IV Iberoamerican Symposium in Computer Graphics - SIACG 2009
O. Rodríguez, F. Serón, R. Joan-Arinyo, J. Madeiras, J. Rodríguez, E. Coto (Editors)

Fluid Warping

Dalia Bonilla[1], Luiz Velho [1], André Nachbin[1] and Luis Gustavo Nonato[2]

[1]IMPA Instituto Nacional de Matemática Pura e Aplicada
[2]Universidade de São Paulo-São Carlos

Abstract

Warping techniques can be complicated and difficult to use, but through the use of fluid dynamics the warping becomes simple and it is intuitively controlled by physical properties such as viscosity and forces. These properties are naturally associated with the image itself or with spatial control handles. The key idea is to think of the image domain as a two-dimensional incompressible and homogeneous fluid, and to use the Navier Stokes equations to change it by applying forces to the image function. In this way, the process does not move the image values as in fluid simulations, but transforms the coordinates of a parametrization of the image through a vector field generated by the simulation equations — effectively acting as a texture mapping. The contribution of this work is a new method for image warping based on fluid simulation.

1. Introduction

The process that changes the shape of objects in an image is called *warping*. The use of warping plays an important role in many applications, from the correction image distortions in medical data to the creation of special effects in the entertainment industry. Warping can also be used together with blending for creating transitions between different objects, a technique known as *morphing* [GV97].

More formally, given an image, $f : U \subset \mathbb{R}^2 \rightarrow C$, the mapping between source space (u,v) and destination space (x,y) is called *warping filter*. Such a map, $W(f) = g$, acts on the input image $f(u,v)$ giving rise to an output image $g(x,y)$ that can be regarded as a deformation of the image domain [Hec89]. Furthermore, in the case of morphing, a composition operator combines the result of two synchronized warping filters applied to different images.

Spatial transformations based on Navier-Stokes equations, and the use of fluid dynamics in general, present a great potential in the above context because they are very powerful to derive warping transformations. In the present paper we develop a framework where such a technique is exploited.

2. Related Work

There are many methods for image warping. They can be classified into: parameter-based; feature-based; free-form; and hybrid [GDCV99].

Parameter-based methods are warping techniques controlled by a family of transformations, such as scaling, twisting, and bending. This type of technique was introduced in Computer Graphics by Alan Barr [Bar84].

Feature-based methods encompass a whole class of warping techniques, which differ regarding types of geometric features and reconstruction functions. In these methods a correspondence of features in the source and destination objects must be provided by the user. Typical reconstruction functions include scattered data interpolation, inverse distance weighted kernels,and radial basis [AR95]. An efficient warping technique, based on feature vectors, was introduced by T. Beier and S. Neely [BN92].

Free-form based methods use specification by coordinate systems. For this purpose, they employ free-form curves (B-splines, Bézier etc.) to define the coordinate curves [BJ03]. These techniques was introduced by Smith [Smi87], developed by Smithe [Smi90] and described in Wolberg [Wol94].

The pioneer work using fluid dynamics in image processing was introduced by Bertalmio et al. [BBS01], with a method for digital inpainting. They think of the image intensity as a *stream function* and the Laplacian of the image

intensity plays the role of the vorticity of the fluid, which is transported into the region to be inpainted by a vector field defined by the stream function. Even though their idea is based on Navier-Stokes equations, our work is different. In our technique, the fluid warping uses a vector field directly created by the velocity of fluid by Navier-Stokes equations: we neither use the stream function nor vorticity.

The proposed fluid warping technique carries the coordinates of a parametrization of the image through of a vector field generated by the Navier-Stokes equations. Warping is controlled by physical parameters associated with the characteristics of the image itself or by other auxiliary images and applied to the dynamic simulation.

Depending on the application (for example in those already mentioned above) an important aspect of fluid warping is user control. Deformations are easy to specify using fluid dynamics through physical properties, such as viscosity and forces.

