Color Image Completion Using Simultaneous Geometry and Texture

Bekir DIZDAROGLU

Department of Computer Engineering, Karadeniz Technical University,
Trabzon, 61080, Turkey

ABSTRACT

In this paper, a hybrid inpainting method is proposed for the simultaneous reconstruction of geometry and texture in missing regions of a color image. Using inpainting methods based on partial differential equation (PDE) to fill in large image regions usually fails if these regions contain textures. On the other hand, texture synthesis algorithms sometimes fail due to complex structure and texture in a given image. However, this study suggesting a hybrid method that consists of both techniques generates satisfactory results in completing the missing parts. In the proposed method, the given image is decomposed into two components. The geometry component is inpainted by the tensor-driven PDE algorithm which takes curvatures of line integral curves into account and the texture component is reconstructed by the modified exemplar-based inpainting algorithm. Both of these methods work based on color information. Experimental results show that the proposed method efficiently completes missing regions.

Keywords: Color Image Decomposition, Color Image Inpainting, Geometry, Texture, Texture Synthesis.

1. INTRODUCTION

In many cases, there might be some missing parts on images and videos, which may affect the appearance far too much in a negative manner. Completing these missing parts properly is very important in editing images and videos for restoration (detection and removal of blotches in old pictures or films), correction (fixing undesirable defects such as red-eyes on photos), and manipulation (making tricky photos and effects on the images).

There are two categories of image reconstruction methods mentioned in the literature. These are called texture synthesis method and inpainting method which has received more attention in recent years. The first is generally used to reconstruct the large regions of the images and the second is used to fill in the small image holes. Tschumperlé *et al.* [1] proposed the trace-based PDE method which is not very successful because of rounding the corners in images. Then, Tschumperlé [2] resolved this problem by employing an algorithm based on curvature-preserving PDE. Criminisi *et al.* [3] presented an exemplar-based image inpainting algorithm to remove large objects from images. The method fills in the missing regions by sample patches. However, sometimes artifacts are seen in output image generated by filling in the missing parts of the input image. Dizdaroglu *et al.* [4] introduced the spatiotemporal exemplar-based image inpainting method to complete damaged regions in image sequences. But this method can also fail if it is tested on only one input image. Bertalmio *et al.* [5] proposed an inpainting method. Since the method does not employ color information, it sometimes is unable to detect the image contours and the filling process cannot be completed successfully. Harald [6] introduced a combined approach which is limited to the removal of small image holes, while continuing to blur the completed region.

Since there are some little deficiencies in previous methods, we presented a method based on processing both geometry and texture images simultaneously in order to obtain the visually plausible reconstructed image.

2. THE PROPOSED METHOD

The proposed method contains image decomposition, inpainting and texture synthesis.

Let $\mathbf{I}: \Omega \to \mathbb{R}^n$ be color image $(n = 3)$, defined on domain $\Omega \to \mathbb{R}^2$, and $I_i: \Omega \to \mathbb{R}$ represents the image channel i of $\mathbf{I}$ $(1 \leq i \leq n)$: $\forall \mathbf{p} = (x, y) \in \Omega$. The method is explained in detail in the following sections.

Image Decomposition

In equation $\mathbf{I} = \mathbf{I}_\mathrm{u} + \mathbf{I}_\mathrm{v}$, which shows the entire image, $\mathbf{I}_\mathrm{u}$ and $\mathbf{I}_\mathrm{v}$ represent geometry image and texture image, respectively. According to the equation, it can be seen that the texture image $\mathbf{I}_\mathrm{v}$ is generated by subtracting $\mathbf{I}_\mathrm{u}$ from $\mathbf{I}$. Texture information can be considered as a kind of noise while geometry information can be considered as a cartoon version with sharp edges. Therefore, any denoising algorithm based on PDE can be used in order to obtain geometry image. In this study, the trace-based method is preferred among those algorithms [1].

Denoising or regularization of an image $\mathbf{I}$ based on PDEs is a smoothing operation along the direction defined based on intensity distributions of pixels on the region. Here, the smoothing operation should not blur the edges, which could be ensured by avoiding smoothing orthogonally to edges. In order to do this, the local geometry of image should be obtained in the first place.

