Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting

M. Bertalmío
Computer Eng. Dept.
University Pompeu Fabra
08003 Barcelona, SPAIN

A. L. Bertozzi
Mathematics Dept.
Duke University
Durham, NC 27708

G. Sapiro
Elec. & Comp. Eng. Dept.
Univ. of Minnesota
Minneapolis, MN 55455

Abstract

Image inpainting involves filling in part of an image or video using information from the surrounding area. Applications include the restoration of damaged photographs and movies and the removal of selected objects. In this paper, we introduce a class of automated methods for digital inpainting. The approach uses ideas from classical fluid dynamics to propagate isophote lines continuously from the exterior into the region to be inpainted. The main idea is to think of the image intensity as a 'stream function' for a two-dimensional incompressible flow. The Laplacian of the image intensity plays the role of the vorticity of the fluid; it is transported into the region to be inpainted by a vector field defined by the stream function. The resulting algorithm is designed to continue isophotes while matching gradient vectors at the boundary of the inpainting region. The method is directly based on the Navier-Stokes equations for fluid dynamics, which has the immediate advantage of well-developed theoretical and numerical results. This is a new approach for introducing ideas from computational fluid dynamics into problems in computer vision and image analysis.

1. Introduction

Image *inpainting* [2, 10, 20, 38] is the process of filling in missing data in a designated region of a still or video image. Applications range from removing objects from a scene to re-touching damaged paintings and photographs. The goal is to produce a revised image in which the inpainted region is seamlessly merged into the image in a way that is not detectable by a typical viewer. Traditionally, inpainting has been done by professional artists. For photography and film, inpainting is used to revert deterioration (e.g., cracks in photographs or scratches and dust spots in film), or to add or remove elements (e.g., removal of stamped date and red-eye from photographs, the infamous "airbrushing" of political enemies [20]). A current active area of research is to automate digital techniques for inpainting [2, 3, 16, 21, 22].

In this paper, we introduce a novel algorithm for digital inpainting of still images that attempts to replicate the basic techniques used by professional restorators. Our algorithm, motivated by a method proposed in [2], involves a direct solution of the Navier-Stokes equations for an incompressible fluid. The image intensity function plays the role of the stream function whose isophote lines define streamlines of the flow. After the user selects the regions to be restored, the algorithm automatically transports information into the inpainting region. The fill-in is done in such a way that isophote lines arriving at the region's boundaries are completed inside. The technique introduced here does not require the user to specify where the novel information comes from. This is done automatically (and in a fast way), thereby allowing for simultaneously fill-in of multiple regions containing completely different structures and surrounding backgrounds. In addition, no limitations are imposed on the topology of the region to be inpainted. The only user interaction required by the algorithm is to mark the regions to be inpainted. Since our inpainting algorithm is designed for both restoration of damaged photographs and for removal of undesired objects on the image, the regions to be inpainted must be marked by the user. Our method inherits a mathematical theory already developed for the fluid equations, including well-posedness and the design of efficient convergent numerical methods.

1.1. Prior Work

First note that image denoising is different to filling-in, since the regions of missing data are usually large. That is, regions occupied by top to bottom scratches along several film frames, long cracks in photographs, superimposed large fonts, and so on, are of significantly larger size than the type of noise treated by common image enhancement algorithms. In addition, in common image enhancement applications, the pixels contain both information about the real data and the noise (e.g., image plus noise for additive noise), while in our application there is no significant information

in the region to be inpainted.

A very active area related to our work is the restoration of damaged films. The basic idea is to use information from past and future frames to restore the current one, e.g., [16, 22]. Of course, this general approach cannot be used when dealing with still images. In addition, it can not deal with movies where the region to be inpainted is static with respect to its background (e.g., a logo on a shirt), since consecutive frames do not provide new information. This is also the case when the region to be inpainted occupies a large number of frames. Another area related to our work is texture synthesis, in which a texture is selected and synthesized inside the region to be filled-in (the hole) [9, 13, 15, 34]. These algorithms often require the user to select the texture and are not often well-designed to fill in structure from boundary data.

The closest methods to our approach are the fundamental works on disocclusion and line continuation. A pioneering contribution in this area is described in [28]. The authors presented a technique for removing occlusions with the goal of image segmentation. Since the region to be filled-in can be considered as occluding objects, removing occlusions is analogous to image inpainting. The basic idea suggested by the authors is to connect T-junctions at the occluding boundaries of objects with elastica minimizing curves. The technique was primarily developed for simple images obtained from a segmentation, with only a few objects with constant gray-levels. Thus, they ended up by connecting T-junctions at the same gray level. (Other researchers, e.g., D. Jacobs, R. Basri, and S. Zucker, have followed this interesting research area, mainly developing techniques for smooth curve continuation.)

