DEFORMABLE ELASTIC 2-D OBJECTS

F. Abram, P. T. Sander

Thème Images/Réalité Virtuelle
Laboratoire UNSA/CNRS I3S
Les Algorithmes, Bâtiment Euclide
2000, route des Lucioles
Sophia Antipolis 06410 BIOT France
abram@i3s.unice.fr http://www.essi.fr/~abram
sander@essi.fr http://www.essi.fr/~sander

ABSTRACT

In this paper, we present the first steps of a 4-D oriented study of object deformation based on both local geometric deformations and global topological constraints of regularization, and shape preservation. The model described handles the shape of the objects to simulate the manipulation of elastic objects that always return to their original shape when no constraints are applied. Many kinds of forces can be applied on the objects in a scene, e.g., some user forces, object interactions or attraction/repulsion fields. Following the placement of objects into a scene, the deformation process consists in three stages, a free deformation stage under the effect of all the forces of the scene, a stage of regularization to constrain the previous deformation with respect to the original shape of the object, and finally a stage of re-positioning the object in the scene using the theory of material systems.

Keywords: regularization, material systems, free form deformation, differential geometry.

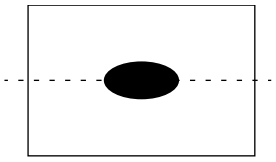
INTRODUCTION

The model we describe in this paper is locally defined and locally deformed with respect to its global rest shape. Most work uses regular shapes to deform, e.g., boxes, spheres, planes, parametric curves, implicit curves, and so on. With our model, the use of arbitrary object shape is permitted. All the deformations resulting from applied forces, done with respect to this shape, are done in a local neighborhood at a local object level. Animation and motion are done at a global object level with respect to the theory of material systems to produce realistic movement during the deformation.

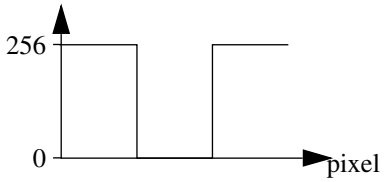
The following section briefly covers some related work. We next present the composition of a scene, the various kinds of forces we use to simulate obstacles, attractions and so on. Then the geometric model and topological constraints that ensure a correct deformation are introduced. We then progressively describe the evolution process, and present some examples using most of the defined forces to give some realistic effects. We conclude with some remarks on extensions currently in progress.

RELATED WORKS

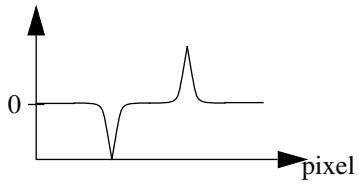
Much work has been done on solid modeling operations, deformable objects and animations. From the theoretical study of deformations [Barr84] as a preliminary stage, the evolution was given by Terzopoulos [Terzopoulos88] where animation of elastic bodies was determined by the physical laws of curves, surfaces and solids. On the other hand, animation work led to results presented by Sederberg [Sederberg93] with



Example of a B/W image used to create a force field

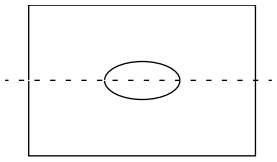


A single line of the B/W image

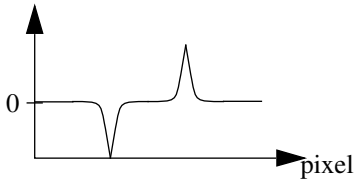


The resulting forces

Figure 1



Example of an obstacle defined by a line



The resulting forces

Figure 2

evolution based on an interpolation method between two curves; by Gascuel [Gascuel93] dealing with the contact surface during collision of objects; by Baraff [Baraff92] and Provot [Provot95] for simulation of clothes with a network of masses and springs; and by Szeliski [Szeliski93] modeling arbitrary shapes with a field of dynamic particles. More recently, with the increasing power of computers, animation uses more and more complicated models to create the best effects. The intent of these models is to simulate natural phenomena in ever-greater detail. Thus for example, Desbrun [Desbrun94] has proposed a model of skin as an application of his work on highly deformable bodies and Terzopoulos [Terzopoulos94] has simulated the movement of fish in a virtual world.

