Medical Image Analysis 17 (2013) 429–441

Contents lists available at SciVerse ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Omnidirectional displacements for deformable surfaces

Dagmar Kainmueller [a,*], Hans Lamecker [a], Markus O. Heller [b], Britta Weber [a], Hans-Christian Hege [a], Stefan Zachow [a]

[a] Zuse Institute Berlin, Berlin, Germany
[b] Julius Wolff Institute, Charité – Universitätsmedizin, Berlin, Germany

ARTICLE INFO

Article history:
Received 13 January 2012
Received in revised form 25 July 2012
Accepted 30 November 2012
Available online 14 December 2012

Keywords:
Segmentation
Deformable surfaces
Markov random field
Mandible
Acetabulum

ABSTRACT

Deformable surface models are often represented as triangular meshes in image segmentation applications. For a fast and easily regularized deformation onto the target object boundary, the vertices of the mesh are commonly moved along line segments (typically surface normals). However, in case of high mesh curvature, these lines may not intersect with the target boundary at all. Consequently, certain deformations cannot be achieved. We propose *omnidirectional displacements for deformable surfaces* (*ODDS*) to overcome this limitation. ODDS allow each vertex to move not only along a line segment but within the volumetric inside of a surrounding sphere, and achieve globally optimal deformations subject to local regularization constraints. However, allowing a ball-shaped instead of a linear range of motion per vertex significantly increases runtime and memory. To alleviate this drawback, we propose a hybrid approach, *fastODDS*, with improved runtime and reduced memory requirements. Furthermore, fastODDS can also cope with simultaneous segmentation of multiple objects. We show the theoretical benefits of ODDS with experiments on synthetic data, and evaluate ODDS and fastODDS quantitatively on clinical image data of the mandible and the hip bones. There, we assess both the global segmentation accuracy as well as local accuracy in high curvature regions, such as the tip-shaped mandibular coronoid processes and the ridge-shaped acetabular rims of the hip bones.

1. Introduction

In this paper, we address the issue of segmenting highly curved anatomical structures in three-dimensional medical image data. The aim is to improve segmentation accuracy. Segmentation methods based on *deformable models* (Terzopoulos, 1988; Xu et al., 2000; He et al., 2008) have been shown to cope in a highly robust manner with typical imaging deficiencies, such as noise, artifacts, partial volume effects, low or no contrast due to adjacent anatomical structures with similar appearance, etc. The basic idea is to deform a given (template) shape in such a way that the deformed shape provides an optimal geometric representation of the corresponding structure in the image.

Among many different types of deformable models, meshes are advantageous in many respects, such as flexibility and topology preservation (Montagnat et al., 2001). Typically, the degrees of freedom of the deformable mesh are increased in a multi-level fashion (Okada et al., 2007; Ma et al., 2010; Yin et al., 2010; Zhang et al., 2010; Seim et al., 2008; Kainmueller et al., 2007). At first, only global deformations like rigid transformations or statistical variations (Cootes et al., 1995; Heimann and Meinzer, 2009) are allowed. This robustly produces initial deformed shapes that roughly capture the structure sought-after in the image. On the finer levels, more local assumptions are made on deformations (Okada et al., 2007; Ma et al., 2010), in order to allow for more flexibility and thus capture the specific details of the structure in the given image data. On the finest level, each vertex position of the mesh can move "freely", subject only to regularity constraints that consider its direct neighborhood (Yin et al., 2010; Zhang et al., 2010; Seim et al., 2008; Kainmueller et al., 2007). We refer to such kind of deformations as *free deformations*.

Usually, the deformable mesh *probes* the image information at each vertex position: The image data is evaluated within a certain *search space* to assess suitable image features. Given these probes, a new shape is computed by *displacing* the vertices of the mesh, following a trade-off between image fidelity and anatomically plausible deformation. Note that for free deformations, search space and resulting displacement of an individual vertex are closely related, while this is in general not the case for global deformations, where individual resulting vertex displacements may deviate arbitrarily from the respective search space.

The details of the image probing play a crucial role in the segmentation process. To this end, *unidirectional* (i.e. linear, one-dimensional) search spaces per vertex of the deformable mesh are commonly used (Heimann and Delingette, 2011) due to a

* Corresponding author.
E-mail address: kainmueller@zib.de (D. Kainmueller).

1361-8415/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.media.2012.11.006

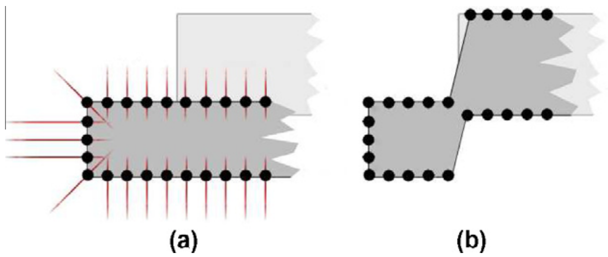

Fig. 1. 2D sketch of an exemplary deformable mesh (dark grey, with vertices as black dots) and target object (light grey). (a) Normal search spaces (directions indicated by lines through vertices) on a tip-like structure detect no target boundary points for a large set of vertices. (b) Resulting unregularized deformation onto target object boundary. Avoiding self-intersection of the mesh suppresses displacement of bottom left-most vertex.

number of benefits: (1) Feature assessment is fast within one-dimensional subsets of the image; (2) It is easy to select the "best" feature, as required by many methods (Cootes et al., 1995; Kainmueller et al., 2007), because a one-dimensional search space is likely to hit the target surface at only one single point (or at most a finite number of points), and hence the set of suitable features is likely to be small; (3) Free deformations can be computed in a globally optimal way for unidirectional search spaces (Li et al., 2006); (4) Normal vertex displacements implicitly restrict the deformation of the surface in a way that reduces (but does not prevent) the risk of generating mesh inconsistencies like self-intersections or fold-overs.

However, unidirectional search spaces suffer from *restricted visibility*: They are prone to miss features in the image data (Fig. 1). In case global deformations are employed, this problem may be alleviated by the fact that individual vertex displacements are not tightly coupled to their respective search spaces. On free deformations, however, the problem has a severe impact: E.g., local translations of highly curved surface regions such as tips or ridges can hardly be achieved (cf. Figs. 1 and 4). This holds true independently of the chosen mesh resolution.

One approach to confront the visibility problem is repeated – i.e. *iterative* – search for image features and respective deformation, where the hope is that visibility will improve in the next iteration. There is, however, no guarantee to this end. Furthermore, iterative deformation of meshes may easily lead to mesh inconsistencies such as self-intersections (Park et al., 2001). This requires additional remedial actions such as adaptive step-size control, adaptive remeshing or mesh surgery (Bucki et al., 2010).

In this paper, we propose a method to overcome the visibility problem for free deformations.[1] The basic idea is to enlarge the search space for image features to allow not only unidirectional but *omnidirectional* displacements at each point of the deformable model. On a deformable mesh, we asses features at – and allow displacements to – a discrete set of points within a ball[2] around each vertex, thus guaranteeing visibility within some radius. Free deformations are modeled by penalizing differences of displacements on edge-connected mesh vertices. This discrete formulation enables us to frame the segmentation problem as a Markov Random Field (MRF), as will be explained in Section 2. MRFs can be optimized efficiently (Komodakis et al., 2008), which has made them attractive for many applications in image processing and computer graphics (see e.g. Glocker et al., 2008; Paulsen et al., 2010). We denote the method of ball-shaped search spaces combined with MRF optimization for surface mesh deformation as *omnidirectional displacements for deformable surfaces*, or *ODDS*.

Allowing a three-dimensional search space per mesh vertex has the drawback of significantly increased run-time and memory requirements as compared to unidirectional search spaces. Therefore, we also propose an extension to ODDS that is faster and less memory-intensive – denoted as *fastODDS*. The key idea for fastODDS, presented in detail in Section 3, is to allow omnidirectional displacements only in regions of high curvature, while restricting displacements to surface normals in "flat" surface regions.

Section 4 provides an extensive evaluation of ODDS and fastODDS on synthetic and clinical data. In Section 5 we will analyze and discuss these results in depth. Here, we will also address the influence of mesh resolution and mesh consistency.

In summary, our results indicate that

1. ODDS can handle free deformations of meshes with high curvature where previous approaches based on normal displacements fail.
2. fastODDS keep all the benefits of ODDS for highly curved surface regions, while being twice as fast and requiring 50% less memory.
3. In contrast to ODDS, fastODDS can also be applied successfully for simultaneous segmentation of multiple objects.

2. ODDS

For a more thorough search for image features in terms of the visibility problem (see Section 1), we propose to extend the search space at each vertex of a deformable surface mesh from a line segment to a ball centered at the respective vertex position. We define the segmentation problem as a trade-off between finding suitable image features within these ball-shaped search spaces and simultaneously considering local regularization.

Volumetric (three-dimensional) ball-shaped search spaces of neighboring vertices overlap heavily in case the ball radius is bigger than the distance between the respective vertices; furthermore, individual search spaces most probably contain a whole region (two-dimensional manifold) of the target surface. Hence highly inconsistent (dissimilar) displacements on neighboring vertices may point to the target surface. The type of local regularization we employ must be able to avoid highly inconsistent displacements of adjacent vertices. We achieve this in a discrete setting (Sections 2.1 and 2.2) via Markov Random Field (MRF) energy minimization (Section 2.3).

We denote the set of vertices v of the deformable surface mesh as $V = \{v_i \in \mathbf{R}^3 | i = 1 \cdots n_V\}$, and the set of pairs of adjacent (i.e. edge-connected) vertices (v, w) as $E \subset V \times V$. Each vertex v can be moved by adding a vector, or *displacement*, $s \in S$, where $S = \{s_i \in \mathbf{R}^3 | i = 1 \cdots n_S\}$ is a discrete set of possible displacements. We refer to a mapping $d : V \to S,\ v \mapsto d(v) =: d_v$ that assigns a displacement to each vertex as *displacement field*. We call a position $v + s$ *sample point*. The set of sample points $v + S$ defines the *search space* for vertex v. Note that this definition has the effect that the search space of a vertex equals its *range of motion*.

2.1. Omnidirectional displacements

We define S as a set of displacements that are uniformly distributed within a ball of radius r_S, i.e. $\forall s \in S : \|s\| < r_S$, where r_S is a parameter of our method. Displacements in S are arranged as a cubic close-packed lattice (Conway et al., 1999); see Fig. 2a for a 2D sketch. We denote the minimum Euclidean distance between unequal displacements $s_i,\ s_j \in S$ as *sampling distance* $\delta_S := min_{s_i \neq s_j} \|s_i - s_j\|$.

