Sparse Finite Elements for Geodesic Contours with Level-Sets*

Martin Weber[1], Andrew Blake[2], and Roberto Cipolla[1]

[1] Department of Engineering, University of Cambridge, UK,
{mw232,cipolla}@eng.cam.ac.uk
http://mi.eng.cam.ac.uk/research/vision/
[2] Microsoft Research, Cambridge, UK

Abstract. Level-set methods have been shown to be an effective way to solve optimisation problems that involve closed curves. They are well known for their capacity to deal with flexible topology and do not require manual initialisation. Computational complexity has previously been addressed by using *banded* algorithms which restrict computation to the vicinity of the zero set of the level-set function. So far, such schemes have used finite difference representations which suffer from limited accuracy and require re-initialisation procedures to stabilise the evolution. This paper shows how banded computation can be achieved using finite elements. We give details of the novel representation and show how to build the signed distance constraint into the presented numerical scheme. We apply the algorithm to the geodesic contour problem (including the automatic detection of nested contours) and demonstrate its performance on a variety of images. The resulting algorithm has several advantages which are demonstrated in the paper: it is inherently stable and avoids re-initialisation; it is convergent and more accurate because of the capabilities of finite elements; it achieves maximum sparsity because with finite elements the band can be effectively of width 1.

1 Introduction

Level-set methods are generally useful for the analysis of image data in 2D and 3D when one has to solve an optimisation problem with respect to an interface [1–7]. In this paper, we will present a novel numerical scheme to solve the problem of minimising a certain geodesic length in two dimensions which was proposed [8, 9] to achieve the attraction to contours in images. For the geodesic model, the cost C (*Riemannian length*) of an interface Γ is the integral of a local density g (*Riemannian metric*) over the interface:

$$C = \int_{\Gamma} g \tag{1}$$

* This work was supported by the EPSRC, the Cambridge European Trust and a DAAD-Doktorandenstipendium (Germany).

where the standard Lebesgue measure is used to integrate the scalar function g over the set Γ. Differential minimisation of C leads to a *gradient descent* scheme.

Level-set methods [1] introduce a level-set function[3] ϕ to represent the interface Γ implicitly as the zero level-set: $\Gamma := \phi^{-1}(0)$. The implicit representation links ϕ (as the introduced *analytic entity*) with the *geometric entity* Γ: $\phi \mapsto \Gamma(\phi)$ and allows for changes in the topology during the evolution. Furthermore, it was pointed out [10] that this relationship can be made one-to-one by imposing the signed distance constraint. The conceptual advantage is then that ϕ is (up to a sign) uniquely determined by Γ and that one can also write $\Gamma \mapsto \phi(\Gamma)$. In this way ϕ gets the intrinsic geometric meaning as the distance function for Γ.

1.1 Differential minimisation and level-set evolution

For the evolution, one introduces an evolution parameter $t \in R$ and ϕ becomes *time*[4] dependent. One starts with an initial function $\phi(0,.)$ and prescribes an evolution $\phi(t,.)$ that tends towards local minima of the cost C using gradient descent. In the level-set formulation, the gradient descent is expressed in the evolution equation, a *partial differential equation* (PDE):

$$\tfrac{d\phi}{dt} = \beta \tag{2}$$

where, at the interface Γ, β is the differential of the cost[5]: $\beta|_\Gamma := -\frac{\delta C}{\delta \phi}$ and is defined globally as in [10] to maintain the signed distance constraint. The signed distance constraint is well known for its desirable conceptual and numerical properties [10]. Where ϕ is differentiable, we have $|\nabla\phi(x)| = 1$ and, for $x \in \Gamma$, one has particularly simple expressions for the curve's *normal* $N(x) = \nabla\phi(x) \in S^1$ and curvature $\kappa(x) = \nabla^2\phi(x) \in \mathbb{R}$.

1.2 Previous numerical problems of evolving level-sets

In the following, u denotes the numerical representation of the level-set function ϕ. There are two major issues in the numerical implementation of the PDE (2): one is efficiency and the other is stability. Potential inefficiency arises from the need to maintain an entire (2D) function u in order simply to obtain the curve Γ. "Banded" schemes have been suggested [11, 2, 12] which restrict computations to the immediate neighbourhood of Γ. Because the interface Γ can leave the band, those schemes require an the algorithm to extend the sparse representation u as signed distance map. However, the extension outside the current band is only consistent if the signed distance property is preserved by the evolution [10].

