 33.28 dB; (j) Result of our multilayer multi-resolution inpainting, PSNR = 34.54 dB.

Fig. 6 shows how our multilayer inpainting algorithm achieves the best structural continuation in a Chinese painting. Our inpainting algorithm has been compared with three other existing techniques, BSCB inpainting [1], fast image inpainting [3], TV inpainting [5] and CDD image inpainting [2] schemes. Considering the PSNR values by using the results presented in Figs. 6 (c) to (j), we conclude that our proposed multilayer multi-resolution inpainting algorithm can provide the best picture quality in general.

5.4 Extreme Cases

To test the extreme power of our algorithms, we randomly generate damages to cover a large percentage of the original picture. The percentages of noise are from 50% up to more than 90%. In Fig. 7, in order to observe the damaged picture clearly, we use

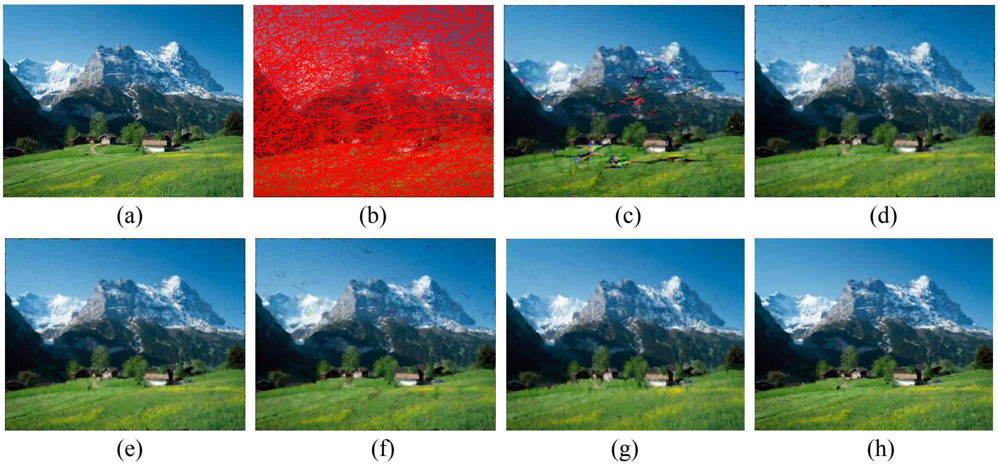


Fig. 7. Extreme results of image quality with different inpainting methods in 640 × 453 image size (Photo). (a) Original picture; (b) Damaged image, noise ratio: 73.1%; PSNR: 5.72 dB; (c) TV inpainting; (d) Fast image inpainting; (e) CCD method; (f) BSCB inpainting; (g) Single resolution; (h) Multi-resolution.

Table 2. The PSNR value and computation time of each inpainting method used in Fig. 7.

Fig. 7	(c)	(d)	(e)	(f)	(g)	(h)
PSNR (dB)	19.58	24.05	23.73	23.38	24.5	25.1
Time (sec)	139.6	199	4472	129	3.8	3.8

red color to show noise. Even the noise ratios are high, our inpainting algorithm can still recover pictures with visually pleasant results and reasonable PSNR values. The result is also compared with others quantitatively. We show the computation time and PSNR values of each algorithm that we used in Fig. 7 in Table 2. Our proposed algorithm shows that Figs. 7 (g) and (h) are superior to others on both PSNR values and performance (50 times to more than 1,000 times faster).

6. CONCLUSION

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. From a unique perspective, our first contribution proposes 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. However, to repair large continuous areas is not our recent focus. We believe that a complete lost of information of an object in a Chinese painting can not be automatically recovered unless human is involved in the loop. In addition to the first contribution, our second contribution is a multi-resolution inpainting algorithm which can recover picture with a very high percentage of damage. In some extreme cases, the ratios of noises are higher than 90%.

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. Several 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.

REFERENCES

1. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 2000, pp. 417-424.
2. T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, Vol. 12, 2001, pp. 436-449.
3. M. M. Oliveira, B. Bowen, R. McKenna, and Y. S. Chang, "Fast digital image inpainting," in *Proceedings of the International Conference on Visualization*, *Imaging and Image Processing*, 2001, pp. 261-266.
4. M. Bertalmío, "Strong-continuation, contrast-invariant inpainting with a third-order optimal PDE," *IEEE Transactions Image Processing*, Vol. 15, 2006, pp. 1934-1938.
5. T. F. Chan and J. Shen, "Mathematical models for local nontexture inpaintings," *SIAM Journal on Applied Mathematics*, Vol. 62, 2002, pp. 1019-1043.
6. K. A. Patwardhan and G. Sapiro, "Projection based image and video inpainting using wavelets," in *Proceedings of the International Conference on Image Processing*, Vol. 1, 2003, pp. 857-860.
7. S. D. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression," in *Proceedings of International Conference on Image Processing*, 2002, pp. 317-320.
8. T. K. Shih, L. C. Lu, Y. H. Wang, and R. C. Chang, "Multi-resolution image inpainting," in *Proceedings of IEEE International Conference on Multimedia and Expo*, Vol. 1, 2003, pp. 485-488.
9. Y. H. Chen, "Chinese painting by Chen Yung-Hao," 1993, pp. 24-42.

10. A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, Vol. 13, 2004, pp. 1200-1212.
11. T. K. Shih, L. C. Lu, and R. C. Chang, "An automatic image inpaint tool," in *Proceedings of the 11th ACM International Conference on Multimedia*, 2003, pp. 102-103.
12. J. Jia and C. Kang, "Inference of segmented color and texture description by tensor voting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, 2004, pp. 771-786.

Rong-Chi Chang (張榮吉) is an assistant professor of Digital Media Design at Asia University, Taiwan. He specializes in image restoration and also works as the program coordinator of applied computer science in the Multimedia Computing Laboratory. He earned both M.S. and Ph.D. degrees from Tamkang University in Computer Science and Engineering. His research focuses on multimedia computing, computer network, and image processing.

Timothy K. Shih (施國琛) is a professor of the Department of Computer Science at National Taipei University of Education, Taiwan. As a senior member of IEEE, Dr. Shih joins the Educational Activities Board of the Computer Society. His current research interests include multimedia computing and distance learning. He is the founder and co-editor-in-chief of the International Journal of Distance Education Technologies. He is also the associate editor of IEEE Transactions on Multimedia and ACM Transactions on Internet Technology.