Iterative Closest Point with Minimal Free Space Constraints

Simen Haugo and Annette Stahl

Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
{simen.haugo,annette.stahl}@ntnu.no

Abstract. The Iterative Closest Point (ICP) method is widely used for fitting geometric models to sensor data. By formulating the problem as a minimization of distances evaluated at observed surface points, the method is computationally efficient and applicable to a rich variety of model representations. However, when the scene surface is only partially visible, the model can be ill-constrained by surface observations alone. Existing methods that penalize free space violations may resolve this issue, but require that the explicit model surface is available or can be computed quickly, to remain efficient. We introduce an extension of ICP that integrates free space constraints, while the number of distance computations remains linear in the scene's surface area. We support arbitrary shape spaces, requiring only that the distance to the model surface can be computed at a given point. We describe an implementation for range images and validate our method on implicit model fitting problems that benefit from the use of free space constraints.

Keywords: 3D model fitting · implicit modeling · visibility constraints

1 Introduction

Iterative Closest Point (ICP) [1] is commonly used for fitting geometric models to sensor data. At its core is a local optimization, requiring only computation of data-to-model distances at observed surface points. This makes ICP computationally efficient and applicable to any model representation where the distance to the surface can be computed. However, when the scene surface is only partially visible, the model's shape and pose may be ill-constrained by surface observations alone. A common solution is to impose constraints or penalties on the parameters [2], but this is a model-specific intervention. Alternatively, one can integrate free space constraints (space observed to be empty) that vision systems often provide, but which is not used by ICP. This introduces a new challenge of managing computational complexity, as free space is inherently volumetric [3]. While efficient methods have been proposed when the model surface (or a bound) is available in explicit form [4–7], these are not applicable when the explicit surface is unavailable or prohibitively expensive to compute.

We propose an extension of ICP for implicit models that integrates free space constraints, while retaining the computational complexity of the original method, *i.e.* the number of distance computations scales linearly by the surface area of the scene. Our method supports arbitrary shape spaces, requiring only that the distance to the model surface can be computed at a given point. We demonstrate on range data that our method is amenable to off-the-shelf optimizers, and can resolve parameters which are ill-constrained by surface observations alone.

2 Related Work

Geometric model fitting has been extensively studied over several decades (see [8] for a survey). Notably, we now have efficient methods that are highly robust against outliers and provide strong optimality guarantees [9–11]. To achieve such performance, state-of-the-art methods exploit the structure of the domain (*e.g.* SE3) and/or rely on the ability to match invariant features between the model and the data. These methods work very well for explicit representations, such as static point sets obtained from laser scanners [9, 10] or deformable landmark models [11, 12]. Unfortunately, they are incompatible with representations where the domain does not allow for easily-exploited structure or invariant features that can be detected and matched.

In this work, we focus on models represented as (parameterized) distance functions. Distance functions have been derived exactly or approximately for several primitives and implicit modeling operations, thereby enabling constructive modeling of objects [13–15]. In robotics and computer vision, distance functions are a natural representation for scene reconstruction and planning [16–18], and have recently fueled research as a neural shape representation in machine learning [19]. The ability to efficiently fit models, using only their distance function, to (incomplete) sensor data, would enrich each of these application areas. We therefore review related work on implicit model fitting, where the explicit surface is not directly maintained and where the implicit function itself may be expensive to evaluate.

The Iterative Closest Point (ICP) method [1, 20–24] can be applied to any model representation, requiring only that the distance to the model surface can be computed at each point on the observed surface. This makes ICP efficient, as the computational cost scales linearly by the observed surface area. However, the model can be ill-constrained when the scene's surface is only partially observed (see Fig. 1), e.g. due to capturing data from predominantly one viewpoint or due to noise and specular reflections. Regularization strategies [8], such as penalizing description length [25, 26], minimizing volume or surface area [2, 27] or imposing constraints on the model parameters [2] can alleviate this problem, but are model-specific interventions that do not generalize.

Alternatively, the fitting method can use visibility information provided by the vision system; *e.g.* a finite range value from a laser scanner indicates not only that there is a surface at that distance from the sensor, but also that there are no surfaces in-between. Visibility information plays a central role in 3D

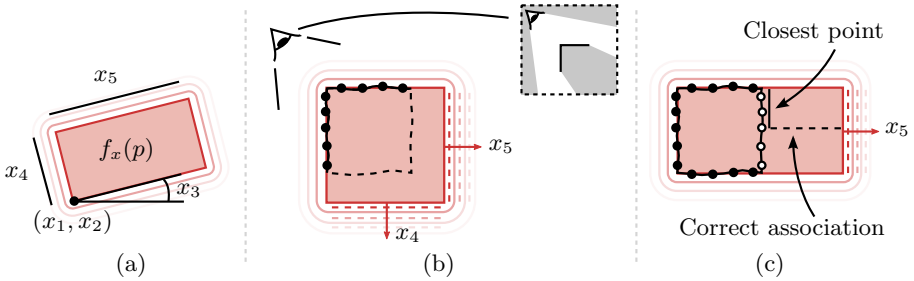


Fig. 1. Illustration of cases where a volume-aware method has advantages over Iterative Closest Point (ICP). Given a model (a) represented by a distance function f_x, we seek to estimate the parameters x such that the resulting solid fits to a partially observed scene. When using ICP, a subset of the parameters can be ill-constrained due to (b) missing data (despite being wrong, the model fits all the observed points) and (c) poor initialization (closest point association leads to uninformative gradients).

