

A General Model of Animated Shape Perturbation

Jean-Michel DISCHLER

Laboratoire M.S.I.

I.U.T. – Université de Limoges

Allée A. Maurois, 87065 Limoges Cedex - France

Abstract

Stochastic shape-perturbation, called shape- or hyper-texturing, represents an attractive way for rendering many complex surface structures including fur, fire, cotton or rocks. While current methods often limit applications to static, isolated and rather simple objects, this paper attempts to provide a more general approach, based on projection schemes. The presented technique not only permits to deal with various shapes, but furthermore combines perturbation with animation using 3D deformation models based on a principle of smoothed particles. The parameters of the deformation are controlled by the user through "projection primitives". Examples of animated surface behaviors (applied to usual polyhedrons such as the Utah teapot) including morphing, fluttering, burning and waving illustrate the possibilities of the approach.

Key words: rendering, realism, surface perturbation, hypertexturing, animation, mass-spring-based deformation, smoothed particles.

1 Introduction

Realism, which represents one of the most challenging problems in computer graphics, is closely related to the level of details at which surface complexity is modeled. Past investigations outline that complex "macroscopic" surface primitives, such as for example fur, cotton, textiles, rocks or erosion, are more efficiently simulated with a special kind of volumetric texturing approach [17, 23, 10, 21, 13, 15], than with explicit "brute force" geometric models (for example polyhedrons). Explicit models often lead to huge geometric data bases and unnecessarily increase the actual scene complexity.

In [23], Perlin and Hoffert introduce a method based on stochastic density modulation, called shape- or hyper-texturing. This method allows for multiple visual simulations beyond the possibilities of bump mapping [3], horizon mapping [19] and displacement mapping [7] to be obtained. Its great polyvalence led to various visual results, such as furry spheres, eroded cubes, or fire balls, and provided among the most advanced "textures" (in fact the concept is very close to geometric modeling) obtained to

date. In [10], the principle of stochastic surface "perturbation" is extended to more complex objects, including polyhedrons and CSG models (provided that the surfaces verify some basic topological conditions). This method introduced the use of skeletons and distance measurement in order to obtain "pseudo-implicit" descriptions of these objects (a volumetric description in the form of a potential field). In [16] and [14], perturbations were also applied to implicit surfaces.

However, as opposed to density variation 3D textures [12, 25], the 3D shape-perturbation methodology has not been deepened much further by researchers. Most of the previously mentioned complex phenomena, such as fire or hairs for example, were rather investigated using specialized physically-motivated approaches [26, 8, 1]. Shape-texturing cannot be compared with these approaches since, in contrast, it uses neither collections of explicit models (for example blobs or individual hairs), nor is it based on physical principles. Shape-texturing remains a purely "visual" approach, where the final appearance of the "macroscopic surface primitives" (hairs, flames, etc.) is entirely determined by the stochastic perturbation function [23]. This direct use of stochastic functions makes the problem of applying and using shape perturbation harder tractable, but also far more attractive. Shape-textures have two major advantages: the same model can be used in the context of a lot of very different effects. That is, only a few parameters need be changed to obtain either fur, or fire, or ice, or cotton or dripping plastic. Second, the description remains extremely compact, even in the case of very complex structures such as fur. This is related to the procedural nature of the methodology. Unfortunately, shape textures also raise several substantial problems, which may explain their limited use at these days (apart from the fact that the computational requirements are extremely important):

- the macroscopic primitives are entirely determined by the 3D perturbation function. It may become difficult to define appropriate functions to obtain precise effects and structures (what function will give

hairs, picks, flames, etc.?). The problem of turning a certain geometry into a corresponding stochastic function is a difficult – not yet – resolved problem;

- as opposed to texture maps or solid textures, the application to common objects (such as the Utah teapot polyhedron) is not straightforward. Basically, the methodology requires that the objects be modeled in a "volumetric" manner. In [23], objects were procedurally modeled in the form of density variation functions, and mainly simple shapes were considered (spheres, cubes, donuts, etc.);
- animating and deforming hypertextures by combining the methodology with physically-based approaches has not been addressed and seems difficult because of the polyvalence of the model. However, interactions among objects (collisions for example) or deformations due to force fields or due to motion (fur fluttering in wind fields, flames flickering, etc.) need to be considered, since applications rarely remain purely static in computer graphics.

