Detection of curved edges at subpixel accuracy using deformable models

M. Kisworo
S. Venkatesh
G.A.W. West

Indexing terms: Curve reconstruction, Deformable models, Subpixel levels

Abstract: One approach to the detection of curves at subpixel accuracy involves the reconstruction of such features from subpixel edge data points. A new technique is presented for reconstructing and segmenting curves with subpixel accuracy using deformable models. A curve is represented as a set of interconnected Hermite splines forming a snake generated from the subpixel edge information that minimises the global energy functional integral over the set. While previous work on the minimisation was mostly based on the Euler–Lagrange transformation, the authors use the finite element method to solve the energy minimisation equation. The advantages of this approach over the Euler–Lagrange transformation approach are that the method is straightforward, leads to positive *m*-diagonal symmetric matrices, and has the ability to cope with irregular geometries such as junctions and corners. The energy functional integral solved using this method can also be used to segment the features by searching for the location of the maxima of the first derivative of the energy over the elementary curve set.

1 Introduction

While there has been substantial work performed on the detection of one-dimensional features such as edges at subpixel accuracy, little has been done on the subpixel detection of two-dimensional features such as curves and conic sections. The main problem of extracting such features from digital images arises because images are noisy, and so the detected edge points are incomplete and, for a given segment, the points are unequally spaced. Furthermore, since the variance of edge data points is generally high, a simple curve fit through all the points will not reconstruct the correct shape even at pixel accuracy. In this paper we outline a new two-dimensional curve extraction and segmentation algorithm based on deformable models. Our technique assumes that the edge information is reliable and accurate but not equally spaced nor complete, which means that the least-squares method is inappropriate. We use a reliable subpixel edge detector [1, 2] to obtain reliable and accurate edge information.

The use of deformable models for curve detection has drawn attention recently [3, 4]. The approach by Zucker *et al.* [4] represents a curve as a cover set of elementary dynamic deformable curves. This method provides better results and is more straightforward than the previous work based on statistical curve fitting. The deformable model works by dividing the domain curve into a set of elementary curves called subdomains and then minimises the energy integral over the set using the Euler–Lagrange integral transformation. This discrete representation allows the use of an alternative mathematical approach based on finite elements [5].

Our technique follows that of Karaolani *et al.* [5] in the use of the finite element model to solve the energy minimisation problem. However, our method differs in that we model the elementary curves as simple one-dimensional cubic Hermite splines. This simplification allows the application of one-dimensional finite element techniques which can be solved easily using standard finite element methods and allows us to deal with irregular features such as sharp corners by removing the problem of corner smoothing in two-dimensional finite elements. We minimise the energy over all subdomains instead of each subdomain, which thus makes simultaneous extraction and segmentation of the global curve possible.

2 Principles of deformable models

2.1 Lagrangian energy minimisation

Let $c(s)$ represents a normalised deformable curve, where $s \in \Omega = [-1, 1]$ is its spatial parameter. The potential energy functional L (see Fig. 1) of curve c is defined as the *functional*

$$L(c) = \frac{1}{2}\int_{\Omega}\left(\alpha(s)\left|\frac{dc}{ds}\right|^2 + \beta(s)\left|\frac{d^2s}{ds^2}\right|^2 + I(c) + S(c)\right) ds \quad (1)$$

where $\alpha(s)|dc/ds|$ is the tension of the curve that represents the axial loads, $\beta(s)|d^2c/ds^2|$ represents lateral forces, $I(c)$ is the energy from the tangent field that controls the curvature and $S(c)$ is the gravitational force between neighbouring curves that glues adjacent components.

The minimal energy principle requires that the curve c must minimise the functional to have a stable system. The value of the curve that minimises the functional (eqn. 1) is called the *extremal* of the functional. Thus to

Paper 2175K (E4), first received 22nd November 1994 and in revised form 26th June 1995

M. Kisworo is with STMIK Bina Nusantara, JL. H. Syahdan, Kemanggisan, Jakarta Barat, Indonesia

S. Venkatesh and G.A.W. West are with the School of Computing, Curtin University of Technology, GPO Box U1987, Perth 6001, Western Australia

compute the curve equation one needs to minimise the functional. There are two approaches to solving the minimisation problem. The first method is the *Euler–Lagrange method* that transforms the energy integral into

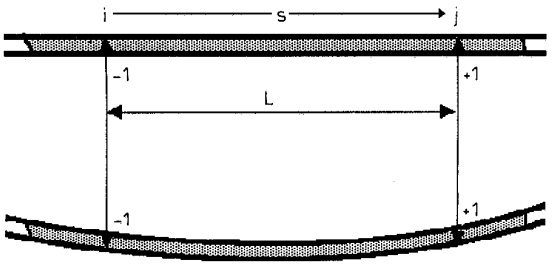

Fig. 1 *Deformed element c(s)*
L is its length and *i* and *j* are its nodal points

a set of differential equations. From the calculus of variations it is known that any minimum of the Lagrangian equation must satisfy the Euler conditions [6]. Zucker *et al.* [4] have developed an algorithm that solves the problem of energy minimisation by applying the Euler–Lagrange method to the energy functional integral. The second method is based on the *finite element method* as proposed by Karaolani *et al.* [5]. This method divides the global energy domain into several small subdomains and minimises the energy integral over each subdomain independently. The global curve is approximated by element equations of each subdomain combined at their joint boundaries. The external energies influencing the model that are independent of the free variables are represented by the loading conditions.

