PYRAMID MODEL BASED DOWN-SAMPLING FOR IMAGE INPAINTING

Miaohui Wang, Bo Yan *

School of Computer Science, Fudan University
Shanghai, PRC 200433
Email: {miaohwang, byan}@fudan.edu.cn

Hamid Gharavi

National Institute of Standards and Technology
100 Bureau Drive, Gaithersburg, MD USA
Email: gharavi@nist.gov

ABSTRACT

Image inpainting is a useful and powerful technique for automatically restoring or removing objects in films and damaged pictures. In the last ten years, many excellent inpainting algorithms have been proposed after Bertalmio *et al.* [1]. However, no paper systematically and theoretically analyzes the factors, which limit the performances of existing algorithms. Based on extensive experiments, we firstly construct a universal framework for image inpainting, which contains three crucial factors — *Area, Shape, and Perimeter (ASP)*. Then we propose a Pyramid model based Down-sampling Inpainting (PDI) model according to the ASP principles. Experimental results show that the performances of existing methods can be tremendously improved after incorporating the PDI model.

***Index Terms*—** Image inpainting, Pyramid model, Down-sampling.

1. INTRODUCTION

Image inpainting was introduced by Bertalmio *et al.* [1]. After their pioneering research, image inpainting algorithms experienced significant progress. Bear in mind that inpainting technology has been widely applied in various fields, such as specific objects removal [2, 3], old photograph restoration [4, 5], film retouching [6], and digital zooming [7]. In this article, we review the inpainting algorithms from two aspects: non-texture and texture based inpainting methods.

The non-texture based inpainting method aims to restore the damaged areas regardless of its texture information, as described in [1,3,5]. Therefore, the non-texture based method will inevitably degrade in the center of the inpainting hole, if the diameter of the hole becomes big enough.

Non-texture algorithms mainly include *Partial Differential Equation (PDE)* based methods and *Total Variation (TV)* based methods. To some level, TV based algorithms also associate with some solutions of PDE equations.

PDE based methods, such as [1, 3], belong to pixel based inpainting technology. The method proposed in [1] uses a second order PDE equation to restore the damaged areas, which expands the information from the boundary areas along isophote lines. Bertalmio *et al.* [3] employed the Navier-Stokes equation to hold the continuity of isophote lines from the boundary to the center of the inpainting holes.

TV based methods, such as [7–9], utilize a noise removal mathematical model with low computational complexity for image inpainting. Rudin *et al.* [8] introduced this model to remove image noises. In [7], Chan *et al.* developed the TV based method and employed new principles, e.g. human visual perception, to restore the edges. Also, Chan *et al.* proposed a new information propagation model — the *Curvature Driven Diffusions (CDD)* scheme in [9], which tries to restore the connectivity of far apart corrupted areas.

The texture based inpainting method can restore the damaged area considering the textures of the damaged images. Literatures like [10, 11] combined *texture synthesis* and traditional inpainting algorithms to restore damaged pictures. For example, Yamauchi *et al.* [10] employed *Discrete Cosine Transform (DCT)* to separate the original image into low frequency and high frequency parts. They arranged these two kinds of frequencies with different schemes: low frequency components using traditional non-texture method, and high frequency components using texture synthesis. Although texture based inpainting methods can achieve better restoration results than non-texture ones, they are time-consuming and suffer high computational complexity, which limit widespread use.

Besides the above two main inpainting classes, there are other ponderable inpainting methods. For example, [5, 12] employ different methods directly to restore the damaged image instead of iteration in [1]; Oliveira *et al.* [5] employ a small filter to propagate the boundary information to the holes. In [12], Sun *et al.* design an interesting scheme to complete damaged images, which first restores the structure of objects and then other small parts isolated by the already restored structure.

Our contributions: firstly, we extract three factors which play an important role in existing pixel based schemes. These are the *Area*, *Shape* and *Perimeter* of the inpainting hole. Secondly, we construct a novel optimization model —— *Pyramid model based Down-sampling Inpainting (PDI)* model for im-

*This work is supported by NSFC (Grant No.: 60703034).

age inpainting. After incorporating PDI, the performances of previous pixel based inpainting algorithms can be improved significantly. Experimental results show that our PDI method can both accelerate the traditional inpainting algorithms and enhance their performance.

