Initialisation and Termination of Active Contour Level-Set Evolutions

Martin Weber*

Department of Engineering
University of Cambridge
Cambridge CB2 1PZ
mw232@cam.ac.uk

Andrew Blake

Microsoft Research
7 J J Thomson Ave
Cambridge CB3 0FB, UK
ablake@microsoft.com

Roberto Cipolla

Department of Engineering
University of Cambridge
Cambridge CB2 1PZ
cipolla@eng.cam.ac.uk

Abstract

This paper deals with the evolution control of level-sets in the context of contour detection. There is a considerable amount of existing work on PDEs for geodesic contour detection and the investigation of level-set implementations has recently resulted in efficient and stable numerical realisations of the differential evolution. This paper is based on a finite element implementation for signed distance level-set evolutions and focuses the attention to the initialisation and termination of level-set evolutions. An initialisation consists of an initial signed distance function which corresponds to some implicit curve. We discuss two types of initialisation. We generalise the commonly used a priori type, which can be a rectangle the size of the image, to include more general initial shapes. We show that the initial shape does not have to be closed and can for instance be a single line. The second type of initialisations is not specified by the user but a result of previous level-set evolutions. This type of initialisation is useful when different evolution equations are to be alternated and can be used for instance to detect nested contours or in multi-resolution techniques. For the termination of a geodesic evolution, we introduce an automatic stopping condition by looking at the Riemannian length of the implicit curve as the quantity that is subject to the minimisation. It turns out that the length can be computed efficiently from the employed finite element representation and used to terminate the gradient descent.

1 Introduction

Level-set methods have become powerful tools for many geometric problems in the analysis of image data in 2D and 3D [11, 10, 14, 5]. Level-set methods are generally useful when one has to solve an optimisation problem with respect to an interface. To that extent, one assigns a cost $C(\Gamma)$ to a given interface Γ. In this paper, we will focus on the problem of minimising a certain geodesic length in two dimensions and write G for the cost in the geodesic case. The idea is then to compute the variation of the cost for deformations of the interface and to use gradient descent in order to minimise the cost.

Level set-methods [11] introduce a level-set function[1] ϕ to represent the interface Γ implicitly as zero level-set:

$$\Gamma := \phi^{-1}(0) \tag{1.1}$$

The implicit representation links ϕ (as the introduced *analytic entity*) with the *geometric entity* Γ and allows for changes in the topology during the evolution. Furthermore, this relationship can be made one to one by imposing the signed distance constraint [7, 12].

By the implicit description, Γ is determined by ϕ and the cost can be viewed as functional $C(\phi)$. Introducing an evolution parameter t, we write $u(x,t) = \phi(x)$ for the level-set function at 'time' t. In this way, the gradient descent equation for $u(x,t)$ becomes a *partial differential equation* (PDE) that controls the velocity of the current interface with the objective to minimise C:

$$\frac{du}{dt} = -\frac{\delta C}{\delta u} \tag{1.2}$$

The numerical realisation of the evolution has been discussed intensively [11, 7, 9, 12] and we adopt here a novel representation in [12] which uses a sparse dynamic *finite element complex* [15] to represent and evolve the level-set function. Using the signed distance constraint, one has the following simple expressions for the key geometric quantities (normal $N \in S^1$ and curvature κ):

$$N = \nabla u \tag{1.3}$$

$$\kappa = \nabla^2 u \tag{1.4}$$

In this paper we focus on the somewhat less discussed issues of *initialising* the evolution and *terminating* the evolution automatically. The non-parametric nature of level-set representations removes the need for manual initialisation that is required by parametric methods (e.g. B-spline snakes [3]). Using the geodesic contour detection as an example, we discuss various options for initialisation and evolution-control which are of practical use.

Section 2 is a brief review of the geodesic problem using the Riemannian length as cost function which we use

*Supported by the EPSRC, the Cambridge European Trust and the DAAD (Germany)

[1] ϕ is a continuous, real valued function

throughout the paper. Various options for initialising level-set functions are then introduced in Section 3. Section 4 on automatic termination completes the theoretical introduction. This is followed by sections on results (Section 5) and conclusions (Section 6).

