Structure and Texture Image Inpainting

Somayeh Hesabi, Mansour Jamzad, Nezam Mahdavi-Amiri

Computer Science Department, Sharif University of Technology, Tehran, Iran
Computer Engineering Department, Sharif University of Technology, Tehran, Iran
Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, Iran

shesabi@mehr.sharif.edu
jamzad@sharif.edu
nezamm@sina.sharif.edu

***Abstract*— Inpainting refers to the task of filling in the missing or damaged regions of an image in an undetectable manner. We have an image to be reconstructed in a user-defined region. We use a fast decomposition method to obtain two components of the image, namely structure and texture. Reconstruction of each component is performed separately. The missing information in the structure component is reconstructed using a structure inpainting algorithm, while the texture component is repaired by a texture synthesis technique. To obtain the inpainted image, the two reconstructed components are composed together. Taking advantage of both the structure inpainting methods and texture synthesis techniques, we designed an effective image reconstruction method. Comparative reconstructed test images show the merits of our proposed approach in providing high quality inpainted images.**

***Keywords*— Image inpainting, image decomposition, structure, texture.**

I. INTRODUCTION

Image inpainting provides means to restore damaged regions of an image so that the image appears normal to the not familiar observers. Its applications include removing scratches in old photos, removing text or logos, repairing damaged areas in unreliably transmitted images, image zooming, removing undesired objects from an image, or even creating artistic effects. Since the missing or damaged areas cannot be simply classified objectively, the user needs to identify them. These specified regions is called inpainting domain.

Some aspects of image inpainting were first introduced by Bertalmio et al. [1]. They were coined inpainting as a reminder of recovery process which museum experts did for old and deteriorating artwork. They first gathered information over the boundary of the inpainting domain. Then, image smoothness, estimated by Laplacian operator, propagated along isophote (i.e. a line which points have the same gray value) directions with anisotropic diffusion.

Several classical inpainting methods have been proposed [2, 3]. These models attempt to restore inpainting domain by diffusion schemes based on partial differential equations (PDEs).

Inspired by the work of Bertalmio et al., Chan and Shen proposed two PDE-based algorithms [4, 5]. The Total Variational (TV) model [4] uses an Euler-Lagrange equation coupled with anisotropic diffusion to preserve the direction of isophotes. The major drawback of TV is that it does not restore a single object well when its disconnected remaining elements are separated far apart within the inpainting domain. The Curvature Driven Diffusion (CDD) model [5] extended the TV algorithm to consider geometric information by defining strength of isophotes. So, it can inpaint larger damaged regions.

Roth and Black [6] presented a simple inpainting algorithm based on *prior* models. The authors modified the diffusion technique used for denoising approaches and applied it to the inpainting domain to repair the regions.

The above approaches perform well on piecewise smooth images (images with only geometric information) but fail to reconstruct areas containing texture with fine details. Hence, methods based on local statistical analysis, such as texture synthesis techniques, were proposed to suitably fill in large regions with pure textures.

The common idea of texture synthesis approaches is to create a new texture from an initial seed. Therefore, the appearance of the synthesized texture (i.e., structure and color) is similar to the initial sample [7].

In [8], Efros and Leung proposed a way to synthesize an unknown pixel by known pixels having similar neighborhood. This pixel-based scheme performs very well but it is very slow since the filling-in process is being done pixel by pixel. In Efros and Freeman's work [9], the speed was greatly improved by using a similar and simpler algorithm, which filled in the image using blocks of pixels.

An algorithm specific for texture inpainting was presented by Criminisi et al. in [10]. The algorithm uses the same texture synthesis techniques as the Efros-Leung algorithm. The only difference is that the pixels that are placed along the edges of the image are filled in with high priority. This slight difference was adequate to give better results as presented in [10]. Note that the texture synthesis techniques fail in geometric regions.

As real images usually contain both geometric and texture regions, Bertalmio et al. in [11] proposed a combination of structure inpainting and texture synthesis to achieve better results. They first decomposed the input image into its structure and texture components and then

978-1-4244-8594-9/10/$26.00 ©2010 IEEE

restored them, separately. At last, two reconstructed images were composed to obtain the final inpainted result.

