Fast Digital Image Inpainting

Manuel M. Oliveira Brian Bowen Richard McKenna Yu-Sung Chang

Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11794-4400
{oliveira|bbowen|richard|yusung}@cs.sunysb.edu

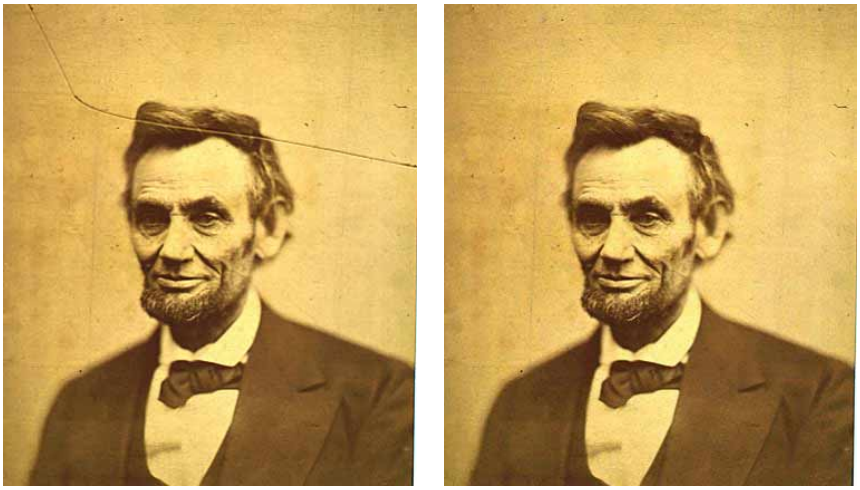

Fig. 1. Left: An 1865 Photograph of Abraham Lincoln taken by Alexander Gardner (courtesy of Wing Yung and Ajeet Shankar [15]). Right: Image restored with our algorithm. The inpainting time took about half of a second.

Abstract. Reconstruction of missing or damaged portions of images is an ancient practice used extensively in artwork restoration. Recently, a few digital inpainting models based on the use of partial differential equations have been proposed. Unfortunately, these algorithms are computationally expensive, usually taking a few minutes to restore small portions of an image, which makes them inappropriate for interactive applications. We discuss the causes of inefficiency of these algorithms and present a simple inpainting model that is two to three orders of magnitude faster, while producing results comparable the ones obtained with current methods.

1 Introduction

Reconstruction of missing or damaged portions of images is an ancient practice used extensively in artwork restoration. Also known as *inpainting* or *retouching*, this activity consists of filling in the missing areas or modifying the damaged ones in a non-detectable way for an observer not familiar with the original images [2]. Uses of image inpainting range from restoration of photographs, films and paintings, to removal of occlusions, such as text, subtitles, stamps and publicity from images. In addition, inpainting can also be used to produce special effects.

Traditionally, skilled artists have performed image inpainting manually. Recently, digital techniques have been used for automatic restoration of scratched films [10] and Bertalmio et al [2] have introduced a technique for digital inpainting of still images. This work has inspired other researchers [4] [5], including us, to explore the subject. The algorithms described in [2] [4] [5] use iterative methods for solving partial differential equations (PDEs), usually requiring several minutes on current personal computers for the inpainting of relatively small areas [2] [15]. Given the large range of applications of image inpainting, it would be desirable to have such a feature included as part of popular image tools such as PhotoShop. Unfortunately, having to wait several minutes is unacceptable for an interactive session, and faster algorithms are needed.

We have analyzed the sources of limitations and inefficiencies of the current digital inpainting algorithms and designed a simple inpainting model based on these observations. The sampling theorem [8] imposes a fundamental limit on the quality of the information that can be restored by any automatic inpainting model, despite of its mathematical sophistication. The human visual system can tolerate some amount of blurring in areas not associated to high contrast edges. We use these facts to design an inpainting

algorithm that produces results comparable to those found in the literature [2] [4] [5], but two to three orders of magnitude faster. Figure 1 (left) shows a famous cracked photograph of Abraham Lincoln taken in 1865. The image to its right shows the result obtained with our algorithm in 0.61 seconds on a 450 MHz Pentium III PC.