[1] This work extends the authors' paper presented at MICCAI 2010 (Kainmueller et al., 2010), from which some text passages and figures are reused with kind permission from Springer Science + Business Media.

[2] Note that we use the term *ball* to refer to the volumetric (three-dimensional) interior of a sphere, while with the term *sphere* we refer to the surface of a ball, i.e. a two-dimensional manifold embedded in 3d.

We refer to this ball-shaped set of displacements as *omnidirectional displacements*. Note that omnidirectional displacements are interpreted in the same (world) coordinate frame for all vertices (see Fig. 2b and c). In consequence, applying the "same" displacement to different vertices means shifting these vertices by the same three-dimensional vector. This is different from traditional, unidirectional sets of displacements (i.e. *unidirectional displacements*), where the set of displacements is specified via a set of lengths, and actual displacement vectors are obtained as vectors of the respective lengths in vertex-individual directions. In other words, each vertex has its individual set of displacements, and "same" displacements on different vertices have same lengths but may have different directions.

2.2. Objective function

For each displacement $s \in S$ and vertex $v \in V$, a scalar cost $\phi(v,s) \geqslant 0$ encodes whether sample point $v+s$ is believed to lie on the target object boundary within the image $I : \mathbf{R}^3 \rightarrow \mathbf{R}$. The stronger the belief, the lower the cost. In other words, $\phi(v,s)$ serves as a penalty for the case that v is displaced by s. In general, however, any $\phi : V \times S \rightarrow \mathbf{R}_0^+$ is feasible as cost function.

For each two displacements s_i, s_j, a scalar "distance" $\psi(s_i,s_j) \geqslant 0$ serves as a penalty for the case that s_i and s_j occur on adjacent vertices. The distance function $\psi : S \times S \rightarrow \mathbf{R}_0^+$ takes care of regularization. In the following, we assume that ψ is monotonically increasing with the Euclidean norm of the difference of displacements, $\|s_j - s_i\|$, and depends on nothing else. Whenever it adds to clarity, we sloppily denote $\psi(s_i,s_j) = \psi(\|s_j - s_i\|)$.

We define the mesh adaptation problem as follows:

$$d = \underset{\{\hat{d}_v : v \in V\}}{\operatorname{argmin}} \sum_{v \in V} \phi(v,\hat{d}_v) + \sum_{(v,w) \in E} \psi(\hat{d}_v,\hat{d}_w) \tag{1}$$

This means we are looking for the displacement field d that minimizes an objective function that sums up the image costs and distance penalties for all vertices. Note that Eq. (1) contains an implicit parameter that controls the trade-off between "image fit" and regularization. It can be adjusted by scaling the cost- or the distance function.

Interpreting displacements in world coordinates (cf. Section 2.1) yields distance-penalties for locally scaling (i.e. growing or shrinking) the mesh, while parallel translations are not penalized (see Fig. 2c). We consider this beneficial as we expect our initial meshes as well as their local features to have approximately correct scale. Alternatively, if scaling should not be penalized, one could interpret displacements in local coordinate systems per vertex.

2.3. Optimal displacement field

We encode the objective function in Eq. (1) as an MRF, with vertices being represented by MRF-nodes, mesh edges by MRF-edges, and displacements by the possible states (also called *labels*) of the nodes. Cost $\phi(v,s)$ defines the unary potential of node v in state s, and distance $\psi(s_i,s_j)$ defines the binary potential of two adjacent nodes in states s_i, s_j. The MRF-state with minimal sum of potentials yields the desired displacement field. We optimize the MRF energy with a solver that is guaranteed to find an approximately optimal solution (Komodakis et al., 2008). This solver can deal with non-metric distance functions ψ – it solely requires ψ to satisfy $\psi(s_i,s_j) = 0 \Longleftrightarrow s_i = s_j$.

2.4. Refined regularization

The condition $\psi(s_1,s_2) = 0 \Longleftrightarrow s_1 = s_2$ has the effect that there is always a distance penalty for unequal displacements on neighboring vertices. In other words, even the smallest distance between displacements, i.e. the sampling distance δ_S, is penalized if respective displacements occur on neighboring vertices. The sampling distance serves as a scale on which features shall be detected in the image data; in general it is not supposed to determine the amount of regularization imposed upon mesh deformation. A straightforward option to "tolerate" some larger distance between displacements while respecting the condition $\psi(s_i,s_j) = 0 \Longleftrightarrow s_i = s_j$ would be to set the respective distance penalties to a very small value with respect to all others. However, setting very small binary potentials to obtain "almost" unpenalized displacement distances impairs the approximate optimality guarantees of the MRF solver (Komodakis et al., 2008), which depend on the ratio between largest and smallest non-zero binary potential.

Alternatively, we propose to approximate a "tolerated distance" with zero penalty as follows: Let $\widetilde{S} = \{\tilde{s}_i \in \mathbf{R}^3 | i = 1 \cdots n_{\widetilde{S}}\}$ be a second cubic close-packed lattice which is coarser than S, i.e. $\delta_{\widetilde{S}} > \delta_S$. $\widetilde{S}$ partitions S into *displacement blocks* B_i by means of nearest-neighborhood to its elements $\tilde{s}_i$. Formally, $B_i = \left\{ s \in S | \tilde{s}_i = \underset{\tilde{s} \in \widetilde{S}}{\operatorname{argmin}} \|s - \tilde{s}\| \right\}$. Given the displacement blocks, we set up an MRF with states $\tilde{s}_i$ via unary potentials $\tilde{\phi}(v,\tilde{s}_i) = min_{s \in B_i} \phi(v,s)$, and binary potentials $\tilde{\psi}(\tilde{s}_i,\tilde{s}_j) = \psi(\|\tilde{s}_i - \tilde{s}_j\|)$. We optimize the respective MRF energy w.r.t. $\tilde{d} : V \rightarrow \widetilde{S}$ and assign to vertex v with $\tilde{d}_v = \tilde{s}_i$ the displacement $s \in B_i$ with minimum cost, i.e. $d_v = \underset{s \in B_i}{\operatorname{argmin}} \phi(v,s)$.

The sampling distance of $\widetilde{S}$ defines an upper bound to the Euclidean norm of displacement differences that are not penalized. More precisely, with block sampling distance $\delta_{\widetilde{S}}$, zero penalty is attributed to displacements with $\|s_i - s_j\| < \delta_{\widetilde{S}}$ in case s_i and s_j belong to the same block, while the minimum non-zero penalty is attributed to displacements with $\|s_i - s_j\| < 2\delta_{\widetilde{S}}$ in case s_i and s_j belong to adjacent blocks.

Note that the proposed approach allows for the "best" approximative optimality guarantee of the MRF solver (Komodakis et al., 2008) given the desired amount of regularization. Furthermore, memory requirements are significantly reduced as compared to the straightforward approach: The size of the MRF to be solved depends only on the desired amount of regularization, and does not increase in case of refined search space sampling. These advantages come at the cost of distance penalties not only depending on actual displacement distances, but also on displacement block organization.

3. FastODDS

ODDS are designed to allow for accurate segmentations of highly curved structures, while methods that employ unidirectional displacements are fundamentally limited here (cf. Fig. 1). The methodological benefits of ODDS come with the drawback of increased runtime and memory. The required number of sample points per vertex for a ball-shaped range of motion with radius r is in $O(r^3)$, while it is in $O(r)$ for line segments of length $2r$, with corresponding runtime and memory requirements. For instance, given a medium-sized mesh with about 8500 vertices, together with a displacement set S with diameter $2r_S = 15$ mm and sampling distance $\delta_S = 0.4$ ODDS take about 2:30 min to compute on a 3.5 GHz core and require more than 4.5 GB of memory (cf. Table 4).

Unidirectional displacements – apart from the above-mentioned limitations – *do* allow for an accurate segmentation of "flat" structures: At least, the non-visibility problem is unlikely to occur here. Anatomical structures of interest in medical image analysis often exhibit mainly flat or only slightly curved surface regions, while high curvature appears on a much smaller amount of their surface.

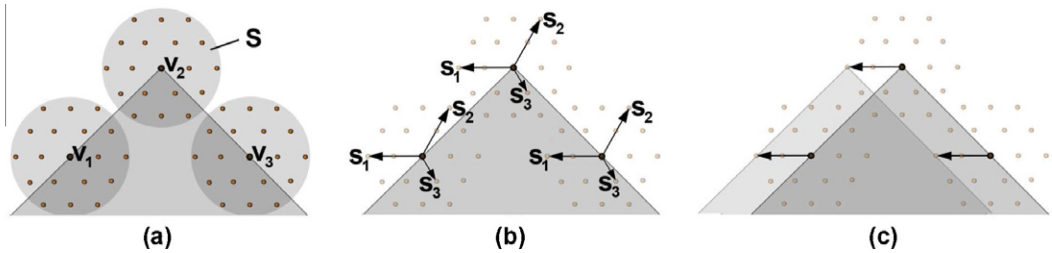

Fig. 2. 2D sketch of omnidirectional displacements: (a) Black dots depict three vertices $v_1,\ v_2,\ v_3$ of a deformable mesh. Ball-shaped ranges of motion S (large grey disks) around each vertex are discretized via sample points (light dots). (b) Exemplary displacements $s_1,\ s_2,\ s_3$ to sample points are shown as black arrows, where equivalent displacements on different vertices are indicated by corresponding numbers. (c) Applying the same displacement to all vertices leads to parallel translation.

Therefore we propose to use omnidirectional displacements only in (and next to) surface regions with high curvature, while employing unidirectional displacements in flat surface regions (see Section 3.2). Thus we exploit the benefits of ODDS, while reducing runtime and memory via an overall reduced amount of sample points. We call this approach *fastODDS*.

We propose to compute unidirectional and omnidirectional displacement sequentially (see Section 3.4). Hence, in general, any method for obtaining unidirectional displacements can be chosen. In this work, we employ the graph cuts based method of Li et al., 2006 (see Section 3.1), because (1) it has proven to be powerful for accurate fine-grain segmentation of medical image data (Yin et al., 2010; Lee et al., 2010; Zhang et al., 2010; Petersen et al., 2011; Seim et al., 2008, 2010), and (2) it allows for simultaneous segmentation of multiple objects (Yin et al., 2010), a property that we call *multi-object ability*. In the following, we refer to this method as *GraphCuts*. Simultaneous segmentation of multiple objects is beneficial in case of low contrast or similar appearance of adjacent objects, e.g. for accurate segmentation of adjacent bones in joints. As described in Section 3.5, the multi-object ability of GraphCuts can be transferred to fastODDS.