Masnou and Morel [25, 26] recently extended these ideas, presenting a formal variational formulation for disocclusion and a particular practical implementation. The algorithm uses geodesic curves to join the isophotes arriving at the boundary of the region to be inpainted. The inpainting regions require a simple topology. In addition, the angle, with which the level lines arrive at the boundary of the holes, is not (well) preserved, and the algorithm uses straight lines to join equal gray value pixels.

1.2. Image inpainting along isophotes

Recently, the concept of smooth continuation of information in the level-lines direction has been addressed in [2]. This is the work, as we will see below, that we follow to make the connection with fluid dynamics and the Navier-Stokes equations. The proposed algorithm propagates the image Laplacian in the level-lines (isophotes) direction. The algorithm attempts to imitate basic approaches used by professional restorators. The algorithm also introduces the importance of propagating both the gradient direction (geometry) and gray-values (photometry) of the image in a band surrounding the hole to be filled-in. Some of the ideas of [2] where adopted in [1], while deviating from the particular model in order to be able to define a formal variational approach to the filling-in/inpainting problem.

The work in [2] inspired a very elegant approach to the filling-in problem recently reported in [7] (this work was performed independently of the one reported in [1]). The authors present a clear and intuitive axiomatic approach to the problem. The main algorithm they propose is to minimize the Total Variation (TV) [33], of the image inside the hole (they also use, as proposed in [1, 2] and here, a band surrounding the region). As in the work of Masnou and Morel, their interpolation is limited to creating straight isophotes, not necessarily smoothly continued from the hole boundary, and mainly is developed (as the authors clearly state) for small holes. Although straight connections give visually pleasant results for small holes, it is important to develop a theory that permits interpolation of level lines across large gaps, where connecting with straight lines will be unpleasant even for simple images. In order to obtain such a smooth interpolation and continuation of isophotes, it is necessary to go into high-order Partial Differential Equations (PDE's) or systems of PDE's, as done in [1, 2] and here (the authors of [7] have also recently introduced higher order models). Research in perception, from the Gestalt to more recent work (e.g. [31]) supports the idea of performing a smooth continuation of the angle of arrival of the level lines at the gap.

The paper [2] proposes an algorithm designed to project the gradient of the smoothness of the image intensity in the direction of the isophotes. The resulting scheme is a discrete approximation of the PDE

$$I_t = \nabla^\perp I \cdot \nabla \Delta I \tag{1}$$

for the image intensity I, where $\nabla^\perp$ denotes the perpendicular gradient $(-\partial_y, \partial_x)$ and Δ denotes the Laplace operator $\partial_x^2 + \partial_y^2$. Additional anisotropic diffusion of the image can produce a PDE of the form

$$I_t = \nabla^\perp I \cdot \nabla \Delta I + \nu \nabla \cdot (g(|\nabla I|)\nabla I). \tag{2}$$

The goal is to evolve (1) or perhaps more appropriately (2) to a steady state solution, satisfying the condition in the inpainting region that the isophote lines, in the direction of $\nabla^\perp I$, must be parallel to the level curves of the smoothness ΔI of the image intensity, which for $\nu = 0$ becomes

$$\nabla^\perp I \cdot \nabla \Delta I = 0. \tag{3}$$

We note that while the authors of [2] present their method as moving image intensity along isophote lines, there is another sense in which to think of the dynamics. Equation (1) is a transport equation that convects the image

intensity I along level curves of the smoothness, ΔI. This can be seen by noting that (1) is equivalent to $DI/Dt = 0$ where D/Dt is the material derivative $\partial/\partial t + v \cdot \nabla$ for the velocity field $v = \nabla^{\perp}\Delta I$. In particular I is convected by the velocity field v which is in the direction of level curves of the smoothness ΔI. In the next section we discuss how this is related to a classical problem in incompressible fluid dynamics. Our goal is to make use of expertise developed in that field to design better inpainting algorithms.