In this paper we used the combination of structure inpainting and texture synthesis schemes to provide a robust algorithm. We used this approach with Aujol and Kang's decomposition algorithm [12].

The remainder of our work is organized as follows. In Section 2, we explain the image decomposition method. We represent the texture synthesis technique in Section 3. In Section 4, we explain the image inpainting method. Our proposed image inpainting approach is established in Section 5. Performance of our technique is examined in Section 6. Section 7 gives the concluding remarks.

II. Image Decomposition

Here, we describe the image decomposition method proposed in [12]. The authors, inspired by [13], present an algorithm to decompose a color image f into $f = u + v$, where u is the structure component and v is the color texture component of the image.

In [13], Meyer introduced an image decomposition model for gray scale images by minimizing the following functional,

$$\inf_{(u,v)\in \mathrm{BV}\times \mathrm{G}/f=u+v}\left(\int_\Omega |\nabla u| + \|v\|_G\right), \quad (1)$$

where, the first term is a total variation (TV) minimization which reduces u as the bounded variation (BV (R^2)) component of the original image f. It is well known that BV is well suited to model the structure component of f, which means that the edges of f are in the BV component u. The second term gives the v component containing the oscillatory part of the image, which is textures and noise. The Banach space G contains such signals with large oscillations. A distribution v belongs to G if v can be written as

$$v = \partial_1 g_1 + \partial_2 g_2 = \mathrm{Div}(g),\ g_1, g_2 \in L^\infty \quad (2)$$

The G-norm $\|v\|_G$ in (1) is defined as the infimum of all $\|g\|_{L^\infty} = \sup_{x\in\Omega}|g(x)|$, where $v = \mathrm{Div}(g)$ and $|g(x)| = \sqrt{|g_1|^2 + |g_2|^2}(x)$. Any function belonging to the space G can present strong oscillations, nonetheless have a small norm.

As pointed out by Aujol et al. [14], solving (1) is equivalent to computing the following image decomposition model:

$$\inf_{(u,v)\in \mathrm{BV},\|v\|_G\le\mu}\left\{J_{1D}(u) + \frac{1}{2\lambda}\|f-u-v\|_{L^2}^2\right\}, \quad (3)$$

where, $J_{1D}(u) = \int_\Omega |\nabla u|$ is the scalar total variation of u and the parameter λ and bound μ control the L_2-norm of the residual $f - u - v$ and the G-norm of the oscillating component v, respectively.

The Legendre-Fenchel transform of F is given by $F^*(v) = \sup(\langle u, v\rangle_{L^2} - F(u))$, where $\langle .,. \rangle_{L^2}$ stands for L_2 inner product. By denoting

$$\mu B_G = \{v \in G \text{ such that } \|v\|_G \le \mu\}, \quad (4)$$

it is demonstrated in [14] that

$$J_{1D}^*\left(\frac{v}{\mu}\right) = \chi_{\mu B_G}(v) = \begin{cases} 0 & \text{if } v \in \mu B_G \\ +\infty & \text{otherwise} \end{cases}. \quad (5)$$

So, (3) can be written as:

$$\inf_{(u,v)\in \mathrm{BV},\|v\|_G\le\mu}\left\{J_{1D}(u) + J_{1D}^*\left(\frac{v}{\mu}\right) + \frac{1}{2\lambda}\|f-u-v\|_{L^2}^2\right\}. \quad (6)$$

The function (6) can be minimized with respect to the variables u and v, alternatively. First, u is fixed and v is obtained as the solution of $\inf_{v\in\mu B_G}(\|f-u-v\|^2)$, and then v is fixed and u is obtained by solving $\inf_u\left\{J_{1D}(u) + \frac{1}{2\lambda}\|f-u-v\|_{L^2}^2\right\}$. So, the structure and texture components of a gray scale image are obtained.