3. Fluid Simulation

In this section we present the fluid simulation setting that will be adopted in our framework for image warping. We are interested in modeling homogeneous, incompressible fluids with variable viscosity. Such formulation gives a good compromise between simplicity and expressive power. Moreover, we require an efficient implementation, that allows large time-steps, for this purpose we will employ the Stable Fluids algorithm, extending it to handle variable viscosity.

3.1. Mathematical Formulation

First we review some mathematical concepts for the definition of the fluid equations. The fluid is defined over a region $D \in R^2$. Let $x = (x,y)$ be any point of D. Let $v(x,t)$ denote the velocity of the particle of fluid moving through x at time t. The *velocity field of fluid* is denoted by v and it is a vector field tangent to the trajectory of the particle. For each time t assume that the fluid has mass density $\rho(x,t)$. The fluid is called *incompressible* when

$$\nabla \cdot v = 0.$$

Assuming that the fluid has mass density constant in space (i.e., the fluid is *homogeneous*) and that ρ is constant in time. Then we have

$$\rho(x,t) = constant \quad \text{in particular } \rho(x,t) = 1.$$

Let μ be the *viscosity* of fluid and p the *pressure*. Then for constant viscosity, the basic Navier-Stokes equations for incompressible fluids are: (see [CM93] and [Nac06])

$$\begin{aligned} \partial_t v &= -\nabla p - v \cdot \nabla v + \mu \triangle v + f \\ \nabla \cdot v &= 0. \end{aligned} \tag{1}$$

where the "$\cdot$" denotes a dot product between vectors, while $\nabla = (\partial/\partial x, \partial/\partial y)$ is the vector of spatial partial derivatives, ∂_t is the partial derivative $\frac{\partial}{\partial_t}$, and $\nabla\cdot$ is the divergent. We also adopt the notation $\triangle = \nabla \cdot \nabla$.

For more expressivity, we want to consider changes of viscosity in space and thus we have to formulate the Navier-Stokes equations for variable viscosity. The equations are:

$$\begin{aligned} \partial_t &= -\nabla p - v \cdot \nabla v \\ &+ \nabla \mu(x)(\nabla v + \nabla v^\top) + \mu(x) \triangle v \\ &+ f \end{aligned} \tag{2}$$

Where $\nabla v = \begin{pmatrix} \partial_x v^1 & \partial_y v^1 \\ \partial_x v^2 & \partial_y v^2 \end{pmatrix}$ and $\nabla v^\top = \begin{pmatrix} \partial_x v^1 & \partial_x v^2 \\ \partial_y v^1 & \partial_y v^2 \end{pmatrix}$ then

$$\nabla v + \nabla v^\top = \begin{pmatrix} 2\partial_x v^1 & \partial_x v^2 + \partial_y v^1 \\ \partial_y v^1 + \partial_x v^2 & 2\partial_y v^2 \end{pmatrix}.$$

We will use the Helmholtz-Hodge Decomposition theorem, where a vector field w on D can be uniquely decomposed in the form

$$w = u + \nabla q$$

such that u has zero divergence and q is a scalar field. If we have $w = u + \nabla p$, then

$$\nabla \cdot w = \nabla \cdot \nabla q = \triangle q, \text{ and } w \cdot n = \nabla q \cdot n = \frac{\partial q}{\partial n} = 0.$$

This is a Poisson equation for a scalar field with the Neumann boundary conditions. A solution to this equation can be used to compute u:

$$u = w - \nabla q.$$

Now we define the operator P which projects any vector field w onto its divergence free part u: $\nabla \cdot u = 0$ and $Pw = u$ (see [Sta99] and [CM93]). P is a linear operator and thus $w = Pw + \nabla p$, $u = Pu$, $P(\nabla p) = 0$. We apply the operator P to both sides of the basic Navier-Stokes equations (1) and obtain

$$\partial_t v = P(\partial_t v + \nabla p) = P(-(v \cdot \nabla)v + \mu \triangle v + f). \tag{3}$$