Our solution is (in retrospect) very simple. Thus, the primary contribution of this work is not the inpainting algorithm *per se*, but instead the analysis of the problem, which shows that sophisticated mathematical models have limited impact on automatic inpainting procedures. We illustrate our point and the effectiveness of our approach with examples of restoration of old photographs, vandalized images, and text removal, similar to the ones found in the literature.

2 Previous and Related Work

Digital image inpainting techniques can be classified according to their primary goal as *film restoration*, *texture synthesis* and *disocclusion* [2]. In the case of film restoration, data obtained from adjacent frames is the primary source of information for the inpainting process [5]. Due to the large number of frames associated with a movie, fully automated and fast techniques are desired. Texture synthesis consists of generating, from a given texture sample, a new texture that is perceived as being statistically similar to the sample [16]. Notice that texture synthesis is an important component of the general image-inpainting problem, since the area to be retouched might involve textured surfaces. Disocclusion techniques try to recover information about surfaces not directly visible in a scene and, therefore, are the closest in relation to inpainting of still images. Disocclusions have been studied both in computer vision [12] and image-based rendering [11] contexts.

Bertalmio et al [2] pioneered a digital image-inpainting algorithm based on a PDE model. A user-provided mask specifies the portions of the input image to be retouched. The algorithm treats the input image as three separate channels (R, G and B). For each channel, it fills in the areas to be inpainted by propagating information from the outside of the masked region along level lines (isophotes). Isophote directions are obtained by computing at each pixel along the contour a discretized gradient vector (this gives the direction of largest spatial change) and by rotating the resulting vector by 90 degrees. This intends to propagate information while preserving edges. A 2-D Laplacian [8] is used to locally estimate the variation in smoothness and such variation is propagated along the isophote direction [2]. After every few step of the inpainting process, the algorithm runs a few diffusion iterations to smooth the inpainted region. Anisotropic diffusion [P] is used in order to preserve edges across the inpainted region.

Inspired by the work of Bertalmio et al. [2], Chan and Shen proposed two image inpainting algorithms [4] [5]. The *Total Variational* (TV) inpainting model [4] uses an Euler-Lagrange equation and inside the impainting domain the model simple employs anisotropic diffusion [13] based on the contrast of the isophotes. This model was designed for inpaintings of small regions and while it does a good job in removing noise, it does not connects broken edges (single lines embedded in a uniform background) [4]. The *Curvature-Driven Diffusion* (CCD) model [5] extended the TV algorithm to also take into account geometric information of isophotes when defining the “strength” of the diffusion process, thus allowing the inpainting to proceed over larger areas. While it can connect some broken edges, the resulting interpolated segments look blurry.

While nonlinear PDE-based image restoration methods have the potential to systematically preserve edges, the inpainting problem is very ill posed in general and fast numerical implementations are difficult to achieve [5]. It is equally hard to find appropriate mathematical models for inpainting [5]. Thus, despite its elegance, it is not clear if Bertalmio’s inpainting model is fully appropriate to the problem. A careful examination of the results presented in [2] (not reproduced here) reveals that sharp edges are not always preserved. For instance, the reconstructed region where the mask crosses the VW Beetle near the windshield appears blurred with broken edges (Figure 6 (top) [2]). Similar artifacts can be found in both restored images shown in Figure 6 [2]. Since no criteria have been defined for locally stopping the inpainting, the process is constantly applied to all masked pixels, regardless of the local smoothness of the region. As a result, computationally expensive operations might be unnecessarily performed, resulting in longer processing time.

2.1 Discussion

Principle 1: Given a region to be inpainted, the Shannon-Whittaker sampling theorem [8] imposes a fundamental restriction on the scale of the features that can be automatically restored by any inpainting model.

Thus, although it might still be possible to reconnect broken edges in some cases, exact reconstruction of the inpainted regions is only feasible for locally smooth areas. In practice, however, images can contain arbitrary spatial discontinuities. Even if some knowledge about the context of the region to be inpainted can be made available, one can only hope to produce a plausible rather than an exact reconstruction. Thus, in order for any inpainting strategy to be reasonably successful for a large class of images *the regions to be inpainted must be locally small*. As the inpainted regions become smaller, simpler diffusion models can be used to locally approximate the result produced by more expensive ones (geometrically, this is analogous to piecewise linear interpolation).

Principle 2: Chromatic perception and separation between regions is affected by the gradient between them [9]. As a result, the human visual system is very sensitive to edges, but can tolerate certain amounts of blurring elsewhere.