2. Analogy to Transport of Vorticity in Incompressible Fluids

Incompressible Newtonian fluids are governed by the Navier-Stokes equations, which couple the velocity vector field $\vec{v}$ to a scalar pressure p:

$$v_t + v \cdot \nabla v = -\nabla p + \nu \Delta v, \quad \nabla \cdot v = 0. \quad (4)$$

In two space dimensions, the divergence free velocity field v possesses a stream function Ψ satisfying $\nabla^{\perp}\Psi = v$. In addition, in 2D the vorticity, $\omega = \nabla \times v$, satisfies a very simple advection diffusion equation, which can be computed by taking the curl of the first equation in (4) and using some basic facts about the geometry in 2D:

$$\omega_t + v \cdot \nabla \omega = \nu \Delta \omega. \quad (5)$$

Note here that in 2D the vorticity is a scalar quantity that is related to the stream function through the Laplace operator, $\Delta \Psi = \omega$. In the absence of viscosity $\nu = 0$, we obtain the Euler equations for inviscid flow. Both the inviscid and viscous problems, with appropriate boundary conditions, are globally well-posed in two space dimensions. Solutions exist for any smooth initial condition and they depend continuously on the initial and boundary data [17, 18, 23, 24, 36, 40].

In terms of the stream function, equation (5) implies that steady state inviscid flows must satisfy

$$\nabla^{\perp}\Psi \cdot \nabla \Delta \Psi = 0 \quad (6)$$

which says that the Laplacian of the stream function, and hence the vorticity, must have the same level curves as the stream function. The analogy to image inpainting in the previous section is now clear: the stream function for inviscid fluids in 2D satisfies the same equation as the steady state image intensity equation (3).

The point is that in order to solve the inpainting problem proposed in the previous section, we have to find a steady state stream function for the inviscid fluid equations, which is a problem possessing a rich and well developed history.

2.1. The stream function-image intensity analogy

The main analogy that we build on in this paper is the parallel between the stream function in a 2D incompressible fluid and the role of image intensity function I in the inpainting method described in Section 1.2. This allows us to design a new inpainting method that will achieve the same steady equation (3).

Let Ω be a region in the plane in which we want to inpaint from surrounding data. Assume that the image intensity I_0 is a smooth function (with possibly large gradients) outside of Ω and we know both I_0 and ΔI_0 on the boundary $\partial\Omega$. We now design a 'Navier-Stokes' based method for image inpainting. In this method the fluid dynamic quantities have the following parallel to quantities in the inpainting method.

Navier-Stokes	Image inpainting
stream function Ψ	Image intensity I
fluid velocity $v = \nabla^{\perp}\Psi$	isophote direction $\nabla^{\perp}I$
vorticity $\omega = \Delta\Psi$	smoothness $w = \Delta I$
fluid viscosity ν	anisotropic diffusion ν

Our goal is to solve a form of the Navier-Stokes equations in the region to be inpainted. The method described next is based on the vorticity-stream form (5) of the Navier-Stokes equations, however it is also possible to consider other methods based on the primitive variables form (4).

For ease of notation we denote by w the smoothness ΔI of the image intensity. Instead of solving a transport equation for I as in (2), we solve a vorticity transport equation for w:

$$\partial w/\partial t + v \cdot \nabla w = \nu \nabla \cdot (g(|\nabla w|)\nabla w), \quad (7)$$

where the function g allows for anisotropic diffusion of the smoothness w. The image intensity I which defines the velocity field $v = \nabla^{\perp}I$ in (7) is recovered by solving simultaneously the Poisson problem

$$\Delta I = w, \quad I|_{\partial\Omega} = I_0. \quad (8)$$

For $g = 1$, the direct numerical solution of of (7-8) is a classical way to solve both the dynamic fluid equations and to evolve the dynamics towards a steady state solution [30].

For fluid problems with small viscosity ν, the above dynamics can take a long time to converge to steady state, making the method less practical. Instead there are *pseudo-steady* methods that involve replacing the Poisson equation (8) with a dynamic relaxation equation

$$I_t - \alpha[\Delta I + w] = 0, \quad \alpha > 0, \quad I_{\partial\Omega} = I_0. \quad (9)$$

where the parameter α determines a rate of relaxation. In our situation, diffusion can result in a blurring of sharp interfaces, gradients of I, in the inpainting region. Hence it

is often desirable to include anisotropic diffusion in the solution of I. This can be added directly to the dynamical problem (9) or as an additional step in conjunction with the Poisson step (8). In our test cases with anisotropic diffusion, we did not find a significant difference between direct solution of the Poisson equation in (8) versus the relaxation method (9).

