TR－H－044

Physical Models for Edge Finding: Snakes

Christophe Begin

1993. 12. 22

ATR人間情報通信研究所

〒619-02 京都府相楽郡精華町光台 2-2　☎07749-5-1011

ATR Human Information Processing Research Laboratories

2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

Telephone: +81-7749-5-1011
Facsimile: +81-7749-5-1008

Physical Models for Edge Finding: Snakes

Christophe Begin
ATR Human Information Processing
Research Laboratory
2-2 Hikari-dai, Seika-cho, Soraku-gun
Kyoto 619-02 Japan

Abstract

A snake is an elastic spline with dynamics determined by internal and external forces - it is a physical model for a one-dimensional curve. In this report, we use a snake to find and track features of interest in images by constructing forces which guide the snake to those features. Construction of these forces requires consideration of the "material" properties of elastic curves and the resulting dynamics and of the image processing required to generate an external force related to features of interest.

Acknowledgements

This work constitutes the internship report of Christophe Begin, who would like to express his gratitude to all the people in the ATR Human Information Processing Laboratory, who have helped him during this internship and contributed to a very friendly work atmosphere. I also wish to express sincere thanks to Dr. Ed Gamble for his continual support and very helpful advice during the length of my stay at ATR.

Contents

List of Figures

Chapter 1

INTRODUCTION

Within the past ten years, computer graphics and image processing research have advanced tremendously due to rapid technological developments. These two research areas share a common body of knowledge [1]:

- *Graphics* deals with the generation of images.Currently this seems to be an area of fast growth. The computational effort required to produce displays (such as production of plots of functions, composition of displays for the common computer game, and production of the scenes used in flight simulators) varies significantly, depending on the task. The term *interactive graphics* refers to devices and systems that accept input from the user to produce a graphical display (the simplest example being the drawing of a line between two points on the screen that the user specifies).

- *Image processing* deals with analysis of images. It can entail noise removal, data compaction, contrast enhancement techniques. Sometimes it desirable to apply more drastic transformations. For example, an image with a wide range of illumination may be reduced into an image where one sees only two levels of illumination.

Clearly, there exist overlapping interests among theses two areas of research.

In the past few years, a new focus of research in computer graphics, known variously as *engineering animation or physical-based modeling* has begun to emerge. The notion is that rendering objects and scenes, however realistically, is only a part of the game. Human and animal figures, mechanical parts and assemblies have physical attributes and obey the laws of classical mechanics. The lack of such attributes is often painfully obvious in animated sequences that account only for geometric properties; animation that incorporates physical properties seems aesthetically and psychologically more satisfying.

Modeling based on physical principles is establishing itself as a potent technique in computer graphics. Physical-based models, while computationally more complex than many traditional models, offer unsurpassed realism in the modeling of natural phenomena. This is a critical advantage for computer animation. Conventional animation is kinematic; objects are set into motion by describing the positions of their geometric components at each instant in time, usually aided by key-frame interpolation techniques. By contrast, in the dynamic approach to animation, forces are applied to objects while standard numerical procedures generate position through time in accordance with Newton's laws. Newton's laws are applied based on the physical properties attributed to the object's model.

A snake is a one-dimensional model of an elastic curve that is influenced by internal and external forces. The internal forces are elastic forces which give the curve a resistance to deformation. The external forces can be specified arbitrarily and are the primary mechanism used to produce a desired behavior. As an example of interest here, image processing techniques can be used to produce a force field within which the snake moves towards object boundaries. Moreover it could track this object through time because the snake is a dynamic model with a mass, so it has a inertial motion like real objects in a sequence of images.

The basic theory of the snake has been developed in recent years[2, 3, 4]. It comes in fact from the theory of computer animation and graphics - the authors of this theory were interested in virtual reality, so they created some physical models that can be used for image processing. It is a animation which receive both nonpictural information and pictural information. So we can no longer say that it is only image processing, and it is certainly not only computer graphics because of its uses. The snake is a flexible object in a physical environment. The energy of and forces on the snake depend on where the snake is placed and how its shape changes locally in space.

The idea is to have the snake lock on to the features of an image structure by minimizing an integral measure which represents the snake's total energy or in fact by computing its displacement in a force field. For such computation, Terzopoulos[2] used an energy-oriented algorithm; within this report we used forces-oriented algorithm.

Chapter 2

BASIC SNAKE BEHAVIOR

2.1 Introduction

In this chapter we will explain how to "build" a snake. So in this chapter we will discuss the physical part of this report. We have to understand what the snake is made of and what are the forces we will have to apply and how it will react to such forces as well as its internal constraints. In the first part of this chapter, we will talk about the space in which the snake will move. We will have to define the most useful frames to study the snake motion. Secondly, we will obtain the equations of motion for the snake which determine its shape and motion when it is submitted to external forces. The third part of this chapter will be dedicated to a deeper study of the elastic force which is the snake's cohesion force. Finally, we will show some examples of different types of snakes.

A snake is modeled as a spline under the influence of external and internal forces. The external forces push the snake in some direction and the internal forces try to return it to its original shape (unless this "original" shape can be changed through time). The external forces can be specified in any way - for example, interactively by the "user".

In this report, the behavior of the snake is based only on the deformation of its elastic model from its original shape. This simplification is discussed further in Chapter 5.

2.2 Inertial Frame and Body Frame

Snakes are represented parametrically as:

$$v(s,t) = (x(s,t), y(s,t)) \tag{2.1}$$

where t is a time coordinate, s is the snake's intrinsic coordinate with domain of Ω=[0,1], $x()$ and $y()$ are coordinates in the snake's two-dimensional Euclidean space, and $v()$ is a time-varying, vector-valued function of the material coordinates (*not* a velocity).

If we define a reference frame ϕ, whose origin coincides with the snake's center of mass as

$$c(t) = \int_\Omega \mu(s)v(s,t)ds$$

where μ is the mass density of the deformable snake and Ω is the domain of integration, then we can express the positions of the mass points relative to ϕ as

$$q(s,t) = r(s) + e(s,t) \tag{2.2}$$

which is composed of a reference component, $r()$, and an elastic component, $e()$ given by

$$\begin{aligned} r(s) &= [r_x(s), r_y(s)] \\ e(s,t) &= [e_x(s,t), e_y(s,t)] \end{aligned} \tag{2.3}$$

for the deformable snake model.

The inertial 'body' frame ϕ translates along with the deformable body. It could rotate as well, however, that changes many things particularly the equations of motion; we will discuss the effect of rotation later in the report. In what follows, the body frame ϕ is only permitted to translate as shown in Figure 2.1.

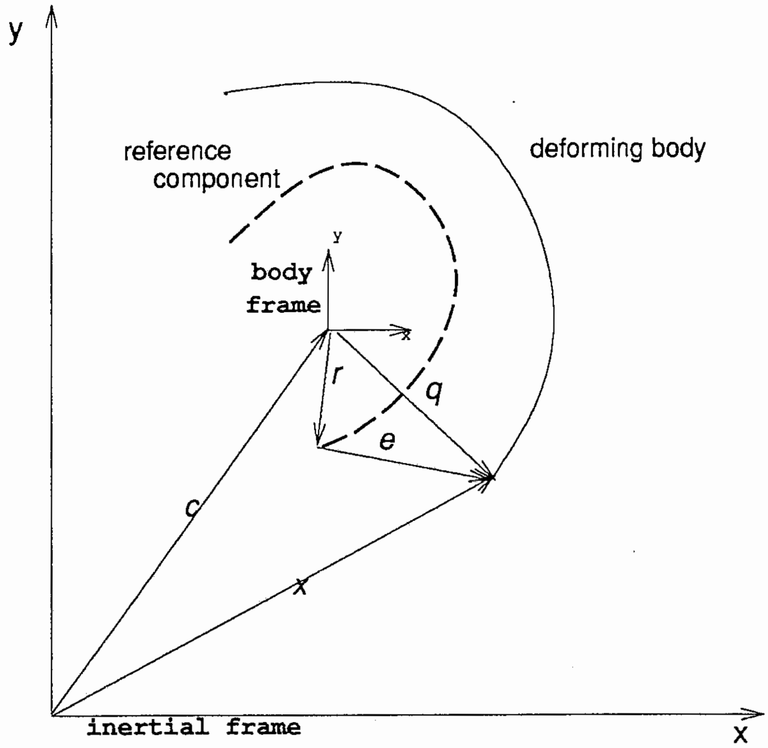


Figure 2.1: *Geometric Representation: A snake's shape is decomposed into reference and deformation components.*

2.3 Equations of Motion

A deformable model is described completely by the positions $v(s,t)$, velocities $\dot{v}(s,t)$, accelerations $\ddot{v}(s,t)$ of each of its mass elements. In the inertial frame ϕ, the Lagrange equations of motion governing $v(s,t)$ take a relatively simple form :

$$\mu\ddot{v} + \gamma\dot{v} + \delta_v\xi = f \tag{2.4}$$

where $\mu(s)$ is the mass density, $\gamma(s)$ a damping factor and $f(x,t)$ is the external force. $\delta_v\xi$ represents the internal force or elastic force which depends only on $v()$ and its partial derivatives. Equation 2.4 is a partial differential equation in s and t.

During motion the net external forces $f(s,t)$ balance dynamically against the following three factors:

- the inertial force due to the mass density $\mu(s)$ - this is a force to resist acceleration.
- the velocity-dependent damping force with damping density $\gamma(s)$ - this force dissipates the kinetic energy of the body's mass points as it moves through a viscous ambient medium.
- the internal elastic force $\delta_v \xi$ - this resists deformation of the snake away from its natural shape.

The elastic force is conveniently expressed as $\delta_v \xi$, a variational derivative of a deformation energy $\xi(x)$ associated with the model. The non-negative functional ξ measures the potential energy associated with an instantaneous elastic deformation of the body. Its value increases monotonically with the magnitude of the deformation.

Given this equation with appropriate conditions for v on the boundary of Ω and the initial shape or configuration $v(s, 0)$ and velocity $\dot{v}(s, 0)$, the problem becomes an initial boundary-value problem.

Using this formulation of Equation 2.4, there can be a numerical degeneration due to the fact that in order to increase rigidity, the values of the parameters must be increased. This increase makes ξ both more non-quadratic and non-linear. This problem can be avoided by decomposing v into the reference component r and the deformation component e of Equation 2.4. In that case, we obtain a purely quadratic elastic functional ξ. However, as mentioned previously, in this work the body frame can only translate in the inertial frame; thus there is no great difference between a new equation of motion and equation 2.4. Still, the notation is useful to express easily the elastic form as we saw in equation 2.4.

