

# Region Filling and Object Removal by Exemplar-Based Image Inpainting

Mrs. Waykule J.M., Ms. Patil V.A.

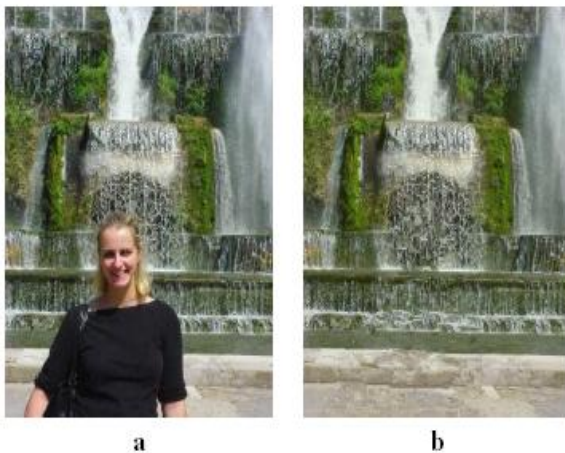
**Abstract**— A new algorithm is proposed for removing large objects from digital images. The challenge is to fill in the hole that is left behind in a visually plausible way. In the past, this problem has been addressed by two classes of algorithms: (i) "texture synthesis" algorithms for generating large image regions from sample textures, and (ii) "inpainting" techniques for filling in small image gaps. The former has been demonstrated for "textures" – repeating two-dimensional patterns with some stochastic; the latter focus on linear "structures" which can be thought of as one-dimensional patterns, such as lines and object contours. This paper presents a novel and efficient algorithm that combines the advantages of these two approaches.

**Index Terms**— Image Inpainting, Texture Synthesis, Simultaneous Texture and Structure Propagation.

## 1 INTRODUCTION

A New algorithm is proposed for removing large objects from digital images. The challenge is to fill in the hole that is left behind in a visually plausible way.

Figure 1 show an example of this task, where the foreground person (manually selected as the *target region*) is automatically



replaced by data sampled from the remainder of the image Fig. 1 Removing large objects from images. (a) Original photograph. (b) The region corresponding to the foreground person (covering about 19% of the image) has been manually selected and then automatically removed. Notice that the horizontal structures of the fountain have been synthesized in the occluded area together with the water, grass and rock textures.

- Mrs. Waykule J.M. is pursuing masters degree program in electronics engineering in Shivaji University, Maharashtra, PH-9422525671. E-mail: jyoti.tate@gmail.com
- Ms. Patil V.A. is currently pursuing masters degree program in electronics engineering in Shivaji University, Maharashtra, PH-9975035677. E-mail: viddulata@gmail.com  
(This information is optional; change it according to your need.)

## 2. PRESENT THEORY AND PRACTICES

In the past, this problem 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. The former work well for "textures" – repeating two-dimensional patterns with some stochastic; the latter focus on linear "structures" which can be thought of as one-dimensional patterns, such as lines and object contours.



Fig.2 Removing large objects from photographs. a) Original image b) The result of region filling by traditional image inpainting. Notice the blur introduced by the diffusion process and the complete lack of texture in the synthesized area c) The final image where the bungee jumper has been completely removed and the occluded region reconstructed by our automatic algorithm

### 3. KEY OBSERVATIONS

#### 3.1 Exemplar-based synthesis suffices

The core of our algorithm is an isophote-driven image sampling process. It is well-understood that exemplar-based approaches perform well for two-dimensional textures [1], [11],[17]. But, we note in addition that exemplar-based texture synthesis is sufficient for propagating extended linear image structures, as well; i.e., a separate synthesis mechanism is not required for handling isophotes.

The core of our algorithm is an isophote-driven image sampling process. It is well-understood that exemplar-based approaches perform well for two-dimensional textures [1], [11],[17]. But, we note in addition that exemplar-based texture synthesis is sufficient for propagating extended linear image structures, as well; i.e., a separate synthesis mechanism is not required for handling isophotes.