Although we have already addressed the first point (the problematic of turning simple shapes such as bumps, picks, etc. into appropriate functions) in [11] by introducing a profile analysis, our approach remains only applicable to some very particular cases. In this paper, we do not discuss this delicate problem, but rather propose solutions for the two other mentioned points, namely the application of shape perturbation to common objects, as well as the consideration of animation. Therefore, the initial model developed in [10] is first improved by extending the principle of skeletons to a more general concept of "projection schemes". The method of [10] is limited by the fact that restrictive topological conditions must be verified, and by the fact that the potential fields are not continuous throughout the 3D space. The latter often results in visual artifacts. The new formulation allows to avoid both restrictions, and permits to extend the range of objects that can be "perturbed". Beyond improving the previous model, we additionally propose a blending technique, with similarity to implicit surfaces, in order to extend the modeling possibilities.

In a second part of this paper, animation is addressed by introducing a deformation model based on systems of smoothed particles. We show how deformations (for example involving simple physical principles, such as mass-spring systems) can be combined with the previously described perturbation methodology in order to obtain visually convincing motions, including waving, fluttering, burning, etc. All of the new concepts introduced in this paper allow for various complex visual simulations, generally hardly realizable using traditional approaches (ex-

PLICIT GEOMETRIC MODELS), to be obtained and easily controlled.

The remainder of the paper is organized as follows. In section 2, the general principle of using projection schemes to obtain pseudo-implicit descriptions of surfaces is described. The blending technique is also presented. In section 3, one deformation model, based on a mass-spring system is developed and applied to our technique. This simple deformation model is used as an example to show how perturbations can be merged with deformation models, using smoothed particles and the projection schemes. Applications to more sophisticated deformation models are straightforward. Finally, some graphical results are presented.

2 Shape perturbation, pseudo-implicit surfaces and skeletons

In the following subsection, we briefly describe the approach developed in [10], and outline its main limitations. This leads us to a more general formulation, explained in subsection 2.2. Finally, in subsection 2.3, the blending technique is presented.

2.1 The skeleton-based model and its limitations

Basically, "distance surfaces" can be obtained using skeletons and potential fields [34]. Often, they represent an easier to control alternative to traditional implicit surfaces, i.e. defined using a direct mathematical formulation. The projection-based shape-perturbation approach follows the inverse path. We first start with a given closed and orientable surface H , that can be for example a polyhedron. Because of its topological conditions, H divides the space into an inside and outside. Then, a geometric primitive S , called "skeleton" in [10], is put strictly inside H (see figure 1). S does not need to be computed as the morphological skeleton of H , but on the contrary, may have any shape and is freely designed by the user. It represents a "tool", used to steer the synthesis of the shape-texture. The possibility of "free" modeling is essential for the control of the global aspects of the perturbation (in fact, using an other "skeleton", results in an other aspect of the shape-texture).

Once modeled, the skeleton is used to derive a "pseudo-implicit" description of the surface H . Let be $W(P)$, where $P(x, y, z)$ represents any point of the 3D space, a function that measures either the proximity to H or to the skeleton S . W returns a scalar value between 0 and 1, and represents a kind of equivalent counterpart to the iso-potential fields commonly used with implicit surfaces [4]. W is defined as $1 - \frac{GP}{GR}$, where G represents the closest projection of P onto S , and R the closest intersection between the line GP and H , that always exists because of the topological conditions. The closest pro-

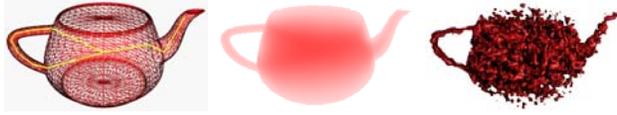


Figure 1: An example of skeleton S inside a Utah teapot polyhedron. We added some polygons in order to make it closed (left). The skeleton S together with H allows us to define a potential field and to build a pseudo-implicit description of this teapot (middle). It can then, (right), be perturbed using a stochastic function such as the solid noise of [22]. The perturbation engenders macroscopic primitives.

jection on a skeleton made of points, segments and faces, can be obtained using simple perpendicular projections. The computation is simple, but unfortunately, the so resulting field is not always continuous (see the next subsection). $W(P)$ is equal to 1, if P is on the skeleton, 0 if P is exactly upon H and for example 0.5, if P is at mid-way between S and H .

$W(P)$ represents a function of the 3D space that describes the object H as a "full" volume. Its border is given by $W(P) = 0$. H can be stochastically "perturbed", that is, shape-textures applied to any kind of closed and orientable surface, by "perturbing" W . We use following model [10]:

$$W(P) = 1 - \frac{GP}{\sigma \cdot GR}, \quad \sigma = 1 - \frac{\alpha}{2}(\Phi(P) - 1)$$

σ represents the perturbing factor, α (between 0 and 1) the "depth" of the perturbation and Φ the perturbation function, almost equivalent to Perlin's dmf [23]. Φ returns scalar values between -1 and 1 and is generally, but not restricted to, a stochastic function of the 3D space. It can be, for example, one of [22, 18, 32]. Note that perturbing a volume H , is not the same as perturbing its mesh. In our case, the topology may change, and very complex "disconnected" structures can be obtained. It is also advantageous to use a volumetric description rather than a mesh, because translucency can be processed straightforwardly. Indeed, it is possible to assign to $W(P)$ a density variation (for example from 1 on S , down to 0 on H) in order to simulate fuzzy and blurry structures such as fur, fire, cotton, clouds, etc. (see figure 3, for an example of fuzzy structure applied to a box).

