Omnidirectional Displacements for Deformable Surfaces

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

[a] *Zuse Institute Berlin, Germany*
[b] *Julius Wolff Institute, Charité University Medicine, Berlin, Germany*

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 intersect with the target boundary at "non-corresponding" positions, or may not intersect 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 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.

Keywords: Segmentation, Deformable Surfaces, Markov Random Field, Mandible, Acetabulum

1. Introduction

Since the pioneering work of Terzopoulos (1988), deformable models have been shown to be very effective for medical image segmentation. 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 (Heimann and Delingette, 2011).

Among many different shape representations, polygonal meshes are advantageous in many respects, such as flexibility and topology preservation (Montagnat et al., 2001). They are therefore widely used in model-based segmentation. Usually, the deformable mesh "probes" the image information at each vertex position. 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. The required amount of regularization depends on the imaging deficiencies present in the data, such as noise, artifacts, partial volume effects, low or no contrast due to adjacent anatomical structures with similar appearance, etc. Regularization approaches range from global methods using prior shape knowledge, such as Active Shape Models (Cootes et al., 1995; Heimann and Meinzer, 2009), to local methods that impose only local constraints upon shape deformation. While global regularization achieves robust results, local methods allow for more flexibility and thus may achieve more accurate results (Kainmueller et al., 2007).

The details of the "image probing" play a crucial role in the segmentation process. The image data is evaluated within a certain "search space" to assess suitable image features. Most commonly, *unidirectional* (i.e. linear, one-dimensional) search spaces are used at each vertex of the deformable mesh. This has a number of advantages: Feature assessment is fast within one-dimensional subsets of the image. It is easy to select the "best" feature, as required by many methods (Cootes et al., 1995; Kainmueller et al., 2007), because the set of suitable features is likely to be zero-dimensional, i.e. finite. Globally optimal deformation subject to local regularization constraints can be achieved for unidirectional search spaces (Li et al., 2006). 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.

At the same time, unidirectional search spaces cause two fundamental problems:

1. **Restricted visibility**. Unidirectional search spaces are prone to miss features in the image data (Fig. 1a and 1b).
2. **Incorrect correspondence**. Unidirectional search spaces frequently detect "wrong", i.e. non-corresponding, features in the image data (Fig. 1c and 1d).

If a-priori knowledge about characteristic deformations is available, one can introduce suitable global regularization in the

*Corresponding Author.
Email address: kainmueller@zib.de (Dagmar Kainmueller)

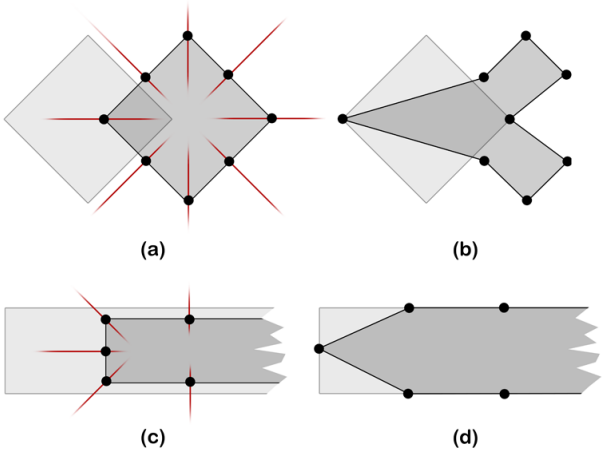


Figure 1: 2D sketches of exemplary deformable meshes (dark grey, with vertices as black dots) and target objects (light grey). Normal search spaces on (a) a cube and (b) a tip-like structure detect no target boundary points or non-corresponding target boundary points for most vertices. (b,d) Resulting unregularized deformations onto target object boundary.

segmentation process, e.g. by allowing only rigid transformations or employing Active Shape Models. This alleviates the impact of these problems, but does not solve them. If, however, only local restrictions are imposed upon deformations, the visibility and correspondence problems have severe impact; e.g., local translations of highly curved surface regions such as tips or ridges can hardly be achieved (cf. Fig. 1 and Fig. 4). Here, iterative deformation together with local mesh smoothing (Kainmueller et al., 2007), adaptive step-size control, adaptive remeshing or mesh surgery (Bucki et al., 2010) may serve as remedies – however, even when taking these measures the deformable surface may not converge to the correct image features, or the deformation process gets stuck because self-intersections of the mesh need to be prevented (Park et al., 2001). Despite these particular drawbacks of local regularization, it is typically employed in the "late" stage of a model-based segmentation process, when global regularization becomes too restrictive to reach optimal segmentation accuracy as permitted by the underlying image data.

