AN ANALYTICAL STUDY OF DIFFERENT IMAGE INPAINTING TECHNIQUES

SUPRIYA CHHABRA
IT Dept., 245, Guru Premsukh Memorial College of Engineering, Budhpur Village
Delhi- 110036
supriyachhabra123@gmail.com

RUCHIKA LALIT
CSE Dept., 245, Guru Premsukh Memorial College of Engineering, Budhpur Village
Delhi- 110036
Delhi
ruchikalalit@gmail.com

Dr. S.K. SAXENA
CSE Dept., Delhi Technological University, Shahbad Daulatpur, Main Bawana Road,
Delhi-110042
sksaxena@dce.ac.in

Abstract

Inpainting is the technique of filling in holes in an image to preserve its overall continuity. Applications of this technique include the restoration of old photographs and damaged film; removal of superimposed text like dates, subtitles, or publicity; and the removal of entire objects from the image like microphones or wires in special effects. In this paper, we analyze different digital inpainting algorithms for still images. The simultaneous propagation of texture and structure information achieved. The texture image repaired by the exemplar –based method; for the structure image, the Laplacian operator is used to enhance the structure information. The Laplacian image is inpainted by the exemplar-based algorithm and the Poisson equation based reconstruction is applied thereafter. In 8 pixel neighborhood method, central pixel value is identified by investigating surrounded 8 neighborhood pixel properties like color variation, repetition, intensity and direction. Finally, in 2e based inpainting technique, original image analyzed at encoder side so that some blocks removed during encoding. At decoder side, the image is restored by 2e-based inpainting and texture synthesis. Finally, we compare the computational cost of all the algorithms.

Keywords : Image inpainting; image restoration

1. Introduction

In real world, many people need a system to recover the damaged photographs, artwork, designs, drawings etc. Damage may be due to various reasons like scratches, overlaid text or graphics, scaled image etc. Traditionally, inpainting has been done by professional artists. However, we could not expect the accuracy and quality if it was done by human and time-consuming process. Image inpainting is an important element in image restoration study. It makes use of the information not lost of the image to fill the lost or damaged part according to certain rules, so that after the inpainting, the images are close to mathematical point of view, it is to repair image in the regions of blank area in accordance with the information around them. Digital repair technology was introduced earliest by Bertalmio *[Bertalmio et al. (2000)]*. After that, it is widely used in image processing, visual analysis and film industries. At present, the image inpainting technology is a hotspot in computer vision and computer graphics, and has an important value in heritage preservation, film and television special effects production, removing redundant objects. This paper presents various algorithms used either for removing objects from digital photographs and replacing them with visually plausible backgrounds or filling the holes in the images. Digital techniques are ranging from attempts to fully automatic detection and removal of scratches in film, all the way to software tools that allow a sophisticated but mostly manual process. In this paper, we com pares the computational cost following image inpainting algorithms:

- Object removal by exemplar based inpainting method
- Poison Equation method
- 8-pixel neighborhood fast sweeping method.
- 2e-Based inpainting method

2. Object Removal by Exemplar-Based Inpainting

The algorithm is based on an isophote-driven image sampling process. The exemplar-based approaches perform well for two-dimensional textures and for propagating extended linear image structures*[Criminisi et al. (2003)]*.