Following conditions consisting of important features are provided in every image point $\mathbf{p} = (x, y) \in \Omega$ [1, 2]:

- Two orthogonal directions $\theta^+(\mathbf{p}), \theta^-(\mathbf{p}) \in \mathrm{S}^1$ (unit vectors of R^2) determined by maximum and minimum intensity variations at point **p** are defined to show the gradient vector, and the edge if there is any, respectively.

- The positive values $\lambda^{+}(\mathbf{p})$, $\lambda^{-}(\mathbf{p})$ are calculated related to clarity of the edges to show effective variations of the image intensities along $\theta^{+}(\mathbf{p})$ and $\theta^{-}(\mathbf{p})$.

The local geometry $\{\lambda^{+/-}, \theta^{+/-} | \mathbf{p} \in \Omega\}$ of scalar images $I: \Omega \rightarrow \mathbb{R}$ is calculated based on the gradient field ∇I or smoothed gradient field $\nabla I_{\sigma} = \mathrm{G}_{\sigma} * \nabla I$. Here, $\mathrm{G}_{\sigma} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$ is a 2D Gaussian kernel with a variance σ.

The *image gradient* is the derivation of a *scalar image I* related to the spatial coordinates $\mathbf{p} = (x, y)$:

$$\nabla I = (I_x, I_y)^T = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)^T \quad (1)$$

A vector $\nabla I: \Omega \rightarrow \mathbb{R}^2$ is created according to image gradient to represent magnitudes of the scalar image I and the maximum variation directions. Scalar and pointwise measure of the image variations are given by the *gradient norm* $\|\nabla I\|$ which is used in image analysis in many cases:

$$\|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (2)$$

Therefore, $\lambda^{+} = \|\nabla I\|^2$ and $\theta^{-} = \eta^{\perp} = \nabla I^{\perp}/\|\nabla I\|$ are possible measures of the intensity and direction of the image edge respectively (See Figure 1).

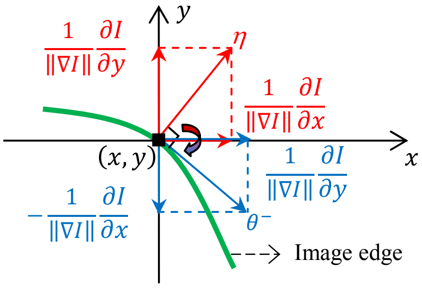

$\frac{1}{\|\nabla I\|}\frac{\partial I}{\partial y}$
y
η
$\frac{1}{\|\nabla I\|}\frac{\partial I}{\partial x}$
(x, y)
x
$\frac{1}{\|\nabla I\|}\frac{\partial I}{\partial y}$
$-\frac{1}{\|\nabla I\|}\frac{\partial I}{\partial x}$
θ^{-}
Image edge

Figure 1: η and θ^{-} vectors at point $\mathbf{p} = (x, y)$

Here, $\mathbf{G}: \Omega \rightarrow \mathrm{P}(2)$ of 2 × 2 symmetric and semi-positive tensors, denoted by $\forall \mathbf{p} \in \Omega$, $\mathbf{G}(\mathbf{p}) = \lambda^{-}\theta^{-}\theta^{-T} + \lambda^{+}\theta^{+}\theta^{+T}$, may be utilized to represent $\{\lambda^{-/+}, \theta^{-/+} | \mathbf{p} \in \Omega\}$ more properly. Eigenvalues of **G** are λ^{-} and λ^{+} and related eigenvectors are θ^{-} and θ^{+}. For example, the tensor $G(\mathbf{p}) = \nabla I(\mathbf{p}) \nabla I(\mathbf{p})^T$ can be used to show the local geometry of scalar-valued images I.

The local geometry for multi-valued images $\mathbf{I}: \Omega \rightarrow \mathbb{R}^n$ can be obtained in a similar way, by computing the field **G** of *geometry tensors*. Therefore, the gradient of multi-valued images is expressed as follows:

$$\forall \mathbf{p} \in \Omega,\ \mathbf{G}(\mathbf{p}) = \textstyle\sum_{i=1}^{n} \nabla I_i \nabla I_i^T \quad \text{where } \nabla I_i = \begin{pmatrix} \frac{\partial I_i}{\partial x} \\ \frac{\partial I_i}{\partial y} \end{pmatrix} \quad (3)$$