The authors in [12] extended this idea to color images. They identified a new definition for total variation as:

$$J(u) = \int_\Omega \sqrt{|\nabla u_r|^2 + |\nabla u_g|^2 + |\nabla u_b|^2} \quad (7)$$

where, r, g, and b stand for RGB channels. They also generalized Meyer's G-norm and defined a 3D G-norm, $\vec{G}$. This resulted in proposing a function for splitting a color image f into a bounded variation component u and a texture component v,

$$\inf_{u+v=f}\{J(u) + \|v\|_{\vec{G}}\}. \quad (8)$$

To derive a numerical scheme, an L_2 residual is added to (8):

$$\inf_{u,v}\left\{J(u) + \frac{1}{2\lambda}\|f-u-v\|^2 + J^*\left(\frac{v}{\mu}\right)\right\}. \quad (9)$$

Now, to obtain the variables u and v, (9) is minimized; i.e., first with v being fixed, u is obtained as the solution of

$$\inf_u\left\{J(u) + \frac{1}{2\lambda}\|f-u-v\|^2\right\}, \quad (10)$$

and then, with u being fixed, v is obtained by solving

$$\inf_v \|f-u-v\|^2. \quad (11)$$

It should be pointed out that, instead of solving (11) to obtain variable $\tilde{v}$, one can solve the following function,

$$\inf_w\left\{J(w) + \frac{1}{2\mu}\|f-u-w\|^2\right\} \quad (12)$$

and obtain the variable $\tilde{w} = f - u - \tilde{v}$ as the solution.

The two problems (10) and (12) are minimized for obtaining their solution. For minimization, the associated Euler-Lagrange equations are computed.

Authors in [12], used the CB model for component u, and the color TV model for component v, and solved the Euler-Lagrange equation of (10) and (12) in an iterative scheme. Hence, two components u and v of image f are estimated (for more details refer to [12]).

III. Texture Synthesis

Here, we describe the method of Criminisi et al. in [10] for repairing texture component v.

This method presents an improvement of the sampling concept proposed in Efros and Leung's approach [8]. It takes isophote into consideration, and gives higher priority to "*interesting points*" on the boundary of the inpainting domain. These interesting points are parts of linear structures, and thus should be extended into the inpainting domain in order to obtain a natural look. To

identify these points, the algorithm assigns priorities to all the pixels on the boundary of the inpainting domain. According to the method, the interesting points get higher priorities, and thus the linear structures would be extended first. For each pixel p on the boundary, a patch (block) ψ_p centered at p is constructed. The patch's priority $P(p)$ is defined as the product of two terms: a *confidence* term $C(p)$, and a *data* term $D(p)$ as follows:

$$P(p) = C(p).D(p), \tag{13}$$

$$C(p) = \frac{\sum_{q\in\psi_p\cap\bar{\Omega}} C(q)}{|\psi_p|}, \quad D(p) = \frac{|\nabla f_p^{\perp}.n_p|}{\alpha}, \tag{14}$$

where, $|\psi_p|$ is the area of ψ_p, α is a normalization factor (e.g., $\alpha = 255$ for a typical gray-level image), and n_p is a unit vector orthogonal to the boundary at the point p. The priority is computed for every border patch, with distinct patch for each pixel on the boundary of the inpainting domain. $C(p)$ describes the amount of reliable information surrounding the pixel p and $D(p)$ is the strength of isophotes hitting the boundary and boosts the priority of a patch that an isophote "flows" into. $D(p)$ is very important, since it encourages linear structures to be synthesized first, and thus propagated securely into the inpainting domain.

We adopt the notations similar to that used in inpainting literature. The region to be filled, i.e., the inpainting domain, is indicated by Ω, and its boundary is denoted δΩ. The *source* region, Φ, which remains fixed throughout the algorithm, provides samples used in the filling process (refer to Fig. 1).

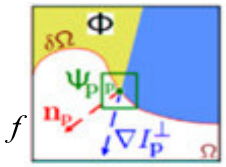


Fig. 1 Notation diagram: Given the patch ψ_p, n_p is the normal to the boundary δΩ of the inpainting domain Ω and $\nabla f_p^{\perp}$ is the isophote (direction and intensity) at point p. The entire image is denoted by f.