2.4 The Elastic Force

In this section, we will discuss the elastic force, ξ, which resists deformation and pushes the snake to recover its original shape. A deformation is termed "elastic" if, upon removal of all external forces, the shape returns to its undeformed original shape. The basic assumption underlying the constitutive laws of classical elasticity theory is that the restoring force (stress) in a body is a single-valued function of the deformation (strain) of the body and moreover that it is independent of the history of the deformation. The restoring forces intrinsic to the deformable models are expressed in term of a deformation potential energy [5]. An energy characterization is always possible for elastic models. As a generalization of the ideal spring, the elastic model stores potential energy during deformation, which is then released as it recovers the original shape.

The elastic force due to deformational displacement away from the original shape is a variational derivative of a elastic potential energy functional ξ. This is an integral of an elastic density E which depends on $e(s,t)$ and its partial derivatives. We can define $\xi(e)$ as:

$$\xi(e) = \int E(s, e, e_s, e_{ss}, \dots) \tag{2.5}$$

with two restrictions: 1) $\xi = 0$ when $e = 0$, and 2) ξ increase monotonically with increasing e.

At our disposal are a class of controlled-continuity generalized spline kernels. These splines are of the previous form with the potential energy defined by

$$E = \frac{1}{2} \sum_{m=0}^{p} \sum_{|j|=m} \frac{m!}{j_1! j_2! \dots j_d!} \omega_j(s) |\partial_j^m e|^2 \tag{2.6}$$

where $j = (j_1, j_2, \dots, j_d)$ is a multi-index with $|j| = j_1 + j_2 + \dots + j_d$, d is the material dimension of the model and the partial derivative operator is

$$\partial_j^m = \frac{\partial^m}{\partial_{s_1}^{j_1} \dots \partial_{s_d}^{j_d}}.$$

$E()$ is a weighted sum of partial derivatives of e of all orders up to p. The weights $\omega_j(s)$ are express the material properties of the deformable model. This is a general model capable of representing any material properties whatsoever. In practice only low-orders of p are used. In a subsequent section we illustrate the consequences of modifying these weights $\omega_j(s)$

In the material domain Ω, the variational derivative of ξ is

$$\delta_e \xi = \sum_{m=0}^{p} (-1)_m \Delta^m_{\omega_m} e \tag{2.7}$$

where

$$\Delta^m_{\omega_m} = \sum_{|j|=m} \frac{m!}{j_1! j_2! \ldots j_d!} \partial^m_j (\omega_j(s) \partial^m_j) \tag{2.8}$$

is a spatially-weighted iterated Laplacian of order m. In our problem d is equal to one because the snake is a simple one-dimension curve. The form of the elastic force simplifies to

$$\delta_e \xi = \sum_{m=0}^{p} (-1)^m \frac{\partial^m}{\partial s^m} (\omega_m(s) \frac{\partial^m e}{\partial s^m}). \tag{2.9}$$

Moreover, for the work described here there is no need for high precision in the snake behavior (we are in a discrete space so the complexity of the model could be a factor of errors). So we employed a second order ($p = 2$) controlled-continuity spline model. Thus the form for our elastic force becomes

$$\delta_e \xi = \omega_0(s) e(s) - \frac{\partial}{\partial s}(\omega_1(s) \frac{\partial e}{\partial s}) + \frac{\partial^2}{\partial s^2}(\omega_2(s) \frac{\partial^2 e}{\partial s^2}). \tag{2.10}$$

2.5 Examples

Before continuing with a discussion of our implementation it is useful to illustrate some examples of simple snake motion. We start with an oval moving in a 2D world. We will show now some examples of the different kinds of rigid or elastic bodies that we can obtain. We changed the the coefficients ω_0, ω_1 and ω_2 of the elastic force ξ to explore the effects of these parameters.

In the five examples, there is only one external force which is the gravitational force (a constant force along the negative y-axis). We gave a initial velocity to the ball in the negative x direction to improve the viewability of the results.

Figure 2.2 shows a very soft ball that reacts more like a viscous fluid after it reaches the wall at $x = 0$. For this figure, all the parameters were nearly zero.

Figure 2.3 shows a soft ball. In this example, $\omega_0 = 0$ and the others elastic

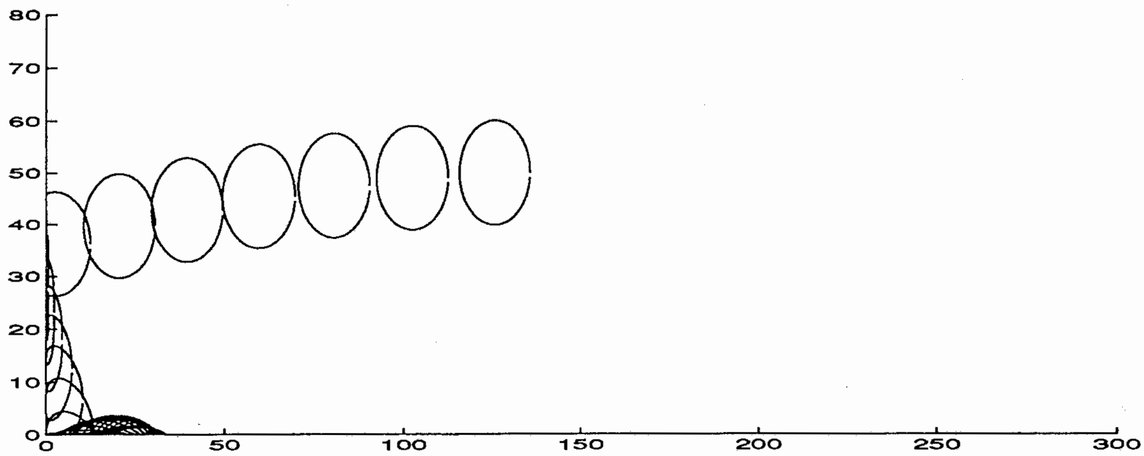


Figure 2.2: *A very soft ball.*

parameters are greater than in the previous example. However, in our model, the ball need not have a constant volume; it is compressible and we see the consequences in this figure.

Figure 2.4 shows a ball that is harder than the previous one; upon hitting the wall, the shape of the ball oscillates around its original shape.

Figure 2.5 shows a hard ball. There is not a great difference between the translational motion of the balls in Figures 2.4 and 2.5 except that the harder ball recovers its original shape more quickly.

Finally, in Figure 2.6 is different from the previous ones because we changed more parameters than solely the plastics parameters. This figure shows that not only the elastic parameters are important - further discussion regarding this is in Section 3.4.

In the this chapter we studied the physical aspects of snakes when submitted to two different types of forces: external forces and internal elastic forces. We derived an equation for the snake's motion under theses conditions.

