Enhanced Target Driven Smoke Morphing

Marcelo Caniato Renhe*, Antonio Oliveira†, Cláudio Esperança† and Ricardo Marroquim†
*Federal University of Juiz de Fora
marcelo.caniato@gmail.com
† COPPE/ Federal University of Rio de Janeiro
oliveira, esperanca,marroquim@lcg.ufrj.br

Fig. 1. Example of a transformation into a multi-component target

***Abstract*—Target driven smoke morphings are processes, based on the simulation of smoke dynamics, which are so flexible that we are induced to believe that they can produce a continuous transformation of any 2D object into any other. Here they are considered with respect to the ability of reproducing targets of more complex geometry or topology, assuming that the process starts at a generic 2D object. Alternatives for several limitations are proposed to make it possible to cope with fine dents, cavities or holes and to obtain the same brightness in all components of a non-connected target object. The original approach uses a diffusion process to define a field that pulls smoke towards the target. Here that field is replaced by one, depending on a distance transform, which is computationally simpler and helps to deal with peculiarities of the objects. The finalization of the morphing process is also focused with the objective of eliminating oscillations and flickering.**

***Keywords*-Morphing, Navier-Stokes equation, Smoke models, Distance Tranform**

I. Introduction

Methods for obtaining visually convincing fluid dynamics simulations have been an important topic of Computer Graphics research for at least the last two decades. The increase in hardware capabilities has allowed the treatment of configurations of growing complexity, and thus have maintained a high level of interest in the area. Moreover, fluid diffusion models have been used for implementing tasks beyond the context of simulation. A remarkable example of this is morphing, which contradicts physical laws. Among the morphing approaches based on fluid dynamics, those not steered by keyframes have scarce examples in the literature and, in consequence, their application to the morphing between two planar shapes still have aspects not entirely understood. This article focuses some of these aspects, giving them specialized treatments so as to produce visually acceptable transformations which are more correct in the sense of fully reaching the target set geometry. We aim to show that by adequately treating topologycal and geometrical aspects of the target and also the way smoke density is distributed and rendered, a reasonable result can be obtained in a totally automatic way, that is, without manually stipulating keyframes or matched pairs.

Smoke presents highly variable densities, transparency and shape. For that reason, it is a natural medium for forming the objects being morphed. Standard schemes for obtaining a morphing employ some kind of warping, followed by a blending procedure which merges characteristics of both the input and target objects. The smoke morphings focused here do not follow that model. There is no warping, and the blending of features is restricted to displaying part of the target while portions of the input object are still recognizable. They have an evolution paradigm ideally consisting of the gradual dilution of the input, the creation of a smoke cloud that hides the rearrangement of structures and the materialization of the target when smoke is totally dissipated.

Some previous works on the theme are mainly interested in showing the ability of physically modeling the transformation. In this article, the questions considered are related to different issues of the process. In fact, its main contributions are relative to problems of other kind like reproducing the geometrical and topological features of the target at the end of the morphing, generating a final density distribution having the same value at every target component and avoiding noticeable oscillations in the convergence process to the target.

We start by overviewing related works in the next section. Section III is dedicated to the generic formulation of smoke

morphing as established in [1]. In section IV we discuss several aspects of a smoke morphing process which demands specific treatment. Implementation and demonstrative examples are the scope of section V. Conclusion and future work constitute section VI.

II. Related work

Several works in the literature have already addressed diffusion based morphing or similar topics, using a variety of methodologies. The first ones to present a method for the generation of shapes through a controlled simulation of smoke were Treuille and McNamara [2]. The smoke control was done through a series of keyframe images defined by the application user. An objective function measuring how well states of simulation match keyframes is optimized to adequately determine control variables affecting forces and velocities. This scheme was revisited in [3] and more recently in [4] and [5].

The work by Treuille and McNamara is based, like many others involving fluid simulation, on the results obtained by Foster and Metaxas [6], Stam, in his Stable Fluids publication [7], and Fedkiw, in a subsequent work [8]. Stam presents an unconditionally stable method for fluid simulation, adopting a semi-Lagrangian approach for the advection component in the system.