There is also a relative independence between the form of a surface and the microstructure of its contour [9]. Smaller inpainting domains also help to hide reconstruction errors in areas of the image containing stochastic textures, such as the leaves of the tree in Figure 6.

3 A Fast Digital Inpainting Algorithm

Let Ω be a small area to be inpainted and let ∂Ω be its boundary. Assuming Ω is small, any simple diffusion algorithm can be used to propagate information from ∂Ω into Ω, with a bounded error. A slightly improved version of this simple algorithm would reconnect edges reaching ∂Ω (for instance, using an approach similar to the one described in [12]), remove the new edge pixels from Ω (thus splitting Ω into a number of smaller sub-regions), and then perform the diffusion process as before.

We devised a simple diffusion algorithm, which consists of initializing Ω by clearing its color information and repeatedly convolving (since we use rotationally symmetric kernels, a simpler correlation operation [7] suffices) the region to be inpainted with a diffusion kernel. ∂Ω is a one-pixel thick boundary and the number of iterations is a user-specified parameter. Figure 2 (left) shows the pseudocode of the algorithm. Despite its simplicity, it performs surprisingly well (or, given the principles discussed in section 2.1, not so surprisingly). All reconstructed images shown in this paper were obtained with this algorithm or with a minor variation of it.

```
initialize Ω;
for (iter =0; iter < num_iteration; iter++)
  convolve masked regions with a diffusion kernel;
```

a	b	a
b	0	b
a	b	a

c	c	c
c	0	c
c	c	c

Fig 2. Left: Pseudocode for the fast inpainting algorithm.. Right: different diffusion kernels used with the algorithm. a = 0.073235, b = 0.176765, c = 0.125.

3.1 Preserving Edges

This simplest version of the algorithm can introduce some artifacts as Ω crosses the boundaries of high contrast edges (Figure 3(b)). Such artifacts are visible on the edges between black regions and horizontal white bars in the façades of the buildings shown in Figure 4. Other inpainting algorithms are also unable to preserve all kinds of edges.

In practice, the intersections between Ω and high contrast edges are the only places where anisotropic diffusion is required, and they account for a very small percentage of the image area. Building a mask is an important and the most time-consuming step of the inpainting process, requiring user intervention. Since

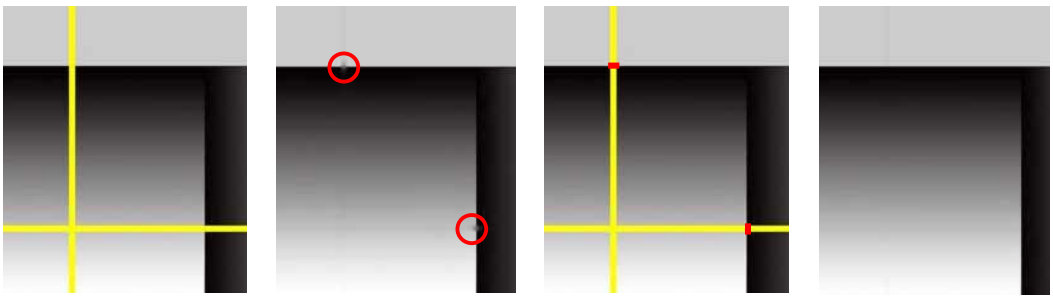

Fig. 3. (a) Image to be inpainted. (b) Result of the isotropic diffusion introduces some blurring along high contrast edges. (c) The user-added diffusion barriers. (d) Result.

our algorithm can inpaint an image in just a few seconds, it can be used for interactive construction of tight masks[1]. We exploit this interactivity to easily implement edge reconnection by defining, during the mask construction process, diffusion barriers. This accomplishes a result similar to anisotropic diffusion, but without the overhead of applying it to every single pixel. Figure 3 illustrates the idea. Figure 3(a) shows a portion of an image to be inpainted. The simple diffusion-based inpainting algorithm produces blurred regions at the intersections between Ω and high contrast edges indicated by the small circles in Figure 3(b). By appropriately adding diffusion barriers (line segments across the mask), the user stops the diffusion process from mixing information from both sides of the mask (see little edges shown in Figure 3(c)). The diffusion barriers are then used in conjunction with the mask to produce the result shown in Figure 3(d).

