Smoothness–Based Forces for Deformable Models: A Long Range Force and a Corner Fitting Force

Zixin Zhang[1] Michael Braun[*]

Department of Applied Physics, University of Technology, Sydney, P.O.Box 123, Broadway, NSW 2007, Australia

Abstract

Deformable models, originally proposed by Terzopoulos *et al* and Kass *et al* in 1988, have been widely used in medical image segmentation. However, they manifest two well known limitations: the lack of an appropriate long range force to drive the model surface towards the object boundary and poor performance at high curvature boundaries (such as corners) due to the models' intrinsic smoothness constraint. In this paper, a new smoothness force with local control is proposed. The local control is used to devise a long range force, referred to as the self–zoom force, and a corner fitting force. The self–zoom force enables the model surface to expand and shrink without a limit in range. The corner fitting force propels the model surface to fit high curvature boundaries. Experiments demonstrate that the model surface is driven to the object boundary by the new forces even if the initial estimate is not close and the object is nonconvex or has a high local curvature.

Key words: deformable models, active contour models (snakes), image segmentation, corner fitting, medical imaging, image visualization

1 Introduction

Deformable models, originally proposed by Terzopoulos *et al* [1,2] and Kass *et al* [3] in 1988, have been used widely in medical image segmentation. However, they manifest certain well known limitations that significantly restrict

[*] Corresponding author.
Email addresses: zzhang@proteomesystems.com (ZZ), Michael.Braun@uts.edu.au (MB).

[1] Present address: Proteome Systems Ltd., 35–41 Waterloo Rd, North Ryde, NSW 2113, Australia

the scope of their applications. One limitation is the lack of appropriate long range forces [4]. The commonly used gradient force operates over a short range. Away from the object boundary, the model's surface may experience little or no attraction towards the boundary. In noisy medical images, this can lead to a segmentation failure and is one of the reasons why, in most implementations, the initial model surface has to be placed close to the object boundary. Several long range forces have been introduced to overcome the limitation. The elastic spring force [3,5–8] pulls model nodes towards object boundary (high gradient) voxels. Its strength depends on the distance from the model node to its corresponding boundary voxel, vanishing when that distance becomes zero. Model nodes have to be reconnected to appropriate boundary voxels at each iteration of model evolution. However, determining the correspondence between model nodes and the boundary voxels is not trivial and can be computationally expensive [6]. Another long range force is the *balloon* force [9–11]. Its direction is normal to the model surface at each node, and its strength is usually preset to a constant value by the user. The balloon force inflates or deflates the model surface without a limit in range. However, difficulties may arise if the object boundary is locally concave. Moreover, setting an optimum strength of the balloon force may not be simple as the optimum is image dependent. If set too high, the model will overshoot the object boundary, whereas if set too low, the model may become trapped by image noise and irrelevant boundaries. Recently, a new long range force, based on *gradient vector flow* (GVF), was introduced [12,13]. This force extends the range of the gradient force and attracts the model towards high gradient points. However, it too may be confounded by noise–induced gradients.

Another limitation arises from the fact that many biological organs manifest complex borders with high-curvature components. The intrinsic smoothness constraint of deformable models prevents them from segmenting such objects accurately. This difficulty has been addressed previously for 2D deformable contour models. Menet *et al* [14] segmented contour corners by B–splines in two steps: 1) the B–spline was allowed to evolve and reach its convergence state; 2) the potential corners were detected as extrema of contour curvature and the smoothness constraint at the candidate corner nodes was relaxed by setting multiple nodes at each candidate, then the B–spline was allowed to continue evolving. Xu *et al* [15] imposed on each candidate corner node an external force which varied with contour curvature and cancelled out the local smoothness force that prevented the deformable contour from reaching corners. Both algorithms work by relaxing the smoothness force at candidate corner nodes to allow the contour to bend at those nodes. However, as neither algorithm provides a force to drive the nodes into the corners, the corners will not, in general, be fully defined.

In this paper, a locally controlled smoothness force is proposed. With the use of the local control, a long range force is constructed, to be referred to as the self–zoom force. It enables the model surface to expand and shrink towards the object boundary without a limit in range. Also based on the locally controlled smoothness force, a corner fitting force is devised, which allows the model to segment objects with high curvature boundary. The rest of the paper is organized as follows. The notation for deformable models is introduced briefly in section 2. In section 3, the locally controlled smoothness force is proposed. The self–zoom force is described in section 4 and the corner fitting force is described in section 5. Section 6 gives the conclusions. An early version of the self–zoom force and the corner fitting force were reported in [16] and [17], respectively.

2 Deformable models

A deformable model is a surface model with a generalized spline subject to internal and external constraints expressed by energies or forces [1,3]. Image segmentation is carried out by fitting the deformable model to image data under the influence of the model's internal and external constraints. The energy functional is expressed [1,18] as

$$E(\vec{q}) = \int_0^1\!\!\int_0^1 \left[w_{\text{int}} E_{\text{int}}(\vec{q}) + w_{\text{ext}} E_{\text{ext}}(\vec{q})\right] \,\mathrm{d}u\,\mathrm{d}v, \tag{1}$$

where E_{int} and E_{ext} are the internal and external energies, respectively, with the corresponding weights w_{int} and w_{ext}, and the model surface $\vec{q}(u,v) = [x(u,v), y(u,v), z(u,v)]^T$ is parameterized by $u, v \in [0,1]$. Ideally, the minimum global energy corresponds to the object boundary.

The internal energy represents prior knowledge about the object. It gives the model a bias towards certain boundary properties. In medical images, object boundaries are commonly presumed to be continuous and smooth. Correspondingly, the internal energy is expressed in terms of first and second order derivatives [1,18],

$$E_{\text{int}}(\vec{q}) = w_{10}\left|\frac{\partial \vec{q}}{\partial u}\right|^2 + w_{01}\left|\frac{\partial \vec{q}}{\partial v}\right|^2 + w_{20}\left|\frac{\partial^2 \vec{q}}{\partial u^2}\right|^2 + w_{02}\left|\frac{\partial^2 \vec{q}}{\partial v^2}\right|^2 + w_{11}\left|\frac{\partial^2 \vec{q}}{\partial u \partial v}\right|^2 . (2)$$

The external energy is usually designed to attract the model to image features, such as high intensity or high gradient voxels [3].

3 Locally controlled smoothness force

In conventional deformable models, the smoothness force cannot inflate or deflate the model surface globally and it prevents the model from fitting high curvature boundaries accurately. It lacks local control though its strength, relative to the continuity force and the external force, can be controlled globally by presetting the weights in equation 2. By local control, we mean control of the strength and direction of the smoothness force at individual nodes. In this section, a locally controlled smoothness force is proposed, which engenders a long range force and a corner fitting force. We will also describe the surface representation, the formulation of smoothness energy, the decomposition of the smoothness force into components and the control of the smoothness force components.

3.1 Surface representation

We represent a closed 3D surface for deformable models by means of a conforming triangular mesh with planar mesh elements (fig. 1). Each edge of the mesh is shared by exactly two triangular mesh elements. The mesh nodes correspond to the vertices of the triangular mesh. Two nodes are *adjacent* if they share an edge, i.e. if they are vertices of a common triangle, and triangles are adjacent if they share a node. The number of nodes adjacent to any one node varies from one node to another, allowing the mesh to deform locally. The total number of model nodes can vary from one iteration to another during the model evolution, allowing the mesh to expand and shrink substantially. These variabilities over space and time give the mesh additional flexibility.

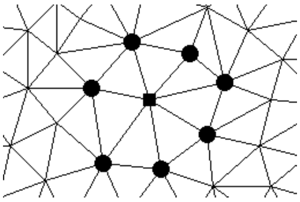


Fig. 1. A conforming triangular mesh. A node (the solid square in the centre) is shown connected to its adjacent nodes (the solid circles), and is shared by its adjacent triangles.

3.2 Local surface smoothness and smoothness energy

Principal curvatures or second derivatives of a surface in two orthogonal directions, which are commonly used to represent smoothness on a continuous surface, are not appropriate for a mesh surface with planar mesh elements.

In the conforming triangular mesh, the local surface smoothness at a node j should account for the orientation of each adjacent mesh element relative to the plane T tangential to the surface at node j. We adopt the notation of [16]. Similar notation was used in [19]. The relative orientation is measured by the smoothness angle ϕ_i subtended by the normal $\vec{N}$ to the surface at node j and the unit normal $\hat{n}_i$ to the adjacent mesh element i (fig. 2)

$$\phi_i = \cos^{-1}\left(\hat{n}_i \cdot \frac{\vec{N}}{|\vec{N}|}\right), \tag{3}$$

where

$$\hat{n}_i = \frac{\vec{e}_i \times \vec{e}_{i+1}}{|\vec{e}_i \times \vec{e}_{i+1}|},$$

$$\vec{N} = \frac{1}{\theta}\sum_{i=1}^{I_j}\theta_i\,\hat{n}_i, \qquad \theta = \sum_{i=1}^{I_j}\theta_i.$$

The smoothness angle $\phi \in [0, \pi/2]$ will be larger at surface parts with higher curvature. The inner angle θ_i is subtended by the edge vectors $\vec{e}_i$ and $\vec{e}_{i+1}$, which connect node j to its adjacent nodes i and $i+1$, respectively. The triangular mesh element i will be parallel to the tangential plane T at node j when the smoothness angle $\phi_i = 0$. The local surface at node j will be planar if all smoothness angles ϕ_i, $i = 1, \ldots, I_j$, are zero.

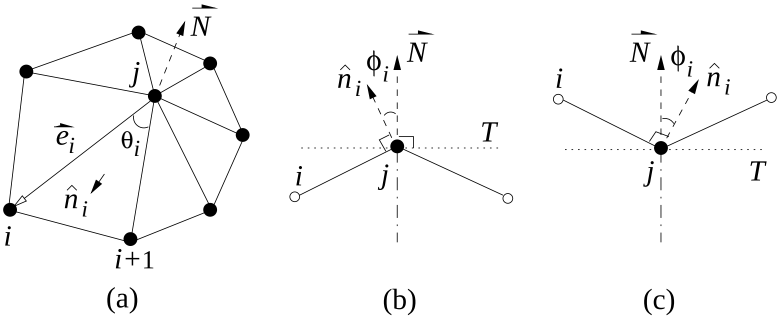


Fig. 2. Definition of the smoothness angle ϕ (see text for notation). (a) Top view of a part of a triangular mesh consisting of node j, its ($I_j = 7$) adjacent nodes and triangular mesh elements. Smoothness angle ϕ_i subtended by the surface normal $\vec{N}$ and the mesh element normal $\hat{n}_i$ for a convex (b) and for a concave (c) surface.

