arXiv:2108.00355v1 [cs.CV] 1 Aug 2021

ELLIPSDF: Joint Object Pose and Shape Optimization with a Bi-level Ellipsoid and Signed Distance Function Description

Mo Shan, Qiaojun Feng, You-Yi Jau, Nikolay Atanasov
University of California San Diego
{moshan,qjfeng,yjau,natanasov}@ucsd.edu

Abstract

Autonomous systems need to understand the semantics and geometry of their surroundings in order to comprehend and safely execute object-level task specifications. This paper proposes an expressive yet compact model for joint object pose and shape optimization, and an associated optimization algorithm to infer an object-level map from multi-view RGB-D camera observations. The model is expressive because it captures the identities, positions, orientations, and shapes of objects in the environment. It is compact because it relies on a low-dimensional latent representation of implicit object shape, allowing onboard storage of large multi-category object maps. Different from other works that rely on a single object representation format, our approach has a bi-level object model that captures both the coarse level scale as well as the fine level shape details. Our approach is evaluated on the large-scale real-world ScanNet dataset and compared against state-of-the-art methods.

1. Introduction

Range sensors, such as RGB-D cameras and LIDARs, have become a primary data source for robot localization and mapping due to their increasing accuracy, affordability, and compactness. This has contributed to the development of RGB-D Simultaneous Localization And Mapping (SLAM) [23, 33, 35, 46] and Structure from Motion (SfM) [2, 11, 43] approaches that provide accurate and efficient ego-motion estimation and map reconstruction. The map representations used in RGB-D SLAM, however, are predominantly geometric, composed of point landmarks [23, 45], surfels [47] or explicit (mesh) and implicit (signed distance field) surface representations [33, 39]. These geometric models do not provide semantic information such as the class, pose, shape, or affordances of objects in the scene. Maps that combine geometric and semantic information are useful and understandable for humans and allow specification of symbolic tasks, such as retrieval, object-directed navigation, grasping, and safety critical operation, in terms of object entities.

We gratefully acknowledge support from ARL DCIST CRA W911NF-17-2-0181 and NSF RI IIS-2007141.

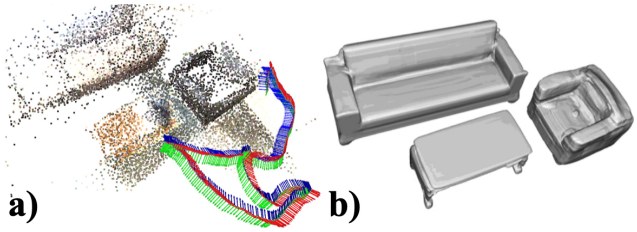

Figure 1. Overview of ELLIPSDF: a) Ground-truth scene reconstruction from colored point clouds in ScanNet [13] scene 0087, where the RGB axes show the camera trajectory, b) Reconstructed object meshes in the world frame using the SDF model decoded from a latent code, and the optimized SIM(3) transformation representing object pose.

Recent works that focus on object-level localization and mapping include [37, 42, 32, 45, 29, 20], which utilize objects as landmarks for localization and navigation [42, 3, 32, 45, 29, 20] or as functional entities for motion, part, and affordance identification [26, 38, 31, 27]. The memory and computational efficiency of the object representations used by semantic SLAM are vital for accommodating online construction, onboard storage, and multi-robot use of large semantic maps. On one hand, a parsimonious way for optimizing and storing object maps is needed to ensure online computation and low onboard memory use. On the other hand, it is desirable to preserve as many details about the object shapes, texture, and affordances as possible. Striking the right balance between a faithful object reconstruction and a compact object representation remains an open research problem.

This paper proposes ELLIPSDF, which is an expressive yet compact model of object pose and shape, and an associated optimization algorithm to infer an object-level map from multi-view RGB-D camera observations, as shown in Fig. 1. ELLIPSDF is expressive because it captures the

identity, scale, position, orientation, and shape of objects in the environment. It is compact because it relies on a low-dimensional latent encoding of the signed distance function (SDF) to an object's surface, allowing onboard storage of large multi-category object maps.

Shape representation using SDF predicted by an autodecoder network was proposed in DeepSDF [36] and DualSDF [18]. In this paper, we extend the SDF prediction network in prior works by proposing a bi-level object model with a shared latent representation. Object primitive shapes and SDF are predicted from a shared latent space. On the coarse-level, an ellipsoid is used as a primitive shape to constrain the overall shape scale. On the fine-level, an autodecoder similar to DeepSDF is used to preserve the object shape details. To summarize, the main **contribution** of this work is the design of

- a bi-level object model with coarse and fine levels, enabling joint optimization of object pose and shape. The coarse-level uses a primitive shape for robust pose and scale initialization, and the fine-level uses SDF residual directly to allow accurate shape modeling. The wo levels are coupled via a shared latent space.
- a cost function to measure the mismatch between the bi-level object model and the segmented RGB-D observations in the world frame.

2. Related Work

Several RGB-D SLAM approaches [33, 14, 23, 15, 47] are able to generate accurate trajectory and a dense 3D model of the environment. However, early RGB-D SLAM techniques focus on obtaining a geometric map and overlook the semantics. Later, object-level SLAM approaches [34, 49] are proposed to model both geometry and semantics. Those works focus on estimating the object pose accurately, but have limited capabilities to model object shape details due to the very simple geometric shape models used, such as cuboids and quadrics.