The reminder of this paper is organized as follows. In Section 2, we analyze the constraints of image inpainting and propose our PDI model. Then we evaluate the proposed model by simulations and present the results in Section 3. Finally, in Section 4, we draw the conclusions.

2. ANALYSIS OF INPAINTING CONSTRAINTS AND OUR PROPOSED PDI MODEL

2.1. Analysis of Inpainting Constraints

We start our research from three aspects: the *Area, Shape* and *Perimeter* of the Inpainting Regions.

When the total area of inpainting region A_{rea} remains unchanged, there are three factors which affect restoring performances:

$$A_{rea} = \frac{1}{2}\sum_{i=1}^{n}\oint_{M_{\partial\Omega^i}} x \cdot dy - y \cdot dx \tag{1}$$

where $M_{\partial\Omega^i}$ represents the boundary of the i^{th} hole with the total number n.

a).*The distribution of the inpainting regions:* firstly, let's employ two new definitions: *Sparse Distribution and Dense Distribution*. We consider the inpainting picture as a *multiply connected region (MTR)* and holes in the MTR are equivalent to each connected inpainting area. Sparse Distribution denotes that the number of inpainting holes is small, while dense distribution represents large amounts of inpainting holes. These two types measure the distribution of holes in the damaged pictures.

Why are the distributions of damaged regions significant for inpainting? Let's analyze the example shown in Fig. 1, where different layers (such as 2^0, 2^2, 2^4,) have different distributions. The illustration in Fig. 1 represents that dense distribution has more peripheral pixels, which can be utilized to directly restore the boundary damaged pixels. Theoretically, since current pixel based methods repair damaged pixels step by step from the boundary to the center of the hole, the boundary pixels of the hole can more easily achieve their authentic values than the central pixels.

Corollary 1. Dense distribution outperforms sparse distribution in the process of inpainting.

b).*The shape of the inpainting regions:* Fig.1 shows that different shapes affect the restoring results. We employ the definition of the *Inscribed Circle of the Hole (ICH)*. The ICH represents the biggest inscribed circle among the whole inpainting holes in a MTR. For pixel based inpainting methods, the *Radius of the Inscribed Circle of the Hole (RICH)* determines the complexity and accuracy of the restoring process:

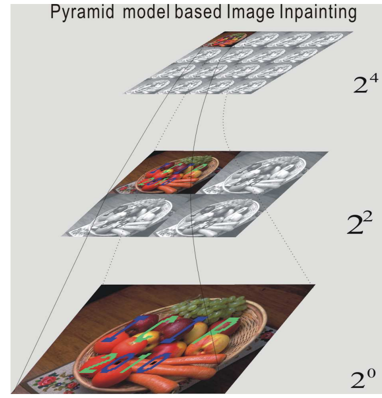

Fig. 1. Pyramid model based downsampling

$$R_{ICH} = max\{R_{\partial\Omega^1}, R_{\partial\Omega^2}, ...R_{\partial\Omega^n}\} \tag{2}$$

Let's give a concise proof: pixel based methods have similar steps. For example: the values of inner damaged pixels are obtained by their boundary pixels layer by layer along the direction of propagation from the boundary to the center of the hole. As we analyzed above, the center pixels of the hole degrade restoring performances.

Corollary 2. The bigger the RICH is, the worse performance restores in the center of the inpainting, and vice versa.

c).*The total perimeter of the inpainting regions:* in an MTR, it is easy to prove that the total $P_{erimeter}$ will increase when the MTR is cut into small MTRs. For instance, a square with the dimension $2m \times 2m$ is cut into two rectangles with the dimension $m \times 2m$. As shown in Fig. 1, we find that a picture with a constant damaged A_{rea} will obtain different restoring results under the same condition when the total perimeters are different.

$$P_{erimeter} = \sum_{i=1}^{n}\oint_{M_{\partial\Omega^i}} \sqrt{1 + y'(x)}dx \tag{3}$$

Corollary 3. The longer the total perimeter is, the better the restoring results are.

Taking Bertalmio *et al.* [1] as an example, the restoring process is expanded from the boundary pixels to the center ones by iterations. Theoretically, when iterations increase, the restoring results will gradually approaching a better visual quality. Actually, it should be noted that the restoring performance will change slightly after the first several iterations (As shown in Fig. 2, we find that the restoring results have little variation among 200∼3000 iterations.).