Once steady state is achieved, we have effectively found a solution of (3) for the intensity I (perhaps modified slightly by the anisotropic diffusion). Hence we expect and indeed find that the Navier-Stokes inpainting method performs in many ways like the method proposed in [2]. There are several advantages to using the Navier-Stokes method. First, there is a well-developed theoretical and numerical literature for this problem on which we can build inpainting algorithms. Second there is the possibility to test the performance of the method against a number of classical examples from fluid dynamics. Finally, we are able to implement this method efficiently and we have a theoretical framework in which to understand the transport of information from the exterior into the inpainting region.

2.2. Isophote continuity and boundary conditions for Navier-Stokes

When using any PDE-based method to do inpainting, the issue of boundary conditions becomes very important. In order to produce a result which, to the eye, does not distinguish where the inpainting has taken place, we must at the very least continue both the image intensity and direction of the isophote lines continuously into the inpainting region. This means that any PDE-based method involving the image intensity I must enforce Dirichlet (fixed I) boundary conditions as well as a condition on the direction of ∇I on the boundary. Immediately we see that this poses a problem for lower-order PDE-based methods. Indeed, any first or second order PDE (including anisotropic diffusion) for the scalar I could typically only enforce one of these boundary conditions, the result being an inpainting with discontinuities in the slope of the isophote lines, or a method with a jump in I itself on the boundary. From a mathematical point of view, to fix this, one can either choose a higher order equation for I, as in [2], that requires more boundary conditions, or consider a vector evolution for ∇I, which is the idea of the Navier-Stokes method.

The Navier-Stokes analogy guarantees, in a very natural way, continuity of the image intensity function I and its isophote directions across the boundary of the inpainting region. First consider a solution of the Navier-Stokes equation (4) in primitive variables form satisfying the classical no-slip condition $v = 0$ on the boundary $\partial\Omega$. This condition guarantees two features: (a) that the stream function Ψ must be constant on the boundary, since the boundary is

trivially a streamline of the flow; (b) that the direction of the fluid velocity $\vec{v}$ is always tangent to the boundary. A more general form of the no-slip boundary condition, for which well-posedness is known, is to prescribe the velocity vector $v = \vec{v}_0$ on the boundary. This would be the natural choice for a moving boundary. Specifying the velocity on the boundary is equivalent to specifying both the normal and tangential derivatives of the stream function Ψ on the boundary, since $v = \nabla^\perp\Psi$. However, specifying the tangential derivative of Ψ determines Ψ on the boundary up to a constant of integration, by simply integrating around the boundary with respect to its arc length. Similarly this information determines the direction of flow on the boundary. The result is that if we solve the Navier-Stokes equations with v fixed on the boundary, we obtain a solution with a stream function Ψ and velocity field v both of which are continuous up to the boundary.

For the Navier-Stokes inpainting method, we inherit the continuity across the boundary. For example, suppose we fix $\nabla^\perp I$ on the boundary. Then solving the Navier-Stokes inpainting equation with these boundary conditions will not only result in continuous isophotes, but also will produce an image intensity function that is continuous across $\partial\Omega$.

In this paper, we are interested in solving the vorticity stream form (7) which, practically speaking, requires boundary conditions for the stream function and the vorticity. For viscous Navier-Stokes, there are several well-studied methods for doing this, all of which involve some sort of Dirichlet condition for ω on the boundary. In the case of a fluid, information about ω outside of the boundary is typically not known, hence first and second order accurate methods use information about the stream function at the previous timestep in order to numerically compute $\Delta\Psi$ on the boundary. In the case of images, we have more information about ΔI outside the boundary and can use this to construct accurate boundary conditions for $w = \Delta I$ at the boundary. For a discussion of how this is treated in fluids, see [30] Chapter 6.

2.3. Existence and uniqueness of solutions

The Navier-Stokes based inpainting method inherits a theoretical framework from the mathematical theory of the Navier-Stokes equations. Although a full treatment of such issues is beyond the scope of this paper, we discuss the problem of uniqueness and its relevance to inpainting.

