

# Minimal Free Space Constraints for Implicit Distance Bounds

Simen Haugo and Annette Stahl

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

**Abstract.** A general approach for fitting implicit models to sensor data is to optimize an objective function measuring the quality of the fit. The objective function often involves evaluating the model’s implicit function at several points in space. When the model is expensive to evaluate, the number of points can become a bottleneck, making the use of volumetric information, such as free space constraints, challenging. When the model is the Euclidean distance function to its surface, previous work has been able to integrate free space constraints in the optimization problem, such that the number of distance computations is linear in the scene’s surface area. Here, we extend this work to only require the model’s implicit function to be a bound of the Euclidean distance. We derive necessary and sufficient conditions for the model to be consistent with free space. We validate the correctness of the derived constraints on implicit model fitting problems that benefit from the use of free space constraints.

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

## 1 Introduction

Infinite resolution, trivial support for constructive modeling operations and the existence of closed-form expressions for a rich variety of primitives, has sparked interest in the use of implicit models as a representation of scenes and objects. However, the lack of an explicit surface and the open-ended domain (which may be non-rigid and even non-differentiable) poses major challenges for their use in model fitting. A general approach, based on optimizing an objective function measuring the quality of the fit, often involves evaluating the associated implicit function at multiple points in space. When the implicit function is expensive to evaluate, the number of evaluations can become a bottleneck, making the use of volumetric free space constraints (space observed to be empty) [1] impractical.

When the model is the Euclidean distance function to its surface, recent work has been able to integrate free space constraints while ensuring that the number of function evaluations remains linear in the scene’s surface area [2]. Often, the Euclidean distance function is too expensive to compute efficiently, and a distance bound is used instead [3].

**Contribution.** We present an extension of [2] that only requires the model’s implicit function to be a bound of the Euclidean distance to its surface. We derive a minimal set of necessary and sufficient conditions for the model to be consistent with free space. We validate the correctness of the derived constraints on implicit model fitting problems that benefit from the use of free space information.

## 2 Related work

**Distance Functions.** Implicit models represent geometric objects indirectly by a function  $f(p)$ , which maps a given point  $p \in \mathbb{R}^3$  to a scalar that, in the simplest case, indicates whether  $p$  is inside or outside the object. These have numerous applications in solid modeling [4–9], robotics [10, 11], graphics [3, 12], reverse engineering [13, 14] and machine learning [15–17]. A special case is when  $f(p)$  is equal to, or approximates, the (signed) Euclidean distance function  $d(p)$ , which is the distance from  $p$  to the closest point on the surface, with the sign indicating whether  $p$  is inside or outside. Discretized distance functions stored in volumetric grids are widely used in 3D reconstruction [1, 18] and can easily be computed from explicit models using a distance transform [19]. Neurally-defined distance functions have recently received interest in the machine learning community as a compact and end-to-end learnable shape representation [20]. The Euclidean distance can be computed from an arbitrary implicit function by solving a constrained optimization problem, although this requires an initial estimate of the closest point on the surface to guarantee convergence [21].

A rich set of modeling operations and primitives have also been proposed for constructively defining distance-like functions [3, 4]. Although the Euclidean distance function is challenging to define constructively, a bounding function can often be obtained instead [3]. In graphics, a desirable condition on the bounding function is that it does not overestimate the true distance. When  $f$  is Lipschitz, this can be ensured by identifying a Lipschitz constant,  $\lambda > 0$ , such that  $|f| \leq \lambda|d|$ . The resulting “signed distance bound”,  $\lambda^{-1}f$ , defines everywhere an “unbounding sphere” which is guaranteed to be intersection-free. Lipschitz functions and constants have been derived for various modeling operations and primitives [22, 3]. Our work uses Lipschitz functions, but we identify the additional condition that  $\lambda^{-1}|f|$  should be bounded from below by a non-decreasing function of  $|d|$ . This lets us also avoid the error caused by underestimation of the true distance, which we use together with the Lipschitz condition to derive a minimal set of necessary and sufficient free space constraints.

**Free Space Constraints.** Besides points on the scene’s surface, vision systems may also provide information about visibility. For example, a range measurement from a laser scanner says that not only is there a surface that far from the sensor, but also that there can be no surface in-between. The resulting “free space constraints” play a central role in 3D reconstruction [1, 23–25], and are used in articulated and non-rigid model fitting to resolve model parameters that are ill-constrained from surface measurements alone [26–34].

Several works have addressed the computational challenges involved in the use of free space constraints for model fitting. When the model surface or a bound is available in explicit form, previous works have formulated efficient silhouette-based or symmetric objective functions [29, 31, 33, 34]. For implicit models, when the model is equal to the Euclidean distance function to its surface, recent work [2] has been able to integrate free space constraints, while keeping the number of function evaluations linear in the scene’s surface area. We extend the previous work [2] to only require a distance bound, thus enabling the use of more diverse implicit models without the need to compute the exact Euclidean distance.