Figure 3 illustrates this point. For ease of comparison, we adopt notation similar to that used in the inpainting literature. The region to be filled, i.e., the target region is indicated by  $\Omega$ , and its contour is denoted  $\delta\Omega$ . The contour evolves inward as the algorithm progresses, and so we also refer to it as the "fill front". The source region,  $\Phi$  which remains fixed throughout the algorithm, provides samples used in the filling process.

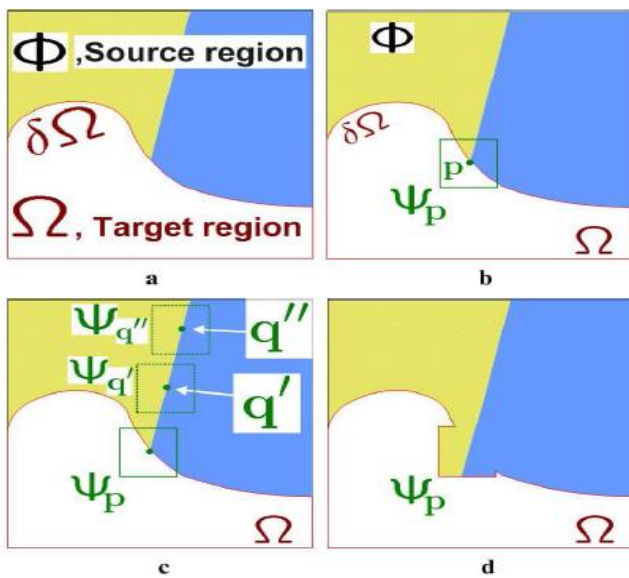


Fig.3 Structure propagation by exemplar-based texture synthesis. (a) Original image, with the target region  $\Omega$ , its contour  $\delta\Omega$ , and the source region  $\Phi$  clearly marked. (b) We want to synthesize the area delimited by the patch  $\varphi_p$  centered on the point  $p \in \delta\Omega$  (c) The most likely candidate matches for  $\varphi_p$  lie along the boundary between the two textures in the source region e.g.  $\varphi_{q'}$  and  $\varphi_{q''}$  (d) The best matching patch in the candidates set has been copied into the position occupied by  $\varphi_p$ , thus achieving partial filling of  $\Omega$ . Notice that both texture and structure (the separating line) have been propagated inside the target region. The target region  $\Omega$  has, now, shrunk and its front  $\delta\Omega$ .has assumed a different shape.

The user will be asked to select a target region,  $\Omega$  manually. (a) The contour of the target region is denoted as  $\delta\Omega$  (b) For

every point  $p$  on the contour  $\delta\Omega$ , a patch  $\Psi_p$  is constructed, with  $p$  in the centre of the patch. A priority is calculated based on how much reliable information around the pixel, as well as the isophote at this point. (c) The patch with the highest priority would be the target to fill. A global search is performed on the whole image to find a patch,  $\Psi_q$  that has most similarities with  $\Psi_p$ . (d) The last step would be copy the pixels from  $\Psi_q$  to fill  $\Psi_p$ . With a new contour, the next round of finding the patch with the highest continues, until all the gaps are filled.

#### 3.2 Filling order is critical.

The previous section has shown how careful exemplar-based filling may be capable of propagating both texture and structure information. This section demonstrates that the quality of the output image synthesis is highly influenced by the order in which the filling process proceeds. Furthermore, we list a number of desired properties of the "ideal" filling algorithm.

A comparison between the standard concentric layer filling (onion-peel) and the desired filling behavior is illustrated in fig. 3. Figures 3b,c,d show the progressive filling of a concave target region via an anti-clockwise onion-peel strategy. As it can be observed, this ordering of the filled patches produces the horizontal boundary between the background image regions to be unexpectedly reconstructed as a curve.