Compared with other similar works [30, 10] on learning implicit function for surface, DeepSDF [36] learns a continuous metric function of distance instead of binary classification function dividing inside or outside, which makes it suitable for gradient-based optimization in SLAM. Subsequent works along the direction of DeepSDF include FroDO [41], MOLTR [25], and DualSDF [18]. FroDO leverages both point cloud and SDF representations, which defines sparse and dense losses to optimize the object shape. An extension of FroDO is MOLTR, which reconstructs an object shape by fusing multiple single-view shape codes to handle both static and dynamic objects. Similar to the coarse-to-fine shape estimation in FroDO and MOLTR, DualSDF uses two levels of granularity to represent 3D shapes. A shared latent space is employed to tightly couple the two levels, and a Gaussian prior is imposed on the latent space to enable sampling, interpolation, and optimization-based manipulation. DeepSDF and the derivatives offer models for accurate shape modeling but few of them consider object pose estimation.

Object pose estimation is a critical step in the construction of an object level map. To estimate the transformation between world frame and the object frame, Scan2CAD [4] estimates the object pose and scale by establishing keypoint correspondences between the objects in the scene and their 3D CAD models. The keypoints are annotated for the CAD models and predicted by CNNs during inference. The Harris keypoints are detected from the 3D scan to be matched with those keypoints on the CAD models. However, both keypoint annotation and model retrieval take a long time for objects with complicated shapes, such as sofa. Later on Avetisyan *et al.*[5] dramatically increased the efficiency of the alignment process by utilizing a novel differentiable Procrustes alignment loss. Firstly, a proposed 3D CNN is used to identify objects in the 3D scan. Secondly, object bounding boxes are used to establish correspondence between scan objects and the CAD models. Lastly, alignment-informed correspondences are learnt via the differentiable Procrustes alignment loss. Furthermore, multi-view constraints are introduced in Vid2CAD [28].

In the proposed ELLIPSDF, a learnt continuous SDF is used to reconstruct the object at arbitrary resolutions, and thus our approach has a more expressive object model in comparison to [20, 44]. Furthermore, our model has two levels of granularity that provide a coarse object prior to optimize the object scale, which is different from FroDO or [1]. Our system is online and more efficient, and unlike prior works that focus on single object estimation, we also present a large-scale, quantitative evaluation using a public benchmark that has multiple objects.

3. Background

Rigid body orientation, pose, and similarity are represented using the $\mathrm{SO}(3)$, $\mathrm{SE}(3)$, and $\mathrm{SIM}(3)$ Lie groups, respectively, defined as:

$$
\begin{aligned}
\mathrm{SO}(3) &\triangleq \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1 \right\}, \\
\mathrm{SE}(3) &\triangleq \left\{ \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \,\middle|\, \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}, \\
\mathrm{SIM}(3) &\triangleq \left\{ \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \,\middle|\, \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3, s \in \mathbb{R} \right\}.
\end{aligned} \tag{1}
$$

We overload $\boldsymbol{\xi}_\times$ to denote a mapping from a vector in $\mathbb{R}^3$ or $\mathbb{R}^6$ or $\mathbb{R}^7$ to the Lie algebra $\mathfrak{so}(3)$, $\mathfrak{se}(3)$, or $\mathfrak{sim}(3)$, associ-

ated with the Lie groups in (1), defined as:

$$\mathfrak{so}(3) \triangleq \left\{ \boldsymbol{\xi}_\times = \begin{bmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{bmatrix} \;\middle|\; \boldsymbol{\xi} \in \mathbb{R}^3 \right\},$$
$$\mathfrak{se}(3) \triangleq \left\{ \boldsymbol{\xi}_\times = \begin{bmatrix} \boldsymbol{\theta}_\times & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \;\middle|\; \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\theta} \end{bmatrix} \in \mathbb{R}^6 \right\}, \quad (2)$$
$$\mathfrak{sim}(3) \triangleq \left\{ \boldsymbol{\xi}_\times = \begin{bmatrix} \sigma\mathbf{I} + \boldsymbol{\theta}_\times & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \;\middle|\; \boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\theta} \\ \sigma \end{bmatrix} \in \mathbb{R}^7 \right\}.$$

We define an infinitesimal change of a Lie group element $\mathbf{T}$ via a left perturbation $\exp\left(\boldsymbol{\xi}_\times\right)\mathbf{T}$, using the exponential map $\exp\left(\boldsymbol{\xi}_\times\right)$ to retract a Lie algebra element $\boldsymbol{\xi}_\times$ to the Lie group. Please refer to [6, Ch.7] or [16] for details.

The coarse shape of a rigid body is represented using a *quadric shape* [19, Ch.3], $\left\{\mathbf{x} \in \mathbb{R}^3 \mid \underline{\mathbf{x}}^\top \mathbf{Q} \underline{\mathbf{x}} \leq 0\right\}$, where $\underline{\mathbf{x}} \triangleq [\mathbf{x}^\top, 1]^\top$ denotes the homogeneous coordinates of $\mathbf{x}$ and $\mathbf{Q} \in \mathbb{R}^{4\times4}$ is a symmetric matrix. An axis-aligned ellipsoid centered at the origin:

$$\mathcal{E}_\mathbf{u} \triangleq \left\{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x}^\top \mathbf{U}^{-\top} \mathbf{U}^{-1} \mathbf{x} \leq 1\right\}, \quad (3)$$