3.1. Unidirectional displacements

In contrast to ball-shaped ranges of motion, unidirectional ranges of motion are defined *per vertex* of the deformable mesh. Usually, directions normal to the deformable surface are chosen, but any other predefined directions and generally also curves can be employed. Directions $\ell_v \in \mathbf{R}^3$ with $\|\ell_v\| = 1$ yield respective discretized displacement sets per vertex, $L_v = \{l_i \in \mathbf{R}^3 | i = 1 \cdots n_{L_v}\}$, with lengths $\|l_i\| < r_L$, where r_L is a parameter of the method.

Given unidirectional (sets of) displacements L_v per vertex, we employ GraphCuts (Li et al., 2006) to compute the displacement field with minimum sum of costs subject to local constraints on the difference between the lengths of adjacent displacements. Formally, GraphCuts compute the optimal solution to

$$d = \underset{\{\hat{d}_v : v\in V\}}{\operatorname{argmin}} \sum_{v\in V} \phi(v,\hat{d}_v) \quad \text{subject to } \forall (v,w)\in E: |\ell_v\cdot\hat{d}_v - \ell_w\cdot\hat{d}_w| < c, \tag{2}$$

where $c \in \mathbf{R}$ is a regularization parameter. Here, the (signed) length of displacements must define a total order, i.e. the set(s) of displacements must be one-dimensional.

3.2. Where to use omnidirectional displacements

When defining the surface region where omnidirectional displacements shall be applied (OmniD-region), we assume that we want to achieve a smooth transition to the region where unidirectional displacements shall be applied (UniD-region). Consider e.g. a sharp ridge surrounded by flat surface regions. Imagine we want to translate this ridge in a direction roughly parallel to the flat surface regions. To achieve a smooth overall displacement field, surface-tangential movements cannot be allowed on the boundary of the OmniD-region. Therefore we need to employ omnidirectional displacements not only in the region of high curvature (i.e. on the ridge and in a very small area around it), but within a larger *transition region* around the ridge. Hence, a band of some width around high-curvature regions has to be included in the OmniD-region (see Fig. 3).

We propose to define the OmniD-region as follows: (1) Identify *ridges* on the surface[3]; (2) Identify the surface region that lies within a certain *geodesic distance* g to those ridges. As for the UniD-region, we define it as the complement of the OmniD-region on the surface. Fig. 8a shows OmniD- and UniD-regions on an exemplary anatomical structure.

The geodesic distance threshold g is a parameter of fastODDS. Informally speaking, it should be large enough to allow for the desired amount of displacement of ridge vertices without too much distance penalty. Consider a deformable mesh with mean edge length e_m. Then, g/e_m roughly estimates the number of edges that connect a ridge to the boundary of the OmniD-region. Stretching (or shrinking) each of these edges by one sampling distance δ can reach a translation of the ridge up to $\delta \cdot g/e_m$. In case we can estimate a desired maximum amount of displacement $t \in \mathbf{R}$, we may define $g = e_m \cdot t/\delta$. This way, the desired displacement of ridge vertices can be achieved with no more than the minimum non-zero distance penalty $\psi(\delta)$ at any edge.

3.3. Objective function

We propose to compute displacements for OmniD- and UniD-region with ODDS and GraphCuts, respectively. If not mentioned otherwise, we use the same notation as in Section 2. Let V_U be the vertices in the UniD-region, V_O the ones in the OmniD-region, with $V = V_U \cup V_O$ and $V_U \cap V_O = \emptyset$. The pairs of adjacent vertices E are partitioned into $E_O = (V_O \times V_O) \cap E$, $E_U = (V_U \times V_U) \cap E$ and $E_\partial = (V_O \times V_U) \cap E$. This means E_O contains the edges in the OmniD-region, E_U the edges in the UniD-region, and E_δ the edges that bridge between V_O and V_U. We refer to the set of vertices in the UniD-region which are part of an edge that bridges to the OmniD-region as *UniD-boundary* $\partial V_U = \{w \in V_U, \exists v in V_O : (v,w) \in E_\partial\}$.

We are dealing with two sets of displacements, namely the discretized ball-shaped range of motion S which applies to all vertices in the OmniD-region, and discretized linear ranges of motion L_v along directions ℓ_v per vertex of the UniD-region. We assume for the moment that $\forall v \in V_U : L_v \subset S$ (see Section 3.4). In this context, for ease of notation, we refine the definition of a displacement field to

[3] Ridges may be computed automatically on the initial segmentation (see Appendix B), or, in case a statistical shape model is used for initial segmentation, defined a priori (automatically or manually) on the model.

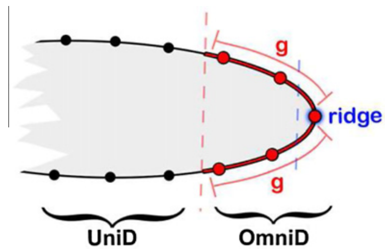

Fig. 3. 2D-sketch of OmniD- and UniD-region on an exemplary tip-like mesh (vertices depicted as dots) with *ridge* at rightmost vertex (blue dot). Only a small region around the ridge exhibits high curvature, as indicated by the light blue, dashed line. Instead, we assign vertices within a certain *geodesic distance g* around the ridge to the OmniD-region; all others belong to the UniD-region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$d : V \to S, \quad v \mapsto d(v) \; =: \; d_v \begin{cases} \in S : v \in V_O \\ \in L_v : v \in V_U \end{cases} \tag{3}$$

Then the objective function of fastODDS is defined as follows:

$$\begin{aligned} d = & \underset{\{\hat{d}_v : v \in V\}}{\operatorname{argmin}} \sum_{v \in V} \phi(v, \hat{d}_v) + \sum_{(v,w) \in E_O \cup E_\partial} \psi(\hat{d}_v, \hat{d}_w) \\ \text{s.t.} \quad & \forall (v,w) \in E_U : |\ell_v \cdot \hat{d}_v - \ell_w \cdot \hat{d}_w| < c \end{aligned} \tag{4}$$

Note that Eq. (4) sums up Eq. (1) on the OmniD-region and Eq. (2) on the UniD-region, and adds to that the distance penalties for edges bridging between OmniD- and UniD-region.

3.4. Optimal displacement field

We follow the simple idea to compute unidirectional and omnidirectional displacements sequentially, in a way that a smooth transition between UniD- and OmniD-region is achieved. Therefore, we first obtain a displacement field for the UniD-region (via GraphCuts or any other method), and second perform ODDS on the OmniD-region, constrained by fixed displacements on the UniD-boundary as computed beforehand.

This approach partitions the objective function in Eq. (4) into two parts that are subsequently solved. First, we compute

$$\begin{aligned} d|_{V_U} = & \underset{\{\hat{d}_v : v \in V_U\}}{\operatorname{argmin}} \sum_{v \in V_U} \phi(v, \hat{d}_v) \\ \text{s.t.} \quad & \forall (v,w) \in E_U : |\ell_v \cdot \hat{d}_v - \ell_w \cdot \hat{d}_w| < c \end{aligned} \tag{5}$$

via GraphCuts. Second, we approximate

$$d|_{V_O} = \underset{\{\hat{d}_v : v \in V_O\}}{\operatorname{argmin}} \left(\sum_{v \in V_O} \phi(v, \hat{d}_v) + \sum_{(v,w) \in E_O} \psi(\hat{d}_v, \hat{d}_w) + \sum_{(v,w) \in E_\partial} \psi(\hat{d}_v, d_w) \right) \tag{6}$$

via MRF optimization. Note that in Eq. (6), $d_w = d|_{V_U}(w)$ is fixed for all $w \in V_U$.

While GraphCuts yield a globally optimal displacement field subject to the given constraints, and MRF optimization guarantees an approximately optimal solution within provable bounds (Komodakis et al., 2008), our hybrid approach for solving Eq. (4) does not guarantee either of these global properties. The optimality bounds guaranteed for MRF optimization would determine bounds for the overall objective function in case we minimized Eq. (4) with respect to the set of displacement fields on the UniD-boundary for which a solution to Eq. (5) exists. This would require solving Eqs. (5) and (6) for all feasible displacement fields on the UniD-boundary. Their number is in $O(c/\delta \cdot |\partial V_U|)$. As this is roughly $3 \cdot 1000$ in our experiments (cf. Section 4), we did not follow this approach for performance reasons.

Practically, to get a "good" solution on the UniD-boundary, we perform GraphCuts on the whole surface except vertices on or next to ridges. The overlap with the OmniD-region serves for an extended regularization in GraphCuts optimization; the resulting displacements are discarded afterwards. Note that "cutting the surface open" along ridges allows for translational movements of surface regions near ridges with GraphCuts at least in one (surface-normal) direction. Otherwise, moving the surface "inward" one side of a ridge and "outward" on the opposite side may not be possible due to regularization.

As for the OmniD-region, practically, we enlarge it by the UniD-boundary, and achieve fixed displacements for each boundary vertex $w \in \partial V_U$ by assigning zero cost $\phi(w,s)$ to $s = d|_U(w)$ and infinite cost to all other displacements $s \in S, \; s \neq d|_{V_U}(w)$. More precisely, as unidirectional displacements are generally *not* in S due to discretization, we assign zero cost to the *closest* displacement in S, i.e. $\operatorname{argmin}_{s \in S} \|d|_{V_U}(w) - s\|$, and infinite cost to all others.

3.5. Multi-object FastODDS

GraphCuts can be used for simultaneous segmentation of multiple objects (Yin et al., 2010) via shared displacement directions for arbitrary adjacent structures (Kainmueller et al., 2009b). Hard constraints on the distance between adjacent surfaces can be enforced. To transfer this capability to fastODDS, we use multiple surfaces that are coupled with shared displacement directions in adjacent regions as input, and partition each surface into OmniD- and UniD-region as for single-object fastODDS. Then, we apply multi-object GraphCuts on the (coupled) UniD-regions. Subsequently, we apply ODDS on the OmniD-regions as for single-object fastODDS (cf. 6), i.e. constrained by fixed displacements on the UniD-boundary as computed beforehand, here via multi-object GraphCuts. This way, fastODDS can handle multi-object situations in case adjacent surface regions are, at least to some extent, flat, and hence equipped with linear range of motion.

