Keyframe Control of Fluid Warping and Morphing using Adjoint Method

Dalia Bonilla, Luiz Velho
VISGRAF
IMPA - Instituto de Matématica Pura e Aplicada
www.impa.br/∼{dalia,lvelho}

Luis Gustavo Nonato
Universidade de São Paulo - São Carlos
www.icmc.usp.br/∼gnonato

***Abstract*— There are many morphing techniques based on geometry, however none of them use physical models. This paper proposes a novel technique of morphing using dynamic of fluids and creating a method to control the deformation using points and keyframe. The adjoint method make the morphing more efficient. The image domain is considered as a two-dimensional incompressible fluid, and to use the Navier- Stokes equations to model the fluid. The image is deformed through a vector field generated by equations and controlling the deformation by the trajectory of points.**

***Keywords*-warping; morhing; fluids**

I. Introduction

Fluids have been extensively studied in computer graphics. Early on, fluids dynamics were simulated by using systems of particles [1], and then modeled by Navier-Stokes equations. The equations used for the simulation were first two-dimensional and then three-dimensional, and went from unstable to unconditionally stable. Nowadays, fluids simulations became physically realistic, and are used in animations.

The warping techniques, until now, are using transformations based on geometry to create deformations in image, with good results, but so far, few know techniques have exploited a deep study based on physics to create the deformations.

In this paper, we present a novel technique of image warping using fluid dynamics. Fluid dinamics for image warping was first used by Jos Stam [2]. For this we develop two warping control mechanisms. In the first one, the warping is controlled by the physical parameters of the fluid, like viscosity and forces. The user manually applies these parameters on the simulation, until the desired deformation is reached. This mechanism is easy and intuitive to handle, and good results are obtained, although being less precise to be used in morphing.

For the second mechanism, we develop an accurate and automatic technique, where the input parameters are points specified directly on the image by the user. This technique is extended from the optimization methods used for control fluids and particle systems.

The recent artistic work of New York City-based photographer Jamie Beck, presented in [3], reminds us the animated photographies seen in the Harry Potter movies. With the techniques of fluid warping and control mechanism mentioned above, we could create animated pictures from just one picture and fluid simulation.

The main contributions of this work are:

- Accurate and efficient control to image warping using fluids. It is shown that the key frame method is a precise and automatic control method. Moreover, the use of the adjoint method makes this technique efficient and precise.

- The objective function is robust. We show that the objective function we use is a quadratic one and that it possess only one minimum. Moreover, to find this minimum does not imply a pre-process of this function.

II. Related Work

According to Gomes et al. [4], the warping methods, can be classified into parameter-based, feature-based, free-form based and hybrid techniques. The parameter-based methods encompasses all the warping techniques that are controlled by parameters, such as scale, twisting and bending. An early work using this technique was done by Alan Barr [5]. Feature-based methods cover a whole class of warping techniques, with a great variety of different geometric features. The warping is defined explicitly by mapping each feature in the source object to its correspondent in the target object. Image warping with scattered data interpolation techniques belongs to the class of the feature-based warping methods. A scattered data interpolation method was introduced by Arad and Reisfel [6]. Free-form-based warping techniques uses free-form curves (B-splines, Bézier, etc.) to define the warping transformations. An early example of these techniques was introduced by Smith [7].

Some important works in a long history in fluid simulation in computer graphics are the work of Foster and Metaxas, who used the full Navier-Stokes equations to model both water [8] and gases [9] but this models are unstable for large time-steps. Then, Stam [2] introducing the Stable Fluids algorithm, which combined semi-Lagrangian advection with an implicit viscosity solver. The work of Foster and Fedkiw [10] and Enright [11] create simulations physically realistic. Here we mentioned two works on fluid control, Treuille et al [12] proposed the general framework for controlling smoke simulations using keyframing and nonlinear optimization. McNamara [12] greatly increased the speed of the optimization using the adjoint method. The adjoint method is used here to compute derivatives quickly and efficient. Some works are Giles and Pierce [13] discuss both the continuous and discrete approach of the adjoint method. And Giering and Kaminski [14] give recipes for adjoint code construction.