where $\mathbf{U} \triangleq \operatorname{diag}(\mathbf{u})$ and the elements of the vector $\mathbf{u} \in \mathbb{R}^3$ specify the lengths of the semi-axes of $\mathcal{E}_\mathbf{u}$. An ellipsoid $\mathcal{E}_\mathbf{u}$ is a special case of a quadric shape with $\mathbf{Q} = \operatorname{diag}(\mathbf{U}^{-2}, -1)$. A quadric shape can also be defined in dual form, as the set of planes $\underline{\boldsymbol{\pi}} = \mathbf{Q}\underline{\mathbf{x}}$ that are tangent to the shape surface at each $\mathbf{x}$. This dual quadric surface definition is $\left\{\boldsymbol{\pi} \in \mathbb{R}^3 \mid \underline{\boldsymbol{\pi}}^\top \mathbf{Q}^* \underline{\boldsymbol{\pi}} = 0\right\}$, where $\mathbf{Q}^* = \operatorname{adj}(\mathbf{Q})$ is the adjugate of $\mathbf{Q}$. A dual quadric defined by $\mathbf{Q}^*$ can be scaled, rotated, or translated by a similarity transform $\mathbf{T} \in \mathrm{SIM}(3)$ as $\mathbf{T}\mathbf{Q}^*\mathbf{T}^\top$. Similarity, a dual quadric can be projected to a lower-dimensional space by a projection matrix $\mathbf{P} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$ as $\mathbf{P}\mathbf{Q}^*\mathbf{P}^\top$.

The fine shape of a rigid body is represented as $\left\{\mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}) \leq 0\right\}$ using the *signed distance field* of a set $\mathcal{S} \subset \mathbb{R}^3$:

$$f(\mathbf{x}) = \begin{cases} -d(\mathbf{x}, \partial\mathcal{S}), & \mathbf{x} \in \mathcal{S}, \\ d(\mathbf{x}, \partial\mathcal{S}), & \mathbf{x} \notin \mathcal{S}, \end{cases} \quad (4)$$

where $d(\mathbf{x}, \partial\mathcal{S})$ denotes the Euclidean distance from a point $\mathbf{x} \in \mathbb{R}^3$ to the boundary $\partial\mathcal{S}$ of $\mathcal{S}$.

4. Problem Formulation

Consider an RGB-D camera whose optical frame has pose $\mathbf{C}_k \in \mathrm{SE}(3)$ with respect to the global frame at discrete time steps $k = 1, \ldots, K$. Assume that the camera is calibrated and its pose trajectory $\{\mathbf{C}_k\}_k$ is known, e.g., from a SLAM or SfM algorithm. At time k, the camera provides an RGB image $I_k : \Omega^2 \mapsto \mathbb{R}^3_{\geq 0}$ and a depth image $D_k : \Omega^2 \mapsto \mathbb{R}_{\geq 0}$ such that $I_k(\mathbf{p})$ and $D_k(\mathbf{p})$ are the color and depth of a pixel $\mathbf{p} \in \Omega^2$ in normalized pixel coordinates. The camera moves in an unknown environment that contains N objects $\mathcal{O} \triangleq \{\mathbf{o}_n\}_{n=1}^N$. Each object $\mathbf{o}_n = (\mathbf{c}_n, \mathbf{i}_n)$ is an instance $\mathbf{i}_n$ of class $\mathbf{c}_n$, defined below.

Definition. An *object class* is a tuple $\mathbf{c} \triangleq (\nu, \mathbf{z}, f_{\boldsymbol{\theta}}, g_{\boldsymbol{\phi}})$, where $\nu \in \mathbb{N}$ is the class identity, e.g., chair, table, sofa, and $\mathbf{z} \in \mathbb{R}^d$ is a latent code vector, encoding the average class shape. The class shape is represented in a canonical coordinate frame at two levels of granularity: coarse and fine. The coarse shape is specified by an ellipsoid $\mathcal{E}_\mathbf{u}$ in (3) with semi-axis lengths $\mathbf{u} = g_{\boldsymbol{\phi}}(\mathbf{z})$ decoded from the latent code $\mathbf{z}$ via a function $g_{\boldsymbol{\phi}} : \mathbb{R}^d \mapsto \mathbb{R}^3$ with parameters $\boldsymbol{\phi}$. The fine shape is specified by the signed distance $f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ from any $\mathbf{x} \in \mathbb{R}^3$ to the average shape surface, decoded from the latent code $\mathbf{z}$ via a function $f_{\boldsymbol{\theta}} : \mathbb{R}^3 \times \mathbb{R}^d \mapsto \mathbb{R}$ with parameters $\boldsymbol{\theta}$.

Definition. An *object instance* of class $\mathbf{c}$ is a tuple $\mathbf{i} \triangleq (\mathbf{T}, \delta\mathbf{z})$, where $\mathbf{T} \in \mathrm{SIM}(3)$ specifies the transformation from the global frame to the object instance frame, and $\delta\mathbf{z} \in \mathbb{R}^d$ is a deformation of the latent code $\mathbf{z}$, specifying the average shape of class $\mathbf{c}$.

We assume that object detection (e.g., [8]) and tracking (e.g., [7]) algorithms are available to provide the class $\mathbf{c}_n$ and pixel-wise segmentation $\Omega^2_{n,k} \subseteq \Omega^2$ of any object n observed by the camera at time k. Our goal is to estimate the transformation and shape $\mathbf{i}_n := (\mathbf{T}_n, \delta\mathbf{z}_n)$ of each observed object n. We consider object instances independently and drop the subscript n when it is clear from the context.