If, however, the coupled region exhibits high curvature, it may overlap with the OmniD-region. Consequently, the resulting deformed surface may intersect with the adjacent surface. This can be prevented in case we know beforehand that one of the adjacent surfaces does not exhibit high curvature. In this case, the multi-object GraphCuts result on the "flat" surface can be used to modify the cost function on the OmniD-region of the "curved" surface such that no overlap can happen. This can be achieved by setting costs to infinite for all sample points that lie inside the deformed "flat" surface. However, in case both adjacent surfaces exhibit high curvature within the coupled region, multi-object fastODDS do not guarantee non-overlapping results.

4. Results

To evaluate ODDS, we applied it to three types of 3D data: (1) Synthetic binary images, (2) synthetic binary images with various amount of noise, and (3) clinical image data. To evaluate fastODDS, we applied it to two cohorts of clinical image data: (1) 106 CBCTs of the mandibular bone – with the coronoid process as an exemplary tip-like structure – to assess the differences to ODDS, and (2) 49 CTs of the hip bones – with the acetabular rim as an exemplary rim-like structure in a multi-bone environment – to assess the multi-object ability of fastODDS.

On synthetic binary images and clinical image data, we also computed results with GraphCuts (Li et al., 2006), as well as repeated, i.e. *iterative* GraphCuts (*iGraphCuts*). Furthermore we compute results with an iterative, locally regularized method we refer to as *FreeForm* (Kainmueller et al., 2007). FreeForm selects the minimum cost displacement for each vertex, truncates it to some maximum length (i.e. "stepsize"), applies it and subsequently regularizes locally via a small displacement toward the centroid of the respective adjacent vertices.

We computed costs $\phi(v,s) \in \mathbf{R}$ from the image $I : \mathbf{R}^3 \to \mathbf{R}$ as proposed by Seim et al., 2008: If the image intensity $I(v+s)$ lies within a certain window $[i_0, i_1]$, and the directional image derivative $\nabla_{n_v} I(v+s)$ along the surface normal n_v at vertex v is negative and its absolute value exceeds a certain threshold j, the cost $\phi(v,s)$ is inversely proportional to $\|\nabla_{n_v} I(v+s)\|$.[4] Otherwise costs are set to a constant, high value ϕ_{high}. The thresholds i_0, i_1 and j are parameters of the strategy and are set per application (see Table 1).

As for the trade-off between image fit and regularization, we scale the cost function such that $a \cdot \psi(\delta_{\widetilde{S}}) < (\phi_{high} - \phi(v,s)) < a \cdot \psi(2\delta_{\widetilde{S}})$ for any cost $\phi(v,s) < \phi_{high}$, where $a = 6$ is the average number of edges per vertex. Image derivatives are scaled such that $0 < \phi(v,s) < 0.5 \cdot \phi_{high}$ for any $\phi(v,s) < \phi_{high}$. This serves for a clear distinction of features from "non-features", but also leaves room to distinguish between the quality of features, i.e. there exist $\phi_1 < \phi_2 < \phi_{high}$ with $a \cdot \psi(\delta_{\widetilde{S}}) < (\phi_2 - \phi_1)$.

Whenever we compare different adaptation methods on the same image data, we use the same cost function ϕ for all methods.

For all omnidirectional displacements (ODDS and fastODDS), we employ displacement blocks with sampling distance $\delta_{\widetilde{S}} = 3\delta_S$; as distance function ψ, we use $\psi(s_1, s_2) = \|(s_2 - s_1)/\delta_{\widetilde{S}}\|^3$ in all experiments. In all GraphCuts experiments ("pure", iterative, as well as in UniD-regions of FastODDS), the regularization parameter c equals the block sampling distance $\delta_{\widetilde{S}}$ as set in the respective ODDS/fastODDS experiment, i.e. $c = \delta_{\widetilde{S}}$.

Whenever we employ unidirectional as well as omnidirectional displacements for the same image data (in multiple methods), the length of the unidirectional range of motion equals the respective ball diameter, i.e. $r_L = r_S$. As for the sampling distance of unidirectional displacements, we set it to half the sampling distance of the respective omnidirectional displacements, i.e. $\delta_{L_v} = 0.5\delta_S$. Whenever we employ fastODDS, we detect ridges automatically as described in Appendix B with significance $0.04\ \mathrm{mm}^{-1}$ and curvature threshold $0.1\ \mathrm{mm}^{-1}$. In contrast to GraphCuts and ODDS/fastODDS, all FreeForm and iGraphCuts adaptations were performed iteratively, with 30 steps.

Table 1 lists the values of application specific parameters.

4.1. Synthetic images

We performed experiments on binary images[5] of a cube and a thin ellipsoid. As initial meshes, we used triangulated cube and tip surfaces with ideal shape, but shifted pose (see Fig. 4a). We chose ball diameters such that the target object boundary was located completely within a band of respective width around the initial mesh. Fig. 4b shows the results of adding normal displacements without any regularization. The results of FreeForm-, GraphCuts- and ODDS adaptation are shown in Fig. 4c, d and e, respectively.

We added various amounts of random noise to the binary cube image and performed ODDS as before. The cube was detected correctly for noise with ranges $[-0.5 \cdots 0.5]$ and $[-2.5 \cdots 2.5]$, and failed for $[-5 \cdots 5]$. Fig. 5 shows slices of the noisy image data and the respective adaptation results.

4.2. Clinical data

4.2.1. Mandible (coronoid process)

In a quantitative evaluation on 106 mandible Cone-Beam CTs with voxel size $0.3^3\ \mathrm{mm}^3$ we compared ODDS, fastODDS, FreeForm, iGraphCuts and GraphCuts results to gold standard surfaces obtained from manual segmentations. Initial meshes were generated automatically by adaptation of a statistical shape model (SSM) (Kainmueller et al., 2009a).

For all omnidirectional displacements, we gave slight preference to displacements that point further "outwards" in curvature gradient direction $\nabla k_1(v)$, where k_1 is the first principal curvature of the deformable surface. (Note that $\nabla k_1(v)$ is perpendicular to the surface normal at v.) We achieve this preference by adding to $\phi(v,s)$ a small cost proportional to $r_S - s \cdot \nabla k_1(v)/\|\nabla k_1(v)\|$. To also exploit information about curvature gradient for all unidirectional displacements, we do not use normal displacements, but rotate displacement directions from surface normals towards curvature gradient direction. We found that a rotation angle of 45 degrees yields the best results for the mandible. By considering the curvature gradient of the mesh, we intend to find "better" points on the sharp, tip-like target structure in the sense of *anatomical correspondence*, and hence reduce the amount of mesh distortion necessary for accurate segmentation of the tip.

For both gold standard as well as automatically determined mandible surfaces, we extracted the right coronoid processes as the region of the mesh that lies above 1/2 of the extension of the mandible in transversal direction, between 1/3 and 2/3 of extension in dorsoventral direction, and above 2/3 in longitudinal direction. Extraction of the left coronoid process worked analogously. We identified the tip point as the upmost vertex in longitudinal direction. See Fig. 6 for an exemplary mandibular bone anatomy.

As error measures for the coronoid process, we assessed the tip-to-tip distances (tip2tip), tip-to-surface distances (tip2surf), and Hausdorff (max) surface distances, as well as the percentage of two-sided surface distances above 1.2 mm (%>1.2 mm). Evaluation results are shown in Table 2 and Fig. 7. As measurements are not normally distributed, we performed Wilcoxon's signed-rank test (Hollander and Wolfe, 1999) to assess the significance of differences between methods.

4.2.2. Hip bones (acetabular rim)

In a quantitative evaluation on 49 hip CTs with voxel size $0.9 \times 0.9 \times 1\ \mathrm{mm}^3$ we compared fastODDS and GraphCuts results to gold standard surfaces obtained from manual segmentations. Initial meshes were generated automatically by adaptation of an articulated statistical shape model (ASSM) of hip bones and femur (Kainmueller et al., 2009c). In case of omnidirectional displacements, we gave slight preference to displacements in surface curvature gradient direction, as described before for the mandible (cf. Section 4.2.1).

Again, we experimented with unidirectional displacements rotated from surface normals towards curvature gradient directions to also approfit from curvature gradient information – however, we found that for the hip bones, this does not improve accuracy. We attribute this to the more complex shape of the hip bones, which exhibit concave and convex structures in close proximity, and do not show such a sharp and long tip as the mandibular coronoid process. Consequently, we stick to normal displacements here.

For an unbiased, reproducible delineation of the acetabular rim, we computed it *automatically* as described in Appendix A on both gold standard segmentations and adaptation results. As error measures for the acetabular rim, we assessed the Hausdorff (max) *curve distance* as well as the percentage of distance above 1.5 mm ($\% > 1.5$ mm). Furthermore, we assessed the max *surface distance* as well as the $\% > 1.5$ mm measure for the whole hip bones.

Evaluation results for both acetabular rim and whole hip bone are shown in Table 3. As for the mandible, error measures are not normally distributed, and hence we performed Wilcoxon's

[4] This means that for the computation of costs at a vertex v, image derivatives are assessed in the *same* direction for all displacements s, namely in direction of the surface normal n_v at v, no matter which type of displacements are employed (unidirectional or omnidirectional).

[5] i.e. intensities $\in \{0, 1\}$.

Table 1
Application specific parameters are the number of vertices $\#V$ of the deformable mesh (which determines the average edge length of mesh triangles el (mm)), the diameter $2r$ (mm) of the range of motion, the sampling distance δ_S (mm) of the set of displacements, the number of sample points $\#S$ and MRF labels $\#\widetilde{S}$ resulting for ODDS/fastODDS, the intensity window $[i_0, i_1]$ of the cost function, the gradient magnitude threshold j (1/mm) and filter length n_f (in number of edges) and geodesic distance g (mm) for definition of the OmniD-region.