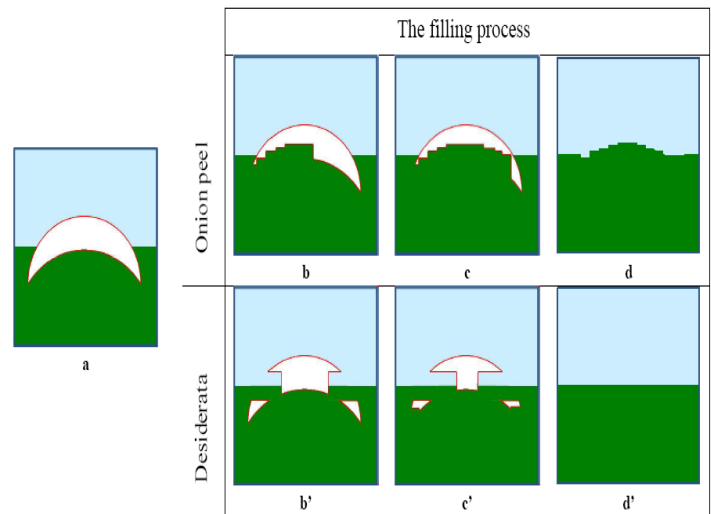


Fig. 4 The importance of the filling order when dealing with concave target regions. (a) A diagram showing an image and a selected target region (in white). The remainder of the image is the source. (b,c,d) Different stages in the concentric-layer filling of the target region. (d) The onion-peel approach produces artifacts in the synthesized horizontal structure. (b', c', d') Filling the target region by an edge-driven filling order achieves the desired artifact-free reconstruction. (d') The final edge-driven reconstruction, where the boundary between the two background image regions has been reconstructed correctly.

A better filling algorithm would be one that gives higher priority of synthesis to those regions of the target area which

lie on the continuation of image structures, as shown in figs. 3b', c', d'. Together with the property of correct propagation of linear structures, the latter algorithm would also be more robust towards variations in the shape of the target regions. A concentric-layer ordering, coupled with a patch-based filling may produce further artifacts (cf. fig. 4) Therefore, filling order is crucial to non-parametric texture synthesis [1], [6], [12], [15]. To our knowledge, however, designing a fill order which explicitly encourages propagation of linear structure (together with texture) has never been explored, and thus far, the default favorite has been the "onion peel" strategy. Another desired property of a good filling algorithm is that of avoiding "over-shooting" artifacts that occur when image edges are allowed to grow indefinitely. The goal here is finding a good balance between the propagation of structured regions and that of textured regions (fig. 3b', c', d'), without employing two adhoc strategies. As demonstrated in the next section, the algorithm we propose achieves such a balance by combining the structure "push" with a confidence term that tends to reduce sharp in-shooting appendices in the contour of the target region.

As it will be demonstrated, the filling algorithm proposed in this paper overcomes the issues that characterize the traditional concentric-layers filling approach and achieves the desired properties of: (i) correct propagation of linear structures, (ii) robustness to changes in shape of the target region, (iii) balanced simultaneous structure and texture propagation, all in a single, efficient algorithm. We now proceed with the details of our algorithm.

#### 4. OUR REGION-FILLING ALGORITHM

First, given an input image, the user selects a target region,  $\Omega$ , to be removed and filled. The source region, may be defined as the entire image minus the target region ( $\Phi = I - \Omega$ ), as a dilated band around the target region, or it may be manually specified by the user.

Algorithm iterates the following three steps until all pixels have been filled.

- 1) Computing Patch Priorities.
- 2) Propagating Texture and Structure Information.
- 3) Updating Confidence Values.

##### 1) Computing Patch Priorities:

The priority computation is biased toward those patches which:

- a) Are on the continuation of strong edges.
- b) Are surrounded by high-confidence pixels. Given a patch  $\varphi_p$  centered at the point 'p'. We defines its priority  $P(P)$  as the product of two terms:

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

$C(P)$  The confidence term that measure of the amount of reliable information surrounding the pixel 'p'

$$C(P) = \frac{\sum_{(q \in \varphi_p) \cap (I - \Omega)} C(P)}{|\Psi_p|}$$

$D(P)$  The data term that is a function of the strength of isophotes hitting the front  $\partial\Omega$  at each iteration.

$$D(p) = \frac{|\nabla I_p^\perp \cdot \eta_p|}{\alpha}$$

Where, (a)  $\eta_p$  estimated as the unit vector orthogonal to  $\varphi_p$  the front  $\partial\Omega$ ... (b)  $\nabla I_p^\perp$  Is the isophote (direction and intensity) at point 'p' It computed as the maximum value of the image gradient in  $\varphi_p \cap I$  (c)  $\alpha$ =Normalization factor (e.g. For Gray level it is  $\alpha=255$ )