4 Results

We have implemented the algorithm described in Figure 2 in C++ and tried two different diffusion kernels (Figure 2 (right)). In both cases the results were similar. All images shown in the paper were generated using a 450 MHz Pentium III PC with 128 MB of memory running Windows98 and using the leftmost kernel shown in Figure 2. In all cases, 100 diffusion iterations were used. Results shown in Figures 4, 6, 7 and 8 were produced with the simplest diffusion version of the algorithm. For Figure 1, two diffusion barriers were used: one at each boundary between Lincoln's hair and the background. For Figure 5, we used our own generated mask and four diffusion barriers: one between the left white border and the gray background, one between the background and the left side of each of the two bigger girls' faces, and one between the right arm and the dress of the girl in the center.

The cost of inpainting is linear on the size of the inpainted region and algorithms are cache intensive. For Lincoln's portrait, the inpainting time of our algorithm was 0.61 seconds. Yung and Shankar [15] have reported an inpainting time of 2 minutes and 25 seconds for the same input image using an implementation of the algorithm described in [2], on a 450 MHz Pentium II PC with 128 MB of memory. Despite the longer time, the results they obtained exhibits blurry spots on the masked boundary between the hair and the background [15]. Figures 4 (left) and 5 (left) were used in Bertalmio et al original paper [2] and were obtained from Bertalmio's web site [3]. For the example shown in Figure 5, Bertalmio et al. [2] reported an inpainting time of approximately 7 minutes, or 2 minutes when a two-level multiresolution approach is used. These times were measured on a 300 MHz Pentium II PC (128 MB of memory under Linux). The image shown in Figure 5 (right) was produced with our algorithm in 1.21 seconds.

Figures 6, 7 and 8 illustrate different kinds of features found in actual photographs. Figure 6, shows a 640x480-pixel photograph exhibiting uncorrelated high frequencies represented by the leaves of the trees. It was superimposed with a textual mask (18 pt font size) covering 18.77% of its original area. The restored image, obtained in 6.37 seconds, essentially recovers all details of the original picture. For instance, notice the children playing in the back, as well as the details of the doors, windows and columns. Figure 7 shows a 640x480-pixel image containing very few high contrast edges, but with 14.54% of its area scratched. The image shown on its right was recovered in 5.87 seconds. Finally, Figure 8 shows an underwater scene (512x384 pixels) containing a large number of high contrast edges and superimposed with a mask covering 16.19% of its area. Figure 8 (right) was reconstructed in 4.06 seconds. Notice that such a reconstruction is mostly fine, except for branches that became disconnected on the top right portion of the image. Due to the relatively small scale of some masked branches, other inpainting techniques will likely to fail to connect these edges. Table 1 summarizes the inpainting results obtained on two different systems.

The quality of an inpainting is a subjective issue. One possible way to quantify the quality of a reconstruction is to start with a known image, mask some of its areas, restore them and compare the result with the original one. However, error measurements should take into account a perceptual metric [14] as opposed to just some Euclidean metric applied to the RGB color space. Unfortunately, perceptual metrics are still not completely reliable due to incomplete understanding of our visual system. Instead, we used the mean-square error (MSE) of the reconstructed region computed for the R, G and B channels as a measure of the quality of the reconstruction. MSE is frequently used in image processing to assess error. For the case of Figures 4 and 5, the MSE was computed against images restored by Bertalmio et al. and available at their web site [3]. The errors associated with the reconstruction of the images shown in Figures 6, 7 and 8 were computed using the original photographs as reference. The results are summarized in Table 2, sorted by increasing error. Notice that for the case of images not containing sharp color or intensity discontinuity

[1] Our system has a suit of image processing and editing tools that make the mask construction process even simpler.

(*e.g*, *three girls* and *baby Lu*) the error is very small. In particular, for the case of the three girls, our result is virtually indistinguishable from the one produce by Bertalmio et al.

As expected, images containing large amounts of high frequencies (Yard and Underwater), present larger reconstruction errors. Despite the error values, the reconstructed images look indeed very good (Figures 6 (right) and 8 (right)). Although we have not performed any rigorous experiments, we have informally shown these results on a monitor screen and asked several people if they could find any problems in these images. Nobody reported problems with the reconstructed picture of the yard. In this case, the high frequency regions correspond to the trees leaves, which due to its stochastic nature help to mask the error. In the case of the underwater image, the error is again distributed across all high frequency regions. However, it only seems to be noticeable in areas containing predictable high contrast edges, such as the branches on the top right portion of the image. Careful observers can identify these disconnected and blurred edges.