2 Geodesic Active Contours

It has been known for some time [2, 8, 10] 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, one starts by introducing a *local* measure for edges g (with $g(x) \in [0;1]$). A simple example[2] is given by [8, 10]:

$$g := \frac{1}{1 + a\ |\nabla I_\sigma|^2} \tag{2.1}$$

where I_σ is the image smoothed with a Gaussian of scale parameter σ and a is a sensitivity constant to image contrast. Interpreting g as *Riemannian metric* for the image, we can see that we are using the Euclidean length measure in the absence of edges ($g \approx 1$) but use a smaller length-measure in areas of large gradient values.

The *global* cost associated with Γ is the *Riemannian length*:

$$G(\Gamma) = \int_\Gamma g \tag{2.2}$$

where the standard Lebesgue measure is used to integrate the function g over Γ.

The velocity function β which asymptotically minimises G can be shown to be [8, 10]:

$$\beta = \mathrm{div}\Big(g \tfrac{1}{|\nabla u|} \nabla u\Big) + cg \tag{2.3}$$

where c is the coefficient of the so-called "balloon force" [8, 10]. We derive (2.3) in Appendix A for the signed distance case:

$$\beta = \langle \nabla g, \nabla u \rangle + g\ \big(\nabla^2 u + c\big) \tag{2.4}$$

and refer to [8] for the connection to parametric snakes that has been discussed in the literature. The numerical implementation of (2.4) for the finite element representation has been recently introduced in [12].

The "balloon force" term does not arise from the cost minimisation but is useful in certain situations:

- It can be used to accelerate the convergence since it adds a velocity c to the interface motion in the absence of any edges ($g \approx 1$) and vanishes at ideal edges ($g \approx 0$).
- There are applications where the cost functional is locally flat (e.g. sky-line initialisation of Figures 5 and 6). Here the "balloon force" is vital to drive the evolution towards the desired minimum (and can be turned off subsequently to refine the contour).
- The formulation as Riemannian length minimisation problem implies that the contour localisation deviates from the 'true edge location' and lies slightly inside convex areas and outside concave areas. The balloon force can be used to adjust the location of the contour, provided the object's curvature is known approximately. In fact this is one of the original motivations for adding the term [4, 6].

[2] Further examples are known in the context of *diffusion filtering [13]*.

3 Initialisation of Level-Set Evolutions

The initialisation of level-set evolutions requires an initial level-set function. By demanding the signed distance constraint, one can equivalently prescribe an initial orientated closed curve (or collections of simple closed curves) that does not self-intersect. This uses the fact that implicit curves are in one-to-one correspondence to signed distance functions [7].

In this section we discuss two sources of initialisation: The first one embarks on *a priori* knowledge and can be used to initialise evolutions independent of any previous evolution. The second initialisation is indirect in that it relies on previous evolutions. The termination criteria of Section 4 is of particular relevance to the indirect case (Section 3.2) where the end of one evolution starts a secondary evolution.

3.1 Direct Initialisation

In the most important example, the initial curve consist of the outline of an image and reflects the assumption that the objects of interest are fully contained in the image. However, we discuss more general options and demonstrate in particular the case of a single infinite line that allows us to capture non-closed contours such as sky-lines.

A rich set of signed distance function initialisations can be obtained by transforming and combining two very basic distance functions, the ones corresponding to a point and a line.

3.1.1 Elementary and Transformed Shapes

We define the signed distance functions of a point and a line:

$$\phi_{\text{point}}(x) := |x| \tag{3.1}$$
$$\phi_{\text{line}}(x) := \langle N, x \rangle \tag{3.2}$$

where $N \in S^1$ denotes the 'outward' normal of the region separated by the line. More interesting examples can be formed by transforming the space ($\mathbb{R}^2$) and the levels:

- **Spatial Transformations:** Signed distance functions are symmetric under the Special Euclidean Group $\mathrm{SE}(\mathbb{R}^2)$ and scaling: If ϕ is a signed distance map,
  - $\psi(x) := \phi(x - T)$ is the signed distance map corresponding to a translation $T \in \mathbb{R}^2$ of the shape.

- $\psi(x) := \phi(R^t x)$ is the signed distance map corresponding to a rotation (with fixed origin) by $R \in SO(2)$ of the shape.
- $\psi(x) := s\,\phi(\frac{1}{s}x)$ is the signed distance map corresponding to a scaling of the shape by a factor $s \in \mathbb{R}$.

- **Level Transformations:** Signed distance functions are symmetric under shift and inversion of levels: If ϕ is a signed distance map,
  - $\psi(x) := \phi(x) - d$ is the signed distance map corresponding to an expansion of the shape's interior by a distance d in normal direction.
  - $\psi(x) := -\phi(x)$ is the signed distance map corresponding to the same shape but with an inversion of inside and outside (complement).