First we note that without the presence of viscosity in the method we do not have a unique *steady-state* solution. This can be seen in the following simple example, motivated by problems involving the mathematical concentration of vorticity in solution sequences of the Euler equations [8, 12]. Consider the problem (6) with $v = 0$ on $\partial\Omega$. Consider a disk D_{2R} of radius $2R$, centered at the point x_0, contained

inside Ω. Let $w_1(r)$ and $w_2(r)$ be two *different* compactly supported functions on $[0, 1]$ so that $\int_0^1 w_i(r)r\,dr = 0$. From each w_i we can construct an exact "radial eddy" solution of the inviscid ($\nu = 0$) fluid equations: the vorticity ω_i is given as $\omega_i(x) = w_i(\frac{|x_0 - x|}{R})$, $|x - x_0| < R$, and zero otherwise. This results in a stream function satisfying (where we denote $r = |x - x_0|$)

$$\Psi_i'(r) = \frac{1}{r} \int_0^r s\omega_i(s)\,ds$$

for $0 < r < 2R$. Note that $\Psi_i'(r) = 0$ for $R < r < 2R$. We can integrate the equation for Ψ_i' to recover Ψ_i where the constant of integration is chosen so that Ψ_i is identically zero in $R < r < 2R$. Outside of the disk D_{2R} we continue Ψ_i and ω_i as zero. Note that this is a C^∞ continuation, by construction. Since the initial functions w_1 and w_2 were chosen to be different, but both satisfying the mean zero constraint, the result is two different smooth solutions of the inviscid steady state problem. In terms of the actual image, the result would be two different bull's eye patterns on a disk in the interior of Ω. Note that this construction works for any disk or for multiple disks. So it is possible to construct a wide range of different solutions satisfying the same zero velocity boundary data.

At the other extreme when viscosity is sufficiently large, for smooth boundaries there is a unique solution of the steady Navier-Stokes equations [11]. For viscous flows with moderate and small viscosity the problem is more complex; for example, there are classical experimental examples in which known steady state solutions have varying stability properties depending on the viscosity. For example, in the case of Couette flow, the fluid is constrained to move between two concentric cylinders rotating at different speeds. There is an exact, radially symmetric, solution that satisfies the Navier-Stokes equation for all viscosities. However this solution is unstable for sufficiently high viscosity, resulting in the creation of multiple eddies in such experiments (see e.g. [37] p. 80-81).

We expect that Navier-Stokes based inpainting may inherit some of the stability and uniqueness issues known for incompressible fluids, although the effect of anisotropic diffusion is not clear. In fact, some degree of non-uniqueness of the steady state problem could be favorable for inpainting, since large inpainting regions might have multiple 'possible' solutions, the best choice of which might be determined by pattern matching or information from a previous frame, as in the case of video. Ultimately we hope that the choice of magnitude and anisotropy of viscosity, as well as the initial condition in the inpainting region, may determine the best possible outcome for the inpainting method.

Existence and uniqueness of time-dependent solutions of the Navier-Stokes equations with isotropic viscosity ($g = 1$ in (7)) is well established [17, 18, 23, 24, 36, 40]. In our problem we consider the dynamics with anisotropic viscosity, both at the level of the smoothness and at the level of the intensity. This method of selective smoothing is known to be ill-posed for a wide class of functions g, however the ill-posedness can be easily removed with a small amount of smoothness applied to the gradient inside the function g. Catte et. al. [6] established an existence and uniqueness theory which could be extended to include nonlinear transport terms from the Navier-Stokes equations. In fact, nonlinear diffusion has a natural physical analogy; Non-Newtonian fluids have a viscosity that depends locally and nonlinearly on the shear ($|\nabla v|$) of the fluid [32]. Another form of viscosity for Navier-Stokes include fourth order hyperviscosity which is commonly used in numerical simulations of turbulence [4].

3. Computational Examples

We now show via some examples how the Navier-Stokes inpainting method performs. In each example described below, the Navier-Stokes equations are solved in the inpainting region. We start by computing the vorticity w from the image I, using information from the exterior to determine the boundary vorticity. We evolve the vorticity stream form (7), using a simple forward Euler time stepping, with centered differences in space for the diffusion and a min-mod method [29] for the convection term. The diffusion is anisotropic.

After one time step of (7) we compute the image intensity I by solving the Poisson equation (8) using the Jacobi iteration method. From this updated I we recompute w and start again. Every few steps we perform anisotropic diffusion on I, which helps to sharpen edges. Steady state is achieved after N iterations of this cycle, typically N=300. We can set the algorithm to stop automatically when I does not change appreciably. Parameters for the algorithm have been chosen in such a way as to work for a wide range of examples: $dt = 0.01$, $dx = dy = 1$, $\nu = 2$, 50 steps for Jacobi, 5 steps of anisotropic diffusion of I every 10 cycles. The evolution (7) might be made more efficient by using an ADI (alternate direction implicit) method [39].