In this paper, we propose a general solution to the problems of restricted visibility and incorrect correspondence on a local level of regularization[1]. The basic idea is an extended search space to allow not only unidirectional but *omnidirectional* displacements at each mesh vertex, thus enlarging visibility and – potentially – improving correspondence. An extended (more precisely three-dimensional, ball-shaped) search space per vertex most probably contains an infinite (two-dimensional) set of suitable image features. Hence the type of local regularization must be able to avoid highly inconsistent displacements of adjacent vertices. In particular, we allow displacements to a discrete set of points within a sphere around each vertex, while 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 explained in Sec. 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 set of displacements 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 Sec. 3, is to allow omnidirectional displacements only in regions of high curvature, while restricting displacements to surface normals in "flat" surface regions.

Sec. 4 and 5 provide an extensive evaluation of ODDS and fastODDS on synthetic and clinical data, which shows that

1. ODDS can handle 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 successful search for image features at each vertex of a deformable surface mesh in terms of the visibility and correspondence problem (see Sec.1), we propose to extend the search space from a line segment to a ball centered at each 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. We find an approximately optimal solution to the segmentation problem in a discrete setting via Markov Random Field (MRF) energy minimization.

We denote the set of vertices v of the deformable surface mesh as $V = \{v_i \in \mathbf{R}^3 \mid i = 1 \dots n_V\}$, 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 $S = \{s_i \in \mathbf{R}^3 \mid i = 1 \dots n_S\}$ is a discrete set of possible displacements. 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*.

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*. The *sampling distance* δ_S denotes the minimum Euclidean distance between unequal displacements $\delta_S := min_{s_i \neq s_j} \|s_i - s_j\|$.

[1] This work extends the authors' paper presented at MICCAI 2010 (Kainmueller et al., 2010).

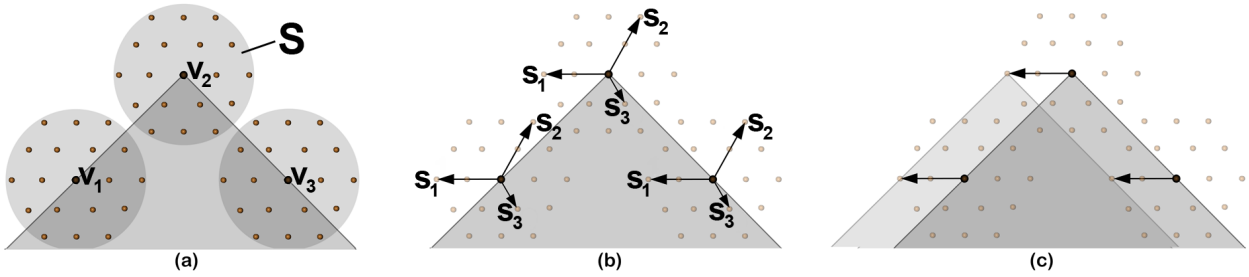


Figure 2: 2D sketch of omnidirectional displacements: (a) Black dots depict three vertices v_1, v_2, v_3 of a deformable mesh. Ball-shaped ranges of motion S (large grey spheres) around each vertex are discretized via sample points (yellow/gray dots). (b) Exemplary displacements s_1, s_2, s_3 to sample 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.

2.1. Omnidirectional Displacements

We define S as a set of displacements that are uniformly distributed within a sphere 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 refer to this ball-shaped set of displacements as *omnidirectional displacements*. Note that in contrast to traditional, unidirectional sets of displacements (i.e. *unidirectional displacements*), omnidirectional displacements are interpreted in world coordinates for all vertices, as opposed to local coordinate frames per vertex (see Fig. 2b and 2c).

2.2. Objective Function

For each displacement $s \in S$ and vertex $v \in V$, a scalar cost $\phi(v, s) \geq 0$ 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" $\psi(s_i, s_j) \geq 0$ 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. It has to satisfy $\psi(s_i, s_j) = 0 \Leftrightarrow s_i = s_j$, but does not have to be a metric (see Sec. 2.3). 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:

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

This means we are looking for the displacement field 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.

Note that interpreting displacements in world coordinates (cf. Sec. 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 its 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 can deal with non-metric distance functions ψ as specified in Sec. 2.2, and is guaranteed to find an approximately optimal solution (Komodakis et al., 2008).

2.4. Refined Regularization

The condition $\psi(s_1, s_2) = 0 \Leftrightarrow s_1 = s_2$ 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.

To this end, a "tolerated distance" with zero penalty[2] can be approximated as follows: Let $\tilde{S} = \{\tilde{s}_i \in \mathbf{R} \,|\, i = 1 \ldots n_{\tilde{S}}\}$ be a second cubic close-packed lattice which is coarser than S, i.e. $\delta_{\tilde{S}} > \delta_S$. $\tilde{S}$ partitions S into *displacement blocks* B_i by means of nearest-neighborhood to its elements $\tilde{s}_i$. Formally, $B_i = \{s \in S : \tilde{s}_i = \operatorname{argmin}_{\tilde{s} \in \tilde{S}} \|s - \tilde{s}\|\}$. Given the displacement

[2] in other words, a smaller sampling distance with fixed regularization

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 $\tilde{\psi}(\tilde{s}_i, \tilde{s}_j) = \psi(\|\tilde{s}_i - \tilde{s}_j\| \cdot \delta_S / \delta_{\tilde{S}})$. We optimize the respective MRF energy w.r.t. $\tilde{d} : V \to \tilde{S}$ and assign to vertex v with $\tilde{d}_v = \tilde{s}_i$ the displacement $s \in B_i$ with minimum cost, i.e. $d_v = \text{argmin}_{s \in B_i} \phi(v, s)$.