##### 2) Propagating Texture and Structure information

Propagate image texture by direct sampling of the source region, the distance  $\delta(\varphi_p, \varphi_q)$  between two generic patches  $\varphi_p$  and  $\varphi_q$  is simply defined as the sum of squared differences.

$$\Psi_q = \arg \min_{\Psi_q \in \Phi} d(\Psi_p, \Psi_q)$$

##### 3) Updating Confidence Values

After the patch  $\varphi_p$  has been filled with new pixel values, the confidence  $C(P)$  is updated in the area delimited by  $\varphi_p$

$$C(P) = C(\bar{P}) \quad \forall_p \in (\varphi_p \cap T)$$

#### 5. SOME PROPERTIES OF EXEMPLAR-BASED IMAGE INPAINTING ALGORITHM

The effect of the confidence term is that of smoothing the contour of the target region by removing sharp appendices and making the target contour close to circular. . Notice that (1) only dictates the order in which filling happens. The use of image patches for the actual filling achieves texture synthesis [11]. Furthermore, since the fill order of the target region is dictated solely by the priority function  $P(p)$ , we avoid having to predefine an arbitrary fill order as done in existing patch-based approaches [11], Our fill order is function of image properties, resulting in an organic synthesis process that eliminates the risk of "broken-structure" artifacts (as in fig. 11f). Furthermore, since the gradient-based guidance tends to propagate strong edges, blocky and misalignment artifacts are reduced (though not completely eliminated), without a patch-cutting (quilting) step [11] or a blur-inducing blending step. It must be stressed that our algorithm does not use explicit nor implicit segmentation at any stage. For instance, the gradient operator in (1) is never thresholded and real valued numbers are employed.

#### 6. IMPLEMENTATION DETAILS

In our implementation the contour  $\partial\Omega$  of the target region is modelled as a dense list of image point locations. These points are interactively selected by the user via a simple drawing interface. Given a point  $P \in \partial\Omega$ , the normal direction  $\eta_p$  is computed as follows: i) the positions of the "control" points of  $\partial\Omega$  are filtered via a bi-dimensional Gaussian kernel and, ii)  $\eta_p$  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  $\nabla I_P$  is computed as the maximum value of the image gradient in  $\phi_P \cap I$ . Robust filtering techniques may also be employed here. Finally, pixels are classified as belonging to the target region  $\Omega$ , the source region  $\Phi$  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.

### 7. RESULT AND COMPARISONS

Here we apply our algorithm to a variety of images, ranging from purely synthetic images to full-color photographs that include complex textures. Where possible, we make side-by-side comparisons to previously proposed methods. In other cases, we hope the reader will refer to the original source of our test images (many are taken from previous literature on inpainting and texture synthesis) and compare these results with the results of earlier work.

In all of the experiments, the patch size was set to be greater than the largest texel or the thickest structure (e.g., edges) in the source region. Furthermore, unless otherwise stated the source region. All experiments were run on a 2.5GHz Pentium IV with 1GB of RAM.

Figure 5 shows yet another comparison between the concentric filling strategy and the proposed algorithm. In the presence of concave target regions, the "onion peel" filling may lead to visible artifacts such as unrealistically broken structures (see the pole in fig. 11f). Conversely, the presence of the data term of (1) encourages the edges of the pole to grow "first" inside the target region and thus correctly reconstruct the complete pole (fig. 11f). This example demonstrates the robustness of the proposed algorithm with respect to the shape of the selected target region tend to have large priority values associated with them.