### 3 Theory and Method

Our work builds on results from [2]. We summarize their main findings in Sec. 3.1. We present our extension in the subsequent sections.

#### 3.1 Notation and Problem Formulation

Let  $\mathcal{S}$  be a solid representing the scene and let  $\mathcal{V}$  be a domain of interest. Free space  $\mathcal{V}_{\text{free}}$  is a closed subset of  $\mathcal{V}$  that is observed to be empty. The free space boundary  $\partial\mathcal{V}_{\text{free}}$  is the boundary between free and unobserved space. The visible surface  $\mathcal{R}$  is the observed subset of the physical scene boundary  $\partial\mathcal{S}$ . Points in  $\mathbb{R}^3$  are denoted  $c$ ,  $p$  and  $q$ . A ball with center  $c$  and radius  $r$  is denoted  $(c, r)$ . 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.

A geometric model is given as a function  $f_x(p) : \mathbb{R}^3 \times \mathcal{X} \rightarrow \mathbb{R}$  and we seek to estimate the parameters  $x \in \mathcal{X}$  such that the solid  $\mathcal{M}_x = \{p \in \mathbb{R}^3 : f_x(p) \leq 0\}$  is consistent with the visible surface and with free space. Formally:

$$f_x(p) = 0, \quad \forall p \in \mathcal{R}, \quad (1)$$

$$f_x(p) > 0, \quad \forall p \in \text{int } \mathcal{V}_{\text{free}}. \quad (2)$$

These conditions can be turned into a constrained optimization problem [2]

$$\min_x \quad E(x) = \sum_{p \in \mathcal{R}} f_x(p)^2 \quad (3)$$

$$\text{subject to} \quad f_x(c) \geq r(c), \quad \forall c \in \mathcal{I}, \quad (4)$$

where  $\mathcal{I}$  is a set of points in  $\mathcal{V}_{\text{free}}$  and  $r(c)$  is the radius of the largest ball at  $c$  that is empty with respect to  $\mathcal{V}_{\text{free}}$  (*i.e.* its interior contains no point of  $\partial\mathcal{V}_{\text{free}}$ ). Thus, each constraint in (4) defines a ball  $(c, r(c))$  which  $\mathcal{M}_x$  must not intersect. When  $f_x = d_{\mathcal{M}_x}$ , previous work [2] has shown that the medial axis [35] of  $\mathcal{V}_{\text{free}}$  provides a minimal set of necessary and sufficient constraints  $\mathcal{I}$  to ensure that a feasible solution of (3)-(4) is consistent with free space, *i.e.* satisfies (2).

### 3.2 Free Space Constraints for Distance Bounds

When  $f_x \neq d_{\mathcal{M}_x}$ , the constraints (4) are neither sufficient nor necessary. To resolve this, we will assume that  $f_x$  satisfies upper and lower bounds of the form

$$g(d_{\mathcal{M}_x}(c)) \leq \lambda^{-1} f_x(c) \leq d_{\mathcal{M}_x}(c), \quad \forall (c, x) \in (\mathbb{R}^3, \mathcal{X}) : f_x(c) > 0, \quad (5)$$

where  $\lambda$  is a Lipschitz constant of  $f_x$  over the domain  $\mathbb{R}^3 \times \mathcal{X}$  and  $g(d) : \mathbb{R} \rightarrow \mathbb{R}$  is a non-decreasing function with  $g(d) > 0$  for  $d > 0$ . We motivate the use and existence of bounds of this form in Sec. 3.4. In this section, we derive a set of necessary and sufficient free space constraints when the true distance  $d_{\mathcal{M}_x}$  is unavailable, but  $f_x$ ,  $g$  and  $\lambda$  are known. For simplicity in the derivation, we will omit subscripts and write  $f = f_x$ ,  $d = d_{\mathcal{M}_x}$  and  $\mathcal{M} = \mathcal{M}_x$ .

**Intuition.** Consider the constraint  $\lambda^{-1} f(c) \geq r(c)$ . Observe that its satisfaction implies  $d(c) \geq r(c)$ . Hence, its satisfaction is a sufficient condition for  $\mathcal{M}$  to not intersect the ball  $(c, r(c))$ . However, as  $f$  may underestimate the true distance, its violation does not imply  $d(c) < r(c)$ . Its satisfaction is therefore not a necessary condition. Suppose there exists a function  $g$ , such that  $g(d) \leq \lambda^{-1} f$ . Consider the constraint  $\lambda^{-1} f(c) \geq g(r(c))$ , obtained by “shrinking” the ball according to  $g$ . Observe that its satisfaction implies  $d(c) \geq g(r(c))$ . Hence, the “shrunk” ball  $(c, g(r(c)))$  is intersection-free. Observe also that violation of the constraint implies  $g(d(c)) < g(r(c))$  which, if  $g$  is non-decreasing, implies  $d(c) < r(c)$ . Hence,  $\mathcal{M}$  intersects the “original” ball  $(c, r(c))$ .