Fattal and Lischinski present an alternative proposal for the morphing of shapes using physical animation of fluids [1]. Differently from the previous approach, the authors do not use any intermediate keyframe to steer the simulation. Only the initial and final shapes are given. The transformation process is carried out through the application of a driving force to the fluid, which directs it to the target. It should be noted that the approach only concerns itself with whether or not the final stage has been reached, but not in the aspect of the smoke in the middle stages of the morphing. Even such a flexible scheme has difficulties for opening holes or excavate cavities Obtaining uniform brightness in a multi-componnent target can also be a problem. An implementation based in this work with results is described in [9].

Other morphing approaches can be found in the literature, some of them using diffusion processes, while others eschew fluid based techniques. Liu *et al.* use a model consisting of ellipsoidal structures (blobs) to approximate 3D meshes ([10] and [11]). Their technique morphs objects by changing their blob approximations through the conjugated application of two operations affecting clouds, which are, essentialy, sets of blobs. One is a diffusion scheme affecting each cloud individually, the other is an aggregating phase which merges several clouds into a more complex one.

Zhao [12] uses a scheme typically employed in fluid dynamics, known as Lattice Boltzmann, or LBM for short. He introduces a modified version of the LBM algorithm where distance fields relative to surfaces influence the evolution process. More specifically, the difference between the distance field of the target surface and that of the transformed surface is used to define the expansion speed of the morphing process. This speed is included in the LBM diffusion scheme as an external force.

Schechtman *et al.* introduce yet another approach for image morphing [13]. He uses an optimization algorithm where the objective function is a bidirectional similarity measure. The idea is to find a smooth transition between the intermediate stages of the morphing while, at the same time, making sure that the regions of an in-between image is, in some way, similar either to the original or to the target image.

This work is based on the approach adopted by Fattal and Lischinski [1] for the morphing of images using a smoke diffusion process. However, instead of using a Gaussian function, the fluid driving force is associated with a special distance function to the target. The definition of that function and the way the driving force is obtained from it is chosen so that density concentration and convergence oscillations are avoided. The gathering term used in the work by [1] was also replaced by a series of mechanisms designed to work around the issues found during the implementation of the method. By using this approach, we can achieve good results in several cases in where the original algorithm fails. Such cases include the generation of objects with holes and narrow cavities, or those with more than one connected component.

III. Original problem formulation

Following the formulation used in [1], let $\rho(\mathbf{x}, t)$ be a scalar field representing the density of the smoke at a given point $\mathbf{x}$ of a grid at time t. For simplicity, we write ρ to refer to the current state of that field during the morphing process, while ρ_0 and ρ^* denote the fields given by the characteristic functions of the source and target sets, respectively. The morphing will be interpreted as a process transforming field ρ_0 into ρ^*. Thus, ρ varies according to the standard Navier-Stokes system for Newtonian fluids with variable densities [1], expressed by the equations below:

$$\frac{\partial V}{\partial t} = -V \cdot \nabla V - \frac{\nabla P}{\rho} + \sigma \nabla^2 V + g_{\text{targ}} + g_{\text{damp}} \quad (1)$$

$$\frac{\partial \rho}{\partial t} = -V \cdot \nabla \rho + \sigma_\rho \nabla^2 \rho + h_{\text{damp}} \quad (2)$$

Expression 1 updates velocity(V). The terms on its right side refer to the velocity advection, pressure(P) action under uncompressibility conditions or not, a viscosity influence weighted by σ, the acceleration (g_{targ}) determined by a force that pulls toward the target and a last term (g_{damp}) introduced to eliminate residual velocities at the end of the process producing a static solution. Equation 2 expresses how density is regulated by the action of advection, viscosity and a damping term. In our implementation we have used Stam's formulation [7] that uses cell velocities instead of staggered grids. The crucial point is the definition of the force which induces g_{Target}. In [1], that force is obtained from a Gaussian

[1] A detailed introduction to Navier-Stokes equations can be found in [14]

diffusion process applied to both current and target objects. Explicitly, it is expressed by equation 3

$$\mathbf{F}(\rho, \rho^*) = \tilde{\rho}\frac{\nabla\tilde{\rho}^*}{\tilde{\rho}^*}, \quad (3)$$

where $\tilde{\rho}$ and $\tilde{\rho}^*$ are fields obtained by convolving ρ and ρ^* with a Gaussian kernel.