We now focus on a single iteration of the algorithm. For every point p on the contour δΩ, a patch ψ_p is constructed, with p in the center of the patch. The patch $\psi_{\hat{p}}$ with the highest priority is found to be filled with data extracted from the source region Φ. A global search is performed on the whole image to find a patch ψ_q that has most similarities with $\psi_{\hat{p}}$. Formally,

$$\psi_{\hat{q}} = \arg\min_{\psi_q\in\Phi} d(\psi_{\hat{p}}, \psi_q) \tag{15}$$

where, the distance d between two generic patches is simply defined as the sum of squared differences (SSD) of the already filled pixels in the two patches.
After finding the source *exemplar* $\psi_{\hat{q}}$, the value of each pixel to be filled, $p'|p' \in \psi_{\hat{q}\cap\Omega}$, is copied from its corresponding position inside $\psi_{\hat{q}}$ (*c.f.* Fig. 2).
In the last step, the confidence term $C(p)$ is updated in the area delimited by $\psi_{\hat{p}}$ as follows:

$$C(q) = C(\hat{p}), \ \forall q \in \psi_{\hat{p}} \cap \Omega. \tag{16}$$

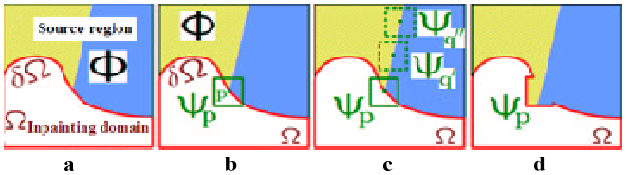


Fig. 2 Structure propagation by exemplar-based texture synthesis: (a) Original image, (b) Synthesis of the area delimited by the patch ψ_p, (c) The most likely candidate matches for ψ_p lying along the boundary between the two textures in the source region, e.g., $\psi_{q'}$ and $\psi_{q''}$, (d) the best matching patch in the candidate set copied into the position occupied by ψ_p, and thus partial filling of Ω achieved.

IV. Image Inpainting

We illustrate the image inpainting algorithm proposed by Roth and Black [6] to repair the structure component u.

Whenever images have noise or uncertainty, prior models are useful approaches to restore them. The authors in [6] propose a framework for learning image priors. The prior model used in their work was Markov random field (MRF). This approach assumes an image as the result of a random process, described by an MRF. In this model, the pixels of an image are considered as a set of nodes V in a graph $G = (V, E)$, where E are the edges connecting nodes. By defining a neighbourhood system, all nodes in an $m \times m$ rectangular region are connected. Every such neighbourhood centered by a node (pixel) k, $k = 1, \ldots, K$, defines a maximal clique $x_{(k)}$ in the graph. A clique x for $G = (V, E)$ is defined as a subset of nodes in V. It consists either of a single node $x = \{i\}$, or of a pair of neighbouring nodes $x = \{i, i'\}$, or of a triple of neighbouring nodes $x = \{i, i', i''\}$, and so on. The collections of single-node, pair-node and triple-node cliques are denoted by C_1, C_2, and C_3, respectively, where,

$$C_1 = \{i|\ i \in V\},$$
$$C_2 = \{\{i, i'\}|\ (i, i') \in E\},$$
$$C_3 = \{\{i, i', i''\}|\ \{(i, i'), (i', i''), (i'', i)\} \in E\}. \tag{17}$$

The Hammersley-Clifford theorem establishes that it is possible to write the probability density of this graphical model as a Gibbs distribution $p(x) = \frac{1}{Z} exp\left(-\sum_k V_k(x_{(k)})\right)$, where x is an image and $V_k(x_{(k)})$ is the potential function for clique $x_{(k)}$. Authors in [6] made an additional assumption that the MRF is homogenous, i.e., the potential function is the same for all cliques $\left(V_k(x_{(k)}) = V(x_{(k)})\right)$. The authors used the Field of Expert (FoE) model for computing the potential function V, i.e., $V(x_{(k)}) = E_{FoE}(x, \Theta)$, with

$$E_{FoE}(x, \Theta) = -\sum_{k=1}^{K}\sum_{i=1}^{N} \log \phi_i(J_i^T x_{(k)};\ \alpha_i), \tag{18}$$

or equivalently,

$$p(x) = \frac{1}{Z(\Theta)} \prod_{k=1}^{K}\prod_{i=1}^{N} \phi_i(J_i^T x_{(k)};\ \alpha_i), \tag{19}$$

where each expert ϕ_i is the response to a linear filter, J_i, with the following form,

$$\phi_i(x) = \phi_i\left(J_i^T x_{(k)};\ \alpha_i\right) = \left[1 + \frac{1}{2}\phi_i(J_i^T x)^2\right]^{-\alpha_i}, \quad (20)$$