The choice of the perturbation function is essential, since it "shapes" the different macroscopic surface primitives (either bubbles, or hairs, or flames, etc.). Figure 2 illustrates different structures, obtained for different functions (all of them based on noise and turbulence [22]).

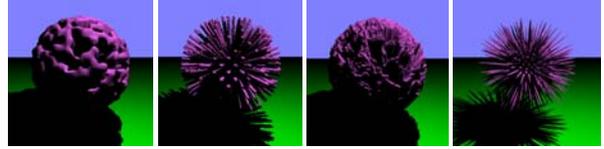


Figure 2: The perturbation function shapes the surface elements (picks, hairs, etc.)

We applied the perturbations to simple balls and used as skeleton their center. For these figures, we evaluated the function Φ at R instead of P . This makes the structures "projected" onto the skeleton (resulting in picks, hairs, etc.). The shape and position of the skeleton strongly influences (in an intuitive manner) the directions and orientations of these picks. For "disconnected" structures, such as shown in figure 1, the shape and position of the skeleton also influences the global aspect of the texture: in particular, the macroscopic structures become denser as they come closer to the skeleton.

Rendering is performed as follows. When a ray strikes and enters the surface H , this ray is sampled into more or less regularly spaced points P_i . According to whether W is positive or negative, these points will lie respectively inside or outside the perturbed surface. The true intersections between the perturbed surface and the ray can be computed by refinement using a binary search for example. In order to apply a shader, normal vectors are computed with the help of the gradient of W . More optimized techniques, as for example presented in [15] in the case of displacement shaders, may be used, but we currently did not develop such techniques. It is also possible to render fuzzy and translucent structures by accumulating densities, as presented in [23].

2.2 Using extended projection schemes

The previously described "skeleton"-based perturbation model turns out to be actually very general. There are no real restrictions concerning the shapes of H and S , except the topological condition of H and the fact that S needs to be put "correctly" inside H . In fact, S materializes a kind of "primitive" that allows us to define a projection scheme, and, thus, to derive a distance measurement function (and a potential field). Keeping this concept of projecting onto simple geometric primitives in mind, it becomes possible to generalize the initial approach of [10], by considering more general "projection-schemes" rather than "skeletons" combined with perpendicular projections (closest distances). A more general formulation allows us to extend the application of shape-perturbation to other complex objects, that do not verify the topological conditions. It additionally permits to avoid problems



Figure 3: Resolving problems of discontinuities using advanced projection schemes.

of discontinuities due to the perpendicular projection. In the following paragraphs, we present two examples of extensions: the first, allows to avoid discontinuities, and the second allows to extend the class of objects that can be perturbed. The latter also permits to construct automatically the projection primitives.

Using closest distances to geometric features such as points, segments and faces introduces discontinuities on equidistant regions. Figure 3 illustrates an example of discontinuity. The object to be perturbed is a box, and the skeleton is given by two joint symmetric segments making a certain angle between them (see the left part of figure 3). Using a perpendicular projection (a closest projection) infers a subdivision of the space into three distinct regions: two regions, whose respective points are closest to one of the two respective segments, and one region of equidistant points (in this case, it is a plane dividing the box into two equal parts). The potential field given by $W(P)$ obviously exhibits a discontinuity on that region. Often, this results in unpleasant visual artifacts, as shown by the middle part of figure 3. In what follows, we show that it is possible to avoid such artifacts by extending the projection technique.

The extension consists of "smoothing" the projection, by considering mean values among the primitives. Let be S_i the different skeleton primitives, and G_i the respective points obtained by projecting P on S_i . R_i represent the respective intersections with H . Instead of keeping only the closest G_i , we shall use all values by computing an average:

$$W(P) = 1 - \frac{\sum w_i G_i P}{\sigma \cdot \sum w_i G_i R}$$

where w_i represents a weight (in fact, a function representing a smoothing kernel depending on the distance $G_i P$). The right part of figure 3 illustrates the result obtained for the previous box by applying the "advanced" projection scheme. As visible, there no longer appear any discontinuities. The whole hypertexture, as well as the potential field have been smoothed, thus canceling out visual artifacts.

