Parallel Implementation of Lagrangian Dynamics for real-time snakes

R.M.Curwen, A.Blake and R.Cipolla

Robotics Research Group
Department of Engineering Science
Oxford University OX1 3PJ
United Kingdom

Abstract

Snakes as originally proposed by Kass, Witkin and Terzopoulous are active contour models which minimise an expression of energy to locate image features. The original formulation involves hundreds of state variables and does not submit to a real-time implementation. We explore the use of a B-spline model of the feature to reduce the state space of the problem. Using a coarse to fine feature search and Lagrangian Dynamics we demonstrate a real-time, parallel implementation of B-spline snakes on a network of transputers.

1 Introduction

Energy-minimising Active Contour models (snakes) were proposed by Kass, Witkin and Terzopoulos [8] as a top-down mechanism for locating features of interest in images and tracking their image motion as long as the feature does not move too fast. The snake is a computational construct, a dynamic curve able to track moving, deforming image features. Since many snakes can be active at once, each tracking its feature contour as a background process, they constitute a versatile mechanism for direction and focus of attention, a second generation of Inoue's window system [7].

1.1 The snake model

The behaviour of a snake is controlled by internal and external forces. The internal forces enforce smoothness and the external forces guide the active contour towards the image feature. In their implementation for image curve localisation and tracking, the external force is computed from the image intensity data $I(\mathbf{x}(s))$, where the position of the snake is represented by $\mathbf{x}(s)$, by differentiating an external energy:

$$E_{external} = -\left|\nabla G(\sigma) * I(\mathbf{x}(s))\right|^2 \tag{1}$$

The internal energy at a point on the snake $\mathbf{x}(s)$:

$$E_{internal} = \frac{\alpha|\mathbf{x}_s|^2 + \beta|\mathbf{x}_{ss}|^2}{2} \tag{2}$$

BMVC 1991 doi:10.5244/C.5.5

is composed of first and second order terms forcing the active contour to act like a membrane (avoiding gaps) or a thin plate (avoiding high curvatures) (controlled by the relative values of α and β). This serves to maintain smoothness of the curve under changing external influences. The tracking behaviour of the snake is then achieved by numerical, iterative solution of the elastic problem using techniques from variational calculus.

Amini et al (1988)[1] have discussed the problems with this approach. These include instability and a tendency for points to bunch up on strong portions of an edge. They have presented an implementation based on Dynamic Programming instead of variational methods which allows the inclusion of hard constraints which can not be violated as well as the original smoothness constraints which do not have to be satisfied exactly.

Snakes can be set up to be attracted to specific shapes. Yuille et al [10] use a variant of snakes - deformable templates - to detect, describe and track features in medical images. The feature of interest is described by a parameterised template with a limited number of parameters, including constraints on the 2D shape of the feature to be detected.

1.2 The B-spline snake

A more economical realisation can be obtained by using far fewer state variables [11]. Blake and Cipolla (1990) [4] proposed cubic B-splines [5], which are deformable curves represented by four or more state variables (control points). The curves may be open or closed as required. The flexibility of the curve increases as more control points are added; each additional control point allows either one more inflection in the curve or, when multiple knots are used [2], reduced continuity at one point.

The B-spline is a curve in the image plane

$$\mathbf{x}(s) = \sum_i f_i(s)\mathbf{q}_i \tag{3}$$

where f_i are the spline basis functions with coefficients $\mathbf{q}_i$ - the vertices or control points of the curve's "characteristic polygon". An energy function of the B-spline snake can then be defined in such a way that it is a minimum when the snake is positioned on a contour of high contrast.

B-splines have the desirable properties of local control - modifying the position of a data-point causes only a small part of the curve to change; continuity control - B-splines are defined with continuity properties; and that the number of variables to be estimated is reduced to the number of control-points. The are no internal forces since the B-spline representation maintains smoothness via hard constraints implicit in the representation.

Blake et al (1991)[3] describe a real-time system in which B-spline snakes could just "hang-around" in the image until they are swept by the motion of the camera over a feature for which they have affinity. The range of the feature search (scale) is controlled by inspecting image gradients around the snake using finite differences. Gaussian blurring is unnecessary since image noise is not, as might be thought, problematic in the unblurred image. CCD cameras have relatively low noise and gradient is sampled at several (currently 20) places along the spline, and those samples combined to compute motions for the spline control points. The combination of those samples itself has an adequate averaging, noise-defeating effect.

2 Snakes with Dynamics

A principal claim of this paper is that tracking performance is greatly enhanced if snake dynamics are carefully modelled. For example a snake can be given *mass*, distributed along its length, and immersed in a simulated viscous fluid. The effect of the mass is that the snake tracker now has a "memory" for its velocity and so prefers to move in a continuous fashion. The subjective effect is dramatic. The snake seems much "stickier" and far less prone to fall off fast-moving objects. (The viscosity is necessary to avoid oscillatory behaviour.) The enhanced performance following the incorporation of mass illustrates the importance of modelling the motion of the 3D object. The inclusion of mass amounts to an assumption that the object is in roughly uniform motion across the image.