	$\#V$	el	$2r$	δ_S	$\#S$	$\#\widetilde{S}$	$[i_0, i_1]$	j	n_f	g
Cube	770	1	26	0.5	105294	3768	[0.1,1.1]	0.1	–	–
Ellipsoid	1797	1	31	0.5	178201	6989	[0.1,1.1]	0.1	–	–
Mandible	8561	1.2	15	0.4	41272	2188	[350,800]	75	6	6
Hip bone	14008	2.1	20	0.5	48078	1714	[120,720]	25	10	10

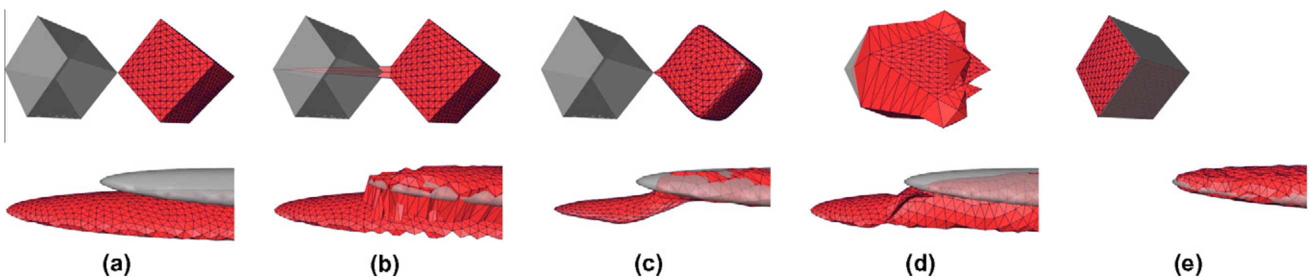

Fig. 4. Results on synthetic data. Deformable mesh (red/dark grey mesh) and target object (transparent grey surface) are shown (a) in their initial situation, and after deformation via (b) displacements along normals without regularization, (c) FreeForm, (d) GraphCuts, and (e) ODDS. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

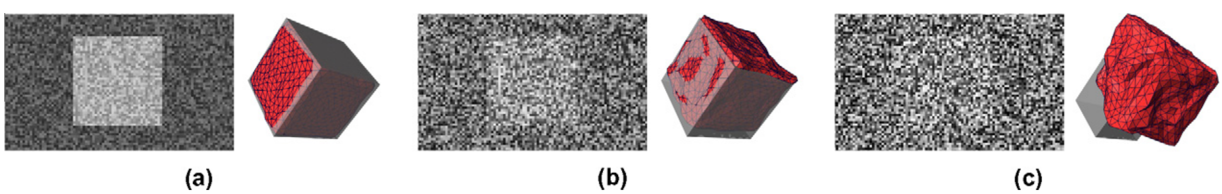

Fig. 5. Performance of ODDS in the presence of noise. We added random noise with range (a) [-0.5..0.5], (b) [-2.5..2.5] and (c) [-5..5] to a binary image of a cube. We show slices of the image data and the respective adaptation result (red/dark grey mesh). The grey transparent surface depicts the ideal target object. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

signed-rank test to assess significant differences. Additionally, Fig. 8b and c shows the averaged directional distance maps, i.e. gold standard-to-result and result-to-gold standard distances, respectively.

4.2.3. Performance

All experiments were performed on a single 3 GHz core with 8 GB main memory. Table 4 lists the average performance of all methods applied to clinical data.

MRF optimization (Komodakis et al., 2008) took between one and six seconds in all ODDS- and fastODDS experiments. Computation of the cost function $\phi(v, l)$ was more time-consuming, accounting for more than 90% of the runtime of ODDS and fastODDS as stated in Table 4. FastODDS and iterative GraphCuts have comparable runtime.

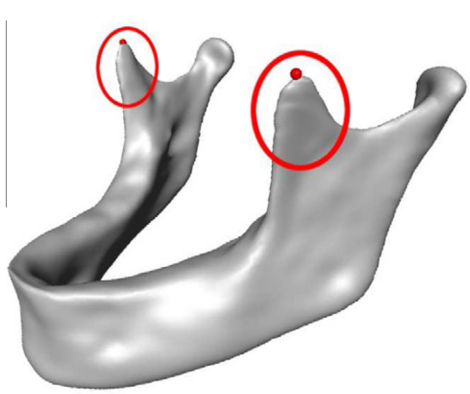

Fig. 6. Exemplary mandibular bone anatomy. Red circles: Left and right coronoid process. Red dots: Respective tip points.

5. Discussion

5.1. Accuracy

Experiments on synthetic binary images show that ODDS are able to handle parallel translations of highly curved surface regions, in contrast to conventional free deformation approaches (GraphCuts and FreeForm) that employ normal displacements. While global deformation models (i.e. rigid) may also yield the desired vertex displacements in the experiments we present, ODDS achieves them with *free* deformations.

Experiments on noisy synthetic images show that ODDS are able to produce well-regularized displacement fields in the presence of noise. However, for a very low signal-to-noise ratio, ODDS may fail to detect the target object.

Experiments on Cone-Beam CTs of the mandible show that both ODDS and fastODDS are able to produce very accurate segmentations of tip-like structures. On 212 mandibular coronoid processes, ODDS and fastODDS clearly outperform the GraphCuts, and FreeForm approach. Here, normal displacements often exhibit the visibility problem. Fig. 9a–c shows exemplary results.

A comparison of ODDS and fastODDS on the mandibular coronoid processes reveals no significant differences for any error

Table 2
Top to bottom: Average error measures (and standard deviation) for initial SSM adaptation (SSM), FreeForm, iterative GraphCuts (itGraphCuts), GraphCuts, fastODDS and ODDS results on 212 coronoid processes and 106 entire mandibles, followed by differences $A-B$ of average error measures for $A, B \in \{\text{GraphCuts}, \text{fastODDS}, \text{ODDS}\}$, together with significance levels of difference (*p*-values) as assessed with Wilcoxon's signed rank test. A positive *p*-value indicates that *B* has lower error than *A* (at the respective level of significance), while a negative sign indicates that *A* has lower error than *B*. Significance levels below 5% are marked with a star (*), below 1% with two stars (**), and below 0.1% with three stars (***).

Coronoid process	tip2tip (mm)	tip2surf (mm)	max (mm)	%> 1.2 mm (%)
SSM	**2.44**(2.10)	**2.12**(2.15)	**2.76**(2.16)	**4.48**(7.12)
FreeForm	**1.72**(2.00)	**1.43**(2.02)	**1.98**(1.98)	**1.51**(4.04)
GraphCuts	**1.79**(2.08)	**1.69**(2.17)	**2.22**(2.18)	**2.30**(5.39)
itGraphCuts	**1.66**(2.02)	**1.32**(2.02)	**2.07**(2.01)	**1.83**(4.62)
fastODDS	**1.38**(1.69)	**1.05**(1.66)	**1.70**(1.76)	**1.23**(3.74)
ODDS	**1.35**(1.52)	**1.03**(1.51)	**1.68**(1.59)	**1.17**(3.39)
FreeForm-ODDS	**0.37**	**0.40**	**0.20**	**0.34**
p-Value [%]	<0.01***	<0.01***	0.02***	0.51**
FreeForm-fastODDS	**0.34**	**0.38**	**0.18**	**0.28**
p-Value (%)	0.03***	<0.01***	0.04***	0.20**
itGraphCuts-ODDS	**0.31**	**0.29**	**0.38**	**0.66**
p-Value (%)	<0.01***	1.27*	<0.01***	<0.01***
itGraphCuts-fastODDS	**0.28**	**0.27**	**0.37**	**0.60**
p-Value (%)	0.72**	3.07*	<0.01***	<0.01***
fastODDS-ODDS	**0.03**	**0.02**	**0.02**	**0.06**
p-Value (%)	31.41	46.00	23.77	−43.05

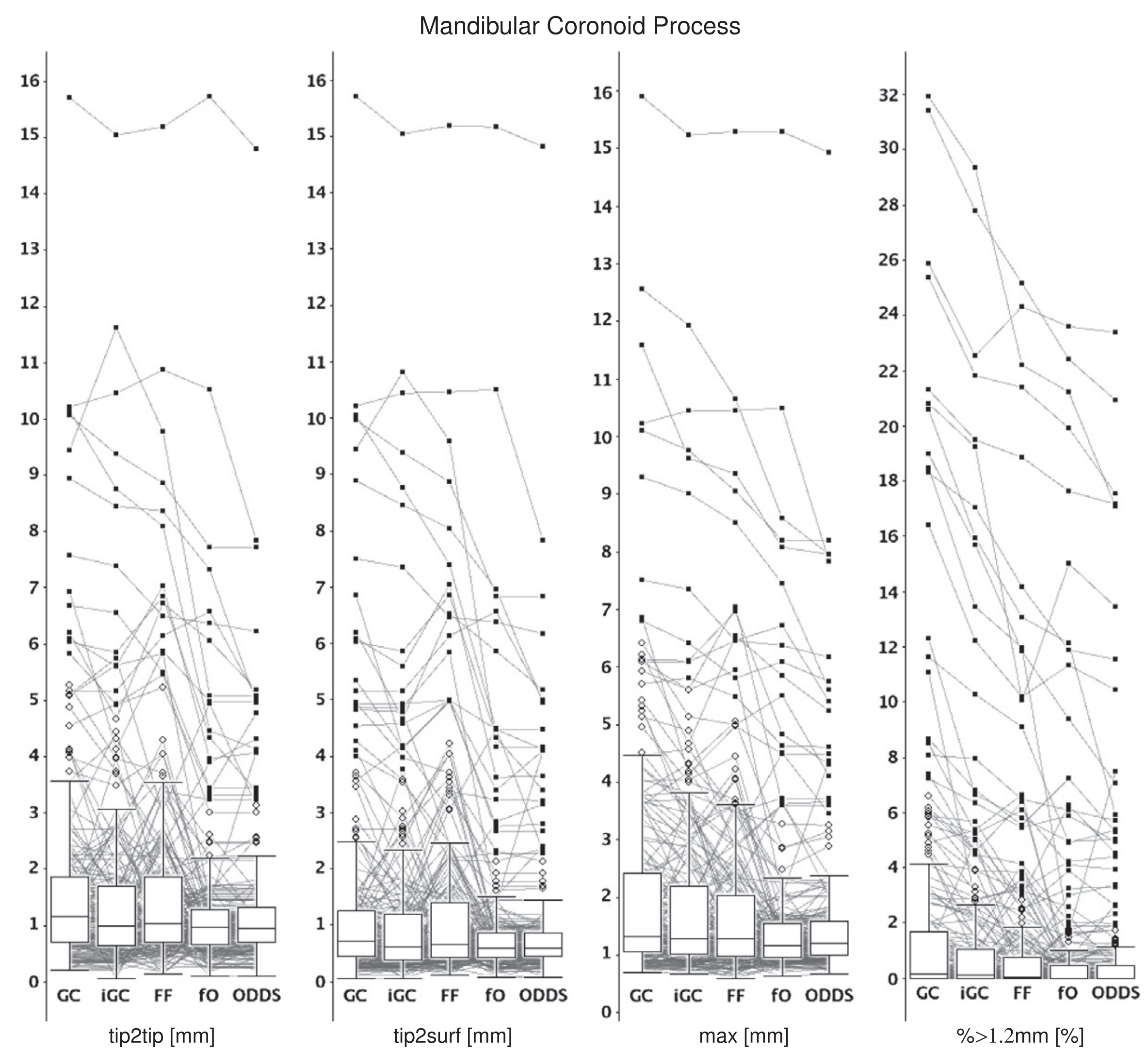

Fig. 7. Box plots (with outliers as circles and extreme outliers as dots; see Chambers, 1983) of error measures for GraphCuts (GC), iterative GraphCuts (iGC), FreeForm (FF), fastODDS (fO) and ODDS results on coronoid processes as listed in Table 2. Underlaid parallel coordinate plots draw lines between errors measured for different methods (GC, iGC, FF, fO, ODDS) on corresponding individual cases, e.g. between the tip2tip errors of fastODDS- and ODDS-result on coronoid process no. 189, etc.