Thus, if the constraint  $\lambda^{-1} f(c) \geq g(r(c))$  is satisfied, the model is guaranteed to be outside the shrunk ball  $(c, g(r(c)))$ . If the constraint is violated, the model intersects the original ball  $(c, r(c))$ , though not necessarily the shrunk ball. This observation leads to Proposition 1.

**Proposition 1.** *Let  $f_x : \mathbb{R}^3 \times \mathcal{X} \rightarrow \mathbb{R}$  be Lipschitz over the domain  $\mathbb{R}^3 \times \mathcal{X}$  and let  $\lambda > 0$  be an associated Lipschitz constant. Let  $g : \mathbb{R} \rightarrow \mathbb{R}$  be a non-decreasing function, with  $g(d) > 0$  for  $d > 0$ , such that  $\forall (x, c) \in (\mathcal{X}, \mathcal{V}_{free}) : g(d_{\mathcal{M}_x}(c)) \leq \lambda^{-1} f_x(c)$ . Let  $r(c)$  be the radius of the largest ball at  $c \in \mathcal{V}_{free}$  that is empty with respect to  $\mathcal{V}_{free}$ . Then*

$$\lambda^{-1} f_x(c) \geq g(r(c)), \quad \forall c \in \mathcal{V}_{free}, \quad (6)$$

*are necessary and sufficient conditions for  $f_x$  to satisfy the free space consistency conditions (2).*

### 3.3 Minimal Set of Necessary and Sufficient Constraints.

While the constraints (6) are necessary and sufficient, many of the associated balls can be fully contained by a larger ball. The constraints are thereby highly redundant. To obtain a minimal set of constraints, we consider each line segment connecting a point  $p \in \partial \mathcal{V}_{free}$  with the point  $c_p$  that lies on the intersection of the boundary normal  $n_p$  and the medial axis of  $\mathcal{V}_{free}$ . We generate the shortest

sequence of (shrunk) balls that cover the line segment without overlapping, by letting  $c_{p,i} = p + t_{p,i}n_p$  and solving

$$t_{p,i} + g(r(c_{p,i})) = t_{p,i-1} - g(r(c_{p,i-1})), \quad i = 1, 2, \dots, \quad (7)$$

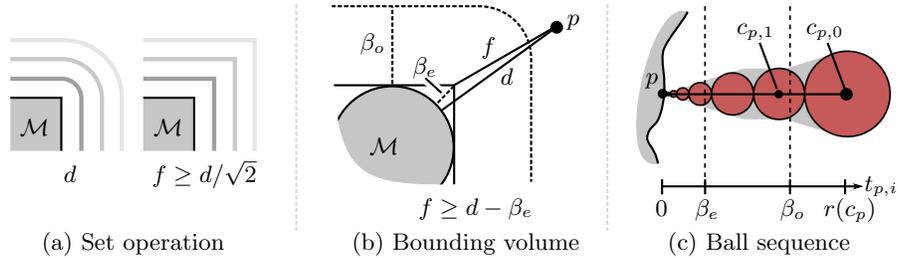
successively for  $t_{p,i}$  starting with  $t_{p,0} = r(c_p)$ . This equation requires ball  $i$  in the sequence to be tangent to ball  $i - 1$ . Depending on  $g$ , this may produce an infinite sequence of balls, which may be truncated in different ways. For example, by stopping or setting  $t_{p,i}$  to its limit value (0) once below a desired tolerance  $t_{\min}$ . If  $g(d) = d$ , the sequence has a length of one, as expected. When the sequences for all points  $p \in \partial\mathcal{V}_{\text{free}}$  are combined, we obtain a minimal set of necessary and sufficient constraints.

**Comparison with Sphere Tracing.** The reader may notice a resemblance between the above and a technique used for rendering implicit surfaces known as sphere tracing [3]. One difference is that sphere tracing generates sample points based on evaluating the model, whereas the above generates sample points based on the input data. When sphere tracing is used for model fitting, it produces different sample points depending on the current model parameters. The above sample points remain the same during the optimization process. This observation enables further improvements, which we describe below (approximate cover). Sphere tracing requires the upper bound  $\lambda^{-1}f \leq d$ , while we also require the lower bound  $g(d) \leq \lambda^{-1}f$ . If  $g(d) = d$ , we only generate a single sample point, while sphere tracing may still take multiple steps to converge.

**Approximate Cover.** The minimal set may still exhibit substantial overlap between the constraints, which can lead to unnecessary computations. Because of our formulation of the bounds (5), the free space constraints are fully determined by the input data. Similar to [2], we can therefore pre-compute an “approximate cover” of free space—a simplified set of constraints that check for consistency with an approximation of free space. We may use the same heuristic as in [2] and [36], in which redundant balls are identified by checking if a given ball can be completely covered by slightly enlarging any of its neighboring balls. Like [2], we can greedily build an approximate cover by iteratively selecting the ball which covers the most uncovered balls, when its radius is increased by  $\delta$ .