The implementation of the evolution (2) on a grid of pixels in finite difference schemes [1, 11, 2, 3] results in a stability problem, illustrated by the bunching of levels in Figure 1. Although the signed distance constraint used by Gomes and

[3] ϕ is a continuous, real valued function

[4] One refers to the parameter t as *time* although it is not related to physical time.

[5] $\frac{\delta}{\delta\phi}$ denotes variational differentiation and $\beta|_\Gamma$ is detailed in (18) for the cost (1).

Faugeras [10] maintains the equal spacing of levels in principle, the numerical implementation (discretisation and finite numerical accuracy) still causes a drift which eventually destroys the equal spacing. The bunching of levels destabilises the evolution and affects the convergence. Therefore, previous methods required a separate re-initialisation procedure in order to restore the signed distance property. One also needs to select a suitable frequency for invoking the re-initialisation procedure to maintain stability of the evolution.

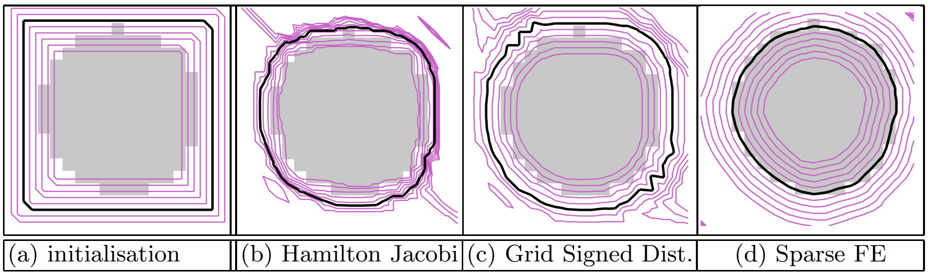


Fig. 1. Finite element approach has better stability: The figure compares three different geodesic contour implementations. The initial shape is a square (18×18 pixels) and the target shape is a discrete circle (shaded pixels). The zero-level is indicated as a dark line in each case and neighbouring levels are drawn with a level spacing of 0.5 pixel units. (a) initialisation by a rectangle. The following images display the propagation of the level sets when a time step of $\Delta t = 0.1$ is used to evolve the level-set function to $t = 20$. (b) The Hamilton-Jacobi evolution [2] causes a bunching of levels which destabilises the evolution and requires a separate re-initialisation procedure. (c) The signed distance evolution [10] in grid representation improves the stability but still has a slow drift from the signed distance property which also requires a re-initialisation procedure. (d) Novel sparse finite element evolution maintains the signed distance constraint indefinitely, with no need for re-initialisation.

1.3 Novel numerical solution: the sparse finite element approach

In this paper, we solve the problems of efficiency and stability by proposing a novel scheme that uses finite elements [13, 14] to represent and evolve u. Finite elements have been used before in the context of level-set methods: in [15] a finite element scheme is used to represent temperature changes along an interface, while the interface itself is evolved using a finite difference scheme. Preußer and Rumpf [16] work with 3D cubical elements (of mixed polynomial degree) and evolve all levels in the computational domain. Our method introduces a sparse simplicial element representation and combines a weak form of the geodesic evolution equation with the inbuilt preservation of the signed distance constraint:

- The band is represented as a simplicial complex, over which simplices are continually added and deleted, in a fashion which is integrated and harmonious with the differential evolution of u. No mode switching is required to deal with the interface Γ falling out of the band.

- The simplicial representation of the band allows it to have minimal width, resulting in enhanced efficiency. Derivatives are treated by our weak formulation with no need for conditional operators. As a consequence, no second order derivatives of the level-set function have to be computed explicitly.
- With finite elements, the function u is defined everywhere, not just at grid locations, and sub-grid accuracy is particularly straightforward.
- The signed distance constraint is maintained actively in a stable, convergent fashion, even over indefinite periods of time. This results in an algorithm which is demonstrably more stable (Figure 1) and more accurate (Figure 2) than previous approaches [2, 10].