Given an object with multi-view segmentation Ω^2_k, we use the depth $D_k(\mathbf{p})$ of each pixel $\mathbf{p} \in \Omega^2_k$ to obtain a set of points $\mathcal{X}_k(\mathbf{p})$ along the ray starting from the camera optical center and passing through $\mathbf{p}$. The sets $\mathcal{X}_k(\mathbf{p})$ is used to optimize the pose and shape of the object instance. For each ray, we choose three points, one lying on the observed surface, one a small distance $\epsilon > 0$ in front of the surface, and one a small distance ϵ behind. Given $d \in \{0, \pm\epsilon\}$, we obtain points $\mathbf{y} \in \mathbb{R}^3$ in the optical frame corresponding to the pixels $\mathbf{p} \in \Omega^2_k$:

$$\mathcal{Y}_k(\mathbf{p}) \triangleq \left\{ (\mathbf{y}, d) \;\middle|\; \mathbf{y} = \left(D_k(\mathbf{p}) + \frac{d}{\|\underline{\mathbf{p}}\|} \right) \underline{\mathbf{p}},\ d \in \{0, \pm\epsilon\} \right\},$$

and project them to the global frame using the known camera pose $\mathbf{C}_k$:

$$\mathcal{X}_k(\mathbf{p}) \triangleq \left\{ (\mathbf{x}, d) \;\middle|\; \mathbf{x} = \mathbf{P}\mathbf{C}_k\underline{\mathbf{y}},\ (\mathbf{y}, d) \in \mathcal{Y}_k(\mathbf{p}) \right\}. \quad (5)$$

We define an error function $e_{\boldsymbol{\phi}}$ to measure the discrepancy between a distance-labelled point $(\mathbf{x}, d) \in \mathcal{X}_k(\mathbf{p})$ observed close to the instance surface and the coarse shape $\mathcal{E}_\mathbf{u}$ provided by $\mathbf{u} = g_{\boldsymbol{\phi}}(\mathbf{z})$. Another error function $e_{\boldsymbol{\theta}}$ is used for the difference between $(\mathbf{x}, d)$ and the SDF value

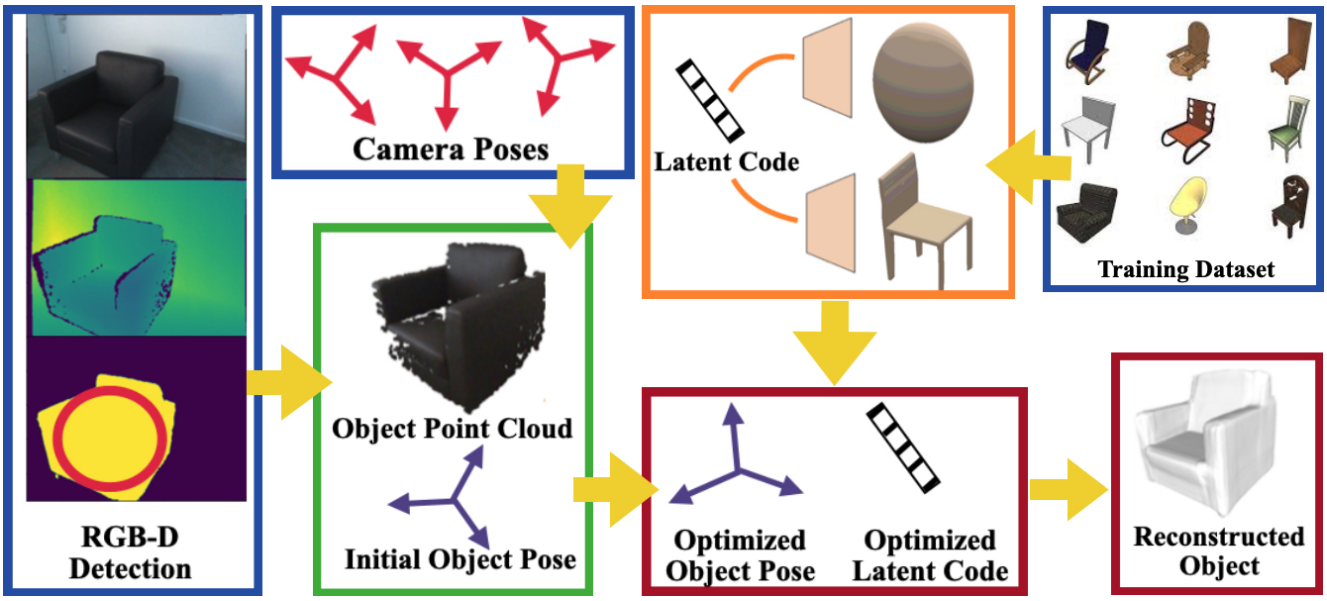

Figure 2. ELLIPSDF Overview: A point cloud and initial pose (*green*) are obtained from RGB-D detections of a chair instance from known camera poses (*blue*). A bi-level category shape description, consisting of a latent shape code, a coarse shape decoder, and a fine shape decoder (*orange*), is trained offline using a dataset of mesh models. Given the observed point cloud, the pose and shape deformation of the newly seen instance are optimized jointly online, achieving shape reconstruction in the global frame (*red*).

$f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ predicted by the fine shape model. The overall error function is defined as:

$$e(\mathbf{T}, \delta\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\phi}; \{\mathcal{X}_k(\mathbf{p})\}) \triangleq \alpha e_r(\delta\mathbf{z}) \tag{6}$$
$$+ \sum_{k=1}^{K} \sum_{\mathbf{p} \in \Omega_k^2} \sum_{(\mathbf{x},d) \in \mathcal{X}_k(\mathbf{p})} \beta e_{\boldsymbol{\theta}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z}) + \gamma e_{\boldsymbol{\phi}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z}),$$

where $e_r(\delta\mathbf{z})$ is a shape deformation regularization term. The coarse-shape error, $e_{\boldsymbol{\theta}}$, fine-shape error, $e_{\boldsymbol{\phi}}$, and the regularization, e_r are defined precisely in Sec. 5.1.