### 3.4 Upper and Lower Bounds for Some Implicit Models

The upper bound in (5) requires that  $f$  can be scaled to not overestimate the true distance  $d$ . Such functions are called signed distance bounds [3], although we only require the bound to hold when  $f > 0$ . When  $f$  is Lipschitz, dividing by a Lipschitz constant provides a distance bound. Several primitives and constructive modeling operations that yield functions satisfying this property, along with associated Lipschitz constants, have been described in the literature, including set operations, linear transformations, tapering and twisting [22, 3].



**Fig. 1.** (a,b): Examples of the lower bound  $f \geq g(d)$  for a max-based intersection and a bounding volume. (c): Sequence of balls produced by (7) for an example bound  $g(d)$ .

Below, we describe two general forms of the function  $g$  required for the lower bound, based on a consideration of two common implicit modeling techniques: set operations and bounding volume hierarchies.

**Set Operations.** In constructive geometry, solids can be defined by successively applying set operations (union, intersection, difference) to primitives. Given the distance functions to two solids, a cheap distance-like function to their union, intersection or difference can be computed by the min and max operators [37]. While these do not yield the Euclidean distance to the resulting solid everywhere, they are guaranteed to not overestimate it [3]. As an example, the intersection between two planes is shown in Fig. 1 (a), along with isolines of the Euclidean distance  $d$  and its max-based approximation  $f$ . Here,  $f$  and  $d$  are equal everywhere except when the closest point on  $\partial\mathcal{M}$  is at the corner. The error between  $f$  and  $d$  grows proportionally to  $d$ , with largest growth on the diagonal extending away from the corner, where  $d = \sqrt{2}f$ . This gives a lower bound of the form

$$g(d) = \alpha d \quad (8)$$

with  $\alpha = 1/\sqrt{2}$ .

**Bounding Volume Hierarchies.** Bounding volumes are often used to avoid computation of geometric details when the evaluation point is sufficiently far away from the surface [3]. Fig. 1 (b) shows a single-stage bounding volume hierarchy, where the object  $\mathcal{M}$  is bounded by a box. A bounding volume hierarchy first computes the distance to the box. If the result is greater than a threshold  $\beta_o$ , it is returned, terminating the computation of  $f$ . Otherwise, computation proceeds with the distance to  $\mathcal{M}$ . The threshold  $\beta_o$  is typically chosen by the model designer to be at the point where the distance to the bounding volume ceases to be a good approximation of the distance to the bounded object.

Consider a single-stage bounding volume hierarchy. Let  $d'$  and  $d$  be the Euclidean distance function to the bounding volume and the bounded object, re-

spectively. Suppose  $f = d'$  when  $d' > \beta_o$  and  $f = d$  otherwise. Then, the error between  $f$  and  $d$  is either zero or bounded by the constant  $\beta_e < \beta_o$ , which is the maximum difference between  $d$  and  $d'$  at the level-set  $d'(p) = \beta_o$ :

$$\beta_e = \max_{p \in d'^{-1}(\beta_o)} d(p) - d'(p). \quad (9)$$

A lower bound  $g$  is

$$g(d) = \begin{cases} d & \text{if } d < \beta_o, \\ \beta_o & \text{if } \beta_o \leq d < \beta_o + \beta_e, \\ d - \beta_e & \text{otherwise,} \end{cases} \quad (10)$$

where the constant value in the transition region  $\beta_o \leq d < \beta_o + \beta_e$  ensures that  $g$  is non-decreasing.

**Proportional or Constant Error.** We observe that the error between  $f$  and  $d$  grew proportionally in the case of a set operation and was bounded by a constant in the case of a bounding volume hierarchy. This lead to two general forms of the function  $g$ . These can naturally be combined. An example sequence of balls produced by (7) for a function  $g$  that combines both of the above is shown in Fig. 1 (c).