reconstruction [28], *e.g.* in space carving [16, 3], and has also been used in model fitting, *e.g.* in volume matching [29–31], minimizing reprojection error [32–37] and matching of occluding contours or silhouettes [38–41, 6, 7]. These methods can be said to be *volume-aware* [42] and have the ability to resolve ill-constrained models by requiring the model to be consistent with free space.

However, existing methods can be prohibitively expensive for implicit models where the associated implicit function is expensive to evaluate. Volume matching (*e.g.* of density, distance or binary occupancy) supports any implicit representation, but is based on densely sampling the volumetric domain. Computing reprojection error or the model silhouette requires the extraction of visible points on the surface. Extraction can be done using accelerated ray casting for distance function models [13], but still requires multiple samples along rays. Extraction can also be done using spatial subdivision schemes [43], but requires an initial resolution that contains sampling points from every connected component of the model's interior and exterior. The availability of the explicit surface (or a close bound) has been used to derive efficient volume-aware methods [4–7]. For implicit models, the explicit surface must in general be recomputed each time the model changes. For small deformations, a set of points can be made to track the implicit surface as the model parameters are optimized [44]. However, efficiently maintaining such an approximation for general domains is an open problem.

Following this line of work, we propose a novel volume-aware method that integrates free space constraints. Similar to ICP, our method only requires the ability to compute the distance to the model at a given point. Unlike previous volume-aware methods, our method does not require the explicit surface or dense volumetric sampling. Instead, our method retains the computational complexity of ICP, in that the number of distance computations per iteration scales linearly by the scene's surface area.

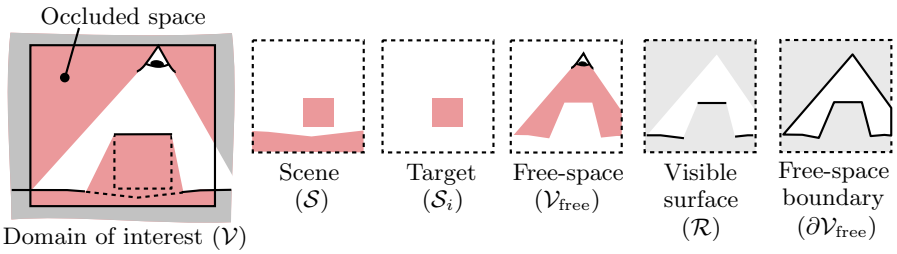


Fig. 2. Notation applied to an example scene containing a target object and outliers.

3 Theory & Method

This section describes our volume-aware extension of the ICP method. We first formalize volume-aware model fitting based on *volume- and surface-consistency* conditions. We then show how these conditions can be expressed in a constrained optimization problem which retains the computational complexity of ICP.

3.1 Notation

Here we introduce notation used in the following sections (see also Fig. 2). A scene, assumed to be some unknown solid $\mathcal{S}$, is observed by a vision system, *e.g.* a range sensor. We consider a domain of interest $\mathcal{V}$ that contains a subset of $\mathcal{S}$, *e.g.* a user-defined box region. The scene is assumed to be decomposed into target and outlier solids, $\mathcal{S}_i$ and $\mathcal{S}_o$, with respective boundaries $\partial\mathcal{S}_i$ and $\partial\mathcal{S}_o$.

Free space $\mathcal{V}_{\text{free}}$ is a closed subset of $\mathcal{V}$, determined by the vision system, which does not intersect the interior of $\mathcal{S}$ (*i.e.* $\mathcal{V}_{\text{free}} \subseteq \mathcal{V}$ and $\mathcal{V}_{\text{free}} \cap \text{ int } \mathcal{S} = \emptyset$). Occluded space is the complement of free space inside the domain of interest. The free space boundary $\partial\mathcal{V}_{\text{free}}$ is the boundary between free and occluded space. The visible surface $\mathcal{R}$ is a subset of $\partial\mathcal{V}_{\text{free}}$, determined by the vision system, which also belongs to the physical scene boundary $\partial\mathcal{S}$. If the scene contains outliers, we assume that the visible surface has been segmented into respective target and outlier surfaces $\mathcal{R}_i \subseteq \partial\mathcal{S}_i$ and $\mathcal{R}_o \subseteq \partial\mathcal{S}_o$. The signed Euclidean distance to a solid $\mathcal{D}$ is denoted $d_{\mathcal{D}}(p) := \pm \min_{q\in\partial\mathcal{D}} ||p-q||_2$, where the sign is negative for p inside $\mathcal{D}$ and positive outside.