The finite element method has advantages over the Euler–Lagrange transformation method because the method is straightforward, leads to matrices that are positive–definite *m*-diagonal symmetric for which good numerical algorithms have been developed, and has the ability to represent irregular geometries as a set of interconnected elements. Since the method does not transform the potential energy functional integral into a set of differential equations, the problem of boundary conditions [6] will not arise. The use of the finite element method also makes the dealing with discontinuous geometric features such as corners or junctions possible. Contrast this with the Euler transformation method that always generates smooth models with simple geometric properties. In the proposed approach, the erratic behaviour of the elements at interelement boundaries that present difficulties [5] is handled by introducing the gravitational force *S* into eqn. 1, which controls the interaction between neighbouring elementary curves.

The energy is minimised over all subdomains instead of within each subdomain, which makes the simultaneous extraction and segmentation of the global curve possible and allows irregular features such as sharp corners and junctions to be dealt with. The elementary curves are modelled as one-dimensional cubic Hermite splines by assuming that the length of each elementary curve is very small compared to the length of the global curve. This simplification allows the application of one-dimensional finite element techniques that are easily solved using standard finite element methods.

2.2 Finite element solution for the energy functional

The global curve c that fits M edge points is modelled as $M-1$ interconnected elementary curves c_k, $k = 1, \ldots, M-1$. The elements are assumed to be connected at their end points called the *nodal points* (Fig. 1). The length of each element is assumed to be small such that the shape of the displacement of each element can be approximated by a simple one-dimensional function. The function that models the displacement over each finite element is termed the *displacement function.* A normalised *shape function* that has unit value at one nodal point and zero value at all other nodal points is used as the basis for the displacement function. The state of an element is described as a matrix called the *nodal parameter matrix.* The elements of this matrix are termed the *nodal parameters.* The value of c_k and its derivative dc_k/ds, i.e. position and orientation of a pair of edge points, are chosen as the nodal parameters. The nodal parameter matrix and the shape function matrix together define the *element equation.*

To compute the global curve, the equations of the elements are assembled into a system equation. The initial nodal point values are derived from the edge detector. By solving the system equation using these nodal point values, the curvature of the elements [7] are obtained. Integrating the curvature and applying boundary conditions yields new nodal point values that are used in subsequent iteration. This process is performed iteratively until the global energy is minimised.

For example, consider a deformable curve $c(s)$, $s \in \Omega = [-1, 1]$, which is divided into a finite number of elements c_k, $k = 1, \ldots, M-1$, such that each element has two nodal points i and j. Assuming that the system is isotropic, the energy functional L for element c_k can be expressed as

$$L(c_k) = \frac{1}{2}\int_{\Omega}\left(\alpha_k(s)\left|\frac{dc_k}{ds}\right|^2 + \beta_k(s)\left|\frac{d^2c_k}{ds^2}\right|^2 + I(c_k) + S(c_k)\right) ds \quad (2)$$

Given a shape function matrix N, the element equation for element c_k can be written as

$$c_k(s) = Na_k \quad (3)$$

where a_k is the nodal parameters matrix for the elementary curve c_k. The value of c_k and its slope dc_k/ds at each of the nodal points (position and orientation of a pair of edge points) are chosen as the nodal parameters, so the elementary curves have four degrees of freedom

$$a_k = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{Bmatrix} = \begin{Bmatrix} c_{k,i} \\ \dfrac{dc_{k,i}}{ds} \\ c_{k,j} \\ \dfrac{dc_{k,j}}{ds} \end{Bmatrix} \quad (4)$$

Since the elementary curve has four degrees of freedom, a shape function that is cubic in s is required to represent the displacements. If the global curve is assumed to be continuous and the elementary curves have first-order continuity with respect to variable s, the cubic Hermites may be used as the shape function [8]. For a one-dimensional element with two nodal points at $i = -1$ and $j = 1$, the cubic Hermitian shape function N is given by

$$N = \{N_1 \quad N_2 \quad N_3 \quad N_4\} = 0.25\{2 - 3s + s^3 \quad 1 - s - s^2 - s^3 \quad 2 + 3s - s^3 \quad -1 - s + s^2 + s^3\} \quad (5)$$

Since the length of each of the elementary curves is very small compared to the length of the global curve, the elementary curves can be modelled as one-dimensional curves. From the edge data the values of c_k and dc_k/ds at the nodal points $s = i, j$ are known. By substituting these values into eqn. 3 the element equation for the elementary curve c_k can then be written as

$$c_k(s) = 0.25\{2 - 3s + s^3 \quad 1 - s - s^2 - s^3 \quad 2 + 3s - s^3 \quad -1 - s + s^2 + s^3\} \begin{Bmatrix} c_{k,i} \\ \frac{dc_{k,i}}{ds} \\ c_{k,j} \\ \frac{dc_{k,j}}{ds} \end{Bmatrix}$$

$$= 0.25\Big((2 - 3s + s^3)c_k(s)\Big|_{s=i} + (1 - s - s^2 - s^3)\frac{dc_k}{ds}\Big|_{s=i} + (2 + 3s - s^3)c_k(s)\Big|_{s=j} + (-1 - s + s^2 + s^3)\frac{dc_k}{ds}\Big|_{s=j}\Big) \tag{6}$$

The curvature κ of the curve c_k is given by:

$$\kappa_k(s) = \frac{d^2c_k}{ds^2}(s)$$

$$= 0.25\left\{\frac{d^2N}{ds^2}\right\}\begin{Bmatrix} c_{k,i} \\ \frac{dc_{k,i}}{ds} \\ c_{k,j} \\ \frac{dc_{k,j}}{ds} \end{Bmatrix}$$

$$= 0.25\Big(6sc_k(s)\Big|_{s=i} - (2 + 6s)\frac{dc_k}{ds}\Big|_{s=i} - 6sc_k(s)\Big|_{s=j} + (2 + 6s)\frac{dc_k}{ds}\Big|_{s=j}\Big)$$

$$= 0.25Ba_k \tag{7}$$

where B is called the curvature matrix.