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

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, ODDS on a medium-sized mesh with 8000 vertices and 40 sample points per sphere diameter take about 2:30 minutes to compute on a 3.5 GHz core and require more than 4 GB of memory (cf. Tab. 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, while the wrong-correspondence problem in practice only affects a "bounding band" of the flat structure. 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.

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 Sec. 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 Sec. 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 Sec. 3.1), because (1) it has proven to be powerful for accurate fine-grain segmentation of medical image data (Seim et al., 2008; Heimann et al., 2010), and (2) it allows for simultaneous segmentation of multiple objects (*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 Sec. 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 $\ell_v \in \mathbf{R}$ with $\|\ell_v\| = 1$ yield respective discretized displacement sets per vertex, $L_v = \{l_i \in \mathbf{R} \mid i = 1 \dots n_{L_v}\}$, 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

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

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 1D.

3.2. Where to use Omnidirectional Displacements

When defining the surface region where omnidirectional displacements shall be applied (OmniD-region), we assume that on the boundary of the OmniD-region, surface-tangential movements are not allowed, as 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, we need 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

[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.

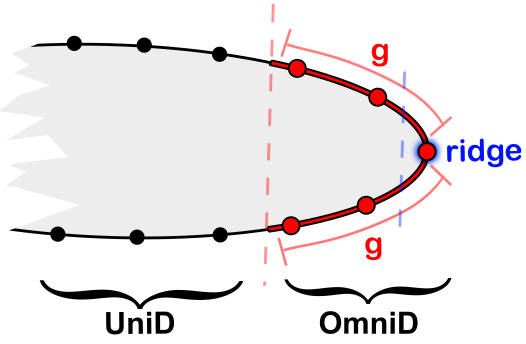


Figure 3: 2D-sketch of OmniD- and UniD-region on an exemplary tip-like mesh (vertices depicted as dots) with *ridge* at rightmost vertex (blue dot). Only a small region around the ridge exhibits high curvature, as indicated by the light blue, dashed line. Instead, we assign vertices within a certain *geodesic distance* g around the ridge to the OmniD-region; all others belong to the UniD-region.

OmniD-region. "Stretching" each of these edges by one sampling distance δ costs no more than the minimum non-zero distance penalty per edge and 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 Sec. 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 Sec. 3.4). In this context, for ease of notation, we refine the definition of a displacement field to

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

Then the objective function of fastODDS is defined as follows:

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

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

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

via GraphCuts. Second, we approximate

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

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 $O(c/\delta \cdot |\{w \in V_U : \exists v \in V_O : (v, w) \in E_\delta\}|)$. As this is roughly $3 \cdot 1000$ in our experiments (cf. Sec. 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 one side of a ridge "inward" and the opposite side "outward" 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, we assign zero cost to the closest displacement in S, i.e. $\text{argmin}_{s \in S} \|d|_{V_U}(w) - s\|$, and infinite cost to all others.

3.5. Multi-object FastODDS

GraphCuts can be used for simultaneous segmentation of multiple objects (Li et al., 2006) 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, and subsequent constrained ODDS on the OmniD-region. 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 OmniD-region overlaps with the coupled region, the resulting deformed surface in this overlap 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 "pure" GraphCuts (Li et al., 2006), and furthermore with a locally regularized method (Kainmueller et al., 2007) (FreeForm), both employing vertex normals as displacement directions, with the exception of multi-object GraphCuts, where shared displacement directions are used. FreeForm selects the minimum cost displacement for each vertex and subsequently regularizes locally via a small displacement toward the centroid of the respective adjacent vertices. In contrast to GraphCuts- and ODDS/fastODDS, all FreeForm adaptations were performed iteratively, with 30 steps.

We computed costs $\phi(v, s)$ from the image $I : \mathbf{R}^3 \rightarrow \mathbf{R}$ as proposed by Seim et al. (2008): If the image intensity $I(v + s)$ lies within a certain window $[i_0, i_1]$, costs are inversely proportional to the directional derivative $\nabla_{n_v} I(v + s)$, where n_v denotes the surface normal at v. Otherwise costs are set to a constant, high value c_{high}. As for the trade-off between image image fit and regularization, we scale the cost function such that $a \cdot \psi(\delta_{\tilde{S}}) < c_{high} < a \cdot \psi(2\delta_{\tilde{S}})$, where $a = 6$ is the average number of edges per vertex. The thresholds i_0 and i_1 are parameters of the strategy and are set per application (see Tab. 1). Whenever we compare different adaptation methods on the same image data, we use the same cost function ϕ for all methods.