3.2 Problem Formulation and Volume-Aware Model Fitting

We assume a geometric model of the target object is given as a real-valued function $f_x(p) : \mathbb{R}^3 \times \mathcal{X} \to \mathbb{R}$ defining the solid $\mathcal{M}_x = \{p \in \mathbb{R}^3 : f_x(p) \leq 0\}$. We assume that f_x is the Euclidean distance function $d_{\mathcal{M}_x}$. The model parameters $x \in \mathcal{X}$ may include the object's pose (SE3 $\subseteq \mathcal{X}$), as in rigid registration, but here we assume x can also include any real-valued, discrete or symbolic parameters

that define the shape. Our goal is to estimate the parameters x such that the resulting solid is consistent with the observed surface and free space, *i.e.*

$$f_x(p) = 0, \forall p \in \mathcal{R}_i \text{ and} \tag{1}$$
$$f_x(p) > 0, \forall p \in \text{int } \mathcal{V}_{\text{free}}. \tag{2}$$

The free space constraints (2) imply that the model cannot occupy free space, but can occupy any subset of occluded space. When (2) is satisfied, we say that $\mathcal{M}_x$ is *volume-consistent* with $\mathcal{V}_{\text{free}}$. When (1) is satisfied, we say that $\mathcal{M}_x$ is *surface-consistent* with $\mathcal{R}_i$. When both are satisfied, we say that $\mathcal{M}_x$ is *consistent*. For brevity, we will say that a solid is volume- or surface-consistent and leave it understood that volume consistency refers to $\mathcal{V}_{\text{free}}$ and that surface consistency refers to the visible surface $\mathcal{R}_i$.

3.3 Distance Constraints for Volume- and Surface-Consistency

When f_x is the Euclidean distance function, free space implies the existence of a set of inequality constraints bounding the distance from below:

$$f_x(p) \geq \min_{q \in \partial\mathcal{V}_{\text{free}}} ||p - q||_2, \forall p \in \mathcal{V}_{\text{free}}. \tag{3}$$

If the closest point $q \in \partial\mathcal{V}_{\text{free}}$ in (3) also belongs to $\partial\mathcal{S}$, the inequality is replaced by an equality. In practice, we do not know $\partial\mathcal{S}$ and therefore cannot identify everywhere that this holds. However, we do know the visible surface $\mathcal{R} \subseteq \partial\mathcal{S}$. Hence, we can identify the subset of equality constraints:

$$f_x(p) = \min_{q \in \mathcal{R} \cap \partial\mathcal{V}_{\text{free}}} ||p - q||_2, \forall p \in \mathcal{V}_{\text{free}}. \tag{4}$$

If the scene contains outliers, $\mathcal{R}$ in (4) is replaced by the target surface $\mathcal{R}_i$. The constraints (3)-(4) are necessary and sufficient conditions for the solid $\mathcal{M}_x$ to be volume- and surface-consistent. In the next sections, we address how to turn these constraints into a tractable optimization problem.

Comparison with Implicit-to-Implicit Methods. Before continuing, it may be helpful to compare the constraints (3)-(4) against the related class of implicit-to-implicit methods [31]. These methods penalize the difference between the distance function of the model and the scene, at each point in a volumetric domain of interest $\mathcal{V}$. Assume for simplicity that $\mathcal{S}_i = \mathcal{S}$. An implicit-to-implicit method can then be viewed as imposing the constraints

$$f_x(p) = \hat{d}_{\mathcal{S}}(p), \forall p \in \mathcal{V}, \tag{5}$$

where $\hat{d}_{\mathcal{S}}$ is an estimate of the unknown true scene distance function $d_{\mathcal{S}}$. One choice for $\hat{d}_{\mathcal{S}}$ is the distance transform of free space [31]. However, the resulting constraints either cannot guarantee volume-consistency or do not admit the distance function to the true solid as a feasible solution.

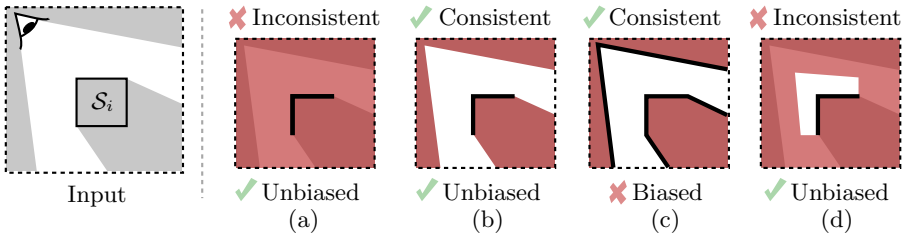


Fig. 3. Comparison between the set of feasible solids for different constraints: surface-consistency (a), surface- and volume-consistency (b), Equation (5) with $\mathcal{V}$ as $\mathcal{V}_{\text{free}}$ (c) or truncated as described in the text (d). The shaded (resp. white) region in (a)-(d) is the region of space that can (resp. must not) be occupied by a feasible solid (*i.e.* a solid whose associated distance function satisfies the constraints). The thick lines indicate where the solid is required to have a boundary according to the constraints. A feasible set is consistent if it does not contain any solid that occupies free space. A feasible set is unbiased if it contains the target solid ($\mathcal{S}_i$).