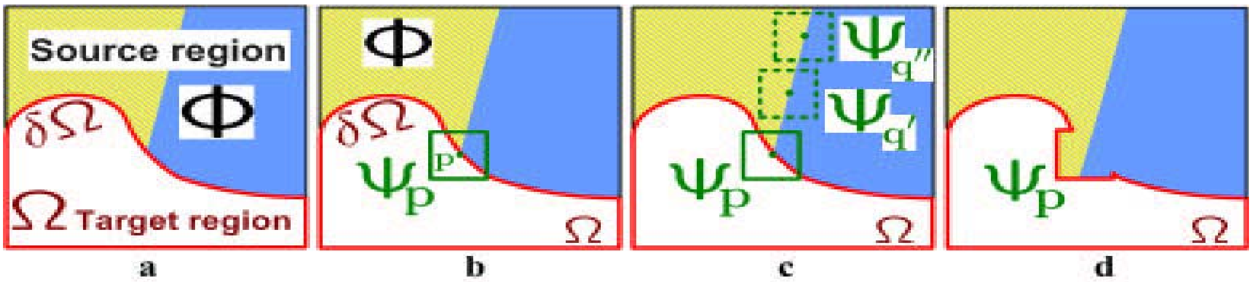

Fig 1*[Criminisi et al. (2003)]*: Structure propagation by exemplar-based texture synthesis.(a) Original image, with the target region Ω, its contour δΩ and the source region Φ clearly marked. (b) We want to synthesize the area delimited by the patch Ψp centred on the point p ∈ δΩ. (c) The most likely candidate matches for Ψp lie along the boundary between the two textures in the source region, e.g., Ψq' and Ψq'' . (d) The best matching patch in the candidates set has been copied into the position occupied by Ψp, thus achieving partial filling of Ω. The target region Ω has, now, shrank and its front has assumed a different shape.

A target region, Ω, is selected to be removed and filled. The source region, Φ, may be defined as the entire image minus the target region (Φ = I−Ω). The size of the template window Ψ must be specified. Each pixel maintains a colour value if in Φ region and empty if in Ω region and a confidence value, which reflects our confidence in the pixel value, and which is set once a pixel has been filled. Patches along the fill front are also given a temporary priority value to determines the order in which they are filled. The algorithm iterates the following three steps until all pixels are filled:

2.1. Computing patch priorities

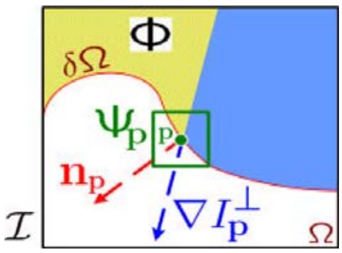

Fig 2*[Criminisi et al. (2003)]*: Notation diagram. Given the patch Ψp, np is the normal to the contour δΩ of the target region Ω and ∇I⊥ p is the isophote (direction and intensity) at point p. The entire image is denoted with I.

The priority computation is partial toward those patches which are on the continuation of strong edges and which are surrounded by high-confidence pixels. Given a patch Ψp centred at the point p for some $p \in \delta\Omega$ (fig. 2), its priority $P(p)$ is defined as the product of two terms:

$$P(p) = C(p)D(p). \qquad (1)$$

$C(p)$ the confidence term and $D(p)$ the data term are defined as follows*[Criminisi et al. (2003)]*:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \bar{\Omega}} C(q)}{|\Psi_p|}, D(p) = \frac{|\nabla I_p^{\perp} \cdot n_p|}{\alpha} \qquad (2)$$

$C(p)$ approximately enforces the desirable concentric fill order.Those patches which have more of their pixels already filled are filled first. The data term $D(p)$ defines function of the strength of isophotes on the front δΩ at each iteration. This term boosts the priority of a patch. This factor tends to synthesized linear structures first, therefore propagated into the target region.

2.2. Propagating texture and structure information

The patch $\psi_{\hat{p}}$ with highest priority is found. We then fill it with data extracted from the source region Φ. Search in the source region for that patch which is most similar to $\psi_{\hat{p}}$.

$$\psi_{\hat{q}} = \arg\min_{\psi_q \in \Phi} d(\psi_{\hat{p}}, \psi_q) \qquad (3)$$

2.3. Updating confidence values.

The *C(p)* is updated in the area enclosed by $\psi_{\hat{p}}$ after the patch $\psi_{\hat{p}}$ has been filled with new pixel values, by:

$$C(q) = C(\hat{p}) \quad \forall q \in \psi_{\hat{p}} \cap \Omega. \qquad (4)$$

3. Poisson-based image inpainting

3.1 Image decomposition

Image I_0 is decomposed into texture and structure image. For an image texture information is assumed to be noise as compared to structure information. A classical variational denoising algorithm i.e. total variation (TV) minimizing process *[Shao et al.]* is used for decomposition of the image. This algorithm yields sharp edges in the output image *I* while maintaining the fidelity to the original noisy input image I_0. The energy function with scalar fidelity controller λ is defined as*[Shao et al.]*:

$$E = \int_\Omega \left(|\nabla I| + \frac{1}{2}\lambda(I - I_0)\right) dxdy \qquad (5)$$