The pioneer work using fluid dynamics in image processing was introduced by Bertalmio et al [15] with a method for digital inpainting.

In our work proposed fluid warping technique carries the coordinates of a parametrization of the image through of a vector field generated by the Navier-Stokes equations. There the warping is controlled by physical parameters such as viscosity and forces.

also the warping is controlled by user-specified keyframes. A continuous quasi-Newton optimization solves for appropriate forces to be applied to the velocity field throughout the simulation. We use a method to efficient compute derivatives of a whole fluid simulation.

III. Fluid Simulation

We adopted and adapted the Stable Fluids algorithm for the implementation of the fluid simulation. This algorithm solves the Navier-Stokes equations, originally for the case of constant viscosity and incompressible fluid. This equations can be written in the following compact form:

$$\partial_t v = P(-v \cdot \nabla v + \mu \triangle v + f) \tag{1}$$

where v is the velocity of fluid, μ is the viscosity and f are the external forces. The operator $P(v) = w$ projects a vector field v onto its incompressible component w (divergence free, i.e, $\nabla \cdot w = 0$).

For the each time step $\triangle t$ the algorithm uses operator splitting and solves the equations in four steps. Starting form a velocity field v_0 of a previous time step, the algorithm decomposes the equations 1 sequentially:

$$v_0 \xrightarrow{\text{add force}} v_1 \xrightarrow{\text{advect}} v_2 \xrightarrow{\text{diffuse}} v_3 \xrightarrow{\text{project}} v_4$$

The step *add force* is the addition of external forces f to the velocity field v. The step *advect* transports the points and velocities through the velocity field. The step of diffusion *diffuse* is the effect of the viscosity in the fluid. The algorithm solves the equation

$$\partial_t v = \mu \vartriangle v$$

and for spacial variable viscosity it is solved the equation

$$\partial_t v = \nabla \mu(\boldsymbol{x})(\nabla v + \nabla v^\top) + \mu(\boldsymbol{x}) \vartriangle v$$

where

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

the extension for the case variable viscosity remains stable.

The last step *project*, projects the velocity field onto the incompressible (divergence free) field.

IV. Warping and Morphing with Fluids

The morphing technique consists of two warpings and a blending operation. The idea is to match the forward and inverse warping in some image features to prepare them to be blended (see [4]). At this point it is used the linear interpolation as blending and this allows the warping to dominate the whole theory of the morphing technique with fluid. Then, it is possible to see the warping technique of image using fluids. This technique is largely explained in the works [16] [17] and call it of *fluid warping*. In the warping using fluid the domain of the image are considered as a two-dimensional incompressible fluid, and use Navier-Stokes equations to create an advection vector field and thus, deform the image.

A. Texture Mapping and Warping with Fluids

Given an image $f : U \to \mathbb{R}^3$, U is a rectangle in the plane filled with a two-dimensional fluid. The intuitive idea it is to paint the image onto the fluid in U and deform the image when the fluid moves.

For this, it is necessary to create a vector field from the Navier-Stokes equations and move the texture coordinates through this field. Moving the coordinates instead of the image intensity because otherwise the image would be lost.