This formulation focuses specially the fact that once the current object has converged to the target, then the Navier-Stokes' regulating process must be able to bring all the velocities to zero. Finalization problems are thus avoided. This is provided in the following way: when $\rho = \rho^*$, then the target driving field becomes $\mathbf{F}(\rho^*, \rho^*) = \nabla\tilde{\rho}^*$, which is the gradient of a potential field. Thus, it can be compensated by a pressure field equal to $\tilde{\rho}$. Velocities, then, only change by advection and are continuously reduced by a damping force until they reach zero.

Considering a circular target, for instance, it is clear that the acceleration determined by 3 increases with the distance to it. For non-convex targets, however, this may not hold. The formulation above requires a new diffusion process at each iteration, to find $\tilde{\rho}$, which must use the same parameters as the one employed to generate $\tilde{\rho}^*$. Otherwise the two fields will not be equal at the final state.

The process used for $\tilde{\rho}^*$, however, cannot lower the target densities too much to avoid eliminating fine geometrical features. This not only makes it unappropriate to convolve the image with a Gaussian kernel repeated times, but also requires the Gaussian variance to be small. Moreover, the result of a low variance Gaussian filter applied to the whole image far from the target, will be a sum of extremely small numbers. Thus, robust numerical methods and high precision are necessary, considering that the initial object can be, in principle, at any distance from the target.

IV. Proposed approach

A. Redefining the driving force

To avoid the computational peculiarities mentioned above and to get a more suitable approach to deal with complex shapes, we propose a driving force based on a distance transform rather than on a Gaussian diffusion. Let us define T as the signed distance function with respect to the boundary of the set of points $\tau = \{\mathbf{x}|\rho^*(\mathbf{x}) = 1\}$. Note that T is differentiable everywhere except on $MA(\tau)$, the medial axis of τ. Disconsidering for now the points in $MA(\tau)$, we propose the following expression for the driving force component:

$$H = \frac{\rho}{\tilde{\rho}^*}[\tilde{\rho}^*\tilde{T} + (1-\tilde{\rho}^*)((1-\alpha)(1-\tilde{T}) + \alpha\tilde{T})]\nabla T, \quad (4)$$

where $\tilde{\rho}^*$ is a nowhere null continuous approximation of ρ^*, such that $\tilde{\rho}^* = 1$ in (τ), $\tilde{T}$ is a normalized version of T so that the maximum value of $|T|$ is 1, and α is a measure in $[0, 1]$ of how far the current iteration is from the final desired state. In our implementation it is given by

$$\alpha \triangleq \frac{\sum_{p\in I}(\tilde{\rho}^*(p)(\tilde{\rho}(p))}{\text{target area}}.$$

First of all, notice that for a point in the target, $H \to \tilde{T}\nabla T$ when the morphing gets closer to the final state where $\rho = \rho^*$. At the same time, $H \to 0$ outside the target. Putting the two cases together, we obtain that $H \to \nabla(G)$, i.e., where G is the scalar field

$$G \triangleq \frac{\tilde{T}\min(T, 0)}{2}.$$

Thus, a pressure field equal to G can compensate H.

Far from τ, $\tilde{\rho}^* \simeq 0$ and H decreases with T at the beginning of the morphing process, when α is small, and increases with T at the end, when $\alpha \simeq 1$. This causes a stretching of the source towards the target at the start and aggregates small amounts of smoke dispersed on the whole image while they are pulled towards the target at the end of the process. To make the acceleration determined by H be a bounded function, we must limit the ratio between $\max((1-\tilde{T}), \tilde{T})$ and ρ^*. This can be ensured, for instance, by making $\rho^* = 1 - \frac{\tilde{T}}{2}$. Observe that at any given iteration, we simply update α, current densities and velocities. In particular, there is with no need for computing costly Gaussian filters.

However, a morphing process driven by distance measures has the drawback that, during the whole transformation, only the portion of the target which is closest to the object being morphed influences the way it changes. This issue is tackled in two different ways. The first consists in adding to the field force a component proportional to the distance vector between the geometrical centers of the current morphing instance and of the target. The translation determined by this vector anticipates the moment the morphing object reaches the target, thus reducing the possibility that an object being morphed is influenced solely by the points of the target closest to it. Secondly, the concentration of density around the convex vertices of the border must be handled with care, since these attract a much larger number points than those in a smooth portion of the boundary. To distribute the smoke density along the target border, when the morphing process reaches it, we need to redefine the distance transform. Consider a discretized version of the problem, where the image domain is represented by a rectangular grid of cells, each one with a label obtained according to following procedure:

1) Label a cell with 0 if it lies on the target border, or with ∞ otherwise.
2) For all cells C such that $label(C) = \infty$, relabel them with $\min_{C'\in N(C)}(label(C') + 1)$, where $N(C)$ stands for the 8-connected neighborhood of C.
3) Repeat (2) until all cells have finite labels.
4) Define X_i as the set of cells with label i.