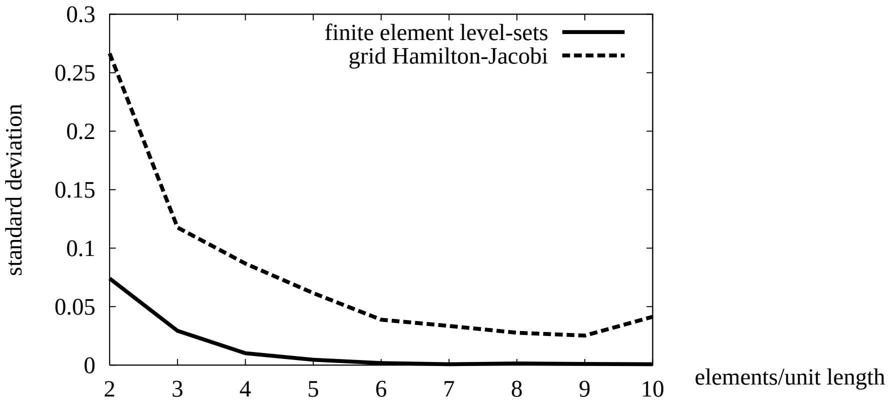


Fig. 2. Superior accuracy: The diagram shows deviations of the detected interface from the unit disc. The unit disc is used as target shape of the geodesic evolution (see Figure 1). The diagram shows the deviations (vertical-axis) between the result of the level-set evolution and the target shape when the pixel resolution is varied (horizontal-axis). The lines in the diagram correspond to the Hamilton-Jacobi scheme and the novel method presented in this paper. The new method clearly performs better. The increase in deviation for the grid method on the right is caused by numerical instabilities that occur when no re-initialisation is used.

2 Efficient representation with 'banded' finite elements

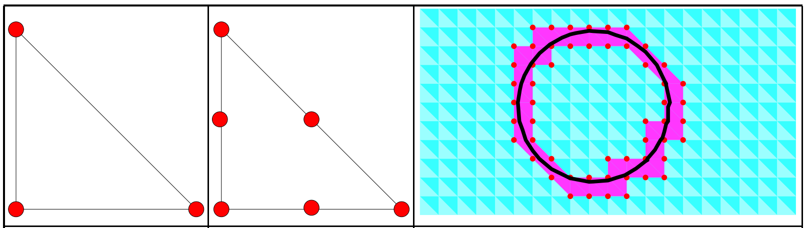
The new numerical representation u consists of a *global* and a *local* component:

- The local component inside each element is a polynomial in $d = 2$ variables which prescribes the location of the zero level-set inside the element.
- The global component, the 'band', is a *simplicial complex* that consists of the minimal set of elements that contain the zero level-set (Figure 3). We refer to elements that are members of the complex as being *active*. The representation differs from standard finite element representations in that the

complex is sparse and in that it has to be changed dynamically to maintain the containment property for the evolving interface Γ.

2.1 Local representation: element polynomial

Following standard methods found in finite element methods [13, 14], we use the standard d-simplex to represent u locally as a polynomial of fixed degree p in d dimensions. The standard simplex T_0^d is defined to be the convex hull of the standard Euclidean basis vectors $b_1, b_2, .., b_d \in \mathbb{R}^d$ and the origin $b_0 = 0$. In $d = 2$ dimensions, the standard simplex is simply a triangle as in Figure 3. We adopt



(a) $p = 1$ (b) $p = 2$ (c) global representation

Fig. 3. Sparse representation (2D): Inside each simplex, the level-set function u is defined by the values of the nodes. (a) shows the location of nodes within a first order element and in (b) for a second order element. (c) For the global representation, the plane is partitioned into standard simplices (shaded lightly) and computations are restricted to the active complex $\mathcal{A}$ (shaded darker) which consists of the minimal set of elements that contain the zero level-set (in this example a circle).

the following terminology from finite element methods [13, 14]:

- a *node* is a location $p_i \in T_0^d$ together with a real value and we position the nodes as indicated in Figure 3 on the grid $\frac{1}{p}\mathbb{Z}^d$.
- the *nodal basis function* e_i associated with node i is the unique [13] polynomial of degree p that evaluates at the nodes to: $\forall j \;\; e_i(p_j) = \delta_{ij}$.
- u is a linear combination of the nodal basis functions: $u = \sum_i \mathbf{u}_i \, e_i$.

The fact that integration is a linear operation will enable us to express all occurring integrals (9),(11) as linear combinations of a few integral constants (15),(20).

2.2 Global representation: active simplicial complex

Our global representation of the functional u consists of the active complex $\mathcal{A}$ covering the area Ω. Each $d-simplex$ of the complex is mapped to a standard element and defines in this way a global functional u on the area Ω. By the sharing of nodes, we obtain a global functional that is automatically continuous[6].

[6] This is a significant advantage over representations that do not enforce continuity (like for instance the *surfel* representation used in [17]).