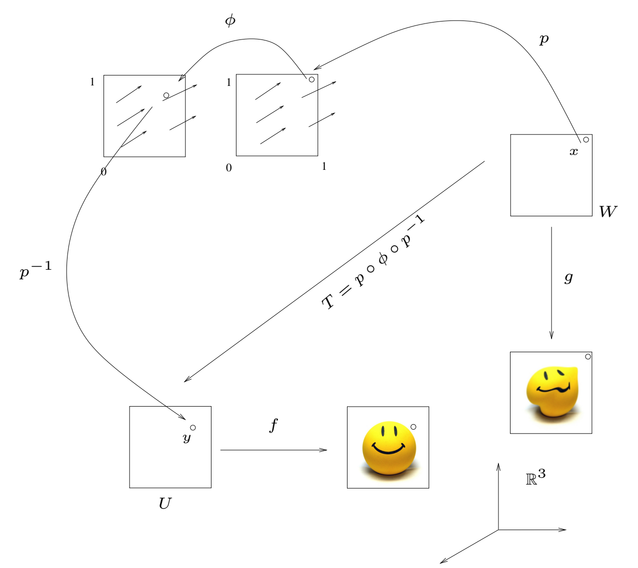


Fig. 1. Texture mapping. Taking a point x in the target image and this point is mapped to the square $[0,1] \times [0,1]$, where we apply the fluid ϕ. The position y result of the movement is mapped back in U. $g(x) = f(y)$ $g(x) = f(T(x)) = f((p^{-1} \circ \phi \circ p)(x))$

V. Control Mechanisms

Now for controlling the warping using fluids, it is necessary to control the fluid simulation. The fluids are modeled by partial differential equations: the Navier-Stokes equations. These partial differential equation (PDEs) are difficult to control because small changes in the parameters may produce different results. Here are some control methods to the *fluid warping*.

A. Indirect control

In this mechanism the fluid simulation is used for controlling the image warping. The simulation is controlled by physical parameters like the viscosity and forces.

It is observed that with viscosity is possible to control the deformation because the fluid resists to the force and then the image gradually deforms in the same direction of the force. Now, without viscosity any force produces a turbulent motion.

The viscosity of the fluid is consider also as a space variable, and the user can specify low viscosity in the area where more deformation is required and have more viscosity for the area whit less deformation. See [16].

This technique allows local control over the deformation, but depends on the intuition of the user and the warping is by trial and errors. For morphing is necessary a more accurate and precise control.

B. Differential Control

In morphing the effect of the transition of an image into another. This transition needs transform objects of the source image in objects of the target image. This transformation must preserve the topology, features of the objects of the source image, these are the principles for a good morphing [4].

Therefore it is necessary a more specific and precise technique. For this a mechanism based in keyframes has been developed. The user has in mind a goal of the deformation and mark this objective with points onto the target image. This set of points defines the keyframe.

The keyframe must be achieved at the end of the simulation. Moreover a scalar function, called objective function compare the difference of the points of the simulation with the keyframe.

The control parameters of the objective function are the forces and these drives the simulation points toward the keyframe. The forces are found using a gradient based optimizer. These optimizer method require the gradient of the objective function.

The more forces in the simulation the objective function depends on more parameters of control and this increases the number the operations to calculate the gradient.

The control mechanism by keyframe is precise, specific and automatic. Furthermore, depends on the calculus of gradient objective function to be efficient or not. Like see in [18] the direct calculus of the gradient is inefficient then we will use the adjoint method as more efficient technique.

C. Adjoint Method

The adjoint method was introduced by McNamara [18] by first time in computer graphics. In general, a simulation can be viewed as a sequence of states $q_1, q_2, ...q_N$, where each state is a finite dimensional vector. We assume that the evolution of the states is governed by the following general equation: $f(q, u) = 0$. Where $q = (q_1, q_2, ...q_N)$ and the variable u contains P the control parameters. We wish to compute some function $g(q, u)$ and perform a minimization of g. In this case the gradient of g with respect to u indicates a useful direction (e.g for nonlinear conjugate-gradient optimization). [19] The adjoint method gives an efficient way to evaluate $\frac{dg}{du}$, with a cost independent of P usually comparable to the cost of solving for q once.

To evaluate the gradient directly, we would do

$$\frac{dg}{du} = g_u + g_q q_u$$

where the subscripts indicate partial derivatives (g_u and g_q are row vectors $1 \times P$ and $1 \times N$ respectively and q_u is an $N \times P$ matrix).