Once this is done, associate each $C \in X_i$ with one of its neighbors in X_{i-1}. The graph thus defined is a forest with roots in X_0. Different associations imply different forests, and to construct one where the pixels are well distributed among its trees, procedure *Redistribution* given below must be applied for every X_i, starting with X_0. Figure 2 shows an example of this process for a square object.

Procedure *Redistribution*:

1) Traverse X_i, adding to X_{i+1} every neighbor of a cell in X_i still unassociated and removing from X_i all cells with no unassociated neighbors. At the end of that process, if X_{i+1} is empty, stop. Otherwise, the remaining cells of X_i will form a set of mutually disjoint closed chains. For each chain S of X_i, let L_S be the list of cells $C \in S$ ordered by the value of $D(C)$, where $D(C)$ is defined as the number of cells in the tree which contains C.
2) For each cell C of X_{i+1} having a single neighbor C' in X_i, associate C' to C.
3) Make associations for aligned groups of 3 cells. This is done using a priority queue organized by $D(C)$. Each cell C of a chain S of X_i is tested as to whether it is aligned (i.e., has same row or column) with its parent C^* in X_{i-1} and an unassociated neighbor C'. In this case, an association between C and C' is created and $D(C)$ is updated in the priority queue.
4) Make associations for aligned groups of 2 cells. This is done in a similar manner to step 3, except that the parent C^* is not taken into account.
5) Finally, repeat the process for remaining cells still having an unassociated neighbor in X_{i+1}.

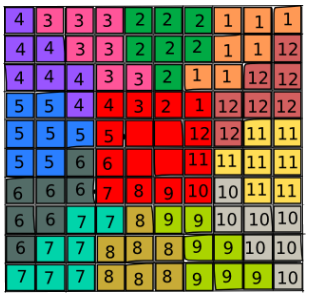

Fig. 2. The modified Distance Transform of the square in red. Each cell is labeled with a number identifying the root of the tree it belongs to.

Notice that the algorithm forces a preference for aligned sequences of parent, child and grandchild cells so as to reduce the chance of having jagged tree paths. The preference for aligned groups of two cells avoid branches intersecting each other.

Finally, this discrete formulation entails that ∇T at a cell C belonging to a tree rooted at a cell C_0 is defined as the unit vector of $\overrightarrow{CC_0}$. T is replaced by the norm of $\overrightarrow{CC_0}$.

B. Average density evolution

Both source and target consist ot sets of cells having density 1 and, as densities are simply advected, the total ammount of smoke will remain constant during the transformation, except for the portions that eventually leave the image. These two facts imply that, in spite of the system's ability to transport densities to the target, a transformation between objects of different sizes cannot be totally completed.

We tackled this problem by estimating, for each simulation step t, an intended total density $D^*(t) = \beta(t)D_1 + (1 - \beta(t))D_0$, where D_0 and D_1 are the sum of all densities at the initial and final configurations, respectively. Then, an uniform correction factor is applied to all the grid cells, forcing $D(t)$, the total density computed at step t to become $D^*(t)$. Eventually, some cells - n, say - might get densities greater than one. This is corrected by distributing the surplus densities among the other cells. In our experiments, this process has not introduced noticeable artifacts or degraded the morphing in our experiments.

The linear interpolation weight $\beta(t)$ must express the proximity between the configuration at t and the target. Let $R(t)$ be the ratio between the sum of the densities inside the target and the total density, both evaluated at step t. As a first attempt, we may define

$$\beta_1(t) = \frac{\max(R(t) - R(0), 0)}{1 - R(0)},$$

which is a number in $[0, 1]$. This expression reduces to $R(t)$ if target and source are disjoint. With this formulation, however, $\beta_1(t)$ is kept null until the target is reached. To correct this, let $\delta_0(t)$ refer to the distance between the geometrical centers of the instance at t and that of the source object. Define $\delta_1(t)$ similarly with respect to the target object. Thus, the actual factor used in our implementation is given by

$$\beta(t) = k\beta_1(t) + (1 - k)\frac{\delta_0(t)}{\delta_0(t) + \delta_1(t)},$$

for some $k \in [0, 1]$.

