Digital Image Inpainting using Finite Volume approach and the Navier-Stokes Equations

B. A. Youssef
Computer Based Engineering Application Department. Informatics Institute
Moubark City for Scientific research and Technological Applications
Alexandria
EGYPT

E.H. Atta
Computer Graphics Department. Informatics Institute
Moubark City for Scientific research and Technological Applications
Alexandria
EGYPT

Abstract:- This paper presents a new numerical method for image inpainting based on the vorticity stream function formulation of the Navier-Stokes Equations. The method employs a finite volume discretization and a Runge-Kutta time stepping scheme with dissipative terms. A stability analysis of the proposed numerical method has been performed to determine the maximum allowable time step for which the computation will be stable. It is shown that the expression for the maximum time step contains terms that are functions of the image isophote gradient and curvature. The method has been implemented for a number of grey level and colored images and the results indicate that the method is effective in terms of solution quality and computational efficiency.

Kew-word :- Image inpainting , Finite Volume, Navier-Stokes, stability ;

1 Introduction

Inpainting is the problem of filling-in missing regions in a digital image. The problem can be stated as: given an image with missing regions, fill in the missing regions or modify the damaged ones in a non-detectable way [1]. Considerable progress has been made in solving the inpainting problem using different techniques ranging from variational approaches [2], convolution methods[3], partial differential equations [1,4,5], and texture synthesis [6,7,8].

Recently, Bertalmio, et. al. [4] proposed a novel approach based on the analogy between the image inpainting equations and the Navier-Stokes equations of fluid dynamics. In this approach, the stream function plays the role of image intensity, the fluid velocity represents image intensity gradient, and fluid vorticity acts as the Laplacian of the image intensity. Using the stream function vorticity formulation of the Navier-Stokes, the equations are solved numerically in the region to be inpainted. The numerical algorithm is implemented iteratively in two steps, it starts by solving for the vorticity using a simple forward Euler time stepping, with centered differences in space for the diffusion and a minmod method [9] for the convection term. After one time step, the image intensity is computed by solving a Poisson equation using the Jacobi iteration method. The process is then repeated until a steady state is reached. Stability of the numerical method, however, is not addressed, and the choice of a suitable time step seems to be determined by numerical experimentation. In the present paper, a different numerical approach is proposed, using a modified vorticity transport equations, finite volume discretization, and a Runge-Kutta time stepping scheme with dissipative terms. The following sections describe the proposed method and present results of inpainting several digital images.

2 Mathematical Model

Following the approach presented in [4], the inpainting equations to be solved are represented as a vorticity transport equation as follows:

$$\frac{\partial \omega}{\partial t} + \nabla^{\perp} I \bullet \nabla\, \omega = \nu\, \Delta\omega \qquad (1)$$

$$\Delta I = \omega \qquad (2)$$

Where I is the isophote intensity, ω is the Laplacian of the image intensity, and ν is a diffusion parameter.

In the present approach the Poisson equation (2) is replaced by:

$$\frac{\partial I}{\partial t} + \nabla^{\perp} I \bullet \nabla\, I = \alpha(\Delta I + \omega) \qquad (3)$$

Where α alpha is a relaxation parameter, $\nabla^{\perp} I, \nabla I$ and are the isophote gradient and normal respectively. At steady state Eq. (3) is equivalent to Eq. (2), since $\nabla^{\perp} I \bullet \nabla\, I = 0$ and $\frac{\partial I}{\partial t}$ vanishes. Then Eqs (1, 3) can be written in vector form as follow:

$$\frac{\partial W}{\partial t} + \frac{\partial F_c}{\partial x} - \frac{\partial G_c}{\partial y} = \eta\left(\frac{\partial 2W}{\partial x^2} + \frac{\partial 2W}{\partial y^2}\right) + \zeta W \qquad (4)$$

Where

$$W = \begin{pmatrix} \omega \\ I \end{pmatrix}, \quad F_c = \begin{pmatrix} I_y \omega \\ I_y I \end{pmatrix}, \quad G_c = \begin{pmatrix} I_x \omega \\ I_x I \end{pmatrix}$$

$$\eta = \begin{pmatrix} \nu & 0 \\ 0 & \alpha \end{pmatrix}, \quad \zeta = \begin{pmatrix} 0 & 0 \\ \alpha & 0 \end{pmatrix}$$