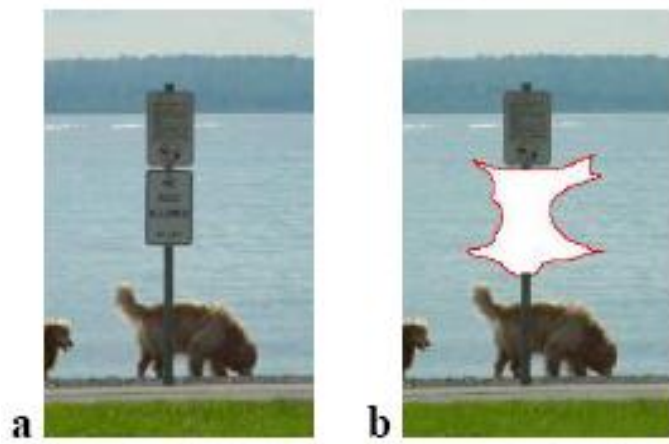
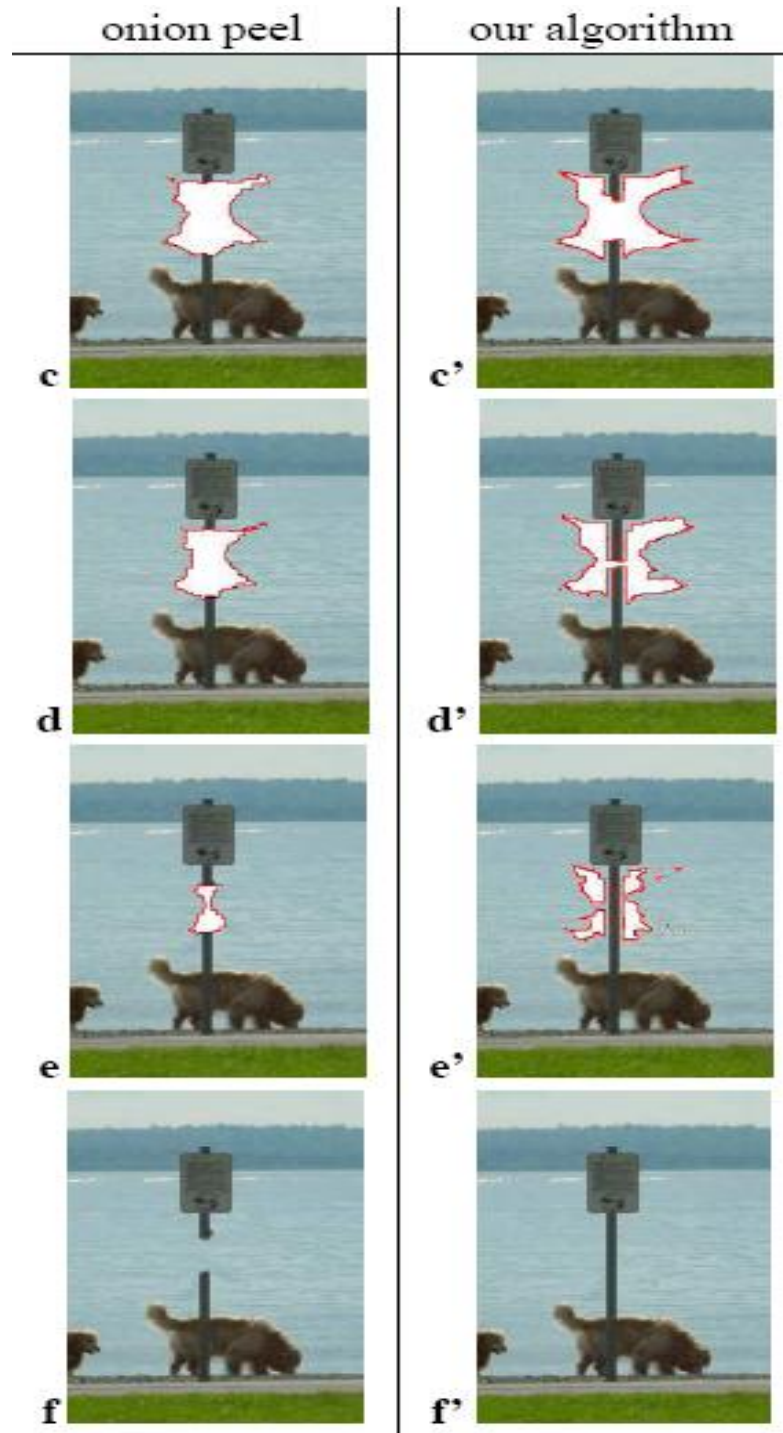