G is defined as the following for color images $\mathbf{I} = (R, G, B)$:

$$\mathbf{G} = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} = \begin{pmatrix} R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_y R_x + G_y G_x + B_y B_x & R_y^2 + G_y^2 + B_y^2 \end{pmatrix} \quad (4)$$

The positive eigenvalues $\lambda^{+/-}$ and the orthogonal eigenvectors $\theta^{+/-}$ of **G** are formulized as follows:

$$\lambda^{+/-} = \frac{g_{11} + g_{22} \pm \sqrt{\Delta}}{2} \quad (5)$$

and

$$\theta^{+/-} = \begin{pmatrix} 2g_{12} \\ g_{22} - g_{11} \pm \sqrt{\Delta} \end{pmatrix} \quad (6)$$

where $\Delta = (g_{11} - g_{22})^2 + 4g_{12}^2$.

The *gradient norm* $\|\nabla \mathbf{I}\|$ of color image is easy to compute as follows since it perceives image structures successfully (See Figure 2):

$$\|\nabla \mathbf{I}\| = \sqrt{\lambda^{+} + \lambda^{-}} = \sqrt{\sum_{i=1}^{n} \|\nabla I_i\|^2} \quad (7)$$

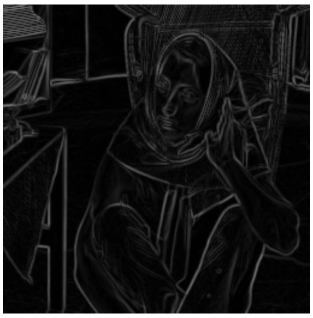


Figure 2: Gradient norm of Barbara's color image.

More consistent geometry is obtained provided that $\mathbf{G}_{\sigma} = \mathbf{G} * \mathrm{G}_{\sigma}$ is smoothed by Gaussian filter. Here, $\mathbf{G}_{\sigma}$ is a good estimator of the local multi-valued (color) geometry of **I** at point **p**, and its spectral elements give the vector-valued variations (by the eigenvalues λ^{-}, λ^{+} of $\mathbf{G}_{\sigma}$) at the same time and the orientations (edges) of the local image structures (by the eigenvectors $\theta^{-} \perp \theta^{+}$ of $\mathbf{G}_{\sigma}$).

Tschumperlé *et al.* [1] suggested designing a particular field $\mathbf{T}: \Omega \rightarrow \mathrm{P}(2)$ of *diffusion tensors* to define the specification of the local smoothing method for the regularization process. Apparently, it should be noticed that **T** depending on the local geometry of **I** is thus defined from the spectral elements $\lambda^{+}, \lambda^{-}$and θ^{+}, θ^{-} of $\mathbf{G}_{\sigma}$:

$$\forall \in \Omega,\ \mathbf{T} = f^{-}(\lambda^{+}, \lambda^{-})\theta^{-}\theta^{-T} + f^{+}(\lambda^{+}, \lambda^{-})\theta^{+}\theta^{+T} \quad (8)$$

The strengths of smoothing along θ^{-}, θ^{+} are set by two functions denoted by $f^{-/+}: \mathbb{R}^2 \rightarrow \mathbb{R}$, where the types of applications determine f^- and f^+. Sample functions for image denoising are proposed in [1]:

$$f^-(\lambda^+,\lambda^-) = \frac{1}{(1+\lambda^++\lambda^-)^{p1}} \text{ and } f^+(\lambda^+,\lambda^-) = \frac{1}{(1+\lambda^++\lambda^-)^{p2}} \quad (9)$$

with $p1 < p2$

Here, the goal of smoothing operation is:

- The pixels on image edge are smoothed along θ^- with a strength inversely relative to the vector edge strength (anisotropic smoothing).
- The pixels on homogeneous regions are smoothed along all possible directions (isotropic smoothing).

The tensor directions for Barbara's color image are shown in Figure 3.

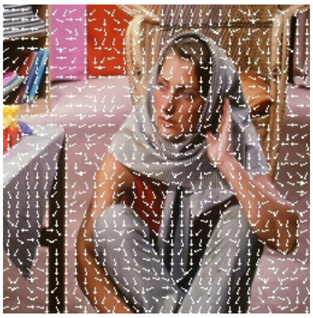


Figure 3: Tensor directions of Barbara's image.