According to Euler- Euler-Lagrange equation,

$$\nabla\cdot\left(\frac{\nabla I}{|\nabla I|}\right) + \lambda(I_0 - I) = 0 \qquad (6)$$

The solution can be achieved by the gradient descent method, which means we solve

$$I^{(n+1)} = I^{(n)} + \nabla T\cdot\lambda\left(I_0 - I^{(n)}\right) + \nabla T\cdot\frac{I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_xI_yI_{xy}}{(I_x^2+I_y^2)^{3/2}} \qquad (7)$$

$$I_x = \frac{\partial I^{(n)}}{\partial x}, I_x = \frac{\partial I^{(n)}}{\partial y}$$

$$I_{xx} = \frac{\partial^2 I^{(n)}}{\partial x^2}, I_{yy} = \frac{\partial^2 I^{(n)}}{\partial y^2}, I_{xy} = \frac{\partial^2 I^{(n)}}{\partial x \partial y}$$

where $I^{(n)}$ is the result of the *n*th iteration, and *ΔT* is the step size. The structure image is extracted by increasing the number of iteration by assigning λ to a small value. Texture image can be extracted from the residual image.

3.2. Method of Poisson-based Structure image inpainting

The exemplar-based inpainting method performs well on texture information and is able to handle large holes. But, due to the direct patch duplication, the algorithm tends to produce block effect in processing of structure information degrading the visual effect of repaired images. To improve the performance of structure inpainting, the exemplar-based inpainting approach is combinedwith the Poisson equation. The theoretical foundation is that the Poisson equation is able to reconstruct a scalar function from a guidance field and a boundary condition*[Shao et al.]*. The Poisson equation can also be used as a least-squares minimization method, so block effect introduced by block duplication can be removed via reconstruction. First apply the Laplacian operator to the structure image and find the Laplacian field. In the Laplacian field edges are enhanced and backgrounds are almost completely removed. It provides a more accurate structure inpainting result when employing the exemplar-based method. Then the structure image is reconstructed by the Poisson equation taking inpainted Laplacian field as the guidance field.

4. Inpainting algorithm based on the 8-neighborhood fast sweeping method

Take a small neighborhood $B_\varepsilon(p)$ of size ε of the known image around p, for ε small enough, we consider a first order approximation $I_q(p)$ of the image in point p, given the image I (q) and gradient $\nabla I(q)$ values of point q*[Xul et al. (2009)]*:

$$I_q(p) = I(q) + \nabla I(q)(p-q) \qquad (8)$$

Next, we inpaint point p as a function of all points q in $B_\varepsilon(p)$ by summing the estimates of all points q, weighted by a normalized weighting function ω(p,q) .

$$I(p) = \frac{\sum_{q\in B_\varepsilon(p)} \omega(p,q)[I(q)(p-q)]}{\sum_{q\in B_\varepsilon(p)} \omega(p,q)} \qquad (9)$$

The weighting function ω(p,q) , is designed such that the inpainting of p propagates the gray value as well as the sharp details of the image over $B_\varepsilon(p)$.
To inpaint the whole Ω, we iteratively apply Eqn.(9) to all the discrete pixels of $\partial\Omega$, in increasing distance from $\partial\Omega_i$'s initial position $\partial\Omega_i$, and advance the boundary inside Ω until the whole region has been inpainted.
Implementing the above requires a method that propagates $\partial\Omega$ into Ω by advancing the pixels of $\partial\Omega$ in order of their distance $\partial\Omega$to the initial boundary $\partial\Omega_i$.
(1) Initialize: Set $u_{ij}=\infty$ for (i, j)∈Γ .For each γ ∈Γ set u(γ) = 0 . For z ∈Γ not in Ω_d compute the exact solution at the vertices of the grid cell in which z lies.
(2) For each sweep direction in {(x+, y+), (y-, x-),(x+, y-), (x, y+)} iterate through each grid pointaccording each of the sweep directions or according to the fast marching heap sort.
(3) At each grid E with index (i, j) If all the neighbors are infinity, skip. If there is at least one non-infinity neighbor in both the x- and y- direction, Let P and Q are the smallest one in each direction. Inpaint (p,q).