Fig. 5 Onion peels vs. structure-guided filling. (a) Original image. (b) The target region has been selected and marked with a red boundary. (c, d, e, f) Results of filling by concentric layers. (c', d', e', f') Results of filling with our algorithm. Thanks to the *data term* in (1) the sign pole is reconstructed correctly by our algorithm

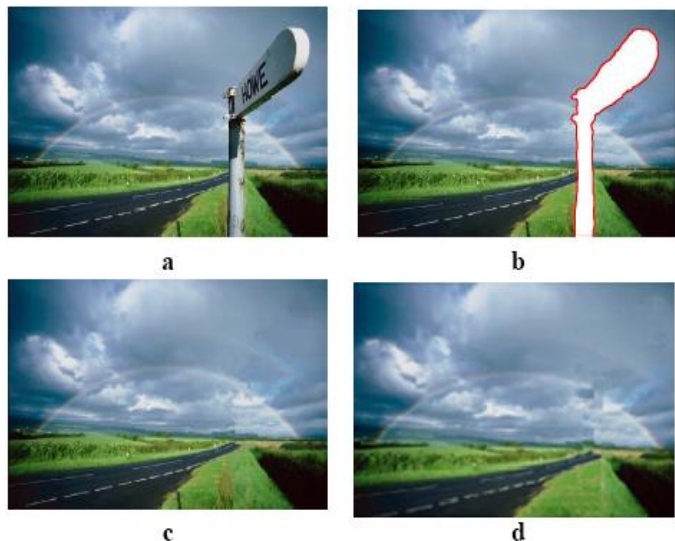


Fig. 6 Comparisons with Jia et al. "Image Repairing" [16]. (a) Original input image. (b) The manually selected target region. (c) The resulting region-filling achieved by Jia et al. (d) the result of our region-filling algorithm. The missing portion of rainbow is reconstructed convincingly. Figures (c) and (d) are of comparable quality, but our algorithm avoids the image segmentation step with considerable increase in speed.

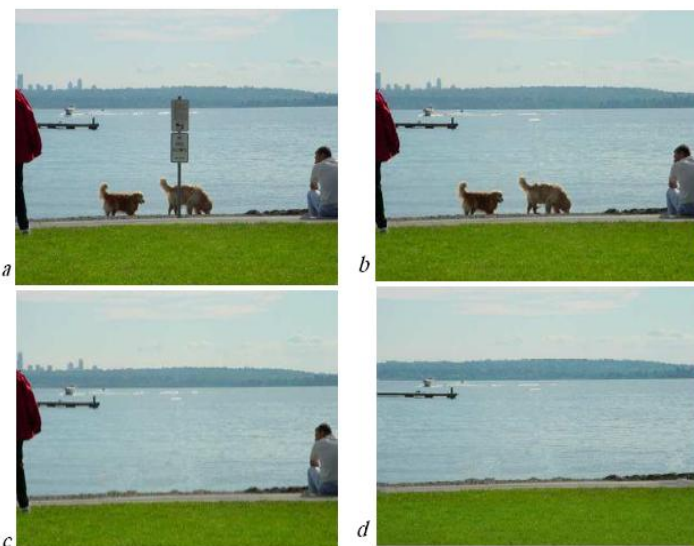


Fig. 7 Removing the multiple objects from photographs. (a) Original photograph of Kirkland (WA). (b,c,d) Several objects are sequentially removed. Notice how well the shore line and other structures have been reconstructed.

## 8. CONCLUSION AND FUTURE WORK

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.

The technique is capable of propagating both linear structure and two-dimensional texture into the target region with a single, simple algorithm. Comparative experiments show that a simple selection of the fill order is necessary and sufficient to handle this task.

Our method performs at least as well as previous techniques designed for the restoration of small scratches, and, in instances in which larger objects are removed, it dramatically outperforms earlier work in terms of both perceptual quality and computational efficiency.