The global curve c is the cover of all elementary curves c_k and each elementary curve contributes $L(c_k)$ to the global energy functional. Since the system is assumed to be isotropic, the energy functional of the global curve is simply a sum of the individual energy functionals derived for all elementary curves, i.e.

$$L(c) = \sum_{k=1}^{M-1} L(c_k) = \frac{1}{2}\sum_{k=1}^{M-1}\int_\Omega \left(\alpha_k(s)\left|\frac{dc_k}{ds}\right|^2 + \beta_k(s)\left|\frac{d^2c_k}{ds^2}\right|^2 + I(c_k) + S(c_k)\right) ds \tag{8}$$

Because the length of each elementary curve is very small compared to the length of the global curve, the elementary curves can be assumed to be homogeneous such that within each element α and β are constants and independent of s, and so eqn. 8 can be simplified to

$$L(c) = \frac{1}{2}\sum_{k=1}^{M-1}\int_\Omega \left(\alpha_k\left|\frac{dc_k}{ds}\right|^2 + \beta_k\left|\frac{d^2c_k}{ds^2}\right|^2 + I(c_k) + S(c_k)\right) ds \tag{9}$$

Substituting eqn. 3 into eqn. 9 yields

$$L(c) = \frac{1}{2}\sum_{k=1}^{M-1}\int_\Omega \left(\alpha_k\left|\frac{dN}{ds}a_k\right|^2 + \beta_k\left|\frac{d^2N}{ds^2}a_k\right|^2 + I(c_k) + S(c_k)\right) ds \tag{10}$$

This integral equation forms the global energy equation that is minimised to reconstruct the global curve.

2.3 Inferring the global curve

The global curve is computed by minimising eqn. 10. The minimisation means picking the stationary values of the first variations of L, so it is required that

$$\delta L(c) = \sum_{k=1}^{M-1}\sum_{i=1}^{4}\frac{\partial L(c_k)}{\partial a_{ik}}\delta a_{ik} = 0 \tag{11}$$

where a_i is the nodal parameter. Since the nodal parameters are independent of each other, δa_i are also independent of each other and may have non-zero values. Since L is positive–definite, then eqn. 11 may hold only if the first variation of L is zero, i.e.

$$\frac{\partial L(c_k)}{\partial a_{ik}} = 0, \quad i = 1, 2, 3, 4 \tag{12}$$

Eqn. 12 comprises a set of $M - 1$ equations each of which characterises the behaviour of an elementary curve with respect to its nodal parameters. Carrying out the minimisation of $L(c_k)$ with respect to its nodal parameters gives the equations for an element as

$$\frac{\partial L(c_k)}{\partial a_1} = \int_\Omega \left(\alpha_k\frac{dN_1}{ds}\left|\frac{dN}{ds}a_k\right| + \beta_k\frac{d^2N_1}{ds^2}\left|\frac{d^2N}{ds^2}a_k\right| + \frac{1}{2}\frac{\partial}{\partial a_1}I(c_k) + \frac{1}{2}\frac{\partial}{\partial a_1}S(c_k)\right) ds = 0$$

$$\frac{\partial L(c_k)}{\partial a_2} = \int_\Omega \left(\alpha_k\frac{dN_2}{ds}\left|\frac{dN}{ds}a_k\right| + \beta_k\frac{d^2N_2}{ds^2}\left|\frac{d^2N}{ds^2}a_k\right| + \frac{1}{2}\frac{\partial}{\partial a_2}I(c_k) + \frac{1}{2}\frac{\partial}{\partial a_2}S(c_k)\right) ds = 0$$

$$\frac{\partial L(c_k)}{\partial a_3} = \int_\Omega \left(\alpha_k\frac{dN_3}{ds}\left|\frac{dN}{ds}a_k\right| + \beta_k\frac{d^2N_3}{ds^2}\left|\frac{d^2N}{ds^2}a_k\right| + \frac{1}{2}\frac{\partial}{\partial a_3}I(c_k) + \frac{1}{2}\frac{\partial}{\partial a_3}S(c_k)\right) ds = 0$$

$$\frac{\partial L(c_k)}{\partial a_4} = \int_\Omega \left(\alpha_k\frac{dN_4}{ds}\left|\frac{dN}{ds}a_k\right| + \beta_k\frac{d^2N_4}{ds^2}\left|\frac{d^2N}{ds^2}a_k\right| + \frac{1}{2}\frac{\partial}{\partial a_4}I(c_k) + \frac{1}{2}\frac{\partial}{\partial a_4}S(c_k)\right) ds = 0 \tag{13}$$