	#V	2r	δ_S	$[i_0, i_1]$	n_f	g
Cube	770	26	0.5	[0.1, 1.1]	-	-
Ellipsoid	1797	31	0.5	[0.1, 1.1]	-	-
Mandible	8561	15	0.4	[350, 800]	6	6
Hip Bone	14008	20	0.5	[120, 720]	10	10

Table 1: Application specific parameters are the number of vertices #V of the deformable mesh, the diameter 2r [mm] of the range of motion, the sampling distance δ_S [mm] of the set of displacements, the intensity window $[i_1, i_2]$ of the cost function, and filterlength n_f (in number of edges) and geodesic distance g [mm] for definition of the OmniD-region.

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

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 sphere 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 $0.04\,\text{mm}^{-1}$ and curvature threshold $0.1\,\text{mm}^{-1}$.

Table 1 lists further parameters that we set individually per application.

4.1. Synthetic Images

We performed experiments on binary images[4] 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 sphere diameters such that the target object boundary was located completely within a band of respective width around the initial mesh. The results of ODDS-, FreeForm- and GraphCuts adaptation are shown in Fig. 4c, 4d and 4e, respectively. Fig. 4b shows the results of adding normal displacements without any regularization.

We added various amounts of random noise to the binary cube image and performed ODDS as before. The cube

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

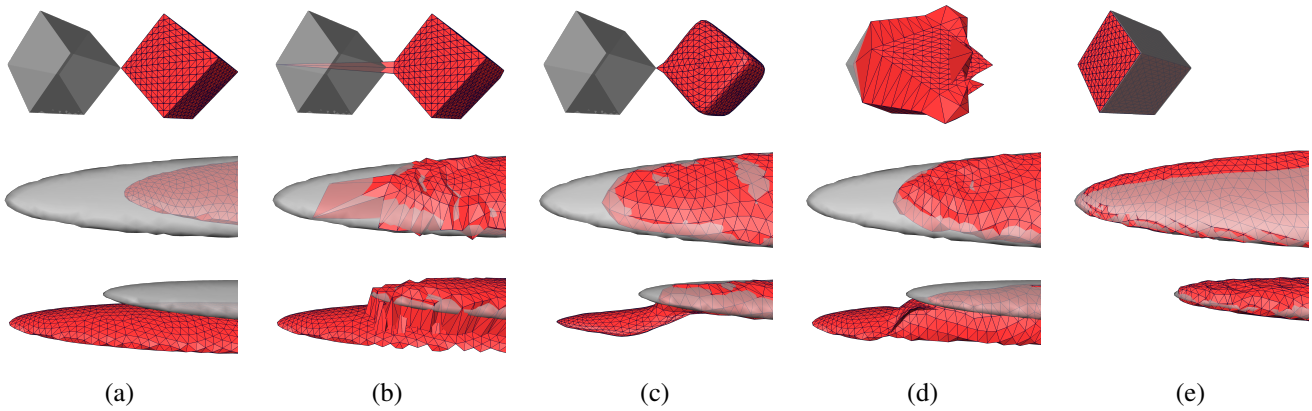


Figure 4: Results on synthetic data. Deformable mesh (red/dark grey mesh) and target object (transparent grey surface) are shown (a) in their initial situation, and after deformation via (b) displacements along normals without regularization, (c) FreeForm, (d) GraphCuts, and (e) ODDS.

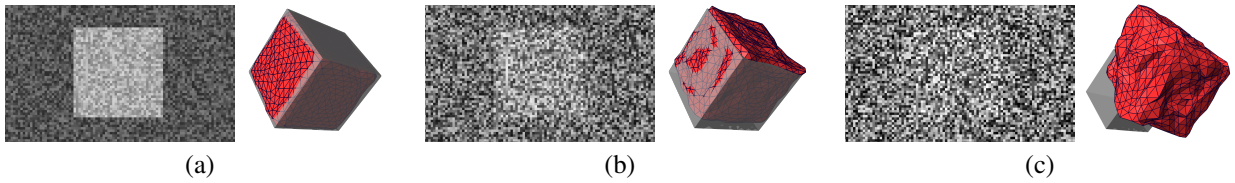


Figure 5: Performance of ODDS in the presence of noise. We added random noise with range (a) [−0.5..0.5], (b) [−2.5..2.5] and (c) [−5..5] to a binary image of a cube. We show slices of the image data and the respective adaptation result (red/dark grey mesh). The grey transparent surface depicts the ideal target object.

was detected correctly for noise with ranges [−0.5..0.5] and [−2.5..2.5], and failed for [−5..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 we compared ODDS, fastODDS, FreeForm and GraphCuts results to gold standard surfaces obtained from manual segmentations. Initial meshes were generated automatically by adaptation of a statistical shape model (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 1st principal curvature of the deformable surface. We achieve this by adding to $\phi(v, s)$ a small cost proportional to $r_S - s \cdot \nabla k_1(v)/\|\nabla k_1(v)\|$.

