IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
ISSN: 2278-2834, ISBN: 2278-8735. Volume 4, Issue 5 (Jan. - Feb. 2013), PP 07-11
www.iosrjournals.org

Image Inpainting Realization of GUI in JAVA Using Exemplar Method

A. J. Thesiya[1], D. U. Shah[2], P. V. Kathiriya[3], G. B. Vasani[4], J. L. Vyas[5]
[1](EC, LJ Polytechnic, India)
[2](EC, Rk University, India)
[3](EC, Balaji institute, India)
[5](EC, Gujarat Technological University, India)

Abstract : *Image Inpainting technique is of filling in holes in an image to preserve its overall continuity. There are many applications of this technique include the restoration of old photographs and damaged film; removal of labeled text like dates, subtitles, or publicity; and the removal of entire objects from the image like microphones or wires in special effects; also repaire the scratches on pahotograph and image. The older techniques, having a problem, which has been addressed by two classes of algorithms: (I) "texture synthesis" algorithms for generating large image regions from sample textures, and (II) "inpainting" techniques for filling in small image gaps. Image inpainting techniques is the art of restoring the lost or scratch part of image and repaired it by using patch from the same image which is to be inpainted. The tool provides the GUI (Graphic User Interface) by which user can easily select the area which is to be inpaint. Then the tool is automatically inpaint the selected region according to the background information.*

Keyword – *Exemplar method, Flowchart,Implementation details, Introduction, Previous work,*

I. INTRODUCTION

Now a day's many applications are benefited from inpainting technology, e.g. object removal, image zooming and super-resolution, stain image recovering etc.

To develop a GUI (General User Interface) to inpaint selected regions from an image. Image Inpainting is the art of restoring lost parts of an image and reconstructing them based on the background information. The inpainting is derived from the ancient art of restoring image by professional image restorers in museums etc. Digital Image Inpainting tries to imitate this process and perform the inpainting automatically. Figure1show an example of this technique where a selected object (manually selected as the target region) is replaced by information from the remaining of the image in a visually plausible way. The algorithms automatically do this in a way that it looks "reasonable" to the human eye.

II. PREVIOUS WORK

The image inpainting field can be carried by many techniques so, in the following section we introduce the most related methods which used in that field. Bertalmio et al pioneered a digital image inpainting algorithm based on the PDE and be the extension to the level lines based disocclusions method or the same basic idea, after the user select the region to be inpainted the two methods iteratively propagate information from the outside of the area along the level lines isophotes (lines of equal gray values), the difference lies in the goal of maintaining the angle of arrival. In order to maintain the angle of arrival, the direction of the largest spatial change is used. The direction may be obtained by computing a discretized gradient vector and rotating this vector by 90 radians. Instead of using geodesic curves to connect the isophotes, the prolongation lines are progressively curved while preventing the lines from intersecting each other.

A Telea proposed a fast marching algorithm that can be looked as the PDE based approach without the computational overheads. It is considerably fast and simple to implement than other PDE based methods, this method produces very similar results comparable to other PDE methods. The algorithm propagating estimator that used for image smoothness into image gradient (simplifies computation of flow), the algorithm calculate smoothness of image from a known image neighbourhood of the pixel as a weighted average to inpaint, the FMM inpaint the near pixels to the known region first which is similar to the manner in which actual inpainting is carried out , and maintains a narrow band pixels which separates known pixels from unknown pixels, and also indicates which pixel will be inpainted next. The limitation of this method is producing blur in the result when the region to be inpainted thicker than 10 pixels.

III. STRENGTHS & LIMITATIONS OF DIFFERENT METHOD

All These methods are generally categorized into three basic groups. Algorithms proposed for Image Inpainting use the information from surrounded portions of image, which is selected to inpaint for the selected region.