The left hand side of equation (4) represents unsteady and convective terms, while the right hand side contains diffusion and source terms. Eq.(4) can be also put in the linearized form

$$\frac{\partial W}{\partial t} + A\frac{\partial W}{\partial x} - B\frac{\partial W}{\partial y} = \eta\left(\frac{\partial 2W}{\partial x^2} + \frac{\partial 2W}{\partial y^2}\right) + \zeta W \qquad (5)$$

Where

$$A = \frac{\partial F}{\partial W}, \quad B = \frac{\partial G}{\partial W}$$

3 Space and time Integration

Integration of Eq.(5) is implemented using a finite volume approach. The discretization procedure follows the method of lines in decoupling the approximation of the spatial and temporal terms [11]. The computational domain is divided into quadrilateral cells representing the pixels to be inpainted (Fig.1). The integration over physical control surface area takes the form:

$$\iint_{\Delta S} \frac{\partial W}{\partial t} dS + R(W) = 0 \qquad (6)$$

Where R(W) is the discretized residual of spatial derivative and can be written as:

$$R(W) = \oint_C \left(F_c\, dx + G_c\, dy\right) - A\oint_C \left(F_v\, dx + G_v\, dy\right) - \iint_{\Delta S} BW\, dS \qquad (7)$$

$$F_v = \frac{\partial W}{\partial x}, \quad G_v = \frac{\partial W}{\partial y}$$

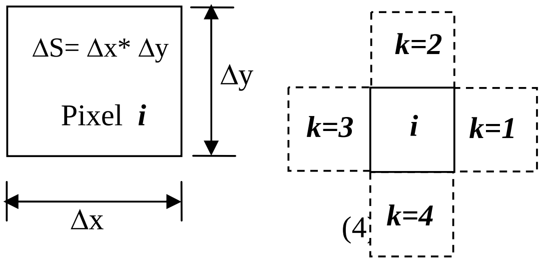


Fig1. The pixel cell definition and the discretization area of computational domain (i and k indices refer to pixel i and its neighbors)

The different terms are evaluated as follows:

$$\oint_C F_c \cdot dx = 0.5\sum_k^4 I_y (W_i + W_k)\ \Delta x_{i,k} \quad ,$$

$$\oint_C G_c \cdot dy = 0.5\sum_k^4 I_x (W_i + W_k)\, \Delta y_{i,k} \qquad (8)$$

$$\oint_C F_v \cdot dx = 0.5\sum_k^4 (F_{v\,i} + F_{v\,k})\ \Delta x_{i,k} \quad ,$$

$$\oint_C G_v \cdot dy = 0.5\sum_k^4 (G_{v\,i} + G_{v\,k})\ \Delta y_{i,k} \qquad (9)$$

Artificial dissipative terms are added to the original form of the Image Inpainting equations in order to eliminate spurious oscillation and accelerate the solution convergence. A fourth order linear dissipation (Dw) is inserted to Eq.(6) as follows:

$$\frac{\partial W}{\partial t} \Delta V + R(W) + Dw = 0 \qquad (10)$$

The differential form of the fourth order dissipation yields:

$$Dw = \frac{\partial}{\partial x}(\varepsilon_x x^3 \frac{\partial^3 W}{\partial x^3}) + \frac{\partial}{\partial y}(\varepsilon_y y^3 \frac{\partial^3 W}{\partial y^3}) \qquad (11)$$

There are two options for the values of $(\varepsilon_x, \varepsilon_y)$, either taken as constant value of order unity, or evaluated from the eigenvalues of the matrices in Eq. (4).

The time integration of Eq.(6) is preformed by using a fourth order Runge-Kutta method as follows:

$$W_1 = W^n \qquad (12\text{-a})$$

$$W_2 = W^n + \frac{\Delta t}{2\Delta V} R(W_1) \qquad (12\text{-b})$$

$$W_3 = W^n + \frac{\Delta t}{2\Delta V} R(W_2) \qquad (12\text{-c})$$

$$W_4 = W^n + \frac{\Delta t}{\Delta V} R(W_3) \qquad (12\text{-d})$$

$$W^{n+1} = W^n + \frac{\Delta t}{6\Delta V}(R(W_1) + 2R(W_2) + 2R(W_3) + R(W_4)) \qquad (12\text{-e})$$