The smoothness energy E_j^{s} at node j is determined by integrating the orientation information of all mesh elements i sharing node j. It is defined as the sum of squared smoothness angles ϕ_i, weighted by the corresponding inner angles θ_i and normalized by the sum θ of all the inner angles θ_i (eq. 3)

$$E_j^{\mathrm{s}} = \frac{1}{\theta}\sum_{i=1}^{I_j}\theta_i\,{\phi_i}^2. \tag{4}$$

Weights θ_i guarantee that the smoothness energy E^{s}_j will not change when any adjacent mesh element merely splits. Minimum E^{s}_j will occur where each ϕ_i is zero, making the surface at node j planar. The sum of E^{s}_j over all N nodes gives the total smoothness energy of the model, expressed as

$$E^{\mathrm{s}} = \sum_{j=1}^{N} E^{\mathrm{s}}_j. \tag{5}$$

3.3 Components of the smoothness force

The measurement of the smoothness energy is local: at any node, it is determined only by that node and nodes adjacent to it (eq. 4). The movement of a node will change the smoothness energy at that node and at its adjacent nodes, but not elsewhere. Let the node's movement during an iteration be confined to a region around the node to be referred to as the search neighborhood. Suppose node j moves from its current position to a new position in the search neighborhood. The total change in the smoothness energy ΔE^{s} caused by the movement of node j is the sum of the changes in the smoothness energies at node j and at all its adjacent nodes. Let ΔE^{s}_i be the change in smoothness energy at adjacent node i, where E^{s}_i is determined by equation 4. Thus ΔE^{s} is expressed as

$$\Delta E^{\mathrm{s}} = \sum_{i=0}^{I_j} \Delta E^{\mathrm{s}}_i \ , \tag{6}$$

where $i = 0$ refers to node j itself. Suppose that the smoothness energy alone drives the evolution of the surface. Node j will move from its current position to a new position where the total energy change ΔE^{s} is most negative, or will stay at its current position if $\Delta E^{\mathrm{s}} \geq 0$ at every voxel position within the search neighborhood.

Alternatively, the movement of node j can be described in terms of the smoothness force $\vec{F}$ acting on node j, expressed as the gradient of the smoothness energy E^{s},

$$\vec{F} = \nabla E^{\mathrm{s}}. \tag{7}$$

By equation 5, the force can comprise contributions from all mesh nodes. However, the local constraint on the smoothness energy changes (eq. 6), restricts contributions to those arising at the node j (the node that has moved) and nodes adjacent to it. Using the indexing notation adopted earlier, we have

$$\begin{aligned}\vec{F} &= \sum_{i=0}^{I_j} \nabla E_i^{\mathrm{s}} \\ &= \sum_{i=0}^{I_j} \vec{F}_i. \end{aligned} \tag{8}$$

Each component $\vec{F}_i$ tends to move node j to a position that minimizes the smoothness energy at node i. In general, each force $\vec{F}_i$ tends to move node j to a different position. The destination of node j is determined by the vector sum of all the components $\vec{F}_i$.

3.4 Control of the smoothness force components

The smoothness force $\vec{F}$ in conventional deformable models depends on local surface smoothness only, and it is not user–controlled as all the smoothness force components $\vec{F}_i$ are assigned a constant weight (eq. 8). In our model, $\vec{F}$ is made controlable by assigning a variable weight w_i^{s} to each of the smoothness force components $\vec{F}_i$. Equation 8 becomes

$$\vec{F} = \sum_{i=0}^{I_j} w_i^{\mathrm{s}} \, \vec{F}_i, \tag{9}$$

where, again, $i = 0$ refers to node j itself. The direction and strength of $\vec{F}$, and thus the movement of node j, can be controlled by tuning the weights w_i^{s}.

Let us first investigate the effect of each force component $\vec{F}_i$ and the net force $\vec{F}$. We consider three mesh node configurations. For simplicity, these are shown in 2D (fig. 3), but the results readily apply to a 3D mesh. Smoothness component forces $\vec{F}_0$, $\vec{F}_1$ and $\vec{F}_2$, acting on node j, are contributed by nodes j, 1 and 2, respectively. The force component $\vec{F}_2$, for example, tends to smooth the local surface at node 2 by moving node j to a position in the search neighborhood such that the local smoothness energy at node 2 will be minimum. In the configurations (a) and (c), $\vec{F}_2$ tends to move node j outwards (upwards), whereas in the configuration (b), $\vec{F}_2$ tends to move node j inwards (downwards). The net force $\vec{F}$ acting on node j is

$$\vec{F} = w_0^{\mathrm{s}} \vec{F}_0 + w_1^{\mathrm{s}} \vec{F}_1 + w_2^{\mathrm{s}} \vec{F}_2. \tag{10}$$

Let us consider the case of $w_i^{\mathrm{s}} > 0$ $(i = 0, \ldots, I_j)$. In figure 3a, the net force $\vec{F}$ will push node j outwards as all the force components $\vec{F}_0$, $\vec{F}_1$ and $\vec{F}_2$ are directed outwards. In figure 3b, the force component $\vec{F}_1$ pushes node j outwards, whereas $\vec{F}_0$ and $\vec{F}_2$ pull it inwards. In figure 3c, $\vec{F}_1$ and $\vec{F}_2$ push node j

outwards, whereas $\vec{F}_0$ pulls it inwards. In figures 3b and 3c, some of the force components pull node j inwards, whereas others push it outwards. Whether node j moves inwards or outwards will be determined by the net force $\vec{F}$, which is in turn controlled by the weights w_i^{s}.

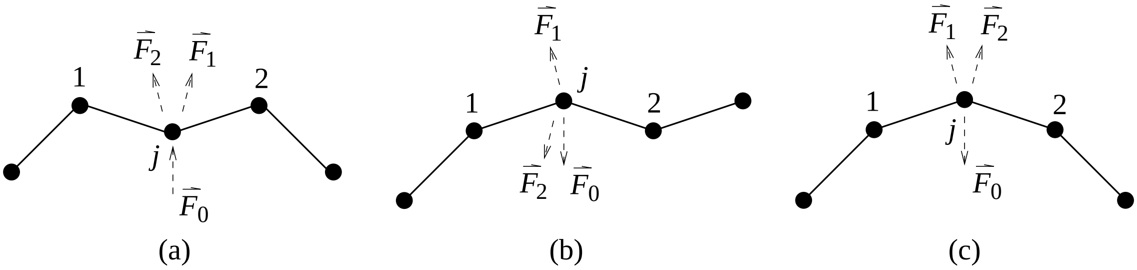


Fig. 3. Three node configurations (shown in 2D) and the corresponding smoothness force components. force components $\vec{F}_0$, $\vec{F}_1$ and $\vec{F}_2$, acting on node j, are contributed by nodes j, 1 and 2, respectively.

Let us examine closely how the weights w_i^{s} control the net force $\vec{F}$. We consider further the configuration of figure 3c, which occurs most often during surface evolution. For the sake of simplicity, we set $w_i^{\mathrm{s}}=1$ $(i=1,\ldots,I_j)$ and vary w_0^{s} only. Increasing $w_0^{\mathrm{s}} > 0$ will strengthen the force component $w_0^{\mathrm{s}}\vec{F}_0$ pulling node j inwards, whereas decreasing w_0^{s} will weaken it. There exists a threshold $\tilde{w}_j^{\mathrm{s}}$ such that if w_0^{s} exceeds it, the net force $\vec{F}$ will act to pull node j inwards. On the other hand, if w_0^{s} decreases below $\tilde{w}_j^{\mathrm{s}}$, $\vec{F}$ will act to push node j outwards. If $w_0^{\mathrm{s}} = \tilde{w}_j^{\mathrm{s}}$, $\vec{F}$ will be zero and node j will be stationary. Threshold $\tilde{w}_j^{\mathrm{s}}$ is local and varies from node to node, depending on local surface smoothness. Globally, the threshold values are bounded by

$$\tilde{w}^{\mathrm{s}}_{\max} = \max_{j\in\mathcal{N}}\{\tilde{w}_j^{\mathrm{s}}\} \qquad \text{and} \qquad \tilde{w}^{\mathrm{s}}_{\min} = \min_{j\in\mathcal{N}}\{\tilde{w}_j^{\mathrm{s}}\}, \tag{11}$$

where $\mathcal{N}$ is the set of all nodes in the mesh. Suppose w_0^{s} is set to a uniform value for all mesh nodes. The model will inflate at every node if $w_0^{\mathrm{s}}<\tilde{w}^{\mathrm{s}}_{\min}$ and deflate if $w_0^{\mathrm{s}}>\tilde{w}^{\mathrm{s}}_{\max}$. If $\tilde{w}^{\mathrm{s}}_{\min}<w_0^{\mathrm{s}}<\tilde{w}^{\mathrm{s}}_{\max}$, the model will inflate at nodes where $w_0^{\mathrm{s}}<\tilde{w}_j^{\mathrm{s}}$ and deflate where $w_0^{\mathrm{s}}>\tilde{w}_j^{\mathrm{s}}$.

Consider a spherical model surface with uniformly distributed mesh nodes. The smoothness angle will be the same at all nodes, and so will the threshold $\tilde{w}_j^{\mathrm{s}}$. We refer to the uniform value of $\tilde{w}_j^{\mathrm{s}}$ as the *global threshold* $\tilde{w}_{\mathrm{g}}^{\mathrm{s}}$. The thresholds $\tilde{w}^{\mathrm{s}}_{\max}$ and $\tilde{w}^{\mathrm{s}}_{\min}$ will be identical and equal to $\tilde{w}_{\mathrm{g}}^{\mathrm{s}}$. The model will experience global inflation when $w_0^{\mathrm{s}} < \tilde{w}_{\mathrm{g}}^{\mathrm{s}}$, global deflation when $w_0^{\mathrm{s}} > \tilde{w}_{\mathrm{g}}^{\mathrm{s}}$ and no change when $w_0^{\mathrm{s}}=\tilde{w}_{\mathrm{g}}^{\mathrm{s}}$. The global threshold $\tilde{w}_{\mathrm{g}}^{\mathrm{s}}$ will equal the uniform value of w_i^{s} $(i = 1,\ldots,I_j)$. When all the weights w_i^{s} $(i = 0,\ldots,I_j)$ are equal, the mesh node configurations (a) and (b) in figure 3 will transit to (c) under the influence of the smoothness force alone.