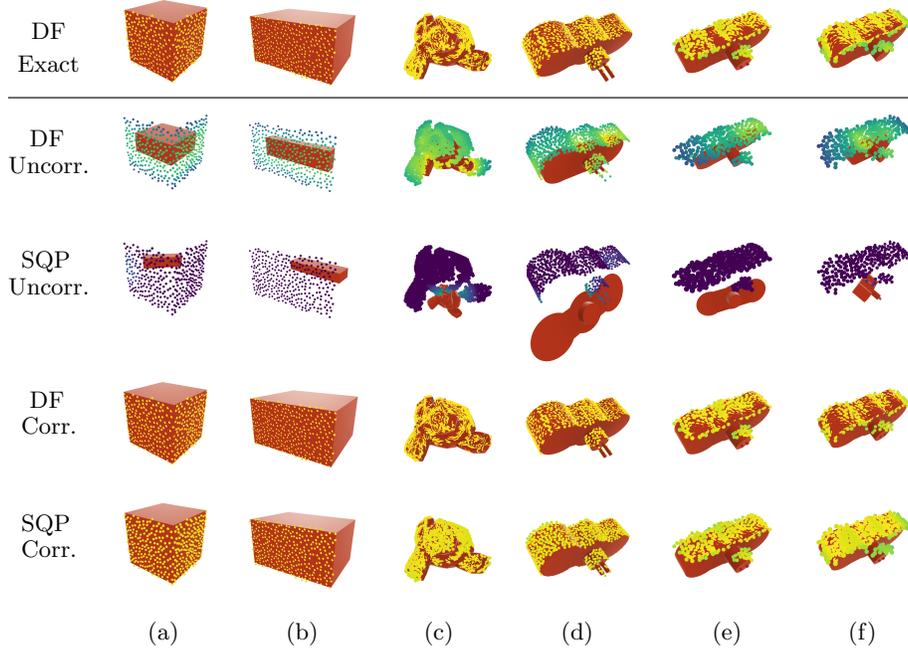
## 4 Experiments

We experimentally validate the proposed constraints. First, we show that the solution obtained using the ‘‘uncorrected’’ free space constraints of [2] is incorrect when  $f_x \neq d_{\mathcal{M}_x}$ . Second, we show that the use of the proposed ‘‘corrected’’ constraints (6) can resolve ill-constrained parameters when the bounds (5) are available. Finally, we study the computational cost of the proposed minimal set of constraints and the approximate cover.

**Datasets and Pre-processing.** We use the dataset in [2], comprised of single-view range images of objects with corresponding Euclidean distance functions. To obtain the necessary quantities ( $\mathcal{R}$  and  $\mathcal{I}$ ) for the constrained optimization problem (3)-(4), we use a pre-processing pipeline similar to [2]. For the corrected constraints,  $\mathcal{I}$  is generated as described in Sec. 3.3. All results use an approximate cover for both methods. To solve the constrained optimization problem, we consider Matlab’s implementation of Sequential Quadratic Programming (SQP). We also consider a non-smooth exact penalty method that replaces the constrained problem by the unconstrained problem

$$\min_x \sum_{p \in \mathcal{R}} f_x(p)^2 + \mu \sum_{(c,r) \in \mathcal{I}} (\min(0, f_x(c) - r))^2, \quad (11)$$

which we solve using a derivative-free (DF) solver.

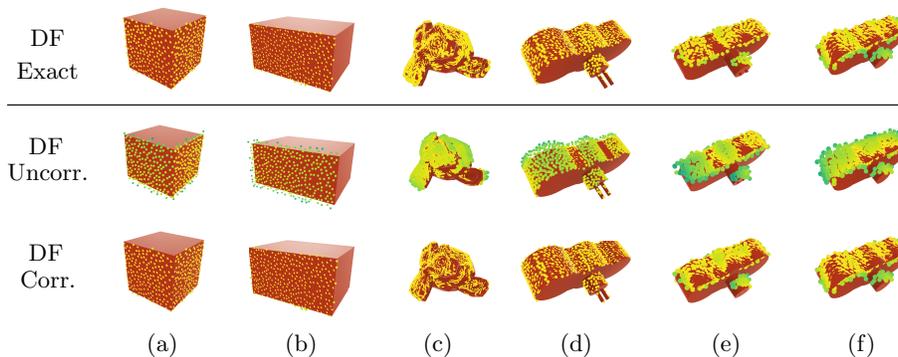


**Fig. 2.** Effect of proportional error with  $\alpha = 0.7$ . (*DF Exact*): Global solution for the exact distance models using the derivative-free solver. (*DF/SQP Uncorr./Corr.*): Global solution for the distance bound models with uncorrected or corrected free space constraints. Points are on the visible surface  $\mathcal{R}$ . Point color indicates distance to the model, brighter being closer.

#### 4.1 Effect of Error Between $f_x$ and $d_{\mathcal{M}_x}$

**Proportional Error.** We study how a proportional error between  $f_x$  and  $d_{\mathcal{M}_x}$  affects the global solution. We simulate a proportional error by replacing each model's distance function by  $f'_x(p) = \alpha f_x(p)$ , for  $\alpha < 1$ . We compare the global solution of both solvers (DF and SQP) with and without corrected free space constraints. To find the global solution, we initialize  $x$  at ground-truth with a small random offset and run the given solver until convergence. We pick the solution with the best objective function value over 100 trials.