1. The first groups of algorithms deals with the restoration of films.
2. The second groups of algorithms deals with the reconstruction of textures in the image.
3. The third group of algorithms that deal with disocclusionsFast Digital Image Inpainting

A.Fast digital inpainting

The algorithm's strength lies in its simplicity, being very easy to understand and implement. Although the algorithm is simple to implement, it gives decent results. This algorithm is very fast, and is able to preserve general color information and some low-frequency texture. The algorithm however, produces results which are quite blurry. More importantly, the visual continuity at the image edges seem to be pretty bad; if one looks enough, the visual discontinuity at the region boundaries can be quite easily seen.

B.Fast Marching Method

This method produced visually nearly identical results with other method. The runtime for this method was in all cases much shorter than other previous method. This C++ implementation took less than 3 seconds on an 800 MHz PC for an 800 × 600 color image with about 15% pixels to inpaint. The main limitation of this method is the blurring produced when inpainting regions thicker than 10 -15 pixels, especially visible when sharp isophotes intersect the region's boundary almost tangentially

IV. EXEMPLAR METHOD

Previous texture synthesis image inpainting algorithms require large amounts of user interaction. The key observations behind this algorithm are that:

1. Exemplar-based synthesis suffices. This means that exemplar-based texture synthesis does very well to preserve, propagate and extend linear image structure. Exemplar-based texture synthesis is based around finding patches in the rest of the image which are similar to the known parts of the image patch to be filled, and then copying the missing information over. This mechanism will do well to preserve and extend and image isophotes we had in the known parts of our target patch.
2. Filling order is critical. This means that the order in which exemplar-based synthesis is carried out has a large impact on the resulting quality.

Fig. 1 Removing large objects from an image. (a) Original Image (b) The object that had been selected by tool manually has been removed from the image and the information from background is merged into the missing region.

We must select a target region, Ω, to be removed and filled. The source region, Φ, may be defined as the entire image minus the target region (Φ = I- Ω), as a dilated band around the target region, or it may be manually specified.

The algorithm iterates the following three steps until whole selected region's pixels have been filled:

- Computing patch priorities
- Texture synthesis
- Filling order
- Updating confidence values

A. Computing patch priorities

The algorithm performs the synthesis task by a best-first filling strategy that depends entirely on the priority values that are assigned to each patch on the fill front. The priority computation is biased toward those

patches which: (i) are on the continuation of strong edges and (ii) are surrounded by high confidence pixels. Given a patch p centred at the point p for some $p \in \delta n$, we define its priority P(p) as the product of two terms:

$P(p) = C(p)D(p)$

C(p) the confidence term and D(p) the data term,

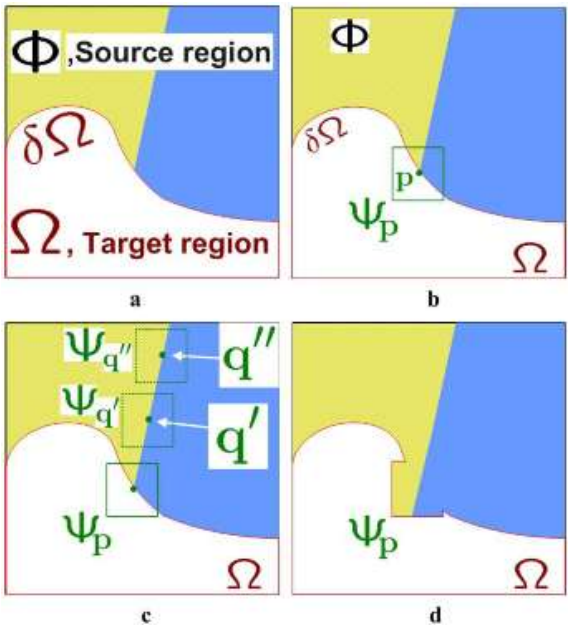


Fig. 2 Structure propagation by exemplar-based texture synthesis

In the above figure we can see that ϕ is Source region, Ω is Target region, δΩ is contour, Ψp is the Patch at point P. The area delimited by patch Ψp centred on point $P \in \delta\Omega$. The source region remains fixed throughout process and provides samples used in the filling process.