When working with a color image, we perform inpainting on its three components separately (one luminance image and two chroma images), and join the results at the end (in the same was as in [2]). The algorithm is programmed in tens of lines of C++ code. The results shown here were obtained in a few seconds of CPU time of a standard PC under Linux.

3.1. Inpainting of stills

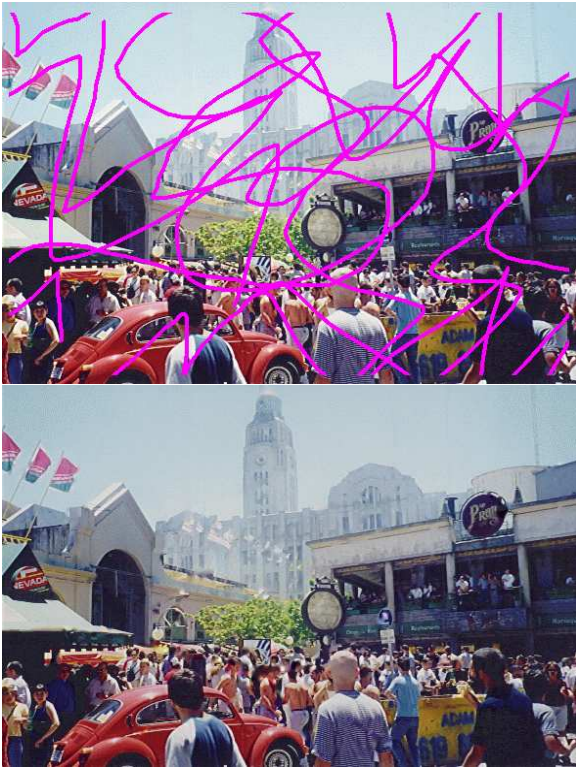In this example we consider a still image, shown in Figure 1 top, with thick lines obscuring parts of the photograph.

Figure 1. (top) Color photograph with lines obscuring parts of the image, (bottom) Navier-Stokes inpainting restoration of the photograph.

Figure 2. Four frames from the 'Foreman' video in which the lettering has been removed using Navier-Stokes inpainting (available electronically).

The bottom image in the figure shows the result of Navier-Stokes inpainting applied to the photo with the inpainting region corresponding to the obscuring lines.

Notice the result is sharp, edges are not blurred inside Ω nor do color artifacts appear. This is a clear example for which texture synthesis or manual selection algorithms are not a good choice, since the inpainting region is complex. The topology of the inpainting region does not pose a problem, unlike the approach in [25, 26].

3.2. Video inpainting

Figure 2 shows four frames of the 'Foreman' video in which lettering overlay has been removed using Navier-Stokes inpainting. The inpainting is done frame-by-frame using the same method outlined in the previous subsection. Notice that this approach, though straightforward, shows very good results: sharp, no color artifacts, no motion artifacts.

3.3. Super-resolution

This example uses Navier-Stokes inpainting to increase the resolution of an image. We have taken a detail I of size (n,m) of the original image, the right eye of the girl in Figure 3. Then we increased its size 9-fold with replication (zero-th order interpolation) to obtain I' of size (9n,9m). Finally we apply Navier-Stokes inpainting to I', while fixing I and $\nabla^{\perp} I$, from the original small image, on the lattice of points with coordinates (9i,9j). Notice how inpainting reproduces a round iris.