Tschumperlé *et al.* [1] suggests a regularization PDE to agree with local smoothing geometry **T** based on a *trace* operator:

$$\frac{\partial \mathbf{I}}{\partial t} = \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{TH}_i) \quad (10)$$

where $\mathbf{H}_i$ is the Hessian matrix of I_i:

$$\mathbf{H}_i = \begin{pmatrix} \dfrac{\partial^2 I_i}{\partial x^2} & \dfrac{\partial^2 I_i}{\partial x \partial y} \\ \dfrac{\partial^2 I_i}{\partial y \partial x} & \dfrac{\partial^2 I_i}{\partial y^2} \end{pmatrix} \quad (11)$$

In this study, $\mathbf{H}_i$ is a *symmetric matrix* since the images are regular ones, $\frac{\partial^2 I_i}{\partial x \partial y} = \frac{\partial^2 I_i}{\partial y \partial x}$.

As Tschumperlé *et al.* [1] have shown, equation (10) can be viewed as local filtering with oriented and normalized Gaussian kernels. Here, a small convolution is locally applied around each **p** with a 2D Gaussian mask $G_t^{\mathbf{T}}$ *oriented* by the tensor $\mathbf{T}(\mathbf{p})$:

$$G_t^{\mathbf{T}}(\mathbf{p}) = \frac{1}{4\pi t}\exp\left(-\frac{\mathbf{p}^T\mathbf{T}^{-1}\mathbf{p}}{4t}\right) \quad (12)$$

As a matter of fact, a link exists between anisotropic diffusion PDE and classical filtering techniques:

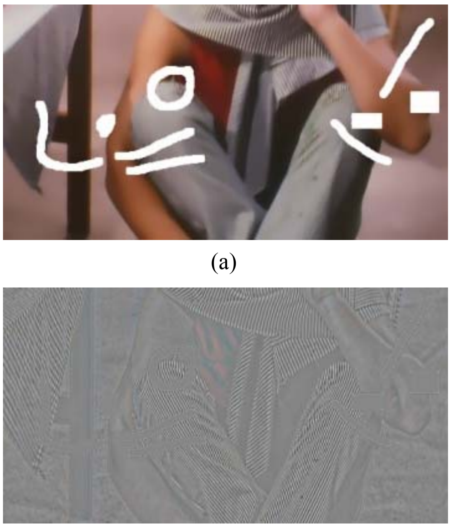
$$\frac{\partial \mathbf{I}}{\partial t} = \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{TH}_i) \Leftrightarrow \partial I_{i(t)} = I_{i(t=0)} * \mathrm{G}_\mathrm{t}^{\mathbf{T}} \quad (13)$$

The regularization PDE shown in equation (14) is compatible with all local geometric properties expressed above:

$$\begin{cases} \mathbf{I}_{(t=0)} = \mathbf{I}_{noisy} \\ \mathbf{I}_{(t+1)} = \mathbf{I}_{(t)} + dt\dfrac{\partial \mathbf{I}_{(t)}}{\partial t} \end{cases} \quad (14)$$

where dt stands for time step.

The suggested technique for image decomposition is used to generate the image geometry. Figure 4 shows results of the decomposition algorithm on the image provided by the proposed method.



(a)

(b)

Figure 4: Decomposed components of the input image: Close-ups of the geometry image (a) and the texture image (b).

Inpainting

The trace-based method behaves locally as an oriented Gaussian smoothing filter. The strength and orientation of this smoothing filter is directly related with the tensor **T**. This method preserves edges, but it tends to cause unacceptable rounding effects on curved structures such as corners. It is because the Gaussian kernel does not consider curvatures. For this reason, Tschumperlé proposed another technique [2] which can be considered filtering the image locally with the normalized one-dimensional Gaussian kernel. In this technique, the smoothing geometry *tensor field* **T** is split into sum of *vector fields*, and then *line integral convolutions* (LICs) is performed on the image with each of these vector fields. One-dimensional Gaussian kernel function is shown below:

$$G_t(x) = \frac{1}{\sqrt{4\pi t}}\exp\left(-\frac{x^2}{4t}\right) \quad (15)$$

Basically, Tschumperlé conducts one-dimensional convolutions on the image using $G_t(x)$ along the vector fields obtained from the tensor **T**:

$$\mathbf{T} = \frac{2}{\pi}\int_{\alpha=0}^{\pi} \left(\sqrt{\mathbf{T}}a_\alpha\right)\left(\sqrt{\mathbf{T}}a_\alpha\right)^T d\alpha \quad (16)$$

where $a_\alpha = (\cos\alpha \quad \sin\alpha)^T$ and $\sqrt{\mathbf{T}} = \sqrt{f^-}\mathrm{uu} + \sqrt{f^+}\mathrm{vv}$ is the square root of $\mathbf{T}$. $\left(\sqrt{\mathbf{T}}\right)^{\mathbf{2}} = \mathbf{T}$ and $\left(\sqrt{\mathbf{T}}\right)^T = \sqrt{\mathbf{T}}$ can be easily verified.

The PDE solved by Tschumperlé [2] is:

$$\frac{\partial I_i}{\partial t} = \mathrm{trace}(\mathbf{TH}_i) + \frac{2}{\pi}\nabla I_i^T \int_{\alpha=0}^{\pi} \mathbf{J}_{\sqrt{\mathbf{T}}a_\alpha}\sqrt{\mathbf{T}}a_\alpha d\alpha \qquad (17)$$

where $\mathbf{J}_{\sqrt{\mathbf{T}}a_\alpha}$ stands for the Jacobian of $\Omega \to \sqrt{\mathbf{T}}a_\alpha$.

This solution can also be expressed as below:

$$\mathbf{I}^{[t+dt]} = \frac{1}{N}\sum_{k=0}^{N-1} \mathbf{I}^{[t]}_{\mathrm{LIC}(\sqrt{\mathbf{T}}a_\alpha)} \qquad (18)$$

where each Gaussian variance has a standard deviation dt, $\mathbf{I}^{[t]}_{\mathrm{LIC}}(.)$ stands for the line integral convolution.

The most difficult operation here is the LIC computation, which requires the tracking of integral curves of a vector field. In order to solve this problem a very simple technique based on the classical Runge-Kutta integration scheme is employed by Tschumperlé [2]. Although another faster LIC computation has developed in [7], it does not utilize Gaussian weighting functions, as needed here:

$$\mathbf{I}^{[t]}_{\mathrm{LIC}}(\mathbf{p}) = \int_{-\infty}^{+\infty} \mathbf{I}^{[t=0]}\left(C^{\mathbf{p}}(x)\right)G_t(x)dx \qquad (19)$$

where $C^{\mathbf{p}}(x)$ is the curve defining the *integral curve* of $\mathbf{w}$, starting from $\mathbf{p}$ and parameterized by $x \in \mathbb{R}$ (See Figure 5):

$$\begin{cases} C^{\mathbf{p}}(0) = \mathbf{p} \\ \dfrac{\partial C^{\mathbf{p}}(x)}{\partial x} = \mathbf{w}\left(C^{\mathbf{p}}(x)\right) \end{cases} \qquad (20)$$

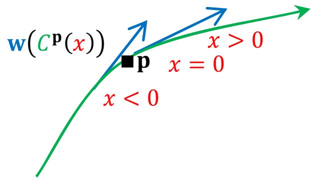


Figure 5: Integral curve $C^{\mathbf{p}}$ of vector fields $\mathbf{w}\colon \Omega \to \mathbb{R}^2$.

The outlines of the implementation of suggested algorithm for one iteration is as follows:

1) The smoothed geometry field $\mathbf{G}_\sigma$ is computed from $\mathbf{I}^{[t]}$.
2) The eigenvalues and eigenvectors of $\mathbf{G}_\sigma$ are computed.
3) The diffusion tensor field $\mathbf{T}$ is computed from $\mathbf{G}_\sigma$.
4) For α in all $[0, \pi]$:
   - The vector field $\mathbf{w} = \sqrt{\mathbf{T}}a_\alpha$ is computed.
   - LIC operation of $\mathbf{I}^{[t]}$ is applied in forward and backward directions along $C^{\mathbf{p}}$.
5) The average of all LICs is computed in step 4.

The result of the inpainting algorithm on the geometry image is depicted in Figure 6.

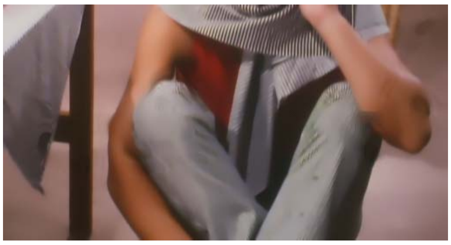


Figure 6: Close-up of the inpainted geometry image.