4 The self–zoom force

Having introduced the locally controlled smoothness force, we will describe its manifestation as the self–zoom force. The adaptability of the self–zoom will also be discussed.

4.1 The direction of the self–zoom force

Consider again the spherical model surface with uniformly distributed mesh nodes. By setting weights $w_i^{\mathrm{s}}=1$ $(i=0,\ldots,I_j)$, the net force $\vec{F}$ in equation 8 will be reduced to that in equation 8 which is the implicit setting for the smoothness force in conventional deformable models. At each node of the sphere, force components $\vec{F_i}$ reach an equilibrium, the net force $\vec{F}$ is zero, and the model energy reaches its global minimum. No node movement will take place and the spherical surface will experience neither global nor local change. If we set $w_i^{\mathrm{s}}=1$ $(i=1,\ldots,I_j)$ and allow w_0^{s} to vary, the net force $\vec{F}$ in equation 8 will become

$$\begin{aligned}\vec{F} &= w_0^{\mathrm{s}}\vec{F_0}+\sum_{i=1}^{I_j}\vec{F_i}\\ &= (w_0^{\mathrm{s}}-1)\vec{F_0}+\sum_{i=0}^{I_j}\vec{F_i}.\end{aligned} \tag{12}$$

The net force $\vec{F}$ in equation 12 is made up of the conventional smoothness force of equation 8 and an extra term,

$$\vec{F_{\mathrm{f}}}=(w_0^{\mathrm{s}}-1)\vec{F_0}. \tag{13}$$

The equilibrium of the smoothness force components at each node of the sphere will be broken if $w_0^{\mathrm{s}}\neq 1$. When $w_0^{\mathrm{s}}>1$, $\vec{F_{\mathrm{f}}}$ will have the same direction as the force component $\vec{F_0}$ (fig. 4a), pulling every node inwards, causing surface deflation. On the other hand, if $w_0^{\mathrm{s}}<1$, $\vec{F_{\mathrm{f}}}$ will have the opposite direction to the force component $\vec{F_0}$ (fig. 4b), pushing every node outwards. When no other forces are applied, the inflation will continue indefinitely, whereas deflation will proceed until all nodes converge to a point (in practice, the limit in deflation will be set by a minimum inter-node separation).

In a nonspherical surface, when $w_0^{\mathrm{s}}>1$, the self–zoom force $\vec{F_{\mathrm{f}}}$ tends to deflate the model surface where it is locally convex (fig. 4a), and to inflate it where it is locally concave (fig. 4c). It reinforces the force component $\vec{F_0}$ which always

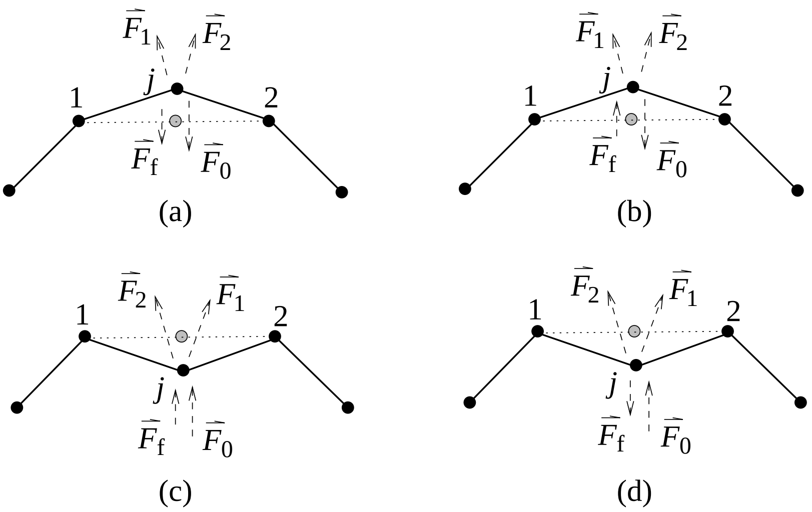


Fig. 4. The self–zoom force $\vec{F}_\mathrm{f}$ (shown in 2D). The force $\vec{F}_\mathrm{f}$ has the same direction as the force component $\vec{F}_0$ when $w_0^\mathrm{s} > 1$ (a, c), and the opposite direction to $\vec{F}_0$ when $w_0^\mathrm{s} < 1$ (b, d). The lightly shaded circle shows the position $\vec{F}_0$ tends to move node j to.

tends to move node j to the rest position (the shaded circle shown in fig. 4) where $\phi_i = 0$, $i = 1, \ldots, I_j$ (eq. 3). On the other hand, when $w_0^\mathrm{s} < 1$, the force $\vec{F}_\mathrm{f}$ opposes $\vec{F}_0$. It tends to inflate the model surface where it is locally convex (fig. 4b), and to deflate it where it is locally concave (fig. 4d). The local direction of the self–zoom force $\vec{F}_\mathrm{f}$ adaptively responds to the local surface shape (convex or concave). Note that the direction of a node's displacement may not coincide with the self–zoom force $\vec{F}_\mathrm{f}$. Instead, it will be determined by the net force $\vec{F}$. Setting $w_0^\mathrm{s} \neq 1$ may cause local inflation at those nodes where $w_0^\mathrm{s} < \tilde{w}_\mathrm{j}^\mathrm{s}$ and local deflation at some other nodes where $w_0^\mathrm{s} > \tilde{w}_\mathrm{j}^\mathrm{s}$, depending on local surface smoothness. In other words, local surface inflation and deflation may occur simultaneously on different parts of a nonspherical model surface. The direction of the self–zoom force $\vec{F}_\mathrm{f}$ has been investigated experimentally, and its effects are demonstrated in figure 5. The initial spherical model surface inflates when $w_0^\mathrm{s} < 1$, deflates when $w_0^\mathrm{s} > 1$, and remains largely unchanged when $w_0^\mathrm{s} = 1$. The simultaneous local surface inflation and deflation is tested by initializing the model surface to an ellipsoid having axis lengths in the ratio of $3:2:1$ (fig. 6), and then allowing it to evolve under the influence of the locally controlled smoothness force alone ($w_0^\mathrm{s} = 1$ in eq. 12). The model surface converges to a nearly spherical shape. It has deflated on the parts with higher curvatures, and inflated on the parts with lower curvatures.

4.2 The magnitude of the self–zoom force

The magnitude of the self–zoom force, $|\vec{F}_\mathrm{f}| = |(w_0^\mathrm{s} - 1)\vec{F}_0|$, is determined by weight w_0^s and the magnitude of the smoothness force component $|\vec{F}_0|$ which is,

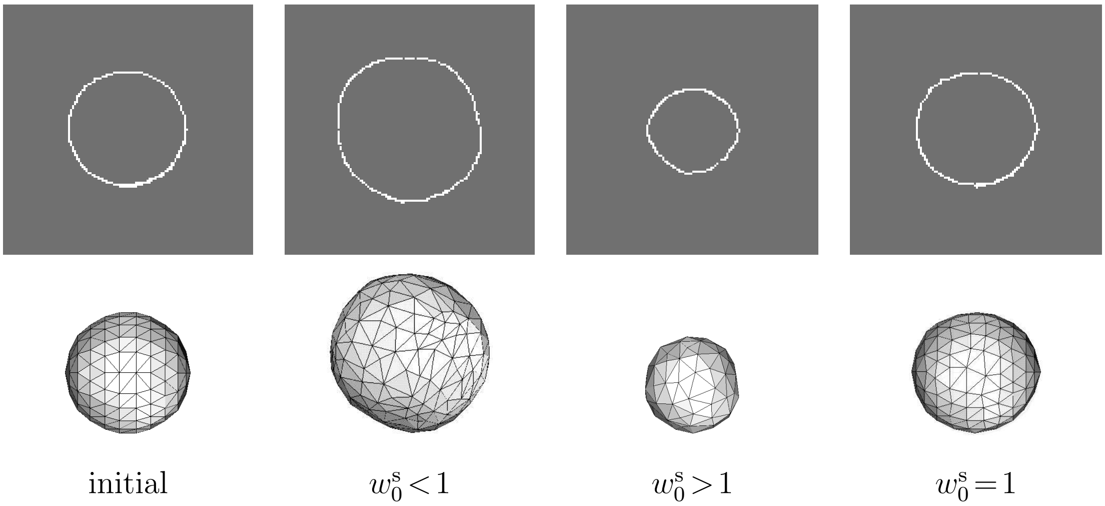


Fig. 5. Global inflation and deflation due to the self–zoom force $\vec{F}_{\mathrm{f}}$. The central cross-sections (top) and the surface views (bottom) of the model demonstrate inflation when $w_0^{\mathrm{s}}\!<\!1$, deflation when $w_0^{\mathrm{s}}\!>\!1$, and little change when $w_0^{\mathrm{s}}\!=\!1$.