This form of equation eliminates the pressure and expresses $\partial_t v$ in terms of v alone. The pressure can then be recovered as the gradient part of $-(v\cdot\nabla)v + \mu \triangle v + f$. In the same way, we obtain for the variable viscosity Navier-Stokes equation (2)

$$\begin{aligned} \partial_t v = \\ P(-(v \cdot \nabla)v + \nabla \mu(x)(\nabla v + \nabla v^\top) + \mu(x) \triangle v + f). \end{aligned} \tag{4}$$

3.2. Computational Method

Now we briefly review the Stable Fluids algorithm that we adopt for the implementation of the fluid simulation. For further details we refer to the papers [Sta99] and [Sta03]. Since the algorithm was designed for fluids with constant viscosity, we adapt it for our setting by developing an extension for variable viscosity fluids. The algorithm solves the Navier-Stokes equations (1) and it is unconditionally stable. This method is based on an operator splitting strategy. For each time step $\triangle t$ the algorithm solves the equations in four stages, starting from a velocity field $w_0 = v(x,t)$ of a previous time step and then sequentially resolving each term of the equations. The stages are

$$w_0 \xrightarrow{\text{add force}} w_1 \xrightarrow{\text{advect}} w_2 \xrightarrow{\text{diffuse}} w_3 \xrightarrow{\text{project}} w_4$$

The first stage is the addition of external force f. It adds the force field multiplied by the time step to velocity $w_1 = w_0 + \triangle t\, f(x,t)$.

The second stage accounts for the effect of advection of the fluid on itself. It is given by an advection equation

$$\partial_t w_2 = -(w_1 \cdot \nabla) w_2$$

and is solved by using a semi-Lagrangian technique [CIR53]

$$w_2(x) = w_1(x - \triangle t w_1(x)).$$

The basic idea behind the advection step is, instead of moving the particle forward in time through the velocity field, to move it backwards in time through the field and calculate a new velocity by interpolation (guaranteeing stability).

The third stage solves for the effect of viscosity and is given by equation

$$\partial_t w_3 = \mu \triangle w_3$$

it uses a simple implicit solver for the diffusion equation

$$(I - \triangle\, t\mu \triangle) w_3 = w_2$$

where I is the identity operator. One way to solve this equation is to get the solution for system

$$A w_3 = w_2$$

by using the Jacobi method.

The fourth stage projects the velocity field onto the incompressible (divergence free) field. This step also involves the solution of a Poisson equation

$$\triangle\, q = \nabla \cdot w_3 \text{ and } w_4 = w_3 - \nabla q.$$

The methods used for solving this stage are finite difference schemes and Jacobi.

We will extend the formulation above to solve our equation (2) using the same stages, except that because we assume variable viscosity the diffusion stage will be different. More precisely, the original Stable Fluids solve the equation

$$\partial_t w_3 = \mu \triangle w_3$$

and we have to solve the equation

$$\partial_t w_3 = \nabla \mu(x)(\nabla w_3 + \nabla w_3^{\top}) + \mu(x) \triangle w_3.$$

To solve this step, let $w_3 = (w^1, w^2)$ and we write again in this form

$$\begin{cases} w_t^1 = 2\mu_x w_x^1 + \mu_y(w_y^1 + w_x^2) + \mu \triangle w^1 \\ w_t^2 = 2\mu_y w_y^2 + \mu_x(w_y^1 + w_x^2) + \mu \triangle w^2. \end{cases}$$

where the sub-indices t, x and y denote the partial derivatives ∂_t, ∂_x and ∂_y. And we discretize again using implicit difference schemes backward in time and central space to w_3 and central space scheme to μ [Str99]. The results are stable, just as we wanted. The discretization of these equations is given in Appendix A. Once again time we have used Jacobi to solve this modified stage.