Texture Synthesis

The texture synthesis method is based on the existing image inpainting method presented in [3]. However, color information is not taken into account in the existing algorithm, while obtaining the data term and attempting to find the best sample patch in the search region.

The method represented in Figure 7 requires a clarification: In the drawing, $\mathbf{I}(\mathbf{p})$ is a pixel value in the input image, Γ_i is the region that needs some reconstruction, $\delta\Gamma_i$ is boundary of Γ_i, and Φ_i is the search region that is composed of sample patches. $\Psi_i^{\mathbf{p}}$ is the actual patch that is to be filled at point $\mathbf{p}$ on $\delta\Gamma_i$. The filling priority of boundary points on the target area is found as follows:

$$\boldsymbol{\rho}(\mathbf{p}) = \boldsymbol{\gamma}(\mathbf{p})\mathbf{D}(\mathbf{p}) \qquad (21)$$

$$\begin{aligned} \boldsymbol{\gamma}(\mathbf{p}) &= \frac{1}{n}\sum\nolimits_{i=1}^{n} \gamma_i(\mathbf{p}) \\ \gamma_i(\mathbf{p}) &= \sum\nolimits_{\mathbf{q}\in\Psi_i^{\mathbf{p}}\cap\Phi_i} \frac{\gamma_i(\mathbf{q})}{Area\left(\Psi_i^{\mathbf{p}}\right)} \end{aligned} \qquad (22)$$

$$\begin{aligned} \mathbf{D}(\mathbf{p}) &= \frac{1}{n}\sum\nolimits_{i=1}^{n} \mathrm{D}_i(\mathbf{p}) \\ \mathrm{D}_i(\mathbf{p}) &= \frac{|\mathrm{trace}(\mathbf{TH}_i)|}{255} \end{aligned} \qquad (23)$$

where $\gamma_i(\mathbf{p})$ shows confidence term indicating the filling priority from the outer layers of the target region towards inner layers, and $\mathrm{D}_i(\mathbf{p})$ is the term giving the priority based on the gradient values such as edge information. $Area\left(\Psi_i^{\mathbf{p}}\right)$ is the area of $\Psi_i^{\mathbf{p}}$.

$\gamma_i(\mathbf{p})$ is set to following values during the initialization:

$$\gamma_i(\mathbf{p}) = \begin{cases} 1, \forall\mathbf{p} \in \Phi_i \\ 0, \forall\mathbf{p} \in \Gamma_i \end{cases} \qquad (24)$$

In this study, the suggested method considers the color information in calculating the distance between two patches, which is defined as the sum of squared differences, and then performs the search process in the search region only. This increases the performance of the algorithm compared to the techniques considering the whole image instead of the search region only. Here, after finding maximum priority patch $\Psi_i^{\hat{\mathbf{p}}}$, the best model patch is investigated within the search regions, where the distance $\mathbf{d} = \sum_{i=1}^{n} \mathrm{d}_i\left(\Psi_i^{\hat{\mathbf{p}}}, \Psi_i^{\hat{\mathbf{q}}}\right)$ between the patches $\Psi_i^{\hat{\mathbf{p}}}$ and $\Psi_i^{\hat{\mathbf{q}}}$ is calculated as the sum of squared differences (SSD) of pixels that have already been filled. The best sample patch is copied from the search region to the target region. The confidence values are updated in the last step.

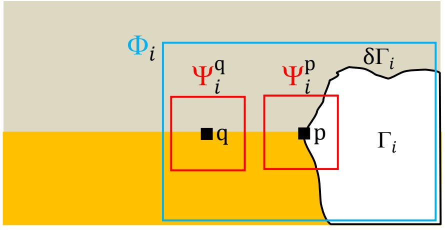


Figure 7: The modified exemplar based inpainting diagram.

Figure 8 shows the result of texture synthesis operation suggested in the study on the texture image.

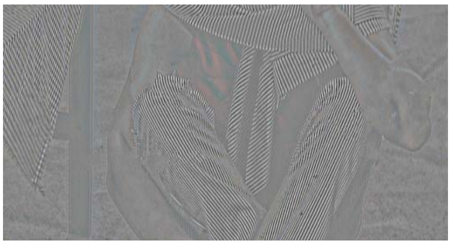


Figure 8: Close-up of the completed texture image.