Bertalmio's [1] algorithm consists of two important steps:

$$\begin{aligned} I^{n+1}(i,j) &= I^n(i,j) + \Delta t I_t^n(i,j) \\ I_t^n(i,j) &= \nabla \overrightarrow{L}^n(i,j) \cdot \overrightarrow{N}^n(i,j), \forall (i,j) \in \Omega \end{aligned} \tag{9}$$

Fig. 2. Results of [1] and PDI based [1]. iteration=200, 600, 1000, 1500, 2000, 3000, 200 (PDI based [1]), original picture

Table 1. The PDI algorithm

Algorithm *PDI*

BEGIN
% i denotes the i^{th} pyramid layer, PID denotes the maximum pyramid layer, $M^{(0)}$=I

$$M^{(i)} = \{(x, y)|x, y \in I\} \tag{4}$$

IF $i <= PID$ Pyramid($M^{(i)}$) **END**
END

Pyramid($M^{(i)}$**)**
BEGIN

$$M_1^{(i+1)} = \{(x, y)|x = 2n, y = 2n, and \; x, y \in M^{(i)}, where, n = 1, 2, ...\} \tag{5}$$

$$M_2^{(i+1)} = \{(x, y)|x = 2n, y = 2n - 1, and \; x, y \in M^{(i)}, where, n = 1, 2, ...\} \tag{6}$$

$$M_3^{(i+1)} = \{(x, y)|x = 2n - 1, y = 2n, and \; x, y \in M^{(i)}, where, n = 1, 2, ...\} \tag{7}$$

$$M_4^{(i+1)} = \{(x, y)|x = 2n - 1, y = 2n - 1, and, x, y \in M^{(i)}, where, n = 1, 2, ...\} \tag{8}$$

END

$I\,(i, j)$ is the pixel value at position(i, j). The superscript n denotes the n^{th} iteration. $I_t^n\,(i, j)$ denotes the correction in the n^{th} iteration and is achieved by equation (9). $\nabla \overrightarrow{L}^n\,(i, j)$ is a Laplacian smoothness estimator which is used to smooth the variation between neighbor pixels. $\overrightarrow{N}^n\,(i, j)$ is the direction of propagation. Δt is the refresh rate. The program steps out at the pixel(i, j) when $I_t^n\,(i, j)$ equals zero.

2.2. Our Proposed PDI Model

When a given picture needs to be restored, shapes, perimeters and distributions of its holes remain fixed. If we want to improve the efficiency and accuracy of algorithms for image inpainting, we must transform the picture in order to conform to either *corollary (1)*, *(2)* or *(3)*. It should be noted that these transforms can not involve frequency domain which will lead to a mismatch between the restored image and its mask. Let's consider a pyramid model based down-sampling, as shown in Fig. 1. For example, the original picture $M^{(i)}$ will be transformed into four small pictures $M_1^{(i+1)}$,$M_2^{(i+2)}$,$M_3^{(i+3)}$ and $M_4^{(i+4)}$ by selecting a different combination of pixels along the rows and columns. Obviously, these four small pictures can completely reconstruct the original one. The algorithm of PDI is listed in Table 1.

From Fig.1, we can find that PDI meets these three corollaries: *a)* the distribution of the holes becoming more dense; *b)* the RICH decreasing; *c)* the total perimeters of these holes increasing. Thus the values of boundary pixels can easily propagate to central pixels in these holes after incorporating PDI. As a result, the speed and accuracy of image inpainting methods can be improved drastically (shown in Section 3).

(a) iteration= 50, 100, 200, 50 (PDI based [5], 20, 15, 10, 5)

(b) iteration=325, 1000, 325 (PDI based [7], 150, 100, 50, 25)

Fig. 3. Simulation results. (a) [5] and PDI based [5]; (b) [7] and PDI based [7].

Fig. 4. Results of [1] and PDI based [1]. iteration= 100, 200, 300, 600, 60 (PDI based [1]), original picture.

3. SIMULATION RESULTS

We implemented our method and other classical inpainting algorithms [1, 5, 7] on *Pentium IV* PC (RAM 512M and CPU 1.65G Hz). The samples used in the experiments are available

(a) iteration= 20, 60, 20 (PDI based [5], 10, 5, 5)

(b) iteration= 100, 500, 100 (PDI based [7], 50, 30, 20)

Fig. 5. Simulation results. (a) [5] and PDI based [5]; (b) [7] and PDI based [7].