By applying this transformations to the point and line, one obtains general (infinite) lines and general circles. The use of a single line as initial shape is illustrated in Figures 5 and 6.

3.1.2 Combining Shapes

Implicit shapes can be combined (union and intersection) by corresponding min/max operations on the distance functions. Applying the transformations of the previous section to combined shapes, one can generate a manifold of further initialisation shapes.

If ϕ_1 and ϕ_2 are signed distance maps,

- **Union:** $\phi(x) := \min(\phi_1(x), \phi_2(x))$ corresponds to the union of the interiors described by ϕ_1 and ϕ_2.
- **Intersection:** $\phi(x) := \max(\phi_1(x), \phi_2(x))$ corresponds to the intersection of the interiors described by ϕ_1 and ϕ_2.

When combined with the level-inversion, this means that we can implicitly perform all usual operations that are known from set-theory on our shapes (complement, subtraction, union, intersection).

In particular, we can now easily obtain signed distance functions of arbitrary closed polygons. For instance, we can obtain the outline of an image by intersecting the four lines of the image-borders (with normals pointing away from the image region).

3.2 Indirect Initialisation

Indirect initialisations are useful when an evolution is to be based on the result of a previous related - but different - evolution. Examples of this case are given by:

- **Detection of all local minima:** A general problem of the minimisation problem lies in the fact that a single evolution will only detect one of the possibly many local minima. However, we can follow a scheme that was suggested in [12] for nested contour detection: The general strategy is use the knowledge of a detected local minimum to drive the subsequent evolution into a different local minimum. The idea is to use two different Riemannian metrics as illustrated in Figure 1: The initial interface Γ_0 is first attracted to the local minima Γ_1 of $G = \int_\Gamma g$. Subsequently, the evolution with a modified cost $\bar{G} = \int_\Gamma \bar{g}$ is used to move the interface beyond the already detected minima. Where the detected interface Γ_1 is used to define $\bar{g}$:

$$\bar{g}(x) = \begin{cases} 1 - g(x) & \text{if } \phi_1(x) < 0 \\ 1 & \text{else} \end{cases} \tag{3.3}$$

- **Global distance to contour map:** The numerical representation is, for computational efficiency, restricted to the vicinity of the interface Γ. However, once the interface is located, one can use a subsequent evolution with constant normal speed ($\beta = 1$) to obtain the global continuation of the signed distance map (see Figure 7).
- **Multi-resolution techniques:** In the finite element representation used, it is natural to consider refinements of elements (Figure 2). Using initially a coarser resolution speeds up the convergence which is relevant for time critical applications (e.g. tracking). In this paper we restrict ourselves to consider techniques that change the resolution for the entire domain. More sophisticated adaptive methods are also possible where one can use the curvature κ to drive a sub-division process.

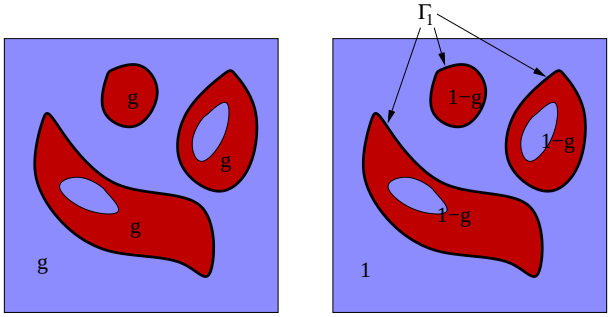

Figure 1: **Modification of the Riemannian Metric for the Detection of General Local Minima:** The original metric g (left) leads initially to the detection of contour Γ_1 and is replaced temporarily by $\bar{g}$ (right) to move the interface away from the already detected minimum.

4 Evolution Control and Termination of Level-Set Evolutions

We analyse the convergence behaviour of level-set evolutions in the case where the evolution is derived from a cost functional C. For instance, as was pointed out in Section 2, the Riemannian length G is the cost for the geodesic contour detection.

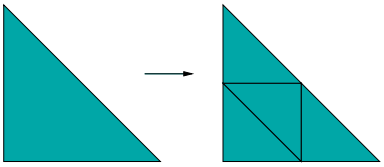

Figure 2: **Element Sub-division for Resolution Refinement:** The original element (left) is sub-divided into four triangles to double the resolution (right). The level-set function of the sub-divided elements is initialised to agree with the original function. This is possible because any coarse scale polynomial yields a polynomial of the same degree in the refined elements.