For the mandible surfaces (gold standard as well as adaptation results), 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.

As error measures for the coronoid process, we assessed the tip-to-tip distances (tip2tip), two-sided roots mean square (rms) and Hausdorff (max) surface distances, as well as the percentage of two-sided surface distances above 1 mm (%>1mm). Furthermore, we also assessed rms, max and %>1mm as error measures for the entire mandible surface.

Evaluation results for both coronoid process and entire mandible are shown in Tab. 2 and Fig. 6. As measurements are not normally distributed, we performed Wilcoxon's signed-rank test (Hollander and Wolfe, 1999) to assess significant differences.

4.2.2. Hip Bones (Acetabular Rim)

In a quantitative evaluation on 49 hip CTs we compared FastODDS and GraphCuts results to gold standard surfaces obtained from manual segmentations. Initial meshes were generated automatically by adaptation of a statistical shape model (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. Sec.4.2.1).

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 root mean square (rms) and Hausdorff (max) curve distances, as well as the percentage of distance above 1 mm (%>1mm). Furthermore, we assessed the rms and max *surfaces* distances as well as the %> 1mm measure as error measures for the whole hip bones.

Evaluation results for both acetabular rim and whole hip bone are shown in Table 3 and Fig. 7. As for the mandible, error measures are not normally distributed, and hence we per-

	Coronoid Process				Mandible		
	tip2tip [mm]	rms [mm]	max [mm]	%>1mm [%]	rms [mm]	max [mm]	%>1mm [%]
FreeForm	**2.58**(2.36)	**0.50**(0.52)	**2.62**(2.23)	**4.06**(6.97)	**1.50**(0.25)	**8.98**(1.37)	**19.92**(4.05)
GraphCuts	**2.25**(2.36)	**0.49**(0.56)	**2.52**(2.39)	**3.54**(6.72)	**1.05**(0.20)	**7.56**(1.54)	**13.42**(2.58)
fastODDS	**1.74**(2.32)	**0.44**(0.54)	**2.11**(2.26)	**2.40**(5.69)	**1.05**(0.21)	**7.54**(1.55)	**13.13**(2.54)
ODDS	**1.68**(2.11)	**0.41**(0.39)	**2.01**(2.04)	**2.15**(4.74)	**1.04**(0.20)	**7.52**(1.50)	**12.94**(2.48)
GraphCuts-ODDS	**0.57**	**0.09**	**0.51**	**1.40**	**0.01**	**0.04**	**0.48**
p-value [%]	<0.01	1.07	<0.01	<0.01	4.90	-18.70	<0.01
GraphCuts-fastODDS	**0.51**	**0.06**	**0.41**	**1.14**	**0.00**	**0.02**	**0.29**
p-value [%]	<0.01	5.13	<0.01	<0.01	1.46	-22.17	<0.01
fastODDS-ODDS	**0.06**	**0.03**	**0.10**	**0.25**	**0.00**	**0.03**	**0.19**
p-value [%]	-23.53	26.91	18.63	16.13	38.41	-41.26	<0.01

Table 2: Top to bottom: Average error measures (and standard deviation) for FreeForm, GraphCuts, fastODDS and ODDS results on 212 coronoid processes and 106 entire mandibles, followed by differences $A - B$ of average error measures for $A, B \in$ {GraphCuts, fastODDS, 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 highlighted by color.