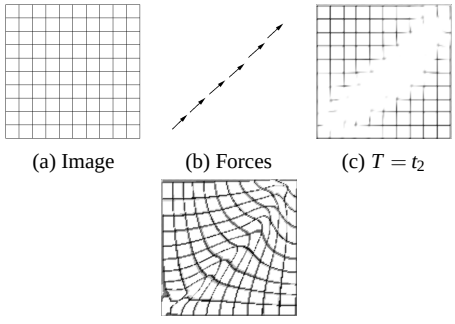
The Stable Fluids also includes a method to compute the motion of a density φ immersed in the fluid. The equation for the evolution of this density is

$$\partial_t \varphi = -(v \cdot \nabla)\varphi + \kappa \triangle \varphi + S \qquad (5)$$

where κ is a diffusion rate and S is a source of density. The density is advected by the fluid using a semi-Lagrangian technique, as in the second stage of the algorithm.

4. Warping with Fluids

The first idea to use fluids for image warping would be to consider the image as a density field immersed in the fluid, and as such could be transported by fluid motion. However, as can be seen in Figure 1 this approach does not work well.



(a) Image (b) Forces (c) $T = t_2$

(d) $T = t_2$

Figure 1: Image warping with fluids.

For this experiment we used the source image in Figure 1a and the force field in Figure 1b. We performed a fluid simulation using these external forces, and then applied the resulting velocity field to the image in two ways.

First, we treated the image intensity as density values immersed in the fluid and computed the motion of the image function using equation (5), which moves the grey level values of the image directly. The result is a blurred image and

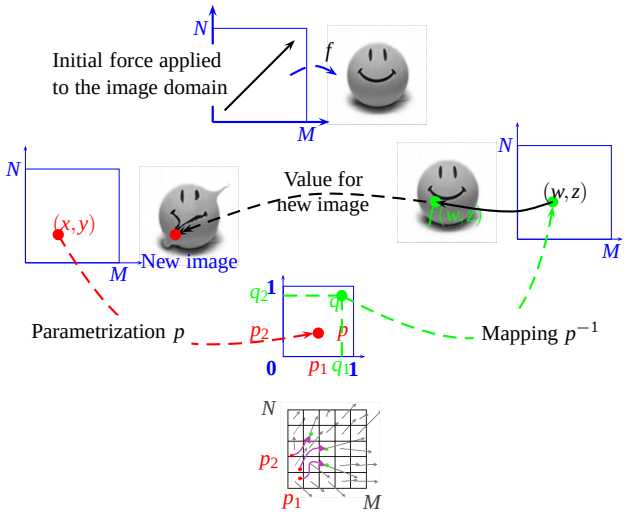


Figure 2: **Texture Warping** Image coordinates p_1 and p_2 carried by vector field generated by the Navier Stokes equations

a fast disappearance of the image after applying additional forces, as shown in Figure 1c.This phenomenon happens because of the intrinsic dissipation of the fluid equation.

The best strategy to avoid the above problem is to move through the fluid velocity field, instead of image values, the coordinates of a parametrization of the image. In this way, the image features are preserved by a texture mapping mechanism, as shown in Figure 1d.

4.1. Texture Mapping and Warping

Based on the conclusions of the previous experiment, the image warping using fluid simulation can be formalized as a deformation induced on a texture.

Let $[0,M]\times[0,N]$ be the image domain. Given a force applied to the image domain, we regard a fluid in $[0,M]\times[0,N]$ and move this fluid taking external forces as initial data. For point (x,y) in the image domain, the coordinates are transported to point (p_1,p_2) in $[0,1]\times[0,1]$ by parametrization p. Now, each coordinate p_1 and p_2 is interpreted as a density immersed in the fluid on $[0,M]\times[0,N]$. We move each coordinate separately through the fluid velocity field, by equation (5) getting for a time step $\triangle t$ the coordinates q_1 and q_2. Finally, for a point (x,y) on domain of a new image the value is computed as follows. If (q_1,q_2) belongs to $[0,1]\times[0,1]$, then it is transported by the inverse parametrization at the point (w,z) of $[0,M]\times[0,N]$. The value of the new image at point (x,y) is the value of the original image at (w,z). If (q_1,q_2) do not belong to $[0,1]\times[0,1]$ then the value of (x,y) is zero. This computational scheme for fluid warping using texture mapping is illustrated in the diagram of Figure 2.