We distinguish between a training phase, where we optimize the parameters $\mathbf{z}$, $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ of an object class using offline data from instances with known mesh shapes, and a testing phase, where we optimize the pose $\mathbf{T}$ and shape deformation $\delta\mathbf{z}$ of a previously unseen instance from the same category using online distance data from an RGB-D camera.

In training, we generate points $\{\mathcal{X}_{n,k}(\mathbf{p})\}$ close to the surface of each available mesh model n in a canonical coordinate frame (with identity pose $\mathbf{I}_4$) and optimize the class shape parameters via:

$$\min_{\{\delta\mathbf{z}_n\}, \boldsymbol{\theta}, \boldsymbol{\phi}} \sum_n e(\mathbf{I}_4, \delta\mathbf{z}_n, \boldsymbol{\theta}, \boldsymbol{\phi}; \{\mathcal{X}_{n,k}(\mathbf{p})\}). \tag{7}$$

In testing, we receive points $\{\mathcal{X}_k(\mathbf{p})\}$ in the global frame, generated by the RGB-D camera from the surface of a previously unseen instance. Assuming known object class, we fix the trained shape parameters $\mathbf{z}^*$, $\boldsymbol{\theta}^*$, $\boldsymbol{\phi}^*$ and optimize the unknown instance transform $\mathbf{T} \in \mathrm{SIM}(3)$ and shape deformation $\delta\mathbf{z} \in \mathbb{R}^d$:

$$\min_{\mathbf{T}, \delta\mathbf{z}} e(\mathbf{T}, \delta\mathbf{z}, \boldsymbol{\theta}^*, \boldsymbol{\phi}^*; \{\mathcal{X}_k(\mathbf{p})\}). \tag{8}$$

5. Object Pose and Shape Optimization

This section develops ELLIPSDF, an autodecoder model for bi-level object shape representation. Sec. 5.1 presents the model and defines the error functions for its parameter optimization. Sec. 5.2 describes how a trained ELLIPSDF model is used at test time for multi-view joint optimization of object pose and shape. An overview is shown in Fig. 2.

5.1. Training an ELLIPSDF Model

Bi-level Shape Representation: The ELLIPSDF shape model consists of two autodecoders $g_{\boldsymbol{\phi}}(\mathbf{z})$ and $f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$, using a shared latent code $\mathbf{z} \in \mathbb{R}^d$. The first autodecoder provides a *coarse* shape representation with parameters $\boldsymbol{\phi}$, as an axis-aligned ellipsoid $\mathcal{E}_{\mathbf{u}}$ in a canonical coordinate frame with semi-axis lengths $\mathbf{u} = g_{\boldsymbol{\phi}}(\mathbf{z})$. The second autoencoder provides a *fine* shape representation with parameters $\boldsymbol{\theta}$, as an implicit SDF surface $\{\mathbf{x} \in \mathbb{R}^3 \mid f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \leq 0\}$ in the same canonical coordinate frame. We implement $g_{\boldsymbol{\phi}}(\mathbf{z})$ and $f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ as 8-layer perceptrons with one cross-connection, as described in Sec. D in the supplementary material of DualSDF [18]. The reparametrization trick [24] is used to maintain a Gaussian distribution $\mathbf{z} = \boldsymbol{\mu} + \mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{\epsilon}$ over the latent code with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Thus, at training time, the ELLIPSDF model parameters are the mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and

standard deviation $\boldsymbol{\sigma} \in \mathbb{R}^d$ of the latent shape code and the coarse and fine shape autodecoder parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. The model is visualized in Fig. 3.

Error Functions: We introduce error terms that play a key role for optimizing the category-level latent code $\mathbf{z}$ and decoder parameters $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, during training time, as well as the transformation $\mathbf{T}$ from the global frame to the canonical object frame and the latent code deformation $\delta\mathbf{z}$ of a particular instance during test time. The training data for an ELLIPSDF model consists of distance-labeled point clouds $\mathcal{X}_{n,k}(\mathbf{p})$ associated with instances n from the same class, as introduced in Sec. 4. A different latent code $\mathbf{z}_n$ is optimized for each instance n, while the decoder parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are common for all instances of the same class.

The fine-level shape error function $e_{\boldsymbol{\theta}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z})$ of a point $\mathbf{x}$ in global coordinates with signed distance label d is defined as:

$$e_{\boldsymbol{\theta}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z}) \triangleq \rho(s f_{\boldsymbol{\theta}}(\mathbf{P}\mathbf{T}\underline{\mathbf{x}}; \mathbf{z} + \delta\mathbf{z}) - d). \quad (9)$$

In the definition above, the point $\mathbf{x}$ is first transformed to the object coordinate frame via $\mathbf{P}\mathbf{T}\underline{\mathbf{x}}$ and the fine-shape model $f_{\boldsymbol{\theta}}$ is queried with the instance shape code $\mathbf{z} + \delta\mathbf{z}$ to predict the SDF to the object surface. Since SDF values vary proportionally with scaling [1], the returned value is scaled back by s before measuring its discrepancy with the label d. Instead of measuring the difference between $s f_{\boldsymbol{\theta}}$ and d in absolute value, we employ a Huber term [21] to make the error function robust against outliers:

$$\rho(r) \triangleq \begin{cases} \frac{1}{2}r^2 & |r| \leq \delta, \\ \delta(|r| - \frac{1}{2}\delta) & \text{else.} \end{cases} \quad (10)$$

Note that the error $e_{\boldsymbol{\theta}}$ relates both the object pose and shape to the SDF residual, which is unique to our formulation and enables their joint optimization.