This equation may be rewritten concisely as

$$K_k = -F_k \tag{14}$$

and defining

$$N_i' = \frac{\partial N_i}{\partial s} \quad \text{and} \quad B_i = \frac{\partial^2 N_i}{\partial s^2} \tag{15}$$

gives

$$\boldsymbol{K}_k = \int_\Omega \alpha_k \begin{Bmatrix} N'_1 N'_1 & N'_1 N'_2 & N'_1 N'_3 & N'_1 N'_4 \\ N'_2 N'_1 & N'_2 N'_2 & N'_2 N'_3 & N'_2 N'_4 \\ N'_3 N'_1 & N'_3 N'_2 & N'_3 N'_3 & N'_3 N'_4 \\ N'_4 N'_1 & N'_4 N'_2 & N'_4 N'_3 & N'_4 N'_4 \end{Bmatrix} ds + \int_\Omega \beta_k \begin{Bmatrix} B_1 B_1 & B_1 B_2 & B_1 B_3 & B_1 B_4 \\ B_2 B_1 & B_2 B_2 & B_2 B_3 & B_2 B_4 \\ B_3 B_1 & B_3 B_2 & B_3 B_3 & B_3 B_4 \\ B_4 B_1 & B_4 B_2 & B_4 B_3 & B_4 B_4 \end{Bmatrix} ds$$

$$= \int_\Omega \left(\alpha_k \begin{Bmatrix} N'_1 \\ N'_2 \\ N'_3 \\ N'_4 \end{Bmatrix} \{ N'_1 \quad N'_2 \quad N'_3 \quad N'_4 \} + \beta_k \begin{Bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{Bmatrix} \{ B_1 \quad B_2 \quad B_3 \quad B_4 \} \right) ds$$

$$= \int_\Omega (\alpha_k [N'][N']^T + \beta_k [B][B]^T)\, ds \tag{16}$$

which is called the stiffness matrix for element c_k,

$$a_k = \begin{Bmatrix} c_{k,i} \\ \dfrac{dc_{k,i}}{ds} \\ c_{k,j} \\ \dfrac{dc_{k,j}}{ds} \end{Bmatrix} \tag{17}$$

is the nodal parameter matrix, and

$$F_k = \int_\Omega \begin{Bmatrix} \dfrac{\partial I}{\partial a_1} \\ \dfrac{\partial I}{\partial a_2} \\ \dfrac{\partial I}{\partial a_3} \\ \dfrac{\partial I}{\partial a_4} \end{Bmatrix} ds + \int_\Omega \begin{Bmatrix} \dfrac{\partial S}{\partial a_1} \\ \dfrac{\partial S}{\partial a_2} \\ \dfrac{\partial S}{\partial a_3} \\ \dfrac{\partial S}{\partial a_4} \end{Bmatrix} ds \tag{18}$$

is the nodal force function.

The energy contributed by the element c_k may be written as

$$L(c_k) = \tfrac{1}{2} K_k \left\{ c_{k,i} \quad \frac{dc_{k,i}}{ds} \quad c_{k,j} \quad \frac{dc_{k,j}}{ds} \right\} \begin{Bmatrix} c_{k,i} \\ \dfrac{dc_{k,i}}{ds} \\ c_{k,j} \\ \dfrac{dc_{k,j}}{ds} \end{Bmatrix}$$

$$+ \frac{1}{2} \int_\omega \left(\frac{\partial I}{\partial a_1} + \frac{\partial I}{\partial a_2} + \frac{\partial I}{\partial a_3} + \frac{\partial I}{\partial a_4} + \frac{\partial S}{\partial a_1} + \frac{\partial S}{\partial a_2} + \frac{\partial S}{\partial a_3} + \frac{\partial S}{\partial a_4} \right) ds$$

$$= \tfrac{1}{2} (K_k [a_k]^T [a_k] + F_k) \tag{19}$$

3 Overview of the algorithm

3.1 Tangent field potential

Before the details of the algorithm are presented, the overview of the computation of the nodal force function follows. The nodal force function consists of the tangent field potential I and the gravitational force S. The tangent field potential is computed from the subpixel edge data. The tangent field provided by the edge detector is expressed as

$$T(s) = \theta(s) \tag{20}$$

where θ is the subpixel edge orientation information provided by the edge detector [9]. For an elementary curve c_k with nodal points i and j, the tangent field potential is

$$I_k(s_{ij}) = \int_{s_{ij}} T(s)\, ds \tag{21}$$

where the integral is taken as a line integral along s.

3.2 Intersplines gravitational energy

An interspline gravitational energy S is introduced to compensate for error generated by the covariance of the edge data. The gravitational energy glues neighbouring curves together such that the two curves are continuous.

For example, given three adjacent elements c_{k-1}, c_k and c_{k+1} with their corresponding nodal points (i_{k-1}, j_{k-1}), (i_k, j_k) and (i_{k+1}, j_{k+1}), for the continuity of the global curve c it is required that

$$c_{k-1}(s)\Big|_{s=j} = c_k(s)\Big|_{s=i}$$
$$c_k(s)\Big|_{s=j} = c_{k+1}(s)\Big|_{s=i} \tag{22}$$

The gravitational energy S is the energy needed to maintain the constraints given in eqn. 22. The gravitational energy S of element c_k is defined as

$$S(c_k) = \rho \sqrt{[(c_{k-1}(s) - c_k(s))^2 + (c_k(s) - c_{k+1}(s))^2]} \tag{23}$$

where ρ is the relaxation parameter that controls the degree of influence of the gravitational energy on the global curve. The choice of ρ is a difficult and in this case, from experiments, it is found that the relaxation parameter of an elementary curve c_k is monotonically proportional to the covariance of the edge data.

