Robust digital image inpainting algorithm in the wireless environment

Karapetyan, G., Sarukhanyan, H., Agaian, S.

G. Karapetyan, H. G. Sarukhanyan, S. S. Agaian, "Robust digital image inpainting algorithm in the wireless environment," Proc. SPIE 9120, Mobile Multimedia/Image Processing, Security, and Applications 2014, 91200O (22 May 2014); doi: 10.1117/12.2049942

Event: SPIE Sensing Technology + Applications, 2014, Baltimore, Maryland, United States

Robust Digital Image Inpainting Algorithm in the Wireless Environment

G.Karapetyan[a], H.G.Sarukhanyan[a], S.S.Agaian[b],
[a]Institute for Informatics and Automation Problems of NAS RA;
gevorgk@ipia.sci.am , hakop@ipia.sci.am
[b]University of Texas at San Antonio, USA; sos.agaian@utsa.edu

ABSTRACT

Image or video inpainting is the process/art of retrieving missing portions of an image without introducing undesirable artifacts that are undetectable by an ordinary observer. An image/video can be damaged due to a variety of factors, such as deterioration due to scratches, laser dazzling effects, wear and tear, dust spots, loss of data when transmitted through a channel, etc. Applications of inpainting include image restoration (removing laser dazzling effects, dust spots, date, text, time, etc.), image synthesis (texture synthesis), completing panoramas, image coding, wireless transmission (recovery of the missing blocks), digital culture protection, image de-noising, fingerprint recognition, and film special effects and production. Most inpainting methods can be classified in two key groups: global and local methods. Global methods are used for generating large image regions from samples while local methods are used for filling in small image gaps. Each method has its own advantages and limitations. For example, the global inpainting methods perform well on textured image retrieval, whereas the classical local methods perform poorly. In addition, some of the techniques are computationally intensive; exceeding the capabilities of most currently used mobile devices. In general, the inpainting algorithms are not suitable for the wireless environment.

This paper presents a new and efficient scheme that combines the advantages of both local and global methods into a single algorithm. Particularly, it introduces a blind inpainting model to solve the above problems by adaptively selecting support area for the inpainting scheme. The proposed method is applied to various challenging image restoration tasks, including recovering old photos, recovering missing data on real and synthetic images, and recovering the specular reflections in endoscopic images. A number of computer simulations demonstrate the effectiveness of our scheme and also illustrate the main properties and implementation steps of the presented algorithm. Furthermore, the simulation results show that the presented method is among the state-of-the-art and compares favorably against many available methods in the wireless environment. Robustness in the wireless environment with respect to the shape of the manually selected “marked” region is also illustrated. Currently, we are working on the expansion of this work to video and 3-D data.

Keywords: Image inpainting, object removal, old photos recovery, texture and structure propagation, endoscopic images texture synthesis

1. INTRODUCTION

The term *inpainting* is taken from the art world, where skillful art conservators or art restorers are tasked with reconstructing the missing or damaged areas of the art work [1]. In the digital domain, the problem of reconstruction of missing areas first appeared under the name of “error concealment” in telecommunications. The term inpainting first was introduced in [2]. The inpainting can also be viewed as an interpolation or a disocclusion problem. The classical image inpainting problem can be stated as follows. Suppose the ideal complete image is I defined on a finite domain C (the plane), and its degraded region Ω . In general, we may formulate the inpainting problem in the following way: given an image with a degraded region Ω , fill-in each pixel inside Ω with a value taken from $I-\Omega$.

An image/video can be damaged due to a variety of factors, such as deterioration due to scratches, laser dazzling effects, wear and tear, dust spots, loss of data when transmitted through a channel, etc. Applications of inpainting include image restoration (removing laser dazzling effects, dust spots, date, text, time, etc.), image synthesis

Mobile Multimedia/Image Processing, Security, and Applications 2014, edited by Sos S. Agaian, Sabah A. Jassim, Eliza Yingzi Du, Proc. of SPIE Vol. 9120, 91200O · © 2014 SPIE
CCC code: 0277-786X/14/$18 · doi: 10.1117/12.2049942

(texture synthesis), completing panoramas, image coding, wireless transmission (recovery of the missing blocks), digital culture protection, image de-noising, fingerprint recognition, and film special effects and production [1,2,5-15].