N is the number of experts which is not prescribed in a particular way, and $\Theta = \{\theta_i = (J_i, \alpha_i)|\ i = 1, \ldots, N\}$ is the set of parameters to be learned. The parameters θ_i are learned from a set of D training images $X = \{x^{(1)}, \ldots, x^{(D)}\}$ by a gradient ascent on log-likelihood. This leads to the parameters being updated by

$$\delta\theta_i = \eta\left[\langle\frac{\partial E_{FoE}}{\partial\theta_i}\rangle_p - \langle\frac{\partial E_{FoE}}{\partial\theta_i}\rangle_X\right], \quad (21)$$

where, η is a user-defined learning rate, $\langle . \rangle_X$ denotes the average over the training data X, and $\langle . \rangle_p$ the expectation value with respect to the model distribution $p(x)$.

Hence, the authors use the FoE model as a part of Bayesian formulation; given an observed image y, the goal in the inpainting literature is to find the true image x that maximizes the posterior probability $p(x|y) \propto p(y|x)\,.\,p(x)$.

As the pixels in the specified regions do not have any observation, there is no likelihood term $p(y|x)$. The gradient of log-prior is simple to compute:

$$\nabla_x \log p(x) = \sum_{i=1}^{N} J_i^- * \psi_i(J_i * x), \quad (22)$$

where, $J_i * x$ denotes the convolution of image x with filter J_i, and J_i^- is obtained by mirroring filter J_i around its center, and $\psi_i(y) = {}^{\partial}/_{\partial y} \log \phi_i(y;\ \alpha_i)$.

By introducing an iteration index t, and an update rate η, the simple inpainting algorithm propagates information using only the FoE prior:

$$x^{(t+1)} = x^{(t)} + \eta M\left[\sum_{i=1}^{N} J_i^- * \psi_i\left(J_i * x^{(t)}\right)\right], \quad (23)$$

where, the matrix M is determined so that the second term in (23) is set to zero for all pixels outside of the inpainted domain.

V. Our proposed method

Our algorithm proceeds as follows.

Algorithm: *Structure and texture image inpainting*.

Step 0: decompose the input image f into a structure component u and a texture component v using the proposed method in [12] (*c.f.* Section 2).

Step 1: repair the structure component using the image inpaiting method given in [6] (*c.f.* Section 4).

Step 2: repair the texture component using the texture synthesis scheme given in [10] (*c.f.* Section 3).

Step 3: compose these two reconstructed components obtained in steps 1 and 2 to yield the result.

Details of our implementation results are given in the following section.

VI. Implementation and experimental results

Performance of the proposed inpainting method is demonstrated by some visual examples. We compare the obtained results with the ones obtained by the methods in which the image is not decomposed, and just one algorithm (either structure inpainting method or texture synthesis technique) is applied. Also, a comparison between our method and the proposed approach in [11] is made, as the proposed algorithm considers simultaneous structure and texture inpainting.

We have used the parameters in the proposed approach unchanged for all the examples considered here. We applied the image decomposition method with the number of numerical steps equal to 5, and for each step we computed the estimates for solutions of (10) and (12) using 30 iterations. The value of λ and μ were set to 0.03 and 0.01, respectively.

In the texture synthesis algorithm, the size of template window ψ is set to 9×9 pixels. This size should be slightly larger than the largest distinguishable texture element or "texel", in the source region (we defined this size to be two times the texel size). During initialization, the function $C(p)$ was valued to be $C(p) = 0, \forall p \in \Omega$, and $C(p) = 1, \forall p \in f - \Omega$.

The image inpainting scheme converted color image to YCbCr color model, and the algorithm was independently applied to all 3 channels. We used a FoE prior with 8 filters of 3×3 pixels. In order to speed up convergence, we ran 5000 iterations of (23) with $\eta = 10$. Since such a large step size may lead to some numerical instabilities, we followed this by 250 more iterations with $\eta = 0.01$.