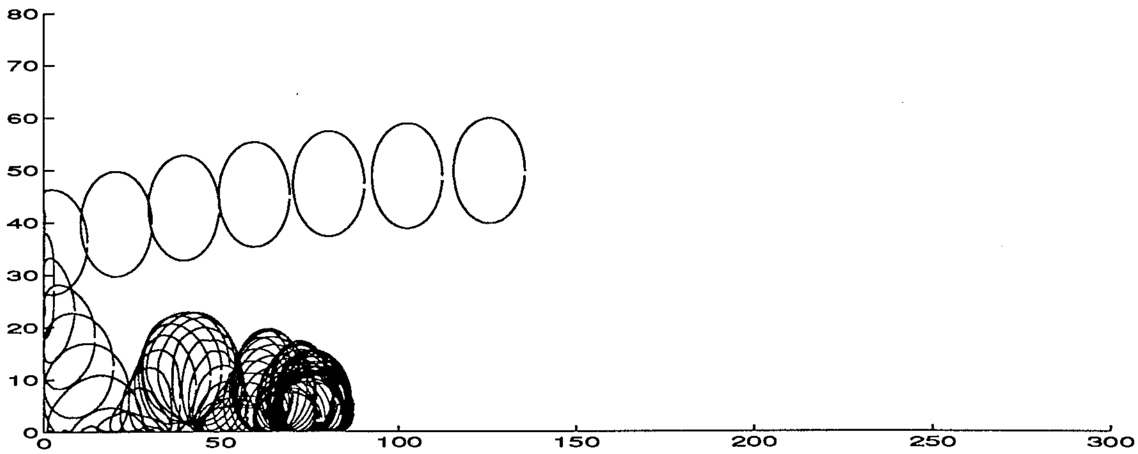


Figure 2.3: *A soft ball.*

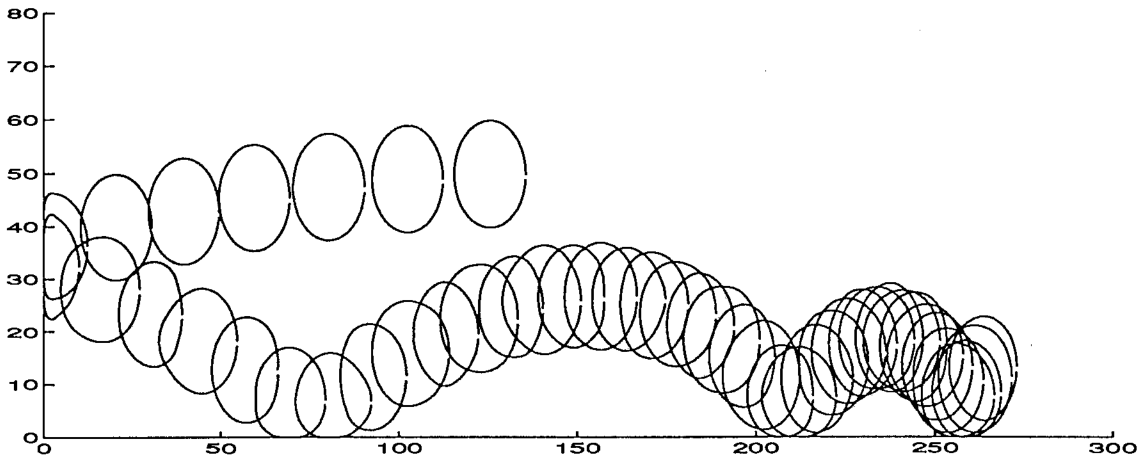


Figure 2.4: *A rigid ball.*

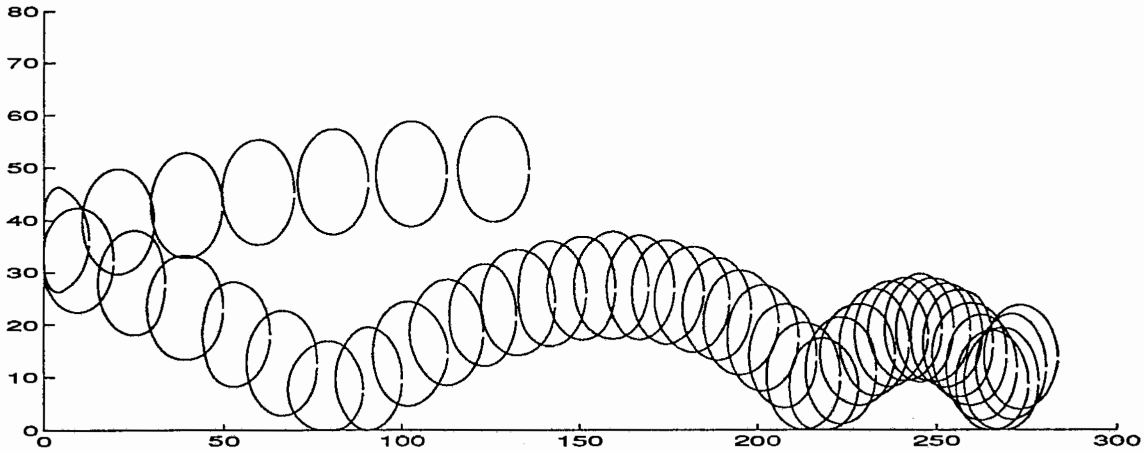


Figure 2.5: *A hard ball.*

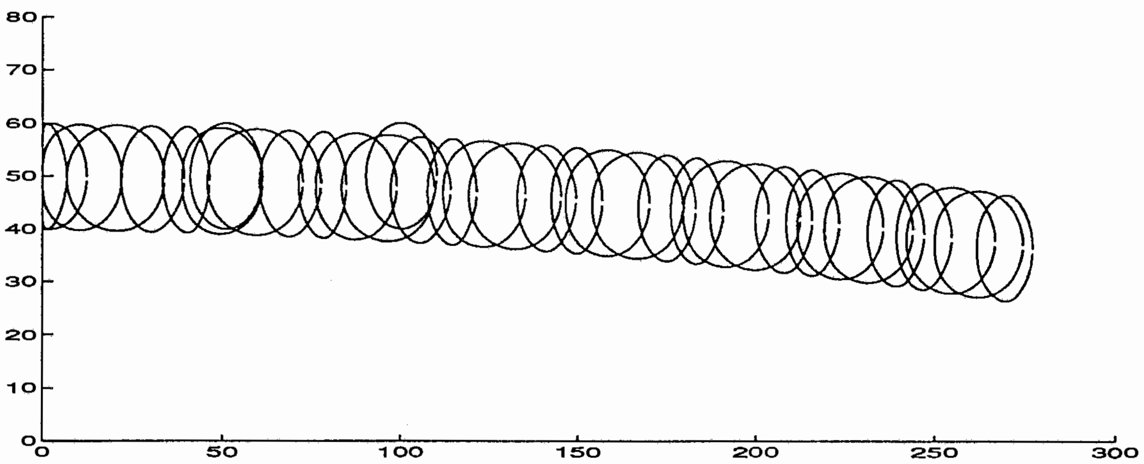


Figure 2.6: *A very hard ball.*

Chapter 3

IMPLEMENTATION

3.1 Introduction

The subject of the present chapter is the solution and implementation of Equation 2.4. In the first part, we will describe the method we used to semi-discretize the snake. It is a semi-discretization because even if the coordinates of the mass points are reals, in our case the snakes' external force is ultimately derived based on a discrete grid (see Chapter 4). Moreover we will consider the snake as a succession of interconnected mass points and not as a continuous elastic curve. After this important part, we will talk about the time-discretization. Because the snake is a dynamic model, we have to solve a dynamic equation to know the motion of the snake through the time. In this chapter, we will introduce some parameters (dynamic parameters, elastic parameters and the time parameter) which are preponderant in the snake's behavior - their study will be the subject of our third section.

3.2 Semidiscretization in Space

The goal of this section is to write the elastic force $\delta_e \xi$ of Equation 2.9 in a discrete form. Our first step in this section will be the discretization of the elastic force for one mass point, then we will introduce new coordinate vectors that will describe the positions of all the mass points along the snake. Our objective then is to express the elastic force acting on the whole snake as a matrix product of the deformation tensor and the position vector for the snake.

We illustrate the semidiscretization step using standard finite-difference approximations. The unit domain, $\Omega = [0; 1]$, of the curve is discretized as an equally-spaced distribution of N nodes. The internode spacing h is constant and is calculated from the original position of the snake. Furthermore, we calculate h as the average value of the different internode spacing; thus h is a constant in time and a constant in space as well. Note that, this is a significant approximation and we will discuss its effect later in this report in chapter 5. Nodes are indexed by integers n where $1 \leq n \leq N$. We approximate continuous functions of s and t by transforming s as $s = nh$; for Equation 2.4 and the external force of Equation 2.4 this approximation yields

$$\begin{aligned} r(s) &= r(nh) &&= r[n] \\ e(s,t) &= e(nh,t) &&= e[n](t) \\ f(s,t) &= f(nh,t) &&= f[n](t). \end{aligned} \tag{3.1}$$

The discretization in time is the subject of future section; So, for now, we will suppress the time dependence notation until section 3.3.

The discrete elastic force requires approximating the nodal variables $e[n]$ and, because we restricted our interest to a second-order approximation, the first and second derivatives of e (with respect to the material coordinates s). In order to compute theses derivatives, we define some differential operators :

- The forward first difference operator

$$D_1^+(e)[n] = \frac{e[n+1] - e[n]}{h}$$

- The backward first difference operator

$$D_1^-(e)[n] = \frac{e[n] - e[n-1]}{h}$$

- The central second difference operator - based on previous operators

$$D_2(e)[n] = D_1^-(D_1^+(e))[n].$$

These difference operators will be used to write the elastic force $\delta_e\xi$ in a discrete form. Recalling that for our model $p = 2$ and with the above difference operators, the elastic force that acts on the node n can be written as:

$$\delta_e\xi \approx \omega_0 e[n] - D_1^-(\omega_1 D_1^+(e))[n] + D_2(\omega_2 D_2(e))[n]. \tag{3.2}$$

We now collect the nodal variables into some N-dimensional vectors. These vectors describe the positions of all the snake's mass points. In our previous discussion on discretization, we were considering one mass point; now the goal is to obtain one equation of motion for the whole snake instead having N equations for each mass point. All underlined variables in the following will describe variables for the whole snake (not only one point but all the points of the snake).

$$\begin{aligned} \underline{e} &= (e[1], e[2], \cdots, e[N]) \\ \underline{v} &= (v[1], v[2], \cdots, v[N]) \\ \underline{r} &= (r[1], r[2], \cdots, r[N]). \end{aligned} \tag{3.3}$$

In our model in fact, $\underline{e}$, $\underline{v}$ and $\underline{r}$ are already vectors in a 2-dimensional space. So in order to represent the whole snake position and elongation, we will use, each time, two N-dimensional vectors which will represent the projection of the vectors $\underline{e}$, $\underline{v}$ and $\underline{r}$ on the x and y axes. So now we will have the vectors $\underline{e}_x$ and $\underline{e}_y$ to describe the snake's elongation , the vectors $\underline{x}$ and $\underline{y}$ to describe the snake's position and the vectors $\underline{r}_x$ and $\underline{r}_y$ to describe the snake's reference position in the body frame.

The discrete approximation of the elastic force may be written in the vector form $K_{xory}\underline{e}_{xory}$ where K_{xory} is an N-dimensional square matrix. To see the form of K, we can have a look on the projection of the discrete elastic force $\delta_e\xi$ (for one point) on the x and y axes. We have to remember that h is a constant; we call these projections

$\delta_x\xi$ and $\delta_y\xi$.

$$\delta_x\xi = \omega_0 e_x[n] - D_1^-(\omega_1 D_1^+(e_x))[n] + D_2(\omega_2 D_2(e_x))[n]$$

and

$$\delta_y\xi = \omega_0 e_y[n] - D_1^-(\omega_1 D_1^+(e_y))[n] + D_2(\omega_2 D_2(e_y))[n].$$

Further simplification of these elastic forces is possible if we assume that ω_0, ω_1 and ω_2 are constants in s and t and thus elastic properties of the snake are the same for each of its points. Then two projections of the elastic force can be written as a product of a matrix by the vector $\underline{e}$. We obtain $\underline{\delta_x\xi} = K_x\underline{e}_x$, $\underline{\delta_y\xi} = K_y\underline{e}_y$, and $K_x = K_y = K$ because K is the representation of the action of the different operators on the vectors $\underline{e}_x$ and $\underline{e}_y$.

With these definitions, K is a pentadiagonal-matrix that can be described as the weighting sum of I_N, J_N and L_N :

$$K = (\omega_0 - 2\omega_1 + 6\omega_2)I_N + (\omega_1 - 4\omega_2)J_N + \omega_2 L_N$$

where

- I_N is a N-dimensional unity matrix
- J_N is the N-dimensional square matrix

$$J_N = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

- L_N is the N-dimensional square matrix

$$L_N = \begin{pmatrix} 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \end{pmatrix}$$

And finally, the equation for whole snake's motion can be written in term of two components in x and y as

$$\begin{aligned} M\frac{\partial^2 \underline{x}}{\partial t^2} + C\frac{\partial \underline{x}}{\partial t} + K\underline{e}_x &= \underline{f}_x \\ M\frac{\partial^2 \underline{y}}{\partial t^2} + C\frac{\partial \underline{y}}{\partial t} + K\underline{e}_y &= \underline{f}_y \end{aligned} \tag{3.4}$$

where the discrete mass density variables $\mu[n]$ have been replaced by a diagonal N-dimensional square matrix M with the $\mu[n]$ variables as diagonal components and respectively where the discrete damping density variables $\gamma[n]$ have been replaced by a diagonal N-dimensional square matrix C with the $\gamma[n]$ variables as diagonal components and where $\underline{f}_x$ and $\underline{f}_y$ are each N-dimensional vectors of the external force projected on the x-axis and y-axis respectively for each mass point on the curve.

3.3 Numerical Integration Through Time

In order to simulate the dynamics of this model, we integrate the semidiscrete system through time. Dividing an open-ended interval from $t = 0$ into time steps Δt, the integration procedure computes a sequence of approximations at $\Delta t, 2\Delta t, 3\Delta t, \ldots, t, t+\Delta t, \ldots$. Each time step requires the solution of our two algebraic equations (Equation 3.5) for x and y to describe the complete motion of the snake in the reference frame.