Moreover, robustness towards changes in shape and topology of the target region has been demonstrated, together with other advantageous properties such as:

### Advantages

- a) Preservation of edge sharpness
- b) No dependency on image segmentation
- c) Balanced region filling to avoid over-shooting artifacts
- d) Patch-based filling helps achieve: (i) speed efficiency, (ii) Accuracy in the synthesis of texture (less garbage growing), and finally (iii) accurate propagation of linear structures.

### Disadvantages

- a) The synthesis of regions for which similar patches do not exist does not produce reasonable results (a problem common to [10], [19]);
- b) The algorithm is not designed to handle curved structures
- c) Our algorithm does not handle depth ambiguities (An expression whose meaning cannot be determined from its context) (*i.e.*, what is in front of what in the occluded area?).

Currently, we are investigating the possibility of constructing Ground-truth data and designing evaluation tests that would allow us to quantify the performance of each algorithm. This turns out to be a non-trivial task. Furthermore we are investigating extensions of the current algorithm to handle accurate propagation of curved structures in still photographs as well as removing objects from video, which promise to impose an entirely new set of challenges.

## REFERENCES

- [1] A.Criminisi\*, P.Perez and K.Toyama. "Region Filling and Object Removal by Exemplar-Based Image inpainting" IEEE TRANSACTION ON IMAGE PROCESSING, VOL. 13, NO. 9, SEP 2004
- [2] P.Harrison. "A non-hierarchical procedure for re-synthesis of complex texture." In Proc. Int. Conf. Central Europe Comp. Graphics, Visua. and Comp. Vision, Plzen, Czech Republic, February 2001
- [3] M.Bertalmio, L. Vese, G. Sapiro, and S. Osher. "Simultaneous structure and texture image inpainting." In Proc. Conf. Comp. Vision Pattern Rec., Madison, WI, 2003

- [4] I. Drori, D. Cohen-Or, and H. Yeshurun. "Fragment-based image completion." In *ACM Trans. on Graphics (SIGGRAPH 2003 issue)*, 22(3), volume 22, pages 303–312, San Diego, US, 2003
- [5] A.Efros and T. Leung. "Texture synthesis by non-parametric sampling." In *Proc. Int. Conf. Computer Vision*, pages 1033–1038, Kerkyra, Greece, September 1999
- [6] A.Efros and W.T. Freeman. "Image quilting for texture synthesis and transfer." In *Proc. ACM Conf. Comp. Graphics (SIGGRAPH)*, pages 341–346, Eugene Fiume, August 2001
- [7] A.Efros and W.T. Freeman. "Image quilting for texture synthesis and transfer." In *Proc. ACM Conf. Comp. Graphics (SIGGRAPH)*, pages 341–346, Eugene Fiume, August 2001
- [8] 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 June 2001*
- [9] J.S.De Bonet. "Multiresolution sampling procedure for analysis and synthesis of texture images." In *Proceedings of SIGGRAPH 1997*
- [10] D.Garber. *Computational Models for Texture Analysis and Texture Synthesis*. PhD thesis, University of Southern California, 1981
- [11] M.M.Oliveira, B. Bowen, R. McKenna, and Y. S. Chung. "Fast Digital Inpainting", Sep 2001
- [12] T.F.chan and J.Shen. "Mathematical model for local non-texture inpainting," 2001
- [13] R.J.Cant and C. S. Langensiepen, "A Multiscale Method for Automated Inpainting", *ESM2003*, 2003
- [14] J.Portilla and E.P.Simoncelli, "A parametric texture coefficients," *International Journal of Computer vision*, vol. 40, no. 1, pp. 49–70, 2000
- [15] D.J.Heeger and J.R.Bergen, "Pyramid-based texture analysis/synthesis," in *SIGGRAPH*, 1995, pp. 229–238.
- [16] J.Jia and C.-K. Tang, "Image repairing: Robust image synthesis by adaptive and tensor voting," *cvpr*, vol. 01, p. 643, 2003
- [17] **Websites:**
  - <http://www.mathworks.com/>
  - <http://www.ieeexplore.ieee.org/>
- [18] **Reference Books:**
  - a) Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing".
  - b) A. K. Jain "Fundamentals of Digital Image processing"