The warping function induced by the velocity field of a fluid simulation has many desirable properties, such as smoothness and continuity, that can be exploited in applications. Additionally, the mapping is naturally time dependent, such that given the initial conditions (i.e., forces and parameters) at time $t=0$, we have a one-parameter family of warpings W_t, $t\in\mathbb{R}_+$, which directly applies for animation, and may also be interpreted in terms of evolution.

4.2. Control Mechanisms

Fluid simulation is capable of producing potentially very complex deformations with good properties for image warping, as we have discussed so far. However, in order to be useful, we need to be able to control the simulation such that the desired transformation is obtained. The simplest way to control the fluid warping is through the direct specification of the simulation parameters. By inspection of the fluid equations (2) is easy to verify that the available parameters are:

- external forces $f(x,t)$; and
- fluid viscosity $\mu(x)$.

An extra parameter is the total duration T of the simulation.

Surprisingly, just this small set of parameters already provide powerful and intuitive mechanisms for controlling the image warping.

First, note that both the forces and viscosity are spatially variable. Thus, they are identified with functions on the image domain which could be associated with image features. Therefore, one natural way to specify forces and viscosity is by auxiliary images.

We define the viscosity from the intensity of an auxiliary image function on $[0,M]\times[0,N]$. The viscosity values are computed from a normalization of the image values to $[0,1]$ and global scaling factor. While the viscosity is defined by a scalar field, forces are defined by a vector field, which can be encoded as an RGB image. However, in many situations it is also convenient to specify forces from point or curve sources. For this we employ procedural definitions.

Furthermore, for specification purposes it is convenient to take the total external force $f=\sum f_i$, as the additive combination of separate individual forces f_i. Since forces vary in time too, we must take this fact into consideration for the definition of forces. More specifically, useful options are: 1) instantaneous forces (i.e., acting at specific time instants t_i, $i=1,\ldots,N$) In this case, it is common to use initial forces at $t=0$; 2) constant forces (i.e., acting during a certain time interval $[t_0,t_1]$; 3) arbitrary forces (i.e., fully variable in time). Also, the application of forces may be defined *a priory* or may depend on a sensing function on the simulation. This second option is related to the so-called force-feedback mechanism of control theory. Up to now, we have adopted only basic specification of forces, such as instantaneous and constant force fields. We have also made use of a simple sensing mechanism, mainly for stopping the simulation.

5. Results

In this section we give some examples of the results that can be obtained with fluid warping.

The first set of examples demonstrate the use of forces and viscosity to control image deformations.

Figure 3 shows a warping of a train leaving the station, defined only by an instantaneous force at $t = 0$ along a curve on the top part of the image. The goal was to get a time-space distortion effect suggesting speed. Note that we are able to create extreme distortions just by running the simulation for a longer period.

Figure 4 exhibits a warping of a Van Gogh's painting, defined by forces and variable viscosity. Here, we generated forces from the gradient of a segmentation of the hat and the viscosity from a quantization of the image values. Note that the forces act to expand the hat and, because the surrounding background has variable viscosity, the hat deforms in a non-uniform way.

The second set of examples is an attempt to evaluate fluid warping as a regular warping technique. For this purpose, we make a comparison using examples from the paper of Arad et al [AR95]. In this seminal work, the authors describe a technique based on radial basis functions.

Figure 5 is an example in [AR95] for lifting the corner of a girl's mouth. To achieve this effect we constructed a viscosity function, shown in Figure 5b, that imposes a restriction on the warping area. In this function, the region outside the desired warping is white and more viscous (i.e., opposing great resistance to fluid motion). The warping area is dark and less viscous (i.e., offering small resistance to fluid motion). The forces used in the process are given by the gradient field of the image shown in Figure 5c. They exert an upward force at the mouth location, producing the desired effect. The results in Figures 5d and 5e demonstrate that we are able to closely match their technique. A similar performance is achieved in the example of Figure 6.