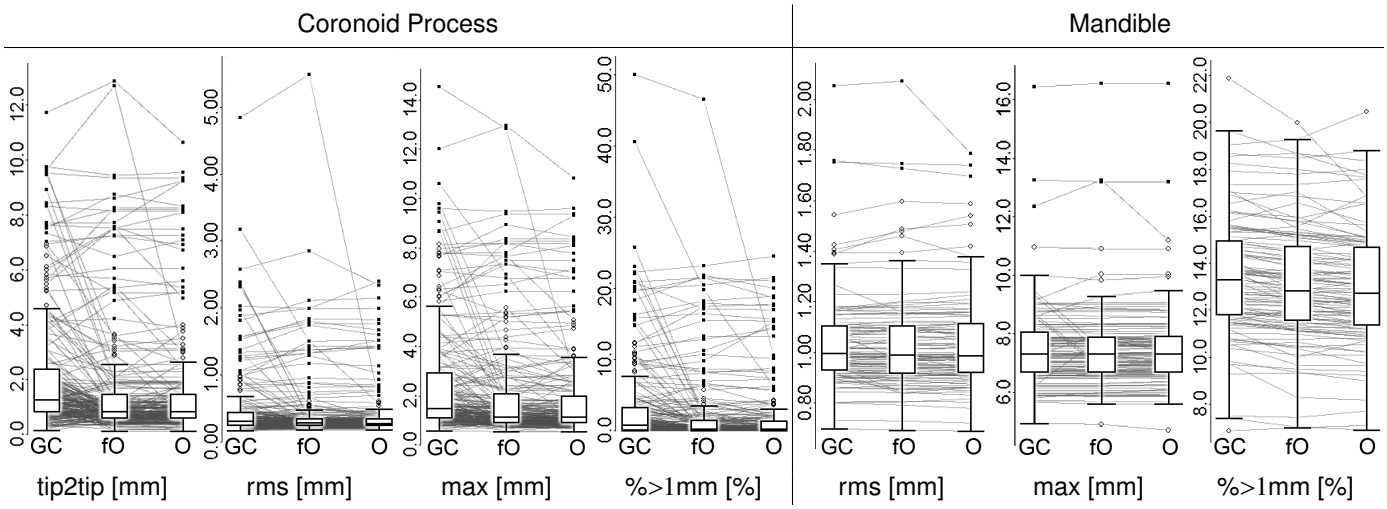


Figure 6: Box plots[6] of error measures for GraphCuts (GC), fastODDS (fO) and ODDS (O) results on coronoid processes and entire mandibles as listed in Tab. 2. Underlaid parallel coordinate plots draw lines between errors measured for different methods (GC, fO, O) on corresponding individual cases, e.g. between the tip2tip errors of GraphCuts- and fastODDS-result on coronoid process no. 189, etc.

[sec / GB]	ODDS	fastODDS	GraphCuts	FreeForm
Mandible	149 / 4.6	85 / 2.2	3 / 0.9	6 / 0.4
Hip Bone	-	319 / 5.4	18 / 2.3	-

Table 4: Performance (computation time in seconds / maximum memory requirement in GB) averaged for 106 mandibles and 98 hip bones.

formed Wilcoxon's signed-rank test to assess significant differences. Additionally, Fig. 8b and 8c show 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 3GHz core with 8GB main memory. Table 4 lists the average performance of all methods applied to clinical data. MRF optimization (Komodakis et al., 2008) took between 1 and 6 seconds in all ODDS- and fastODDS experiments. However, 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 Tab. 4.

5. Discussion

5.1. Accuracy

Experiments on synthetic binary images show that ODDS are able to handle parallel translations, in contrast to a globally and a locally regularized approach (GraphCuts and FreeForm) that employ normal displacements. Experiments on noisy synthetic images show that ODDS are able to produce well-regularized

[6] with outliers as circles and extreme outliers as dots; see Chambers (1983).

	Acetabular Rim			Hip Bone		
	rms [mm]	max [mm]	%>1mm [%]	rms [mm]	max [mm]	%>1mm [%]
GraphCuts	**1.90**(0.85)	**5.00**(2.53)	**61.17**(12.89)	**0.59**(0.18)	**6.93**(2.31)	**5.20**(2.30)
fastODDS	**1.62**(0.89)	**4.69**(2.75)	**45.20**(13.37)	**0.54**(0.18)	**6.74**(2.42)	**3.80**(2.07)
GraphCuts-fastODDS	**0.28**	**0.32**	**15.98**	**0.05**	**0.19**	**1.40**
p-value [%]	<0.01	1.04	<0.01	<0.01	6.43	<0.01

Table 3: Top: Average error measures (and standard deviation) for 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 highlighted by color.