The coarse-level shape error function $e_{\boldsymbol{\phi}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z})$ is defined similarly, using a signed distance function for the coarse shape. Since the coarse shape decoder, $\mathbf{u} = g_{\boldsymbol{\phi}}(\mathbf{z})$, provides an explicit ellipsoid description, we first need a conversion to SDF before we can define the error term. An approximation of the SDF of an ellipsoid $\mathcal{E}_{\mathbf{u}}$ with semi-axis lengths $\mathbf{u}$ can be obtained as:

$$h(\mathbf{x}, \mathbf{u}) = \frac{\left\|\mathbf{U}^{-1}\mathbf{x}\right\|_2 \left(\left\|\mathbf{U}^{-1}\mathbf{x}\right\|_2 - 1\right)}{\left\|\mathbf{U}^{-2}\mathbf{x}\right\|_2}. \quad (11)$$

Then, the coarse-level shape error of a point $\mathbf{x}$ in global coordinates with signed distance label d is defined as:

$$e_{\boldsymbol{\phi}}(\mathbf{x}, d, \mathbf{T}, \delta\mathbf{z}) \triangleq \rho(sh(\mathbf{P}\mathbf{T}\underline{\mathbf{x}}, g_{\boldsymbol{\phi}}(\mathbf{z} + \delta\mathbf{z})) - d). \quad (12)$$

During training, the object transformation is fixed to be the canonical coordinate frame $\mathbf{T} = \mathbf{I}_4$ because the training point-cloud data is collected directly in the object frame. The regularization term $e_r(\delta\mathbf{z})$ in (6) is defined as the KL divergence between the distribution of $\delta\mathbf{z}$ and a standard normal distribution [18].

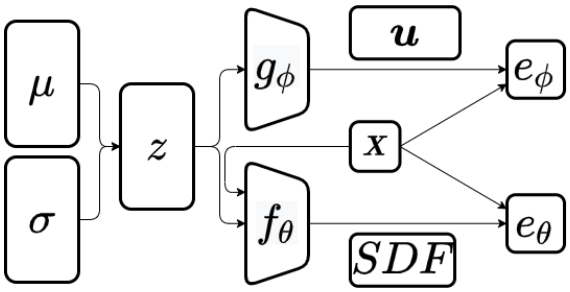

Figure 3. Overview of our ELLIPSDF bi-level object shape model. A latent shape code, $\mathbf{z}$, with distribution $\mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma})^2)$ is shared by a coarse shape decoder g_ϕ, providing an ellipsoid shape description, and a fine shape decoder f_θ, providing an SDF shape description. During training, the decoder parameters ϕ and θ are optimized by minimizing the errors between the SDF values of the training points $\mathbf{x}$, obtained close to the object surface, and the coarse and fine shape models.

5.2. Joint Pose and Shape Optimization with an ELLIPSDF Model

This section describes how a trained ELLIPSDF model is used to initialize and optimize the pose and shape of a new object instance at test time.

Initialization: We follow [12, 40, 17] to initialize the SIM(3) scale and pose of an observed object, relying on its coarse ellipsoid shape representation. We fit ellipses to the pixel-wise segmentation Ω_k^2 of an object at each time k:

$$\left\{\mathbf{q} \in \Omega^2 \mid (\mathbf{q} - \mathbf{c}_k)^\top \mathbf{E}_k^{-1} (\mathbf{q} - \mathbf{c}_k) \leq 1\right\}, \quad (13)$$

where the center and symmetric matrix are obtained as $\mathbf{c}_k = \frac{1}{|\Omega_k^2|}\sum_{\mathbf{p}\in\Omega_k^2}\mathbf{p}$ and $\mathbf{E}_k = \frac{2}{|\Omega_k^2|}\sum_{\mathbf{p}\in\Omega_k^2}(\mathbf{p} - \mathbf{c}_k)(\mathbf{p} - \mathbf{c}_k)^\top$. The axes lengths are the eigenvalues λ_0, λ_1 of $\mathbf{E}_k$. The 2D quadric surface corresponding to the ellipse in (13) and its dual are defined by the matrix $\mathbf{H}_k$ and its inverse $\mathbf{H}_k^*$:

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{E}_k^{-1} & -\mathbf{E}_k^{-1}\mathbf{c}_k \\ -\mathbf{c}_k^\top \mathbf{E}_k^{-1} & \mathbf{c}_k^\top \mathbf{E}_k^{-1}\mathbf{c}_k - 1 \end{bmatrix}, \quad \mathbf{H}_k^* = \begin{bmatrix} \mathbf{E}_k - \mathbf{c}_k\mathbf{c}_k^\top & -\mathbf{c}_k \\ -\mathbf{c}_k^\top & -1 \end{bmatrix}.$$