SCENE

The objects in out scene react to applied forces. The notion of scene, in this paper, groups objects, obstacles, force fields, user-defined forces and so on, almost anything that will interact with a deformable object. All the forces that we define, except the user-defined force, obey the equation of electrostatic interactions between charges

$$\vec{F} = (-k)\frac{qq'}{r^2}\frac{\vec{r}}{||\vec{r}||} \tag{1}$$

where q and q' represent the magnitudes of the charges, $\vec{r}$ is the distance vector between them and k some physical constant. In the following discussions, we will ignore the constant k term.

Force Fields

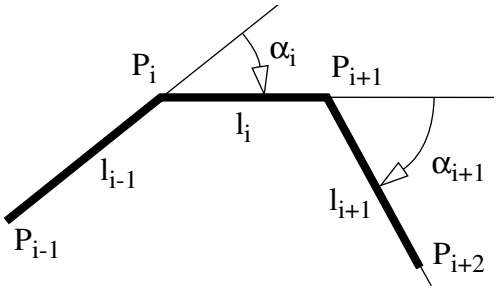
The scene may be superposed on a force field defining a static environment of forces. This is described by an intensity image I where we state that level transitions represent the presence of force with, as a magnitude, the value of the gradient ∇I. From this image, we deduce the force field by a 2D convolution method using the centered kernel defined by Eq. 1, where $q = \nabla I$. To control the magnitude of the force, we just shift Eq. 1 by 1 unit along the distance so the equation becomes

$$\overrightarrow{F_{ff}} = \frac{\nabla I}{(r+1)^2} \cdot \frac{\vec{r}}{||\vec{r}||} \tag{2}$$

Fig. 1 illustrates the force field notion, from an image to the force field itself. The force field is an easy and cheap way to define a bunch of forces. To know the force applied at any point P in the plane that belongs to an object, we just multiply the result of the preprocessed image by the magnitude of the charge of the object, $\vec{F} = q'\overrightarrow{F_{ff}}$.

Obstacles

In the scene, we can place obstacles that will interact on the deformable object close to them. By definition, an obstacle will be defined as an object that is not affected by any sort of force, but because of its charge, will create a local field of forces around itself. When the obstacle is static, it can be directly integrated into the force field image used to describe the environment. But the obstacle may also be in motion. This movement will not be affected by any kind of force in the scene. The local-force field created around an obstacle is define as in Eq. 2 where $\vec{r}$ will denote the distance vector to the closest shape point of the deformable object. Fig. 2 shows the obstacle corresponding to the Fig. 1 force field. The same local field is defined. The main difference between the two implementations is that the first is static and does not allow evolution,



Definition of the characteristics of a form.
Figure 3

whereas the second is dynamic, e.g., forces are calculated at every iteration, to undergo movement.

Objects

Objects contained in the same scene will interact with each other. The same laws are used to compute the forces between objects, and the force formula respects Eq. 2. The difference with the other kinds of force is that the interaction will create the deformation of both objects with respect to the deformation process described in the following sections

User Defined

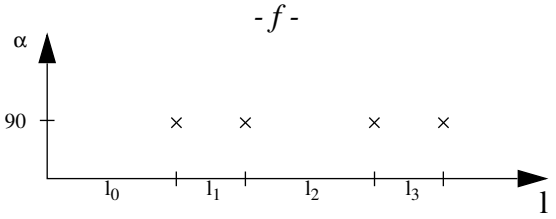
The user-defined force is a force applied on a single point of an object. The magnitude of the force is calculated with respect to the displacement of the pointer motion. The user can apply multiple forces to the objects of the scene.

MODEL

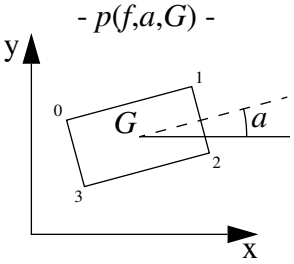
Intuitively, a 2D-object in an image is a curve that delimits an object, and could be represented in various ways - by a hand-drawn curve, a broken line, a parametric curve, differential parameters and so on. Before introducing in the next section what we use as an object, we need to introduce some basic definitions. Let us define the notion of a *form*, f, as a closed broken line described by an ordered set of n pairs

$$f = \{(\alpha_i, l_i), i \in [0, n-1]\}$$

where α_i represents the angle of the object at P_i, more precisely, the angle between the vectors $\overrightarrow{P_{i-1}P_i}$ and $\overrightarrow{P_iP_{i+1}}$, and l_i represents the distance between P_i and P_{i+1} (l_i is the length of the vector $\overrightarrow{P_iP_{i+1}}$), see Fig. 3. At point 0, we define α_0 as the angle between the vectors $\overrightarrow{P_{n-1}P_0}$ and $\overrightarrow{P_0P_1}$, at point n, we define l_{n-1} as the length of vector $\overrightarrow{P_{n-1}P_0}$. This definition of a form gives a uniform role to all points, i.e., there are no extremities on a form. We denote F_n the set of all forms of length n.