Since g is a given function, g_u and g_q are supposedly easy to calculate, but on other hand, computing directly q_u is highly costly.

The adjoint method is as follow. Differentiating the f equation, we find $f_q q_u + f_u = 0$ then $q_u = -f_q^{-1} f_u$. Now we write

$$\frac{dg}{du} = g_u + g_q q_u = g_u - g_q(f_q^{-1} f_u) = g_u - (g_q f_q^{-1}) f_u$$

Then we solve the linear system

$$f_q^T \lambda = g_q^T$$

and we obtain

$$\frac{dg}{du} = g_u - \lambda^T f_u.$$

Then, we observed that to calculate $g_q q_u$ is equivalent to compute $\lambda^T f_u$. Then we make of calculus of directly $g_q q_u$ from the code using the rules described in [14] to crate the adjoint code.

VI. Technique Keyframe and Fluid to Morphing

Until now the animation using fluid simulation have used the keyframe technique but not in the context of image warping and morphing. Treulli et al see [12] present this technique for control fluid simulations. Where the keyframes guided the simulation toward the desired conditions in a given time.

In the indirect control, based on deformation goal, the user specify physical parameters and in the end of the simulation the user consider that the deformation of the image is close to the goal.

By using keyframes it can be specified the aim of the deformation At the end of the simulation, which compares the difference between the result and the target by a function, which makes the technique an automatic process.

It is prevailing to define keyframes. Treulli et al [12] defined the keyframe as densities carried by the fluid, while Wojtan [20] works with a simple movement of particles. This proposal combines these two ideas and defining keyframe as particles carried by the fluid.

A. Specification of Parameters

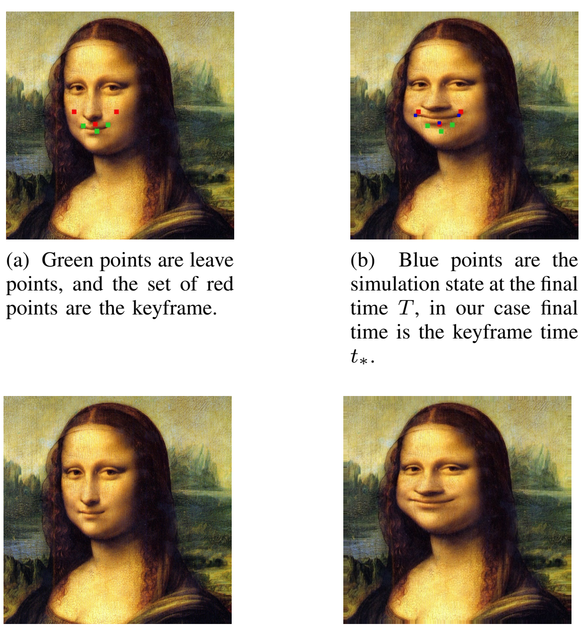
Particles are transported through the fluid and these particles are defined. Over the source image the user specific the position of the points area going to be modified and these points is called as source points.

Then over the target image the user define the position q^* of the point that he want to archive s at the end of the simulation. The point q^* the user mark the deformation objective. Let be p^l where $0 \leq l \leq T$ the sequence of points for each step of time of the simulation, such that in the time zero $p^0 = s$ where T is final time of the simulation. See Figure 2.

Over the simulation it is necessary to compare p^T and q^*. If the user marks M points over the source image then to must mark the correspondent points $\{q_i^*\}_{i=1,\ldots,M}$ over the target image. The point set $\{q_i^*\}_{i=1,\ldots,M}$ constitute the keyframe and it was store in the vector q^* while the vector p^T store the set $\{p_i^T\}_{i=1,\ldots,M}$ of points results of the simulation.

The forces to control and acting over the simulation are organized into a grid and added directly to the velocity. These force parameters are combined into the vector u and this vector have the control parameters of the objective function.