C. Targets with more than one connected component

Within the target, the driving force makes the smoke oscilate from one side to the other before velocity gradually decreases in value until the morphing reaches the final state. In consequence, density propagation to unexplored parts of the target can be predominantly determined by the action of viscosity, which is implemented by a Laplacian operator. That operator performs the uniformization necessary to equalize density distribution inside the target. That uniformization, however, is a slow process, which led us to apply it consecutively n times during a single iteration. A good choice for n is case dependent. It must be large enough to sufficiently speed up the uniformization process to avoid discontinuities in density evolution rhythm when the predominant factor for it changes from advection to viscosity. On the other hand, it cannot be too high to cause noticeable jumps between successive frames.

When dealing with connected targets, the Laplacian operator works quite well. However, if shapes have more than one connected component, a new problem arises. In most cases, density converges to a different value for each component, meaning that, for some, that limit is less than 1. To compensate for that, other components would exhibit densities greater than 1. The average density correction discussed in the previous section avoids this behaviour, but convergence suffers, since densities are redistributed outside the target.

To fix this problem, it is necessary to take energy from one component to another, which implies in moving densities from

a target component to the surrounding background region. The system of forces in use, however, hinders that.

A solution then consists in replacing the Laplacian operator, which in essence computes an average of neighbor cell densities, by an operator – called *MaxViz* – which takes the maximum value, when processing cells inside the target area. This worked well in all of our tests, where the advection process took sufficiently large density values from the initial object to any target component. When that happens, the *MaxViz* operator propagates densities to all of the extension of the component.Simultaneously, average density treatment will increase their values. Figure 1 shows an example of a successful morphing with a target with multiple components.

One might imagine, however, situations where some isolated component will attain a very small overall density. In such cases, *MaxViz* might be combined with some forward advection scheme, making it possible for cells to accumulate density. This is a good example where the physical model must be relaxed to fill the requirements of a good morphing.

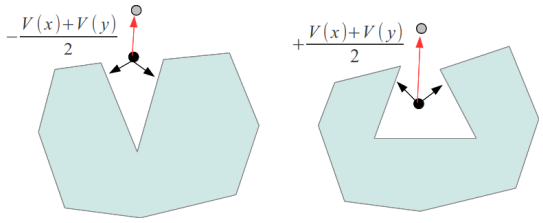

Fig. 3. Advection scheme for pulling low densities towards dents and cavities

D. Density versus brightness

As the density value is the only information relative to the ammount of smoke at a cell, it is natural to think of rendering it by using brightness as a linear function of its density. During the course of thesimulation, however, many cells may assume density values which are insufficient to make them visible. Using a function of higher slope to overcome this problem would saturate the representation of cells with medium densities, thus ruining the visual aspect of the transformation.

Detaching density from brightness, not only allows to address the difficulty above, but may also produce desired visual effects. For example, we could present the target shrouded in a mist, or show the transforming object as if it were more transparent than it really is. In order to achieve these effects and provide correction for cases in which important cells may not be visible, we used a non-linear function to amplify the brightness for densities below some threshold d_0. In our experiments, the brightness of $B(d)$ of a cell with density d is set to

$$B(d) = \begin{cases} cd + \lambda(d_0 - d)^2 \left[1 - \frac{(d_0-d)^2}{d_0^2}\right], & \text{if } d < d_0 \\ cd, & \text{if } d \geq d_0 \end{cases} \quad (5)$$

where λ is chosen to ensure that B is an increasing function.

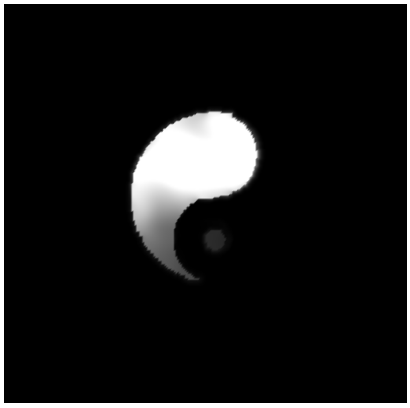

Fig. 4. Difficulties with smoke transportation may cause insufficient density in some connected components and holes which are full of smoke

E. Dents and holes

One of the drawbacks of diffusion processes is that they eliminate high frequencies, making it difficult to produce, for instance, a narrow dent in the target contour. Besides, it is not easy to open a hole in the target after it has been totally filled with smoke. If we entrust this task to the advection step of the simulation, we will not be able to obtain any good results, because the cells in the hole will swap densities with cells on the other side of its medial axis without ever reducing the total amount of smoke inside it.