Example of a rectangular form f. The illustration draws the discrete function of α with respect to $\sum l$



Example of the representation of the rectangle form f in the 2-D plane by the placement p.
Figure 4

Let us now introduce the notion of a *placement*, p, as being the transformation that maps a form $f \in \mathrm{F}_n$ into a set of $n+1$ 2-D points

$$p: \mathrm{F}_n \times \mathbf{R} \times \mathbf{R}^2 \rightarrow (\mathbf{R}^2)^{n+1}$$
$$p((f, a, G) \in \{(x_i, y_i) \in \mathbf{R}^2, i \in [0, n]\})$$

given the orientation a of the principal axis and the position G of the center of gravity that we want to give to the form in the 2-D plane of the scene.

Fig. 4 shows a description of a form f of F_4 and its placement in the 2-D plane with an orientation a and a center of gravity G. This notion of placement comes directly from differential geometry where a curve in the plane is uniquely defined as a function of the curvature, with a given starting angle and position [doCarmo76]. We consider the center of gravity rather than a starting point and orientation in order to avoid the use of local geometric characteristics in the global notion of placement; that could result in increasing the instability of the whole process.

Finally, we define p^{-1}, the inverse transformation of the placement p, which produces the form f of F_n, the orientation a of the principal axis and the position G of the center of gravity, from a set of $n+1$ 2-D points

$$p^{-1}: (\mathbf{R}^2)^{n+1} \rightarrow \mathrm{F}_n \times \mathbf{R} \times \mathbf{R}^2$$

Properties

The advantage of such a curvature-based object description is its invariance with respect to Euclidean

transformations; whatever the position and orientation of a placement in the 2-D plane, p^{-1} will give the same form, f.

Closed curves rely on two essential characteristics. The first one is an intuitive property that states the fact that the first and the last point of a closed curve must superpose. Applied to the discrete closed curve forms we manipulate, this translates to the fact that the sum of the oriented vectors linking all pairs of successive points must be equal to 0.

$$\sum_{i=0}^{n-1} \overrightarrow{P_i P_{i+1}} = \sum_{i=0}^{n-1} l_i \cdot \begin{pmatrix} \cos \sum_{j=1}^{i} \alpha_i \\ \sin \sum_{j=1}^{i} \alpha_i \end{pmatrix} = 0 \qquad (3)$$

In this equation, we assume that $P_n = P_0$. In the 2-D case, this property gives a system of 2 equations expressed in the second members of Eq. 3.

The second property states the fact that a closed curve can be traversed several times while following the same path. This characteristic is written in [doCarmo76] as

$$\int_{\Omega} \kappa(s) = 2\Pi \cdot I$$

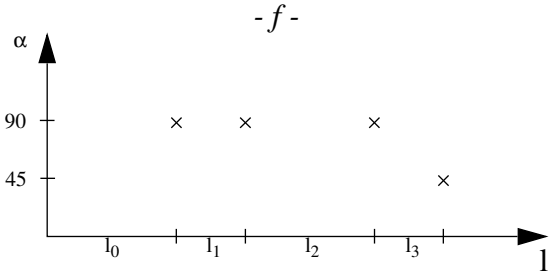
where I denotes the *rotation index* of the curve, and applied to our discrete closed form this becomes