6. Conclusions

In this paper we introduced *fluid warping*, a framework for image deformation using fluid dynamics. Our technique provides good results and is simple to use. Future work includes two avenues of investigation. One direction is towards a finer and more precise control of the warping transformation. This can be achieved by exploiting sophisticated force-feedback mechanisms, possibly combined with an optimization strategy. Another direction is to extend the framework for image morphing. In this context, the coupling of two parallel fluid simulations could be considered together with a composition operator for image blending. Ideally, such an operator should be physically inspired.

References

[AR95] ARAD N., REISFELD D.: Image warping using few anchor points and radial functions. *Computer Graphics Forum 14*, 1 (1995), 35–46.

[Bar84] BARR A. H.: Global and local deformations of solid primitives. *ACM SIGGRAPH Computer Graphics 18*, 3 (1984), 21–30.

[BBS01] BERTALMIO M., BERTOZZI A., SAPIRO G.: Navier-stokes fluid dynamics and image and video inpainting. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)* (December 2001), 9–14.

[BJ03] BIRKHOLZ H., JACKÈL D.: Image warping with feature curves. *In Proceedings of SIGGRAPH* (2003), 199–202.

[BN92] BEIER T., NEELY S.: Feature-based image metamorphosis. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1992), ACM, pp. 35–42.

[CIR53] COURANT R., ISAACSON E., REES M.: On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics 5* (1953), 243–255.

[CM93] CHORIN A., MARSDEN J. E.: *A mathematical Introduction to Fluid Mechanics*. Springer-Verlag, 1993.

[GDCV99] GOMES J., DARSA L., COSTA B., VELHO L.: *Warping and Morphing of Graphical Objects*. Morgan Kaufmann Publ., 1999.

[GV97] GOMES J., VELHO L.: *Image Processing for Computer Graphics*. Springer Verlag, 1997.

[Hec89] HECKBERT P. S.: *Fundamentals of Texture Mapping and Image Warping*. Master's Thesis, University of California, Berkeley, 1989.

[Nac06] NACHBIN A.: *Notas do Curso: Dinâmica dos Fluidos*. 2006.

[Smi87] SMITH A. R.: Planar 2-pass texture mapping and warping. *In Proceedings of SIGGRAPH* (1987).

[Smi90] SMITHE D. B.: A two-pass mesh warping algorithm for object transformation and image interpolation. *Technical memo, Industrial Light and Magic* (1990).

[Sta99] STAM J.: Stable fluids. *SIGGRAPH 99 Conference Proceedings, Annual Conference Series* (August 1999), 121–128.

[Sta03] STAM J.: Flows on surfaces of arbitrary topology. *ACM Transactions On Graphics (TOG), Proceedings of SIGGRAPH* (July 2003), 724–731.

[Str99] STRIKWERDA J. C.: *Finite Difference Schemes and Partial Differential Equations*. CRC Press, 1999.

[Wol94] WOLBERG G.: *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1994.

Appendix A: Discretization for Variable Viscosity

In this appendix we provide the discretization of the diffusion stage of the simulation, given by the equations below:

$$\begin{cases} u_t = 2\mu_x u_x + \mu_y(u_y + v_x) + \mu \triangle u \\ v_t = 2\mu_y v_y + \mu_x(u_y + v_x) + \mu \triangle v. \end{cases}$$

The derivation is obtained by a finite difference scheme. We assume that the grid spacing is $h = 1/N$.