We restrict our current exposition of global representations to the 2-dimensional case. Note however, that our formulation is equally applicable for hyper-surfaces of higher dimensions (e.g. in the 3D case, one can partition the space by choosing the *Delaunay tetrahedrisation* [18] of the node set $\mathbb{Z}^3 \cup \left((\frac{1}{2}, \frac{1}{2}, \frac{1}{2})^\top + \mathbb{Z}^3\right)$ where all tetrahedrons are of the same type). In 2D, a rectangular area (e.g. the image plane) can be partitioned using standard simplices as illustrated in Figure 3(c).

3 Stable dynamics to evolve the novel representation

Having defined the efficient numerical representation of u, we now show how a stable evolution can be defined which is at the heart of our method. In order to avoid re-initialisation procedures, we integrate the signed distance property into the evolution equations by introducing an error functional r which penalises deviations from the desired interface motion $\beta|_\Gamma$ as well as deviations from the signed distance property. The evolution algorithm then minimises this functional.

3.1 Components of the evolution equations

Firstly, unlike [10], we express the signed distance constraint in the following form:

$$(\nabla_x \, u)^2 - 1 = 0 \tag{3}$$

Secondly, the desire to move the interface at a normal speed $\beta|_\Gamma$ simply implies

$$u_t|_\Gamma = \beta_\Gamma \tag{4}$$

for the update of u by (2). We consider interface motion of the general form [2]

$$\beta|_\Gamma(t, x, N, \kappa) \tag{5}$$

which involves *external forces* by the dependence on $x \in \mathbb{R}^d$ and $t \in \mathbb{R}$ as well as the *intrinsic quantities* orientation N and curvature κ. Note that this means that $\beta|_\Gamma$ depends on the 2^{nd} derivative of u and that we have $\beta|_\Gamma(t, x, N, \kappa) = \beta|_\Gamma(t, x, \nabla u, \nabla^2 u)$ due to the signed distance constraint. In this paper we apply the general method to the geodesic problem (1) for which $\beta|_\Gamma$ is given by (18).

3.2 Discrete dynamics

Now the evolution of the level-set function is set-up in discrete space and time, in terms of the displacement v of the function u over a time-step Δt:

$$u(t + \Delta t, .) = u(t, .) + v(t, .). \tag{6}$$

Here v is represented over the finite element basis, in the same way as u is, and represents displacement for a time Δt at velocity β:

$$v = \Delta t \; \beta(u + v) \tag{7}$$

where we have chosen to evaluate β at $u + v$ (instead of u in the explicit scheme) to obtain an implicit scheme [19] which does not limit the magnitude of Δt.

3.3 ***Weak formulation*** of evolution dynamics

Inspired by the *Petrov-Galekin formulation* [13, 14] used in finite element methods, we employ a *weak formulation* of (3) and (4) with the following advantages:

- It allows us to measure and use the constraint equations for the entire active area Ω, and not just at discrete, sampled locations [10].
- It allows for curvature dependent interface motion (5) even in the case of first order elements ($p = 1$) by the use of Green's theorem.
- It gives an appropriate regularisation of derivative operators without the need of switch-operators found in grid representations [2, 10].

In the Petrov-Galekin form, one uses the nodal basis functions e_i $i \in \{1, ..., n\}$ as *test functions* to measure deviations from the desired evolution properties (3) and (4). First, the signed distance equation (3) becomes a set of equations:

$$z_1^i = 0, \quad \text{for} \quad i = 1, \ldots, n \tag{8}$$

where

$$z_1^i := \int_\Omega \left((\nabla u + \nabla v)^2 - 1 \right) e_i. \tag{9}$$

Secondly, the velocity law (7) is expressed as

$$z_2^i = 0, \quad \text{for} \quad i = 1, \ldots, n \tag{10}$$

where

$$z_2^i := \int_\Omega (v - \Delta t\, \beta)\, e_i. \tag{11}$$

We now introduce[7] an optimisation problem to determine the update of the level-set function which minimises deviations from (8) and (10).

3.4 Level-set update equations as optimisation problem

The two sets of equations (9) and (11) represent an overdetermined system of $2n$ equations in n unknowns. We measure the deviations in the following functional:

$$r^2 := |z_1|^2 + \alpha^2 |z_2|^2, \tag{12}$$

where $z_1 = (z_1^1, \ldots, z_1^n)^\top$ and similarly for z_2, and $\alpha \in \mathbb{R}_+$ is an arbitrary positive constant that balances the competing terms in the optimisation problem.