2.1 Distributed mass and damping

Consider the snake described by a quadratic B-spline with L spans (Figure 1), in which the position in the ith span, $\mathbf{x}_i$, is a function of the control points $\mathbf{Q}_i = (\mathbf{q}_i, \mathbf{q}_{i+1}, \mathbf{q}_{i+2})$, the shape matrix $\mathbf{M}_i$ and the snake parameter vector $\mathbf{s} = (1, s, s^2)^T$, where $0 \leq s \leq 1$ over a single span:

$$\mathbf{x}_i(s) = \mathbf{s}^T \mathbf{M}_i \mathbf{Q}_i \tag{4}$$

The equations of motion of a snake are derived using Lagrangian dynamics, defining energy functions for each of the desired properties of the snake in terms of the global control point vector $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_N)^T$. These properties are inertia, attraction towards the feature and velocity damping. The corresponding energy functions are derived by analysis of small elements ds of the snake spline, with a linear density ρ, compliance k and damping λ per unit s. These parameters are selected to give critical damping so that the snake captures contours as rapidly as possible without any tendency to oscillate.

2.2 Implementation

The resulting second order differential equation is integrated using an Euler scheme for speed. The snakes are implemented on a network of transputers. Each span is

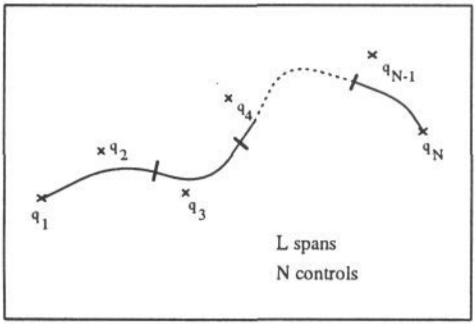

Figure 1: *A B-spline snake*

modelled as a sequential process [6], executing a three step algorithm.

1. The span position and normal samples are determined, and used to search for the feature using coarse to fine search.
2. The span communicates the effects of the feature to the rest of the snake.
3. The span uses the feature information from the rest of the snake to perform the Euler step and find the new values of the snake control points.

3 Results

Figure 2 shows the performance of a six span snake following straight linear features moving with constant velocity across the image. The root mean square distance between snake and feature is plotted against frame number. The snake underdamped to illustrate its dynamics, hence the transient oscillation of the curves as it accelerates to catch up with the feature. Notice the steady state error in following a moving feature. If an accurate localisation is required, for instance for motion measurements, this can be done in a separate process. In the steady state the root mean square error is proportional to the feature velocity, as would be expected from a one dimensional control theory analysis.

Using the full processing capacity of 4 iterations per frame, the snake can successfully track features whose velocity is such that the lag caused by viscous drag does not exceed the radius of the tracking window. With a tracking window radius of approximately 35 mrad (in a field of view of 0.3 rad) maximum tracking velocity is about 3.1 rad/sec, for our system.

For a feature accelerating uniformly from rest, the snake will eventually cease to track when the displacement between frames exceeds the tracking window radius. The snake is therefore able to track a feature accelerating from rest at a maximum of 4.3 rad/sec/sec.

Figure 3 shows the snake following a moving occluding contour in the image plane. Note how it deforms over time to accommodate the shape of moving features.

References

[1] A.A. Amini, S. Tehrani, and T.E. Weymouth. Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. In *Second International Conference on Computer Vision (Tampa,, FL, December 5–8, 1988)*, pages 95–99, Washington, DC,, 1988. Computer Society Press.

[2] R.H. Bartels, J.C. Beatty, and B.A. Barsky. *An Introduction to the Use of Splines in Computer Graphics and Geometric Modeling.* Morgan Kaufmann, Los Altos, CA, 1987.

[3] A. Blake, J.M. Brady, R. Cipolla, Z. Xie, and A. Zisserman. Visual navigation around curved obstacles. In *IEEECRA*, 1991.