To see this, consider a point $p \in \mathcal{V}_{\text{free}}$ and the boundary point $q \in \partial\mathcal{V}_{\text{free}}$ closest to p. At p, the distance transform $d_{\mathcal{V}_{\text{free}}}(p)$ and the true distance function $d_{\mathcal{S}}(p)$ are equal if and only if $q \in \partial\mathcal{V}_{\text{free}} \cap \partial\mathcal{S}$. Where this is not the case, the equality constraint may require the boundary of the model solid to erroneously fit to a non-physical boundary, *e.g.* the boundary of the shadow behind the box in Fig. 3. Therefore, depending on the domain $\mathcal{V}$, the equality constraints (5) may not all be correct. If $\mathcal{V}$ is truncated to exclude the subset of $\mathcal{V}_{\text{free}}$ where $q \notin \partial\mathcal{V}_{\text{free}} \cap \partial\mathcal{S}$, the constraints (5) are all correct, but insufficient.

A comparison of the set of feasible solids for different constraints is illustrated in Fig. 3. The surface-consistency constraints (1) alone do not prevent a solid from occupying free space. We therefore say that the feasible set in Fig. 3 (a) is inconsistent. However, because the true solid ($\mathcal{S}_i$) is contained by the feasible set, we say that the feasible set is unbiased. If we include the volume-consistency constraints (2), the feasible set (Fig. 3 (b)) is both consistent and unbiased.

In comparison, the equality constraints (5) cannot produce a feasible set that is both consistent and unbiased. If $\mathcal{V} = \mathcal{V}_{\text{free}}$, a feasible solid is required to have a boundary where there may not be a physical scene boundary, as indicated by the thick lines in Fig. 3 (c). If $\mathcal{V}$ is truncated as described above, the feasible set (Fig. 3 (d)) is no longer biased, but has become inconsistent, as a feasible solid may occupy regions of free space where there are no constraints.

3.4 Optimization Problem Formulation

To turn the constraints (3)-(4) into a tractable optimization problem, we will use the geometric structure known as the *medial axis transform* of a solid, which is the locus of centers of balls which are maximal within the solid, along with their associated radii [45, 46]. These are also called *medial balls*. We will use the

medial axis of free space, $mat(\mathcal{V}_{\text{free}})$, which is the set of points $c \in \mathcal{V}_{\text{free}}$ that are equally close to two or more points on the boundary $\partial\mathcal{V}_{\text{free}}$. We define $\mathcal{B}(p)$ and $r_{\mathcal{B}(p)}$ as the largest ball, and its radius, centered at p that contains no point outside $\mathcal{V}_{\text{free}}$. A medial ball at $c \in mat(\mathcal{V}_{\text{free}})$ is then given by $\mathcal{B}(c)$. Note that $\mathcal{B}(p)$ is tangent to one or more points on the boundary $\partial\mathcal{V}_{\text{free}}$. Note also that, by this definition, $r_{\mathcal{B}(p)} = d_{\mathcal{V}_{\text{free}}}(p)$.

It will also be helpful to define the medial ball as seen from the surface. Let p be a point on $\partial\mathcal{V}_{\text{free}}$ and let n_p be its associated surface normal. Assume that $\partial\mathcal{V}_{\text{free}}$ is C^2-continuous such that each surface point has a unique normal. We then define c_p as the point on the medial axis which also lies on the positive extension of n_p. Note that the medial ball $\mathcal{B}(c_p)$ is tangent to p and that $r_{\mathcal{B}(c_p)}$ is the distance along n_p from p to c_p, *i.e.* $c_p = p + r_{\mathcal{B}(c_p)} n_p$.

The following results relate the medial axis to the constraints from the previous section.