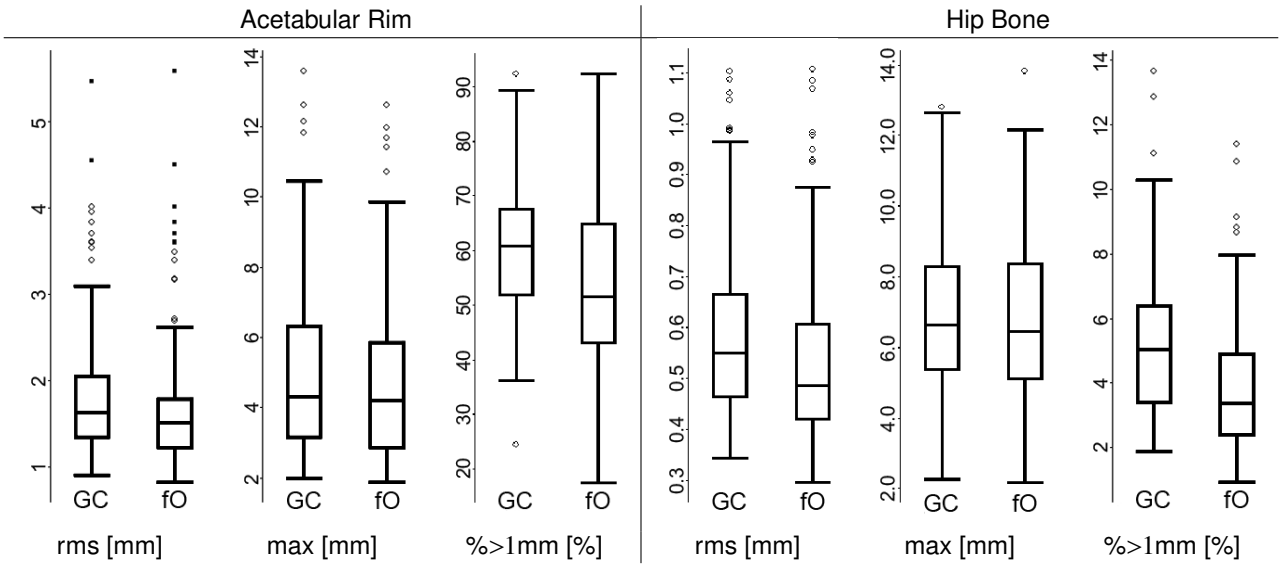


Figure 7: Box plots of error measures for GraphCuts (GC) and fastODDS (fO) results on acetabular rims and hip bones as listed in Tab. 3.

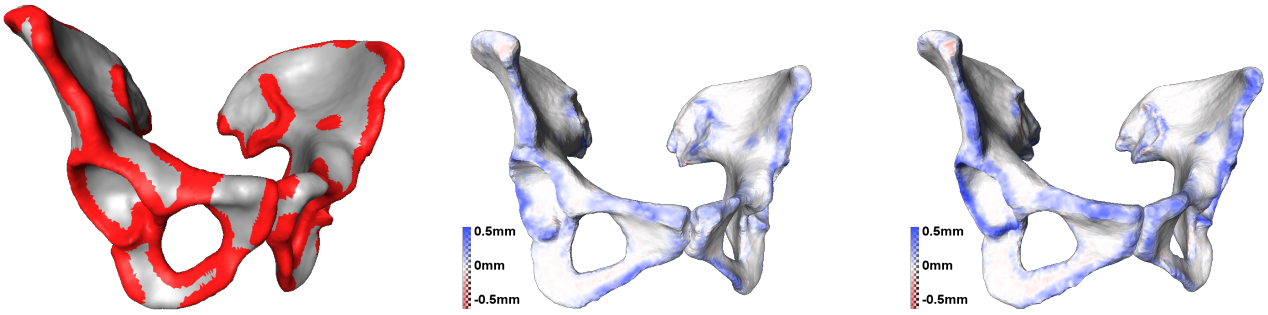
displacement fields in the presence of noise. However, for a very low signal-to-noise ratio, ODDS may fail to detect the target object. Experiments on clinical data show that both ODDS and fastODDS are able to produce very accurate segmentations of tip-like structures. On 212 mandibular coronoid processes, ODDS and fastODDS clearly outperform the GraphCuts and FreeForm approach. Here, normal displacements often exhibit the "wrong-visibility" problem, see Fig. 9(a-c), which omnidirectional displacements resolve.

A comparison of ODDS and fastODDS on the mandibular coronoid processes reveals no significant differences for any error measure. However, the parallel-coordinate plots that underlay the box plots in Fig. 6 show that there are at least two of 212 coronoid process cases where the ODDS error measures are considerably smaller than the fastODDS error measures. There are also some individual cases in which the fastODDS error measures are smaller than the ODDS error measures, yet with a smaller difference. We conclude that fastODDS is not guaranteed to produce an equally good result in the individual case, but overall does not perform significantly different than ODDS.

As for the whole mandible surface, both fastODDS and ODDS perform significantly better than GraphCuts and FreeForm, except for the Hausdorff surface distance error measure. This can be attributed to 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 (e.g. around the teeth or the chin which often lies outside the FoV of the CBCT scanner).

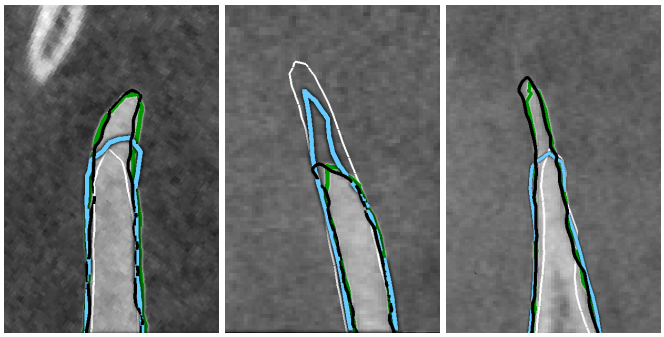
As for a comparison of ODDS and fastODDS on the whole mandible surface, no significant differences occur for rms and Hausdorff error measures, while, surprisingly, ODDS outperforms fastODDS in terms of the %>1mm error measure. As this measure is not significantly different when comparing ODDS and fastODDS on the (highly curved) coronoid process, we hypothesize that the difference on the whole surface is caused by a better performance of ODDS vs. GraphCuts in flat surface regions in terms of a higher amount of very small distances (<1 mm). This may be due to the fact that, from a "continuous" point of view, a ball-shaped range of motion contains an infinite (2D) set of target boundary points, while a unidirectional range most probably contains a finite (0D) set. If image features happen to be weak or not present in this finite set, GraphCuts have no other choice than to interpolate the displacement for the respective vertex.