We used the following two discrete-time approximation for the first and second temporal derivative

$$\frac{\partial z}{\partial t} \approx \frac{z(t) - z(t-\Delta t)}{\Delta t}$$

$$\frac{\partial^2 z}{\partial t^2} \approx \frac{z(t+\Delta t) - 2z(t) + z(t-\Delta t)}{\Delta t^2}$$

and substitute these approximations into Equation 3.5 to obtain the two procedures

$$\begin{aligned}
(\frac{1}{\Delta t^2}M)\underline{x}_{t-\Delta t} &= \underline{f}_x + \Theta_1 \underline{x}_t + \Theta_2 \underline{x}_{t-\Delta t} - K\underline{e}_x \\
(\frac{1}{\Delta t^2}M)\underline{y}_{t-\Delta t} &= \underline{f}_y + \Theta_1 \underline{y}_t + \Theta_2 \underline{y}_{t-\Delta t} - K\underline{e}_y
\end{aligned} \tag{3.5}$$

where

$$\Theta_1 = \frac{2}{\Delta t^2}M - \frac{1}{\Delta t}C$$

and

$$\Theta_2 = \frac{1}{\Delta t}C - \frac{1}{\Delta t^2}M.$$

3.4 Parameters

We have to explain the real role of all the parameters in the snake's motion equations. We saw in the second chapter that the influence of these parameters is significant because modifications in the parameters produce very "hard" or "soft" snakes. Understanding the parameters is important in order to produce or design snakes with desired behavior.

3.4.1 The Elastic Parameters

The elastic parameters control the material properties of the snake:

- $\omega_0(s)$ describes an elastic force which links each mass point to its original position in the body frame. You have to imagine that there is a little spring between the mass point of the body and its reference position. We can't say that this force is really the main force that characterize the elastic behavior of the body because there is no connection in the motion of two different mass point. They all react independently.

- $\omega_1(s)$ controls the tension along the curve; it makes the snake act like a membrane.

- $\omega_2(s)$ controls the rigidity along the curve; it makes the snake act like a thin-plate. Setting $\omega_2(s_0) = 0$ permits a *tangent* discontinuity to occur at s_0 - it allows the snake to develop a corner at s_0. Moreover setting $\omega_1(s_0) = \omega_2(s_0) = 0$ permits a *position* discontinuity where the snake can (almost) break.

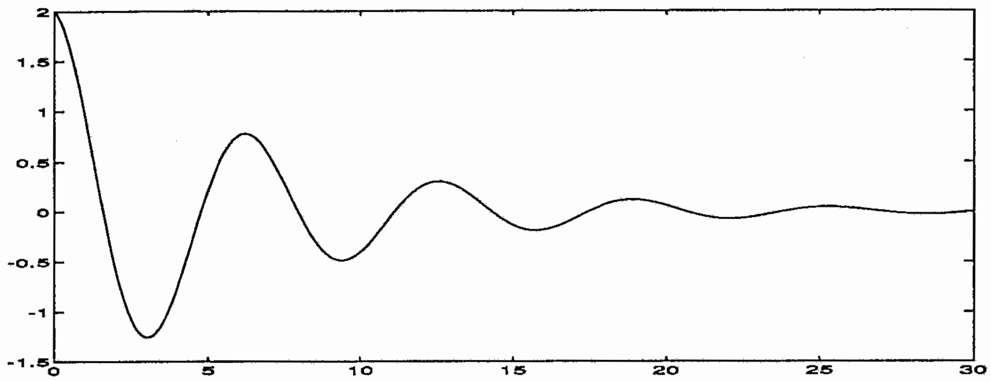
In the model we implemented, the elastic parameters had constant values. We saw that even with these constant values, the snake produced reasonable behavior. It could be a significant improvement to permit these parameters to be real function of s instead just parameters and also to allow these parameters to change values through time.

3.4.2 The Mass and Damping Densities Coefficients

The mass and damping coefficients are very highly correlated. $\mu(s)$ is the mass density of the snake which is responsible of the inertial force and $\gamma(s)$ the damping density which is responsible of the velocity-dependent damping force. To see these correlations, we will illustrate a simple model of a damped spring, $m\ddot{u} + c\dot{u} + ku = 0$, with free oscillations. This study is very simple and well known but we have to keep in mind that the coefficients are linked[6]. In this simple example, the oscillations are free but, in fact, our problem is, when and how does the spring (or the snake) reach its stable position?

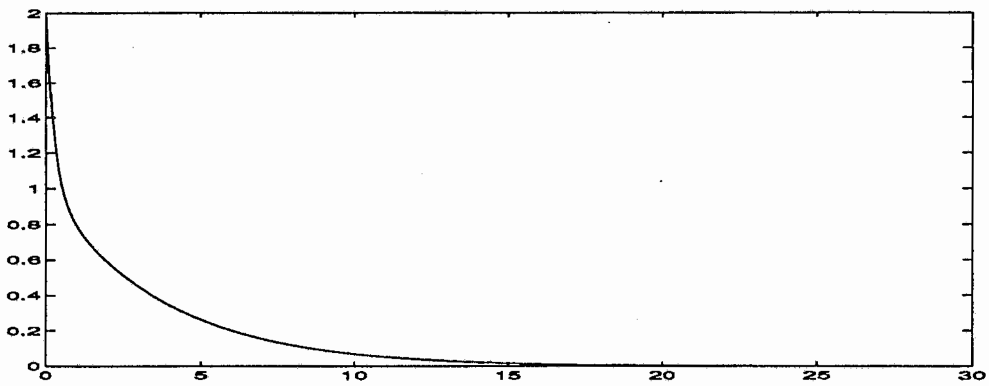
There are three possible types of motion for the spring. For all the examples, we have $m = k = 1$.

- When $c^2 < 4mk$: Underdamping ($c = 0.3$)



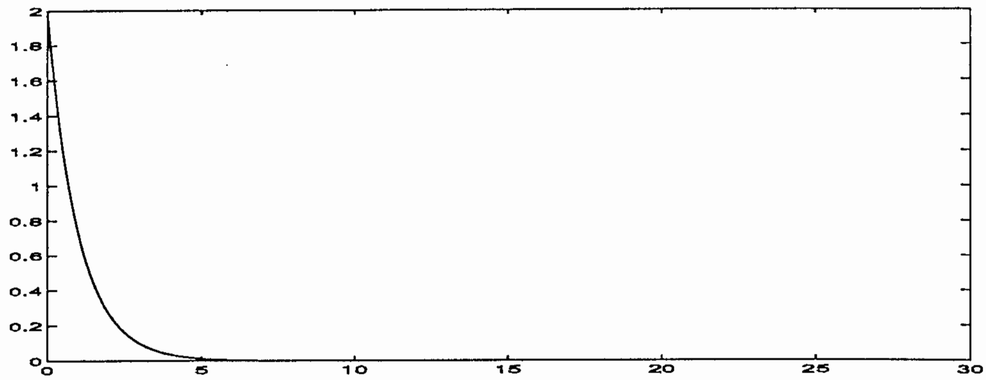
This produces lots of oscillations around the stable position. Given any displacement from its original shape, our snake would oscillate for a long time about the original position.

- When $c^2 > 4mk$: Overdamping ($c = 4$)



Now there is no oscillation, but the spring acts like it is moving in oil or molasses. Consequently it does not reach the stable position very quickly.

- When $c^2 = 4mk$: Critical damping ($c = 4$)



Here we have the most convenient behavior; the spring goes straight to equilibrium (and gets there at $t = \infty$).

If you compare the drawing of the critical damping and the drawing of the over-damping, you can see that with critical damping the spring goes near its stable position more quickly. So from all theses examples, we can say that the best type of motion for the snake is a critically damped behavior. But the problem is that the elastic force is not a constant (neither in time nor in space), so we have to compute the best damping coefficient for each point at each time. We did not do it, but it could be an future improvement.

3.4.3 The Time Parameter

Δt is the time parameter in our discrete equations of motion. This parameter is very important; using a large t value can lead to instability in the relaxation procedure while a small value will slow the overall time to equilibrium of the relaxation.

Chapter 4

IMAGE

4.1 Introduction

The two first parts of our report were oriented towards graphics computation because we wanted to display an elastic curve and make it move in a force field. But now, we have to make the snake useful in an image processing task and thus we need a force field that attracts the snake to salient features in an image. We now enter a totally different part of our report, because we will speak about image processing. We have to link the model to something physical, so we have to find something in an image that could replace a force (or, as Terzopoulus used, an energy). In the physical world, there are "forces" between two (or more) objects because there are contrasts and similarities between objects. In this chapter we will try to define an appropriate force derived from an image and then show the results when theses forces are applied to a snake.

4.2 Force Field

In this part our approach will proceed in two steps: first, we will find a force for one-dimension; then as a second step, we will try to apply our one-dimensional solution to the two-dimensional problem faced in images.

4.2.1 One-Dimensional Problem

In this part, we will begin with a very simple example. Lets have a signal $S(x)$ where $1 \leq x \leq 100$ and

$$S(x) = \begin{cases} 0 & \text{if } x \leq 50 \\ 1 & \text{if } x > 50 \end{cases}$$

First of all, we have to smooth this signal; we cannot use a signal with such a discontinuity. A suitable filter to smooth a signal is the Gaussian distribution:

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-x^2/2\sigma^2}.$$

There are two good reasons for using this filter. First, it smoothes the image thus the effects of noise are diminished[8] and the influence of such discontinuities as in our example are almost suppressed. The space constant of the Gaussian, σ, is used to modify the smoothing (the larger the value of σ is, the more smoothed will be the signal). And second, the Gaussian is separable and, due to the central limit theorem, we can decompose it into many smaller functions.

There are many ways to smooth a signal with a Gaussian .

- The Fourier Transform. According to the property of the convolution, smoothing a signal with a Gaussian can be computed by taking the Fourier transform of the input signal and of the Gaussian and then multiplying both and taking the inverse fourier transform of the result. For our present problem, it is not the simplest way to do.

- Direct Convolution. We convolve directly the input signal with the Gaussian (discretized). That is the simplest way to smooth an image with a Gaussian but it is also the longest in time. So even if here (in the one-dimensional problem) time is not really a problem, because there are only one hundred points, we won't use this method.

- Binomial Convolution. It is an iterated application of the mask [1/4 1/2 1/4] along the x-axis on the input signal to simulate the smoothing with a Gaussian.

This solution is only an approximation but as you could see one the figure below, it work rather well when the number of iterations is sufficient.