3.3 Algorithm to estimate the global curve

Eqn. 19 gives the characteristics of a particular elementary curve and it is called the element equation. To compute the global curve, the element equations are assembled into $M - 1$ equations that form the system equation. The model is initialised with the initial values of $c_{k,i}$, $c_{k,j}$, $dc_{k,i}/ds$ and $dc_{k,j}/ds$ which are obtained from the edge detector [10], and thus for each element of the system equation only two unknown variables (the curvature at each nodal point, $\kappa_{k,i}$ and $\kappa_{k,j}$) for the $M - 1$ elements are to be solved from $M - 1$ equations. We assume the positions $c_{k,i}$ and $c_{k,j}$ are known accurately, but we only have an estimate of the orientations. This is because the edge detector uses a one-dimensional mask which is oriented to be perpendicular to the edge. The one-dimensional mask gives good edge location but poor edge orientation. Since the number of unknowns is twice as many as the number of equations, the system equation is not solvable. Fortunately, since every elementary curve is connected to its neighbours, the curvature of an

element at its end point is the same as the curvature of its neighbouring element. In other words, a constant curvature constraint is imposed by making the curve approximation C^1 continuous. Therefore

$$\left|\frac{d^2c_{k,j}}{ds^2}\right| = \left|\frac{d^2c_{k+1,i}}{ds^2}\right| \tag{24}$$

leaving $M - 1$ unknowns, and thus the system equation can now be solved. Integrating the curvatures yields new nodal parameters c_k and dc_k/ds for the next iteration. The process is repeated until the energy in eqn. 10 is minimised. This means that sharp features with discontinuities of curvature cannot be modelled.

The final curve that gives the minimal global energy is stored in a set giving the global curve function. This cover set is analysed to segment the global curve into its distinct components. The global energy information is used to decompose the cover set of the elementary curves into subsets that each represent a curve segment.

3.4 Decomposition of the cover set into segment sets

The cover set is an ordered set of elementary curves that represent the global curve. Each element of the cover set stores information about the spline parameters of an elementary curve and its minimum energy value. This information can be analysed to break the curve into its geometric components. When a segment breaks, there is a curvature transition in the global curve. Since the stiffness matrix $\boldsymbol{K}$ is a function of curvature, then the stiffness matrix will change at the location where the segment breaks.

The change in $L(c_k)$ is equal to the partial derivative of L with respect to c_k multiplied by a small incremental value δc_k. Thus when $\delta c_k \rightarrow 0$, if $\delta L(c_k) \rightarrow \infty$ then

$$\frac{\partial L(c_k)}{\partial c_k} = \infty \tag{25}$$

This expression implies that the maxima of the first derivative of the energy correspond to the location where segments break. Since the segmentation is based on curvature, geometric features that have nonconstant curvature such as ellipses will be segmented into more than one segment, while geometric features that have constant curvature such as a circle will be detected as one segment.

4 Results and discussions

4.1 Synthetic data

The algorithm is first tested with simulated data to verify that it works correctly. An image consisting of two segments, a straight line and a circle is geometrically created. After that the edge points are manually measured to simulate an ideal subpixel edge detector to get the edge information. The digitised edge points are spaced far enough apart to simulate the distance between pixels to subpixel positions where crosses denote the subpixel edge position (Fig. 2). The algorithm is then applied to the computed data points and the final curve giving the minimal energy is plotted (Fig. 3). The curve is then geometrically segmented using the method described in Section 3.4. The results verify that the algorithm can interpolate values between the detected edges and can segment the reconstructed curve into its geometric components.

To test the performance of the algorithm with respect to noise, an image of a circle with a step height of 200 is generated at 512 by 512 pixels resolution. The circle is centred at (255.0, 255.0) with a radius of 128.0. After that the image is deresolved into 32 by 32 pixels, and lowpass filtered to remove high spatial frequencies. Additive

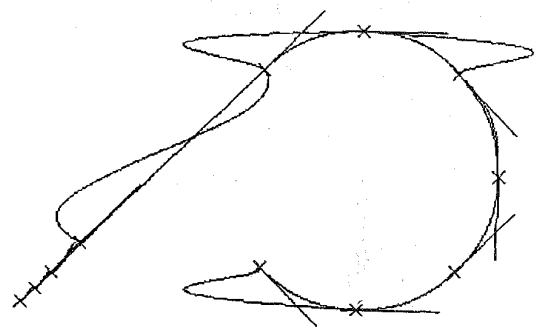

Fig. 2 *Intermediate stage in curve minimisation for a synthetic image*

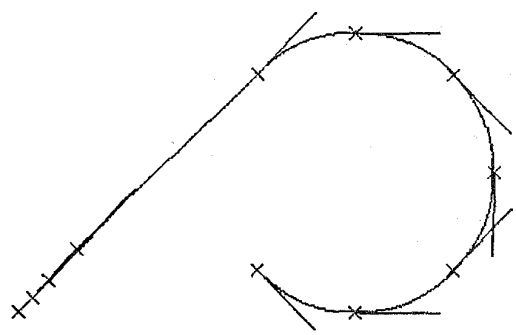

Fig. 3 *Final curve giving the minimum energy for a synthetic image*

Gaussian noise is introduced to the resulting image to produce a set of images at SNR = ∞, SNR = 100, SNR = 16, and SNR = 4 (Figs. 4–7).

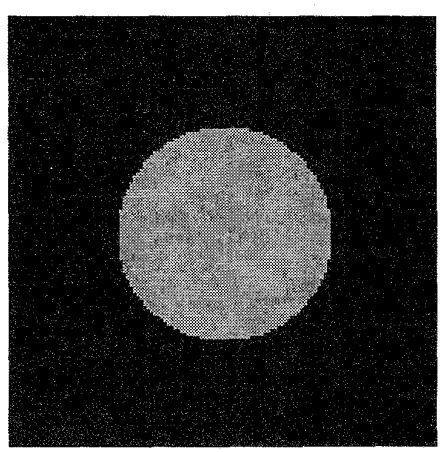

Fig. 4 *Synthetic image of circle with radius of 128 pixels for SNR = ∞*

The subpixel edge detection [2] is applied to the images. Threshold is applied at a measure-of-match of 0.68. Figs. 8–11 show the result of the edge detection displayed at 512 by 512 resolution. Boundaries are formed by fitting a circle to the detected points. Table 1 shows the results of the fitting.