That exchange of densities may be reduced by using a multi-stage advection method such as, for instance, those proposed by Lax-Wendrof and McCormack [15]. In fact, it is sufficient to modify the standard semi-Lagrangean advection scheme so that it updates the density $\rho(x)$ at a point x with the value of

$$\min\{\rho(x - \frac{V(x)+V(y)}{2}\Delta t), \rho(x + \frac{V(x)+V(y)}{2}\Delta t)\} \quad (6)$$

where $y = x - V(x) \times \Delta t$. That change must be made only for $x \notin \tau$ and when the angle between $V(x)$ and $V(y)$ is $\geq a$, for some angle $a \in (\frac{\pi}{2}, \pi)$. That new formulation enlarges the velocity component parallel to $MA(\tau)$ favoring the displacement of smoke inside the dent. The $\min$ operator aims to ensure that null densities in the already excavated part migrate to the portion of the dent still filled with smoke by distinguishing between dents and cavities having openings narrower than their interiors (see Figure 3).

After a certain time, the field force will lower all the densities out of the target's convex hull. This is sufficient to trigger the excavation process of structures such as dents, which are connected to the exterior (see Figure 5 for an example). This, unfortunately, is not the case of a hole. If it is full of smoke, the advection will not be able to find low densities nearby to clear it. In Figure 4, for instance, the hole in the "yin-yang" symbol will remain filled. Thus, to start the opening process we must force the clearance of some cell within the hole. Since any hole has at least one local maximum of the distance transform of the target ($DT(\tau)$), some such cells are selected to start an excavation

TABLE I
DENSITY DISPERSION

Example	Cell Average Density
Psi-omega	0.2528
Omega-psi	0.2157
Yin-yang	0.6173
2012-Sibgrapi	0.3715

process. Rather than explicitly identifying holes, if the rhythm of density evolution becomes too slow, we simply set the density of the k highest local maxima of $DT(\tau)$ to zero. If this is not sufficient to start an opening process that recovers the evolution rhythm, replace every local maximum already selected by its unselected neighbor of highest $DT(\tau)$ and repeat the process. We use the variation of $R(t)$ defined in IV-B during a few iterations to evaluate the morphing rhythm. Figure 7 shows an example of this hole opening process.

F. Flickering elimination and antialiasing procedures

Even after the smoke distribution has visually converged to the target model, a flickering effect may occur, usually constrained to a neighborhood of the target border, and too narrow to give the target a steamy appearance which is more desireable. The flickering is just the result of large velocity values that keep trying to move very small densities towards the target. This effect may induce the observer to think that the morphing process is not yet finished.

Although antialiasing procedures can reduce flickering intensity, to produce a stable final picture, we adopt the simple solution of gradually reducing the time step while the simulation gets closer the final state. Near the end of the process δt becomes very small, producing no visually noticeable oscillations. Before that, however, the time interval need not be reduced, suggesting a decay at least quadratic. We regulated the time step by

$$\Delta t^* = \Delta t(1 - \beta(t)^2) + \epsilon\beta(t)^2,$$

where $\beta(t)$ was defined in Section IV-B, ϵ is a small number and Δt is the non-adjusted time step.

To give a steamy aspect to the target, the small densities around the target are propagated nearby and represented through a scheme that enhances the brightness of low densities. Table I shows the average density of the cells where it is not null at an instance visually equivalent to the target image in four different examples. If all these cells were in the target these densities would be close to 1. As they are much lower,this indicate that there is activity in a large number of cells outside the target. Stopping that activity by means of a damping force may take some time, justifying the finalization approach used here.