There is another small problem to consider with convolution to produce a Gaussian. The points at the boundaries $S(1)$ and $S(100)$ have to be convolved. But there are no points $S(0)$ and $S(101)$. In order to perform the convolution we chose 'mirror' boundary conditions and effectively added two points as $S(0) = S(1)$ and $S(101) = S(100)$.

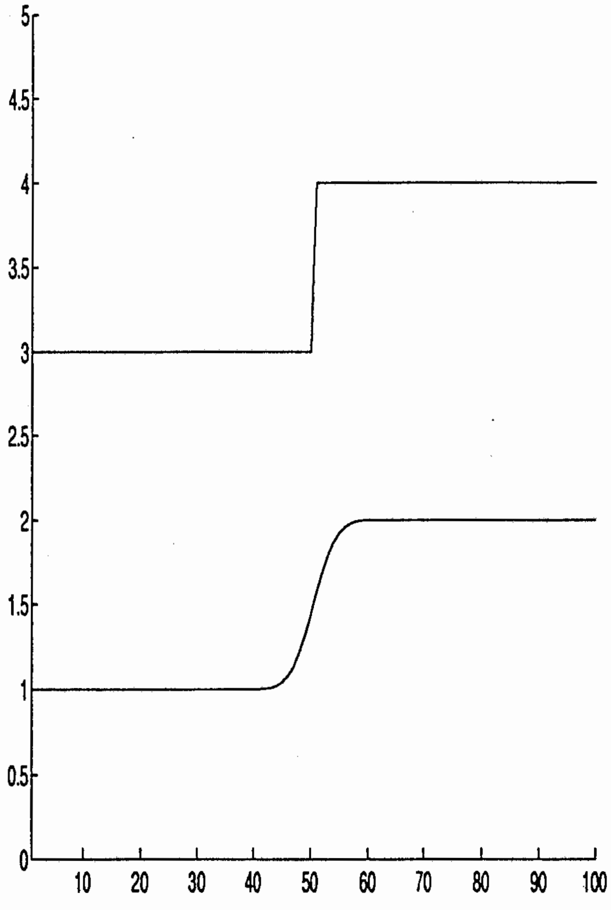


Figure 4.1: *Binomial convolutions to produces a Gaussian. Top - the input signal. Bottom - the smoothed signal (we applied twenty times the mask [1/4 1/2 1/4]).*

Now we have a smoothed signal, we have to find an appropriate force function. The force has to pull the snake toward the edge, so the force function must satisfy certain conditions that are:

$$\begin{cases} F(x) > 0 & \text{if } (50-\delta) < x < 50 \\ F(x) < 0 & \text{if } 50 < x < (50+\delta) \\ F(x) = 0 & \text{elsewhere.} \end{cases}$$

We introduced the parameter δ because we don't want the force field to be too strong. If a force of the same sign (positive or negative) is applied to the snake during too long time, the velocity of the snake increases far too much and so it would pass the equilibrium point (where the force is nulls) because of its inertia leading to excessive oscillations.

The second derivative of the smoothed signal seems to be perfect for our objectives.

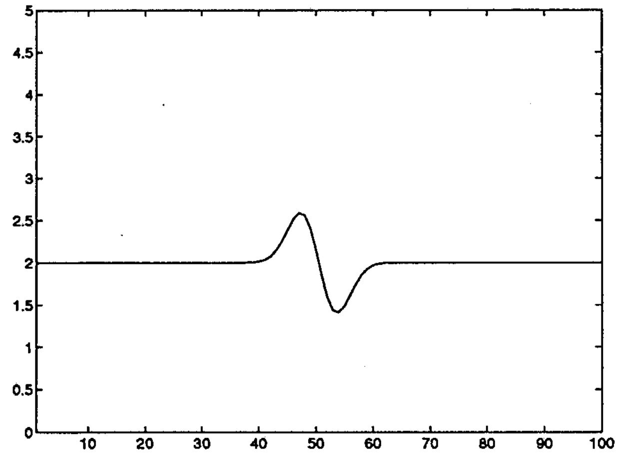


Figure 4.2: *The second derivative of the smoothed signal*

However our step function example was to simple. If the input signal is different as in

$$S(x) = \begin{cases} 0 & \text{if } x \leq 35 \\ 1 & \text{if } 35 < x < 65 \\ 0 & \text{if } x \geq 65 \end{cases}$$

and we apply the smoothing and the second derivative we obtain the result shown in Figure 4.3.

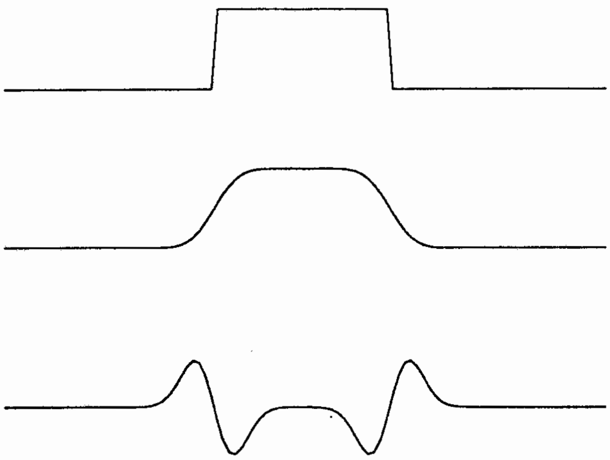


Figure 4.3: *A Box input. Top - the input signal. Middle - the smoothed signal. Bottom - the second derivative.*

It is obvious from Figure 4.3 that our force field does not work because the force field is inverted for the second edge. In our first example, there was an edge for a quick increase of S, but now there is also an edge for a decrease. The only difference in mathematical term is: for the first edge the first derivative is positive, and for the second one the first derivative is negative. So we can solve this problem by taking the absolute value of the first derivative and so an edge will be equivalent of a positive first derivative.

If F is the force we want obtain, then we can write:

$$F = \nabla(|\nabla(G * S)|)$$

where ∇ is the gradient which is equivalent to the first derivative in fact, G is the Gaussian and S the input signal.

Figure 4.4 shows the different steps from the input signal to the force. So now we solved the one-dimensional problem, we have to apply this type of resolution to the two-dimensional problem.

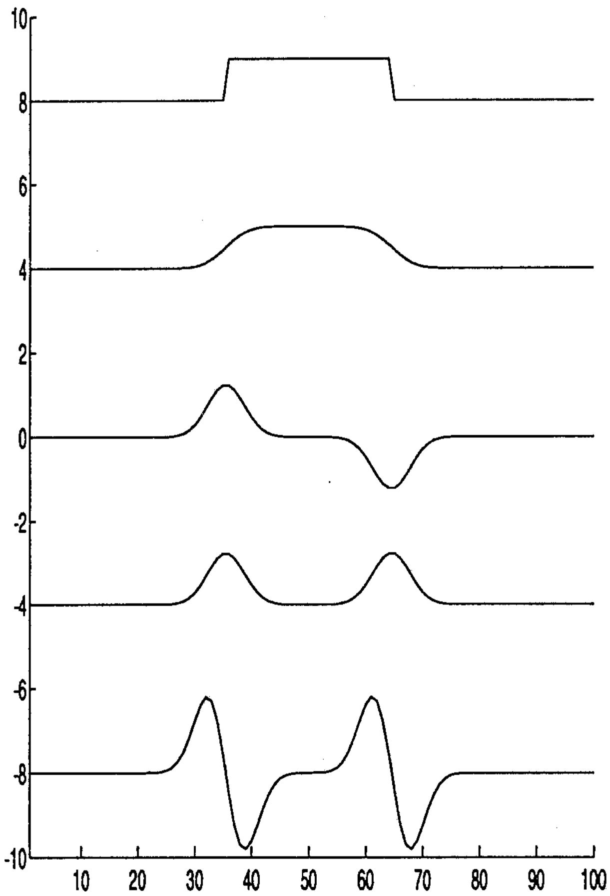


Figure 4.4: *A 'Box' input and the resultant force. From top to bottom: the input signal, the smoothed signal, the first derivative, the absolute value, the resultant force.*

4.2.2 Two-Dimensional Problem

The equations of the snake's motion require two different forces: one for the x-axis, and one for the y-axis. But in fact theses two forces are very similar in their construction. So there is no need to develop separate explanations for their construction. We will use the same algorithm for the one-dimensional problem to build our forces, the only difference will be in the dimension of the problem: we don't use vectors anymore, but a matrix. The input signal (the image) is a matrix with elements that are the pixel gray-level values. We will describe in this section the successive steps to build the x-force (we use the same operations to build the y-force).

In order to smooth the image, we will use the binomial convolution previously discussed based on the iterated application of the mask [1/4 1/2 1/4] along the x-axis. In order to compute the gradient ∇_x of the smoothed image, we used the following 3X3 mask :

$$\begin{bmatrix} 1/4 & 0 & -1/4 \\ 1/2 & 0 & -1/2 \\ 1/4 & 0 & -1/4 \end{bmatrix}$$

A similar mask for ∇_y can be easily found by rotating the above mask by 90 degrees. Theses masks compute, in fact, both a differentiation and a local averaging. This is a sort of smoothing but in fact the number of discontinuities is reduced, that is the reason we applied this mask instead of using a simple derivative operators [1/2 0 -1/2]. The force field has to be the more continuous as possible.

After theses two first steps, we have only to take the absolute value of the gradient and to apply another type the differentiation mask. We will now show one example of the force field.

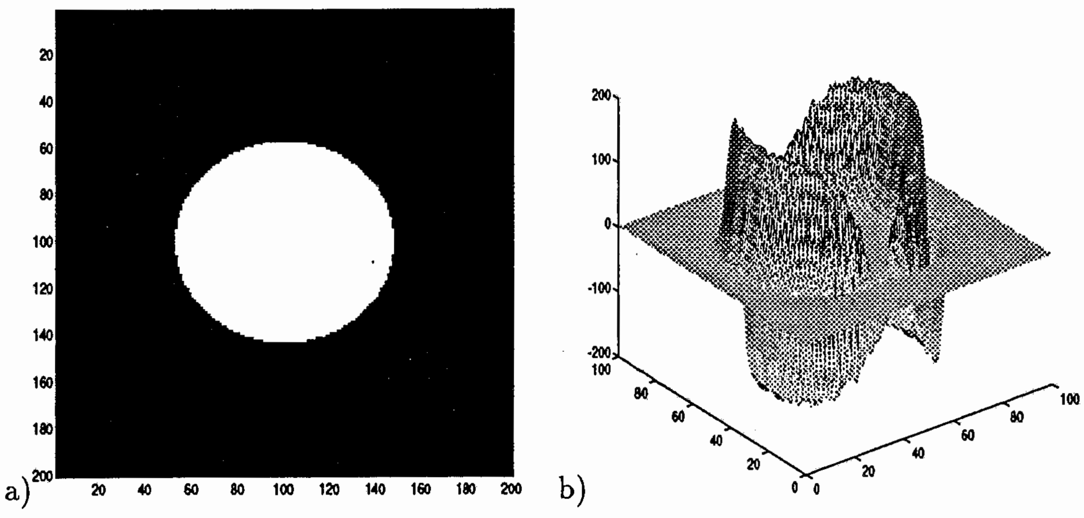


Figure 4.5: *a) The input image; b) the force field*

In the previous chapter, we talk about semidiscretization in space. This becomes important here because the snake moves in a semi-continuous space but the force field is a matrix (two in fact, one for the x-force and one for the y-force). The x-force for example is known only on a discrete grid corresponding to the image, and therefore, there can be a zero-crossing without any zero in the grid (in fact it is the general case). We used a linear interpolation of the external force at non integer positions. Thus with a discrete grid, we have a continuous definition of the force field and equilibrium points correspond now to zeros of the force.