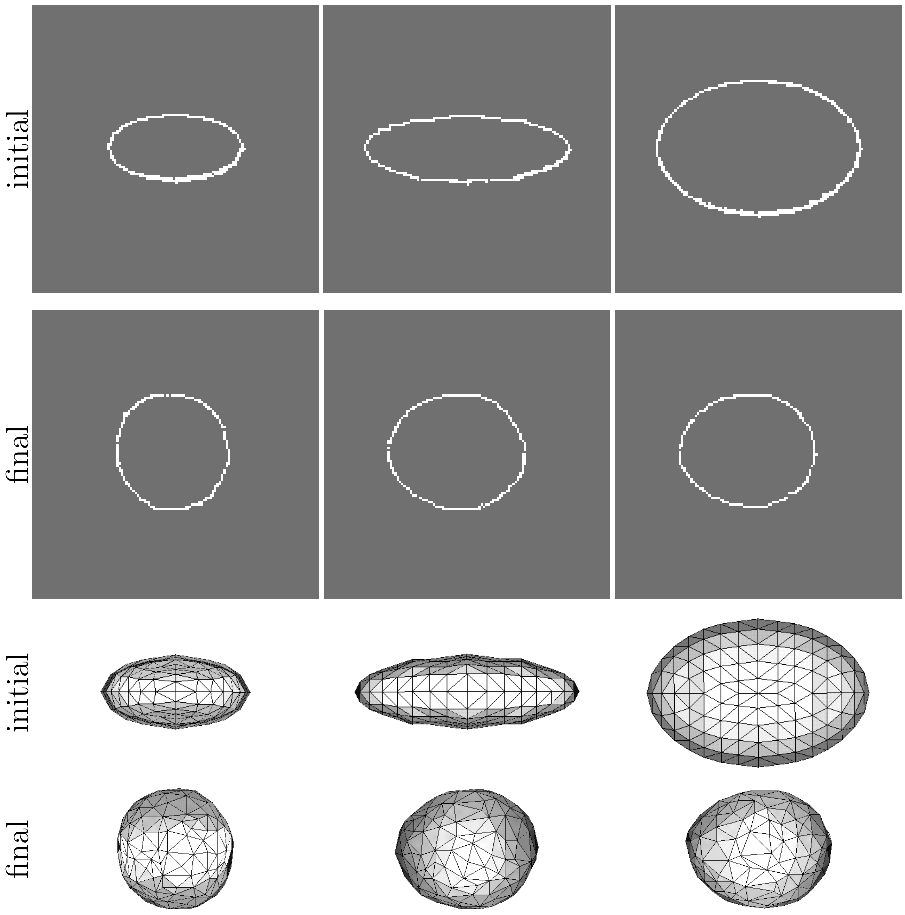


Fig. 6. Simultaneous local inflation and deflation with the smoothness force only ($w_0^{\mathrm{s}}=1$). The initial ellipsoid and the final mesh are shown in central orthogonal cross sections (top two rows) and orthogonal surface views (bottom two rows).

in turn, determined by the smoothness angle ϕ_i. It follows that the magnitude $|\vec{F}_\mathrm{f}|$ adaptively responds to local surface smoothness, rising with increasing ϕ_i and vanishing when $\phi_i = 0$. The user can control the magnitude of the self–zoom force by tuning weight w_0^s. The larger the discrepancy $|w_0^\mathrm{s} - 1|$, the stronger the self–zoom force.

The strength of the self–zoom force has been examined experimentally by measuring the convergence time for $w_0^\mathrm{s} = 0.1, 0.2, \cdots, 0.9$. A stronger self–zoom force, corresponding to a smaller value of w_0^s when $w_0^\mathrm{s} < 1$, will let the model converge faster to the object boundary. The image tested is a noise–free synthetic image containing an ellipsoid with axis lengths in the ratio of $2.25 : 1.5 : 1.0$. All segmentations start with the same initial spherical model surface, which comprises 86 mesh nodes, is totally contained within, and is concentric with, the ellipsoid (fig. 7a). The final meshes comprise almost the same number (430) of nodes. All meshes appear similar. One example ($w_0^\mathrm{s} = 0.6$) is shown in figure 7b. The processing time in seconds required for the model to converge to the ellipsoid boundary from its initial surface, is plotted against w_0^s in figure 8. The plot shows that the computational time decreases for lower values of w_0^s, demonstrating that the larger the discrepancy $|w_0^\mathrm{s} - 1|$, the stronger the self–zoom force and thus the faster the model expansion. The strength as well as the direction of the self–zoom force $\vec{F}_\mathrm{f}$ can be controlled by tuning only the weight w_0^s.

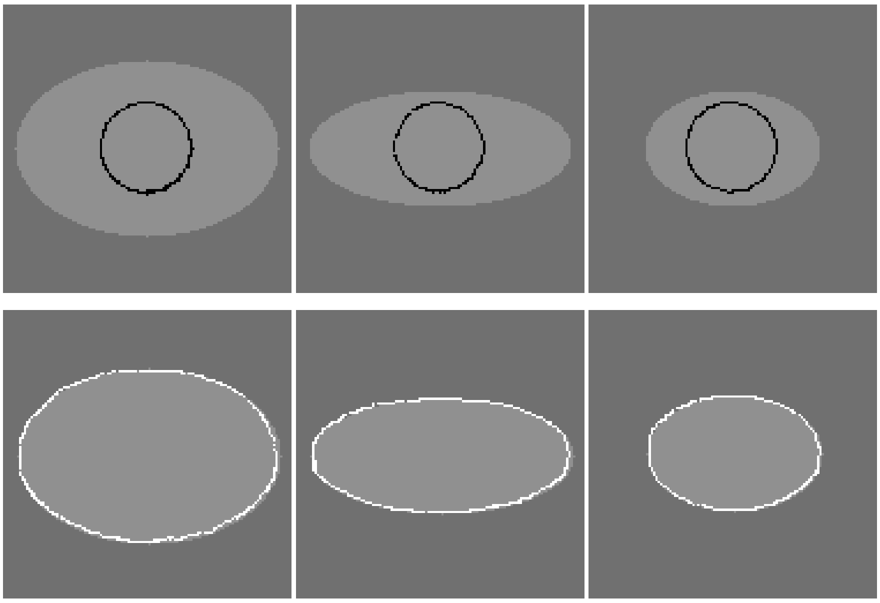


Fig. 7. A segmentation experiment used in evaluating the effect of the value of parameter w_0^s on the rate of evolution. The ellipsoid object is shown in lighter grey. The columns represent central orthogonal cross-sections. The initial sphere is superimposed in black (top row), and the final mesh, obtained with $w_0^\mathrm{s} = 0.6$ is shown in white (bottom row).

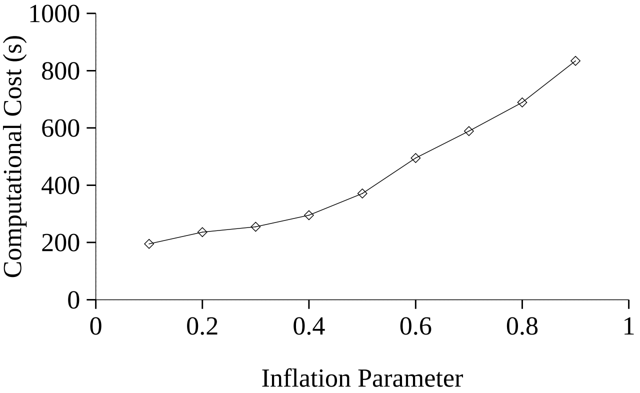


Fig. 8. The speed of surface evolution and the strength of the self–zoom force. The plot shows the dependence of the computational cost (in seconds) on the value of the parameter w_0^{s}.

4.3 Adaptability of the self–zoom force

The direction of the self–zoom force $\vec{F}_{\mathrm{f}}$ adaptively responds to the sign of local surface curvature (convex or concave). When $0 < w_0^{\mathrm{s}} < 1$, the self–zoom force $\vec{F}_{\mathrm{f}}$ tends to push node j outwards (upwards) where the local shape is convex (fig. 4b), and to pull it inwards (downwards) where the shape is concave (fig. 4d). This characteristic will help the model fit an object boundary with local convex and concave shapes.

We adopt a coarse–to–fine mesh approach to segmentation. The coarse mesh of the early evolution may result in the model overshooting a concave boundary and undershooting a convex boundary (fig. 9a) due to the low mesh resolution. New nodes (shaded) are inserted during the later stages of the evolution to increase the mesh resolution. Consider node j newly inserted between nodes 2 and 3, and node k between nodes 4 and 5 (fig. 9b). The self–zoom force $\vec{F}_{\mathrm{f}}$ at each node is zero, but will rise once nodes j and k have moved due to, say, the smoothness force $\vec{F}$. The force $\vec{F}_{\mathrm{f}}$ tends to move node j outwards and node k inwards (fig. 9c), thus helping the model capture both convex and concave boundaries. This manifests one advantage of the self–zoom force over the balloon force [9–11]. The latter force would not help the model capture convex and concave boundaries simultaneously as it has a fixed direction (either inwards or outwards) at all nodes regardless of whether the local surface shape is convex or concave.

The magnitude of the self–zoom force $\vec{F}_{\mathrm{f}}$ adaptively responds to the local surface smoothness, rising with increasing ϕ_i and vanishing to zero when $\phi_i = 0$. This characteristic will help the model fit high curvature boundaries and elongated objects containing high and low boundary curvatures. When the local shape is convex and $0 < w_0^{\mathrm{s}} < 1$, the smoothness force component $\vec{F}_0$ tends to pull node j inwards (fig. 4b). It will be larger at nodes with higher

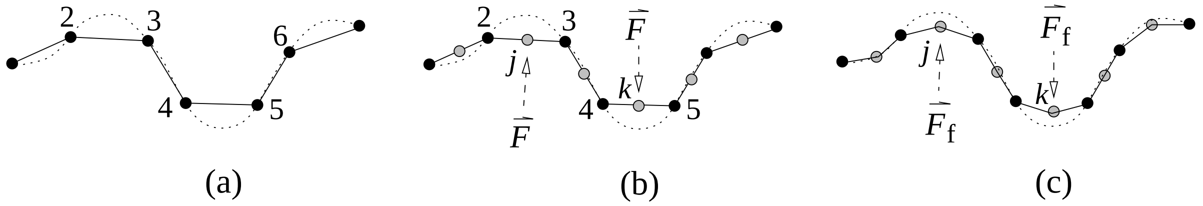


Fig. 9. Fitting convex and concave boundaries simultaneously (shown in 2D) with the help of the self–zoom force $\vec{F}_{\mathrm{f}}$. The object boundary is represented by the dotted line and the model by the solid line. Coarse mesh evolution gives the intermediate result (a), which is refined (b) by inserting new nodes (lightly shaded circles). The self–zoom force $\vec{F}_{\mathrm{f}}$ tends to move node j upwards and node k downwards (c).

curvatures where the smoothness angle ϕ is larger, as illustrated in figure 10. In order to inflate the model to the object boundary, a stronger inflation force is required to counterbalance the larger force component $\vec{F}_0$ at these nodes. The self–zoom force $\vec{F}_{\mathrm{f}}$ is well suited to counterbalance $\vec{F}_0$ as it will be large when $\vec{F}_0$ is large and will have the opposite sense to $\vec{F}_0$. This manifests another advantage of the self–zoom force $\vec{F}_{\mathrm{f}}$ over the balloon force which is uniform in magnitude at all nodes. Uniform inflation force may not be strong enough to counterbalance the smoothness force component $\vec{F}_0$ at higher curvature nodes, while it may be too strong at lower curvature nodes where $\vec{F}_0$ is weaker and overshoots the object boundary.