Proposition 1. *If $f_x(c) \geq r_{\mathcal{B}(c)}$ for all $c \in \text{mat}(\mathcal{V}_{free})$ then $\mathcal{M}_x$ is volume-consistent with $\mathcal{V}_{free}$.*

Proposition 2. *Let $\partial\mathcal{V}_{free}$ be C^2-continuous and let $\mathcal{M}_x$ be volume-consistent with $\mathcal{V}_{free}$. If $f_x(p + t_p n_p) = t_p$ for all $p \in \mathcal{R}_i$ and some $t_p \in [0, r_{\mathcal{B}(c_p)})$, then $\mathcal{M}_x$ is surface-consistent with $\mathcal{R}_i$.*

Intuitively, these results imply that volume-consistency can be determined simply by comparing f_x against the radii of the medial balls in free space, *i.e.* without densely sampling the volume. Furthermore, given volume-consistency, surface-consistency can be determined by sampling one or more points along each surface normal. This suggests the following constrained optimization problem:

$$\min_x \qquad E(x) = \sum_{p \in \mathcal{R}_i} \left(f_x(p + t_p n_p) - t_p\right)^2 \tag{6}$$

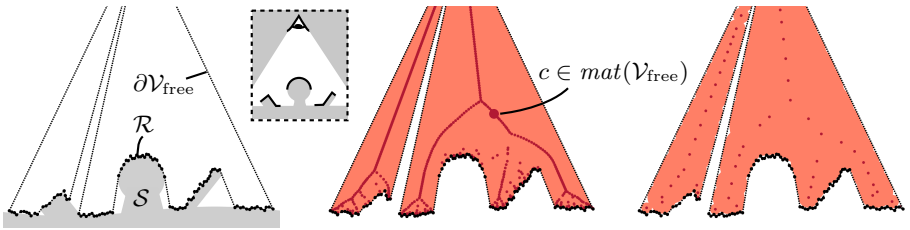
$$\text{subject to} \qquad f_x(c) \geq r_{\mathcal{B}(c)}, \forall c \in mat(\mathcal{V}_{\text{free}}) \ . \tag{7}$$

Before we describe how this can be solved in practice, we want to highlight some key properties.

Proposition 3. *If x^* is a feasible solution to (6)-(7) and $E(x^*) = 0$, then $\mathcal{M}_{x^*}$ is volume-consistent with $\mathcal{V}_{free}$ and surface-consistent with $\mathcal{R}_i$.*

Proposition 4. *The number of distance computations per iteration scales linearly by the surface area of the scene.*

The objective function and the constraints require evaluating f_x only on the visible surface $\mathcal{R}_i$ and the medial axis $mat(\mathcal{V}_{\text{free}})$. Since the medial axis is a deformation retract of $\partial\mathcal{V}_{\text{free}}$ when $\partial\mathcal{V}_{\text{free}}$ is C^2-continuous [47], the complexity of the latter scales linearly by the surface area of $\partial\mathcal{V}_{\text{free}}$. The objective function is the sum of squared data-to-model distances, but taken at different level sets as determined by t_p, and likewise scales linearly by the surface area of $\mathcal{R}_i$. The number of distance computations per iteration therefore scales linearly by the surface area of the scene rather than the domain volume.



(a) Free space boundary (b) All medial balls (c) Approximate cover

Fig. 4. A 2D example of computing an approximate cover of free space. (a): Point samples on free space boundary from a noisy range image; the "spike" represents missing range values that are treated as zero. (b): The centres of all medial balls. (c): The centres of the medial balls remaining in the approximate cover.

Proposition 5. *Volumetric data structures are not required.*

Evaluating the objective function and constraints (6)-(7) requires the centers and radii of the medial balls. It also requires that these can be parameterized by a scalar (t_p) associated with each point on the visible surface. Importantly, this information can be stored without volumetric data structures, *e.g.* by associating each point on the free space boundary with a single floating point number, representing the distance to the medial axis along the normal.

Choosing t_p Although t_p can be non-zero, we will in the remainder assume that $t_p = 0$. Thus, (6) becomes

$$E(x) = \sum_{p \in \mathcal{R}_i} f_x^2(p) \ , \tag{8}$$

which is recognized as the point-to-implicit ICP objective function [1, 21–23]. We show experimentally that this is sufficient to resolve ill-constrained parameters.

3.5 Approximate Cover

While the described method avoids a dense volumetric sampling of free space, the medial axis can still contain prohibitively many balls. We can greatly reduce the computational cost if we only need to check if the model is consistent with an approximation of free space—an approximate cover. To obtain this, we use the heuristic of [48], and detect if a ball is redundant by checking if it can be completely covered by slightly enlargening any of its neighboring balls by an amount δ. We greedily build a simplified medial axis by iteratively selecting the ball that covers the most uncovered balls when its radius is increased by δ. An example approximate cover is shown in Fig. 4.

Fig. 5. Range images and target objects (a)-(f) used in the experiments.

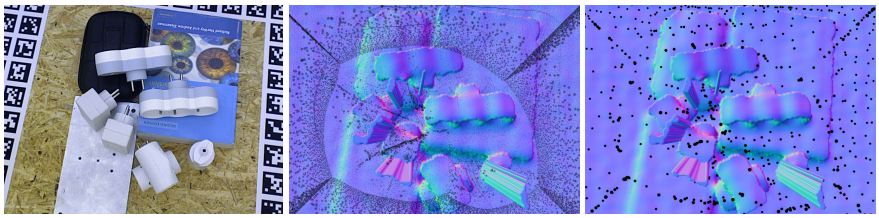
4 Experiments

We experimentally verify the following. First, that our volume-aware method constrains more parameters in partially observed scenes than the original ICP method (*i.e.* the optimization of (8) alone). Second, that our proposed optimization problem can be solved using off-the-shelf solvers. Third, that the approximate cover reduces computational cost.