B. Propagate texture and structure Synthesis

Once all priorities on the fill front have been computed, the patch P with highest priority is computed. We then fill it with data extracted from the source region. In traditional inpainting techniques, pixel-value information is propagated via diffusion. As noted previously, diffusion necessarily leads to image smoothing, which results in blurry fill-in, especially of large regions. On the contrary, we propagate image texture by direct sampling of the source region. We search in the source region for that patch which is most similar to p.

C. Filling Order

These filling may be capable of propagating both texture and structure information. In this section demonstrates that the quality of the output image synthesis is highly influenced by the order in which the filling process proceeds. The ordered of the filled patches produces the horizontal boundary between the background image regions to be unexpectedly reconstructed as a curve.

A concentric-layer ordering, coupled with a patch-based filling may produce further artefacts. Another desired property of a good filling algorithm is that of avoiding "over-shooting" artefacts that occur when image edges are allowed to grow indefinitely.

D. Updating confidence values

After the patch Ψp has been filled with new pixel values, the confidence C(p) is updated in the area delimited by Ψˆp as follows:

$C(p) = C(\hat{p}) \; \forall p \in \Psi\hat{p} \cap \Omega$

This simple update rule allows us to measure the relative confidence of patches on the fill front, without image-specific parameters. As filling proceeds, confidence values decay, indicating that we are less sure of the colour values of pixels near the centre of the target region.

V. IMPLEMENTATION DETAILS

In this implementation the contour δΩ of the target region is modeled as a dense list of image point locations. These points are interactively selected by the user via a simple drawing interface. Given a point $p \in \delta\Omega$, the normal direction n_p is computed as. (i) The positions of the "control" points of δΩ are filtered via a bi-

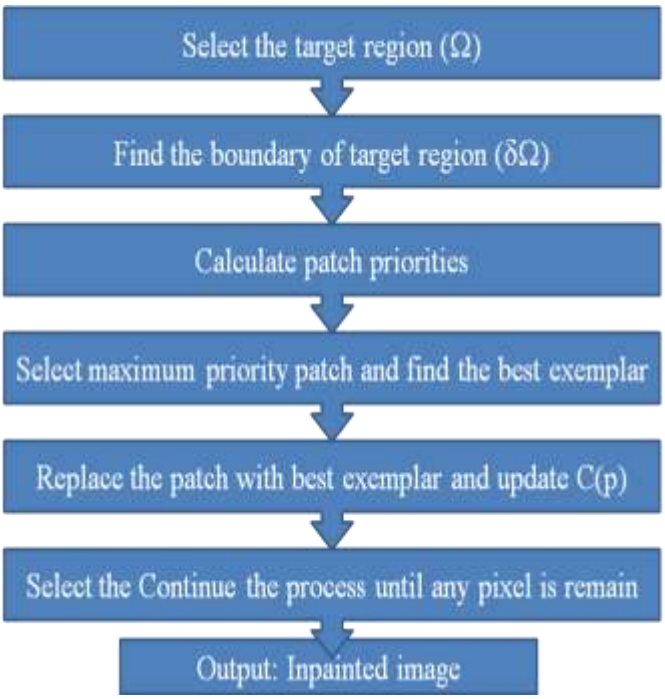
dimensional Gaussian kernel (ii) n_p is estimated as the unit vector orthogonal to the line through the preceding and the successive points in the list. Alternative implementation may make use of curve model fitting. The gradient ∇Ip is computed as the maximum value of the image gradient in $\Psi_p \cap I$. Robust filtering techniques may also be employed here.

Finally, pixels are classified as belonging to the target regionΩ, the source region ϕ or the remainder of the image by assigning different values to their alpha component. The image alpha channel is, therefore, updated (locally) at each iteration of the filling algorithm.