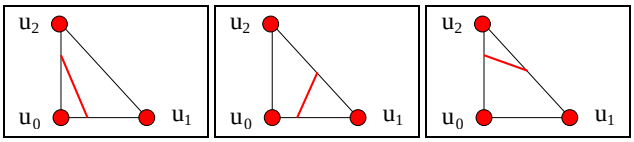

Figure 3: **Element Nodes:** Three relevant examples of active elements. Each element of the complex is a standard simplex and u_0, u_1, u_2 denote the node values of the function u. The zero level-set of u is indicated (red line) for each example. For the element on the left, the sign of u_k with $k = 0$ differs from the signs of the other node values. Similarly, for the elements in the middle $k = 1$ and on the right $k = 2$.

We assume in the following, that one can evaluate the cost functional and its derivative during the evolution.[3] We show in Section 4.1 how to compute the Riemannian length numerically and will use this information to terminate the gradient descent (1.2).

We first investigate the differential changes in the cost induced by the evolution. From Appendix A, (A.6) we obtain the following result for the geodesic case: for small deformations v of the level-set function u, the changes in cost G are given by:

$$G(u+v) - G(u) \approx -\int_\Gamma \beta\, v \tag{4.1}$$

where we have written $\Gamma(u) = u^{-1}(0)$ for the zero level-set. In particular, pursuing the steepest descent $v = \Delta t\, \beta$ (with time step Δt) results in

$$\Delta G \approx -\Delta t \int_\Gamma \beta^2 = -\Delta t\, ||\beta||^2_{\mathcal{L}^2(\Gamma)} \tag{4.2}$$

or

$$\tfrac{dG}{dt} = -||\beta||^2_{\mathcal{L}^2(\Gamma)} \tag{4.3}$$

4.1 Measuring Zero Level-Sets: Length in Euclidean and Riemannian Sense

The aim of this section is to compute the length of a zero level-set (or individual loop-components of it). The novel finite element representation [12] proves useful by defining the level-set unambiguously inside each active element. For simplicity, we resrict the explicit computations of this section to the case of first order elements where we can perform all necessary computations without approximation.

The Riemannian length can be expressed directly in terms of the node-values and the result for individual elements can be summarised as follows (proof omitted here). Using the notation of [12] (see Figure 3) $k \in \{0,1,2\}$ denotes the index such that the sign of u_k is different from the (equal) sign of the two other coefficients. We then obtain the following expressions for the length of the zero level-set in the element:

[3] The method can also be applied to evolutions where the cost is not explicitly known by using equation (4.3): The integral $\int_\Gamma \beta^2$ can be computed similar to the Riemannian length integral for 1st order elements.

$$L_1(U) = \frac{u_k \sqrt{(u_1-u_0)^2 + (u_2-u_0)^2}}{\prod_{i\neq k}(u_i - u_k)} \tag{4.4}$$

$$L_g(U) = L_1(U) \sum_{i=0}^{2} {y^{(k)}}_i\, g_i \tag{4.5}$$

where L_1 denotes the Euclidean and L_g the Riemannian lengths and where we have used ${y^{(k)}}_i$ defined by

$$\begin{aligned} {y^{(k)}}_i &:= \frac{1}{2}\frac{u_k}{u_k - u_i} \quad \text{for } i \neq k \\ \sum_i {y^{(k)}}_i &= 1 \end{aligned} \tag{4.6}$$

4.2 Termination Condition

Evaluating the cost functional during the evolution enables us to define a function $c(t) := C(u(t))$ which is expected to become stationary when a local minimum is approached (Figure 4). Note that we have shown in Section 4.1 how the cost can be computed for 1st order elements without having to introduce any approximation. Theoretically, $c(t)$ decreases indefinitely and will not become stationary at finite time. However, due to the finite numerical accuracy, c(t) stays monotonous until the finite numerical accuracy leads to a slight fluctuation. We use this limited numerical accuracy to stop the evolution.

5 Results

Figures 5, 6, 7, 8 and 9 show experimental results obtained with the first order elements of [12].

The use of a direct initialisation from a single line (see Section 3.1) is demonstrated in Figures 5 and 6. Figure 7 demonstrates the use of a user-specified initialisation (here a circle) in order to obtain the desired contour in an image with a complex background.