From Table 1 it can be seen that even in an ideal condition (SNR = ∞), performing a statistical circle fit still produces errors. This is due to the underlying assumption of the subpixel edge detector that the edges, within a neighbourhood, form a straight line. This assumption

may lead to incorrect results when the image consists of nonstraight boundaries.

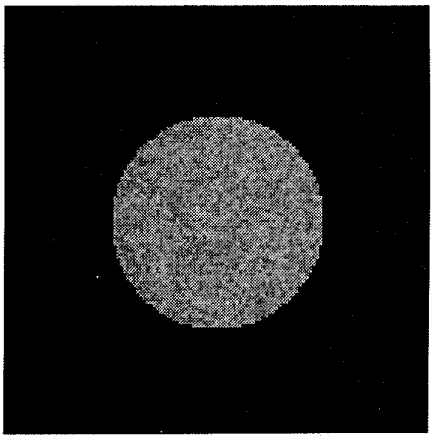

Fig. 5 *Synthetic image of circle with radius of 128 pixels for SNR = 100*

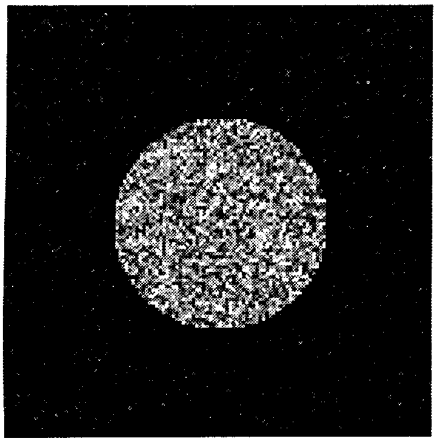

Fig. 6 *Synthetic image of circle with radius of 128 pixels for SNR = 16*

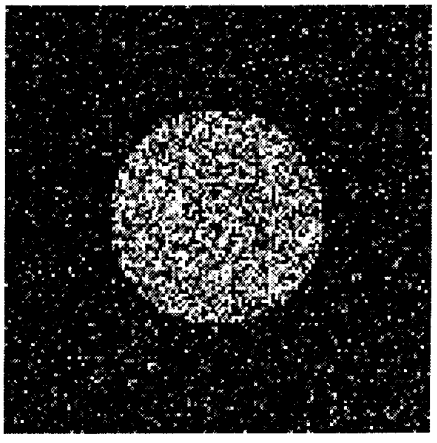

Fig. 7 *Synthetic image of circle with radius of 128 pixels for SNR = 4*

The curve detection algorithm is applied to the detected edge points, and the results are shown in Fig. 12–15. The results are displayed at 512 by 512 pixels, overlayed with the input images magnified 16 times to 512 by 512 pixels to visualise subpixel values. Table 2 shows the result of fitting a circle to the reconstructed image at 512 by 512 pixels resolution. The table shows that after the application of the curve reconstruction algorithm, the reconstructed edges fit better to a circle than before reconstruction (Table 1), and thus it can be seen that the algorithm has refined the detected edge point locations. The performance of the algorithm for

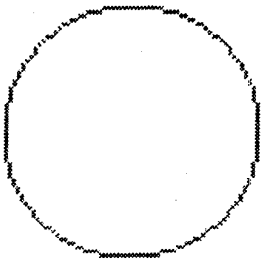

Fig. 8 *Result of subpixel edge detection for SNR = ∞*

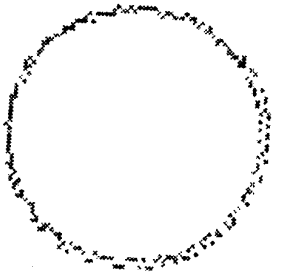

Fig. 9 *Result of subpixel edge detection for SNR = 100*

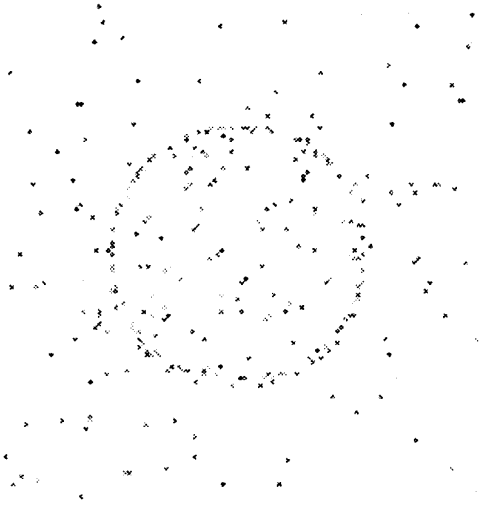

Fig. 10 *Result of subpixel edge detection for SNR = 16*

Table 1: Fitting a perfect circle to the detected points of Fig. 3. The true radius is 128.00 pixels

Parameters	SNR = ∞	SNR = 100	SNR = 16	SNR = 4
centre (*x*, *y*)	255.00, 255.00	255.00, 255,00	255.00, 255.00	255.00, 255.00
radius	128.09	129.03	129.41	133.07
deviation	0.09	0.63	4.27	16.28

low SNRs is also very good. The algorithm produces a standard deviation of 0.53 pixels, which is just slightly higher than the worst case of the uncertainty of the edge location. (Since the sampling interval is one pixel, the uncertainty of the edge location is also one pixel [11].)