**Constant Error.** We repeat the above experiment, adding instead a single-stage bounding volume hierarchy such that  $f'_x(p) = f_{x,\text{bound}}(p)$  if  $f_{x,\text{bound}}(p) > \beta_o$  and  $f'_x(p) = f_x(p)$  otherwise, where  $f_{x,\text{bound}}$  is taken as the distance to a constant offset surface:  $f_{x,\text{bound}} = f_x(p) - \beta_e$ . Because the switch statement makes the model non-differentiable, we only report results using the derivative-free solver.



**Fig. 3.** Effect of constant error with  $\beta_o = 0.03$  and  $\beta_e = 0.01$ . (*DF Exact*): Global solution for the exact distance models using the derivative-free solver. (*DF Uncorr./Corr.*): Global solution for the distance bound models with uncorrected or corrected free space constraints. Points are on the visible surface  $\mathcal{R}$ . Point color indicates distance to the model, brighter being closer.

**Table 1.** Mean Euclidean distance from  $\mathcal{R}$  to  $\partial\mathcal{M}$  at the global solution obtained with the derivative-free method. (*Exact*): Results for the exact distance models. (*Bound, uncorr./corr.*): Results for the distance bound models described in Sec. 4.1: proportional error (\*) and constant error (\*\*), with uncorrected or corrected constraints.

	Box (a)	Box (b)	Head (c)	Plug (d)	Plug (e)	Plug (f)
Exact	0.38	0.18	0.27	0.28	1.67	1.27
Bound*, uncorr.	17.79	18.29	10.44	11.38	16.37	17.24
Bound*, corr.	0.40	0.21	0.25	0.27	1.37	1.30
Bound**, uncorr.	2.33	3.09	2.29	2.35	2.78	3.12
Bound**, corr.	0.29	0.24	0.19	0.24	1.24	1.21

**Results.** The results in Fig. 2 and Fig. 3 show that when  $f_x \neq d_{\mathcal{M}_x}$ , uncorrected free space constraints lead to a global solution that is different from the true solution obtained using the exact distance models ( $f_x = d_{\mathcal{M}_x}$ ). They also show that the correction with the distance bound models yields a global solution that is visually similar to the true solution. Table 1 quantifies the difference using the mean Euclidean distance from the visible surface to the model. It shows that the mean distance is one or two orders of magnitude greater without the correction and within 5-20 percent of true solution with the correction.

## 4.2 Ability to Resolve Ill-Constrained Parameters

We quantify how constrained the model parameters are at the global solution. Initializing  $x$  to ground-truth, we perform a random walk of length  $k = 50$ , where

**Table 2.** Estimated number of constrained parameters out of all parameters and sum of singular values in parenthesis. (*Exact*): Results for the exact distance models. (*Bound, corr.*): Results for the distance bound models with corrected constraints.

	Box (a)	Box (b)	Head (c)	Plug (d)	Plug (e)	Plug (f)
Exact	9/9 (0.0)	8/9 (2.1)	7/7 (0.0)	9/13 (4.0)	9/13 (5.6)	9/13 (6.0)
Bound, corr.	9/9 (0.0)	8/9 (2.2)	7/7 (0.0)	9/13 (3.6)	9/13 (5.9)	9/13 (5.8)

at each step  $i = 1 \dots k$  the current parameters are perturbed by a uniformly drawn vector  $\delta \in [-\sigma, \sigma]^{|X|}$ ,  $\sigma = 0.1$ , and re-optimized (using the derivative-free solver) from the perturbed position. We perform  $N = 100$  random walks, which gives a matrix of solutions  $X = [x_1 \dots x_N]$  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  $\sigma$ . If a parameter is well-constrained, it should return to its original value after re-optimization. Singular values larger than  $\sigma$  indicates that one or more parameters were not constrained.

**Results.** Table 2 shows that the number of constrained parameters at the global solution for the distance bound models, with corrected constraints, is the same as for the exact distance models. This supports the visual results in Fig. 2 and Fig. 3 (bottom rows), where the solution is seen to acquire a fit that is only possible by the use of free space constraints (*e.g.* the dimensions of the Box (a) are correctly estimated, despite only two out the six faces being visible).

### 4.3 Computational Cost

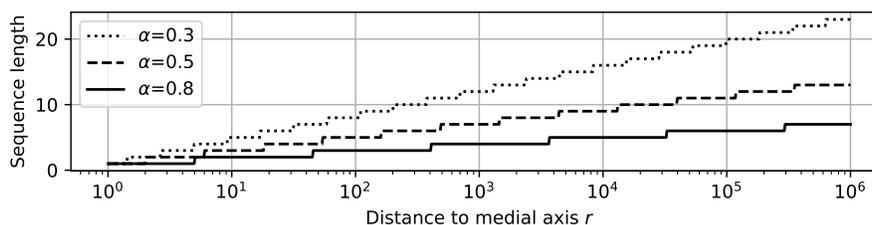
Table 3 reports the number of free space constraints used by [2] and our method. It can be seen that our extension yields an increase in the number of constraints. The number of constraints produced by [2] scales linearly by the scene’s surface area. A natural question is how our method scales as the scene volume grows.