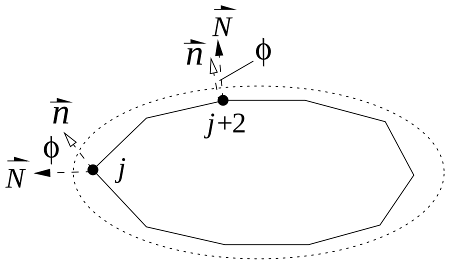


Fig. 10. The smoothness angle ϕ, and thus the smoothness force component $\vec{F}_0$ and the self–zoom force $\vec{F}_{\mathrm{f}}$, is larger at higher curvature node j than at lower curvature node $j{+}2$. The object boundary is represented by the dotted line and the model by the solid line.

4.4 Results

The self–zoom force was tested with a synthetic image, a clinical x–ray CT image and a clinical MR image. The synthetic image is of size $128\times128\times128$ and contains two ellipsoids that partly overlap. One ellipsoid has the half–axis lengths $r=\{25,50,40\}$ and the centre at $c=(30,64,64)$, and the other ellipsoid has $r=\{40,25,50\}$ and $c=(82,64,64)$. The synthetic image is contaminated with additive Gaussian noise at signal–to–noise ratio (SNR) of 1.0. Three initial surfaces were used: a sphere of radius $r=20$ and centre $c=(30,64,64)$

(fig. 11a), a sphere of the same radius but $c = (82, 64, 64)$ (fig. 11b), and an ellipsoid with $r = \{40, 13, 20\}$ and $c = (64, 64, 64)$ (fig. 11c). Segmentations with all three initial surfaces were carried out by the coarse–to–fine mesh evolution scheme and with the same parameter settings. The coarse–to–fine evolution starts with a coarse mesh which is maintained until it reaches the object boundary, whereupon the mesh is subdivided; the fine mesh then evolves until convergence. The model succeeded in converging to the object boundary in all three cases. One of the intermediate results (the result of the coarse mesh evolution) is shown in figure 11d. The three final results are visually almost identical; as an example, figure 11e shows the result of evolution that began with the initial surface of figure 11a. Figure 12 displays three orthogonal views of the intermediate mesh (top row) and the final mesh (bottom row). The clinical x–ray CT image contains a kidney (fig. 13), which fits in a cubic volume $150 \times 150 \times 150$. The initial surface was an ellipsoid with the half–axis lengths $r = \{20, 20, 25\}$ and the centre at $c = (82, 75, 80)$. Figure 13 shows selected slices: every 24th transverse, every 18th sagittal and every 18th coronal slice, as well as the end slice at each orientation, of the initial surface and the final result, as indicated. Note that the initial surface does not appear in all slices due to its small size. Figure 14 displays three views of the final mesh of the segmented kidney. The clinical MR image contains a segment of tibia (fig. 15), which fits in a cubic volume $256 \times 256 \times 85$. The cavity bounding the trabecular bone was to be segmented. The initial surface was an ellipsoid with the half–axis lengths $r = \{60, 50, 30\}$ and the centre at $c = (145, 135, 52)$ (fig. 15–1). The corresponding segmentation result is shown in Figures 15–2. Figures 15a,b,c show the transverse, sagittal and coronal cross–sections, respectively.

The tests show that the model converges to the object boundary even where the initial surface is far from the object boundary for all the synthetic overlapped–ellipsoids image, the CT kidney image and the MR tibia image, demonstrating the effectiveness of the self–zoom force in driving the model surface towards the object boundary and reducing the model's sensitivity to initial conditions. Visual inspection of the intermediate result (fig. 11d and fig. 12a) and the final result (fig. 11e and fig. 12b) of the segmentation of the synthetic joined–ellipsoids image reveals that even though the model overshoots the concave boundary during the coarse mesh evolution (fig. 11d and fig. 12a), the concave boundary is largely captured during the fine mesh evolution (fig. 11e and fig. 12b). This is attributed to the adaptive response of the direction of the self–zoom force to the sign of local surface curvature.

The advantage of the self–zoom force over the balloon force in fitting the model to a concave boundary was experimentally confirmed. Starting from an intermediate (or coarse mesh) segmentation result (fig. 16a), we segmented the synthetic joined–ellipsoids image using the balloon force as well as the self–zoom force. The two segmentations were carried out with the same parameter settings. The segmentations started with 104 mesh nodes and ended up with

628 (balloon) and 600 (self–zoom) mesh nodes. The results are shown in figures 16b and c, respectively. It can be observed that the concave boundary was better fitted using the self–zoom force than using the balloon force.

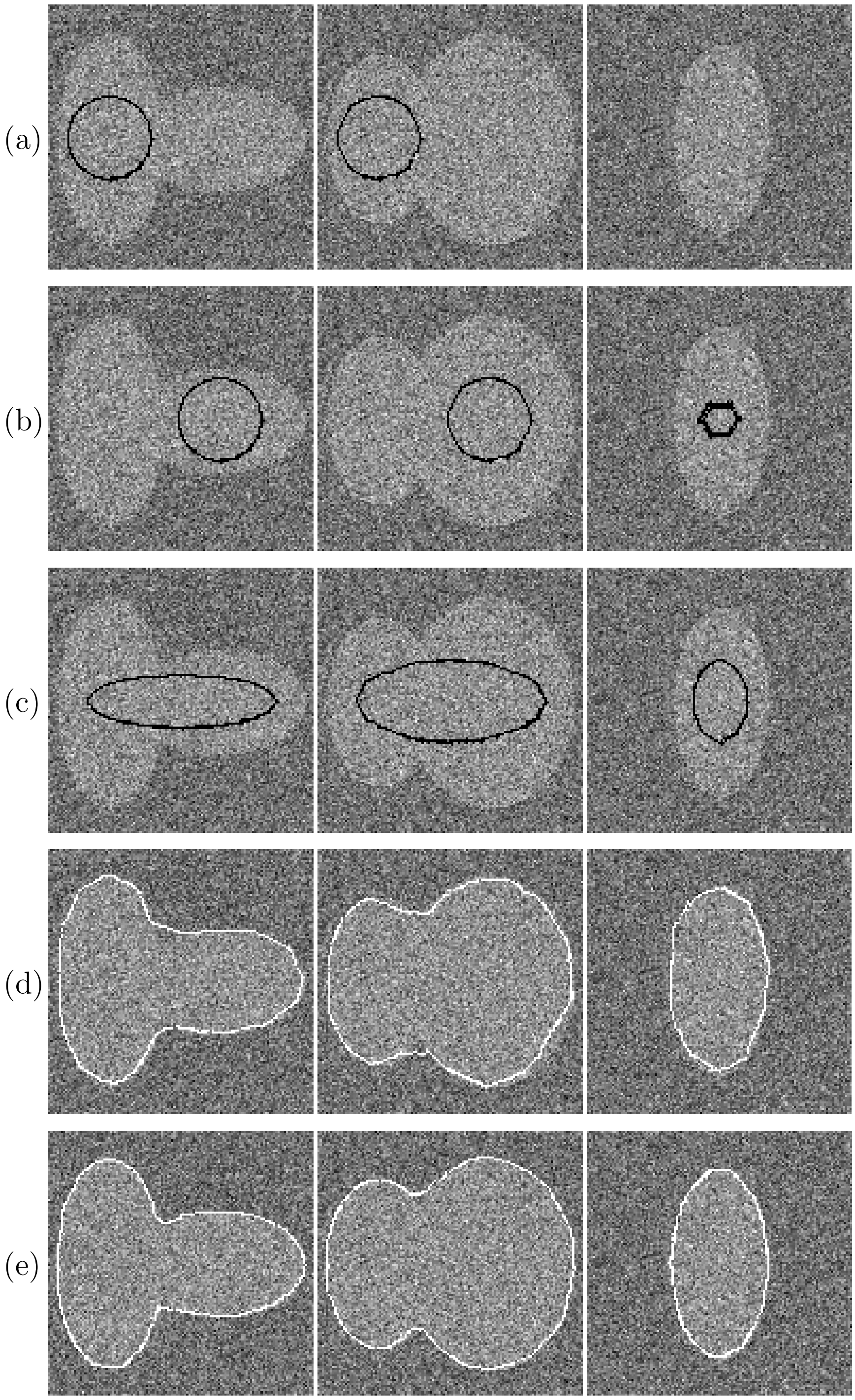


Fig. 11. Test of the self–zoom force. The three columns display the central orthogonal cross–sections of the synthetic joined–ellipsoids image. Shown are initial surfaces: (a-b) two spheres at different locations, and (c) an ellipsoid. Also shown are (d) the intermediate and (e) final outcomes.