$$\sum_{i=0}^{n-1} \alpha_i = 2\Pi \cdot I \qquad (4)$$

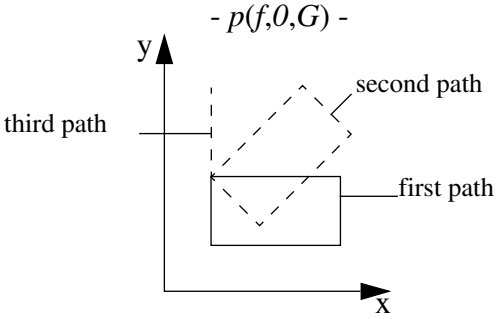
Notice that a curve that satisfies Eq. 3 is not a form if it does not respect Eq. 4. Fig. 4 illustrates a regular form while Fig. 5 highlights the case of a non closed curve. For the purpose of the discussion, we will denote by C_n the set of broken lines that are defined like forms but do not verify the 2 conditions. On the set C_n, we define the following + operator applied to two curves c and c'

$$+ : \mathrm{C}_n \times \mathrm{C}_n \rightarrow \mathrm{C}_n$$
$$c + c' = \{(\alpha_i + \alpha'_i, l_i + l'_i), i \in [0, n-1]\}$$

Notice that the addition of two forms f and f', or of a form f and a curve c does not necessarily give a curve that respects the conditions of a form.



Example of a rectangular curve. The illustration draws the discrete function of α with respect to $\sum l$



Example of the representation of the rectangle curve in the 2-D plane by the placement p. Notice the fact that, because of the value of α_3, there is an angle shift from one path to the following.

Figure 5

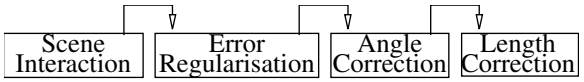
OBJECT

Now, we can introduce the object model that we will be using all along this work. An object consists of the following elements: a form f_o representing the reference form of the object; a form f_d being the deformed form that will be used to find an equilibrium state representing in the scene; an orientation a; and a 2-D position G.

EVOLUTION PRINCIPLE

The evolution of the deformable object in the force field is performed as two separable processes. The first manages the local shape deformation of f_d under constraints with respect to the original shape f_o; the second manages the position with respect to the theory of material systems. In this section, we briefly sketch the processes that will be further detailed in the following sections.

The deformation of the object is done with respect to the reference form f_o. To do so, after having applied the external constraints of the image, we process the error of the deformable form f_d with respect to f_o in the (α, l) coordinate system.



Details of the deformation process.
Figure 6

In addition to deformation, the deformable object moves over the surface of the scene under the applied forces. Motion management, the second part of the evolution process, computes the orientation and the position using the laws of material systems [Remion96].

DEFORMATION PROCESS

In this part, we describe the deformation process to compute the form f_d at step $i+1$ considering the external and internal constraints. In order to deform the form f_d compared to the original form f_o, let us introduces a curve c_e such that

$$f_d = f_o + c_e$$

c_e can be regarded as the deformation of the form f_d compared to f_o, the reference form. To preserve similarity between f_o and f_d, the process of deformation will maintain smoothness of c_e through a regularization. Thus, a local external force applied to f_d will have an effect over some neighborhood. But, modifying the curve c_e implies that the curve $f_o + c_e$ may no longer satisfy the constraints for a form (see the properties section). Let us call this temporary curve c_d. The last step of this part of the deformation process will be a closure of c_d to get the form f_d. Fig. 6 highlights the different stages of the deformation process.

Deformation of the model

First, the deformation of the model is applied to the form f_d placed in the scene. To each point of the placed form, the various forces, summed each other, will create a reaction R_i. This reaction is interpreted as the displacement of the corresponding point. Finally, we compute the new form f_d, and then the error c_e.

Regularization

The point of regularization is to ensure a degree of resemblance between the current shape of the object described by the form f_d and its original shape described by the form f_o. The regularization is performed on the error curve c_e at order 1, elasticity, and at order 2, plasticity, according to the following formula

$$[K + k_v \cdot \mathrm{Id}] \cdot X_{i+1} = k_v \cdot X_i \qquad (5)$$

where K is the regularization matrix, k_v, the viscosity coefficient and the vectors X_i and X_{i+1} are the data at step i and at step $i+1$, before and after regularization. More information about this regularization technique is available in [Kass87]. The matrix K is a band-diagonal matrix that contains on each line

$$K_{i,i-2} = k_p$$
$$K_{i,i-1} = -4k_p - k_e$$
$$K_{i,i} = k_v + 6k_p + 2k_e$$
$$K_{i,i+1} = K_{i,i-1}$$
$$K_{i,i+2} = K_{i,i-2}$$

The coefficients k_e and k_p, which act on elasticity and plasticity respectively, make it possible to balance the action of regularization on each criterion. Only relative values of k_e, k_p and k_v are important and will modify the process of regularization and as a result the evolution of the shape of the object.