4.3 Results

Figures 4.6, 4.7 and 4.8 show the evolution of the snake in an image. We defined the initial position as shown in the first picture, then we let our model move in the force field. At the beginning of this sequence, the top of the snake is attracted by a disturbing edge at the top of the image. But the snake's extremities are stuck on the contour of the apple, thus an external force attracts the middle part of the snake to the top of the image but, moreover, the internal elastic force pushes it toward the apple's contour. The more we progress in the sequence, the bigger is the part of the snake stuck on the apple. The almost tangent discontinuity of the snake curve, which can be seen in picture "g", increase the value of the internal force. Thus the middle part of the snake is suddenly extracted from the neighborhood of the disturbing edge in the top of the image and the snake can reach the apple's contour which is its equilibrium.

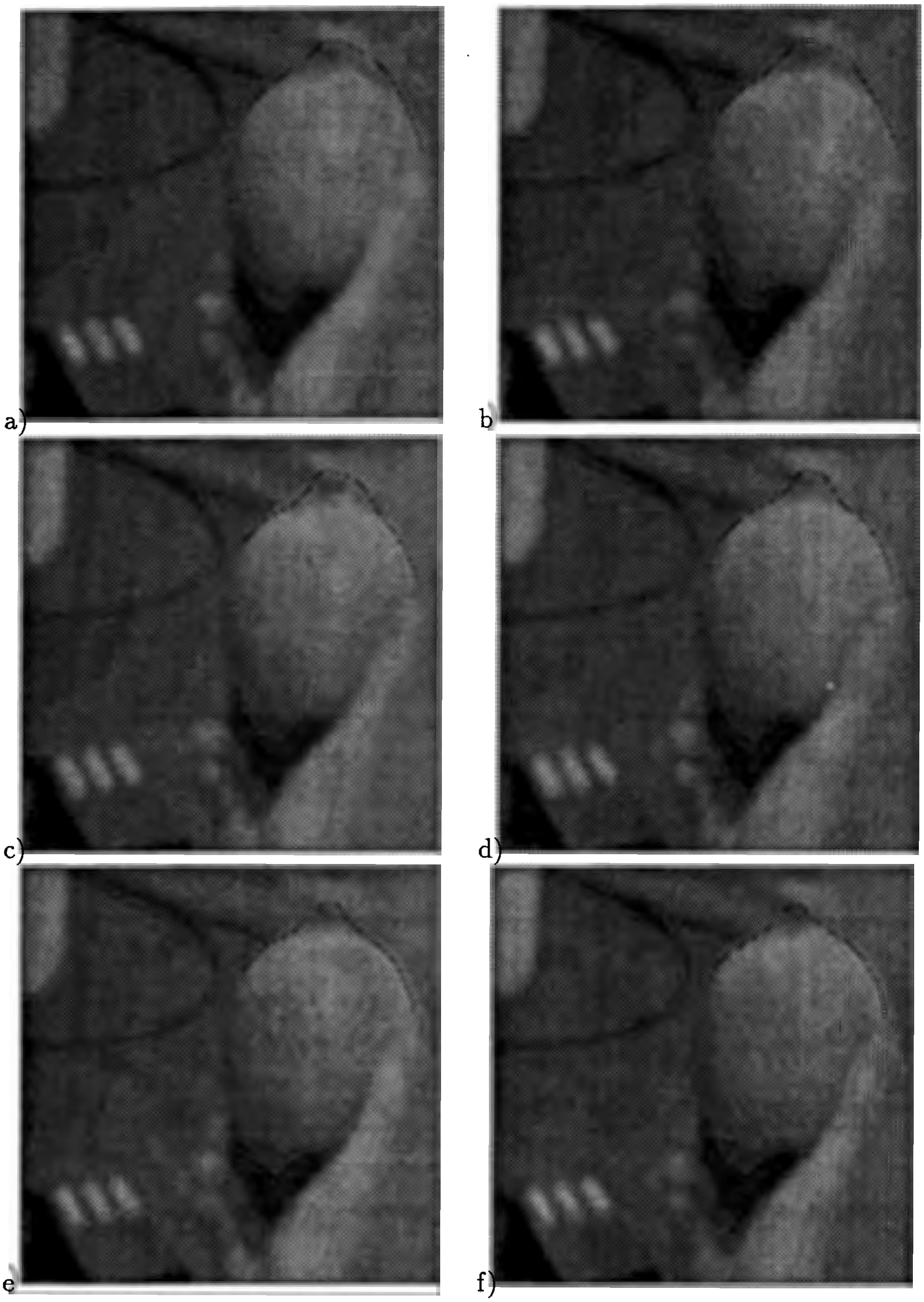


Figure 4.6: *The first sixth steps of the snake's motion.*

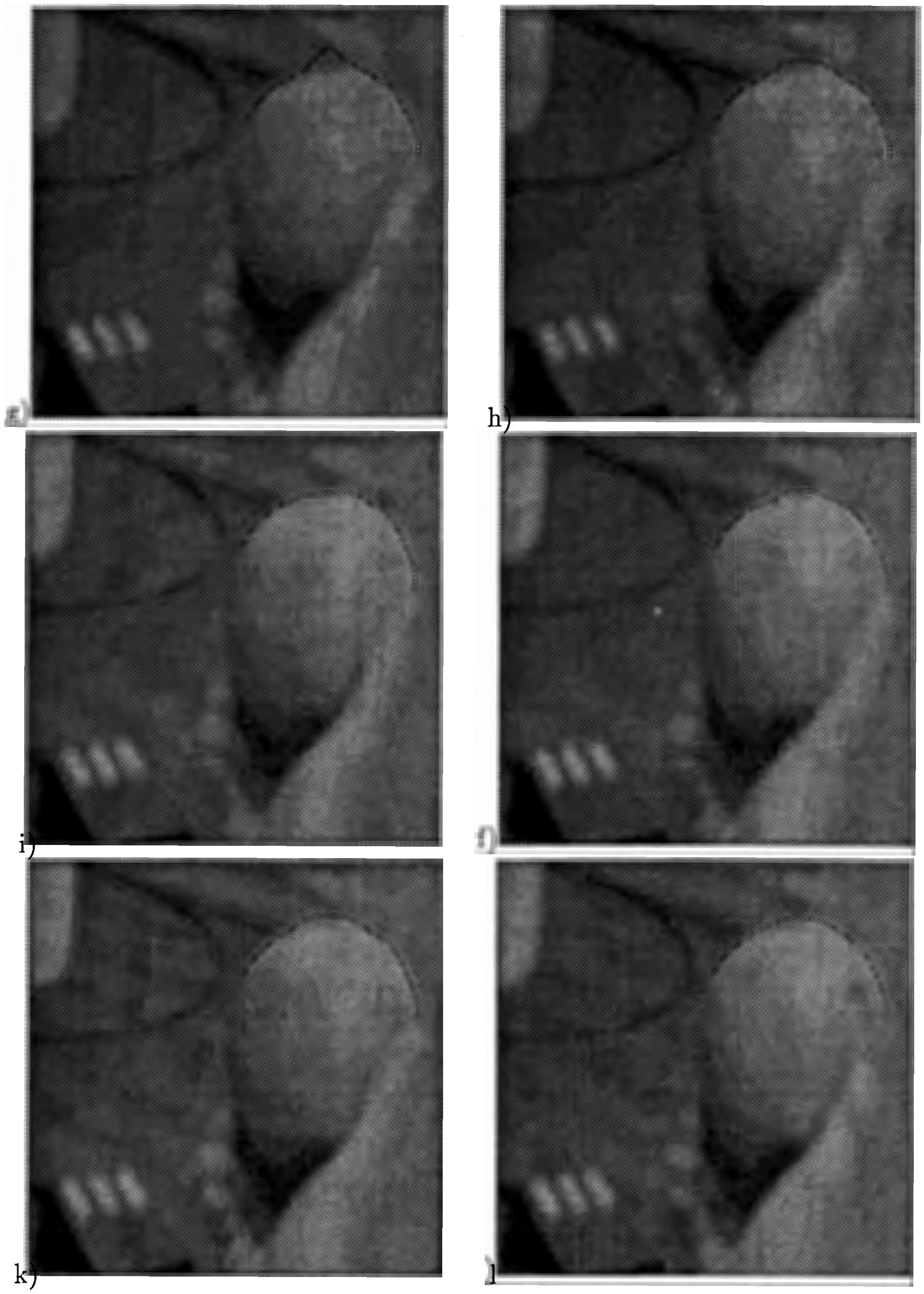
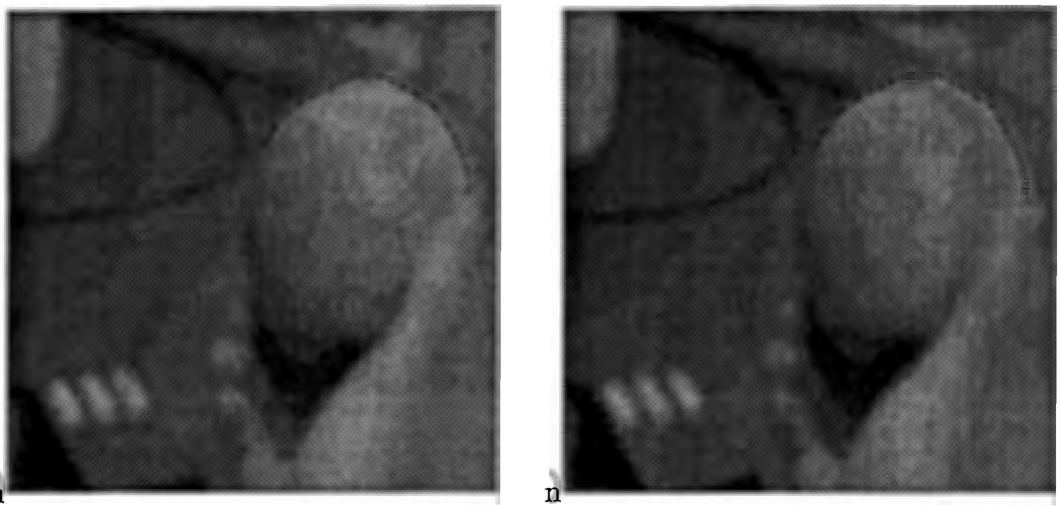


Figure 4.7: *Steps 7 to 12 in the snake's motion.*



m n

Figure 4.8: *The final two steps in the snake's motion.*

Chapter 5

IMPROVEMENTS

5.1 Introduction

In the previous chapter, our subject was the construction of the force field from the image and we showed the results of our snake on a few images. But the snake is a physical model, its motion takes place in a virtual physical space. Our modeling of the snake's motion was basic, but you could improve this model by increasing the complexity of the virtual space. In order to show the possibility of such a model, we will describe in this chapter some possible improvements. The first improvement that we will describe is the rotating body frame. This could be very useful for the following of an object's motion during a sequence of images. The second part will be dedicated to the curvature estimation in a discrete model, and this take a major place in the implementation of the elastic force. We will discuss in the third and last section about the possibility to add a new force (a pressure force) to the snake to make it pass the insignificant edges.

