International Journal of Electrical, Electronics and Computer Engineering **4**(1): 1-4(2015)

ISSN No. (Online): 2277-2626

Exemplar Based Image Inpainting to remove object

Shital D. Suryawanshi*, J.B. Jadhav** and P.V. Baviskar***

**ME (Comm. Engg.) Scholar, Department of Electronics, SSVPS COE, Dhule, (MS), India*
***Assistant Professor, Department of Electronics, SSVPS COE, Dhule, (MS), India*
****Assistant Professor, Department of Electronics, SSVPS COE, Dhule, (MS), India*

(Corresponding author: Shital D. Suryawanshi)
(Received 04 November, 2014 Accepted 05 December, 2014)

ABSTRACT: The process of removing the specific object or area or repairing the damaged area in a image is known as image inpainting. This algorithm is proposed for removing large objects from digital image. The challenge is to fill in the hole that is left behind in a visually plausible way. We first note that exemplar-based texture synthesis contains the essential process required to replicate both texture and structure; the success of structure propagation, however, is highly dependent on the order in which the filling proceeds. We propose a best algorithm in which the confidence in the synthesized pixel values is propagated in a manner similar to the propagation of information in inpainting. The actual color values are computed using exemplar based synthesis. In this paper, the simultaneous propagation of texture and structure information is achieved by a single, efficient algorithm. Computational efficiency is achieved by a block-based sampling process.

Keywords: Inpainting algorithm, Exemplar based inpainting, image inpainting, block-based sampling process

I. INTRODUCTION

In painting algorithm are classified into three category:
(i) Inpainting algorithm based on PDE,
(ii) Exemplar based inpainting.

This paper presents a exemplar based algorithm for removing large objects from digital photographs and replacing them with visually plausible backgrounds. The algorithm effectively hallucinates new color values for the target region in a way that looks “reasonable” to the human eye. This paper builds upon and extends the work with a more detailed description of the algorithm and extensive comparisons with the state of the art. In previous work, several researchers have considered texture synthesis as a way to fill large image regions with “pure” textures repetitive two-dimensional (2D) textural patterns with moderate stochasticity. As effective as these techniques are in replicating consistent texture, they have difficulty filling holes in photographs of real world scenes, which often consist of linear structures and composite textures multiple textures interacting spatially. The main problem is that boundaries between image regions are a complex product of mutual influences between different textures. In contrast to the 2D nature of pure textures, these boundaries form what might be considered more one dimensional (1D), or linear, image structures.

A number of algorithms specifically address the image filling issue for the task of image restoration, where speckles, scratches, and overlaid text are removed. These image in painting techniques fill holes in images by propagating linear structures (called isophotes in the inpainting literature) into the target region via diffusion. They are inspired by the partial differential equations of physical heat flow and work convincingly as restoration algorithms. Their drawback is that the diffusion process introduces some blur, which becomes noticeable when filling larger regions [2]. The technique presented here combines the strengths of both approaches into a single, efficient algorithm. As with inpainting [1], we pay special attention to linear structures. However, linear Structures abutting the target region only influence the fill order of what is at core an exemplar-based texture synthesis algorithm.

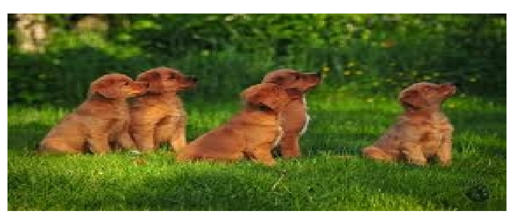


Fig. 1. (a) Original image.

Fig. 1. (b) The object to be removed is selected by marking.

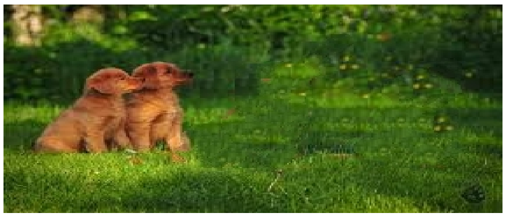


Fig. 1. (c) The Object removed.