Adjustments

As we saw in the evolution principle section, the regularization does not lead to an expected form but to a curve. To restore the characteristics of this result, we must turn this curve into a form, applying corrections on it to satisfy the 2 conditions of a form expressed in the model section, Eq. 3 and Eq. 4. The constraints are applied to the angles and the lengths separately. We will adjust the angles first and then the lengths.

Angle Adjustment. The adjustment of the angles is distributed evenly over all the angles and is carried out by a simple shift of each angle by a single value,

$$\Delta\alpha = \frac{2\Pi - \sum \alpha_i}{n}$$

where $\forall i, \alpha_i = \alpha_i - \Delta\alpha$

We then obtain a form f_d which satisfies the condition Eq. 4.

Length Adjustment. To finish the process, it is necessary to adjust the lengths in order to close the curve c_d, thus satisfying the condition Eq. 3. To do so, we use the method of Lagrange multipliers [Luenberger69]. To describe the application of this method in our case, let us write $\hat{l}_i$ for the lengths obtained after regularization, l_i for the final lengths, and Δl_i for the correction to apply to each $\hat{l}_i$. We can then write the relation

$$l_i = \hat{l}_i + \Delta l_i \qquad (6)$$

Let us define the function θ to be minimized as the sum of the squares of the errors $\theta(\Delta l_i) = \sum \Delta l_i^2$

The constraints to be applied are those expressed by the system of Eq. 3

$$\varphi_1(\Delta l_i) = \sum_{i=1}^{n-1} l_i \cdot \begin{pmatrix} \cos\beta_i \\ \sin\beta_i \end{pmatrix} = 0$$

where $\beta_i = \beta_{i-1} + \alpha_i$ and $\beta_0 = 0$

The Lagrange method will solve the problem by calculating the Δl_i starting from

$$\Phi(\lambda_1, \lambda_2, \Delta l_i, \forall i) = \theta + \lambda_1\varphi_1 + \lambda_2\varphi_2$$

where λ_1 and λ_2 are the multipliers. From the system with $n+2$ equations defined as follows,

$$\begin{cases} \dfrac{\partial\Phi}{\partial\Delta l_i} = 2\Delta l_i + \lambda_1\cos\beta_i + \lambda_2\sin\beta_i = 0 \\ \varphi_1 = 0 \\ \varphi_2 = 0 \end{cases}$$

Solving the system gives us the Lagrange multipliers λ_1 and λ_2. The calculation of each value Δl_i enables us to correct of all the lengths according to Eq. 5.

OBJECT PLACEMENT

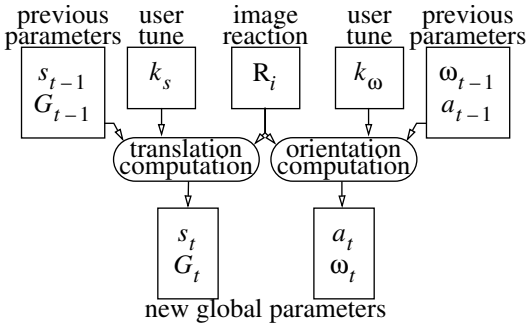
To be in accordance with the external forces, the movement of the object during the animation process must respect the laws of solid mechanics. In our application, in spite of the fact that our form is being deformed, we apply the principle of material systems [Remion96] where the center of gravity and the moment of inertia pertain to translation and rotation respectively. To calculate the parameters a and G, we apply the Newtonian laws of motion, expressed as

$$m\frac{\partial^2 X}{\partial t^2} + k\frac{\partial X}{\partial t} = F$$

At each iteration, the two global parameters of the object are computed and used to place its deformed form in the 2-D plane. The general outline of placement is described in detail in Fig. 7.