$$\begin{aligned} \frac{u^{n+1}_{m,l} - u^n_{m,l}}{\triangle t} = \\ 2\mu_x \left[\frac{u^{n+1}_{m+1,l} - u^{n+1}_{m-1,l}}{2h} \right] + \\ \mu_y \left[\frac{u^{n+1}_{m,l+1} - u^{n+1}_{m,l-1}}{2h} + \frac{v^{n+1}_{m+1,l} - v^{n+1}_{m-1,l}}{2h} \right] + \\ \mu \left[\frac{u^{n+1}_{m-1,l} + u^{n+1}_{m+1,l} + u^{n+1}_{m,l-1} + u^{n+1}_{m,l+1} - 4u^{n+1}_{m+1,l} - 4u^{n+1}_{m,l}}{h^2} \right] \end{aligned}$$

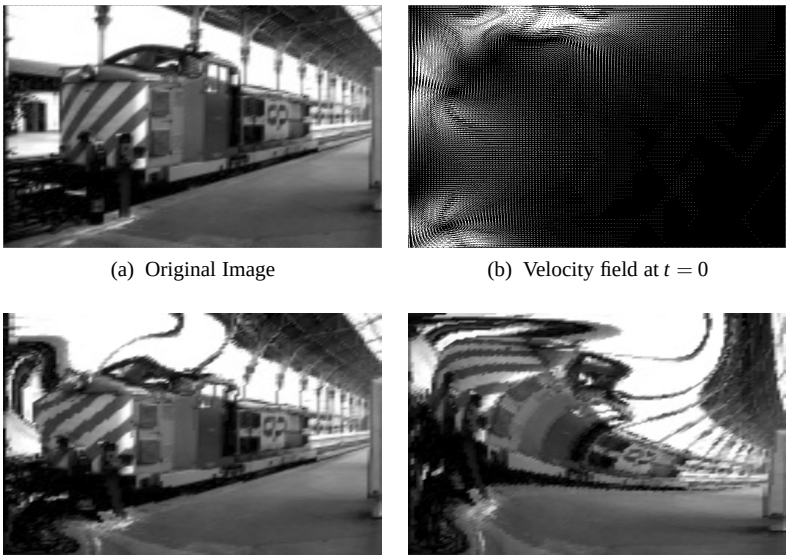
And then we have

$$\begin{aligned} u^{n+1}_{m,l} + \frac{4 \triangle t\mu}{h^2} u^{n+1}_{m,l} = \\ 2 \triangle t\mu_x \left[\frac{u^{n+1}_{m+1,l} - u^{n+1}_{m-1,l}}{2h} \right] + \\ \triangle t\mu_y \left[\frac{u^{n+1}_{m,l+1} - u^{n+1}_{m,l-1}}{2h} + \frac{v^{n+1}_{m+1,l} - v^{n+1}_{m-1,l}}{2h} \right] + \\ \triangle t\mu \left[\frac{u^{n+1}_{m-1,l} + u^{n+1}_{m+1,l} + u^{n+1}_{m,l-1} + u^{n+1}_{m,l+1} - 4u^{n+1}_{m+1,l}}{h^2} \right] + u^n_{m,l} \end{aligned}$$

$$\begin{aligned} u^{n+1}_{m,l} = \frac{1}{1+4\mu\triangle t} \Big\{ u^n_{m,l} + \frac{\triangle t\mu}{h^2} \left(u^{n+1}_{m+1,l} + u^{n+1}_{m-1,l} + u^{n+1}_{m,l+1} + u^{n+1}_{m,l-1} \right) \\ \frac{\mu_x \triangle t}{n} \left[u^{n+1}_{m+1,l} - u^{n+1}_{m-1,l} \right] + \\ \frac{\triangle t\mu_y}{2h} \left[u^{n+1}_{m,l+1} - u^{n+1}_{m,l-1} + v^{n+1}_{m+1,l} - v^{n+1}_{m-1,l} \right] \Big\} \end{aligned}$$