The aim is to find a vector u that minimizes the objective function and in other words the goal is to find a set of forces such that drive the source points toward the keyframe.



(a) Green points are leave points, and the set of red points are the keyframe.

(b) Blue points are the simulation state at the final time T, in our case final time is the keyframe time t_*.

(c) Original Image

(d) Final Result

Fig. 2. Control by Particle-Keyframe.

B. Objective Function

Now to decide if the deformation reaches the goal is the objective function that measures the distance between source points of the simulation and the keyframe. The function are defined by

$$\varphi(u) = \frac{1}{2}\sum_{l=0}^{M} \|p_l^T - q_l^*\|^2$$

This function compares directly the distance of points, therefore the function is quadratic and with a absolute minimum. This function is used then to control a fluid simulation, it is more robust that the function present in [4] that compare densities and for this need the term necessary to control the problem of relative minimum or stagnation points when to compare densities.

In our formulation we also do not use smoothing term $\|u\|^2$ because the aim is to reach more precision for the keyframe and the system does not penalized for excessive use of the control.

C. How Our Technique Works

Let be the images $f : U \rightarrow \mathbb{R}^3$ and $g : W \rightarrow \mathbb{R}^3$, where $U, W \in \mathbb{R}^2$. The morphing between f and g is a continuous deformation from f to g. The image f is called source image and g is called target image. If O_l, $0 \leq l \leq T$ is a morphing sequence of images from the image f to the image g. Then each image O_l is a blending of an image from the source image and an image form target image. In our case the blending is a linear interpolation.

To create the morphing between f and g, constructing a sequence of images C_l, $0 \leq l \leq T$ in this form: the user marks the source points over the source image f and marks the keyframe over the target image g. Then using the optimizer method to find the forces that drive the source points to the keyframe. With this forces the simulation is run and in each step time l the warping image C_l is saved. The sequence C_l, $0 \leq l \leq T$ is a *forward warping*.

From the forward warping it is possible to build another sequence $\widehat{C}_l$ from C_l of images where each $\widehat{C}_l$ is a warping of g. This sequence constitute the *inverse warping*. To create the inverse warping the following method is used. This system is by steps. The first step is to make a warping of g using C_{T-1} as target image and the image result is $\widehat{C}_1$. The second step is a warping of $\widehat{C}_1$ using C_{T-2} as a target image, the result is the image $\widehat{C}_2$, and so on.

Then O_l is a blending of $\widehat{C}_l$ and C_{T-l}. This method is used to create the *inverse warping* because the researcher consider that is more simple that others. The development of a different technique is a future work.

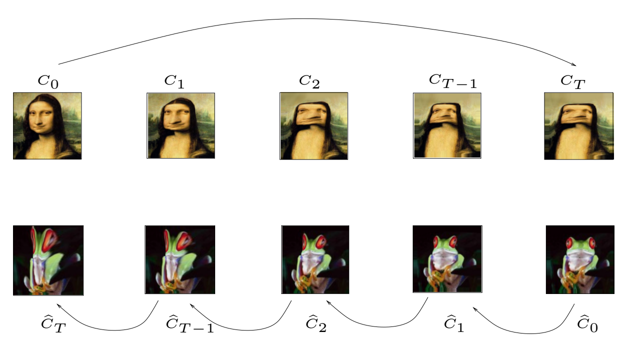


Fig. 3. Inverse Warping

VII. Results and Examples

The warping technique with fluids has been used to create animated image with a natural effect. Figures 4, 5, 6 show three examples of this image. Indirect control has been used to obtain this result In the image of frog the forces are located in the region of the neck and use viscosity here. The forces oscillate during the simulation, because the forces are multiples of the function sin that depends of the time. These forces have vertical direction.

For the example of the nose, see Figure 5, obliques forces was used located in the region of the nose that they oscillate. For the laugh, see Figure 6 we use four group of vertical functions. Two are located in the shoulders and two in the cheeks.