4.1 Datasets and Implementation Details

As input we consider single-view range images from real and synthetic datasets. The scenes (Fig. 5) contain one or more target objects, labelled (a)-(f), which have a corresponding distance function model. The model for (a) and (b) is a cuboid with 9 parameters (pose and side lengths). The model for (c) is a discretized distance field with 7 parameters (similarity transform). The model for (d)-(f) is defined using constructive solid geometry and has 13 parameters. Parameters were standardized such that a unit change in any single parameter visually affected the shape with similar magnitude.

Our method requires the set of points on $\mathcal{R}_i$ and medial balls in $\mathcal{V}_{\text{free}}$. We implemented a pre-processing pipeline to obtain these quantities from single-view range images. For the former, backprojected range pixels provide a point sampling of the visible surface $\mathcal{R}$, from which we manually segment the subset of target points $\mathcal{R}_i$. Many methods have been proposed for estimating the medial axis from various inputs [49]. A fast and precise method is the shrinking ball method of Ma *et al.* [48], which operates on oriented point sets and computes the distance to the medial axis along each point's normal. To apply their method, we need a sufficiently dense and oriented point sampling of the free space boundary. We obtain this by connecting adjacent backprojected pixels into a piecewise linear mesh. This is similar to [50], but we include triangles at jump edges, as the shrinking ball method requires the surface to be a 2-manifold; otherwise balls may protrude into occluded space. We sample evenly-distributed points from the surface. We estimate their normals using [51] and impose a consistent orientation using the viewing direction. We apply a bilateral filter and a median filter to smoothen the range image and fill in small, isolated regions of missing values. To prevent erroneous constraints, larger missing regions are conservatively set to zero, which can cause "spikes" as seen in Fig. 6.



(a) Scene (b) All medial balls (c) Approximate cover

Fig. 6. (a): Scene used for Fig. 5 (e,f) from [52]. (b,c): Free space boundary and medial balls in original and approximate cover. The "spikes" are due to missing range values.

Table 1. Top row pair: Estimated number of constrained parameters (higher is better) and sum of singular values in parenthesis (lower is better). Bottom four row pairs: Success rate using standard deviation 0.1, 0.25, 0.5 and 1.0 (top to bottom) as initialization uncertainty.

	Box (a)	Box (b)	Head (c)	Plug (d)	Plug (e)	Plug (f)
ICP	5/9 (4.4)	3/9 (9.9)	7/7 (0.0)	7/13 (7.1)	7/13 (14.4)	7/13 (12.7)
Our	9/9 (0.0)	8/9 (2.1)	7/7 (0.0)	9/13 (4.0)	9/13 (5.6)	9/13 (6.0)
ICP	100%	100%	100%	87%	98%	77%
Our	100%	100%	100%	100%	100%	100%
ICP	100%	100%	89%	54%	69%	55%
Our	100%	100%	98%	87%	90%	92%
ICP	86%	88%	48%	22%	26%	22%
Our	99%	100%	74%	39%	58%	42%
ICP	62%	47%	9%	3%	5%	1%
Our	100%	100%	26%	15%	16%	8%

4.2 Ability to Resolve Ill-Constrained Parameters

We quantify how constrained the model parameters are at the global solution. We acquire an initial fit using a global search method. We then perform a random walk of length $k = 50$, where at each step $i = 1...k$ the current parameters are perturbed by a uniformly drawn vector $\delta \in [-\sigma, \sigma]^{|\mathcal{X}|}$, $\sigma = 0.1$, and re-optimized from the perturbed position. We perform $N = 100$ random walks, which gives a matrix of solutions $X = [x_1 \cdots x_N]$ from each random walk. As a measure of the number of constrained parameters, we use the rank of X, estimated as the number of singular values less than σ. If a parameter is well-constrained, it should return to its original value after re-optimization. Hence, singular values larger than σ indicates that one or more parameters are not constrained. The results in Table 1 (top) show that our method constrained more parameters (except for the Head model), thus indicating that it successfully utilizes free space information to resolve ill-constrained parameters.

Table 2. Number of surface points and number of medial balls, before and after computing the approximate cover, for the scenes shown in Fig. 5.