The result is an algorithm that has the efficiency and qualitative performance of exemplar based texture synthesis, but which also respects the image constraints imposed by surrounding linear structures.

The algorithm [3] we propose in this paper builds on very recent research along similar lines. The work decomposes the original image into two components, one of which is processed by inpainting and the other by texture synthesis. The output image is the sum of the two processed components. This approach still remains limited to the removal of small image gaps; however, as the diffusion process continues to blur the filled region. In this paper, we present a simpler and faster region filling algorithm which does not suffer from blur artifacts.

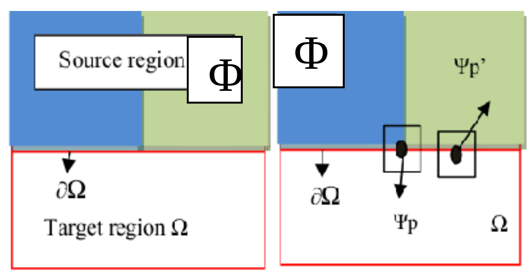
One of the first attempts to use exemplar-based synthesis specifically for object removal was by Harrison. There, the order in which a pixel in the target region is filled was dictated by the level of “texturedness” of the pixel’s neighborhood. Although the intuition is sound, strong linear structures were often overruled by nearby noise, minimizing the value of the extra computation. A related technique drove the fill order by the local shape of the target region, but did not seek to explicitly propagate linear structures.

II. LITERATURE SURVEY

The PDE based Inpainting algorithms have drawback that it introduce some blur in the image after filling algorithm. It cannot fill the large missing region and it cannot renovate the texture pattern. The analysis proved that the exemplar based image Inpainting will create better results for Inpainting the huge missing region also that these algorithms can inpaint both the formation and textured image efficiently

A. Exemplar Based Synthesis

The core of our algorithm is an isophote driven image sampling process. It is well understood that exemplar based approaches perform well for 2D textures. However, we note, in addition, that exemplar based texture synthesis is sufficient for propagating extended linear image structures[5], as well; i.e. a separate synthesis mechanism is not required for handling isophotes. The region to be filled, i.e., the target region is indicated by Ω, and its contour is denoted by $\partial\Omega$. The contour evolves inward as the algorithm progresses, and so we also refer to it as the “fill front”. The source region Φ which remains fixed throughout the algorithm, provides samples used in the filling process. We now focus on a single iteration of the algorithm to show how structure and texture are adequately handled by exemplar based synthesis. Suppose that the square template centered at the point [Fig. 2(b)] is to be filled. The best match sample from the source region comes from the patch, which is most similar to those parts that are already filled in. In the example in Fig. 2(b), we see that if lies on the continuation of an image edge, the most likely best matches will lie along the same (or a similarly colored) edge (e.g., and in Fig. 3(a)).

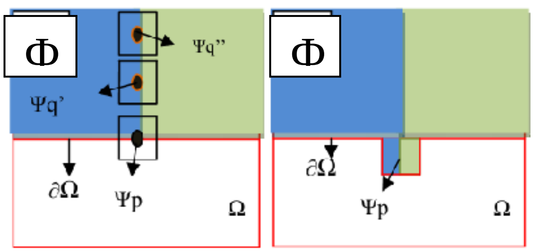


(a) (b)

Fig. 2. (a). Shows that Ω is the target region, Φ is the source region and $\partial\Omega$ is the fill front of patch propagation (b)shows the two examples of known image parts surrounding patch Ψp and $\Psi p'$ which locate at edge and flat texture region respectively.

Fig.3 (a) shows that for the selected patch Ψp , sparse linear combination of candidate patches $\{\Psi p', \Psi p'' ... \Psi p_N\}$ is used to infer the missing pixels in patch Ψp. (b) Shows the best matching patch in the candidates set has been copied into the position occupied by Ψp , thus achieving partial filling of Ω.