Beyond resolving problems of discontinuities, advanced projection schemes can also be used to extend the range of objects that can be perturbed. The texel map-

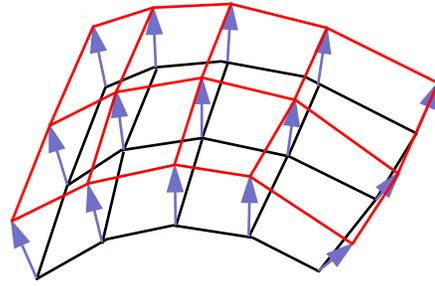


Figure 4: Construction of a texel-like "skin".

ping principle [17], for example, that consists of adding a "skin" to objects, can be instantly transposed to the new formulation in the following way. For each polygon of the surface H , we construct a new corresponding polygon using an interpolated normal vector on its vertices (see figure 4), plus a certain given span. Doing this for each polygon of the object yields a kind of dilatation, i.e. a new larger surface H' . Between H and H' , we obtain a "skin" which is characterized by a certain given depth. Each couple of polygons determines one elementary volume, i.e. a "texel". Now, H can be used as projection primitive and H' as surface to be perturbed. Therefore, we simply use a projection scheme based on an interpolation of the normal vectors (see figure 5) of the vertices. That is, the interpolated normal vector on P defines a line, that can be used to compute intersections with H and H' , i.e. we obtain respectively G and R . The global use of the perturbed distance function W remains the same as previously described, but there are no longer topological restrictions concerning the objects. Also the "projection primitive" is computed automatically.

We note however, that unfortunately, this approach is not universal, and cannot be used for all kinds of objects. Some restrictions remain. In particular, the skin might self-intersect if the span is chosen too large, thus yielding "impossible" objects.

Figure 6 illustrates a cotton-like texture applied to a torus (note the "soft" translucent structure of the texture). We used the previously described technique, by first constructing a skin around the shape, which gives a certain profoundness to the texture.

The two previously described techniques allow us to avoid problems of discontinuities, as well as problems of limitations concerning the range of objects that can be perturbed. The choice between both techniques mainly depends on the type of object that is to be perturbed. In the remainder of this paper, we essentially used the first technique to apply the hypertextures to polyhedrons (mainly because of simplicity). However, all principles also apply to the second "texel-like" approach.

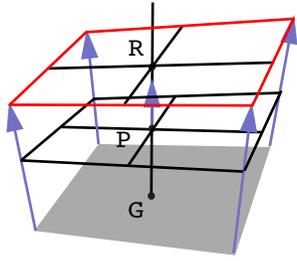


Figure 5: Defining a new texel appropriated projection scheme to apply perturbations.

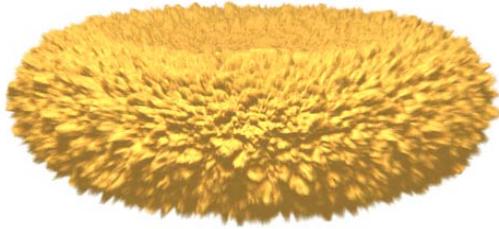


Figure 6: A "soft" texture applied to a torus using a texel-like skin.

Note that the consideration of general projection schemes opens the way to the creation of various new perturbation schemes, that can be adapted to particular classes of objects. With appropriate schemes, it seems that nearly all objects can be hypertextured and perturbed. In addition, many more different types of perturbations are imaginable using more, different projection schemes.

2.3 Blending

The described projection-based approach is a powerful technique for applying shape-perturbations to the most complex and various objects. It considers, however, all objects independently, without accounting for interactions. We now extend the model to multiple interacting structures, by considering blending.

The simplest way of dealing with interactions is basi-

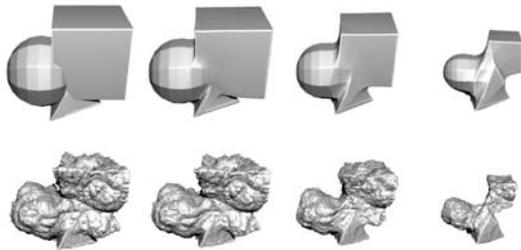


Figure 7: Blending different perturbed polyhedrons.