	(a,b)	(c)	(d)	(e,f)
Surface points ($\|\mathcal{R}\|$)	686	15 806	6 827	8 730
Medial balls (all)	48 386	32 019	42 013	40 193
Medial balls (approximate cover)	432	272	320	479

4.3 Ability to Use Off-the-Shelf Solvers

We compare the success rate of our method against ICP, using an off-the-shelf solver. For both methods, we use Matlab's Sequential Quadratic Programming (SQP) implementation. As a proxy for success, we use the rotational parameters at the end of optimization compared with ground-truth, taking possible symmetries into account. If the trace of the relative rotation matrix for a given solution is higher than a threshold (2.99), it is considered successful. We count successful runs among 100 independent trials, drawing initial parameters from a uniform distribution around ground-truth parameters, repeated for four different standard deviations. The results in Table 1 show that our method had equal or higher success rate, indicating that our optimization problem is amenable to off-the-shelf solvers.

4.4 Reduction of Computational Cost by the Approximate Cover

Table 2 shows the number of surface points $|\mathcal{R}|$ and the number of inequality constraints, before and after computing the approximate cover. We set the ball enlargement parameter δ equal to 10 times the shortest distance between any pair of neighboring point samples of $\mathcal{R}$. The approximate cover decreased the number of inequality constraints (and thereby the number of evaluations of f_x per iteration) by $98.8\% - 99.2\%$, or about two orders of magnitude. The results in Table 1, which were obtained using the approximate cover, shows that the simplified constraints were still sufficient to resolve ill-constrained parameters.

5 Conclusion

We have presented a method for efficiently incorporating free space constraints in the classical Iterative Closest Point algorithm. It is able to resolve parameters that are ill-constrained by partial surface observations and supports any implicit model for which the distance to its surface can be computed.

Presently, our method only allows for outliers in the form of alternate structures or corrupt range values (which can be set to zero). Such outliers decrease the extent of utilizable free space. How to handle incorrect free space observations, which introduces bias, is an important question for future work.

Acknowledgments. This work is partly supported by the Research Council of Norway through the Centre of Excellence funding scheme, project number 223254, NTNU AMOS.

References

1. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992) 239–256
2. Solina, F., Bajcsy, R.: Recovery of parametric models from range images: The case for superquadrics with global deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990) 131–147
3. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. International Journal of Computer Vision **38** (2000) 199–218
4. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real-time human pose tracking from range data. In: European Conference on Computer Vision, Springer (2012) 738–751
5. Schmidt, T., Newcombe, R.A., Fox, D.: DART: dense articulated real-time tracking. In: Robotics: Science and Systems. (2014)
6. Tagliasacchi, A., Schröder, M., Tkach, A., Bouaziz, S., Botsch, M., Pauly, M.: Robust articulated-ICP for real-time hand tracking. In: Computer Graphics Forum. Volume 34., Wiley Online Library (2015) 101–114
7. Tkach, A., Pauly, M., Tagliasacchi, A.: Sphere-meshes for real-time hand modeling and tracking. ACM Transactions on Graphics (ToG) **35** (2016) 1–11
8. Tam, G.K., Cheng, Z.Q., Lai, Y.K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.F., Rosin, P.L.: Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. IEEE Transactions on Visualization and Computer Graphics **19** (2012) 1199–1217
9. Zhou, Q.Y., Park, J., Koltun, V.: Fast global registration. In: European Conference on Computer Vision, Springer (2016) 766–782
10. Yang, J., Li, H., Campbell, D., Jia, Y.: Go-ICP: A globally optimal solution to 3D ICP point-set registration. IEEE Transactions on Pattern Analysis and Machine Intelligence **38** (2016) 2241–2254
11. Yang, H., Carlone, L.: In perfect shape: Certifiably optimal 3d shape reconstruction from 2D landmarks. In: IEEE Conference on Computer Vision and Pattern Recognition. (2020) 621–630
12. Zhou, X., Zhu, M., Leonardos, S., Daniilidis, K.: Sparse representation for 3D shape estimation: A convex relaxation approach. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2016) 1648–1661
13. Hart, J.C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. The Visual Computer **12** (1996) 527–545
14. Barr, A.H.: Global and local deformations of solid primitives. In: SIGGRAPH, ACM (1984) 21–30
15. Pasko, A., Adzhiev, V., Sourin, A., Savchenko, V.: Function representation in geometric modeling: concepts, implementation and applications. The Visual Computer **11** (1995) 429–446
16. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH, ACM (1996) 303–312
17. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: ISMAR, IEEE (2011) 127–136