All that is required to propagate the isophote inwards is a simple transfer of the pattern from the best-match source patch [Fig. 3(b)]. Notice that isophote orientation is automatically preserved. In the figure, despite the fact that the original edge is not orthogonal to the target contour, the propagated structure has maintained the same orientation as in the source region. In this work, we focus on a patch-based filling approach because, as noted, this improves execution speed. Furthermore, we note that patch based filling improves the accuracy of the propagated structures.



(a) (b)

Fig. 3. Patch inpainting.

III. REGION FILLING ALGORITHM

First, given an input image, the user selects 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 by the user. Next, as with all exemplar based texture synthesis, the size of the template window must be specified. We provide a default window size of 9× 9 pixels but, in practice, require the user to set it to be slightly larger than the largest distinguishable texture element, or "texel," in the source region [5]. Once these parameters are determined, the region filling proceeds automatically. In our algorithm, each pixel maintains a color value (or "empty," if the pixel is unfilled) and a confidence value, which reflects our confidence in the pixel value, and which is frozen once a pixel has been filled. During the course of the algorithm, patches along the fill front are also given a temporary priority value, which determines the order in which they are filled. Then, our algorithm iterates the following three steps until all pixels have been filled.

A. Computing Patch Priorities

Our algorithm performs the synthesis task through 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 [7] is based toward those patches which are on the continuation of strong edges an are surrounded by high-confidence pixels. Given a patch Ψp centered at the point p for some p ε ∂Ω, we define its priority P(p)as the product of two terms

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

We call C(p) the confidence term and D(p) the data term , and they are defined as follows:

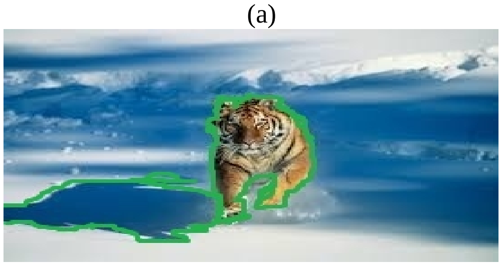
$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\Psi_{\mathbf{p}}\cap(\mathcal{I}-\Omega)} C(\mathbf{q})}{|\Psi_{\mathbf{p}}|} \qquad D(\mathbf{p}) = \frac{|\nabla I_{\mathbf{p}}^{\perp}\cdot \mathbf{n}_{\mathbf{p}}|}{\alpha} \qquad \ldots(2)$$

Where |Ψp| is the area of Ψp, is α a normalization factor (e.g., α=255 for a typical grey level image), is np a unit vector orthogonal to the front ∂Ω in the point P. The priority P(p) is computed for every border patch, with distinct patches for each pixel on the boundary of the target region. The confidence term may be thought of as a measure of the amount of reliable information surrounding the pixel. The intention is to fill first those patches which have more of their pixels already filled, with additional preference given to pixels that were filled early on (or that were never part of the target region). For example, patches that include corners [10] and thin tendrils of the target region will tend to be filled first, as they are surrounded by more pixels from the original image. These patches provide more reliable information against which to match. the term approximately enforces the desirable concentric fill order[4].As filling proceeds, pixels in the outer layers of the target region will tend to be characterized by greater confidence values and, therefore, be filled earlier; pixels in the center of the target region will have lesser confidence values. The data term D(p)is a function of the strength of isophotes hitting the front ∂Ω at each iteration. This term boosts the priority of a patch that an isophote "flows"into.This factor is of fundamental importance in our algorithm [9]because it encourages linear structures to be synthesized first and, therefore, propagated securely into the target region[8]. Broken lines tend to connect, thus realizing the "connectivity principle" of vision psychology.