4 Time Step and Stability Analysis

A stability analysis of the proposed numerical method has been performed to determine the maximum allowable time step for which the computation will be stable. To simplify the analysis, the maximum time step for the convective (hyperbolic) and the diffusive (elliptic) terms in Eq.(6) are determined separately and then combined to arrive at the maximum allowable time step. Using the Von Neumann stability analysis method[10], the amplification factor (Z) for the hyperbolic part yields

$$Z = J\,\Delta t_c \left(\frac{A Sin(\theta)}{\Delta x} - \frac{B\, Sin(\varphi)}{\Delta y}\right) \qquad (13)$$

Then

$$\Delta t_c = \frac{Z}{Max_{\theta,\varphi}(\lambda_A\, Sin(\theta)\,\Delta y - \lambda_B\, Sin(\varphi)\,\Delta x)} \qquad (14)$$

The eigenvalues λ_A, λ_B will be obtained as the following

Since $A = \frac{\partial F}{\partial W}$

then $$A = \begin{bmatrix} \frac{\partial(I_y \omega)}{\partial \omega} & \frac{\partial(I_y \omega)}{\partial I} \\ \frac{\partial(I_y I)}{\partial \omega} & \frac{\partial(I_y I)}{\partial I} \end{bmatrix}$$

$$A = \begin{bmatrix} I_y + \omega \frac{\partial I_y}{\partial \omega} & \omega \frac{\partial I_y}{\partial I} + I_y \frac{\partial \omega}{\partial I} \\ I \frac{\partial I_y}{\partial \omega} + I_y \frac{\partial I}{\partial \omega} & I_y + I \frac{\partial I_y}{\partial I} \end{bmatrix}$$

The eigenvalues of matrix A are extracted from the equation

$$\left(\lambda_A - I_y - \omega \frac{\partial I_y}{\partial \omega}\right)\left(\lambda_A - I_y - I \frac{\partial I_y}{\partial I}\right) - \left(\omega \frac{\partial I_y}{\partial I} + I_y \frac{\partial \omega}{\partial I}\right)\left(I \frac{\partial I_y}{\partial \omega} + I_y \frac{\partial I}{\partial \omega}\right) = 0 \qquad (15)$$

$$\lambda_A \left(\lambda_A - 2I_y - \omega \frac{\partial I_y}{\partial \omega} - I \frac{\partial I_y}{\partial I}\right) = 0 \qquad (16)$$

Then $\lambda_A = 0$ or $\lambda_A = 2I_y + \omega \frac{I_{yy}}{\omega_y} + I \frac{I_{yy}}{I_y}$

Similarly, the eigenvalues of matrix B :

$$\lambda_B = 0, \quad \lambda_B = 2I_x + \omega \frac{I_{xx}}{\omega_x} + I \frac{I_{xx}}{I_x} \qquad (18)$$

Where $J = \sqrt{-1}$, Δx and Δy are the pixel width and height respectively and are equal to unity.

Since the fourth order Rung Kutta is used then the stability condition is realized on the imaginary axis of the Rung Kutta stability criteria. The stability criteria [ ] realizes a maximum value of Z equal to $2\sqrt{2}$. The maximum value of Δt_c is achieved when θ, is equal to $\frac{\pi}{2}$ and φ is equal to $\frac{3\pi}{2}$, then Δt_c yields:

$$\Delta t_c = \frac{2\sqrt{2}}{2(I_y + I_x) + \omega(\frac{I_{yy}}{\omega_y} + \frac{I_{xx}}{\omega_x}) + I(\frac{I_{yy}}{I_y} + \frac{I_{xx}}{I_x})} \tag{19}$$

The values of $I, I_y, I_x, I_{yy}, I_{xx}$, $\omega, \omega_x, \omega_y$ are taken as absolute values

Performing a similar analysis to the elliptic (diffusive) part of Eq.(6), one gets:

$$\frac{\partial W}{\partial t} = \eta\left(\frac{\partial 2W}{\partial x^2} + \frac{\partial 2W}{\partial y^2}\right) + \zeta W \tag{20}$$

$$\Delta t_v = \frac{1}{4\lambda_{\eta \max} + \lambda_{\zeta \max}} \tag{21}$$

The combined time step is then evaluated as:

$$\Delta t = \frac{\Delta t_v \Delta t_c}{\Delta t_v + \Delta t_c} \tag{22}$$

5 Boundary conditions

To complete the implementation of the numerical solution, boundary conditions need to be specified on the boundary of the inpainted region. The required conditions can be obtained or computed from outside the inpainted region as follows:

1. Compute $I_x, -I_y$ using one-sided three-point finite difference formulae for the first derivative of the intensity variable I
2. Compute I_{xx}, I_{yy} using one-sided three-point finite difference formulae for the first derivative of intensity derivative in the x-direction I_x and in the y-direction I_y
3. Compute ω from the poisson equation $\omega = I_{xx} + I_{yy}$
4. Compute ω_x, ω_y using the one-sided three-point finite difference formulae for the first derivative of vorticity variable ω.