18. Jones, M.W., Baerentzen, J.A., Sramek, M.: 3D distance fields: A survey of techniques and applications. IEEE Transactions on Visualization and Computer Graphics **12** (2006) 581–599
19. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: IEEE Conference on Computer Vision and Pattern Recognition. (2019) 165–174
20. Chen, Y., Medioni, G.: Object modelling by registration of multiple range images. Image and Vision Computing **10** (1992) 145–155
21. Fitzgibbon, A.W.: Robust registration of 2D and 3D point sets. Image and Vision Computing **21** (2003) 1145–1153
22. Mitra, N.J., Gelfand, N., Pottmann, H., Guibas, L.: Registration of point cloud data from a geometric optimization perspective. In: Symposium on Geometry Processing, ACM (2004) 22–31
23. Pottmann, H., Huang, Q.X., Yang, Y.L., Hu, S.M.: Geometry and convergence analysis of algorithms for registration of 3d shapes. International Journal of Computer Vision **67** (2006) 277–296
24. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: International Conference on 3D Digital Imaging and Modeling. (2001) 145–152
25. Pentland, A.: Recognition by parts. In: Technical Report 406, SRI International. (1986)
26. Fayolle, P.A., Pasko, A.: An evolutionary approach to the extraction of object construction trees from 3D point clouds. Computer-Aided Design **74** (2016) 1–17
27. Yezzi, A., Soatto, S.: Stereoscopic segmentation. International Journal of Computer Vision **53** (2003) 31–43
28. Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A., Silva, C.T.: A survey of surface reconstruction from point clouds. In: Computer Graphics Forum, Wiley Online Library (2017) 301–329
29. Xiao, J., Furukawa, Y.: Reconstructing the world's museums. International Journal of Computer Vision **110** (2014) 243–258
30. Du, T., Inala, J.P., Pu, Y., Spielberg, A., Schulz, A., Rus, D., Solar-Lezama, A., Matusik, W.: InverseCSG: Automatic conversion of 3D models to CSG trees. In: SIGGRAPH Asia 2018 Technical Papers, ACM (2018) 213
31. Slavcheva, M., Kehl, W., Navab, N., Ilic, S.: SDF-2-SDF registration for real-time 3D reconstruction from RGB-D data. International Journal of Computer Vision **126** (2018) 615–636
32. Whitaker, R.T., Gregor, J.: A maximum-likelihood surface estimator for dense range data. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 1372–1387
33. Gargallo, P., Prados, E., Sturm, P.: Minimizing the reprojection error in surface reconstruction from images. In: IEEE International Conference on Computer Vision. (2007) 1–8
34. Ganapathi, V., Plagemann, C., Koller, D., Thrun, S.: Real time motion capture using a single time-of-flight camera. In: IEEE Conference on Computer Vision and Pattern Recognition. (2010) 755–762
35. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1106–1113
36. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: IEEE Conference on Computer Vision and Pattern Recognition. (2020) 3504–3515

37. Loper, M.M., Black, M.J.: Opendr: An approximate differentiable renderer. In: European Conference on Computer Vision, Springer (2014) 154–169
38. Terzopoulos, D., Witkin, A., Kass, M.: Constraints on deformable models: Recovering 3d shape and nonrigid motion. Artificial Intelligence **36** (1988) 91–123
39. Prisacariu, V.A., Segal, A.V., Reid, I.: Simultaneous monocular 2D segmentation, 3D pose recovery and 3D reconstruction. In: Asian Conference on Computer Vision, Springer (2012) 593–606
40. Tsai, A., Yezzi, A., Wells, W., Tempany, C., Tucker, D., Fan, A., Grimson, W.E., Willsky, A.: A shape-based approach to the segmentation of medical imagery using level sets. IEEE Transactions on Medical Imaging **22** (2003) 137–154
41. Liu, S., Zhang, Y., Peng, S., Shi, B., Pollefeys, M., Cui, Z.: Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In: IEEE Conference on Computer Vision and Pattern Recognition. (2020) 2019–2028
42. Tagliasacchi, A., Olson, M., Zhang, H., Hamarneh, G., Cohen-Or, D.: Vase: Volume-aware surface evolution for surface reconstruction from incomplete point clouds. In: Computer Graphics Forum. Volume 30., Wiley Online Library (2011) 1563–1571
43. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3D reconstruction in function space. In: IEEE Conference on Computer Vision and Pattern Recognition. (2019) 4460–4470
44. Witkin, A.P., Heckbert, P.S.: Using particles to sample and control implicit surfaces. In: SIGGRAPH, ACM (1994) 269–277
45. Siddiqi, K., Pizer, S.: Medial representations: mathematics, algorithms and applications. Volume 37. Springer Science & Business Media (2008)
46. Blum, H., et al.: A transformation for extracting new descriptors of shape. Models for the Perception of Speech and Visual Form **19** (1967) 362–380
47. Wolter, F.E.: Cut locus and medial axis in global shape interrogation and representation. In: MIT Design Laboratory Memorandum 92-2 and MIT Sea Grant Report. (1992)
48. Ma, J., Bae, S.W., Choi, S.: 3D medial axis point approximation using nearest neighbors and the normal field. The Visual Computer **28** (2012) 7–19
49. Tagliasacchi, A., Delame, T., Spagnuolo, M., Amenta, N., Telea, A.: 3D skeletons: A state-of-the-art report. In: Computer Graphics Forum. Volume 35., Wiley Online Library (2016) 573–597
50. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: SIGGRAPH, ACM (1994) 311–318
51. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: SIGGRAPH, ACM (1992) 71–78
52. Hodan, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In: IEEE Winter Conference on Applications of Computer Vision, IEEE (2017) 880–888