Applications of indirect initialisations are shown in Figures 7, 8 and 9: in Figure 7 the global signed map is obtained by using the contour as initial curve for a constant speed evolution (see Section 3.2). Figure 8 demonstrates the detection

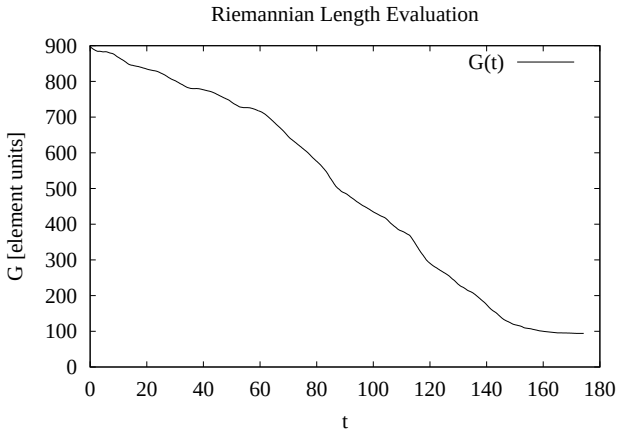

Figure 4: **Geodesic Length Evaluation:** The diagram shows the numerically computed values for the Riemannian length during (the first part of) the evolution from Figure 8.

of nested contours (see Section 3.2) and Figure 9 shows how an initially coarse contour can be used for a fast contour detection by initialising a refined evolution in the vicinity of the local minimum.

6 Conclusion and Future Work

We have presented methods to initialise and terminate level-set evolutions in 2D. Our investigation adds further practical advantages to the use of level-set methods in delivering tools towards user-input free methods.

We exploited the correspondence between *implicit curves* and *signed-distance constrained level-set functions* for the initialisation. Two types of sources for initial level-set functions were presented: While the direct initialisation is based on geometric shapes (e.g. polygons), the indirect initialisation uses signed-distance level-sets of previous evolutions as starting point. The use of indirect initialisations was demonstrated in the examples of global signed distance map computation, nested contour detection and multi-resolution techniques.

We introduced a numerical condition to terminate level-set evolutions when convergence to local minima is achieved. The paper demonstrates how the numerical method (adopted from [12]) allows for a direct computation of the Riemannian length which is used to determine the point of convergence.

Future work is concerned with

- Investigation of methods to automatically suggest values of the parameters of the metric (σ, a) from the image.
- Applications to more general metrics (such as the ones suggested in [6, 9]).
- 3-dimensional implementation and evolution control for geodesic implicit surface detection.

Figure 5: **Sky-line Example:** Initialisaton of a single line along the top of an image can be used to detect non-closed contours such as a 'sky-line'. The top part shows the initial curve and the bottom part the converged curve. A value of $c = 0.5$ was used for the 'balloon force'.

Figure 6: **Sky-line Example:** Contour detection with a single line along the top as initialisation.