Fig. 3 shows some results obtained by our method in comparison with the proposed methods in [6] (using only the structure inpainting approach), [10] (using only the texture synthesis technique), and [11] (using a combination of these two approaches[1]).

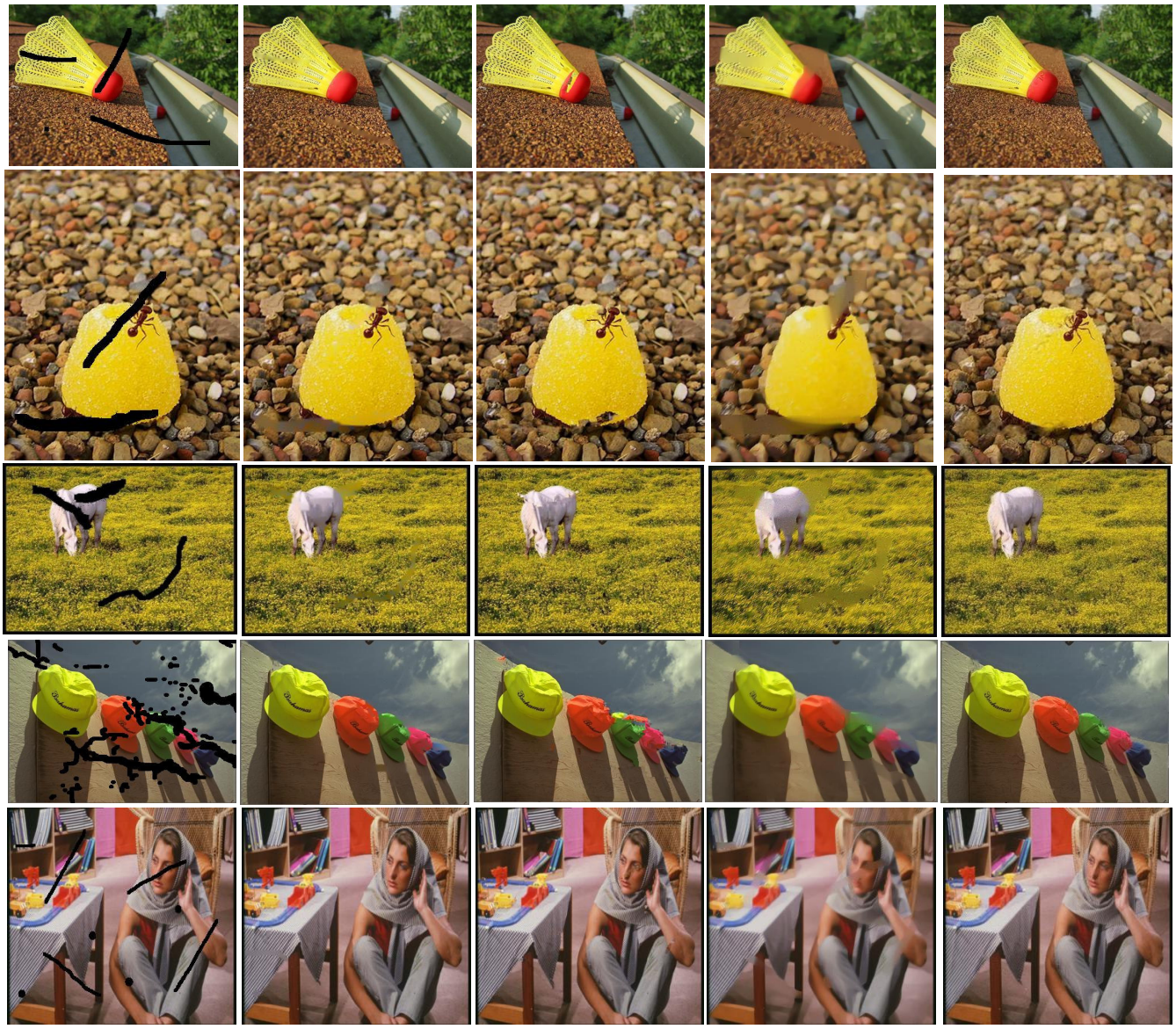
Experiments were carried out on the database presented in [16] and Kodak database [17]. Each input image is shown with inpainting domain. These regions are highlighted in black color. We found out that the restored images look plausible in general. It is obvious that the structure inpainting methods tend to blur the inpainted image, while the texture synthesis techniques fail to reconstruct areas having additional geometric information, as being observed in Fig. 3.