6 Results

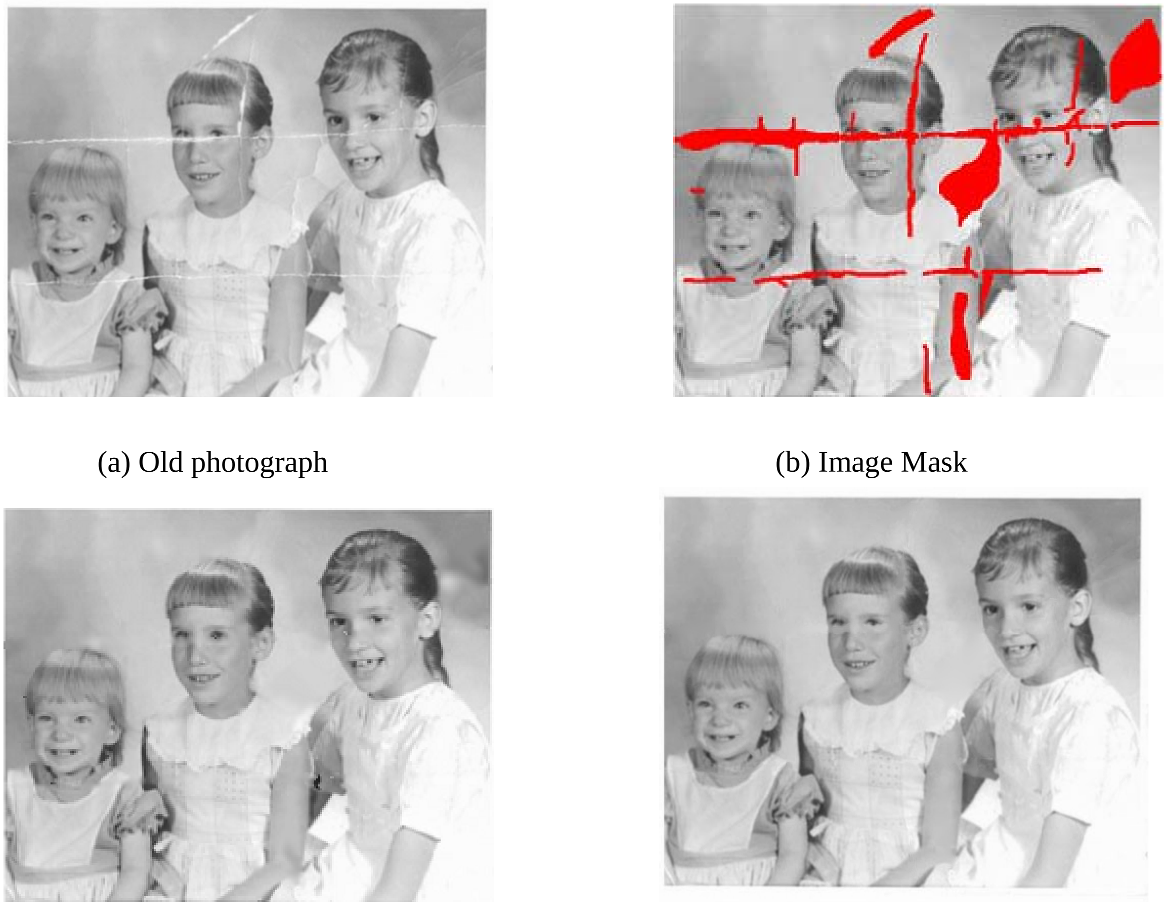
The developed method has been tested on grey and colored images. The inpainting algorithm is implemented using Compaq visual FORTRAN on a Pentium IV PC (265Mb RAM, 2GHz) under windows platform. All the examples use images available from public databases over the Internet. and foregoing articles. The CPU time required for inpainting depends on the size of the inpainted region. The cpu time for each inpainted pixel is roughly equal to 8E-06 second per iteration for each color channel (red, green, and blue) The number of iterations ranges from 500 to 5000. In all examples presented, the inpainting process is completed in less than 2 minutes.