The resolution of the image 600×600. The number of cells used in the fluid simulation is 64×64. The number of forces are 1936. The result is in real time.

For the morphing we use the differential control to create a sequence of animation like the one in Figure 7. For the forward warping are located 14 source points over the Mona Lisa and the corresponding target points over the frog.

It was placed 500 control parameters over the image. The simulation used 10 timesteps and then it was saved 10 images. The simulation produce very close points the keyframe. The simulation spend 5 minutes.

For the inverse warping the same quantity of control forces were used. For each step of the inverse warping the simulation used 5 timesteps and each step spend 3 minutes.

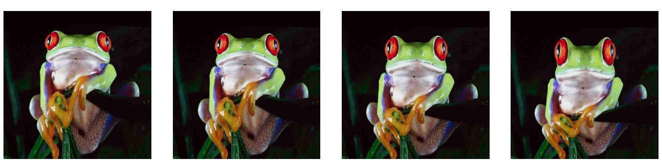


Fig. 4. Frog Animation. From image we create an animation, using oscillator force applied on the fluid defined on the domain image.

Fig. 5. Here an animation of a picture of Oliver Hardy. We move the mouth and nose with a oscillator force in the diagonal direction, centered on the mouth.

Fig. 6. Laugh

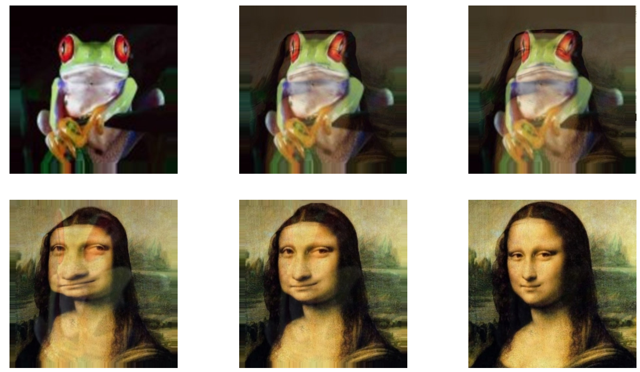


Fig. 7. Morphing sequence.

VIII. Conclusions and Future Work

This study has introduced a technique of warping and morphing using fluids. The adjoint method was used to compute derivatives efficiently. Defining keyframe using point to control the fluid. And this make possible and more accurate control of the image deformation using fluids. Moreover a most robust objective function used in differential control of fluid has been defined. Using a simple blend and inverse warping.

References

[1] C. Csuri, R. Hackathorn, R. Parent, W. Carlson, and M. Howard, "Towards an interactive high visual complexity animation system," *SIGGRAPH Comput. Graph.*, vol. 13, no. 2, pp. 289–299, Aug. 1979.

[2] J. Stam, "Stable fluids," *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, pp. 121–128, August 1999.

[3] "From me to you," http://fromme-toyou.tumblr.com/, 2012, [Online; accessed 13-May-2012].

[4] G. Jonas, D. L, B. Costa, and L. Velho, *Warping and Morphing of Graphical Objects*. Morgan Kaufmann Publ., 1999.

[5] A. Barr, "Global and local deformations of solid primitives." *In Proceedings of SIGGRAPH*, 1984.

[6] N. Arad and D. Reisfeld, "Image warping using few anchor points and radial functions," *Computer Graphics Forum*, vol. 14, no. 1, pp. 35–46, 1995.

[7] A. R. Smith, "Planar 2-pass texture mapping and warping," *In Proceedings of SIGGRAPH*, 1987.

[8] N. Foster and D. Metaxas, "Realistic animation of liquids," *Graph. Models Image Process.*, vol. 58, no. 5, pp. 471–483, 1996.

[9] ——, "Modeling the motion of a hot, turbulent gas," in *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 181–188.