An ellipsoid in dual quadric form $\mathbf{Q}^*$ in global coordinates and its conic projection $\mathbf{H}_k^*$ in image k are related by $\beta_k \mathbf{H}_k^* = \mathbf{P}\mathbf{C}_k^{-1}\mathbf{Q}^*\mathbf{C}_k^{-\top}\mathbf{P}^\top$ defined up to a scale factor β_k. This equation can be rearranged to $\beta_k \mathbf{h}_k = \mathbf{G}_k \mathbf{v}$, where $\mathbf{h}_k = \mathrm{vech}(\mathbf{H}_k^*)$, $\mathbf{h}_k \in \mathbb{R}^6$, $\mathbf{v} = \mathrm{vech}(\mathbf{Q}^*)$ and $\mathbf{v} \in \mathbb{R}^{10}$. The operator vech serializes the lower triangular part of a symmetric matrix, and $\mathbf{G}_k$ is a matrix that depends on $\mathbf{P}\mathbf{C}_k^{-1}$. The explicit form of $\mathbf{G}_k$ is derived in (5) in [40]. Next, a least squares system is constructed from the multi-view observations. By stacking all observations, we obtain

$\mathbf{Mw} = \mathbf{0}$, where $\mathbf{w} = (\mathbf{v}, \beta_1, \ldots, \beta_k)^\top$, and $\mathbf{M}$ is defined in (8) in [40]. This leads to a least squares system:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{Mw}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{w}\|_2^2 = 1, \tag{14}$$

which can be solved by applying SVD to $\mathbf{M}$, and taking the right singular vector associated to the minimum singular value. The constraint $\|\mathbf{w}\|_2^2 = 1$ avoids a trivial solution. The first 10 entries of $\hat{\mathbf{w}}$ are $\hat{\mathbf{v}}$, which is a vectorized version of the dual ellipsoid $\hat{\mathbf{Q}}^*$ in the global frame. To avoid degenerate quadrics, a variant of the least squares system in (14) is proposed in [17], which constrains the center of the ellipse and the reprojection of the center of the 3D ellipsoid to be close. Thus, we modify $\mathbf{M}$ using the version in (9) in [17] to improve the estimation.

The object pose $\hat{\mathbf{T}}^{-1}$ can be recovered by relating the estimated ellipsoid $\hat{\mathbf{Q}}^*$ in global coordinates to the ellipsoid $\mathbf{Q}_\mathbf{u}^*$ in the canonical coordinate frame predicted by the coarse shape decoder $\mathbf{u} = g_{\boldsymbol{\phi}}(\mathbf{z})$ using the average class shape $\mathbf{z}$:

$$\hat{\mathbf{Q}}^* = \hat{\mathbf{T}}^{-1}\mathbf{Q}_\mathbf{u}^*\hat{\mathbf{T}}^{-\top} = \begin{bmatrix} \hat{s}^2\hat{\mathbf{R}}\mathbf{U}\mathbf{U}^\top\hat{\mathbf{R}}^\top - \hat{\mathbf{t}}\hat{\mathbf{t}}^\top & -\hat{\mathbf{t}} \\ -\hat{\mathbf{t}}^\top & -1 \end{bmatrix}.$$

The translation $\hat{\mathbf{t}}$ can be recovered from the last column of $\hat{\mathbf{Q}}^*$. To recover the rotation, note that $\mathbf{A} \triangleq \mathbf{P}\hat{\mathbf{Q}}^*\mathbf{P}^\top + \hat{\mathbf{t}}\hat{\mathbf{t}}^\top = \hat{s}^2\hat{\mathbf{R}}\mathbf{U}\mathbf{U}^\top\hat{\mathbf{R}}^\top$ is a positive semidefinite matrix. Let its eigen-decomposition be $\mathbf{A} = \mathbf{V}\mathbf{Y}\mathbf{V}^\top$, where $\mathbf{Y}$ is a diagonal matrix containing the eigenvalues of $\mathbf{A}$. Since $\mathbf{U}\mathbf{U}^\top$ is diagonal, it follows that $\hat{\mathbf{R}} = \mathbf{V}$, while the scale $\hat{s}$ is obtained as $\hat{s} = \frac{1}{3}\sqrt{\mathrm{tr}(\mathbf{U}^{-1}\mathbf{Y}\mathbf{U}^{-\top})}$. Note that although the SIM(3) pose could also be recovered from the object point cloud, other outlier rejection methods are required [48] when the point cloud is noisy.

Optimization: Given the initialized instance transformation $\hat{\mathbf{T}}$ and initial shape deformation $\delta\hat{\mathbf{z}} = \mathbf{0}$, we solve the joint pose and shape optimization in (8) via gradient descent. Note that the decoder parameters $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ and the mean category shape code $\mathbf{z}$ are fixed during online inference. Since $\mathbf{T}$ is an element of the SIM(3) manifold, the gradients and gradient steps need to be computed by projecting to the tangent sim(3) vector space and retracting back to SIM(3). We introduce local perturbations $\mathbf{T} = \exp\left(\boldsymbol{\xi}_\times\right)\hat{\mathbf{T}}$, $\delta\mathbf{z} = \delta\tilde{\mathbf{z}} + \delta\hat{\mathbf{z}}$ and derive the Jacobians of the error function in (6) with respect to $\boldsymbol{\xi}$ and $\delta\tilde{\mathbf{z}}$.