And finally in the same way we obtain the discretization of v_t

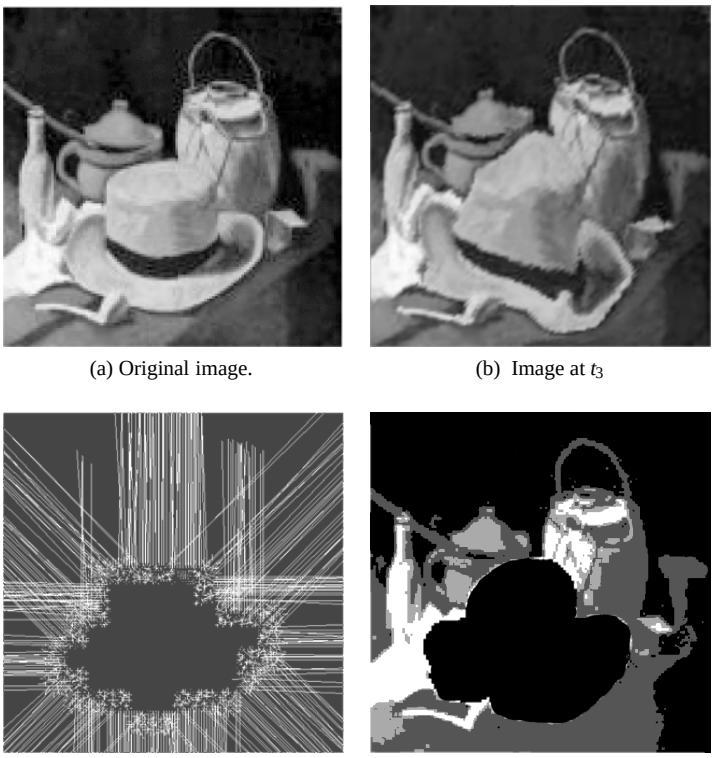
$$\begin{aligned} v^{n+1}_{m,l} = \frac{1}{1+4\mu\triangle t} \Big\{ v^n_{m,l} + \frac{\triangle t\mu}{h^2} \left(v^{n+1}_{m+1,l} + v^{n+1}_{m-1,l} + v^{n+1}_{m,l+1} + v^{n+1}_{m,l-1} \right) \\ \frac{\mu_y \triangle t}{n} \left[v^{n+1}_{m,l+1} - v^{n+1}_{m,l-1} \right] + \\ \frac{\triangle t\mu_x}{2h} \left[u^{n+1}_{m,l+1} - u^{n+1}_{m,l-1} + v^{n+1}_{m+1,l} - v^{n+1}_{m-1,l} \right] \Big\}. \end{aligned}$$



(a) Original Image (b) Velocity field at $t = 0$

(c) Image warping at t_1 (d) Image warping at t_n

Figure 3: Train warping.



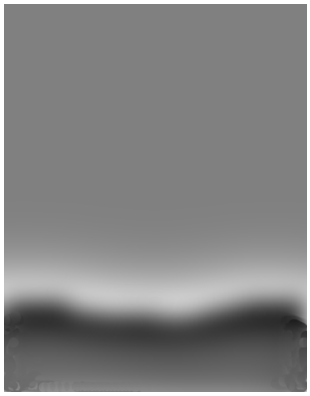
(a) Original image. (b) Image at t_3

(c) Gradient field. (d) quantization for viscosity

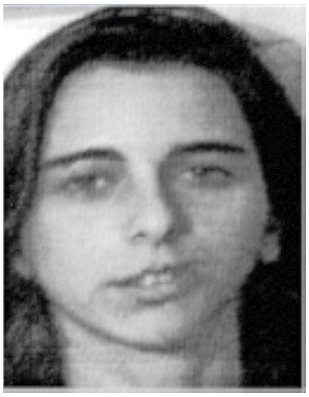
Figure 4: Melting the Van Gogh hat.

(a) Original image.

(b) Viscosity function



(c) Edges for force field.



(d) Fluid Warping

(e) Result of Arad et al. [AR95]

Figure 5: Comparison with example in Arad et al. [AR95]

(a) Original image.

(b) Fluid Warping

(c) Result of Arad et al [AR95]

Figure 6: Comparison with example in Arad et al. [AR95] -Note a small difference in the eye.