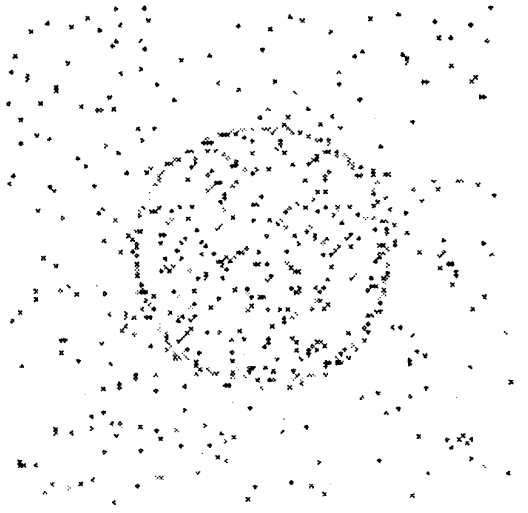

Fig. 11 *Result of subpixel edge detection for SNR = 4*

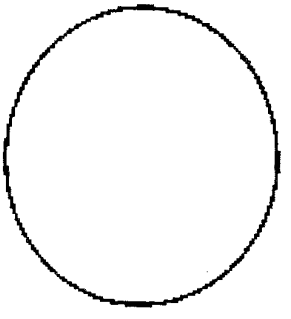

Fig. 12 *Result of the curve detection algorithm magnified 16 times for SNR = ∞*

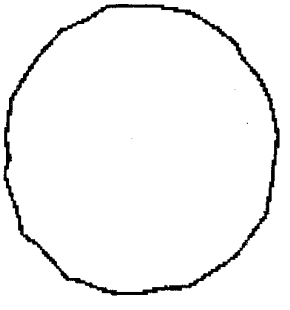

Fig. 13 *Result of the curve detection algorithm magnified 16 times for SNR = 100*

4.2 Real image

To test the performance of the algorithm for real images, an image of a lens cap is generated by digitising the lens cap at 512 by 512 pixel resolution using a Panasonic

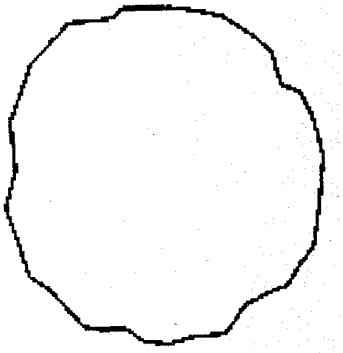

Fig. 14 *Result of the curve detection algorithm magnified 16 times for SNR = 16*

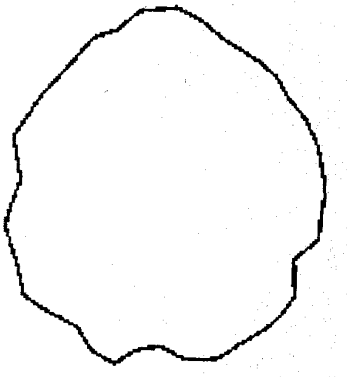

Fig. 15 *Result of the curve detection algorithm magnified 16 times for SNR = 4*

S-VHS camera attached to a Videopix frame-grabber on a Sun/Sparcstation. To visualise the subpixel reconstruction the image is deresolved into 32 by 32 pixels using a local averaging method. The resulting image is then lowpass filtered using a Butterworth filter to remove the high-frequency content of the original image. The local energy based subpixel detector is then applied to the resulting image to detect the edge points at subpixel values and then the boundary curve is reconstructed on a 512 by 512 grid. The detected curve is then overlaid on the image and magnified 16 times into a 512 by 512 image to validate the reconstructed curve against the original image at subpixel accuracy (Fig. 16). The Figure shows that the reconstruction algorithm reconstructs the boundary of the image at subpixel values. The result also show that the algorithm produces glitches on the reconstructed curve. This may be due to noise that corrupts the input data and thus the algorithm produces incorrect elementary curves.

Table 2: Fitting a circle to the reconstructed object of Fig. 5. The true radius is 128.00 pixels

Parameters	SNR = ∞	SNR = 100	SNR = 16	SNR = 4
centre (*x*, *y*)	255.00, 255.00	255.00, 255.00	255.00, 255.00	255.00, 255.00
radius	128.01	128.03	128.11	128.27
deviation	0.01	0.09	0.33	0.53

To test the algorithm with a more complicated image, the backlighted image of a precisely machined object is used. The object is digitised at 512 by 512 pixels (Fig. 17) and then deresolved and lowpass filtered into 32 by 32