Mobile computing technologies have a rapid growth these years. New mobile devices have multicore processors, which give ability to perform a number of computationally expensive computer vision algorithms such as augmented reality, object detection, image editing etc., [3, 4]. One of the major usage of the mobile phone is taking images via camera of the device. Before sharing the images via various social networks such as Facebook, Twitter, Google+ etc. users usually enhance their images. Such enhancement is removing of the unwanted objects from digital images via digital inpainting [5, 6]. The mobile device can be used for another inpainting task such as, removal of specular reflections in endoscopic images [7]. In general the inpainting algorithms are computationally expensive and are not suitable for the wireless environment.

In recent years, there have been great progresses on image inpainting, see [1, 2, 5-15] and the references therein. Most inpainting methods can be classified in two key groups: global and local methods. Global methods are used for generating large image regions from samples while local methods are used for filling in small image gaps. Each method has its own advantages and limitations. For example, the global inpainting methods such as exemplar based inpainting [5] perform well on textured image retrieval but are computationally expensive. The local methods which fill the missing regions via propagating the information from adjacent areas perform relatively fast but cannot be applied on large regions [11-17].

This paper presents a new and efficient scheme that combines the advantages of both local and global methods into a single algorithm. Particularly, it introduces a blind inpainting model to solve the above problems by adaptively selecting support area for the inpainting scheme. We analyze the binary mask of the image, which indicates the regions to be inpainted, separate the areas to be inpainted into independent objects. Then for each object we choose inpainting method global or local and define appropriate restricted support area for it. We apply our algorithm to a variety of images, ranging from purely synthetic images to full-color photographs that include complex textures. We also make comparisons to previously proposed methods.

The remaining sections are organized as follows: Section II describes the framework for robust digital inpainting which form the basis of our algorithm. The computer simulation results on both synthetic images and real-scene photographs are presented in Section III. Finally, we conclude this work in Section IV.

2. FRAMEWORK FOR ROBUST DIGITAL INPAINTING

The main procedure of the proposed inpainting algorithm is illustrated in Figure. 1. The proposed framework starts by selection of targeted regions to be inpainted in digital image. After selection of targeted regions a binary mask is generated, which indicates the targeted areas.
In next step we analyze the binary mask, and separate the targeted regions into independent working areas. Based on the size of the working area we perform appropriate inpainting method. Each step of the inpainting process (See Figure 1) is described in detail below.

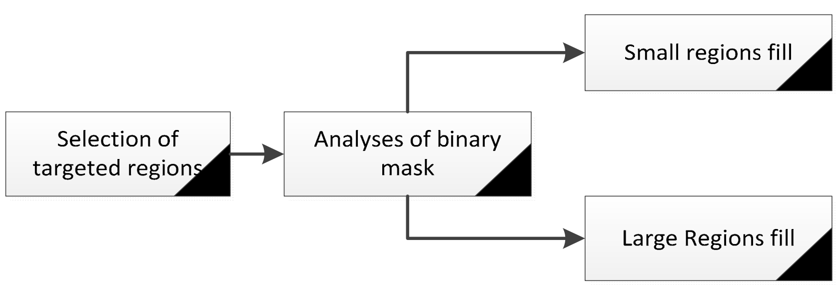


Figure 1.Block diagram of targeted regions inpainting process.

The inpainting framework consists of 5 main parts :

Input : An input image with damaged/unwanted regions to be inpainted
Step 1. Selection of damaged/targeted regions
Step 2. Analyses of the binary mask, obtain the separate working areas
Step 3. Classify the working areas into small areas and large areas

Step 4. For small areas implement small regions fill algorithm. Update input image
Step 5. For large areas implement large regions fill algorithm. Update input image
Output: Inpainted Image

2.1 Selection of targeted regions

This section presents the targeted regions selection procedure. The first step is selection of targeted regions to be inpainted in digital images. The regions to be inpainted can be selected automatically for specific type of target regions such as specular reflections medical images, cracks and dust in old photo images, etc., We consider user interactive selection of targeted regions, when user selects the unwanted regions via screen sensor of mobile device.
The selection curves are drawn with the help of Bezeir [18] curves. On screen touch, we can obtain the coordinates of current point $P_0(x_0, y_0)$. Then on finger move we measure the distance of the current point from the starting pixel. If it equals to predefined threshold t_B, then we store that value and mark it as $P_1(x_1, y_1)$.
Then the user continues the movement of his finger we calculate the distance of ongoing point from P_1. When the distance is equal to predefined threshold, t_B we store that point as P_2. The set of points P_0, P_1, P_2 are used for building $B(t)$ Bezeir quadratic curve, which is calculated below:

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, \quad t \in [0,1]$$

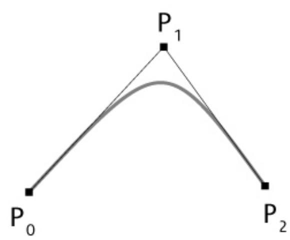
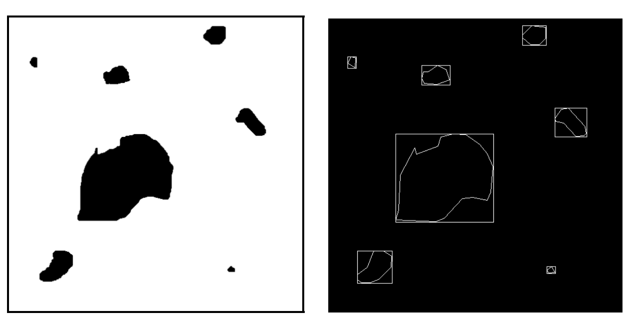


Figure 2. Bezeir curve built with usage of P_0, P_1, P_2 points.

Then P_2 is set as P_0 and the curve building procedure continues till the user takes the finger up. After selection of all targeted regions we generate binary mask which will be used in concealment inpainting algorithm.

2.2 Analyses of binary mask

This section presents the binary mask selection and switching procedure. After obtaining the binary mask, which swiindicates the regions to be inpainted we separate the regions into independent working areas. The separation is made with usage of contour detection algorithm proposed by Suzuki [19]. For each component we calculate the area to be inpainted and based on it choose appropriate inpainting method. If the area is smaller than a predefined T threshold, we use modification of frequency selective extrapolation for inpainting that region otherwise we use exemplar-based inpainting method for inpainting of large regions. Fig. 3 illustrates the binary mask, which indicates the targeted regions and result of contour analyses procedure. In result of binary mask analyses procedure we obtain 2 lists of targeted areas: small regions and large regions.


a) b)
Figure 3. Separation of targeted regions into working areas: a) Input binary mask b) detected contours

2.3 Inpainting of small regions

This section presents the procedure of inpainting of small regions. For inpainting of each small region we use fast and accurate a modified method of frequency selective extrapolation algorithm [11-15]. The algorithm performs sparse modelling in frequency domain, with help of Discrete Fourier Transform (DFT) basis function [20].

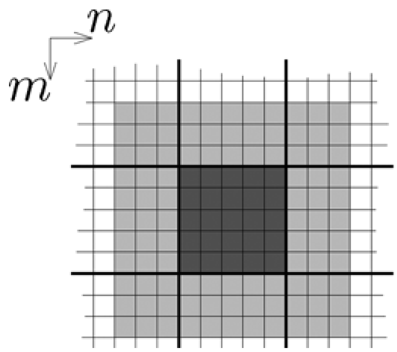


Figure 4: Part of digital image with missing area

Assume that the small region to be inpainted is Ω', we divide the targeted area into components of fixed size. Then for each component choose support area, and obtain $f[m,n]$ input signal which is composed of unwanted region form Ω' and support area. Fig. 1 schematically shows example of area α, is composed of dark grey area B (part of targeted/corrupted region), which has to be estimated with usage of elements in light grey area A (support area).

For estimation of corrupted regions in $f[m,n]$, the support area is approximated by linear combination of two-dimensional DFT basis functions. The approximation is dome with usage of $g[m,n]$ parametric model, which is defined over area α.

The key steps of the presented frequency domain based small regions inpainting algorithm are:

1. Input: Image I with Ω' small targeted region to be inpainted
2. Divide the Ω' small targeted area into fixed size components.
3. For each component define $f[m,n]$ function over α area which is composed on unknown region and support area.
4. Input $f[m,n]$, $g^{(0)}[m,n]=0, \quad \forall(m,n)\in\alpha,\ r^{(0)}[m,n]=f[m,n]w[m,n], \quad \forall(m,n)\in\alpha.$
5. Compute the DFT of the $g^{0}[m,n]$, $f[m,n]$ and $r^{0}[m,n]$:
   $F[m,n]=DFT\{f[m,n]\},\ G^{(v=0)}[m,n]=DFT\{g^{(v=0)}[m,n]\},\ R^{(v=0)}[m,n]=DFT\{r^{(v=0)}[m,n]\}$
6. Calculate of the difference of error criterion $\Delta E_A^{(v+1)}$ between v and $(v+1)$ iterations.
7. Select the basis function indexes (u,v) value pair, for which $\Delta E_A^{(v+1)} \quad (u,v)=\arg\max \Delta E_A^{(v+1)}$.
8. Update of $c_{k,l}^{(v+1)}$ expansion coefficient update.
9. Repeat steps 4 – 8 until the $\Delta E_B^{(v+1)}$ drops predefined threshold $E_{\min}=15$ or the number of iteration becomes more than max number of iterations. We used max number of iterations 100.
10. Apply invers discrete Fourier transform on parametric model:
    $g[m,n]=IDFT\{G^{(v)}[m,n]\}$
11. replace the element from $f[m,n]$ which contain specular reflections with corresponding elements from $g[m,n]$: $f[m,n]=g[m,n],$ where $[m,n]\in A.$
12. Repeat steps 3 – 11 until the whole Ω' region will be inpainted
13. Output: Image I with inpainted region Ω'

2.4 Inpainting of large regions

Selective Extrapolation performs well on small targeted regions, for large regions we use exemplar-based inpainting [5], which performs well on large structure and texture regions. The exemplar-based inpainting the targeted regions are filled with samples taken from known regions of the image. Algorithm search similar samples in source region and copy them into targeted unknown regions as it is done in texture synthesis [5-8].

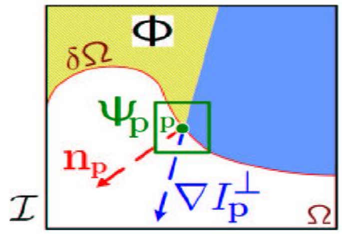


Figure 5: Notation diagram taken from [5]. Input image I, which includes targeted region Ω ; Ψ_p is the given patch centered at point p of the $\partial\Omega$ fill front; n_p is the normal to contour $\partial\Omega$ at the point p ; $\nabla I_p^\perp$ is isophote at the point p.

Assume we have an input image I with targeted region Ω to be inpainted. In general the source region is $\Phi = I - \Omega$ (See Figure 5), but it also can be manually specified by the user. The exemplar-based inpainting [5] implement the following steps until the all pixels in targeted region being inpainted:

1. **Computation of the patch priorities**. In exemplar-based inpainting algorithm the filling order depends on priority values which are assigned to each patch of the fill front. Given a patch Ψ_p centered at point p of the $\partial\Omega$ fill front, priority $P(p) = D(p)C(p)$ is computed via multiplication of two terms: a *data* term $D(p)$, which is a function of strength of isophotes crossing the fill front at each iteration; the second term is the *confidence* term $C(p)$, which indicates the number of reliable information surrounding the pixel p. Thus, the patches to be inpainted are being chosen based on these two terms, the patches which are on the continuation of strong edges and which have high confidence adjacent pixels are being inpainted first.

$$C(p) = \frac{\sum_{q \in \Psi_p \cap \overline{\Omega}} C(q)}{\left|\Psi_p\right|}, \qquad D(p) = \frac{\left|\nabla I_p^\perp n_p\right|}{\alpha},$$

   where $\left|\Psi_p\right|$ is the area of Ψ_p, α -is the normalization factor (e. g. α =255 for grey level image), n_p is the unit vector orthogonal to the fill front $\partial\Omega$ in the point p .

2. **Structure and texture information propagation.** One the $\Psi_{\hat{p}}$ patch with highest priority is being found, it is being filled with data extracted from the most similar $\Psi_{\hat{q}}$ region from Φ source region. The computation of $\Psi_{\hat{q}}$ shown below:

$$\Psi_{\hat{q}} = \underset{\Psi_{\hat{q}} \in \Phi}{\arg\min}\, d(\Psi_{\hat{p}}, \Psi_q),$$

   where the distance $d(\Psi_{\hat{p}}, \Psi_q)$ is the sum of squared differences of known or already filled pixels.
   Φ support area is selected automatically for each $\Psi_{\hat{p}}$ patch.