To quantify this, we characterize the scale of the scene by the length  $r$  of the longest line segment  $pc_p$  between  $\partial\mathcal{V}_{\text{free}}$  and the medial axis (*c.f.* Sec. 3.3). The algorithm described in Sec. 3.3 will generate the longest sequence along this line segment. Hence, an upper bound of the number of constraints is the length of the longest sequence multiplied by  $|\partial\mathcal{V}_{\text{free}}|$ . Fig. 4 shows the sequence length as a function of  $r$ , for the proportional lower bound (8) with different values of  $\alpha$ . We observe that the number of balls increases logarithmically as a function of  $r$ .

Therefore, our method does not scale linearly, since  $r$  may increase as the scene increases in volume. However, if the increase in  $r$  can be bounded by an appropriate factor, our method can achieve linear scaling. For example, when  $\alpha = 0.8$ , the number of balls along the longest line segment is constant within approximately each 10-fold increase in  $r$ . Thus, if the scene growth is bounded such that  $r$  grows by no more than a factor of 10, the number of constraints produced by our method will be linear in the scene’s surface area.

**Table 3.** Number of surface points and (un)corrected free space constraints for the distance bound models described in Sec. 4.1: proportional error (\*) and constant error (\*\*), with (*approx.*) and without (*all*) the approximate cover.

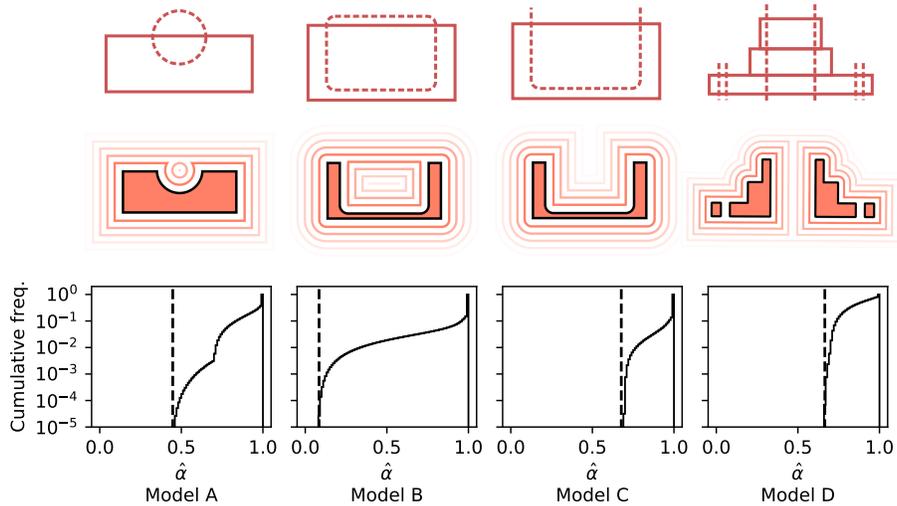
	(a,b)	(c)	(d)	(e,f)
Surface points $ \mathcal{R} $ :	686	15 806	6 827	8 730
Constraints $ \mathcal{I} $ (all), uncorr.	48 386	32 019	42 013	40 193
Constraints $ \mathcal{I} $ (all), corr.*	118 950	93 826	98 768	99 116
Constraints $ \mathcal{I} $ (all), corr.**	83 421	69 036	71 017	71 727
Constraints $ \mathcal{I} $ (approx), uncorr	432	272	320	479
Constraints $ \mathcal{I} $ (approx), corr.*	2 722	1 046	1 776	2 646
Constraints $ \mathcal{I} $ (approx), corr.**	1 240	493	793	1 289



**Fig. 4.** Number of balls produced by the sequence (7) along a single line segment, as a function of the distance to the medial axis  $r = r(c_p)$ , for different  $\alpha$ , using  $t_{\min} = 1$ .

**Analysis of the Lower Bound for Some Implicit Models.** The number of constraints added by our method depends on the lower bound  $g$ . Here, we analyze  $g$  for the implicit models shown in Fig. 5. These models are defined using min/max set operations on primitives with closed-form expressions for the Euclidean distance function. Therefore, a suitable form of the lower bound  $g$  is the form (8), parameterized by the constant  $\alpha$ . Fig. 5 (bottom) shows, for each model, the value of  $\alpha$  and the cumulative histogram of local point-wise estimates  $\hat{\alpha}(p)$  in a box region twice the size of the model. The histograms show that  $\alpha$  is approximately 0.45, 0.09, 0.71 and 0.71, for the respective models.

