Image Inpainting via Fluid Equations

Dalia Fishelov
AFEKA-Tel-Aviv Academic College of Engineering
and Tel-Aviv University
Tel-Aviv, Israel
Email: daliaf@afeka.ac.il

Nir Sochen
Department of Applied Mathematics
Tel-Aviv University
Tel-Aviv, Israel
Email: sochen@post.tau.ac.il

Abstract **An inpainting technique for images and videos is introduced. The idea is to use fluid equations – the Navier-Stokes equations- as a PDE based method for the image processing. The representation of the Navier-Stokes in terms the of streamfunction eases the implementation and the analysis of the method. We applied the new method to color images and show some numerical results.**

I. Introduction

Image inpainting is a process for filling in part of an image, using information from its surrounding area. We choose a PDE (Partial Differential Equation) based method, i.e., we pick up a differential equation which assumes the present distorted image as its initial data, and evolve the data in time so that the missing information will be filled in by data from the surroundings of the defected region; the latter are given as boundary conditions of the differential formulations. The boundary data include RGB (Red-Green-Blue) components of the image in spherical coordinate system and their normal derivatives. This data naturally dissipate into the regions in which RGB components of the color image are missing or corrupted, so that it fills in the gap in information by reconstructing the colored image inside this region.

II. The Partial Differential Equations MODEL

Following ideas represented in [4], the equations we have picked to model the evolution of the image are the Navier-Stokes equations with non-linear viscous term. The justification for choosing this model is as follows.
We denote by I the image intensity (grey level); the values of I range between 0 and 256.
The Laplacian of $I = \Delta I = \nabla^2 I$ denotes the smoothness of I. In the direction of the gradient of I, ∇I, one expects the largest variation of I. Therefore, in the direction which is orthogonal to ∇I, i.e., in the direction $\nabla^{\perp} I$, one expects the flow quantities to remain smooth. Hence, the smoothness of the flow, ΔI, is expected to remain unchanged. Thus, ΔI convects along the direction $\nabla^{\perp} I$.

This means that

$$\frac{\partial(\Delta I)}{\partial t} + \left(\nabla^{\perp} I\right)\cdot \nabla(\Delta I) = 0. \quad (0.1)$$

In order to stabilize the equation, we add some non-linear dissipation $\nu\Delta(g\Delta I)$, where ν is the viscosity coefficient and $g(\nabla I)$ is a non-linear function of ∇I. We get

$$\frac{\partial(\Delta I)}{\partial t} + \left(\nabla^{\perp} I\right)\cdot \nabla(\Delta I) = \nu\Delta(g\Delta I), (x,y)\in D, \quad (0.2)$$

where D is the region to be filled in, and ∂D is its boundary.

The equation above is just the Navier-Stokes equations, with a non-linear viscous term. We usually pick $g(\nabla I) = \dfrac{1}{1+\alpha|\nabla I|^2}$, where α is a parameter.

The identification with fluids is as follows. The grey level of image I is the streamfunction ψ, which is used in fluid flow.
Therefore, $\Delta I = \Delta\psi = \xi$, i.e., the vorticity, and $\nabla^{\perp} I = \nabla^{\perp}\psi = \underline{u}$ is the velocity vector.

We supplement (0.2) with boundary conditions

$$I(x,y) = f_1, \frac{\partial I}{\partial n}(x,y) = f_2, \ (x,y)\in\partial D \quad (0.3)$$

Equations (0.2)-(0.3) are supplemented with the initial condition

$$I(x,y,0) = I_0(x,y), \ (x,y)\in D, \quad (0.4)$$

where $I_0(x,y)$ is the initial original image to be inpainted.

1-4244-0859-8/06/$20.00 ©2006 IEEE

Note here that the originality of the method is that equations (0.2)-(0.3) contain the grey levels I only, whereas the formulation in [3] is

$$\frac{\partial \xi}{\partial t} + \left(\nabla^{\perp} I\right) \cdot \nabla \xi = \nu \nabla \cdot (g \nabla \xi), (x, y) \in D, \qquad (0.5)$$

with

$$\xi = \Delta I. \qquad (0.6)$$

Here, ξ is evolved in time via (0.5). The first boundary condition in (0.3), i.e.,

$I(x, y) = f_1$, $(x, y) \in \partial D$

is used to solve I in terms of ξ.