Table 1. Inpainting time measured using two systems: a 450 MHz PIII PC and a 1 GHz Athlon PC

Image	Time PIII 450 MHz	Time Athlon 1 GHz	Image	Time PIII 450 MHz	Time Athlon 1 GHz
Lincoln	0.61 sec.	0.30 sec.	*Yard*	6.37 sec.	1.90 sec.
New Orleans	2.53 sec.	0.71 sec.	*Baby Lu*	5.87 sec.	1.70 sec.
Three girls	1.21 sec.	0.49 sec.	*Underwater*	4.06 sec.	1.11 sec.

Table 2. Mean Square Error for the RGB channels of the restored images

Image	MSE r	MSE g	MSE b	Masked pixels
Three Girls	33.88	33.88	33.88	9,264
Baby Lu	61.32	66.9	72.40	42,061
New Orleans	347.53	269.57	290.59	20,795
Yard	729.76	725.73	732.90	57,688
Underwater	802.10	589.26	510.06	31,831

5 Conclusions and Future Work

A recent interest for image restoration has lent to the creation of a few PDE-based inpainting models [2] [4] [5], which are relatively expensive to evaluate. Signal theory constrains the quality of reconstruction that can be achieved by any automatic inpainting procedure regardless of its underlying mathematical model. In order to be applicable to images containing arbitrary contents, the inpainting domain must be kept locally small. Moreover, small domains of irregular shapes help to the reconstruction error in areas containing stochastic textures. Based on these facts, we have presented a very simple isotropic diffusion model extended with the notion of user-provided diffusion barriers. Such a simple model produces results comparable to previously known non-linear models, but are two to three orders of magnitude faster, thus making inpainting practical for interactive applications. For instance, our algorithm can be easily incorporated as a plug-in to standard image tools such as Photoshop.

Ideally, the mask Ω should include exactly the region to be retouched. If smaller, $\partial\Omega$ will contain spurious information, which will be carried into the restored area. If bigger, some possibly important information might be discarded. Since changes in Ω imply changes in $\partial\Omega$, changes can be expected to produce different results. Being able to create and refine Ω interactively can greatly improve the quality of the reconstruction and avoid error correlation.

Although diffusion barriers could be used to reconnect edges in Figures 4 and 8, an automatic procedure similar to the one described by Nitzberg et al [12] is preferable and we are currently investigating alternative solutions to this problem. It is desirable to be able to fill in areas of textured surfaces as a way to relax the constraint on the local size of the inpainting domains imposed by the sampling theorem. We are currently investigating the integration of our approach with a fast texture synthesis algorithm for natural textures [1].