Experiments on clinical data 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 "pure"



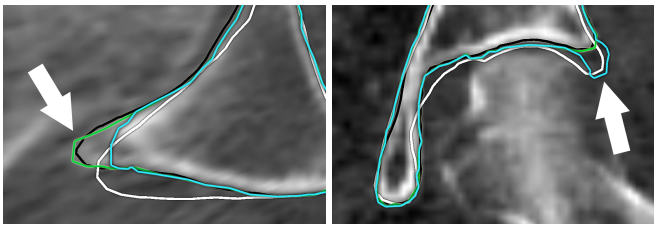
(a) (b) (c)

Figure 8: FastODDS on hip bones. (a) Exemplary hip bones with OmniD-region (red) and UniD-region (grey). (b,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.



(a) (b) (c)

Figure 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. ODDS work fine while FreeForm results are inaccurate because of the visibility and correspondence problems. (b) FreeForm can also cause the deformable mesh to get stuck due to prevention of self-intersections.



(a) (b)

Figure 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.

multi-object GraphCuts approach. Here, unidirectional displacements often struggle with incorrect correspondences as well as restricted visibility, see Fig. 10a and 10b, respectively.

FastODDS also performs significantly better in terms of error measures evaluated on the whole hip bones, except for the Hausdorff error measure, where the p-level of 6.43% is not doubtlessly convincing. As for the mandible, this may be due to 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 range.

5.2. Fairness of Comparison

As described in Sec. 4, we set the regularization parameter c of GraphCuts to the displacement block sampling distance δ_B as set in the respective ODDS/fastODDS experiment. For GraphCuts, differences of displacements on neighboring vertices are "for free" up to a Euclidean norm of c, while being impossible $> c$. 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 c, while the penalty increases cubically for larger differences (see Sec. 2.3).

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 $c = \delta_{\bar{S}}$, 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 Tab. 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 displacements in terms of an effective denser sampling in surface-normal direction due to additional adjacent sampling points. It *did* slightly improve the <1 mm measure for GraphCuts results – however, the accuracy of ODDS/fastODDS in terms of this measure could not be reached, not even with still smaller (nor bigger) sampling distances from 0.25 to $1\delta_S$. We discussed a potential reason for this in Sec. 5.1

5.3. 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. 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.

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 (normals) in regions with high curvature (tips) 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 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 includes an analysis of our hypothesis as to why the percentage of very small errors on entire surfaces is higher for ODDS than for fastODDS, as well as 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 University Medicine, Berlin, Germany) for manually delineating the acetabular rims.

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

Table A.5: Automatic acetabular rim delineation: Average root mean square (rms) and Hausdorff (max) distance from manually defined landmarks, and percentage of distance above 1,mm (%>1mm), assessed on 147 hip bone surfaces. Standard deviations in brackets.

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 fine grain adaptation methods. 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 initial rim estimate, sample a set of points on the hip bone surface perpendicular to the rim within some geodesic distance. (3) Define a *cost* per sample point as the distance from the approximate rim-plane in outward[7] 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 optimization 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 Tab. A.5.

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

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

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 *filterlength*, specified by a number of edges n_f in our discrete setting. We call the difference of the average heights, $\min\{\overline{h} - \overline{h}_{left}, \overline{h} - \overline{h}_{right}\}$ [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* [mm^{-1}], 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. Medical Image Analysis 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.

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. 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.

Forman, R., 1998. Morse theory for cell complexes. Advances in Mathematics 134, 90–145.

Glocker, B., Komodakis, N., Tziritas, G., Navab, N., Paragios, N., 2008. Dense image registration through MRFs and efficient linear programming. MIA 12, 731 – 741.

Heimann, T., Delingette, H., 2011. Model-based segmentation, in: Deserno, T.M. (Ed.), Biomedical Image Processing. Springer Berlin Heidelberg. Biological and Medical Physics, Biomedical Engineering, pp. 279–303.

Heimann, T., Meinzer, H.P., 2009. Statistical shape models for 3d medical image segmentation: A review. Medical Image Analysis 13, 543 – 563.

Heimann, T., Morrison, B.J., Styner, M.A., et al, 2010. Segmentation of knee images: A grand challenge, in: Proc. Medical Image Analysis for the Clinic: A Grand Challenge, Bejing, China. pp. 207–214.

Hollander, M., Wolfe, D., 1999. Nonparametric statistical methods. Wiley series in probability and statistics: Texts and references section, 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 (1), 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: MICCAI (1), Springer. pp. 76–83.

Kainmueller, D., Lamecker, H., Zachow, S., 2009b. Multi-object segmentation with coupled deformable models. Annals of the British Machine Vision Association (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.

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.

Montagnat, J., Delingette, H., Ayache, N., 2001. A Review of Deformable Surfaces: Topology, Geometry and Deformation. Image Vis. Comput. 19, 1023–1040.

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.

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.

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.