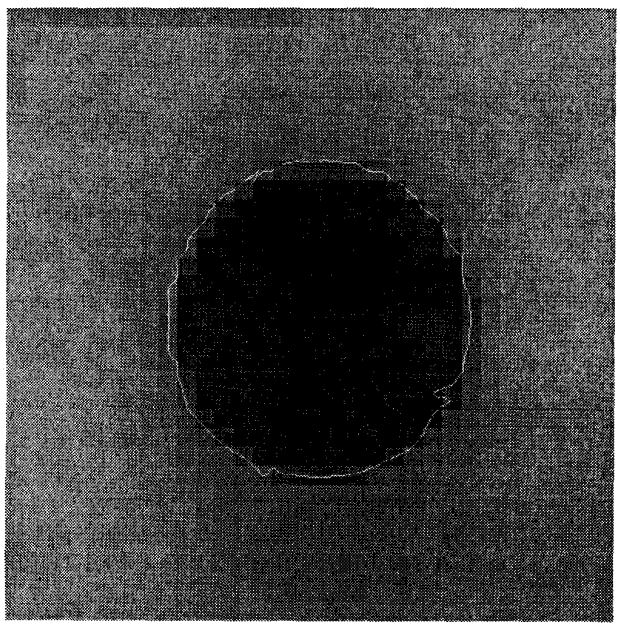

Fig. 16 *Reconstructing boundary of a lens cap image*

pixels resolution. The local energy based subpixel detector is applied to the resulting image, and then the boundaries are reconstructed at 512 by 512 grid. The result shown in Fig. 17 indicates that the algorithm has reconstructed the boundaries at subpixel accuracy. The resulting boundaries are then analysed and geometrically segmented. Fig. 18 shows the reconstructed boundaries on their own for clarity.

The problem of systematically selecting the correct values of the parameters α, β and ρ in the computation of energy functional has not been solved. Currently an *ad hoc* approach is used to determine the values. Recent work by Cootes and Taylor [12] indicates that the parameters can be estimated by looking at the properties of the input image and the algorithm is trained with shapes prior to application.

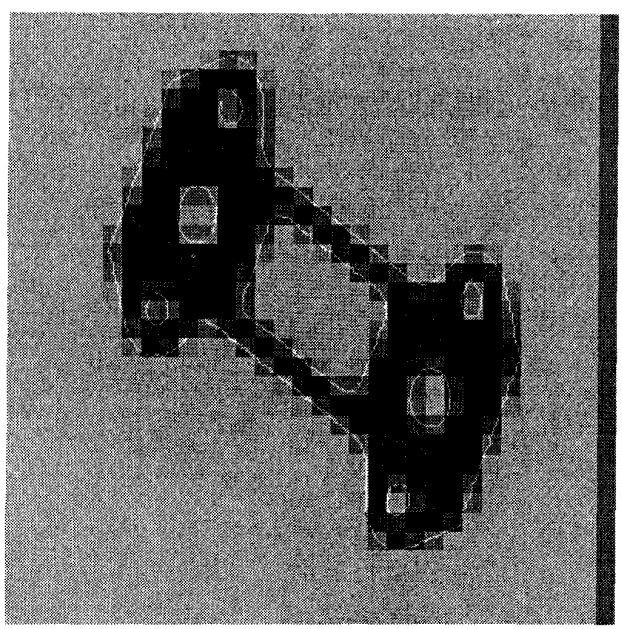

Fig. 17 *Reconstructing boundaries of a complicated object*

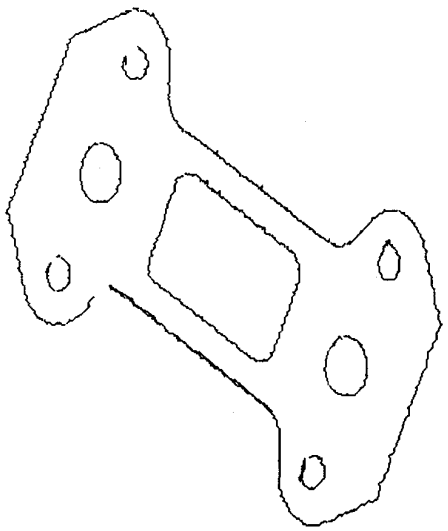

Fig. 18 *Boundaries of Fig. 17 shown on their own for clarity.*

5 Conclusions

A new approach for using deformable models in the reconstruction and segmentation of curves from subpixel edge data has been described. The approach computes the global curve by recovering the cover set from the elementary curves set. The global curve is computed by minimising the energy functional. The new approach has two advantages. First, the global curve can be generated from the cover set. Secondly, the cover set can be analysed and decomposed into subsets thus enabling the segmentation of the global curve into its components each represented by the decomposition subsets of the cover set. The results have shown that the method performs the subpixel curve extraction well at low SNRs without having *a priori* knowledge of the curve.

6 References

1 KISWORO, M., VENKATESH, S., and WEST, G.: '2-D edge feature extraction to subpixel accuracy using the generalised energy approach'. Proc. IEEE TENCON 1991 Region 10 Conf. New Dehli, India, 1991

2 KISWORO, M., VENKATESH, S., and WEST, G.: 'Subpixel precision edge modelling using the generalised energy approach'. Proc. 7th Scandinavian Conf. on Image Analysis, Aalborg, Denmark, 1991

3 KASS, M., WITKIN, A., and TERZOPOULOS, D.: 'Snakes: active contour models', *Int. J. Comput. Vis.*, 1988, **1**, pp. 321–331

4 ZUCKER, S.W., DAVID, C., DOBBINS, A., and IVERSON, L.: 'The organization of curve detection: Coarse tangent fields and fine spline coverings'. Proc. 2nd Int. Conf. Computer Vision, Tampa, Florida, USA, 1988, pp. 568–577

5 KARAOLANI, P., SULLIVAN, G., BAKER, K., and BAINES, M.: 'A finite element method for deformable models'. Proc. British Machine Vision Conf., Oxford, UK, 1990, pp. 73–78

6 GELFAND, M., and FORMIN, S.: 'Calculus of variations' (Prentice-Hall, 1963)

7 ZIENKIEWICZ, O.: 'The finite element method' (McGraw-Hill, 3rd ed., 1977)

8 HINTON, E., and OWEN, D.: 'An introduction to finite element computations' (Pineridge Press Ltd., UK, 1979)

9 KISWORO, M.: 'Investigations into subpixel-precision feature