[10] N. Foster and R. Fedkiw, "Practical animation of liquids," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 2001, pp. 23–30.

[11] D. Enright, S. Marschner, and R. Fedkiw, "Animation and rendering of complex water surfaces," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 736–744, 2002.

[12] A. Treuille, A. McNamara, Z. Popović, and J. Stam, "Keyframe control of smoke simulations," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 716–723, 2003.

[13] M. B. Giles and N. A. Pierce, "An introduction to the adjoint approach to design," *Flow, Turbulence and Combustion*, vol. 65, pp. 393–415, 2000.

[14] R. Giering and T. Kaminski, "Recipes for adjoint code construction," *ACM Transactions on Mathematical Software*, vol. 24, no. 4, pp. 437–474, 1998.

[15] B. M., B. A., and S. G., "Navier-stokes fluid dynamics and image and video inpainting." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, pp. 9–14, December 2001.

[16] D. Bonilla, L. Velho, A. Nachbin, and L. G. Nonato, "Fluid warping," in *Proceedings of the IV Iberoamerican Symposium in Computer Graphics*, Sociedad Venezolana de Computación Gráfica. DJ Editores, C.A., June 2009.

[17] D. Bonilla and L. Velho, "Control methods for fluid-based image warping," in *Proceedings of XXIV Sibgrapi Conference on Graphics, Patterns and Images*. Institute of Mathematics of the Universidade Federal de Alagoas, August 2011.

[18] A. McNamara, A. Treuille, Z. Popović, and J. Stam, "Fluid control using the adjoint method," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 449–456, 2004.

[19] S. G. Johnson, "Notes on adjoint methods."

[20] C. Wojtan, P. J. Mucha, and G. Turk, "Keyframe control of complex particle systems using the adjoint method," in *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 15–23.

[21] R. Fedkiw, J. Stam, and H. W. Jensen, "Visual simulation of smoke," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 2001, pp. 15–22.

[22] H. Birkholz and D. Jackel, "Image warping with feature curves," *In Proceedings of SIGGRAPH*, pp. 199–202, 2003.

[23] A. J. Chorin and J. E. Marsden, *A mathematical Introduction to Fluid Mechanics*. Springer-Verlag, 1993.

[24] J. Gomes and L. Velho, *Image Processing for Computer Graphics*. Springer Verlag, 1997.

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

[26] J. Stam, "Flows on surfaces of arbitrary topology," *ACM Transactions On Graphics (TOG), Proceedings of SIGGRAPH*, pp. 724–731, July 2003.

[27] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*. CRC Press, 1999.

[28] R. Courant, E. Isaacson, and M. Rees, "On the solution of nonlinear hyperbolic differential equations by finite differences," *Communications on Pure and Applied Mathematics*, vol. 5, pp. 243–255, 1953.

[29] T. Beier and S. Neely, "Feature-based image metamorphosis." *SIGGRAPH Comput.*, vol. 2, no. 26, pp. 35–42, July 1992.

[30] L. S, W. G, C. K-Y, and Y. S. S, "Image metamorphosis with scattered feature constraints," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 4, pp. 337–354, December 1996.

[31] A. Nachbin, *Notas do Curso: Dinâmica dos Fluidos*, 2006.

[32] S. S, McPhail.T, and W. J, "Image deformation using moving least squares," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, July 2006.

[33] A. R. Smith, "Planar 2-pass texture mapping and warping," *In Proceedings of SIGGRAPH*, 1987.

[34] D. B. Smithe, "A two-pass mesh warping algorithm for object transformation and image interpolation." *Technical memo, Industrial Light and Magic*, 1990.

[35] G. Wolberg, "Skeleton based image warping," *Visual Computer*, vol. 5, no. 1/2, pp. 95–108, March 1989.

[36] ——, "Digital image warping." *IEEE Computer Society Press, Los Alamitos, CA.*, 1990.