Evolution of G

A global force R, the sum of all forces R_i applied to any point of the form in the scene, will be applied globally on the object (independently of the local deformation). The objective is then not only to find the new position G_t of the center of gravity but also to calculate the new value of its translation speed s_t. The Newtonian laws give the following differential equation (ode), according to the mass m of the system and the viscosity coefficient k_s



Detailed scheme of the global evolution.
Figure 7

$$m\dot{s_t} + k_s s_t = \mathrm{R}$$

where $\dot{s_t} = ds_t/dt$

The solution is given by

$$s_t = \frac{\mathrm{R}}{k_s} + \left(s_{t-1} - \frac{\mathrm{R}}{k_v}\right)\exp\left(-\frac{k_s t}{m}\right)$$

with $v_0 = 0$ and $G_t = G_{t-1} + s_t t$

Thus, it will be easy to place the object at the correct position by translating the center of gravity of the evolving form G_{t-1} to G_t.

Evolution of a

Each force R_i applied to each point of $p(f_d)$ creates a torque T_i which will tend to turn it around its center of gravity. The sum T of all the T_i represents the global torque applied. The goal is not only to find the orientation a_t of the object but also to calculate the angular speed ω_t. The Newtonian laws determine the evolution from the following ode according to the mass m of the system and the viscosity coefficient k_v

$$m\dot{\omega}_t + k_\omega\omega_t = \mathrm{T}$$

where $\dot{\omega}_t = d\omega_t/dt$

ω and t being the angular speed and the applied torque respectively.

The solution is given by

$$\omega_t = \frac{\mathrm{T}}{k_\omega} + \left(\omega_{t-1} - \frac{\mathrm{T}}{k_v}\right)\exp\left(-\frac{k_\omega t}{m}\right)$$

with $\omega_0 = 0$ and $a_t = a_{t-1} + \omega_t t$

The computed angle a represents the orientation of the vector corresponding to the principal axis of inertia of the form. This axis corresponds to the eigenvector associated with the greatest eigenvalue of the matrix of inertia expressed relative to the center of gravity

$$I = \sum_{i=0}^{n-1} \begin{bmatrix} y_i^2 & x_i y_i \\ x_i y_i & x_i^2 \end{bmatrix}$$

Thus, it will be easy to orient the form correctly by rotating the object from orientation a_{t-1} to a_t.

RESULTS

The animations presented vertically in Fig. 8 show some results of object deformations. Parts (a.) and (b.) propose an interesting usage of our deformable model, where the initial shape is quite different from the rest shape of the object. As a result, it is a kind of morphing process from one shape to another. Parts (c.), (d.) and (e.) present the evolution of an object in various force fields attracted by a gravitational force. We can appreciate the realism of the deformation. The design of the objects' shape is very easy to define because of the simplicity of the model. The force field and other interactions are also very easy to build. The implementation program of the process has been developed in Java and is available as a demonstration applet at

http://www.essi.fr/~abram/deformable/CLObjects.html

CONCLUSIONS

For the 2D model, some work remains to implement and test all the forces we plan to incorporate. The gobal results we obtained with this method are satisfactory but some adjustements have to be performed to ensure stability and robustness of the deformation process.

The work will move to 3D and 4D. The next stage, the 3D model, will be developed starting from the experience gained with the 2D model presented here.

REFERENCES

[Baraff92] David Baraff and Andrew Witkin. Dynamic simulation of non-penetrating flexible bodies. In *Computer Graphics26*, pages 303–308, 1992.

[Barr84] Alan H. Barr. Local and global deformations of solid primitives. In *Computer Graphics(18) ACM Siggraph*, 1984.

[Desbrun94] Mathieu Desbrun and Marie-Paule Cani-Gascuel. Peau implicite active pour l'habillage de modèles très déformables. In *AFIG'97, Journées de l'Association Fran,caise d'Informatique Graphique*, Rennes, Décembre 1997.

[Gascuel93] Marie-Paule Gascuel. An implicit formulation for precise contact modeling between flexible solids. In *Computer Graphics, Proceedings of SIGGRAPH'93*, pages 313–320, Anaheim, August 1993.