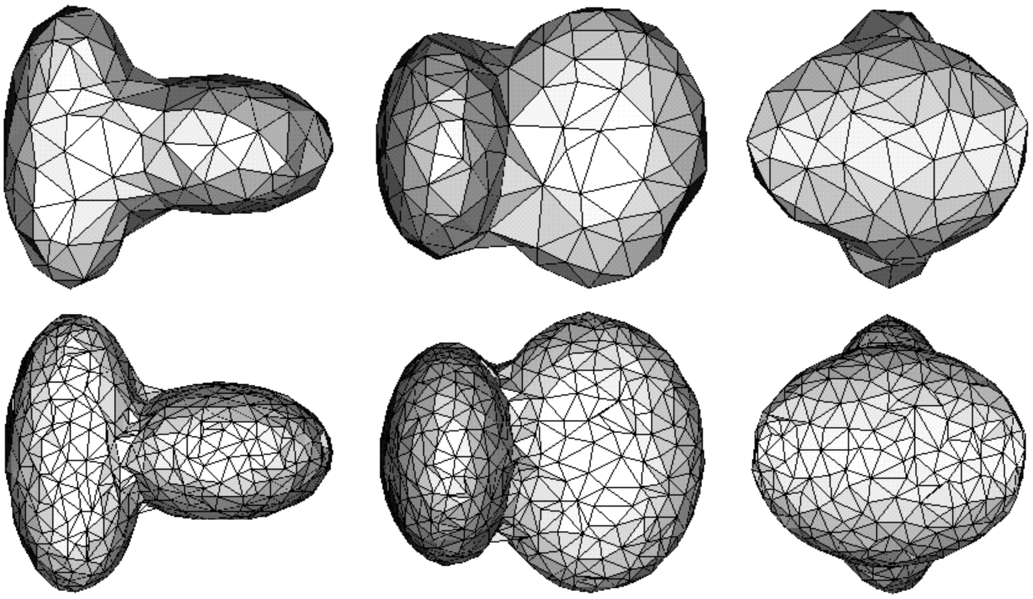


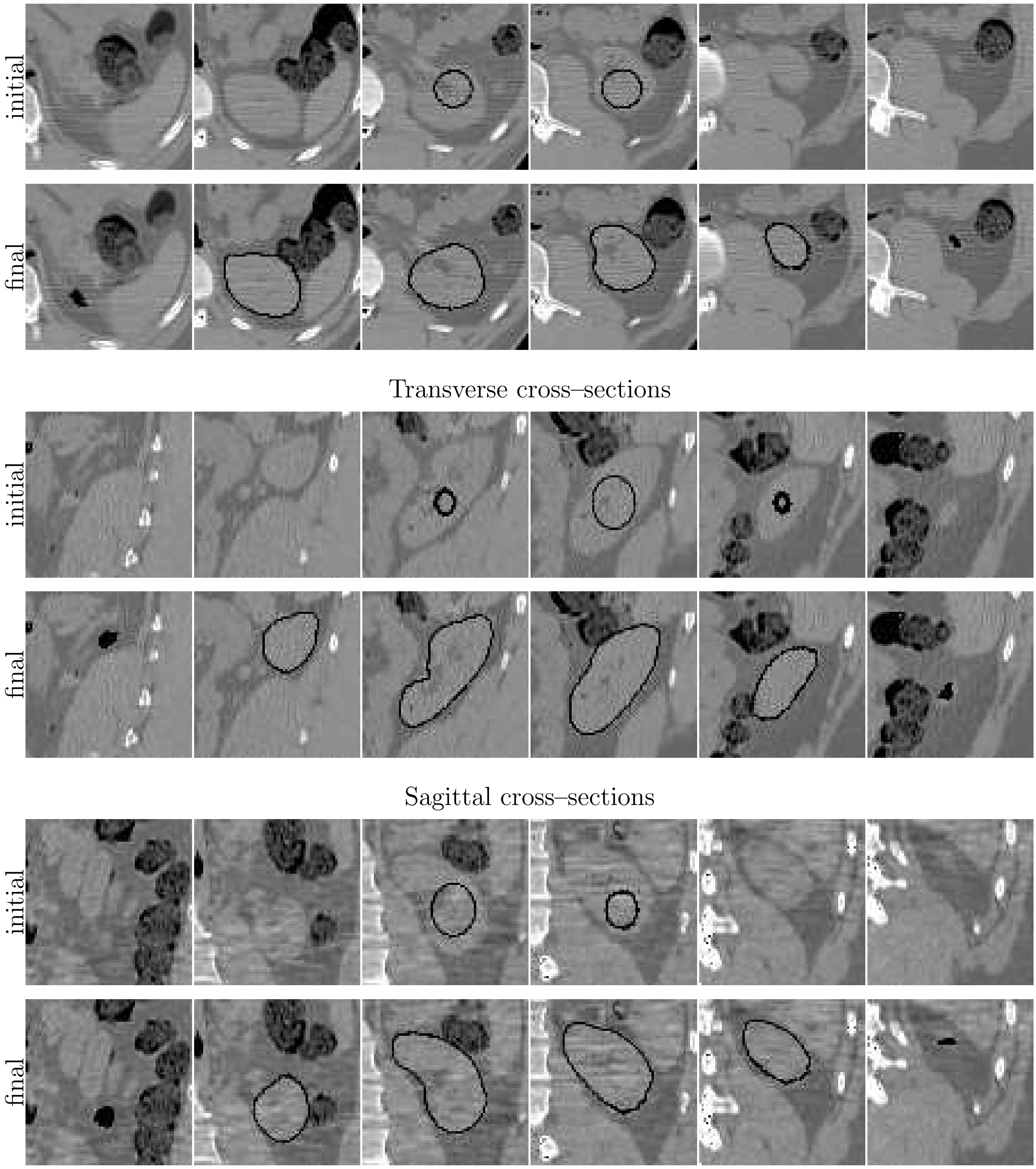
Fig. 12. Segmentation of the joined ellipsoids. Shown are three orthogonal views of the intermediate mesh (top row) and final mesh (bottom row). With the coarse–to–fine mesh evolution approach, the concave boundary is overshot during the coarse mesh evolution (top row), but it is largely captured during the fine mesh evolution (bottom row).

5 The corner fitting force

As discussed in the Introduction, conventional deformable models define high curvature boundaries (corners) poorly, due to the intrinsic smoothness constraint of deformable models. Segmentation of such objects is likely to result in the corners being smoothed ("cut"). An example is shown in figure 17. In this section, a corner fitting force, which makes use of the locally controlled smoothness force, is devised to address the problem.

5.1 Construction of the corner fitting force

For the sake of simplicity, we start with a 2D contour where, for each node j, there are only two adjacent nodes, $j+1$ and $j+2$ (fig. 18a). Each node has two equal smoothness angles ϕ_j subtended by the contour normal vector $\vec{N}_j$ at node j and the normal vector $\vec{n}_i$ of adjacent contour element i. Suppose that node j moves one step along the boundary towards the corner (fig. 18b). The smoothness angles at node j, as well as at its adjacent nodes $j+1$ and $j+2$, will change by $\Delta\phi_j$, $\Delta\phi_{j+1}$ and $\Delta\phi_{j+2}$, respectively. The change $\Delta\phi_j$ will be positive if node j moves towards the corner. The sum of the changes



Transverse cross–sections

Sagittal cross–sections

Coronal cross–sections

Fig. 13. Segmentation of the CT kidney image. Shown are selected transverse, sagittal and coronal slices. The black contours represent the initial ellipsoid surface and the final surface.

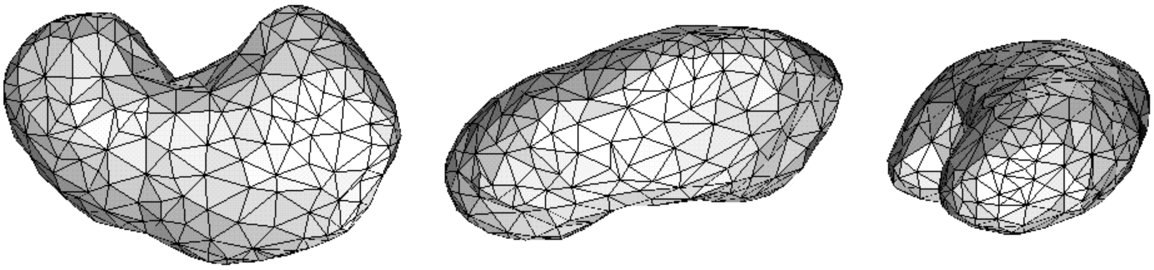


Fig. 14. Three views of the final mesh of the segmented kidney.

in smoothness angles will be zero,

$$\Delta\phi = \sum_{i=0}^{2} 2\,\Delta\phi_{j+i} = 2\,(\Delta\phi_j + \Delta\phi_{j+1} + \Delta\phi_{j+2}) = 0. \tag{14}$$

Assume that the external energy at all boundary pixels is uniform and that it is smaller than that at non–boundary pixels. Assume also that the internal energy consists of the smoothness energy only. Node j will move along the boundary to a new position where the smoothness energy is smallest locally. The change in the smoothness energy ΔE^{s} caused by the movement of node j will be negative. It is given (eq. 6) by the sum of the changes in the smoothness energy components $\Delta E^{\mathrm{s}}_{j+i}$ at nodes $j+i$ $(i = 0, 1, 2)$,

$$\Delta E^{\mathrm{s}} = \sum_{i=0}^{2} \Delta E^{\mathrm{s}}_{j+i} \;<\; 0, \tag{15}$$

$$\text{where} \qquad \Delta E^{\mathrm{s}}_{j+i} = 2\,[\,(\phi_{j+i} + \Delta\phi_{j+i})^2 - \phi^2_{j+i}\,].$$

By assigning a weight to each smoothness energy component in (15), ΔE^{s} becomes a weighted sum of $\Delta E^{\mathrm{s}}_{j+i}$,

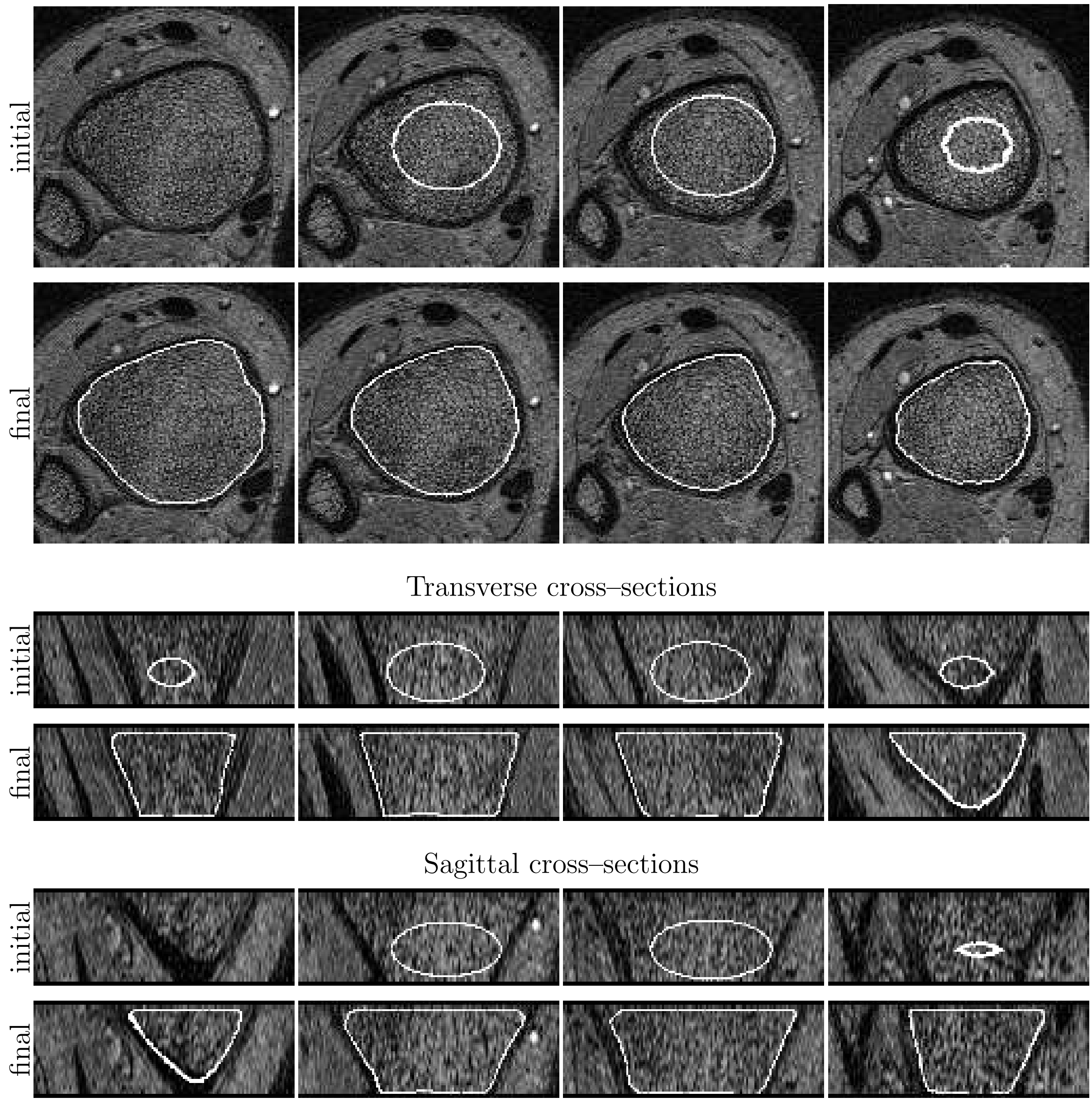
$$\Delta E^{\mathrm{s}} = \sum_{i=0}^{2} w^{\mathrm{s}}_{j+i}\,\Delta E^{\mathrm{s}}_{j+i} \;<\; 0. \tag{16}$$