Table 3
Top: Average error measures (and standard deviation) for initial ASSM adaptation as well as GraphCuts and FastODDS results on 98 acetabular rims and 98 hip bones. Bottom: Differences of average errors and respective levels of significance (p-values) as assessed with Wilcoxon's signed rank test. Significance levels below 5% are marked with a star (*), below 1% with two stars (**), and below 0.1% with three stars (***).

	Acetabular rim		Hip bone	
	max (mm)	%>1.5 mm (%)	max (mm)	%>1.5 mm (%)
ASSM	**5.95**(2.53)	**66.88**(17.64)	**8.44**(2.53)	**26.74**(7.80)
GraphCuts	**5.00**(2.53)	**36.61**(15.98)	**7.07**(2.36)	**2.39**(1.79)
fastODDS	**4.69**(2.75)	**22.91**(15.34)	**6.92**(2.40)	**1.88**(1.65)
GraphCuts-fastODDS	**0.32**	**13.70**	**0.15**	**0.52**
p-Value (%)	1.04*	<0.01***	5.58	<0.01***

Table 4
Performance (computation time in seconds/ maximum memory requirement in GB) for ODDS, fasODDS (fO), GraphCuts (GC), iterative GraphCuts (itGC) and FreeForm (FF) averaged for 106 mandibles and 98 hip bones.

[sec/ GB]	ODDS	fO	GC	iGC	FF
Mandible	149/ 4.6	85/ 2.2	3/ 0.9	90/ 0.9	6/ 0.4
Hip bone	–	319/ 5.4	18/ 2.3	–	–

measure. However, the parallel-coordinate plots that underlay the box plots in Fig. 7 show that there are some individual cases with considerable differences between ODDS and fastODDS error measures. We conclude that fastODDS is not guaranteed to produce equally accurate results in the individual case, but overall does not perform significantly different than ODDS.

As for the whole mandible surface, we found an evaluation of error measures to be "overshadowed" by large regions on the initial segmentation that are too far away from the target structure such that it does not lie within either ball-shaped or linear search space. These are regions that exhibit misleading or missing features – the best one can do is to keep them at their initial position, i.e. not deform them at all after initial SSM-based segmentation. This holds for the region around the teeth, where the teeth themselves are potentially mistaken as features, and also the chin, which often lies outside the FoV of the CBCT scanner and hence exhibits no features at all. Therefore, we only state the Hausdorff error measure here (Table 5) for reasons of completeness. Note that differences are not significant for *any* couple of methods, including the initial SSM-based segmentation. However, the average errors suggest a slight tendency in favour of ODDS/fastODDS.

Experiments on CTs of the pelvis show that fastODDS are able to produce very accurate segmentation of ridge-like structures in a multi-object environment. On 98 acetabular rims of the hip bones, multi-object fastODDS clearly outperform the multi-object GraphCuts approach. Here, unidirectional displacements often struggle with restricted visibility, see Fig. 10a and b, respectively.

FastODDS also performs better in terms of error measures evaluated on the whole hip bones. As for the Hausdorff error measure, the relatively small improvement from initial (SSM-based) to resulting (GraphCuts and fastODDS) segmentation suggests that the *p*-value stemming from the comparison of GraphCuts and fastODDS, 5.58%, may, as for the mandible but less prominent, be influenced by regions on the initial segmentation that are too far away from the target structure to be within reach of either ball-shaped or linear search range. This observation has also been reported by Seim et al. (2008).

5.2. Comparability of regularization

For GraphCuts, differences of displacement lengths on neighboring vertices are "for free" up to *c*, while larger differences are impossible. For ODDS, differences of displacements on neighboring vertices are "for free" or cost the minimum non-zero distance penalty up to a Euclidean norm of $\delta_{\tilde{S}}$, while the penalty increases cubically for larger differences (see Section 2.3). FreeForm regularizes the surface mesh and not displacements themselves – however, the maximum edge length that can occur on a needle-shaped (i.e. infinitesimally thin tip) mesh is bounded by displacement stepsize and internal smoothing weight.

To achieve comparable regularization, we set the regularization parameter *c* of GraphCuts to the displacement block sampling distance $\delta_{\tilde{S}}$ as set in the respective ODDS/fastODDS experiment. Furthermore, we parametrized FreeForm such that when stretching a needle-like tip, the maximum achievable edge length is double the average initial edge length *el*, and hence the maximum difference of displacements is *el*, where $el = c$. In summary, $\delta\tilde{S} = c = el$ in all experiments.

We think this allows for a fair comparison of methods. However, to make sure that the superior accuracy of ODDS/fastODDS is not an effect of "more or less" regularization, we performed GraphCuts not only with $c = \delta_{\tilde{S}}$, but with *c* ranging from the sampling distance δ_S up to an absurdly large $10\delta_S = 5$ mm in nine extra experiments on the hip bones. Considering segmentation accuracy, significant improvements of fastODDS over GraphCuts as stated via colored entries in Table 3 hold for *any* of the respective GraphCuts results.

Cutting the sampling distance by half, i.e. $\delta_{L_v} = 0.5\delta_S$, was intended to compensate for a potential advantage of omnidirectional

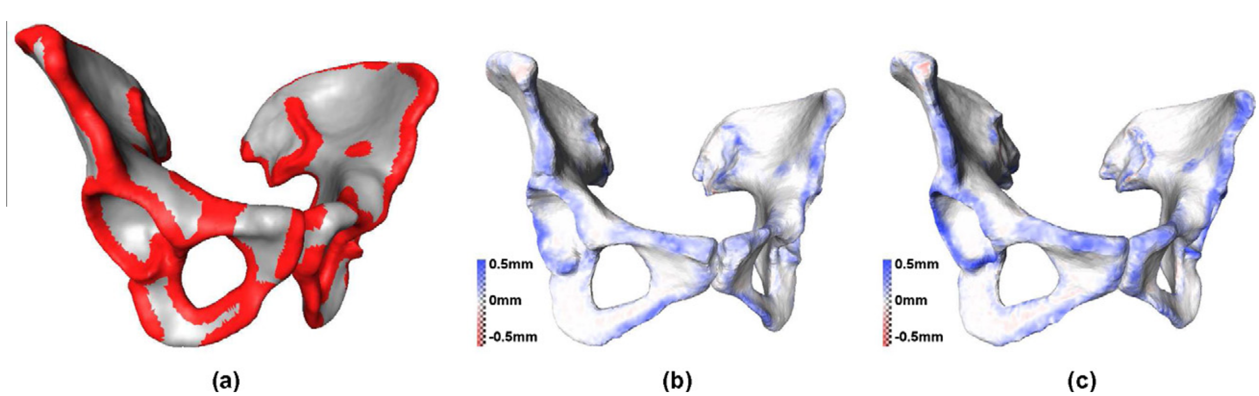

Fig. 8. FastODDS on hip bones. (a) Exemplary hip bones with OmniD-region (red) and UniD-region (grey). (b and c) Comparison to GraphCuts: *Differences* of directional surface distances (GraphCuts-FastODDS) from/to gold standard averaged over 49 cases. (b) Difference of distances from gold-standard to results. (c) Difference of distances from results to gold-standard. On average, fastODDS perform better than GraphCuts in blue regions, while GraphCuts perform better than fastODDS in red regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

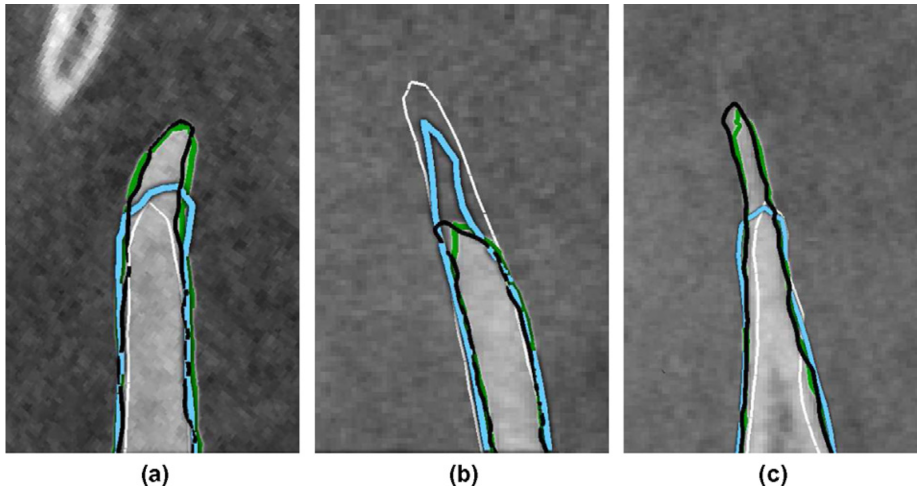

Fig. 9. ODDS: Exemplary results on clinical data: Coronoid processes of the mandible. Contours – Black: gold standard. White: initial mesh. Green/gray: ODDS result. Blue/light gray: FreeForm result. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Average Hausdorff error assessed on 106 mandible surfaces for initial SSM-based segmentation, FreeForm (FF), GraphCuts (GC), iterative GraphCuts (iGC), fastODDS (fO) and ODDS.