At the beginning and at the end of the transformation process, when constrast is sharper and the object seems to remain still, aliasing is much more noticeable. Non-smooth shapes and low resolution grids aggravate this propblem. During the transformation, however, the mist cloud that usually surrounds the object attenuates this effect. We have treated aliasing by applying 4×4 versions of Catmull's [16] and Lanczos's [17] filters. This last one is well known by its effective aliasing reduction, while maintaining the contrast and collaborating with the removal of oscillations. It is, however, more costly, and in the tested simulations we have restricted its application to a narrow band around the object borders

V. RESULTS

A program integrating the proposed morphing approach and a visualizer has been implemented using the *C++* language and the OpenGL graphics library. Experiments were run on a computer with an Intel i3 processor running at 2.1GHz, 4 Gb memory and an nVidia GeForce 540M graphics adapter. The program equalizes the resolution used to represent the objects to be morphed and puts them, after adequately scaled, at places specified by the user. The transformation is produced at frame rates which are essentially proportional to the grid sizes, although the complexity of the field force and the variable time cost of the Poisson solver may also influence the overall speed. In particular, simulations on a 64×64 grid run at roughly 60 fps, while simulations on a 256×256 grid run at close to 6 fps. The duration of the morphing process also depends on many other factors which certainly include the position of the objects, their geometrical difference and topologies. The simulation complexity also depends on parameters such as the number of times the Laplacian operator is applied within the target at each iteration.

The system does not need a strict termination rule. Rather, the time step is successively lowered when most of the density reaches the target object. The chart in Figure 6 shows a typical evolution of the transformation when object and target are disjoint sets, namely, the morphing between two greek letters shown in Figure 8. We remark that from the 80th iteration on, the result is visually equivalent to the final state, though the chart still indicates a small difference.

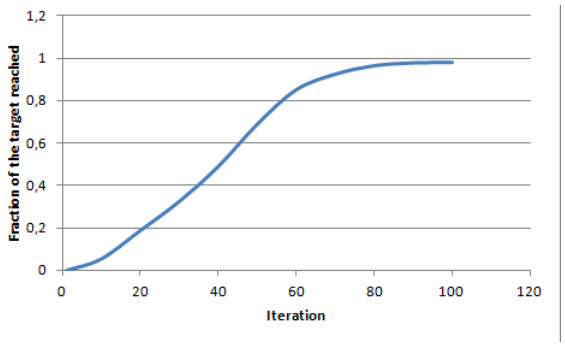

Fig. 6. Portion of the target reached $\times$ number of iterations

The figures below show four examples of morphings produced by the method introduced here. In all these examples resolution is 256×256, time step is 0.2 and the driving force weight is 1. Figure 8 is the transformation of greek letter psi into letter omega, presented here for comparison purposes, since this is also shown in Fattal's original paper. Figure 9 is a simple example where both objects have holes and multiple

Fig. 5. Example of the excavation of target with dents

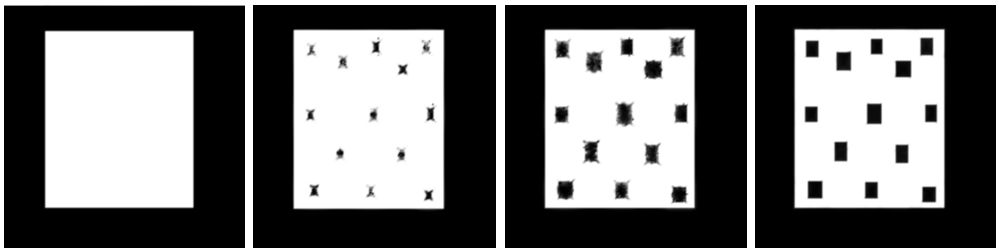

Fig. 7. Example of the opening of holes in a target

components. Figure 10 shows an object with thin features and dents morphing into a target with holes. Finally, Figure 11 presents a sequence of consecutive morphings involving different objects, similar to those used in the works of Fattal [1] and Fedkiw [8].

VI. Limitations

The methodology described herein is, in essence, a fluid simulation performed in an Eulerian environment. Thus, it inherits all limitations of such methods. These are related to the discretization process and to the fact that velocity and density variation are not computed simultaneously. Compared with other approaches, target driven smoke morphing does not allow for direct control over correspondences between parts of the two objects. The most usual form of blending process is the simultaneous exibition of features of both objects. An extension for transforming objects with color is far from straightforward, due to the difficulty to keep parts of different colors separated. In relation to the original approach, we have replaced an atractor field which is differentiable everywhere by one with no gradient at $MA(\tau)$. So, a pressure field cannot adequately compensate the atractor effect at the points on that set. That compensation was used to accelerate the convergence to the final state. However, since $MA(\tau)$ has no interior, losing that property along it does not affect the duration of the morphing process as shown in Table I.