3. **Update confidence values.** The next step after filling the $\Psi_{\hat{p}}$ patch, the $C(p)$ confidence is updated in area of $\Psi_{\hat{p}}$ which has been filled as follows:

$$C(p) = C(\hat{p}), \quad \forall q \in \Psi_{\hat{p}} \cap \Omega.$$

3. COMPUTER SIMULATION AND COMPARISON

In this section, we test the proposed algorithm on natural and synthetic images. We apply the presented algorithm to the applications of scratch removal, object removal and specular reflections removal. Experiments were conducted on Samsung Galaxy s4 mobile device, which operates in open source Android OS. The mobile device is power by octa core Processor with 1,6 GHz Quad + 1,2 GHz Quad CPU Speed. The proposed algorithm is implemented on C++ and the user interface is developed on Java. The input image can be taken form mobile device camera or from remote and local file systems. We apply our algorithm to a variety of images, ranging from purely synthetic images to full-color photographs that include complex textures.

In section 2 to we have found out that the proposed algorithm is composed of 2 types of inpainting techniques: small regions fill and large regions fill. We compare the algorithm with algorithms targeted to small regions fill such as Telea [16] and Navier-Stokes[17], included in openCV open source library[21] and algorithm for large region fill such as Criminisi [5].

Figure 6 demonstrates the comparison of proposed method for small regions fill with Telea and Navier-Stokes inpainting methods. We can see that proposed method provides better PSNR value. Figure 7, 8 demonstrate the usage of algorithm for large object removal from color real life images, where the proposed method is compared with local inapinting method described in [17]. Figure 9 demonstrates the result of the proposed inpainting algorithm on endoscopic image, which is compared with global inpainting method described in [5]. Figure 9 demonstrates the result of the proposed inpainting algorithm on synthetic image, with local inapinting method described in [16].

Figure 6. a) Original image; b) Corrupted image; c) Inpainting via Navier-Stokes method, PSNR: 23.36dB; d) Inpainting via Telea method, PSNR: 23.19 dB; e) Inpainting via proposed method, PSNR: 26.43 dB.

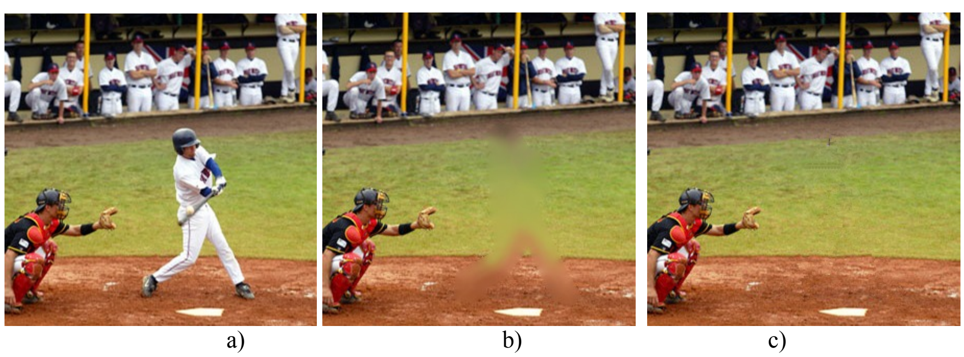


Figure 7. a) Original image c) Inpainted with method proposed in [17]; b) Inpainted image via proposed method.

a) b) c)

Figure 8. a) Original image; c) Inpainted with method proposed in [17]; b) Inpainted image via proposed method.

a) b) c)

Figure 9. a) Original endsoscopic image with specular reflection; b) Inpainted specular reflections with method proposed in [5]; Inpainted specular reflections with proposed method.

a) b) c)

Figure 10. a) Synthetic image; b) Inpainted image with method proposed in [16]; d) Inpainted image with proposed method.

4. CONCLUSION

This paper proposed a novel and efficient inpainting scheme that combines the advantages of both local and global methods into a single algorithm. Particularly, it introduces a blind inpainting model to solve the above problems by adaptively selecting support area for the inpainting scheme. The proposed method is applied to various challenging image restoration tasks, including recovering old photos, recovering missing data on real and synthetic images, and recovering the specular reflections in endoscopic images. A number of computer simulations demonstrate the effectiveness of our scheme and also illustrate the main properties and implementation steps of the presented algorithm. Furthermore, the simulation results show that the presented method is among the state-of-the-art and compares favorably against many available methods in the wireless environment. Robustness in the wireless environment with respect to the shape of the manually selected “marked” region is also illustrated. This work could also be simply extended to video inpainting. Several interesting perspectives of this work are under investigation. Recent researches have given much attention to Three- Dimensional sensing. Depth Image Based Rendering (DIBR) technique has been recognized as a capable tool for supporting innovative 3D image/video systems. On the other hand, an essential problem with DIBR is to fill newly exposed holes caused by disocclusions. Currently, we are working on the expansion of this work to 3D image/video inpaiting. In the future, we will further investigate in incorporating the human-labeled structures into our framework in order to recover the completely removed structures.