3. EXPERIMENTAL RESULTS

The performance tests of our image completion method are done on 304 x 192 color image by removing the balustrade and 510 x 516 Barbara's image by completing the artificially damaged regions as seen in Figure 9. The search regions for the modified exemplar-based inpainting method on the images are taken as 40 x 40 and 45 x 45 pixels respectively. Other parameters are kept the same as in referenced papers.

Results of the tests can be viewed in Figure 10 and 11. Figure 10.a-b shows the result of methods proposed in [6] and [3] respectively. Not like both of those methods, our modified exemplar-based inpainting method produced a perfect result on brick wall and trees as seen in Figure 10.c. Therefore there is no need to apply the other proposed method based on the summation of geometry and texture components. Figure 11.a-c show the constrained PDE based inpainting method [2], the modified exemplar-based inpainting method and the proposed method, respectively. As can be seen in those results, the proposed method gives the best visual quality compared to the others.

The methods were implemented in Microsoft Visual C++ 2005 by taking advantage of *CImg Library* [8]. The program was run on a PC with Pentium 2.20 GHz processor and 2 GB RAM. The runtime of the proposed method is about 4 minutes.

4. CONCLUSION

In this work, we proposed a method that combines the advantages of inpainting and texture synthesis approaches. Both approaches are separately applied to the decomposed images. Results of both approaches are then combined to reconstruct the output image. The result shows that the output image is of acceptable quality.

As a future task automatic search capability could be developed for search regions in the modified exemplar inpainting method since the dimension of search region is manually arranged at present. There is also color scattering on some of the completed parts in the constrained PDE based method. This drawback is also to be eliminated as another future task.

5. REFERENCES

[1] D. Tschumperlé and R. Deriche, "Vector-Valued Image Regularization with PDEs : A Common Framework for Different Applications", **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 27, No.4, 2005.

[2] D. Tschumperlé, "Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDEs", **International Journal of Computer Vision**, Vol. 68, No.1, 2006, pp. 65-82.

[3] P. Perez Criminisi and K. Toyama, "Object Removal by Exemplar-based Inpainting", **IEEE Conference on Computer Vision and Pattern Recognition**, Vol. 2, June 2003, pp. 721–728.

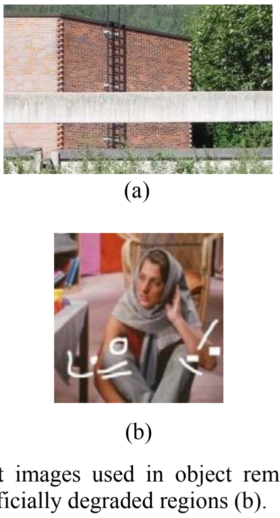
[4] B. Dizdaroglu and A. Gangal, "A Spatiotemporal Algorithm for Detection and Restoration of Defects in Old Color Films", **Lecturer Notes in Computer Sciences**, Vol. 4678, 2007, pp. 509-520.

[5] M. Bertalmio, L. Vese, G. Sapiro and S. Osher. "Simultaneous Structure and Texture Image Inpainting", **IEEE Transactions on Image Processing**, Vol. 12, 2003, pp. 882-889.

[6] Harald G. "Combined PDE and Texture Synthesis Approach to Inpainting". **Proceedings of ECCV**, 2: 214–224, 2004.

[7] D. Stalling and H.C. Hege. "Fast and Resolution Independent Line Integral Convolution", **ACM SIGGRAPH, 22nd Annual Conference on Computer Graphics and Interactive Technique**, pp. 249–256, 1995.