cally to ignore them (see the left part of figure 7). But, this is only possible in some very limited and particular cases, since it not matches the actual physical reality. Because of the close relationship between the "pseudo-implicit" surfaces that we introduced in section 2.1 and "real" implicit surfaces, it seems possible, at least at the first sight, to consider physical interactions, such as collisions, as it has been investigated for example in [6]. Unfortunately, we do not dispose of iso-potential fields in our case, thus impeding a direct application of these methods. Still, interactions can be considered, using unphysical approaches (not volume preserving, for example). We implemented a simple way of blending perturbed surfaces, by using again the distance measurement function, W , and by computing an average over neighboring structures as follows:

$$W(P) = \frac{\sum m_i W_i(P)}{\sum m_i}, \quad m_i = \left(\frac{G_i R_i}{G_i P}\right)^{\frac{1}{pw}}$$

i represents the different interacting surfaces. Intuitively, the equation formally describes that, while points lie inside the surfaces (where W is positive), material is added. On the contrary, material is removed, if points lie outside (where W is negative). In both of these cases, the amount depends on the proximity of the projection primitive. The coefficient pw is used to control how much the shapes do influence neighbors. Figure 7 illustrates an example of blending in the case of three overlapping objects (three polyhedrons). The upper row shows the objects without perturbation, and the lower one the result obtained with a certain rock-like perturbation. The left part represents the reference, i.e. no blending. The three remaining couples of pictures, from left to right, illustrate the influence of an increasing coefficient pw (respectively 0.2, 0.5 and 1.0). Note that the shapes' sizes decrease because of the subtractive effect.

The main application field of blending, is to be able to model complex shapes using basically simpler ones, preserving a certain global coherence. Mount Rushmore (see figure 8), for example well illustrates this. Each United States president's bust represents one individual "basic" shape, that can be modeled and stochastically perturbed to get the typical eroded rock-like structure. But, as visible on the illustration, Mount Rushmore obviously forms only one unique coherent object. This is precisely the kind of effect that can be well reproduced using the previously described "blending" technique (see figure 7).

3 Applying deformations using smoothed particles

The projection-based shape perturbation methodology, including blending, is only depending on a geometric in-

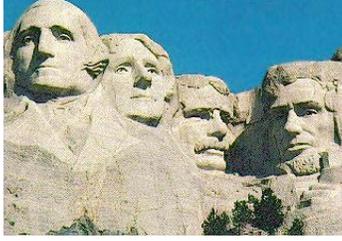


Figure 8: Mount Rushmore (real photograph) illustrates a possible application of blending, in the case of a rock structure

formation, given by the shapes and positions of the surfaces and the projection primitives. Consequently, global space deformation approaches, as surveyed in [2], directly become applicable. In fact, global space deformation approaches turn out to be extremely useful, whenever one needs to deal with an indistinct geometry that has no apparent border or precise shape, as it is the case with perturbation. For a similar reason, a space deformation approach was used by Neyret in [20] to animate texels. However, the latter technique does not apply to ours, because of the more complex structures that the perturbation engenders, as opposed to the repeated elementary volumes characterizing texels.

Let be Ω a global space deformation. In order to apply Ω to the perturbed object, we proceed as follows. Ω is first applied to H in order to compute the new boundaries of the object. The primitives (bubbles, hairs, etc.) are deformed during the rendering process using Ω^{-1} , an inverse deformation (in fact, a transformation that maps back each point to its initial position). That is, when the ray strikes and enters the surface of $\Omega(H)$, each point P_i is replaced by $\Omega^{-1}(P_i)$, while the values of W are still computed with the original surface H using the associated projection scheme.

Physically-motivated deformation models have a long history in computer graphics [28, 29, 30]. Many approaches use particles systems [31, 5, 33], as initially introduced by [24]. To define simple 3D space deformations, we shall also use a particles system (i.e. a set of finite moving points that are carrying some information). The extension to the entire space is then performed using a smoothing kernel in a similar way as presented in [9]. The advantage of the particles system is that the motion can be viewed and modified interactively for small sets. In the remainder of this section, we more particularly develop a simple mass-spring based deformation model, with similarity to [27] (in [27] mass-spring systems were used to animate spline curves representing vegetation branches). This simple model is used as an

example, but the global methodology applies to any other deformation scheme based on particles. The main objective of this section is to show how the deformation can be merged with the perturbation principle, and how some of the parameters can be linked to the projection primitive in order to control some global aspects.

Let be x_i the particles. To each x_i an own mass m_i is assigned. The material's "solidity" is simulated as all x_i are suspended at their initial position $x_i(0)$ by a spring of stiffness k_i . The behavior of a single mass-spring system is trivially governed by following differential equation:

$$m_i \ddot{x}_i = m_i g_i - k_i x_i - f \dot{x}_i + F$$

where f represents a friction coefficient, g the gravity constant and F an external force. This system can be numerically resolved using an Eulerian scheme, i.e. using finite approximations of the differentials.