The second boundary condition in (0.3), i.e.,

$$\frac{\partial I}{\partial n}(x, y) = f_2, \ (x, y) \in \partial D$$

is used in [3] to construct a boundary condition for ξ. However, as pointed out in [5], there is no local boundary condition for ξ. Our method avoids the necessity to form such a boundary condition, as (0.2)-(0.3) is formulated in terms of I alone.

It was shown in [2] that the solution of equations (0.1)-(0.3) when $g(\nabla I)$ is constant is unique solution and that the solution decays exponentially to zero in case of homogenuous boundary conditions. The last statement hold in the case that $g(\nabla I)$ is not necessarily a constant but with the assumption that $g(\nabla I) > 0$.

We refer the reader to [7] for a review about various methods of inpainting.

III. Numerical Implemetation

The numerical scheme we use for the approximation of (0.2)-(0.3) is described in [1] and analyzed in [2]. The way we step equation (0.2) in time is the Crank-Nicholson scheme for the viscous term (on the right-hand-side of (0.2)), and an explicit modified-Euler scheme for the convective non-linear term in the left-hand-side of (0.2). The viscous term is approximated via a Stephenson [6] compact scheme. Grid values of both I and its gradient $\nabla I = (I_x, I_y)$ are used as unknowns. The first order derivatives I_x, I_y are related to I via the following equations:

$$\frac{1}{6}\left(I_x\right)_{(i-1,j)} + \frac{2}{3}\left(I_x\right)_{(i,j)} + \frac{1}{6}\left(I_x\right)_{(i+1,j)} = \frac{I_{(i+1,j)} - I_{(i-1,j)}}{2h}$$

$$\frac{1}{6}\left(I_y\right)_{(i,j-1)} + \frac{2}{3}\left(I_y\right)_{(i,j)} + \frac{1}{6}\left(I_y\right)_{(i,j+1)} = \frac{I_{(i,j+1)} - I_{(i,j-1)}}{2h}$$

where h is the size of the mesh.

The fourth-order derivative, which appears in the viscous term,
is approximated as follows:

$$\left(I_{xxxx}\right)_{(i,j)} \approx \frac{12}{h^2}\left[\left(\delta_x I_x\right)_{(i,j)} - \left(\delta_x^2 I\right)_{(i,j)}\right]$$

where $\left(\delta_x I_x\right)_{(i,j)} = \dfrac{\left(I_x\right)_{(i+1,j)} - \left(I_x\right)_{(i-1,j)}}{2h}$,

and $\left(\delta_x^2 I\right)_{(i,j)} = \dfrac{I_{(i+1,j)} - 2I_{(i,j)} + I_{(i-1,j)}}{h^2}$.

$\left(I_{yyyy}\right)_{(i,j)}$ is approximated in a similar way.

The approximations of the pure fourth-order derivatives are of fourth order (see [1] and [2]).

The mixed fourth order derivatives are approximated via a standard centered scheme

$$\left(I_{xxyy}\right)_{(i,j)} \approx \left(\delta_y^2 \delta_x^2 I\right)_{(i,j)},$$

Which is of second order (see [1],[2]), but may be improved to fourth-order.

The convective non-linear term

$$\left(\nabla^{\perp} I\right) \cdot \nabla(\Delta I) = -I_y \Delta I_x + I_x \Delta I_y$$

is approximated by a centered finite-difference scheme in the following way

$$-I_y \Delta I_x + I_x \Delta I_y \approx -I_y \Delta_h I_x + I_x \Delta_h I_y,$$

where $\Delta_h I_x = (\delta_x^2 + \delta_y^2)\, I_x$ and $\Delta_h I_y = (\delta_x^2 + \delta_y^2)\, I_y$.

It was proven in [2] that the scheme above converges to a solution of the Navier-Stokes equation in case that $g(\nabla I)$ is a constant. It was also shown in [1] that the scheme is stable in time under the assumption

$\Delta t / h \le \min(\sqrt{2}/3, \sqrt{8\nu\Delta t}/(3h)$, where Δt is the time step and h is the spatial mesh size.