For example, in Model A,  $\hat{\alpha}(p) = 1/\sqrt{2}$  on the corner diagonals and  $\hat{\alpha}(p) \approx 0.45$  inside the subtracted disk. The value of  $\alpha$  is the minimum of these. The cumulative histogram for Model A shows that  $\hat{\alpha}$  was less than  $1/\sqrt{2}$  for only 0.5% of the points, indicating that the subset of points affected by the error between  $f_x$  and  $d_{\mathcal{M}_x}$ , caused by the disk subtraction, is very small. This is a shortcoming of our formulation of the lower bound, as a local large error affects the lower bound globally. This is also seen in Model B and Model C. These models define identical solids, but Model B subtracts a closed primitive while Model C subtracts an open primitive. In Model B, the subtracted primitive’s proximity



**Fig. 5.** Analysis of the constant  $\alpha$  in the lower bound (8) for four models A-D. **Top:** Construction diagram of model. A dashed outline indicates that the shape is subtracted. **Middle:** Cross section of the model solid and isolines of  $f_x$ . **Bottom:** Cumulative histogram of point-wise estimates  $\hat{\alpha}(p) = f_x(p)/d_{\mathcal{M}_x}(p)$  where  $p$  is taken over a region twice the size of the model and  $f_x(p) > 0$ . The dashed line indicates  $\alpha = \min \hat{\alpha}$ .

near the top of the cavity causes  $f_x$  to be very small relative to  $d_{\mathcal{M}_x}$ , which in turn causes a long tail in the histogram. This is improved by the subtraction in Model C, which yields a smaller error between  $f_x$  and  $d_{\mathcal{M}_x}$ .

## 5 Conclusion

We have presented a method for fitting implicit models subject to free space constraints, requiring only that a bound of the distance to the model surface can be computed at a given point. Our method supports arbitrary shape spaces and provides a minimal set of sampling locations to ensure that the model is consistent with surface- and free space measurements.

Although our method does not retain the linear complexity compared to previous work [2], we do achieve linear complexity if the scene growth is bounded in terms of the largest distance to the medial axis. A relevant direction for future research is to investigate methods for further reducing the computational cost.

**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. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH, ACM (1996) 303–312
2. Haugo, S., Stahl, A.: Iterative closest point with minimal free space constraints. In: International Symposium on Visual Computing, Springer (2020)
3. Hart, J.C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer* **12** (1996) 527–545
4. Pasko, A., Adzhiev, V., Sourin, A., Savchenko, V.: Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer* **11** (1995) 429–446
5. Pasko, A., Fryazinov, O., Vilbrandt, T., Fayolle, P.A., Adzhiev, V.: Procedural function-based modelling of volumetric microstructures. *Graphical Models* **73** (2011) 165–181
6. Barr, A.H.: Global and local deformations of solid primitives. In: SIGGRAPH, ACM (1984) 21–30
7. Savchenko, V., Pasko, A.: Transformation of functionally defined shapes by extended space mappings. *The Visual Computer* **14** (1998) 257–270
8. Pasko, G.I., Pasko, A.A., Kunii, T.L.: Bounded blending for function-based shape modeling. *IEEE Computer Graphics and Applications* **25** (2005) 36–45
9. Sourin, A.I., Pasko, A.A.: Function representation for sweeping by a moving solid. *IEEE Transactions on Visualization and Computer Graphics* **2** (1996) 11–18
10. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. *Computer* **22** (1989) 46–57
11. Dragiev, S., Toussaint, M., Gienger, M.: Gaussian process implicit surfaces for shape estimation and grasping. In: IEEE International Conference on Robotics and Automation. (2011) 2845–2850
12. Seyb, D., Jacobson, A., Nowrouzezahrai, D., Jarosz, W.: Non-linear sphere tracing for rendering deformed signed distance fields. *ACM Transactions on Graphics (TOG)* **38** (2019) 1–12
13. 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
14. 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
15. 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
16. 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
17. 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
18. 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
19. 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

20. 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
21. 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
22. Kalra, D., Barr, A.H.: Guaranteed ray intersections with implicit surfaces. In: SIGGRAPH. (1989) 297–306
23. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. *International Journal of Computer Vision* **38** (2000) 199–218
24. 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
25. 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
26. 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
27. Delaunoy, A., Prados, E.: Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility. *International Journal of Computer Vision* **95** (2011) 100–123
28. 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
29. 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
30. 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
31. Schmidt, T., Newcombe, R.A., Fox, D.: DART: dense articulated real-time tracking. In: *Robotics: Science and Systems*. (2014)
32. Xiao, J., Furukawa, Y.: Reconstructing the world’s museums. *International Journal of Computer Vision* **110** (2014) 243–258
33. 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
34. Tkach, A., Pauly, M., Tagliasacchi, A.: Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics (ToG)* **35** (2016) 1–11
35. Siddiqi, K., Pizer, S.: *Medial representations: mathematics, algorithms and applications*. Volume 37. Springer Science & Business Media (2008)
36. 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
37. Ricci, A.: A constructive geometry for computer graphics. *The Computer Journal* **16** (1973) 157–160