The functional can be written compactly by expressing z_1^i and z_2^i in terms of the node values $\mathbf{v} = (v_1, \ldots, v_n)^\top$ for the displacement v, and similarly for u:

$$z_1^i = \mathbf{u}^\top Q^i \mathbf{u} - k^i + 2\mathbf{u}^\top Q^i \mathbf{v} + h_i(\mathbf{v}) \tag{13}$$

$$z_2^i = P^i\, \mathbf{v} - \Delta t \int_\Omega e_i\, \beta \tag{14}$$

[7] The optimisation problem introduced here is not to be confused with the optimisation problem (1) that gives rise to the differential evolution $\beta|_\Gamma$ in the first place.

where $h_i(\mathbf{v}) := \mathbf{v}^\top Q^i \mathbf{v}$ and where constants k, P, Q are defined as:

$$k^i := \textstyle\int_\Omega e_i, \quad P_{ab} := \int_\Omega e_a\, e_b, \quad Q^i_{ab} := \int_\Omega \langle \nabla e_a, \nabla e_b \rangle\, e_i \tag{15}$$

The quantities (k, P, Q) can be pre-computed analytically, and stored as constants. Note that the deviation z_1 is almost affine in the unknown speed v since the $h_i(\mathbf{v})$ are small, provided u approximates the signed distance property and if the time step Δt is sufficiently small. In that case (13) can be linearised, ignoring h, by replacing z_1^i by $\tilde{z}_1^i = \mathbf{u}^\top Q^i \mathbf{u} - k^i + 2\mathbf{u}^\top Q^i \mathbf{v}$. We solve the linear least-square problem numerically by using the *conjugate gradient method* [20]. We exploit the sparsity over the computational grid which allows the linear simultaneous equations to be expressed in banded form over the nodal basis. Using the banded form, we solve for $\mathbf{v}$ in $\mathcal{O}(n)$ where n denotes the number of nodes and is proportional to the length of the interface in element units.

3.5 How the global evolution ensures the containment property

For our method to be viable, we require to detect changes in the active complex $\mathcal{A}$ efficiently. The new method is outlined in Algorithm 1. After each local evolution (lines 2-4 of **evolve**), we adjust the active complex by adding and removing elements (lines 5-14 of **evolve**) to ensure that it contains the current interface Γ and that it is minimal with that property. The initialisation of neighbouring elements is well defined; this is the case because although we restrict the *numerical representation* to the sparse complex, u is indeed defined globally by the signed distance constraint. This justifies the use of the extrapolation procedure in **activate()**. Extrapolation[8] is natural since u is represented in full functional form and does not require any separate interpolation mechanism for evaluation.

The maintenance of $\mathcal{A}$ (activation and removal of elements) is performed efficiently in **evolve** by looking at the edges of the complex. Note that the level-set function along an edge is a polynomial of degree p and that it is determined by the $p + 1$ nodes located along the edge [13]. Hence, the problem of deciding where $\mathcal{A}$ needs modification reduces to the problem of finding the roots of the edge-polynomial which is straightforward for linear and quadratic elements.

4 Geodesic contour detection with sparse finite elements

It has been known for some time [8, 9] that the problem of contour detection can be cast into the problem of minimising a Riemannian length functional that is induced by the image. In order to define the cost functional C (1), one starts by introducing the *local* measure for edges g, which is normalised so that $g(x) \in [0, 1]$. The basic construction adopted here uses a positive *edge detector* function f. In this paper, we employ the following monotonic function [19]

$$g := 1 - \exp\left(-\tfrac{a}{|\nabla f_\sigma|^q}\right) \tag{16}$$

[8] For 1st order elements the value u_c in element (abc) is $u_c = u_a + u_b - \bar{u}_c$ where $\bar{u}_c$ is the known value of the node that is obtained by reflecting node c along the edge ab.

Algorithm 1 Geodesic level-set algorithm with sparse finite elements

detect geodesic contour:
smoothen image with Gaussian(σ)
compute image metric g {see (16)}
initialise level-set u as sparse finite element complex $\mathcal{A}$
repeat
  evolve(u,g)
until converged
output interface $\Gamma(u)$
return

activate(T):
for all nodes V of element T **do**
  if $V \notin \mathcal{A}$ **then**
    initialise V (extrapolate from active T-adjacent elements)
  end if
end for
add element to $\mathcal{A}$
return

evolve(u,g):
{Update u:}
compute A_1, A_2 and b_1,b_2 such that $z_l = A_l\mathbf{v} + b_l$ {see (13), (19)}
solve the least square equation $A^\top(A\mathbf{v} + b) = 0$ (C.G. method)
$u \leftarrow u + v$ {see (6)}
{Now update the active complex $\mathcal{A}$:}
for all edges $E \in \mathcal{A}$ that contain a root **do**
  for all E-adjacent elements $T \notin \mathcal{A}$ **do**
    activate(T)
  end for
end for
for all $T \in \mathcal{A}$ **do**
  if no edge of T is active **then**
    remove T from $\mathcal{A}$
  end if
end for
return

where f_σ is smoothed with a Gaussian of scale parameter σ and a, q are real constants that control the scale of sensitivity and the slope as parameters of the normalising function. For the experiments in this paper, we employ the following edge detector functions f:

– **Colour edges:** $f(x) = I(x)$ where the image I has values in rgb-space.
– **Gaussian colour edges:**

$$f(x) = \exp\left(-\tfrac{\gamma}{2}(I(x) - \bar{y})^\top \Sigma^{-1}(I(x) - \bar{y})\right) \tag{17}$$

here, $\bar{y}$ is a fixed colour, Σ is a covariance matrix, γ a positive constant and I has values in rgb-space. For our examples, we obtain the covariance matrix by sampling m pixel-values $\{y_j\}$ in a user-defined region:

$$\bar{y} := \tfrac{1}{m}\sum_j y_j \qquad \Sigma := \left(\tfrac{1}{m}\textstyle\sum_j y_j\, y_j^\top\right) - \bar{y}\,\bar{y}^\top$$

4.1 Geodesic evolution equation

The velocity function β which asymptotically minimises C for signed distance functions can be shown [12, 21] to be:

$$\beta|_\Gamma = \langle \nabla g, \nabla\phi \rangle + g\left(\nabla^2\phi + c\right) \tag{18}$$

where c is the coefficient of the so-called “balloon force” [22, 3, 12] and we refer to [7] for the connection to parametric snakes [23] that has been discussed in the literature. The “balloon force” term does not arise from the cost minimisation but is useful in certain situations [21].

4.2 Numerical form of the evolution

In order to apply the new numerical method of Section 3, we have to perform the evaluation of z_2 in (14) for the geodesic case. It is well known ([19], *implicit scheme*) that the evaluation of $\beta(u+v)$ instead of $\beta(u)$ in (14) is preferable since it does not impose a limit on the time-step Δt. Note that unlike in the case of the diffusion on images [19], the implicit scheme is computationally feasible due to the sparse representation. Using Green's formula, it can be shown that the weak form (14) of the velocity equation (18) becomes:

$$z_2^i = \Big(P_i - \Delta t\,[(B_i - Q_i)\mathbf{g}]^\top\Big)\mathbf{v} - \Delta t\,\big(c\,P_i\,\mathbf{g} + \mathbf{u}^\top(B_i - Q_i)\,\mathbf{g}\big) \tag{19}$$

where we have also represented g in nodal basis and where we have introduced

$$B_{ik}^j := \int_{\partial\Omega} e_k\, e_i\, \langle \nabla e_j, \mathcal{V}\rangle \tag{20}$$

as boundary integral constants with $\mathcal{V}$ denoting the outward normal along $\partial\Omega$.

Proof. (Outline) by (18) the interface speed is $\beta|_\Gamma = \mathrm{div}(g\,N) + c\,g$ and, by the signed distance property $N = \nabla u$ in the explicit scheme and $N = \nabla(u+v)$ in the implicit scheme. Inserting β into (14) one obtains the following non-trivial term in the expression for z_2^i: $-\Delta t \int_\Omega \mathrm{div}(g\,N)\,e_i$. Using Green's formula, we can move one differentiation onto the test function e_i and hence trade the second order derivatives for a boundary integral:

$$\int_\Omega \mathrm{div}(g\,N)\,e_i = \int_{\partial\Omega} e_i\, g\,\langle N, \mathcal{V}\rangle - \int_\Omega g\,\langle N, \nabla e_i\rangle$$

now (19) follows by writing $u = \sum_j \mathbf{u}_j\, e_j$ and $g = \sum_k \mathbf{g}_k\, e_k$. □