5.2 The Rotating Body Frame

In Chapter 2 we express both a reference component $r(s)$ and an elastic component $e(s)$ in body coordinates relatively to a reference frame ϕ. But we only allowed the body frame to translate and thus a rotation of the whole snake was considered as a deformation, but, in fact, it is not. So to have a more accurate description of the snake's motion we could have allowed the body frame to rotate. In order to do that, we need to know the orientation $\theta(t)$ of ϕ relative to Φ (the reference frame) and the displacement $c(t)$ between the two frames. $c(t)$ is the the position of the origin of the body frame, it is the body's center of mass:

$$c(t) = \int_{\Omega} \mu(s)v(s,t)ds \tag{5.1}$$

where the terms in this equation have been defined in Chapter 2. The velocity of mass elements relative to Φ is

$$\dot{v}(s,t) = \dot{c}(t) + \omega(t)xq(s,t) + \dot{e}(s,t) \tag{5.2}$$

where an overstrike denotes a time partial or total derivative, $\omega(t)$ is the angular velocity of ϕ relative to Φ

Once again, to obtain the equations of motion, you have to apply Lagrangian mechanics to the kinetic energy that governs the model given below as

$$T = \frac{1}{2}\int \mu\dot{v}\dot{v}ds = \frac{1}{2}\int \mu(\dot{c} + \omega xq + \dot{e})(\dot{c} + \omega xq + \dot{e})ds \tag{5.3}$$

There is a velocity dependent energy dissipation (in terms of the Raleigh dissipation functional) of

$$F = \frac{1}{2}\int \gamma\dot{v}\dot{v}ds$$

and, given this, we can express the equations of motion as (ξ the elastic energy does not depend on $\dot{c}$ or ω):

$$\begin{aligned} \delta_c T + \delta_{\dot{c}} F &= \int f ds = f^{\dot{c}} \\ \delta_\theta T + \delta_\omega F &= \int qxf ds = f^{\omega} \\ \delta_e T + \delta_{\dot{e}} F + \delta_e \xi &= f \end{aligned} \tag{5.4}$$

where $f^{\dot{c}}$ is a net translating force applied on the center of mass and f^{ω} is a net torque acting also on the center of mass. The two first equations describe the rigid motion of the snake (or the motion of the body frame ϕ relative to the inertial frame Φ) and the last equation describes the deformation of the body in the body frame from its reference shape.

So you can obtain the equations of motion for the unknown functions $\dot{c}$, ω and e under the action of the force f, and assuming small deformations, you have three coupled differential equations:

-
$$m\ddot{c} + \frac{d}{dt}\int \mu\dot{e}ds + \int \gamma\dot{v}ds = f^{\dot{c}}$$

  where m is the total mass of the body. On the left hand of this equation, the first term is the total inertia of the body as if it was concentrated on its center of mass, while the second term is the motion due to the total displacement from the reference shape. The third term is the damping force.

-
$$\frac{d}{dt}(I\omega) + \frac{d}{dt}\int \mu q \times \dot{e}ds + \int \gamma q \times \dot{v}ds = f^{\omega}$$

  where I is the two-dimensional inertia tensor with $I_{ij} = \int \mu(\delta_{ij}q^2 - q_iq_j)ds$. There is an analogous interpretation (term by term) of this equation in terms of inertial torques. The first two terms are due to the body moment of inertia about c and the total angular momentum due to the motion of mass elements and the third term is the total damping torque.

-
$$\mu\ddot{e} + \mu\ddot{c} + \mu\omega \times (\omega \times q) + 2\mu\omega \times \dot{e} + \mu\dot{\omega} \times q + \gamma\dot{v} + \delta_e\xi = f$$

  This equation is the compilation of several inertial forces applied on individual mass elements as they deform in the body frame ϕ. the first term is the inertial force of a mass element, the second is due to the linear acceleration of the center of mass, the third is the centrifugal force, the fourth one is the Coriolis force, the fifth one is the transverse force. The penultimate term is the term of the damping force and the last term is the elastic force due to the deformation of the body in the body frame ϕ.

This rotating body frame increase the complexity of the model because you have to solve three coupled equations of motion. The equations can be simplified, for example, by removing the Coriolis force; however they remain coupled. These equations now express rigid rotation without any deformation. In the case of following one object in a sequence of images, the rotational component is indispensable. Yet as described earlier in the report, we considered only simple edge finding and did not proceed with implementation of this more involved model.

5.3 A Correction in Curvature Estimation

In Chapter 3, when we talked about semidiscretisation, we used the assumption that the mass points of the snake were separated by a constant distance. In fact, this is not true and the derivatives have to be corrected by a factor of d_i where d_i is the distance between points i and $i-1$. We made another assumption, which is: the parameter s is the arc length. When this assumption is true, then the curvature is given by $|v_{ss}| = \sqrt{x_{ss}^2 + y_{ss}^2}$. However, when the parameter is not arc length, curvature is given by

$$K = \frac{|x_s y_{ss} - x_{ss} y_s|}{(x_s^2 + y_s^2)^{3/2}}. \tag{5.5}$$

The measure of curvature is a very important point because of its great influence on the computation of the elastic force along the snake. We developed it in a very simplistic way in the Chapter 3 but, in fact, it is a complicated problem in discretization[9]. The more accurate the curvature estimation is, the more accurate the elastic force will be.

It is not clear what measure of curvature is the best reflection of the geometric situation depicted by the contour. Nevertheless, the mathematical definition of curvature is $K = \frac{d\theta}{ds}$, where θ is the angle between the positive x-axis and the tangent vector to the curve. This is a coordinate independent measure as the same values will be obtained for $d\theta$ when any line is substituted for the x-axis (the measure is invariant under rotation). In this section we will give two examples of possible curvature estimation in discrete contour.

The first possibility for approximating curvature is to apply the definition of curvature directly. Let the vector $u_i = (x_i - x_{i-1}, y_i - y_{i-1})$. If $\Delta\theta$ is the angle between the two vectors u_i and u_{i+1}, the formula for $\Delta\theta$ is given by

$$\begin{aligned}
\Delta\theta &= \arccos \frac{u_i \cdot u_{i+1}}{|u_i||u_{i+1}|} \\
&= \arccos \frac{(x_i - x_{i-1})(x_{i+1} - x_i) - (y_i - y_{i-1})(y_{i+1} - y_i)}{\sqrt{[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2][(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]}}.
\end{aligned}$$

Moreover the centered difference,

$$\Delta s = (\Delta s_{i+1} - \Delta s_i)/2 = (|u_{i+1}| + |u_i|)/2$$

averages the distance from the point i to its two neighbors and thus gives the best estimate of ds. So we have here a measure of curvature which is intuitively satisfying but it requires a lot of computation. So we could gain some precision but we could lose a great time in computation.

There is another possible measure of curvature which has the advantage of being computationally more efficient. If u_1 and u_2 are the vectors defined above, $|u_1 - u_2|_2$ reflects not only the difference between the two vectors, but also the difference in length.

Normalizing the two vectors before taking the difference removes the length differential, and the measure depends solely on relative direction. The length of $\frac{u_{i+1}}{|u_{i+1}|} - \frac{ui}{|u_i|}$ is given by $2\sin\theta/2$ where θ is the difference in direction of the two vectors as shown in the figure before.

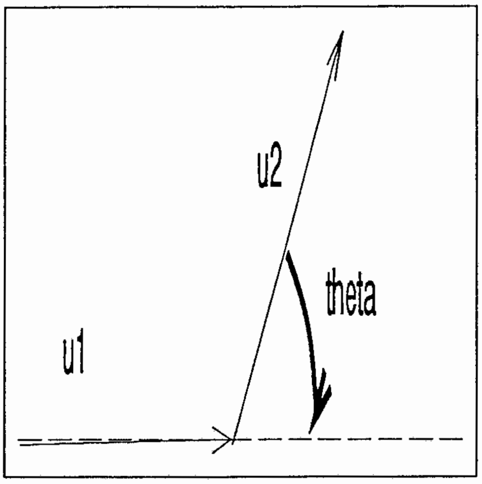


Figure 5.1: *difference in direction of two vectors*

5.4 Snake and Balloon

In our implementation, an initial guess of the contour has to be provided. This has a major consequence for the evolution of the curve: if the snake is not initialized close enough to an edge, it is not attracted by it. So we could add another force which would make the contour have a more dynamic behavior. We can now consider the curve as a "balloon"[10] (in a two-dimensional space) that we inflate. From our initial physical model, we could add a pressure force pushing outside as if we have been introducing air inside the balloon. This additional force could be written as:

$$P = kn(s) \tag{5.6}$$

where $n(s)$ is the normal unitary vector to the curve at point $v(s)$ and k is the amplitude of this pressure force. It is obvious that the sign of k or the orientation of the curve could change the effect of this force. So we could have an effect of deflation instead of inflation. if we chose an inflation force for example, the curve then expands and it is attracted and stopped by edges as before. But since there is is a pressure force, if the edge is too weak (if the gradient is not important enough) the curve can pass through this edge if it is a singularity with regard to the rest of the curve being inflated. So if it tends to have a tangent discontinuity at a point, the smoothing and the pressure force could remove the discontinuity and the curve passes through the edge. This improvement can be very useful if you want to detect the contour of a convex shape but in many cases the objects to detect are not convex. We just talk about the main default but one of the majors advantages of this extra force is that it makes the snake much less sensitive to the initial conditions.