Figure 2a shows a deteriorated grey level image that has been used by several researchers. The image mask shown in Figure 2b is either supplied or drawn manually using a graphical user interface developed as part of the developed method. The restored photograph is shown in Figure 2c is very satisfactory and similar to the result of reference (2).

Two other examples are displayed in Figures 3 and 4 showing cases of text and scratch removal from high resolution colored images. These results indicate the effectiveness of the developed method in inpainting grey level and colored images.

Convergence performance of the numerical method is displayed in Figure 5 for a representative inpainting case. It shows that the present model realizes a good convergence rate whereas the residual drops five orders of magnitude in approximately in 600 iterations

The same trend is achieved for all the cases presented in the paper.



(a) Old photograph

(b) Image Mask

(c) Restored photograph

(d) Restored result of Ref. (1)

Figure2: Inpainting of old photograph

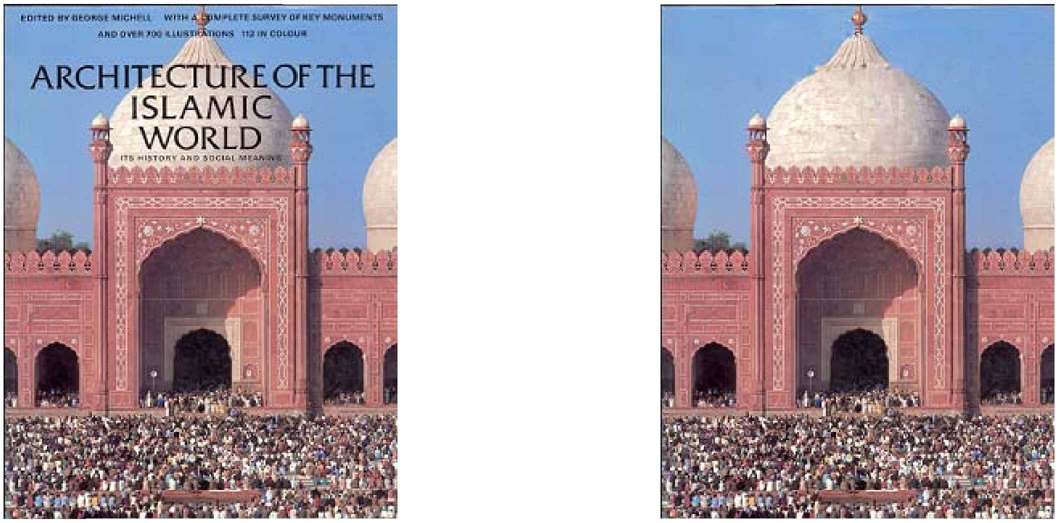

EDITED BY GEORGE MICHELL WITH A COMPLETE SURVEY OF KEY MONUMENTS
AND OVER 700 ILLUSTRATIONS 112 IN COLOUR
ARCHITECTURE OF THE
ISLAMIC
WORLD
ITS HISTORY AND SOCIAL MEANING

Figure 3: Removing undesired superimposed text from colored image.