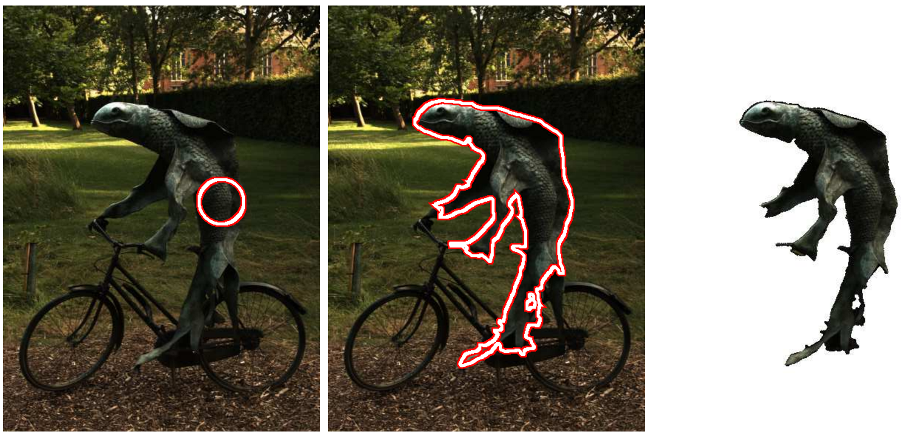
5 Results

This Section presents experimental results (Figures 4, 5, 6) obtained with our method using first order elements ($p = 1$). The smoothing constant σ and the type of metric are selected manually for each example. In order to complete the definition of the metric we determine the parameters a and q in (16) automatically such that the average gradient magnitude $\langle|\nabla f_\sigma|\rangle$ over the image results in $g = \frac{1}{2}$ and that the slope of g with respect to the gradient magnitude equals $-1/\langle|\nabla f_\sigma|\rangle$ at this point.

Numerically, we compute the gradient using central difference operators and define the finite element functional $\mathbf{g}$ by introducing one node per pixel. In this way g is defined over the entire image domain (unlike the sparse functional u). For the evolution, we choose the balloon force constant c individually and set $\alpha = 1$ in (12). In our experiments we also set $\Delta t = 1$. In principle, one could choose a larger time step in the implicit scheme but we limit Δt here to ensure a small value of h in (13) and not to overshoot any details during the evolution.

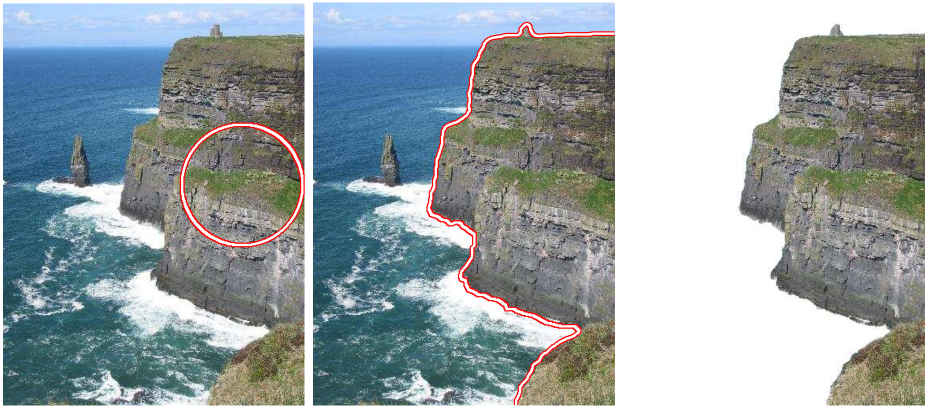
(a) (b) (c)

Fig. 4. Nested contour detection: (a) The outline of an image of a wooden Krishna-figure (266×560 pixels) is used to initialise the level-set function and to extract a Gaussian distribution of the background. (b) Geodesic minimisation (parameters: $\gamma = 0.1, \sigma = 2, c = 0.4$) leads to the detection of the outer contour. Note the faithful capture of sculptural details such as the pipe which require a stabilised method. (c) Using our metric inversion method [21] the nested contours (holes) are detected automatically.



(a) (b) (c)

Fig. 5. Cycling fish sculpture: (a) A user-specified circle is placed on the image (429×577 pixels) as initial level-set and to define a Gaussian colour distribution. (b) Geodesic evolution (parameters: $\gamma = 0.1, \sigma = 1.5, c = -0.3$) with an 'inflating' balloon-force results in the displayed contour. (c) shows the pixels for which $u < 0$.



(a) (b) (c)

Fig. 6. Cliff example: The input image (384×512 pixels) is displayed in (a) with the user defined initial level-set superimposed. (b) shows the converged contour and (c) the obtained segmentation. In order to define the metric g , a Gaussian in rgb-space that represents the colour distribution inside the circle was defined ($\sigma = 1.5, \gamma = 10^{-1.5}$). A negative balloon-force (c= -0.3) was employed to 'inflate' the initial circle towards the boundaries of the region.