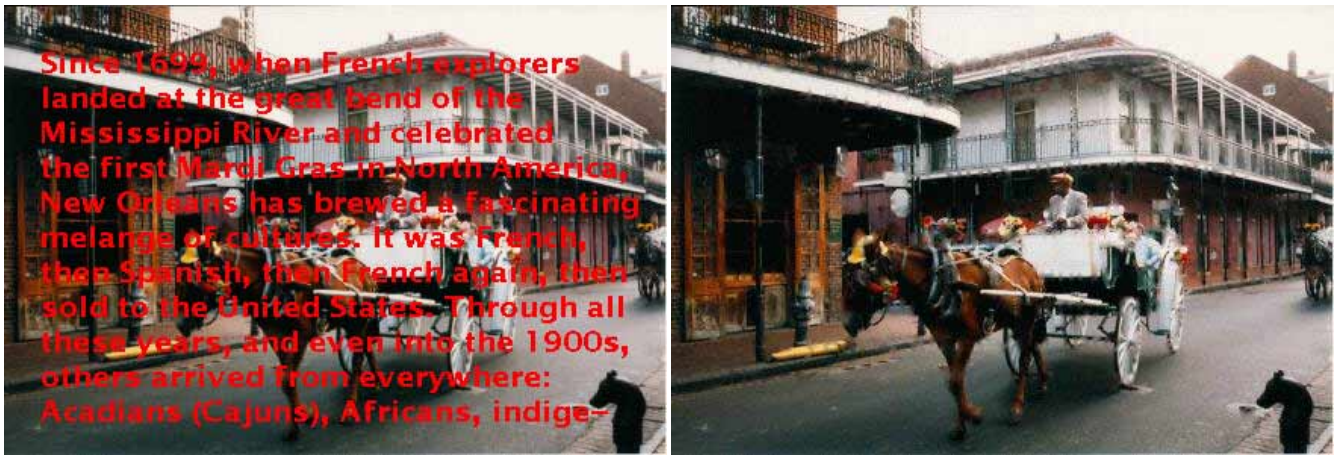

Fig. 4.Left: Picture with superimposed text (courtesy of Marcelo Bertalmio [3]). Right: Restored image obtained with our algorithm.

Fig. 5.Left: Old photograph (courtesy of Marcelo Bertalmio [3]). Right: Restored image obtained with our algorithm.

References

1. Ashikhmin, M. Synthesizing Natural Textures. In *2001 Symposium on Interactive 3D Graphics*, pages 217-226.
2. Bertalmio, M, Sapiro, G., Caselles, V., Ballester, C. Image Inpainting. *SIGGRAPH 2000*, pages 417-424.
3. Bertalmio, M. Marcelo Bertamio's web site: http://www.ece.umn.edu/users/marcelo/restoration.html
4. Chan, T., Shen, J. Mathematical Models for Local Deterministic Inpaintings. UCLA CAM TR 00-11, March 2000.
5. Chan, T., Shen, J. Non-Texture Inpainting by Curvature-Driven Diffusions (CCD). UCLA CAM TR 00-35, September 2000.
6. Chan, T., Mulet, P. Iterative Methods for Total Variation Image Restoration. UCLA CAM TR 96-38, October 1996.
7. Efford, N. Digital Image Processing: a practical introduction using Java. Addison-Wesley, 2000.
8. Gomes, J., Velho, L. Image Processing for Computer Graphics. Springer-Verlag, 1997.
9. Kanizsa, G. Organization in Vision: Essays on Gestalt Perception. Praeger Publishers, 1979.
10. Kokaram, A. Morris, R., Fitzgerald, W., Rayner, P. Interpolation of Missing Data in Image Sequences. IEEE Transactions on Image Processing, 11(4), pages 1509-1519, 1995.
11. Mark, W. Post-Rendering 3D Image Warping: Visibility, Reconstruction and Performance for Depth-Image Warp. PhD. Dissertation, UNC Chapel Hill, April 1999.
12. Nitzberg, M., Mumford, D., Shiota, T. Filtering, Segmentation and Depth. Lecture Notes in Computer Science Number 62, Springer-Verlag, 1993.
13. Perona, P. Malik, J. Scale-space and edge detection using anisotropic diffusion. IEEE-PAMI 12, pp. 629-639, 1990.
14. Ramasubramanian, M. Pattanaik, S., Greenberg, D. Perceptually Based Physical Error Metric for Realistic Image Synthesis. *SIGGRAPH 1999*, pages 73-82.
15. Yung, W., Shankar, A. Image Inpainting project. http://www.people.fas.harvard.edu/~shankar2/inpainting/
16. Wei, L., Levoy, M. Fast Texture Synthesis using Tree-structured Vector Quantization. *SIGGRAPH 2000* pp 479-488.