Chapter 6

SUMMARY

The first chapter of this report was an introduction to the relatively new area of research of physical-based models. We saw also that computer graphics and image processing were not two totally separated worlds, and that a lot of overlap exist between these two research areas. One application of physical-based modeling is the "snake", a elastic curve with motion influence by internal and external forces. The main interest of this physical curve is that it has a dynamic behavior that can be useful in edge detection or a motion tracking, for example.

The second chapter was dedicated to the physical study of an elastic curve in a force field. We introduced a body frame to make the equations of the snake's motion easier. We also described the elastic force which gives the snake its reaction against deformation. At the end of this chapter, we showed some examples of different behaviors for a snake.

The third chapter was dedicated to the implementation of our physical elastic curve. We first discretized in space the snake: the snake was no longer an continuous elastic curve but a linear net of mass points; each of these points was related to its two-nearest neighbours. After having described the elastic force on one point, we wrote it for the whole snake. We obtained an equation of motion for the snake where the elastic force was expressed as the product of a matrix of elasticity and the snake's vector of position. After that we integrated this equation through time. All throughout Chapters 2 and 3, we introduced some parameters (dynamic, time and elastics parameters), and we explained their influence on the model at the end of Chapter 3.

The fourth chapter was dedicated to the image processing part of our work. We described there the way we used to obtain a force field from an image. We always kept in mind that this force will apply on the snake so the force had to push the snake toward the edges. So, we studied a sucession of mathematical operations which could transform a input signal into an output force. Then we applied this operators on our two dimensionnal problem. At the end of the chapter 4, we showed the results of our work.

The last chapter was dedicated to some possible improvements. We first studied the case of the rotating body frame which could allow the snake to have rigid rotation without deformation. This improvement should be useful to the motion tracking of an object in a sequence of images. The second improvement was a correction in the curvature estimation to make the elastic force more accurate. We introduced a new external force as the last possible improvement. This force was an pressure force that could make the snake pass through insignificant edges.

Bibliography

[1] Theo Pavlidis. *Algorithms for Graphics and Image Processing.* Comput. Sci. Press, 1982.

[2] Andrew Witkin Michael Kass and Demetri Terzopoulos. Snakes:active contour models. *International Journal of Computer Vision*, 1:321–323, 1987.

[3] Andrew Witkin Demetri Terzopoulos and Michael Kass. Symmetry-seeking models and 3d object reconstruction. *International Journal of Computer Vision*, (1):211–221, 1987.

[4] Demetri Terzopoulos and Andrew Witkin. Physically-based models with rigid and deformable components. In *Graphics' Interface*, pages 146–154, June 1988.

[5] Demetri Terzopoulos and Kurt Fleischer. Deformable models. *The Visual Computer*, (4):306–331, 1988.

[6] Gilbert Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, 1986.

[7] Saeid Terhani Amir A. Amini and Terry E. Weymouth. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1988.

[8] Demetri Terzopoulos Andrew Witkin and Michael Kass. Signal matching through scale space. *International Journal of Computer Vision*, pages 133–144, 1987.

[9] Donna J. Williams and Mubarak Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics, and Image Processing*, 55(1):14–26, January 1992.

[10] laurent D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing*, 53(2):211–218, March 1991.

Appendix A

MATLAB PROGRAMS

MATLAB is a technical computing environment for numeric computation and visualization. It integrates numerical analysis, matrix computation, signal processing and graphics in an easy-to-use environment. This is an interactive system whose basic data element is a matrix that does not require dimensioning. This allows solution of numerical problems in a fraction of the time it would to write a program in a language such as Fortran, Pascal or C.

```
%%%%%%%%%%% FORCE FIELD'S PROGRAM %%%%%%%%%%


forcex=[];
forcey=[];
imax=ov0;
imay=ov0;
[ny,nx]=size(imax);

%******MASKS DEFINITION*******
ax=[0 0 0;1/4 1/2 1/4;0 0 0];
ay=[0 1/4 0;0 1/2 0;0 1/4 0];
bx=[1/4 0 -1/4;1/2 0 -1/2;1/4 0 -1/4];
by=[1/4 1/2 1/4;0 0 0;-1/4 -1/2 -1/4];


%*********SMOOTHING************
for j=1:20
        res=imax';
        imax=[res(1,:);res;res(nx,:)]';
```

```
        imax=conv2(imax,ax);
        imax=imax(2:ny+1,3:nx+2);
        imay=[imay(1,:);imay;imay(ny,:)];
        imay=conv2(imay,ay);
        imay=imay(3:ny+2,2:nx+1);
end


%%%%%%%%%%%%%% FOR THE X-FORCE %%%%%%%%%%%%%


%*********FIRST DERIVATIVE************
res1=imax';
imax=[res1(1,:);res1;res1(nx,:)]';
forcex=conv2(imax,bx);


%**********ABSOLUTE VALUE*************
forcex=abs(forcex);


%*****SECOND FIRST DERIVATIVE**********
forcex=forcex(2:ny+1,3:nx+2);
res2=forcex';
forcex=[res2(1,:);res2;res2(nx,:)]';
forcex=conv2(forcex,bx);
forcex=10*forcex(2:ny+1,3:nx+2);


%%%%%%%%%%%%%% FOR THE Y-FORCE %%%%%%%%%%%%%


%*********FIRST DERIVATIVE*************
imay=[imay(1,:);imay;imay(ny,:)];
forcey=conv2(imay,by);


%***********ABSOLUTE VALUE************
forcey=abs(forcey);


%*******SECOND FIRST DERIVATIVE********
forcey=forcey(3:ny+2,2:nx+1);
forcey=[forcey(1,:);forcey;forcey(ny,:)];
forcey=conv2(forcey,by);
```

```
forcey=10*forcey(3:ny+2,2:nx+1);


%%%%%%%%%%%%%% SNAKE'S PROGRAM %%%%%%%%%%%%%%%


%*****INITIALISATION********


%****************parameters

%               number of mass points of the snake

n=input('number of mass points ');


%               dynamic coefficients
gam=input('damping parameter');
mu=input('mass density');

%               elastic coefficients
w0=input('w0=');
w1=input('w1=');
w2=input('w2=');

%               time parameters

dt=0.01;
dt2=dt*dt;


%************** initial  snake's position

clf
ov0=0.15*apple_01_6;
colormap(gray)
image(ov0)
hold on

% Initially, the list of points is empty
x=[];
y=[];
n1=0;

%  loop picking up the points
```

```
disp('left mouse button picks points.')
disp('right mouse button picks last point.')

but=1;
while but ==1
        [xi,yi,but]=ginput(1);
        plot(xi,yi,'go')
        n1=n1+1;
        x(n1,1)=xi;
        y(n1,1)=yi;
end

%   interpolate with two splines and finer spacing
t1=1:n1;
ts=1:(n1-1)/(n-1):n1;
xt=spline(t1,x,ts);
yt=spline(t1,y,ts);
h=mean(sqrt(diff(xt).^2+diff(yt).^2));
%   plot the curve
plot(xt,yt,'c-');
hold off
cx0=mean(xt);
cy0=mean(yt);
xt0=xt-cx0;
yt0=yt-cy0;
xt_1=xt;
yt_1=yt;
fx=zeros(1,n);


%**************elastic matrix k initialisation

k=zeros(n,n);
for i = 1:n-2
        k(i,i)= w0+2*w1/(h^2)+6*w2/(h^4);
        k(i,i+1)= -w1/(h^2)-4*w2/(h^4);
        k(i+1,i)= -w1/(h^2)-4*w2/(h^4);
        k(i,i+2)=w2/(h^4);
        k(i+2,i)=w2/(h^4);
end
k(n,n)=w0+2*w1/(h^2)+6*w2/(h^4);
k(n-1,n-1)=w0+2*w1/(h^2)+6*w2/(h^4);
k(n-1,n)=-w1/(h^2)-4*w2/(h^4);
k(n,n-1)=-w1/(h^2)-4*w2/(h^4);
```

```
%************** calculation of the force field

        force_field;


%********** some variables to increase the velocity of the snake

alpha1=2*mu/dt2-gam/dt;
alpha2=gam/dt-mu/dt2;
id=ones(1,n);
dis=70;
g=1:1:n;
gs=1:0.1:n;
t=0;
tprim=0;
axis([0 nx 0 ny]);


%%%%%%%%%% MAIN PART : THE SNAKE'S MOTION %%%%%%%%%


while t<500


%******************displaying of the snake and the image

        if dis==70
        axis ([0 nx 0 ny]);
        axis('equal');
        image(ov0)
        hold on
        plot(xt,yt,'w');
        pause(1);
        plot(xt,yt,'i');
        hold off
        dis=0;
        end
        dis=dis+1;


%******calculation of the deformation from the initial shape
%                   at the time t

        cxt=mean(xt);
```

```
        cyt=mean(yt);
        dex=xt-cxt - xt0;
        dey=yt-cyt - yt0;


%*******comptutation of the external force which is applied
%                      to the whole snake
%                      (linear interpolation)

        entx=round(xt);
        enty=round(yt);
        for i=1:n
        if enty(i)==0
                enty(i)=1;
        end
        fx(i)=(xt(i)-entx(i))*
               (forcex (enty(i), entx(i)+1)  -
                forcex (enty(i), entx(i)  )) +
               forcex (enty(i), entx(i));

        fy(i)=(yt(i)-enty(i))*
               (forcey (enty(i)+1, entx(i))  -
                forcey (enty(i),   entx(i))) +
               forcey (enty(i), entx(i));
        end


%******************* computation of the snake's next position


        xtplus=(fx+alpha1*xt+alpha2*xt_1-dex*k)*dt2/mu;
        ytplus=(fy+alpha1*yt+alpha2*yt_1-dey*k)*dt2/mu;



%****************** adjustement

        xt_1 = xt;
        yt_1= yt;
        xt=xtplus;
        yt=ytplus;


%*******************boundaries conditions
```

```
        for ii=1:n
        if yt(ii)<=1
                yt(ii)=1;
        end
        if yt(ii)>=ny
                yt(ii)=ny;
        end
        end

%*****
t=t+dt;
tprim=tprim+1;

end
%%%%%%%%%%% END OF THE MAIN PART %%%%%%%%%%%%%
```