Letting $w^{\mathrm{s}}_{j+1} = w^{\mathrm{s}}_{j+2} = 1$, without loss of generality, the condition (16) for node j to move to a new position can be written as

$$w^{\mathrm{s}}_j\,\Delta\phi_j\,(2\,\phi_j + \Delta\phi_j) + \Delta\phi_{j+1}\,(2\,\phi_{j+1} + \Delta\phi_{j+1}) + \Delta\phi_{j+2}\,(2\,\phi_{j+2} + \Delta\phi_{j+2}) < 0. \tag{17}$$

To appreciate its effect, let $\Delta\phi_{j+1} = 0$ and substitute $-\Delta\phi_j$ for $\Delta\phi_{j+2}$ (eq. 14). The inequality (17) becomes

$$w^{\mathrm{s}}_j\,\phi_j + \frac{1 + w^{\mathrm{s}}_j}{2}\Delta\phi_j \;<\; \phi_{j+2}. \tag{18}$$



Transverse cross–sections

Sagittal cross–sections

Coronal cross–sections

Fig. 15. Segmentation of a 3D MR trabecular bone image of size 256 × 256 × 85. Shown are: every 20th transverse, every 35th sagittal, and every 35th coronal slice. The white contours represent the initial and final surfaces. The initial ellipsoid has half–axis lengths r={60,50,30} and the centre c=(145,135,52).

Setting $w_j^{\mathrm{s}} = 1$, this becomes

$$\phi_j + \Delta\phi_j \; < \; \phi_{j+2}, \tag{19}$$

suggesting that, although node j may be able to move a few steps towards the corner, it will never reach the corner. As the node advances towards the

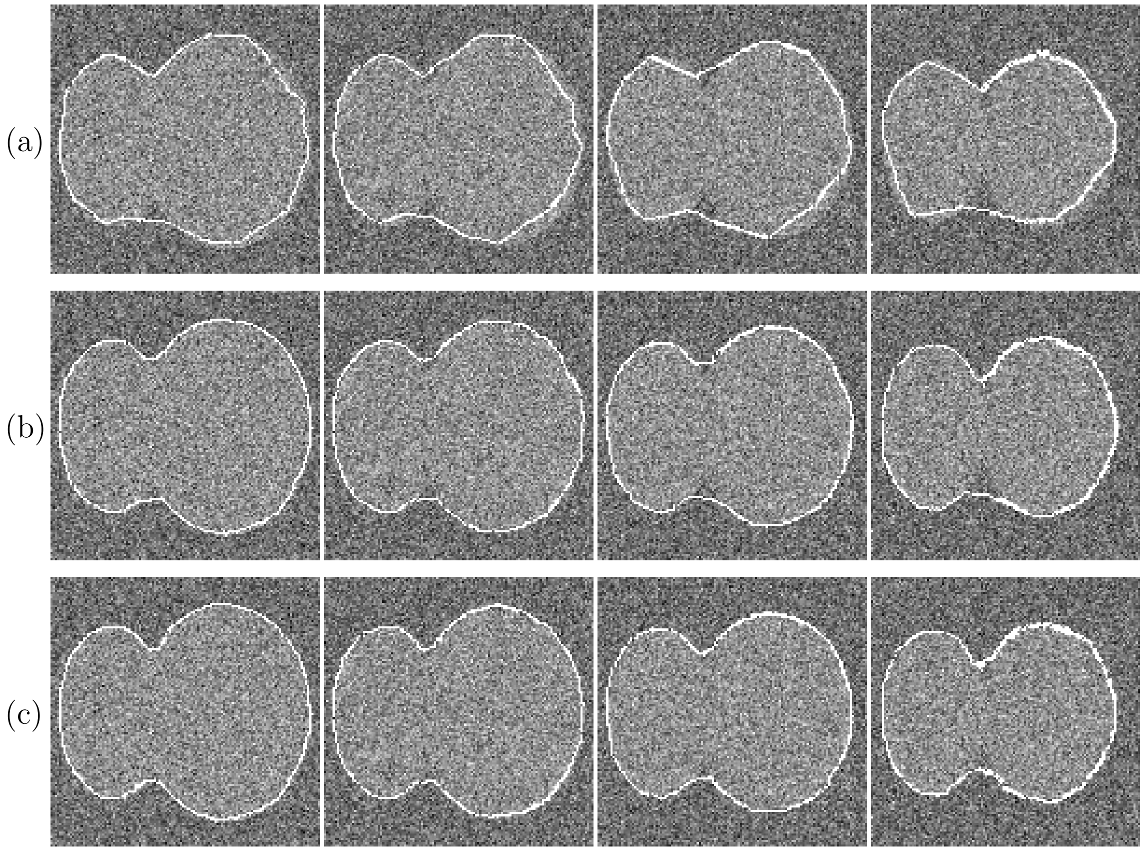


Fig. 16. A comparison of the adaptable self–zoom force and the balloon force. The synthetic joined–ellipsoids image, shown in four successive slices, was segmented with the coarse–to–fine mesh scheme. To facilitate the comparison, the fine stage of both segmentations started from the same intermediate result (a). Both final meshes had a similar number of nodes. (b) and (c) show the final results using the balloon force and the self–zoom force, respectively. It can be seen that the concave boundary was better fitted using the adaptable self–zoom force.

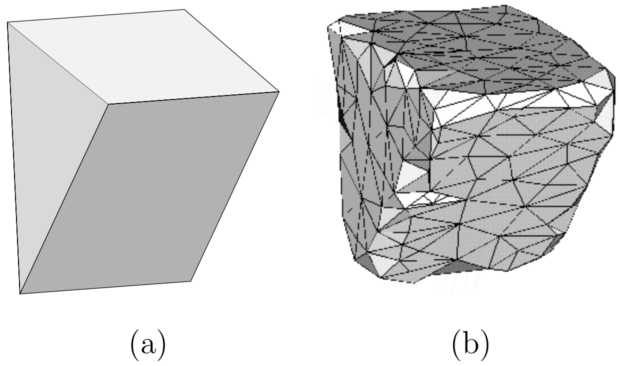


Fig. 17. Poor segmentation of an object with high curvature boundary. Edges and corners of the triangular prism (a) become smoothed (b).

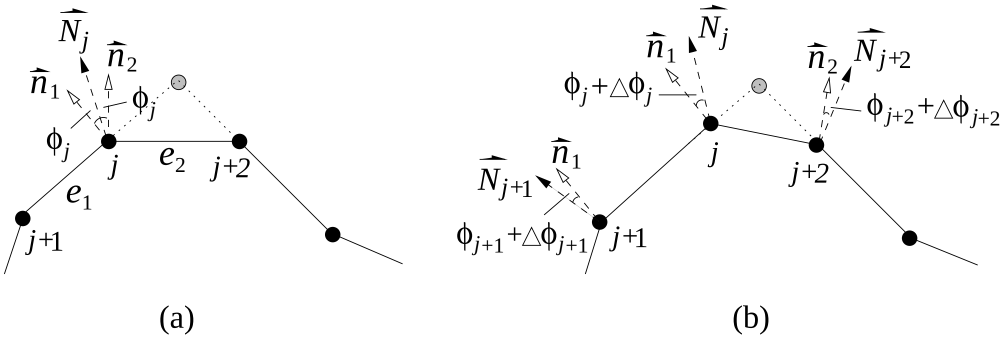


Fig. 18. Node movement towards a corner. Shown are configurations (a) before and (b) after a single movement of node j. Solid lines represent the model contour and dotted lines represent the object boundary.

corner, the smoothness angle ϕ_j will increase while ϕ_{j+2} will decrease, due to the conditions $\Delta\phi_j > 0$ and $\Delta\phi_{j+2} < 0$. Node j will stop moving once it has reached the equilibrium point where $\phi_j \approx \phi_{j+2}$. This is the case commonly encountered in conventional deformable models. If we now set $0 \leq w_j^{\mathrm{s}} < 1$ in eq. i.e. if we reduce the relative weight for the smoothness energy at node j, the equilibrium point will shift towards the corner, allowing node j to move closer to the corner. Node j will ultimately reach the corner if w_j^{s} is sufficiently small.

The condition for a node to move into a corner can be explained in terms of the locally controlled smoothness force. Each term in inequality 16 corresponds to a smoothness force component $\vec{F}_{j+i}$, $i = 0, 1, 2$ (refer to eq. 8). The force driving node j into the corner is the force component $\vec{F}_{j+2}$ acting on node j but arising from adjacent node $j+2$ (see fig. 18 and fig. 3). The force component $\vec{F}_{j+2}$ tends to reduce the smoothness energy at node $j+2$ by moving node j towards the corner. However, it is counterbalanced by the force component $w_j^{\mathrm{s}}\vec{F}_j$ acting on and arising from node j, while the force component $\vec{F}_{j+1}$ does little to oppose the movement of node j into the corner. The strength of the force component $w_j^{\mathrm{s}}\vec{F}_j$ can be controlled by weight w_j^{s}. By reducing w_j^{s}, node j will be driven towards the corner along the object boundary by the force component $\vec{F}_{j+2}$. The smaller the magnitude of weight $w_j^{\mathrm{s}} \geq 0$, the weaker the counter–force $w_j^{\mathrm{s}}\vec{F}_j$, and thus the stronger the net force driving node j into the corner. It is clear that the corner fitting force shares the same mechanism with the self–zoom force.