Color Plates

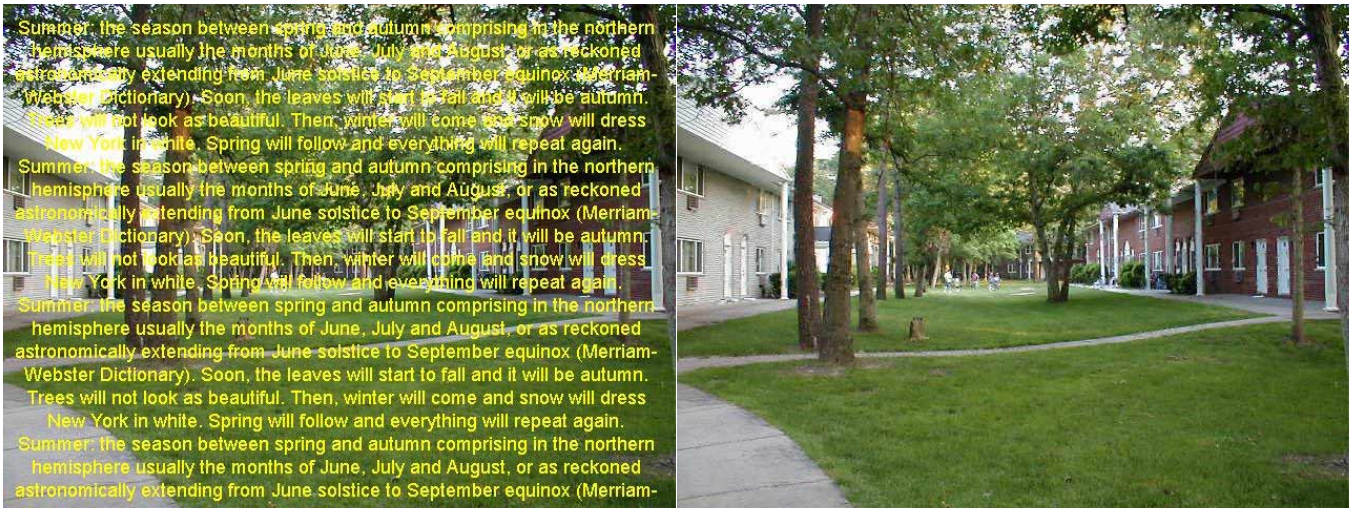

Fig. 6. Yard. Left: Image containing uncorrelated high frequency with text covering 18.77% of its area. Right: restored image obtained with our algorithm. Notice the children playing in the back, and the details of the doors, windows and columns.

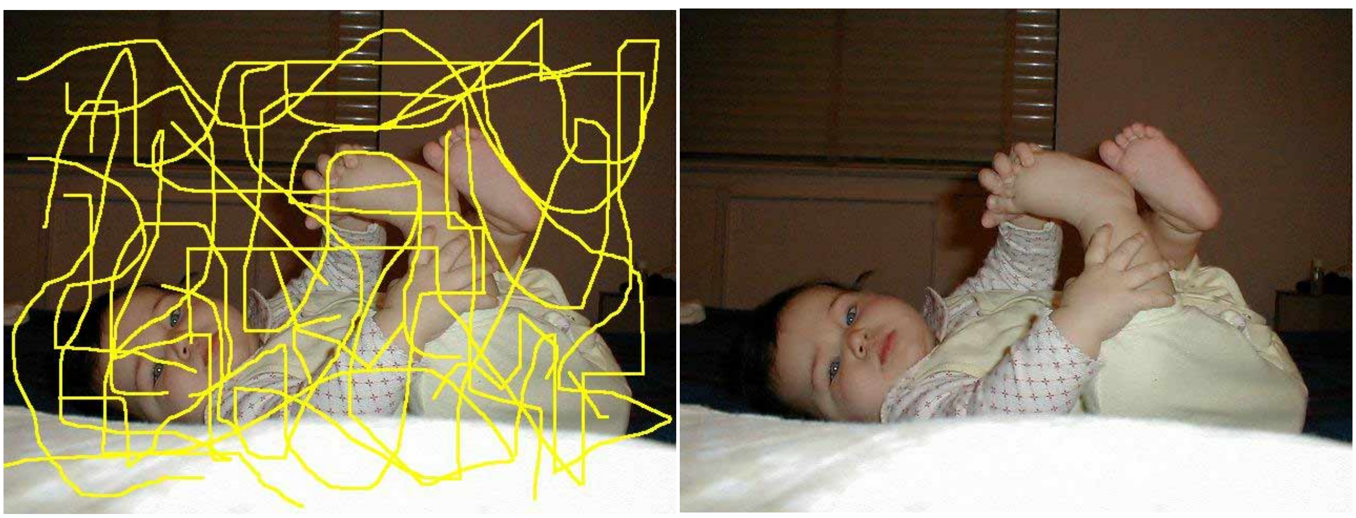

Fig. 7. Baby Lu. Left: Image containing few high contrast edges. The mask covers 14.54% of its area. Right: restored image.

Fig. 8. Underwater. Left: Image containing many high contrast edges, with text covering 16.19% of its area. Right: restored image. Although the error is distributed across all high frequency regions, it is noticeable at the broken and blurred white edges on the top right.