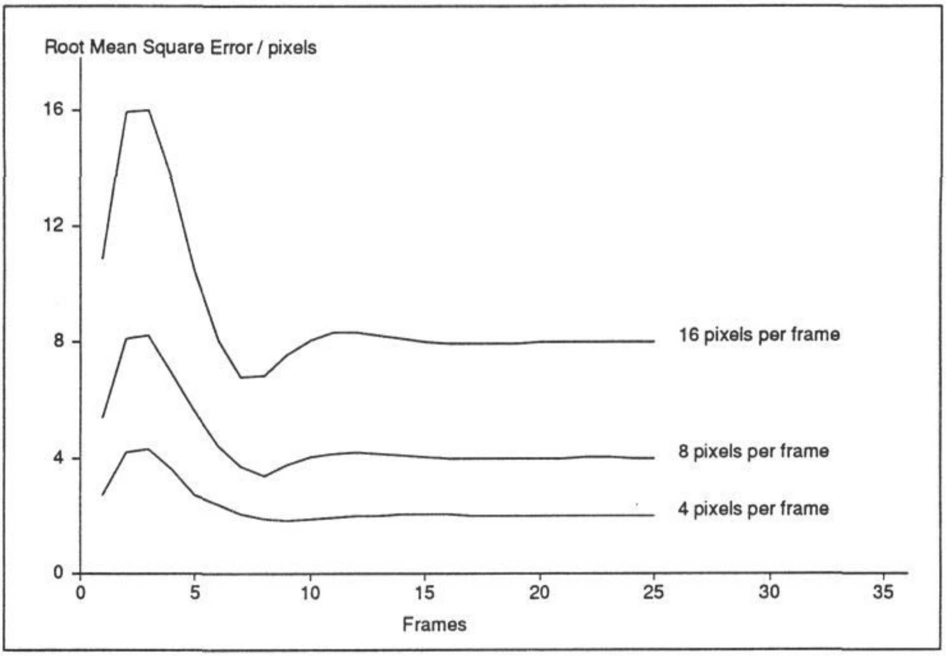

Figure 2: *Root mean square error in position for an initially stationary snake tracking a feature moving with constant velocity across the image. The snake is slightly under damped to illustrate dynamics.*

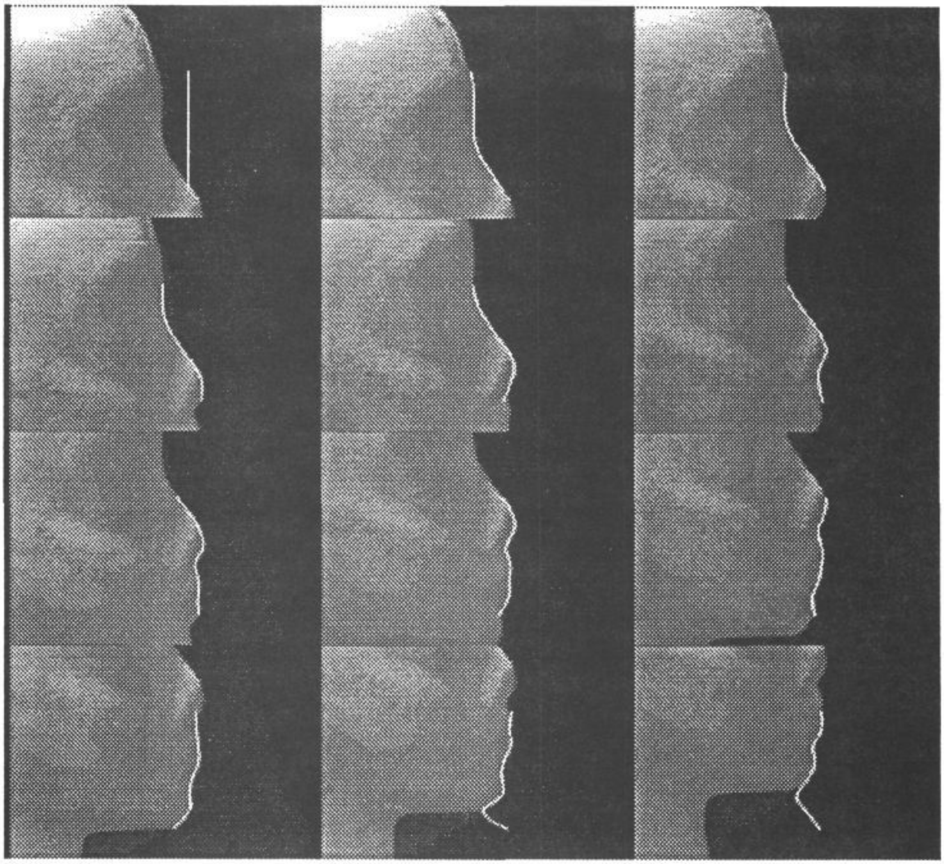

Figure 3: *A six span snake tracking a moving feature. The image is moving upwards with an average velocity of forty-five pixels per frame.*

[4] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. Technical Report Technical Report OUEL 1828/90, University of Oxford, April 1990.

[5] I.D. Faux and M.J. Pratt. *Computational Geometry for Design and Manufacture.* Ellis-Horwood, 1979.

[6] C. Hoare. *Communicating Sequential Processes.* Prentice-Hall, 1985.

[7] H. Inoue and H. Mizoguchi. A flexible multi window vision system for robots. In *Proc. 2nd Int. Symp. on Robotics Research*, pages 95–102, 1985.

[8] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision, (London, England, June 8–11, 1987)*, pages 259–268, Washington, DC., 1987. IEEE Computer Society Press.

[9] L.D. Landau and E.M. Lifshitz. *Mechanics and Electrodynamics*, volume 1 of *A Shorter Course of Theoretical Physics.* Pergamon Press, 1972.

[10] P. Lipson, A. Yuille, D.O. Keefe, J. Cavanaugh, J. Taaffe, and D. Rosenthal. Deformable templates for feature extraction from medical images. In O. Faugeras, editor, *Proc. 1st European Conference on Computer Vision*, pages 413–417. Springer-Verlag, 1990.

[11] G. Scott. The alternative snake. In *Proc. 3rd Alvey Vision Conference*, 1987.