5. Edge-based inpainting

Edge extraction is first performed on the original image. Then, according to exemplar selection, some blocks will be removed and the others will be encoded. Here, the coded blocks are called exemplars because they will be used as exemples in inpainting and synthesis. For the removed blocks, corresponding edges will be encoded and transmitted. At decoder side, edge-based inpainting and texture synthesis are

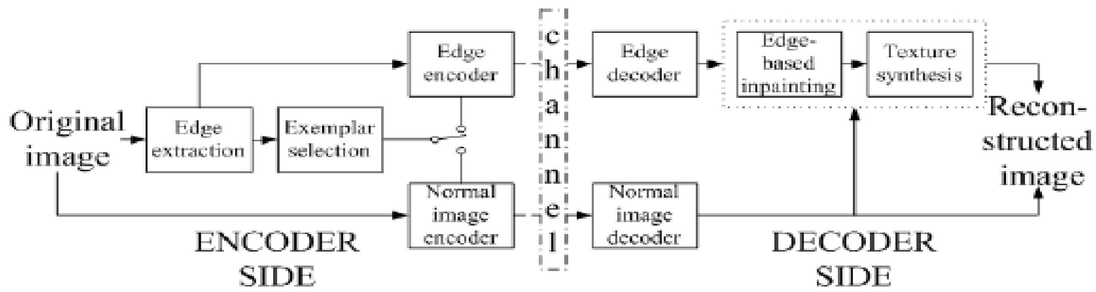

Fig. 1 *[dong liu et al. (2007)]* Image coding scheme.

edge-Based Inpainting is done in two steps.First, a linear interpolation, to generate the unknown pixels on the edge from the known ones on the same edge. Second, the neighborhood of an edge, known as influencing region, is progressively filled-in by pixel generation.

Conclusion

We apply algorithm to damaged frames in old films. The experiment results show that the inpainted images are visually pleasant and computational efficiency is improved in Successive Elimination Method. Exemplar Method works well for large objects. For single dimensions like line, Arc then Poisson method is useful. To improve the efficiency of filling one dimension and two dimension objects Poisson method is best. If the picture is like natural scenario then 8 pixel is good for computation speed and accuracy than Exemplar method. In 8 pixel -neighborhood fast sweeping method experimental result shows that there is substantive increase in the rate of image inpainting for small region i.e. One dimension and blur in large region. By using non-pixel based approach 2e based inpainting is better in visuals and acquire less memory space but the computation cost is high comparatively.

Type	The Computation Cost(sec)			
	Examplar	Poission	8-pixel neighhbourhood	Edge-based Inpainting
Natural Scenery 1	454	612	57	689
Natural Scenery 2	319	454	38	513
Batch Filling 1	227	317	50	369

References

[1] Criminisi, P. Pérez, and K. Toyama. “Object removal by exemplar-based inpainting.” in Proc. Conf. Computer Vision and Pattern Recognition, Madison, WI, June 2003.

[2] Dong Liu, Xiaoyan Sun, Feng Wu, ICME 2007, Edge-Based Inpainting And Texture Synthesis For Image Compression

[3] Jing Xu1, Daming Feng2, Jian Wu1, Zhiming Cui. 2009 International Conference on Communications and Mobile Computing: An Image Inpainting Technique Based on 8-NeighborhoodFast Sweeping Method.

[4] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, “Image inpainting”, in Proc. ACM SIGGRAPH, pp. 417-424, 2000.

[5] Xiaowei Shao, Zhengkai Liu, Houqiang Li. Department of Electronic Engineering and Information Science, University of Science and Technology: An Image Inpainting Approach Based on thePoisson Equation,