VII. Conclusions and future work

In this article, target-driven smoke morphings are analyzed with respect to their geometrical and topological correction, and to their visual aspect. Section IV discusses techniques for the treatment of several issues which are neglected by other works, namely, the handling of multiple connected components, dents and holes. Also, by changing the field force to one based on the special distance transform introduced in Section IV-A, we enhance the computational performance without degrading the morphing quality.

The continuation of this work will evolve in a few predictable directions. In the context of smoke modeling, the next step is to introduce temperature and soot to help to control brightness and transparency during the morphing. The extension to 3D objects is a natural upgrade. The main challenge is to maintain quasi-real time performance, which must be achieved by means of parallelization. Incidentally, writing GPU versions of almost all procedures implementing the ideas proposed here seems quite straightforward. Applying the scheme to colored objects is another challenge. Possible trends to be followed in that direction include partitioning the objects in regions of similar color, pre-compute source-target cells correspondence, possibly with the user's intervention, and the use of multi-phase fluid approaches to try to avoid the merging of different color regions.

References

[1] R.Fattal and D.Lisckhinski, "Target driven smoke animation," *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 23, no. 3, pp. 441–448, 2004.

[2] Z. P. A. Treuille, A. McNamara and J. Stam, "Keyframe control of smoke simulations," *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 22, no. 3, pp. 716–723, 2003.

[3] Z. P. A. McNamara, A. Treuille and J. Stam, "Fluid control using the adjoint method," *ACM Trans. Graph. (SIGGRAPH Proc.)*, vol. 23, no. 3, pp. 449–456, 2003.

[4] A. N. D. Bonilla, L. Velho and L. Nonato, "Fluid warping," in *Proceedings of the IV Iberoamerican Symposium in Computer Graphics*, 2009.

[5] D. Bonilla and L. Velho, "Control methods for fluid-based image warping," in *Proceedings of WTD- Workshop of Theses and Dissertations - SIBGRAPI 2011*, 2011.

[6] N. Foster and D. Metaxas, "Modelling the motion of a hot, turbulent gas," *Computer Graphics Proceedings Annual Conference Series*, pp. 181–191, 1997.

Fig. 8. Example of a transformation between two greek leters. In addition to the standard driving force given by equation 4, another component, pointed to the target center of mass(TCM), has been added. This last force values 0.3 of the distance to TCM.

Fig. 9. Example of a transformation between objects with two connected components and holes

Fig. 10. Example of a transformation between an source with dents and a target with holes

Fig. 11. A sequence of several Smoke Morphings

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

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

[9] J. Fustin and Z. Mihajlovic, "Fluid flow animation," . [Online]. Available: http://bib.irb.hr/datoteka/252360.FluidFlowAnimation.pdf

[10] C. C. L. W. S. Liu, X. Jin and K. Hui, "Ellipsoidal-blob approximation of 3d models and its applications," *Computers and Graphics*, vol. 31, no. 2, pp. 243–251, 2007.

[11] C. C. L. W. J. F. X. Jin, S. Liu and H. Sun, "Blob-based liquid morphing," *Computer Animation and Virtual Worlds*, vol. 16, no. 3, pp. 391–403, 2005.

[12] Y. Zhao, "Gpu-accelerated surface denoising and morphing with lattice boltzmann scheme." in *Proc. of Shape Modeling International/2008*. IEEE, 2008, pp. 19–28. [Online]. Available: http://dblp.uni-trier.de/db/conf/smi/smi2008.html#Zhao08

[13] M. I. E Shechtman, A Rav-Acha and S. Seitz, "Regenerative morphing," in *Proc IEEE Comput Soc Conf Comput Vis Patter.*, 2010.

[14] R. Bridson, "Fluid simulation for computer graphics," 2008.

[15] J. D. Anderson, Ed., *Computational Fluid Dynamics: The Basics with Applications*, ser. Lecture Notes in Statistics. McGraw Hill, 1994.

[16] E. Catmull and R. Rom, "A class of local interpolating splines," *Computer Aided Geometric Design*, pp. 317–326, 1974.

[17] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of Applied Meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979.