SSM	FF	GC	iGC	fO	ODDS
7.36 (3.03)	**7.30** (2.99)	**7.36** (3.01)	**7.33** (3.00)	**7.14** (2.71)	**7.14** (2.69)

displacements in terms of an effective denser sampling in surface-normal direction due to additional adjacent sampling points. It *did* slightly improve error measures for GraphCuts results – however, the accuracy of ODDS/fastODDS could not be reached, not even with still smaller (nor bigger) sampling distances from 0.25 to $1\delta_S$.

5.3. Mesh resolution

The flexibility of the deformable mesh is determined by regularization, namely the "tolerated distance" δ_{tol} between neighboring displacements as set via regularization parameters (cf. Section 5.2), and mesh resolution, namely the average edge length *el* of mesh triangles. Tolerated distance, divided by triangle edge length, δ_{tol}/el, serves as a measure for mesh flexibility. However, coarser mesh resolution at constant mesh flexibility in terms of δ_{tol}/el has a smoothing effect on the displacement field. To this end, to assess the influence of mesh resolution at constant flexibility $\delta_{tol}/el = 1$ as set in our original experiments (see Table 1), we performed an additional evaluation of fastODDS, iterative GraphCuts and FreeForm on a series of different mesh resolutions. We achieve different resolutions with an approach for isotropic remeshing described by Surazhsky and Gotsman, 2003. The resulting Hausdorff distances on 212 mandibular coronoid processes are plotted in Fig. 11. For fastODDS, we did not evaluate the finest mesh resolution due to unbearable memory requirements ($\geqslant 64$ *GB*). The significance statements given in Table 2 hold for *any* of the examined resolutions, and furthermore also when comparing fastODDS at 1.2 mm edge length to iterative GraphCuts and FreeForm at the smaller edge length of 0.6 mm.

5.3.1. Mesh consistency

Avoiding self-intersections of the deformable mesh is crucial for approaches that employ unidirectional displacements together with iterative feature search (FreeForm, iGraphCuts). This is because "loops" in the deformable mesh can invert surface normals and hence render successive search directions unfeasible, if not counterproductive. To this end, we prevent self-intersections with the method proposed in (Kainmueller et al., 2007), and hence no self-intesections occur (while consequently the deformable mesh may "get stuck"). Instead, for approaches that perform feature search just once (ODDS, fastODDS, GraphCuts), self-intersections of the deformable mesh do not necessarily affect segmentation accuracy. However, they do indicate some sort of quality of mesh deformation. To this end, we assessed the number of self-intersections for the mandibular coronoid process and hip bone results as

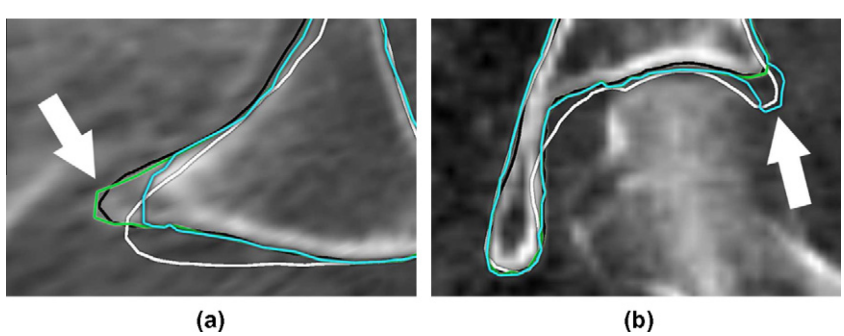

Fig. 10. FastODDS: Exemplary results on clinical data: Acetabular rim of the pelvis. Contours: Black: Gold standard. White: Initial mesh. Green: FastODDS result. Blue: GraphCuts result. FastODDS works nicely while GraphCuts do not reach the corresponding image features that are located (a) in outward direction and (b) in inward direction from the initial mesh. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

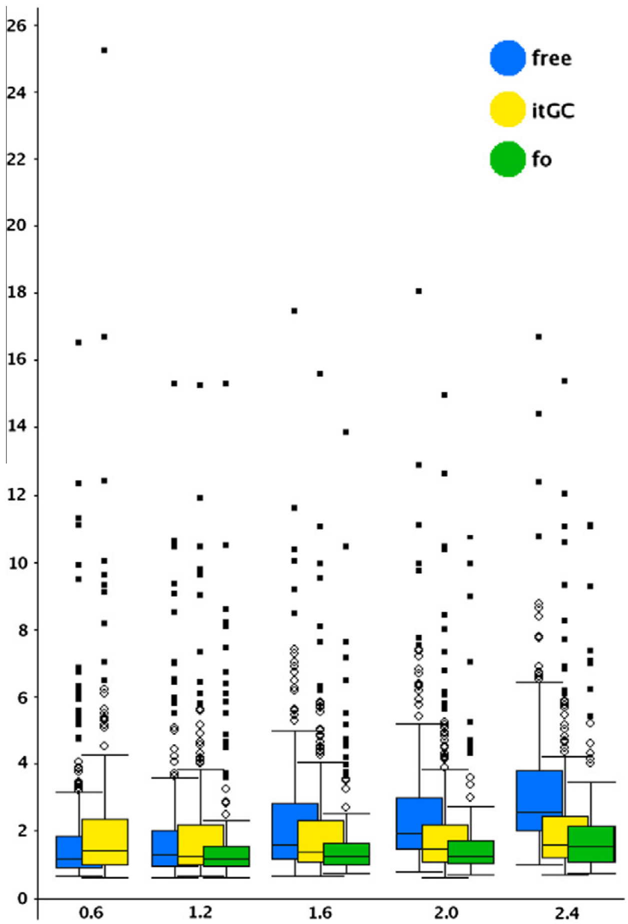

Fig. 11. Comparison of fastODDS, iGraphCuts and FreeForm at a series of mesh resolutions. *X*-axis: Average edge length (mm) of mesh triangles. *Y*-axis: Hausdorff distance (mm) assessed for 212 coronoid processes of the mandible. Mandible meshes contain about 34,100, the original 8561, 5000, 3100 and 2300 vertices, respectively.

presented in Sections 4.2.1 and 4.2.2, respectively. Each triangle edge that intersects with a non-adjacent triangle counts as a self-intersection. Table 6 lists the results. While fastODDS and ODDS exhibit more self-intersections than GraphCuts on the mandibular coronoid processes, fastODDS produces slightly less self-intersections than GraphCuts on the hip bones.

5.4. Performance

A comparison of ODDS and fastODDS on the mandibular bone shows that fastODDS requires less than half the memory, while being almost twice as fast as ODDS. The runtime of fastODDS is comparable to iterative GraphCuts. In general, the gain in performance achieved by fastODDS depends on the "curvedness" of the anatomical structure of interest. Hence we hypothesize that the gain is even bigger for structures like the heart or the liver, where a higher percentage of the structure exhibits low curvature, while it may be little to none on highly folded structures like the cerebral cortex or the intestinal mucosa.

Table 6
Average number of self-intersections in deformed surface mesh, assessed for ODDS, fastODDS and GraphCuts on 212 mandibular coronoid processes, and for fastODDS and GraphCuts on 98 hip bones. Standard deviation in brackets.

	ODDS	fastODDS	GraphCuts
Coronoid process	**30.8** (29.2)	**22.2** (23.1)	**7.6** (17.4)
Hip bone	–	**305.81** (140.28)	**311.70** (163.07)

Table 7
Automatic acetabular rim delineation: Average root mean square (rms) and Hausdorff (max) distance from manually defined landmarks, and percentage of distance above 1 mm (%>1 mm), assessed on 147 hip bone surfaces. Standard deviations in brackets.

rms (mm)	max (mm)	%>1 mm (%)
1.21(0.31)	**3.06**(0.90)	**34.84**(12.37)

6. Conclusion

We proposed ODDS, a method that allows omnidirectional displacements for all vertices of a surface mesh during deformable model adaptation. We encode the adaptation problem as a Markov Random Field, which allows us to approximate globally optimal mesh deformation subject to local regularization constraints. In an evaluation on synthetic as well as clinical data, we showed that this approach can outperform traditional mesh adaptation along line segments (e.g. surface normals) in regions with high curvature (tips and ridges) in terms of segmentation accuracy.

To save runtime and memory as required by ODDS, we developed a hybrid approach, fastODDS. Here, we employ omnidirectional displacements adaptively, i.e. only where high curvature calls for them, and traditional unidirectional displacements elsewhere. In an evaluation on clinical data we showed that fastODDS achieve the same segmentation accuracy as ODDS in regions of high curvature, while requiring only half the runtime and memory.

An additional benefit of fastODDS is that it can be applied for simultaneous adaptation of multiple, adjacent meshes, i.e. multi-object segmentation. In an evaluation on clinical data we showed that fastODDS can outperform traditional multi-object mesh adaptation along line segments.

Future work will focus on a more efficient computation of the image cost function ϕ via parallelization and exploitation of overlapping domains.

Acknowledgments

D. Kainmueller is funded by the DFG Collaborative Research Center SFB760. H. Lamecker is funded by the German Research Center MATHEON. Thanks to Max Zinser (Universiätsklinikum Köln, Germany) for providing image data and manual segmentations of the mandible. Thanks to Jana Malinowski (Zuse Institute Berlin, Germany) for manually segmenting the hip bones. Thanks to Alexander Wurl (Julius Wolff Institute, Charité – Universitätsmedizin Berlin, Germany) for manually delineating the acetabular rims.

Appendix A. Automatic acetabular rim detection

The statistical shape model of the hip bones we employ for initial segmentation contains a particular region (aka *patch*) that defines the acetabulum, cf. Seim et al., 2008. Consequently, this acetabular patch is inherent on every initial segmentation, and is preserved during deformation with any of the adaptation methods we employ in this work. The boundary of the deformed acetabular patch serves as an initial estimate of the acetabular rim. It is represented by a set of vertices that are connected by edges which form a closed contour. Starting from this initial estimate, the algorithm for automatic detection of the acetabular rim proceeds as follows.

(1) Define an approximate "rim-plane" via plane-fit to the initial acetabular rim estimate. (2) For each vertex on the initial rim estimate, sample a set of points on the hip bone surface in direction perpendicular to the rim within some geodesic distance. (3) Define a *cost* per sample point as the signed distance from the approximate rim-plane in outward[6] direction. (4) Construct a graph: For every pair of neighboring vertices, connect corresponding sample points by and edge in the graph; Connect sample points +- the corresponding one to achieve the desired amount of regularization; (5) Perform Dijkstra's algorithm (Dijkstra, 1959) to obtain the minimum-cost rim. The result serves as automatically detected acetabular rim.