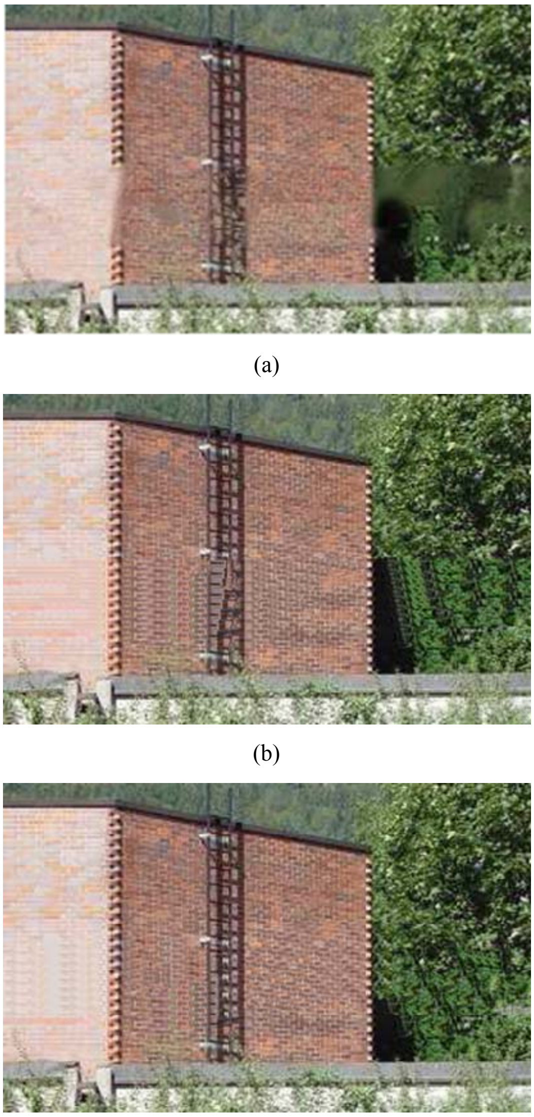
[8] D. Tschumperlé, The CImg Library: **http://cimg.sourceforge.net**, The C++ Template Image Processing Library.



(a)

(b)

Figure 9: Input images used in object removal (a) and completing artificially degraded regions (b).

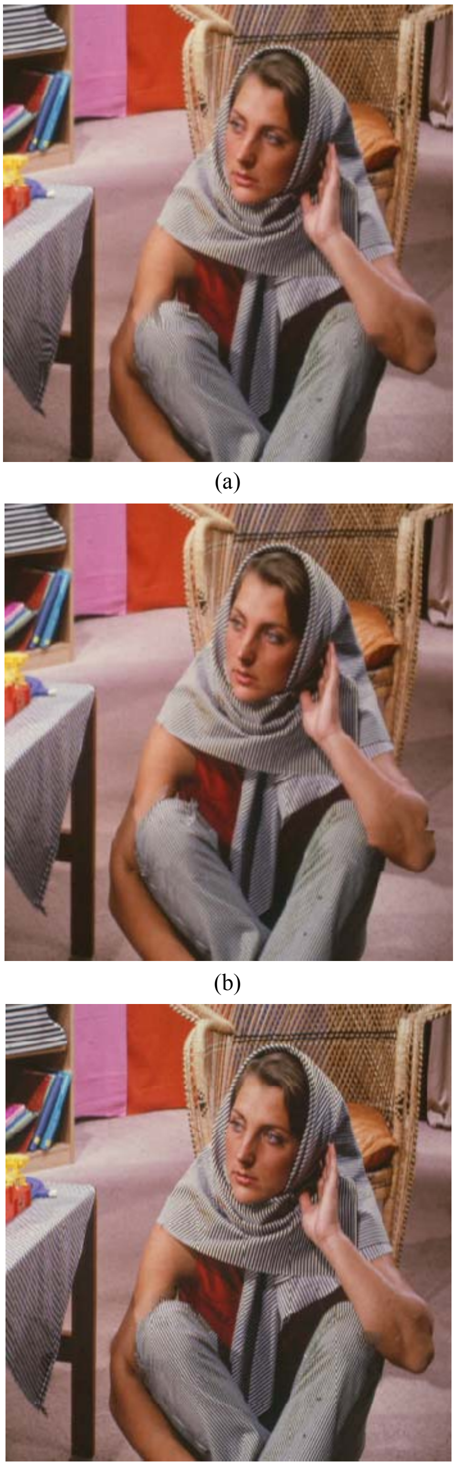


(a)

(b)

(c)

Figure 10: The method in [6] (a), the exemplar-based inpainting method [3] (b) and the modified exemplar-based inpainting method (c).



(a)

(b)

(c)

Figure 11: The constrained PDE based method [2] (a), the modified exemplar-based inpainting method (b) and the proposed method (c).