The particles inside the volume need to be chosen carefully, in order to fill the volume densely enough to prevent artifacts due to the smoothing operation and to the related interpolation. The selection of particles can be made regular, by using the vertices of a grid, or adaptive, according to the shape of the object and according to regions of high frequencies. In practice, we used a regular grid, plus the particles representing the vertices of the mesh of H . Figure 9 illustrates an example of particles distribution. The two left pictures show the meshes (the initial one and the deformed one). The two pictures on the right show the particles. We used a 3D force field based on a low frequency noise which can be assimilated to wind.

None of the model's parameters (stiffness, weight, etc.) need to be constant inside the volume. On the contrary, the projection primitive can (must) be used to modulate the values, in order to provide a certain user control and to avoid incoherently moving particles. Often the projection primitive (in the case of the pawn of figure 9, it is a segment matching the central axis) materializes a kind of rigid "deepest interior". Therefore, we made the stiffness coefficient dependent on the distance to the axis. The stiffness increases as the points come closer to the axis. This variation can be made linear or exponential, resulting in different global behaviors and visual aspects. More generally, all of the parameters characterizing a particular physical model of deformation, can be set with respect to the distance to the projection primitives.

Once the motion has been "designed" in real time, it needs to be applied to the actual "perturbed" object. The deformation Ω corresponds to a displacement of all points with respect to their initial position p_i . The inverse deformation, is obtained using the inverse displacement vector. Ω^{-1} , which is at this time only defined on the finite points x_i , is smoothly extended to the entire continuous

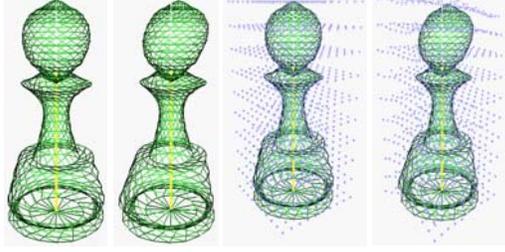


Figure 9: Particles help to visualize the motion in real time.

volume, by using an interpolation according to a certain smoothing kernel K :

$$\Omega^{-1} = \sum K(P - x_i) \overline{\Omega^{-1}}(x_i)$$

where $\overline{\Omega^{-1}}$ represents the inverse deformation on the discrete locations. It is defined as the displacement $D_{x_i(0) \leftarrow x_i(t)}$, where $x_i(t)$ represents the position of a particle at time t . In practice, we used a smoothing kernel K inversely proportional to the distance, that takes the value 0 beyond a certain maximal distance. In order to directly access the closest particles inside the volume, and thus to get a more efficient computation, especially if huge numbers of particles are used, we also subdivide the volume of $\Omega^{-1}(H)$ into a regular grid of voxels, each voxel containing an own list of particles $x_i(t)$. For computing the inverse deformation on P , we only use the particles of the voxel including P , as well as those of the 26-connected neighboring voxels. The size of the voxels depends on the size of the smoothing kernel.

4 Results

4.1 Direct animation

Making the perturbation function Φ time dependent directly yields animation. Animation is also possible, if some parameters are modified according to time, for example α that represents the depth of the shape-perturbation or by changing and moving the projection primitive and/or the interacting polyhedrons. The first upper filmstrip of figure 10 represents five frames out of a video sequence, that progressively increases the texture depth α . As function Φ , we used the solid noise of [22]. This animation, randomly perturbing the teapot, looks somewhat like "disintegration". An other kind of animation consists of "morphing" the surface aspect. We simply need to linearly interpolate the coefficients and the parameters of the projection-based model. The second strip of figure 10 shows some frames out of a video sequence, that consists of morphing a ground-like surface into a metallic teapot, "bombed" with holes. The

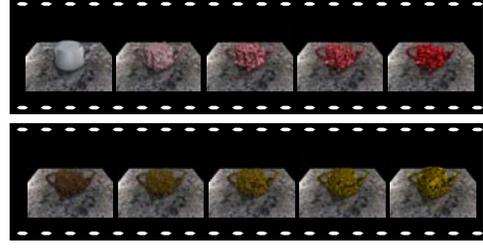


Figure 10: Two examples of direct animation: disintegration and morphing.



Figure 11: (left) peaks without deformation, (middle) applying a noise-based stochastic force field. (right) applying the deformation to a "soft" texture.

obtained morphing is rather of good quality. Many more types of animation are conceivable and simple to realize. Concerning timings, one frame of each of these sequences required about half an hour computation time on a Silicon Graphics O2 workstation, with a R5000 processor and 128Mb RAM for a resolution of 400*300 pixels.