We evaluated automatic rim delineation vs. manually defined landmarks on 147 hip bone surfaces stemming from manual and automatic segmentation results. Resulting error measures are listed in Table 7.

Appendix B. Automatic ridge detection

For ridge detection on surfaces, we utilize the ridge definition first introduced by Rothe (1915), more recently described by Koenderink and van Doorn (1993). Intuitively a ridge of a height function can be imagined as the way one would take when walking up a mountain. One usually chooses the path with the lowest slope since it is the least exhausting. We apply this definition for ridges to the maximum principal curvature on surfaces as height function, i.e. $\kappa = \max(|\kappa_1|, |\kappa_2|)$, yielding curves along sharp edges as well as sharp wrinkles of a surface.

At first sight, the above ridge definition requires computing the fifth derivative of the surface to find the ridges. Because computing derivatives is very sensitive to noise, we use a more robust property of these ridges which we describe intuitively here. Suppose we descend along a ridge for a fixed distance *f*, starting at a certain height *h*. If, instead, we do not start at the ridge, but on the isoline of height *h* a little to the right or to the left of the ridge, and walk the same distance in direction of steepest descent, we will end up lower. Consequently also the *integral* of the heights we pass when starting on the ridge, *H*, is higher than the integrals when starting beneath the ridge, H_{left} and H_{right}, and the same holds for the average height of the walk, $\overline{h} = H/f$. We approximate the respective *integral curves* of the (discrete) gradient vector field of κ on the triangular surface mesh (Forman, 1998) as described by Cazals et al., 2003.

We call the walking distance *f*, i.e. the arc length for integration, the *filter length*, specified by a number of edges n_f in our discrete setting. We call the difference of the average heights, $\min\{\overline{h} - \overline{h}_{left}, \overline{h} - \overline{h}_{right}\}$ (unit:mm^{-1}), the *significance* of a ridge. If significance is low, the ridge might not be sharply peaked. We therefore discard ridge pieces if significance does not exceed a user given threshold. We also discard ridge pieces if their average height $\overline{h}$ is below a user defined *curvature threshold*, because they are not necessarily strong features of the surface. Details can be found in Weber, 2008.

References

Bucki, M., Lobos, C., Payan, Y., 2010. A fast and robust patient specific finite element mesh registration technique: application to 60 clinical cases. Med. Image Anal. 14, 303–317.
Cazals, F., Chazal, F., Lewiner, T., 2003. ******Molecular shape analysis based upon the Morse-Smale complex and the Connolly function, in: SCG '03: Proceedings of the nineteenth Annual Symposium on Computational Geometry. ACM, New York, NY, USA, pp. 351–360.

[6] The "outward" direction of the acetabular rim plane can be determined by means of the orientation of the acetabular patch.

Chambers, J., 1983. Graphical Methods for Data Analysis. Chapman & Hall Statistics Series Wadsworth International Group.
Conway, J., Sloane, N., Bannai, E., 1999. Sphere packings, lattices, and groups. lattices. A Series of Comprehensive Studies in Mathematics. Springer.
Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J., 1995. Active shape models – their training and application. Comput. Vis. Image Underst. 61, 38–59.
Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271.
Forman, R., 1998. Morse theory for cell complexes. Adv. Math. 134, 90–145.
Glocker, B., Komodakis, N., Tziritas, G., Navab, N., Paragios, N., 2008. Dense image registration through MRFs and efficient linear programming. Med. Image Anal. 12, 731–741.
He, L., Peng, Z., Everding, B., Wang, X., Han, C.Y., Weiss, K.L., Wee, W.G., 2008. A comparative study of deformable contour methods on medical image segmentation. Image Vis. Comput. 26, 141–163.
Heimann, T., Delingette, H., 2011. Model-based segmentation. In: Deserno, T.M. (Ed.), Biomedical Image Processing. Biological and Medical Physics, Biomedical Engineering. Springer Berlin, Heidelberg, pp. 279–303.
Heimann, T., Meinzer, H.P., 2009. Statistical shape models for 3d medical image segmentation: a review. Med. Image Anal. 13, 543–563.
Hollander, M., Wolfe, D., 1999. Nonparametric statistical methods. Wiley Series in Probability and Statistics. Wiley.
Kainmueller, D., Lamecker, H., Seim, H., Zachow, S., Hege, H.C., 2010. Improving deformable surface meshes through omni-directional displacements and MRFs. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (Eds.), MICCAI. Lecture Notes in Computer Science, vol. 6361. Springer, pp. 227–234.
Kainmueller, D., Lamecker, H., Seim, H., Zinser, M., Zachow, S., 2009a. Automatic extraction of mandibular nerve and bone from cone-beam CT data. In: Yang, G.Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (Eds.), MICCAI. Lecture Notes in Computer Science, vol. 5762. Springer, pp. 76–83.
Kainmueller, D., Lamecker, H., Zachow, S., 2009b. Multi-object segmentation with coupled deformable models. Ann. Br. Mach. Vis. Assoc. (BMVA) 5, 1–10.
Kainmueller, D., Lamecker, H., Zachow, S., et al., 2009c. An articulated statistical shape model for accurate hip joint segmentation. In: Int. Conf. of the IEEE Eng. in Med. and Biol. Society (EMBC), Minneapolis, USA, pp. 6345–6351.
Kainmueller, D., Lange, T., Lamecker, H., 2007. Shape constrained automatic segmentation of the liver based on a heuristic intensity model. In: 3D Segmentation in the Clinic: A Grand, Challenge, pp. 109–116.
Koenderink, J., van Doorn, A., 1993. Local features of smooth shapes: ridges and courses. SPIE 2031 Geometric Methods in Computer Vision II, 2–13.
Komodakis, N., Tziritas, G., Paragios, N., 2008. Performance vs computational efficiency for optimizing single and dynamic MRFs: setting the state of the art with primal-dual strategies. Comput. Vis. Image Underst. 112, 14–29.
Lee, K., Johnson, R.K., Yin, Y., Wahle, A., Olszewski, M.E., Scholz, T.D., Sonka, M., 2010. Three-dimensional thrombus segmentation in abdominal aortic aneurysms using graph search based on a triangular mesh. Comput. Biol. Med. 40, 271–278.
Li, K., Wu, X., Chen, D.Z., Sonka, M., 2006. Optimal surface segmentation in volumetric images – a graph-theoretic approach. IEEE TPAMI 28, 119–134.
Ma, J., Lu, L., Zhan, Y., Zhou, X., Salganicoff, M., Krishnan, A., 2010. Hierarchical segmentation and identification of thoracic vertebra using learning-based edge detection and coarse-to-fine deformable model. In: Jiang, T., Navab, N., Pluim, J., Viergever, M. (Eds.), Medical Image Computing and Computer-Assisted Intervention MICCAI 2010. Lecture Notes in Computer Science, vol. 6361. Springer, Berlin/Heidelberg, pp. 19–27.
Montagnat, J., Delingette, H., Ayache, N., 2001. A review of deformable surfaces: topology, geometry and deformation. Image Vis. Comput. 19, 1023–1040.
Okada, T., Shimada, R., Sato, Y., Hori, M., Yokota, K., Nakamoto, M., Chen, Y.W., Nakamura, H., Tamura, S., 2007. Automated segmentation of the liver from 3d CT images using probabilistic atlas and multi-level statistical shape model. Proceedings of the 10th International Conference on Medical Image Computing and Computer-Assisted Intervention, vol. Part I. Springer-Verlag, Berlin, Heidelberg, pp. 86–93.
Park, J.Y., McInerney, T., Terzopoulos, D., Kim, M.H., 2001. A non-self-intersecting adaptive deformable surface for complex boundary extraction from volumetric images. Comput. Graph. 25, 421–440.
Paulsen, R.R., Baerentzen, J.A., Larsen, R., 2010. Markov random field surface reconstruction. IEEE Trans. Vis. Comput. Graph. 16, 636–646.
Petersen, J., Nielsen, M., Lo, P., Saghir, Z., Dirksen, A., de Bruijne, M., 2011. Optimal graph based segmentation using flow lines with application to airway wall segmentation. In: SzTkely, G., Hahn, H. (Eds.), Information Processing in Medical Imaging. Lecture Notes in Computer Science, vol. 6861. Springer, Berlin/Heidelberg, pp. 49–60.
Rothe, R., 1915. Zum Problem des Talwegs. Sitz. ber. d. Berliner Math. Gesellschaft 14, 51–69.
Seim, H., Kainmueller, D., Heller, M., Lamecker, H., Zachow, S., Hege, H.C., 2008. Automatic segmentation of the pelvic bones from CT data based on a statistical shape model. In: Proc. VCBM, pp. 93–100.
Seim, H., Kainmueller, D., Lamecker, H., Bindernagel, M., Malinowski, J., Zachow, S., 2010. Model-based auto-segmentation of knee bones and cartilage in MRI data. In: Ginneken, B. et al. (Eds.), Proc. MICCAI Workshop Medical Image Analysis for the Clinic: A Grand, Challenge, pp. 215–223.
Surazhsky, V., Gotsman, C., 2003. Explicit surface remeshing. In: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, Eurographics Association, Aire-la-Ville, Switzerland, pp. 20–30.

Terzopoulos, D., 1988. The computation of visible-surface representations. IEEE Trans. Pattern Anal. Mach. Intell. 10, 417–438.

Weber, B., 2008. Merkmalskurven auf triangulierten Oberflächen. Master's thesis. Department of Mathematics and Computer Science, FU-Berlin. Advised by Konrad Polthier and Steffen Prohaska.

Xu, C., Pham, D.L., Prince, J.L., 2000. Image segmentation using deformable models. In: Sonka M., Fitzpatrick, J.M. (Eds.), Handbook of Medical Imaging. Medical Image Processing and Analysis, vol. 2, pp. 129–174.

Yin, Y., Zhang, X., Williams, R., Wu, X., Anderson, D.D., Sonka, M., 2010. Logismos-layered optimal graph image segmentation of multiple objects and surfaces: cartilage segmentation in the knee joint. IEEE Trans. Med. Imag. 29, 2023–2037.

Zhang, X., Tian, J., Deng, K., Wu, Y., Li, X., 2010. Automatic liver segmentation using a statistical shape model with optimal surface detection. IEEE Trans. Biomed. Eng. 57, 2622–2626.