REFERENCES

[1] Zinovi Tauber, “Disocclusion and Photorealistic Object Reconstruction and Reprojection", Ph. D. Thesis, Simon Fraser University (2004).

[2] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image Inpainting", Proceedings of SIGGRAPH 2000, New Orleans, USA, pp 417-424 (2000).

[3] K. Pulli, A. Baksheev, K. Kornyakov, and V. Eruhimov, “Real-time computer vision with OpenCV,” Communications of the ACM, vol. 55, no. 6, pp. 61–69 (2012).

[4] X. Yang and K.-T. Cheng, “LDB: An ultra-fast feature for scalable Augmented Reality on mobile devices,” in IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 49–57 (2012).

[5] A. Criminisi, P. Perez, and K. Toyama, “Object removal by exemplar-based inpainting,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 721–728 vol.2 (2003).

[6] Y. Xu, G. Guo, and H.-Y. Shum, “Chaos mosaic: Fast and memory efficient texture synthesis,” Tech. Rep., Microsoft Research (2000)

[7] M Ashikhmin, “Synthesizing natural textures,” in ACM Symposium on Interactive 3D Graphics, 2001.

[8] M. Zontak and M. Irani, “Internal statistics of a single natural image,” in IEEE Conference on Computer Vision and Pattern Recognition(2011)

[9] Somayeh Bakhtiari , Elmira Mohyedinbonab, Sos Agaian, Mo Jamshid, “Image noise removal using image inpainting”, Proc. SPIE 8295, Image Processing: Algorithms and Systems X and Parallel Processing for Imaging Applications II, 82950N (2012).

[10] Somayeh Bakhtiari , Sos S. Agaian, Mo Jamshidi,” Local fingerprint image reconstruction based on gabor filtering”, Proc. SPIE 8406, Mobile Multimedia/Image Processing, Security, and Applications 2012, 840602 (2012).

[11] Seiler, J., Kaup, A., “A Fast Algorithm for Selective Signal Extrapolation with Arbitrary Basis Functions”, Conference on Signals, Systems and Computers, pp. 1256 –1260 (2009).

[12] G.Karapetyan, H.Sarukhanyan. “Concealment of Targeted Regions in Digital Images on Mobile Devices, Mathematical Problems of Computer Science”, vol. 40, 7p. (2013).

[13] G.Karapetyan, H.Sarukhanyan, “On a modification of the frequency selective extrapolation method“, Information Models and Analyses, vol.2, pp.139-145 (2012).

[14] G.Karapetyan, “Modification of FSE method based on coefficients of homogeneity”, Mathematical Problems of Computer Science , vol. 35, pp. 109-115 (2011).

[15] G.Karapetyan, H.Sarukhanyan. “Automatic Detection and Concealment of Specular Reflections for Endoscopic Images” Ninth International Conference Computer Science and Information Technologies, Revised Selected Papers, IEEE Explore, 8 pages (2013).

[16] A. Telea, “An image inpainting technique based on the fast marching method”, Journal of graphics tools 9.1 pp. 23-34 (2004).

[17] M. Ebrahimi, M. Holst, “The Navier-Stokes-Voight Model for Image Inpainting” arXiv:0901.4548 (2009)

[18] T. Farouki, “The Bernstein polynomial basis: A centennial retrospective”, Computer Aided Geometric Design, vol. 29, Issue 6, pp. 379 – 419 (2012).

[19] Suzuki, S. and Abe, K., “Topological Structural Analysis of Digitized Binary Images by Border Following”, CVGIP 30 1, pp 32-46 (1985).

[20] S.Agaian, H.Sarukhanyan,K.Egiazarian, J.Astola, “Hadamard Transforms”, SPIE Press, 520p (2011).

[21] Open Source Computer Vision Library home page , Online. Available: http://www.opencv.org