4.2 Animation based on particles

Figure 11 illustrates a waving deformation obtained using a low frequency noise-based vector field. This deformation corresponds to the one presented in figure 9 in the previous section. This animation well outlines the importance it has to modulate the stiffness according to the proximity of the projection primitive. Indeed, without modulation the entire object would unnaturally and elastically move, as opposed to the "branches' extremities" only. The picture on the left (without deformation) required about 40 minutes for rendering. With deformation (right), it required about 2 hours (in both cases using 9 stochastic rays per pixel, for antialiasing). The right part of figure 11 shows the same deformation applied to a "soft" (translucent) texture.

Figure 12 illustrates two more examples of deformations. The left part shows a fluttering behavior, obtained using a similar approach as for figure 11, applied to a

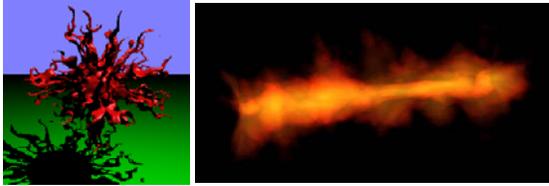


Figure 12: More examples: (right) fluttering, (left) burning.

perturbed ball. The animation however looks different because the random force field has a higher frequency. The right part represents a kind of burning stick. In this case, only the perturbation function was deformed (not the object itself). This deformation is based on particles moving away from the projection primitive (the central axis of the stick). No springs were used. We note that this approach for the simulation of "burning", remains purely empirical, as opposed to the model presented in [26]. Nonetheless, the result looks visually convincing, which outlines the great polyvalence of the shape perturbation methodology, especially for rendering complex phenomena. The fire structure was obtained using a squared turbulence function [22], plus different colors on the projection primitive and the surface H (respectively yellow and red), as suggested in [23, 10].

5 Conclusion

In this paper, we have presented a general method of applying stochastic surface perturbations to the most various objects. We also considered blending, and showed how the model can be combined with particles based deformations. As a result, we obtained various complex visual simulations, such as morphing or fluttering. The simulations are computed according to parameters controlled by the user through the projection primitives. The whole method allows to considerably extend the application field of shape perturbation, while it remains simple to implement.

Beyond texturing, shape perturbation offers a powerful alternative to traditional geometric modeling (explicit models, such as polyhedrons). A major restriction, however, lies in the high computational requirements inherited from the related volume rendering technique. This is actually the price for memory space preservation (because of the procedural nature). The main problem is the computation of the distance measurement function W . If the projection scheme is complex and the object modeled using many polygons, the timings might become rapidly important. Still, the method remains an interesting way to deal with complex, more or less stochastic animated behaviors.

Apart from excessive computational requirements, there are additional topics that need to be furthermore investigated. For example, it is always difficult to compute "exact" intersections with complex potential fields: some small features might be missed. An other important point is how to control the perturbation function to obtain precise effects (first point mentioned in the introduction). All of these problems are currently representing important obstacles to the use of shape perturbation as an open modeling tool.

Some future topics that we aim to develop, consist of using well chosen noise- and turbulence-based [22] perturbations, combined with simplified physical models of flowing liquids, in order to simulate rapid turbulent water flows. We also intend to extend interactions to collisions with rigid objects.

6 References

- [1] K. Anjyo, Y. Usami and T. Kurihara, "A Simple Method for Extracting the Natural Beauty of Hair", *Computer Graphics* **26(2)**, 1992, pp. 111–120.
- [2] D. Bechmann, "Space Deformation Models Survey", *Computers and Graphics* **18(4)**, 1994, pp. 571–586.
- [3] J.F. Blinn, "Simulation of Wrinkled Surfaces", *Computer Graphics* **12(3)**, August 1978, pp. 286–292.
- [4] J. Bloomenthal, "Introduction to Implicit Surfaces", *Morgan Kaufmann*, 1997.
- [5] D.E. Breen, D.H. House and P.H. Getto, "A physical-based particle model of woven cloth", *The Visual Computer* **8(5)**, June 1992, pp. 264–275.
- [6] M.P. Cani-Gascuel and M. Desbrun, "Animation of Deformable Models Using Implicit Surfaces", *IEEE Transactions on Visualization and Computer Graphics* **3(1)**, March 1997, pp. 39–50.
- [7] R.L. Cook, "Shade Trees", *Computer Graphics* **18**, July 1984, pp. 223–231.
- [8] A. Daldegan N.M. Thalmann, T. Kurihara and D. Thalmann, "An Integrated System for Modeling, Animating and Rendering Hair", *Computer Graphics forum* **12(3)**, *Eurographics'93*, 1993, pp. 211–221.
- [9] M. Desbrun and M.P. Gascuel, "Smoothed Particles: a New Paradigm for Animating Highly Deformable Bodies", *Proceeding of Eurographics Workshop on Computer Animation and Simulation*, Poitiers, 1996, pp. 61–76.
- [10] J.M. Dischler and D. Ghazanfarpour, "A Geometrical Based Method for Highly Complex Structured Textures Generation", *Computer Graphics forum* **14(4)**, 1995, pp. 203–215.