In this example, we use Navier-Stokes inpainting only for the brightness component of the color image. A simpler filter is used for the chroma components. Better results may be achieved via harmonic map smoothing of vectors (see [35]). For an axiomatic/PDE based technique for im-
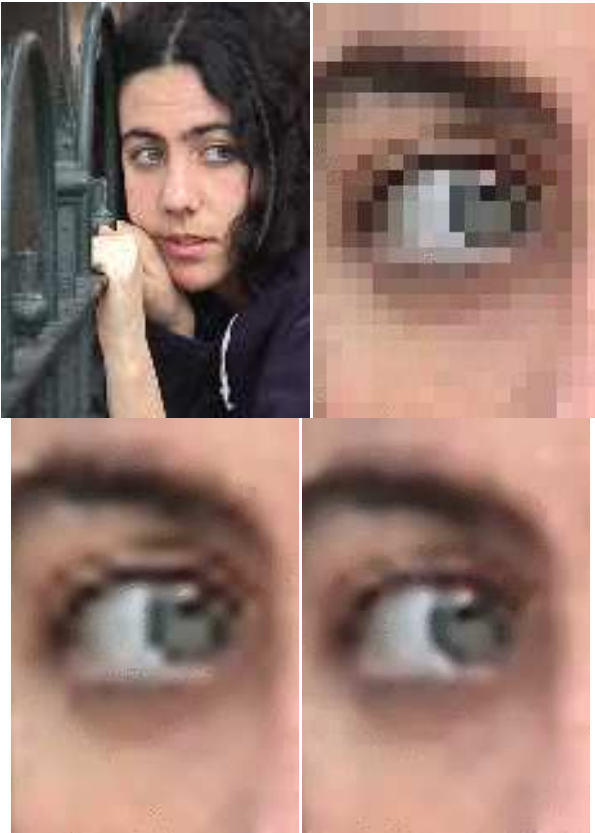
Figure 3. Top left: Original image of girl, 120X160 pixels. Top right: one of her eyes (19x27 pixels) magnified with zeroth order zoom to 169x241 pixels, Bottom left: bicubic interpolation, Bottom right: Navier-Stokes inpainting. Note how the edges are smoother with the NS approach, e.g., the eye ball.

age iterpolation, see [5].

4. Summary and Conclusions

In this paper, we show the importance of computational fluid dynamics (CFD) in general, and Navier-Stokes equations in particular, to vision problems. Although CFD ideas have been used in computer graphics for such problems as modeling natural phenomena, in shape analysis following [29], and as an interpretation for anisotropic diffusion [27], the connection and application here is to the best of our knowledge novel. Having such a direct interdisciplinary connection has several benefits for the computer vision community. First it brings well-established numerical methods and theory to a problem of central importance in vision research. Second, the very close connection to CFD has the potential to draw more people from that community

to explore problems in image processing and computer vision. This can only serve to accelerate the development of the field.

Although we present a new method that seems to perform well, there are several interesting problems and issues to be worked out that should give improvements. First there is the design of the numerical algorithm. We use a vorticity-stream based method due to its direct connection to the method introduced in [2]. However one can consider many other methods, including direct solution of the primitive variables form of the Navier-Stokes equation (4) with linear viscosity replaced by anisotropic diffusion. There is a vast literature of CFD methods that can be modified to suit the image problems. Moreover, the close analogy between Navier-Stokes and the method described in [2] suggests the possibility to try hybrid methods that cross between the two equations, thereby giving a range of parameters that can be tuned to optimize the end result. Moreover, the theoretical knowledge base developed in connection with Navier-Stokes might provide more insight into a theory for the PDE proposed in [2]. Finally there is a very interesting connection that we have not yet explored related to the dynamical evolution of the fluid equations and stability of solutions. For example it is possible that classical two-dimensional fluid dynamics instabilities, like the Kelvin-Helmholtz instability [19, 14] for shear layers, might allow us to generate interesting pattern formation in large inpainting regions. The control of such instabilities should be linked to the magnitude of diffusion present in the model and to the choice of initial condition.

At the very practical level, we have not dealt with textures, and the parameters are set manually. In this respect we believe that the user may be required to tune both the parameters and initial data in the inpainting region to suit the particular problem at hand. However, our experience suggest that large classes of problems can be efficiently solved with the same or similar ranges of parameters and initial conditions.

We also expect that this connection, when extended to higher dimensions, will allow us to generalize the inpainting problem to other features (e.g., optical flow) and other modalities. We are currently working on a video inpainting technique, based on the framework presented here, to automatically switch between texture and structure inpainting, as well as to permit the use of information from other frames when it is available.

Acknowledgments

References

[1] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, *IEEE Trans. Img. Proc.*, **10**, pp. 1200-1211, 2001.

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

[3] C. Braverman. *Photoshop retouching handbook*. IDG Books Worldwide, 1998.

[4] G. L. Browning and H. O. Kreiss, *Math. Comp.*, **52(186)**, 369-388, 1989.

[5] V. Caselles, J. M. Morel, and C. Sbert, *IEEE Trans. Img. Proc.*, March 1998.

[6] F. Catte, P. Lions, J. Morel, and T. Coll, *SIAM J. Num. Anal.*, **29**, 1: 182-193, 1992.

[7] T. Chan and J. Shen, *UCLA CAM Report* **00-11**, March 2000.

[8] R. J. DiPerna and A. J. Majda, *Commun. Pur. Appl. Math.*, **40**, 301-345, 1987.