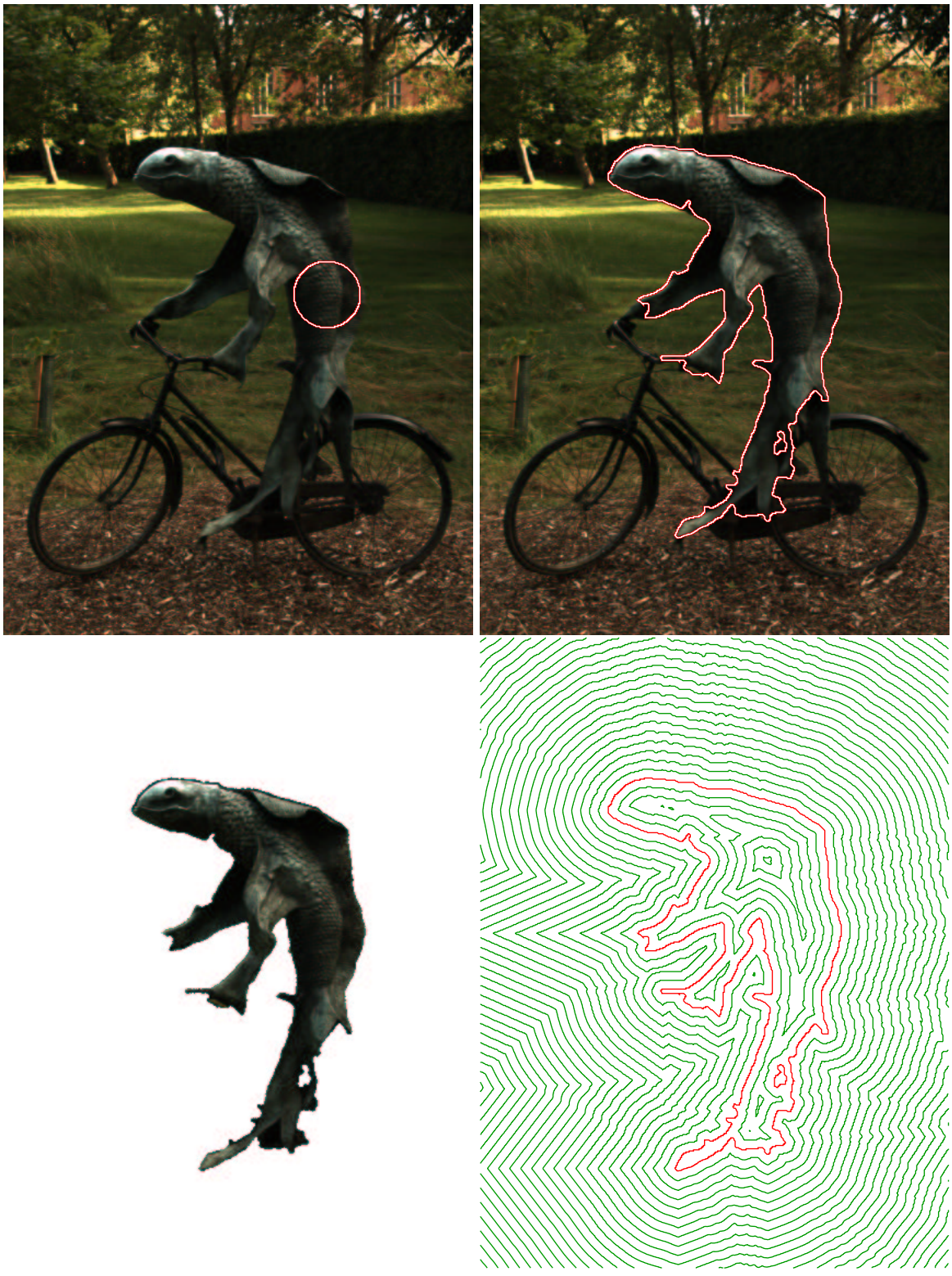

Figure 7: **Cycling Fish Sculpture:** The image on the top-left shows the original image with a user-selected circle as initial level-set. A negative 'inflating' balloon-force is used to drive the evolution towards normalised colour edges and the result of the evolution is displayed on the top-right. Applications based on the obtained contour are displayed in the bottom row of the figure: the bottom-left shows the implied segmentation (by drawing all pixels for which the level-set function is negative) and the bottom-right indicates the levels of the global signed distance map (with a spacing of 8 element units and the zero level-set coloured in red).