Also, note that our results are better than the ones obtained by the method of [11], since the decomposing method used in our approach is expected to give good quality result, as pointed out in [12].

VII. Conclusion

We presented a new approach by combining structure inpainting method and texture synthesis technique in a decomposition framework. The combination of these two powerful approaches enables us to simultaneously recover texture and geometric information. We demonstrated the quality of the results obtained by our method on a variety of defected input images and compared them to the results obtained by several other methods. In all cases, we could verify higher quality for inpainted images using our approach.

[1] Results of this method were obtained using the codes due to Shane Brennan [15].



(a) (b) (c) (d) (e)

Fig. 3. Our results in comparison with the proposed methods in [6], [10], and [11]: (a) Input image, (b) Inpainted image from [6], (c) Inpainted image from [10], (d) Inpainted image from [11], (e) Inpainted image using our method.

Since one of the applications of image inpainting is to repair old photos, finding a decomposing method which can provide good results on gray level images as well as color ones can be done as the future work.

ACKNOWLEDGMENT

The authors would sincerely thank S.H. Kang for providing us her image decomposition code and S. Brennan for his simultaneous structure and texture image inpainting code.

REFERENCES

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," *Computer Graphic* (*SIGGRAPH 2000*), pp. 417–424, July 2000.

[2] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier- stokes, fluid dynamics, and image and video inpainting," *IEEE Computer Vision and Pattern Recognition* (*CVPR*), December 2001.

[3] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and grey levels," *IEEE Trans. Image Processing*, vol. 10, pp. 1200–1211, August 2001.

[4] T. F. Chan, and J. Shen, "Mathematical models for local non-texture inpainting," *SIAM Journal of Applied Mathematics*, vol. 62, no. 3, pp.1019–1043, 2002.

[5] T. F. Chan, and J. Shen, "Non-texture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, issue 4, pp. 436-449, 2001.

[6] S. Roth, and M. J. Black, "Fields of Experts", International Journal of Computer Vision Computer, vol. 82, no. 2, pp. 205-229, 2009.

[7] M. Ashikhmin, "Synthesizing natural textures," ACM Symposium on *Interactive 3D Graphics*, pp. 217–226, 2001.
[8] A. A. Efros, and T.K. Leung, "Texture synthesis by non-parametric sampling," *IEEE International Conference on Computer Vision*, vol. 2, pp. 1033–1038, 1999.
[9] A. A. Efros, and W.T. Freeman, "Image quilting for texture synthesis and transfer," *Proc. ACM Conf. Comp. Graphics* (*SIGGRAPH*), Eugene Fiume, pp. 341–346, 2001.
[10] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar based inpainting," *IEEE Computer Vision and Pattern Recognition* (*CVPR*), vol. 2, pp. 721–728, 2003.
[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. F. Aujol and S. H. Kang. "Color image decomposition and restoration", *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, pp. 916–928, 2006.
[13] Y. Meyer, "Oscillating patterns in image processing and nonlinear evolution equations," *The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures*, vol. 22 of University Lecture Series. AMS, Providence, 2001.
[14] J.-F. Aujol, G. Aubert, L. Blanc-Fe´raud, and A. Chambolle, "Image decomposition into a bounded variation component and an oscillating component," *J. Math. Imaging Vis.*, vol. 22, no. 1, 2005.
[15] S. Brennan, "Simultaneous structure and texture image inpainting," Department of Computer Engineering, University of California at Santa Cruz, EE264 Rep. 2007, (http://users.soe.ucsc.edu/~shanerb/vitae.html).
[16] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum, "Learning to detect a salient object," *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp. 1–8, July 2007.
[17] Kodak database [online]. Available: http://rok.us/graphics/kodak/.