VI. Algorithm

1. The Find all boundary point of cracked portion
2. Find data pattern & confident for each boundary point
3. Multiply data pattern & confident of each Boundary point (It is total point/strength/priority of particular pixel)
4. Find the pixel on boundary point which one having highest priority
5. Create patch window by keeping that highest priority pixel as central one
6. Match this window with entire image from $(0,0)^{th}$ pixel to $(x,y)^{th}$ pixel and identify the matched window
7. Collect all the matched window position and again find data pattern & confident for each matched window, and multiply data pattern & confident
8. Find highest priority matched window then replace the patch window, which has been created in point number (5), with currently got matched window
9. Repeat the above steps unless there is no further inpainting needed

VII. Flowchart



VIII. Region Selection

The region selection is done through co-ordinate of x-axis and y-axis of the each point we place on image to select area in polygon shape. These are the basic mechanism behind the selection of region. First thing we have to save the first co-ordinates of the first rectangle we place on image and also use different color to represent it for reminder that where we have to complete the polygon. The steps of selection of region are as follows.

1. Place first rectangle on image to start the region selection.
2. Use the different color to represent this rectangle.
3. Now put necessary rectangle at each point of the region which you won't to select.
4. Use different color to represent this all rectangle.
5. Complete the polygon.
6. Fill the selected region with different color to view selected region.

7. Not the all co-ordinate of the place rectangle for process boundary region.
8. Then you can process on the selected region for inpainting.

IX. Conclusion And Future Scope

An exemplar based inpainting along with a priority term that defines the filling order in the image to produce the plausible output. In this algorithm, pixels maintain a confidence value and are chosen based on their priority that is calculated using confidence and data term. The confidence term defines the validity of that pixel whereas data term is focused towards maintaining the linear structures in the image.

In future this technique can be used to implement on high definition images and also for video inpainting using frame extraction algorithm. There will be chance to make a GUI for video inpainting for removing unwonted moving objects from the motion film.

REFERENCES

[1] S. Muthukumar, "Analysis of image inpainting techniques with exemplar, poisson, successive elimination and 8 pixel neighborhood methods," International Journal of Computer Applications IJCA, vol. 9, no. 11, pp. 24-28, 2010.
[2] S. CHHABRA, R. LALIT, and S. SAXENA, "An analytical study of different image inpainting techniques," Indian Journal of Computer Science and Engineering, vol. 3.
[3] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2, pp. II-721-II-728 vol. 2, IEEE.
[4] P. Elango and K. Murugesan, "Digital image inpainting using cellular neural network," Int. J. Open Problems Compt. Math, vol. 2, no. 3, pp. 439-450, 2009.
[5] P. Goyal and S. Diwakar, "Fast and enhanced algorithm for exemplar based image inpainting," in Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on, pp. 325-330, IEEE.
[6] F. Cao, Y. Gousseau, S. Masnou, and P. Perez, "Geometrically guided exemplar-based inpainting," International journal of computer vision, 2009.
[7] Q. Chen, Y. Zhang, and Y. Liu, "Image inpainting with improved exemplar-based approach," Multimedia Content Analysis and Mining, pp. 242-251, 2007.
[8] J. Wu and Q. Ruan, "Object removal by cross isophotes exemplar-based inpainting," in Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol. 3, pp. 810-813, IEEE.
[9] Y. Hong, "Object removal using exemplar-based inpainting," 2004.
[10] T. Kwok, H. Sheung, and C. Wang, "Fast query for exemplar-based image completion," Image Processing, IEEE Transactions on, vol. 19, no. 12, pp. 3106-3115, 2010.
[11] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar based image inpainting," Image Processing, IEEE Transactions on, vol. 13, no. 9, pp. 1200-1212, 2004.
[12] A. Criminisi, P. Perez, K. Toyama, M. Gangnet, and A. Blake, "Image region filling by exemplar-based inpainting," 2006.
[13] M. Richard and M. Chang, "Fast digital image inpainting," in Appeared in the Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain.
[14] W. Cheng, C. Hsieh, S. Lin, C. Wang, and J. Wu, "Robust algorithm for exemplar-based image inpainting," in Proceedings of the International Conference on Computer Graphics, Imaging and Visualization, 2007.
[15] A. Telea, "An image inpainting technique based on the fast marching method," Journal of graphics tools, vol. 9, no. 1, pp. 23-34, 2004.
[16] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, "A variational framework for exemplar based image inpainting," International journal of computer vision, vol. 93, no. 3, pp. 319- 347, 2011.