6 Conclusions and future work

We have proposed a new level-set scheme and algorithm for automatically fitting image contours, robustly and efficiently. Even nested contour structures are detected automatically by applying the *metric-inversion* method [21] to the algorithm. We exploited the signed distance constraint systematically to obtain a sparse representation while having a well defined global continuation. For the numerical representation, we replaced previously used finite difference methods and use a dynamically changing finite element complex instead. We incorporated the signed distance constraint into the evolution equations and obtained an algorithm that avoids the periodic re-initialisations required by others. We demonstrated the resulting improvements with respect to stability and accuracy. The efficiency, in common with previous schemes, derives from the banded representation, and this is enhanced by the introduction of finite elements which minimises the band width. Using a weak formulation of the evolution equations, we were able to accurately implement curvature dependant evolutions without having to explicitly compute second order derivatives. Various further developments and investigations are underway:

- Extension to the 3D case with tetrahedral elements. As mentioned in Section 2.2, 3-space can be partitioned using a regular mesh of tetrahedrons with a single element shape. Applications include medical imaging and surface reconstruction for model acquisition.
- Implementation of efficient second order finite elements.
- Applications using more sophisticated metrics (e.g. texture segmentation).

References

1. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. J. of Comp. Phys. **79** (1988) 12–49
2. Sethian, J.: Level Set Methods. Cambridge University Press, Cambridge (1999)
3. Sapiro, G.: Geometric Partial Differential Equations and Image Processing. Cambridge University Press (2001)
4. Yezzi, A., Soatto, S.: Stereoscopic segmentation. In: Proc. IEEE Int. Conf. on Computer Vision. Volume I. (2001) 56–66
5. Faugeras, O.D., Keriven, R.: Complete dense stereovision using level set methods. In: Proc. European Conf. on Computer Vision. LNCS 1406, Springer (1998) 379–393
6. Osher, S., Paragios, N.: Geometric Level Set Methods in Imaging Vision and Graphics. Springer, New York (2003)
7. Kimmel, R.: Curve Evolution on Surfaces. PhD thesis, Dept. of Electrical Engineering, Technion, Israel (1995)
8. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. IEEE Int. Conf. on Computer Vision. (1995) 694–699
9. Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, Jr., A.: Gradient flows and geometric active contour models. In: Proc. IEEE Int. Conf. on Computer Vision. (1995) 810–815
10. Gomes, J., Faugeras, O.: Reconciling distance functions and level sets. Journal of Visual Communication and Image Representation **11** (2000) 209–223
11. Malladi, R., Sethian, J., Vemuri, B.: Shape modeling with front propagation: A level set approach. IEEE Trans. Pattern Analysis and Machine Intelligence **17** (1995) 158–175
12. Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M.: Fast geodesic active contours. IEEE Trans. Image Processing **10** (2001) 1467–1475
13. Zienkiewicz, O., Morgan, K.: Finite Elements & Approximation. John Wiley & Sons, NY (1983)
14. Johnson, C.: Numerical Solution of Partial Differential Equations by the Finite Element Method. Cambridge University Press (1987)
15. Ji, H., Chopp, D., Dolbow, J.E.: A hybrid extended finite element/level set method for modeling phase transformations. International Journal for Numerical Methods in Engineering **54** (2002) 1209–1233
16. Preußer, T., Rumpf, M.: A level set method for anisotropic geometric diffusion in 3D image processing. SIAM J. on Applied Math. **62(5)** (2002) 1772–1793
17. Carceroni, R., Kutulakos, K.: Multi-view scene capture by surfel sampling: From video streams to non-rigid 3D motion, shape and reflectance. In: Proc. IEEE Int. Conf. on Computer Vision. (2001) II: 60–67
18. Edelsbrunner, H.: Geometry and Topology for Mesh Generation. Cambridge University Press, Cambridge (2001)
19. Weickert, J., ter Haar Romeny, B., Viergever, M.: Efficient and reliable schemes for nonlinear diffusion filtering. IEEE Trans. Image Processing **7** (1998) 398–410
20. Schwarz, H.: Numerische Mathematik. Teubner, Stuttgart (1993)
21. Weber, M., Blake, A., Cipolla, R.: Initialisation and termination of active contour level-set evolutions. In: Proc. IEEE workshop on Variational, Geometric and Level Set Methods in Computer Vision. (2003) 161–168
22. Cohen, L.: On active contour models and balloons. Computer Vision, Graphics and Image Processing **53** (1991) 211–218
23. Cipolla, R., Blake, A.: The dynamic analysis of apparent contours. In: Proc. IEEE Int. Conf. on Computer Vision. (1990) 616–623