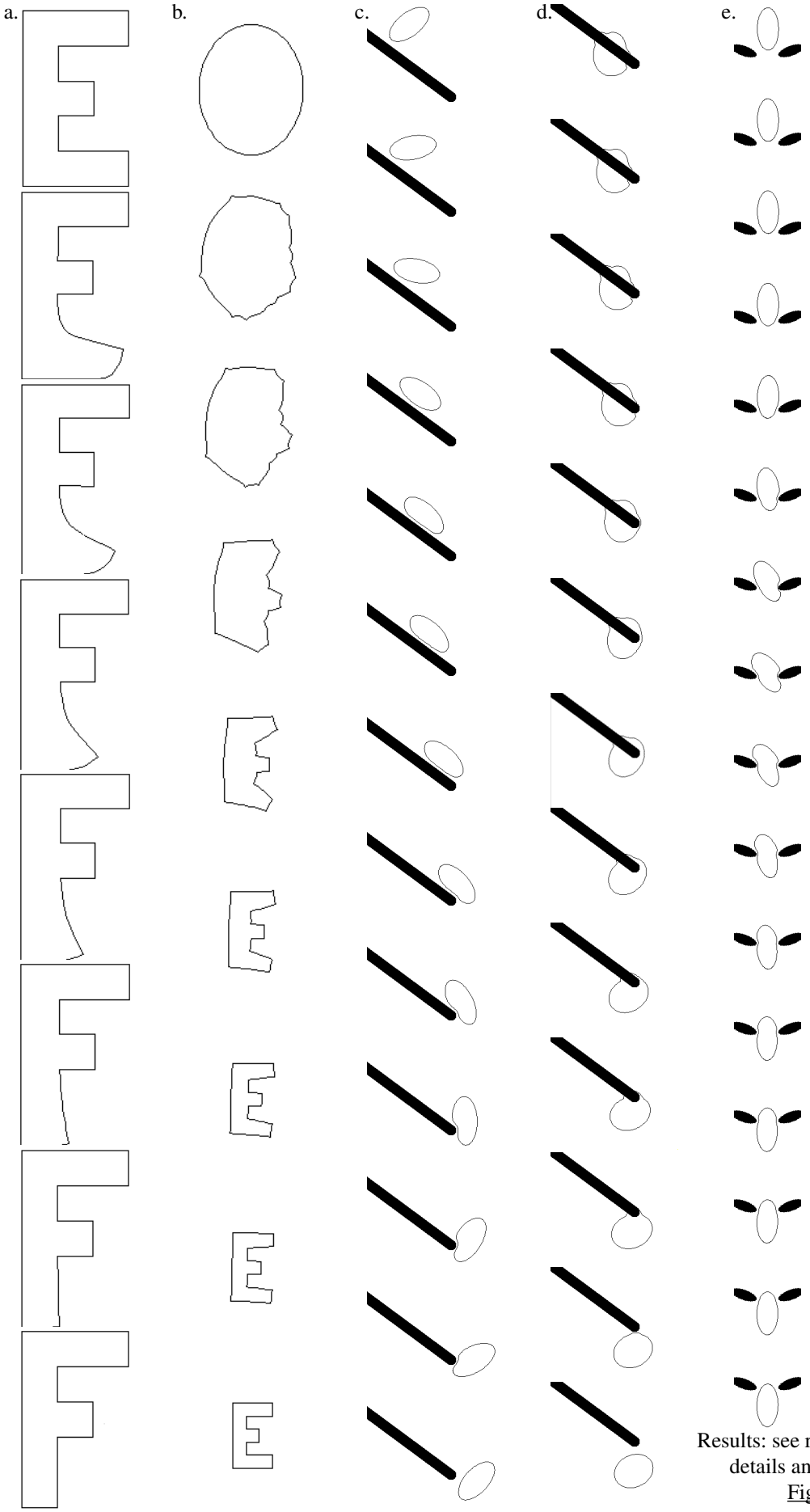
[Provot95] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proceedings Graphics Interface'95*, pages 147–154, 1995.

[Sederberg93] Thomas W. Sederberg, Peisheng Gao, Guojin Wang, and Hong Mu. Shape blending: An intrinsic approach to the vertex path problem. In *Computer Graphics*, pages 15–18, 1993.

[Szeliski93] Richard Szeliski, David Tonnesen, and Demitri Terzopoulos. Modeling surfaces of arbitrary topology with dynamic particles. In *Transactions of the IEEE conference on Computer Vision and Pattern Recognition*. New York City, NY, June 1993.

[Terzopoulos88] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *Proceedings ACM SIGGRAPH'87 Conference*, pages 118–128, Anaheim, July 1988.

[Terzopoulos94] Demetri Terzopoulos, Xiaoyuan Tu, and Radek Grzeszczuk. Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, 1(4):327–351, December 1994.



Results: see result section for details and discussion

Figure 8