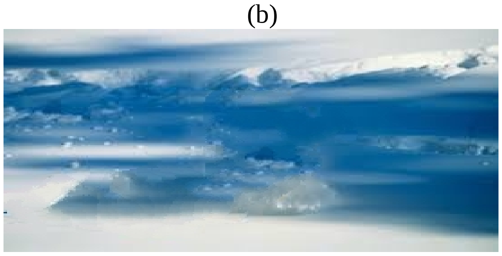
B. Propagating Texture and Structure Information

Once all priorities on the fill front have been computed, the patch Ψp with highest priority is found. We then fill it with data extracted from the source region Φ.

(a)



(b)



(c)

Fig.4 (a): Original image (b): For inpainting selected image and (c): removed region is filled by background portion of image.

C. Updating Confidence Values

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

$$C(\mathbf{p}) = C(\hat{\mathbf{p}}) \quad \forall\, \mathbf{p} \in \Psi_{\hat{\mathbf{p}}} \bigcap \Omega. \qquad ...(3)$$

This simple update rule [11] 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 color values of pixels near the center of the target region [12].

IV. CONCLUSION

The PDE based Inpainting algorithms cannot fill the large missing region and it cannot renovate the texture pattern. The analysis proved that the exemplar based image Inpainting will create better results for Inpainting the huge missing region also that these algorithms can inpaint both the formation and textured image efficiently This paper has presented a novel algorithm for removing large objects from digital photographs. The result is an image in which the selected object has been replaced by a visually plausible background that mimics the appearance of the source region. Our approach employs an exemplar based texture synthesis technique modulated by a unified scheme for determining the fill order of the target region. Pixels maintain a confidence value, which together with image isophotes, influence their fill priority.

REFERENCES

[1]. Jino Lee, Dong-Kyu Lee, and Rae-Hong Park, “Robust Exemplar-Based Inpainting Algorithm Using Region Segmentation” in *IEEE Transactions on Consumer Electronics*, Vol. **58**, No. 2, May 2012.

[2]. Zongben Xu and Jian Sun “Image Inpainting by Patch Propagation” Using Patch Sparsity” *in IEEE Transactions On Image Processing*, vol. **19**, no. 5, MAY 2010.

[3]. Jianchao Yang, John Wright, Thomas S. Huang, “Image Super-Resolution Via Sparse Representation” *IEEE Transactions on Image Processing*, vol. **19**, no. 11, november 2010.

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

[5] .R. Bornard, E. Lecan, L. Laborelli, andJ.H.Chenot, “Missing data correction in still images and image sequences,” *in ACM Multimedia, France,* Dec. 2002.

[6]. C. Ballester, V. Caselles, J. Verdera, M.Bertalmio, and G. Sapiro, “A variational model for filling-in gray level and color images,” *in Proc. Int. Conf. Computer Vision, Vancouver,* BC, Canada, June 2001, pp. 10–16.

[7]. Antonio Criminisi, Patrick Perez,and Kentaro Toyama “Region Filling and Object Removal by Exemplar Based Image Inpainting”*in IEEE Transctions on Image processing* , vol. **13**, no. 9,september 2004.

[8]. T. Chan and J. Shen, “Local inpainting models and tv inpainting,” *SIAMJ. Appl. Math.*, vol. **62**, no. 3, pp. 1019–1043, 2001.

[9]. T. Chan and J. Shen, “Non-texture inpainting by curvature-driven diffusions,”*J. Vis. Commun. Image Represent.*, vol. **4**, no. 12, pp. 436–449, 2001.

[10]. A. Efros and T. Leung, “Texture synthesis by non-parametric sampling,” in *Proc. Int. Conf. Comp. Vision*, 1999, pp. 1033–1038.

[11]. M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” *IEEE Trans. Image Process.*, vol. **12**, pp. 882–889, 2003.

[12]. J. Wu and Q. Ruan, “Object removal by cross isophotes examplar-based image inpainting,” in *Proc. Int. Conf. Pattern Recognition*, 2006, pp.

[13]. A. D. Boloorani, S. Erasmi, and M. Kappas, “Multi-Source Remotely Sensed Data Combination: Projection Transformation Gap-Fill Procedure,” *Sensor*, pp. 84429-4440, July2008.