[9] A. Efros and T. Leung, *Proc. IEEE International Conference Computer Vision*, 1033-1038, Corfu, Greece, September 1999.

[10] G. Emile-Male, *The Restorer's Handbook of Easel Painting*. Van Nostrand Reinhold, New York, 1976.

[11] G. P. Galdi, *An Introduction to the Mathematical Theory of the Navier-Stokes Equations, Vol II., Nonlinear Steady Problems*, Springer Tracts in Natural Philosophy, vol. 39, Springer-Verlag, New York, 1994.

[12] C. Greengard and E. Thomann, *Commun. Pur. Appl. Math.*, **41**, 295-303, 1988.

[13] D. Heeger and J. Bergen. *Pyramid based texture analysis/synthesis*. Computer Graphics, 229-238, SIGGRAPH 95, 1995.

[14] H. von Helmholtz. *Monatsber. Berlin. Akad.*, 215–228, 1868.

[15] A. Hirani and T. Totsuka. *Combining Frequency and spatial domain information for fast interactive image noise removal*. Computer Graphics, pp. 269-276, SIGGRAPH 96, 1996.

[16] L. Joyeux, O. Buisson, B. Besserer, S. Boukir, *Proceedings of CVPR'99, IEEE Int. Conf. on Computer Vision and Pattern Recognition,* Fort Collins, Colorado, USA, June 1999.

[17] V. Judovich, *Math. Sb. N. S.*, **64**, 562-588, 1964.

[18] T. Kato, *Arch. Rat. Mech. Anal.*, **25**, 188-200, 1967.

[19] Lord Kelvin. *Nature*, 50:524, 549, 573, 1894.

[20] D. King. *The Commissar Vanishes*. Henry Holt and Company, 1997.

[21] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, P.J.W. Rayner, *IEEE Trans. Img. Proc.* **11(4)**, 1496-1508, 1995.

[22] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, P.J.W. Rayner, *IEEE Trans. on Img. Proc.* **11(4)**, 1509-1519, 1995.

[23] O. A. Ladyzhenskaya, *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach Science Publishers, New York-London, 1963.

[24] A. J. Majda and A. L. Bertozzi, *Vorticity and incompressible flow*, Cambridge Univ. Press, 2001.

[25] S. Masnou, *Filtrage et Desocclusion d'Images par Méthodes d'Ensembles de Niveau*, Thèse, Univ. Paris-Dauphine, 1998.

[26] S. Masnou and J.M. Morel, *Level-lines based disocclusion*. 5th IEEE Int'l Conf. on Image Processing, Chicago, IL. Oct 4-7, 1998.

[27] J. Monteil and A. Beghdadi, *IEEE Trans. Patt. Anal. Mach. Int.*, **21(9)**, 940-946, 1999.

[28] M. Nitzberg, D. Mumford, and T. Shiota, *Filtering, Segmentation, and Depth*, Springer-Verlag, Berlin, 1993.

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

[30] R. Peyret and T. D. Taylor, *Computational methods for fluid flow*, Springer Verlag, 1983.

[31] D. L. Ringach and R. Shapley, *Vision Research* **36**, 3037-3050, 1996.

[32] M. Renardy, *Mathematical Analysis of Viscoelastic Flows*, CBMS-NSF Conference Series in Applied Mathematics 73, Soc. Ind. Appl. Math., Philadelphia, 2000.

[33] L. Rudin, S. Osher and E. Fatemi. *Phys. D* **60**, 259-268, 1992.

[34] E. Simoncelli and J. Portilla, *Texture characterization via joint statistics of wavelet coefficient magnitudes*. 5th IEEE Int'l Conf. on Image Processing, Chicago, IL. Oct 4-7, 1998.

[35] B. Tang, G. Sapiro, and V. Caselles, *Int. Journal Computer Vision* **36:2**, pp. 149-161, 2000.

[36] R. Temam, *Navier-Stokes Equations,* North–Holland Publishing Company, Amsterdam, second edition, 1986.

[37] M. VanDyke, *An Album of Fluid Motion,* The Parabolic Press, Stanford, CA, 1983.

[38] S. Walden, *The Ravished Image*. St. Martin's Press, New York, 1985.

[39] J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever, *IEEE Trans. Img. Proc.*, 7(3):398–410, March 1998.

[40] W. Wolibner, *Math. Zeit.*, **37**, 698-726, 1933.