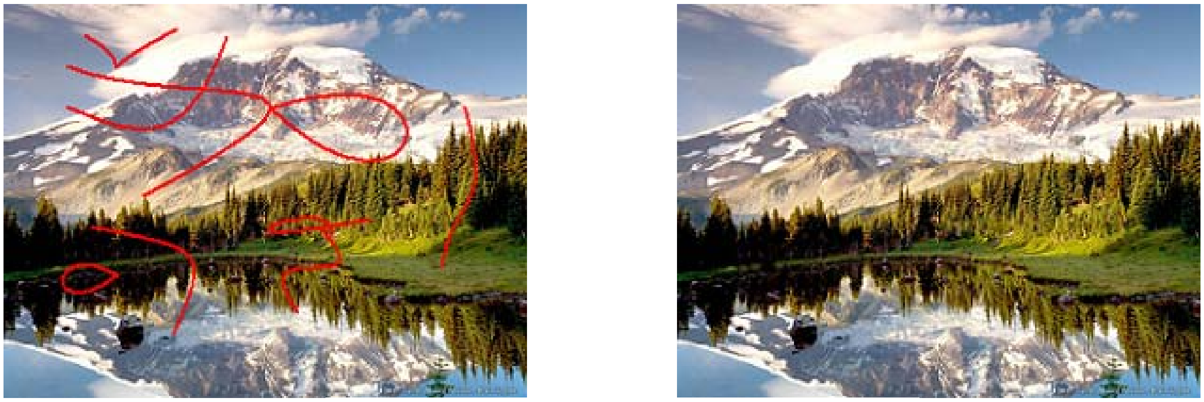


Figure 4. Scratch removal from colored image

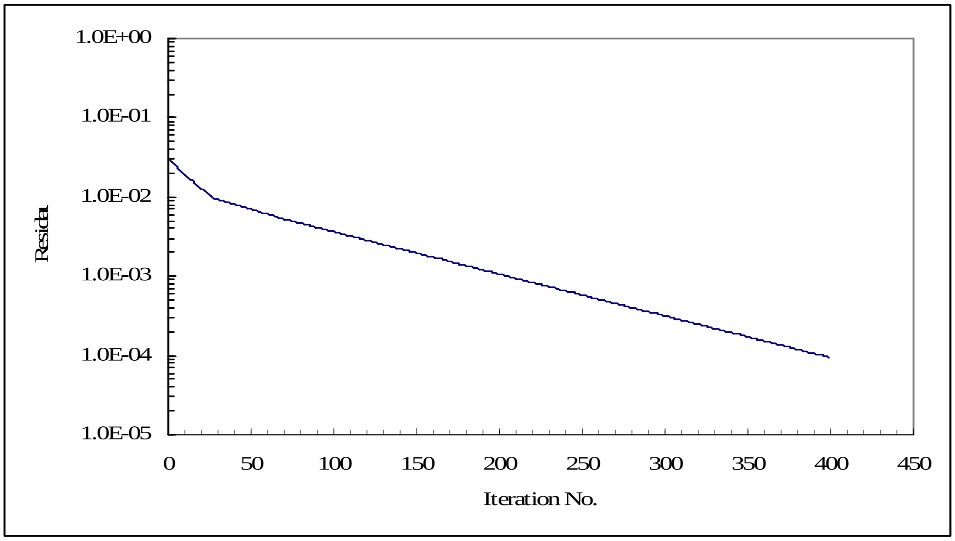


Figure 5: Convergence performance of the solution residual

.

7 Conclusions

In this paper we present a new numerical method for image inpainting based on the vorticity-stream function formulation of the Navier-Stokes Equations. The method employs a finite volume discretization and a Runge-Kutta time stepping scheme with dissipative terms. The finite volume approach permits the inpainting of arbitrary and non-rectangular regions. The modification of the Poisson equation of the isophote intensity enables the construction and implementation of efficient numerical method exhibiting fast convergence. The stability analysis of the proposed numerical method has been performed to determine the maximum allowable time step for which the computation will be stable. It is shown that the expression for the maximum time step contains terms that are functions of the image isophote gradient and curvature. The method has been implemented for a number of grey level and colored images and the results indicate that the method is effective in terms of solution quality and computational efficiency.

References:

[1] M. Bertalmio, G. Sapiro, C. Ballester and V. Caselles, Computer Graphics, SIGGRAPH 2000, pp. 417-424, July 2000.

[2] Chan, T., Shen, J. Non-Texture Inpainting by Curvature-Driven Diffusions (CCD) UCLA CAM TR 00-35, September 2000.

[3] M. Oliveira, B. Bowen, R. McKenna, and Y. -S. Chang Fast Digital Image Inpainting

In Proc. VIIP 2001, pp. 261.266, [CITY]: [PUB], 2001.

[4] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting" Proc. IEEE Computer Vision and Pattern Recognition (CVPR), 2001

[5] A.Telea. An Image Inpainting Technique based on the Fast Marching Method. Journal of Graphics Tools, vol. 9, no. 1, ACM Press, 2004

[6] Ashikhmin, M. Synthesizing Natural Textures, 2001 Symposium on Interactive 3D Graphics, pages 217-226.

[7] Wei, L., Levoy, M. Fast Texture Synthesis using Tree-structured Vector Quantization. SIGGRAPH 2000 pp 479-488.

[8] A. Efros and T. Leung, Texture synthesis by non-parametric sampling. In International Conference on Computer Vision, volume 2, pages 1033.8, Sep 1999.

[9] S. Osher and J. Sethian, *J. Comp. Phys.*, **79**, 12-49, 1988.

[10] C. Hirsch "Numerical Computational of Internal and External Flow", John Wiley, New York, 1990

[11] A. Jameson, W. Schmidt, and E. Turkel , Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes, AIAA Paper 81-1259, 1981.