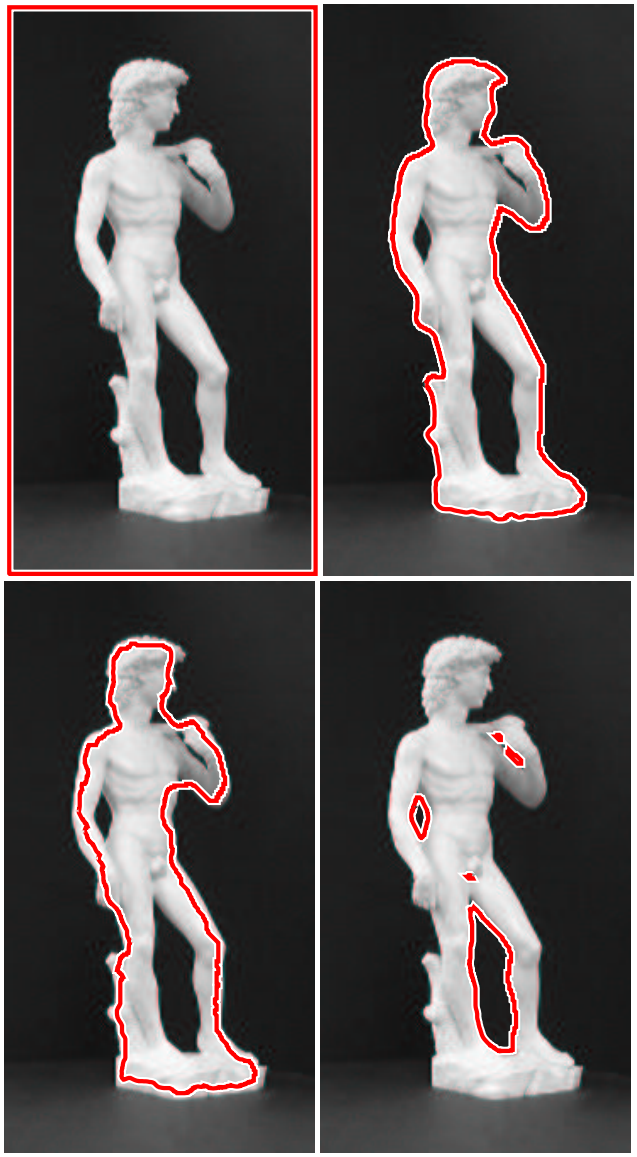

Figure 8: **Nested Contour Example:** The top-left image shows the trivial initialisation (image outline Γ_0). Detection of the first local minimum leads to the result (Γ_1) displayed in the top-right image. Temporary modification of the cost (see Figure 1) leads to the departure (Γ_2) from the already detected contour in the bottom-left image and finally, minimisation with the original cost functional leads to the detection of the nested contours (Γ_3) in the bottom-right image.

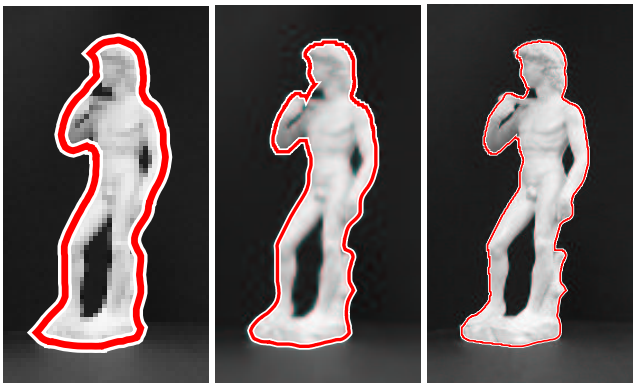

Figure 9: **Multi-Resolution Example:** The spatial resolution of the level-set evolution is increased by a factor of 2 in each of the following steps. The trivial initialisation (image outline) leads with a coarse resolution version of the algorithm to the contour displayed on the left. The approximative contour is then used to initialise a refined evolution by sub-dividing elements (see Figure 2) and the converged result (middle) is used in turn to initialise the final evolution with the final result shown on the right.

A Derivation of the Geodesic Evolution Equation

This section gives a direct poof that β of (2.4) does indeed correspond to the gradient descent of the Riemannian length in the case that $c = 0$. Use of the signed distance constraint allows us to simplify previous proofs in the literature [8].

Proposition A.1. *Geodesic Evolution PDE:*

Defining the Riemannian length for the zero level-set of u,

$$G(u) := \int_{u^{-1}(0)} g \tag{A.1}$$

we will prove that

$$\begin{aligned} \beta &:= \langle \nabla g, \nabla u \rangle + g\, \nabla^2 u \\ &= \operatorname{div}(g\, \nabla u) \end{aligned} \tag{A.2}$$

corresponds to the gradient descent which we use to minimise G.

Proof. Before we detail the proof, we would like to point out an intuitive geometric interpretation: a change in the level-set function implies a change in the interface and changes the length functional G. The two summands in (A.2) correspond to two independent causes for such a change. The first term ($\langle \nabla g, \nabla u \rangle$) accounts for the fact that as u changes, the evaluation of g has to be translated to the new position of the interface (i.e. moved in normal direction ∇u). Note that the first term vanishes in the case of a constant metric such as the Euclidean metric. The second term ($g\, \nabla^2 u$) corresponds to the change in length of the interface assuming that g remains

constant under the deformation (the first term accounts for the transport of g). The change in length is proportional to the curvature $\kappa = \nabla^2 u$ and can be understood intuitively by looking at a circular interface of radius r: While the translation of a line in normal direction does not change its length, the (proportional) length of a circle changes under normal motion proportional to $\kappa = \frac{1}{r}$.