(a) (b)

Fig. 6. **(a):** Zoom ins from Fig. 2 and Fig. 3. Left side: iteration=325, 100, 3000 for [7], [5] and [1] from top to bottom; Right side: iteration=325, 50, 200 for PDI based [7], [5] and [1] respectively. **(b):** Zoom ins from Fig. 5 and Fig. 4. Left side: iteration=500, 60, 600 for [7], [5] and [1] from top to bottom; Right side: iteration=100, 20, 60 for PDI based [7], [5] and [1] respectively.

on the Internet.

Fig. 2 illustrates the performance of Bertalmio *et al.* [1] under different iterations and PDI based [1] with the total iterations 200 (with 100, 50, 30, 20 iterations for different pyramid layers 2^8, 2^4, 2^2 and 2^0 respectively). Fig.3 represents comparison results of Oliveira *et al.* [5] and Chan *et al.* [7] with different iterations. As we can observe from these figures, our PDI method is able to save more time with better inpainting visual quality than [1], [5], and [7] (the PDI method needs less iterations, while each iteration consumes one unit time).

Fig. 4 shows the restoring results of another picture by Bertalmio *et al.* [1]. Here the PDI based [1] with total 60 iterations (with 30, 20 and 10 iterations for different pyramid layers 2^4, 2^2 and 2^0 respectively) shows the advantages. It is obvious that PDI has a better restoring result by taking fewer iterations than that of [1]. Fig. 5 shows the results of Oliveira *et al.* [5] and Chan *et al.* [7]. We can see that the inpainting performances are improved significantly after incorporating the proposed PDI model.

4. CONCLUSION

Starting from Bertalmio's pioneering work [1], new concepts for inpainting algorithms, which employ more and more complex mathematical models [2, 4], have been designed. Unfortunately, none focus on what contributes to retarding the inpainting performance. Based on a rigid analysis of the classical inpainting theory, we firstly propose ASP principles to measure the inpainting technology, which is useful for directing future inpainting research. In addition, we propose a PDI method to testify the accuracy of ASP principles. Experimental results represent that our PDI method can greatly improve the efficiency and performance of previous pixel based inpainting algorithms, which can help to commercialize these classical time consuming inpainting algorithms.

5. REFERENCES

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of Computer graphics and interactive techniques, SIGGRAPH 2000,*, 2000, pp. 417–424.

[2] K.A. Patwardhan, G. Sapiro, and M. Bertalmío, "Video inpainting under constrained camera motion," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 545, 2007.

[3] M. Bertalmio, A.L. Bertozzi, G. Sapiro, et al., "Navier-stokes, fluid dynamics, and image and video inpainting," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, 2001, vol. 1.

[4] F. Bornemann and T. Marz, "Fast image inpainting based on coherence transport," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 3, pp. 259–278, 2007.

[5] 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 (VIIP 2001)*, 2001, pp. 261–266.

[6] K.A. Patwardhan and G. Sapiro, "Projection based image and video inpainting using wavelets," in *IEEE International Conference on Image Processing (ICIP 2003)*, 2003, pp. 857–860.

[7] T.F. Chan and J. Shen, "Mathematical models for local nontexture inpaintings," *SIAM Journal on Applied Mathematics*, vol. 62, no. 3, pp. 1019–1043, 2001.

[8] LI Rudin, S. Osher, C. Inc, and CA Santa Monica, "Total variation based image restoration with free local constraints," in *IEEE International Conference Image Processing (ICIP 1994)*, 1994, vol. 1, pp. 31–35.

[9] T.F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436–449, 2001.

[10] H. Yamauchi, J. Haber, and H.P. Seidel, "Image restoration using multiresolution texture synthesis and image inpainting," *Proc. Computer Graphics International (CGI 2003)*, vol. 8, pp. 120–125, 2003.

[11] H. Grossauer, "A combined pde and texture synthesis approach to inpainting," *Lecture Notes in Computer Science*, pp. 214–224, 2004.

[12] Jian Sun, Lu Yuan, Jiaya Jia, and Heung yeung Shum, "Image completion with structure propagation," *ACM Transactions on Graphics*, vol. 24, pp. 861–868, 2005.