The principle of the corner fitting force in the case of 2D contour can be generalized directly to the case of the 3D surface. By setting weights $w_i^{\mathrm{s}} = 1.0$ $(i = 1, \ldots, I_j)$ and controlling weight w_0^{s} in equation 8, node j of a 3D mesh can be driven towards a corner.

The corner fitting force was tested with a synthetic image containing a triangular prism with angles 45°, 60° and 75° (fig. 19). In order to reduce extraneous influences, the image was noise–free, and the edges of the triangular prism were thin, continuous and uniform in strength. The initial surface was spherical and contained within the triangular prism. Segmentation of the triangular prism image was performed in two steps: first, the initial surface was allowed to inflate until it reached the triangular prism's boundary; then, the mesh nodes with high surface curvature were selected as corner candidates, and the smoothness energy weight for each candidate node was set to a small value ($w_0^s = 0.3$) while the weights for its I_j adjacent nodes were set to one.

In the absence of the corner fitting force, edges and corners were poorly defined (top row of fig. 19) even though about 98% of the mesh nodes have reached the object boundary. This result is consistent with the available mesh resolution defined in terms of mean internodal distance. As shown in the bottom row of figure 19, the corner fitting force successfully drives the mesh nodes into edges and corners along the object boundary. The superior definition of the edges and corners represents a locally increased mesh resolution achieved without decreasing the mean internodal distance.

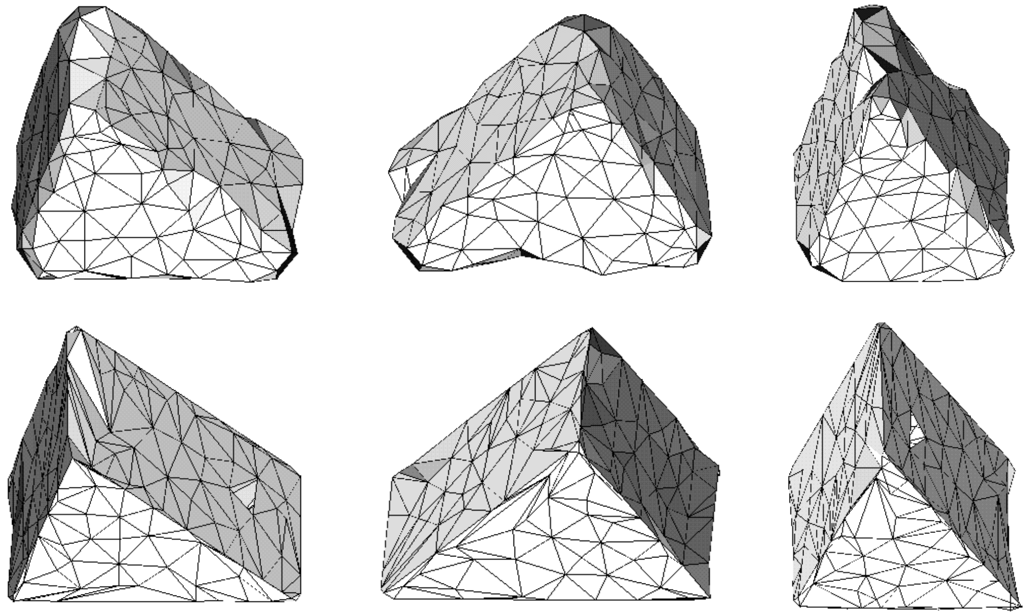


Fig. 19. Segmentation of an object with corners. Shown are three views of a mesh obtained without (top row), and with (bottom row) the corner fitting force.

6 Conclusion

A new locally controlled smoothness force is described. It is based on the smoothness force that is locally resolved into the constituent components. By controlling the relative strengths of the components, the smoothness force allows the model to relax or strengthen its smoothness constraint at individual nodes, thus facilitating local inflation, deflation and corner fitting.

Based on the locally controlled smoothness force, a long range force, the self–zoom force, has been devised. This force enables the model to travel any desired distance towards the object boundary. Both the direction and strength of the self–zoom force can be easily controlled by the user through tuning a single parameter. Like the smoothness force, the self–zoom force is locally adaptive. Its direction depends on the sign of local surface curvature (convex or concave), while its magnitude varies with local surface smoothness. The adaptive direction of the self–zoom force helps the model fit local concavities and convexities simultaneously. Experimental results demonstrate these characteristics of the self–zoom force.

Also based on the locally controlled smoothness force, a corner fitting force is devised to fit the model into high curvature boundaries. It has the advantage of providing a force to drive mesh nodes towards corners, unlike the previous corner fitting algorithms [14,15] based only on relaxing the smoothness force. The magnitude of the corner fitting force can be easily controlled by a single parameter. A test with a prism object shows that the corner fitting force significantly improves the segmentation of high curvature boundaries.

Acknowledgment

Z. Zhang received financial support from the Faculty of Science, the University of Technology, Sydney, Australia.

References

[1] D. Terzopoulos, A. Witkin and M. Kass, Constraints on deformable models: recovering 3D shape and nonrigid motion, *Artificial Intelligence* **36**, pp. 91–123, 1988.

[2] D. Terzopoulos and K. Fleischer, Deformable models, *The Visual Computer* **4**(6), pp. 306–331, 1988.

[3] M. Kass, A. Witkin and D. Terzopoulos, Snakes: active contour models, *International Journal of Computer Vision* **1**, pp. 321–331, 1988.

[4] F. Leymarie and M.D. Levine, Tracking deformable objects in the plane using an active contour model, *IEEE Trans. Pattern Analysis and Machine Intelligence* **15**(6), pp. 617–634, 1993.

[5] Y.F. Wang and J.F. Wang, Surface reconstruction using deformable models with interior and boundary constraints, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **14**(5), pp. 572–579, 1992.

[6] W.C. Huang and D.B. Goldgof, Left ventricle motion modeling and analysis by adaptive–size physically–based models", In *SPIE—Biomedical Image Processing and Three–Dimensional Microscopy* **1660**, pp. 299–310, 1992.

[7] H. Delingette, Adaptive and deformable models based on simplex meshes", In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition* pp. 152–157, 1994.

[8] A.J. Bulpitt and N.D. Efford, An effective 3D deformable model with a self–optimizing mesh", *Image Vision Computing*, **14**, pp. 573–580, August 1996.

[9] L.D. Cohen, On active contour models and balloons", *Computer Graphics and Image Processing: Image Understanding* **53**(2), pp. 211–218, 1991.

[10] I. Cohen and L.D. Cohen and N. Ayache, Using deformable surfaces to segment 3–D images and infer differential structures, *Computer Graphics and Image Processing: Image Understanding* **56**(2), pp. 242–263, 1992.

[11] L.D. Cohen and I. Cohen, Finite–element methods for active contour models and balloons for 2–D and 3–D images", *IEEE Trans. Pattern Analysis and Machine Intelligence* **15**(11), pp. 1131–1147, 1993.

[12] C. Xu and J.L. Prince, Gradient vector flow: a new external force for snake, In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition* pp. 66–71, 1997.

[13] C. Xu and J.L. Prince, Snakes, shapes and gradient vector flow, *IEEE Trans. on Image Processing* **7**(3), pp. 359–369, 1998.

[14] S. Menet, P. Saint-Marc and G. Medioni, Active contour models: overview, implementation and application, In *Proc. IEEE Int. Conf. on Syst. Man and Cybern* pp. 194–199, 1990.

[15] G. Xu, E. Segawa and S. Tsuji, A robust active contour model with insensitive parameters, In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition* pp. 562–566, 1993.

[16] Z. Zhang and M. Braun, Fully 3D active surface models with self–inflation and self–deflation forces", In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition* pp. 85–90, 1997.

[17] Z. Zhang and M. Braun, A deformable model–based image segmentation algorithm for shapes with corners, In *Proc. of Int. Conf. on Pattern Recognition* pp. 392–394, 1998.

[18] T. McInerney and D. Terzopoulos, Deformable models in medical image analysis: a survey, *Medical Image Analysis* **1**(2), pp. 91–108, 1996.

[19] A. Ghanei, H. Soltanian-Zadeh and J.P. Windham, A 3D deformable surface model for segmentation of objects from volumetric data in medical images, *Computers in Biol. & Med.* **28**, pp. 239–253, 1998.

Summary

Deformable models, originally proposed by Terzopoulos *et al* and Kass *et al* in 1988, have been widely used in medical image segmentation. However, they manifest certain well known limitations which restrict the scope of their applications. One limitation is the lack of an appropriate long range force. Away from the object boundary, the model surface may experience little or no force capable of attracting it towards the boundary, leading to a segmentation failure. This is one of the reasons why, in most implementations, the model is initialized close to the object boundary. Another limitation arises from the fact that many biological organs manifest complex borders with high-curvature components. The intrinsic smoothness constraint of deformable models prevents them from segmenting such regions accurately.

In this paper, a new smoothness force with local control is proposed. It is based on the smoothness force that is locally resolved into the components. By controlling the relative strengths of the components, the smoothness force allows the model to relax or strengthen its smoothness constraint at individual nodes, thus facilitating local inflation, deflation and corner fitting. Based on the local control, a long range force, referred to as the self–zoom force, has been constructed. It enables the model surface to expand and shrink without a limit in range so that the model surface can reach the object boundary. Both the direction and strength of the force can be easily controlled through tuning a single parameter. The self–zoom force is locally adaptive. Its direction depends on the sign of local surface curvature (convex or concave), while its magnitude varies with local surface smoothness. The adaptive direction of the force helps the model fit local concavities and convexities simultaneously. Experimental results demonstrate these characteristics of the self–zoom force.

Also based on the locally controlled smoothness force, a corner fitting force is devised that propels the model surface to fit high curvature boundaries (corners). Unlike previous corner fitting algorithms, which merely relax the smoothness force, this algorithm actively drives the surface nodes towards object corners. The magnitude of the corner fitting force can be easily controlled by a single parameter. A test with a prism object shows that the corner fitting force significantly improves the segmentation of high curvature boundaries even if the initial estimate is not close.