For the proof, consider the change in the length functional G when the zero level-set of u is deformed in normal direction ($u \to u + v$). If $v : \mathbb{R}^2 \to \mathbb{R}, x \mapsto v(x)$ denotes the change in the level-set function, the deformation transforms an original level-set location x into $y = T(x)$, where

$$T : \mathbb{R}^2 \to \mathbb{R}^2, x \mapsto y = x - v(x)\ \nabla u(x) \qquad \text{(A.3)}$$

and where we have assumed signed distance representation (hence $|\nabla u| = 1$).

We will now assume that the zero level-set consists of a single loop so that we can use a simple parametrisation for the proof. The general case follows then by summation over individual loops and requires some additional thoughts in the case of topological changes which we omit here. Let $\Gamma := u^{-1}(0)$ be parametrised by its Euclidean arc-length on the interval $I = [0; L_1(u)]$. We then have

$$G(u) = \int_I |\tfrac{dx}{ds}(s)|\ g(x(s))\ ds \qquad \text{(A.4)}$$

For the change in Riemannian length, we can write[4]:

$$\begin{aligned}
&G(u+v) - G(u) \\
&= \int_I \Big[|\tfrac{dy}{ds}(s)|\ g(y(s)) - |\tfrac{dx}{ds}(s)|\ g(x(s)) \Big]\ ds \\
&\approx \int_I v\ \tfrac{\delta}{\delta v}|_{v=0}\ \big[|\tfrac{d}{ds}(x - v\nabla u(x))|\ g(x - v\nabla u(x)) \big]\ ds \\
&= -\int_I v(x)\ \big[g(x)\ \nabla^2 u(x) + \langle \nabla g(x), \nabla u(x) \rangle \big]\ ds \\
&= -\int_I v(x)\ \beta(x)\ ds \qquad \text{(A.5)}
\end{aligned}$$

here the second last step follows from the product rule together with $|\frac{dx}{ds}| = 1$ from

$$\tfrac{\delta}{\delta v}(g(x - v\nabla u)) = -\langle \nabla g, \nabla u \rangle$$

and

$$\tfrac{\delta}{\delta v}|_{v=0}\ \big[|\tfrac{d}{ds}(x - v\nabla u(x))| \big] = -\big\langle \tfrac{dx}{ds}, \tfrac{dN}{ds} \big\rangle = -\kappa = -\nabla^2 u$$

(Gauss map [1]) where we have used the fact that $N = \nabla u$ is the normal of the interface. (A.5) means that

$$\tfrac{\delta G}{\delta u} = -\beta \qquad \text{(A.6)}$$

and hence β is the steepest decent in the space of normal deformations (and β vanishes at local minima of G). □

[4] We have assumed that v is sufficiently small so that no self-intersections occur. This is justified since we are looking at the infinitesimal case here.

References

[1] M. Berger and B. Gostiaux. *Differential Geometry: Manifolds, Curves and Surfaces*. Springer, 1987.

[2] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 694–699, 1995.

[3] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 616–623, 1990.

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

[5] O. D. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *Proc. LNCS European Conf. on Computer Vision*, volume 1, pages 379–393, 1998.

[6] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. In *IEEE Tran. on Image Processing*, pages 10(10):1467–1475, 2001.

[7] J. Gomes and O.D. Faugeras. Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation*, 11(2):209–223, June 2000.

[8] R. Kimmel. *Curve Evolution on Surfaces*. PhD thesis, Dept. of Electrical Engineering, Technion, Israel, 1995.

[9] N. Paragios. *Geodesic Active Regions and Level Set Methods: Contributions and Applications in Artificial Vision*. PhD thesis, INRIA Sophia Antipolis, France, January 2000.

[10] G. Sapiro. *Geometric Partial Differential Equations and Image Processing*. Cambridge University Press, 2001.

[11] J.A. Sethian. *Level Set Methods*. Cambridge University Press, Cambridge, 1999.

[12] M. Weber, A. Blake, and R. Cipolla. An efficient and stable level-set method for active contours. CUED/F-INFENG/TR 452, Department of Engineering, University of Cambridge, March 2003.

[13] J. Weickert, B.M. ter Haar Romeny, and M.A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Processing*, 7(3):398–410, March 1998.

[14] A.J. Yezzi and S. Soatto. Stereoscopic segmentation. In *Proc. IEEE Int. Conf. on Computer Vision*, pages I: 56–66, 2001.

[15] O.C. Zienkiewicz and K. Morgan. *Finite Elements & Approximation*. John Wiley & Sons, NY, 1983.