- [11] J.M. Dischler and D. Ghazanfarpour, "A Procedural Description of Geometric Textures by Spectral and Spatial Analysis of Profiles", *Computer Graphics forum* **16(3)**, Eurographics'97, Budapest 1997, pp. 129–139.
- [12] D.S. Ebert and R.E. Parent, "Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-buffer Techniques", *Computer Graphics* **24(4)**, 1990, pp. 357–366.
- [13] E. Groller, R.T. Rau and W. Strasser, "Modeling Textiles as Three Dimensional Textures", Eurographics Workshop on Rendering, Porto, June 1996, pp. 205–214.
- [14] J.C. Hart, "Implicit Representations of Rough Surfaces", *Computer Graphics forum* **16(2)**, 1997, pp. 91–99.
- [15] W. Heidrich and H.P. Seidel, "Ray-Tracing Procedural Displacement Shaders", *Proceedings of Graphics Interface*, June 1998.
- [16] Z. Kacic-Alesic and B. Wyvill, "Controlled Blending of Procedural Implicit Surfaces", *Proceeding Graphics Interface*, 1991, pp. 236–245.
- [17] J.T. Kajiya and T.L. Kay, "Rendering fur with Three Dimensional Textures", *Computer Graphics* **23(3)**, July 1989, pp. 271–280.
- [18] J. Lewis, "Algorithms for solid Noise Synthesis", *Computer Graphics* **23(3)**, July 1989, pp. 263–270.
- [19] N. Max, "Horizon Mapping: Shadows for Bump-Mapped Surfaces", *The Visual Computer* **4**, 1988, pp. 109–117.
- [20] F. Neyret, "Animated Texels", *Proceedings of Eurographics Workshop on Animation and Simulation*, 1995.
- [21] F. Neyret, "Synthesizing Verdant Landscapes using Volumetric Textures", Eurographics Workshop on Rendering, Porto, June 1996, pp. 215–224.
- [22] K. Perlin, "An Image Synthesizer", *Computer Graphics* **19(3)**, July 1985, pp. 287–296.
- [23] K. Perlin and E.M. Hoffert, "Hypertexture", *Computer Graphics* **23(3)**, July 1989, pp. 253–262.
- [24] W.T. Reeves, "Particle systems - a technique for modeling a class of fuzzy objects", *Computer Graphics* **17(3)**, 1983, pp. 359–376.
- [25] G. Sakas, "Modeling and Animating Turbulent Gaseous Phenomena Using Spectral Synthesis", *The Visual Computer* **9**, 1993, pp. 200–212.
- [26] J. Stam and E. Fiume, "Depicting Fire and Other Gaseous Phenomena Using a Diffusion Process", *Computer Graphics, Siggraph'95, annual conference series*, 1995, pp. 129–136.
- [27] J. Stam, "Stochastic Dynamics: Simulating the Effects of Turbulence on Flexible Structures", *Computer Graphics forum* **16(3)**, Eurographics'97, Budapest, September 1997, pp. 159–164.
- [28] D. Terzopoulos, J. Platt, A. Barr and K. Fleischer, "Elastically Deformable Models", *Computer Graphics* **21(4)**, July 1987, pp. 205–214.
- [29] D. Terzopoulos and K. Fleischer, "Modeling inelastic deformation: viscoelasticity, plasticity, fracture", *Computer Graphics* **22**, Siggraph'88, 1988, pp. 269–278.
- [30] D. Terzopoulos, J. Platt and K. Fleischer, "Heating and melting deformable models (from goop to glop)", *Proceedings of Graphics Interface'89, June 1989*, pp. 219–226.
- [31] D. Tonnesen, "Modeling liquids and solids using thermal particles", *Proceedings of Graphics Interface'91, June 1991*, pp. 255–262.
- [32] S. Worley, "A Cellular Texture Basis Function", *Computer Graphics, Siggraph'96, annual conference series*, 1996, pp. 291–294.
- [33] Y. Wu, D. Thalmann, and N. Magnenat-Thalmann, "Deformable surfaces using physically based particle systems", *Proceedings of CGI'95 conference, June 1995*, pp. 205–215.
- [34] G. Wyvill, C. McPheeters and B. Wyvill, "Data Structure for Soft Objects", *The Visual Computer* **2(4)**, August 1986, pp. 227–234.