IV. Color Images and Complex boundary treatment

We implement our methodology to color images. Color images are represented by a vector valued function (R, G, B) (Red, Green, Blue). Each component represent the intensity of each of the colors above at each point (x, y) of the image.

We use a spherical coordinate system

$$B = r\cos\theta \cdot \cos\phi,\ R = r\cos\theta \cdot \sin\phi,\ G = r\sin\theta.$$

The three components to inpaint are

$$I_1 = r,\ I_2 = \sin\theta,\ I_3 = \sin\phi. \qquad (0.7)$$

We write down equation (0.2)-(0.4) to each of the components in (0.7).

We picked $\nu = 2$ and

$g(\nabla I_j) = 1/(1 + \alpha |\nabla I_j|^2),\ j = 1, 2, 3$

with $\alpha = 0.01$ is a parameter.

The domain D, which is the region to be inpainted, may have complex boundary, i.e., the boundary is not a rectangle or any other simple shape. The way we treat the complex geometry D is by defining a flag:

$flag = 0 \; if \; |R| \le J_0 \; or \; |G| \le J_0 \; or \; |B| \le J_0$,

otherwise $flag = 1$.

Here J_0 is some threshold, which was picked up to be 200, as the inpainted region is colored originally with white, and therefore all the colors there take approximately the value of 256. We also assigned $flag = 1$ to 36 neighboring points satisfying the condition above in order to enlarge the inpainted region, so it contains also points near the original boundary of this region.

Therefore, if $flag = 1$ we are in a region D to be inpainted, and if $flag = 0$ this pixel of the image is remain unchanged.

We have picked $\nu = 2$ and $\Delta t = 10^{-3}$ in our numerical experiments. The equations were evolved in time for about 500 iterations.

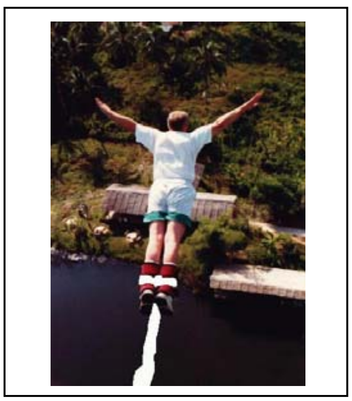

Figure 1. Before Inpainting

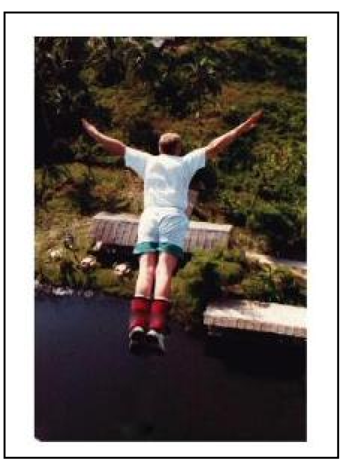

Figure 2.After Inpainting

Figure 3. Before Inpainting.

Figure 4. After Inpainting-the microphone is removed.

REFERENCES

[1] M. Ben-Artzi, J-P. Croisille, D. Fishelov and S. Trachtenberg, "A Pure Compact Scheme for the Streamfunction Formulation of Navier-Stokes Equations", J. Comp. Phys., Vol. 205, pp. 640-664, 2005.

[2] M. Ben-Artzi, J-P. Croisille, and D. Fishelov, "Convergence of a Compact scheme for the pure streamfunction formulation of the unsteady Navier-Stokes system", SIAM J. Numer. Anal., in press

[3] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting", *IEEE CVPR 2001, Hawaii, USA, December 2001.*

[4] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image Inpainting"*, Proceedings of SIGGRAPH 2000, New Orleans, USA, July 2000.*

[5] P. M. Gresho, "Incompressible fluid dynamics: some fundamental formulation issues," *Annu. Rev. Fluuid Mech., Vol.. 23, pp. 413-453, 1991.*

[6] J. W. Stephenson, "Single cell discretizations of order two and four for biharmonic problems," *J. Comp. Phys.,, Vol.. 55, pp. 65-80, 1984.*

[7] Zinovi Tauber, Ze-Nian Li and Mark S. Drew, "Review and Preview: Disocclusion by Inpainting for Image-based Rendering", IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews, In Press, 2006.