Proposition 1. The Jacobian of $e_{\boldsymbol{\theta}}$ in (9) with respect to the transformation perturbation $\boldsymbol{\xi} \in \mathfrak{sim}(3)$ is:

$$\frac{\partial e_{\boldsymbol{\theta}}}{\partial \boldsymbol{\xi}} = \frac{\partial \rho(r)}{\partial r}\left(\hat{s}[\mathbf{0}_6, 1] f_{\boldsymbol{\theta}}(\mathbf{x}, \delta\hat{\mathbf{z}}) + \hat{s}\nabla_\mathbf{x} f_{\boldsymbol{\theta}}(\mathbf{x}, \delta\hat{\mathbf{z}})^\top \mathbf{P}\left[\hat{\mathbf{T}}\underline{\mathbf{x}}\right]^\odot\right)$$
$$\frac{\partial e_{\boldsymbol{\theta}}}{\partial \delta\tilde{\mathbf{z}}} = \frac{\partial \rho(r)}{\partial r}\hat{s}\nabla_\mathbf{z} f_{\boldsymbol{\theta}}(\mathbf{x}, \delta\hat{\mathbf{z}}),$$

where $f_{\boldsymbol{\theta}}(\mathbf{x}, \delta\hat{\mathbf{z}}) = f_{\boldsymbol{\theta}}(\mathbf{P}\hat{\mathbf{T}}\underline{\mathbf{x}}; \mathbf{z} + \delta\hat{\mathbf{z}})$ is defined in (9) and $\frac{\partial \rho(r)}{\partial r}$ is the derivative of the Huber term in (10) evaluated at $r = \hat{s} f_{\boldsymbol{\theta}}(\mathbf{x}, \delta\hat{\mathbf{z}}) - d$:

$$\frac{\partial \rho(r)}{\partial r} = \begin{cases} r & |r| \le \delta \\ \mathrm{sign}(r)\delta & \text{else.} \end{cases}$$

The operator $\underline{\mathbf{x}}^\odot$ is defined as:

$$\underline{\mathbf{x}}^\odot \triangleq \begin{bmatrix} \mathbf{I}_3 & -\mathbf{x}_\times & \mathbf{x} \\ \mathbf{0}^\top & \mathbf{0}^\top & 0 \end{bmatrix} \in \mathbb{R}^{4\times 7}.$$

Proof. Using the chain rule and the product rule:

$$\frac{\partial e_{\boldsymbol{\theta}}}{\partial \boldsymbol{\xi}} = \frac{\partial e_{\boldsymbol{\theta}}}{\partial r}\frac{\partial r}{\partial \boldsymbol{\xi}} = \frac{\partial e_{\boldsymbol{\theta}}}{\partial r}\left(\frac{\partial s}{\partial \boldsymbol{\xi}} f_{\boldsymbol{\theta}}(\mathbf{x}, \delta\mathbf{z}) + s\frac{\partial f_{\boldsymbol{\theta}}}{\partial {}_O\mathbf{x}}\frac{\partial {}_O\mathbf{x}}{\partial \boldsymbol{\xi}}\right),$$

where ${}_O\mathbf{x} = \mathbf{P}\mathbf{T}\underline{\mathbf{x}}$ is a point in the object frame. We have $\frac{\partial s}{\partial \boldsymbol{\xi}} = e^\sigma[\mathbf{0}_6, 1] = s[\mathbf{0}_6, 1]$. The term $s\frac{\partial f_{\boldsymbol{\theta}}}{\partial {}_O\mathbf{x}}$ is the gradient of the fine-level SDF decoder with respect to the input $s\nabla_\mathbf{x} f_{\boldsymbol{\theta}}(\mathbf{x}, \delta\mathbf{z})$, which could be obtained from auto-differentiation. Finally, we have:

$$\begin{aligned} {}_O\mathbf{x} &= \mathbf{P}\mathbf{T}\underline{\mathbf{x}} \approx \mathbf{P}(\mathbf{I} + \boldsymbol{\xi}_\times)\hat{\mathbf{T}}\underline{\mathbf{x}} \\ &= \mathbf{P}\hat{\mathbf{T}}\underline{\mathbf{x}} + \mathbf{P}\boldsymbol{\xi}_\times\hat{\mathbf{T}}\underline{\mathbf{x}} \\ &= \mathbf{P}\hat{\mathbf{T}}\underline{\mathbf{x}} + \underbrace{\mathbf{P}[\hat{\mathbf{T}}\underline{\mathbf{x}}]^\odot}_{\frac{\partial {}_O\mathbf{x}}{\partial \boldsymbol{\xi}}}\boldsymbol{\xi}. \end{aligned}$$

□

In the second equality in Prop. 1, the term $\frac{\partial \rho(r)}{\partial r}\hat{s}\nabla_\mathbf{z} f_{\boldsymbol{\theta}}(\mathbf{x}, \delta\hat{\mathbf{z}})$ is the gradient of the fine-level SDF loss with respect to the input $\mathbf{z}$ and can be obtained via auto-differentiation. The Jacobians of the coarse-level SDF error $\frac{\partial e_{\boldsymbol{\phi}}}{\partial \boldsymbol{\xi}}$, $\frac{\partial e_{\boldsymbol{\phi}}}{\partial \delta\tilde{\mathbf{z}}}$ can be obtained in a similar way.

After obtaining the Jacobians, the pose and latent shape code can be optimized via:

$$\mathbf{T}^{i+1} \triangleq \exp\left(-\eta_1\frac{\partial e(\mathbf{T}, \delta\mathbf{z}, \boldsymbol{\theta}^*, \boldsymbol{\phi}^*; \{\mathcal{X}_k(\mathbf{p})\})}{\partial \boldsymbol{\xi}}\right)\mathbf{T}^i$$
$$\delta\mathbf{z}^{i+1} \triangleq \delta\mathbf{z}^i - \eta_2\left(\frac{\partial e(\mathbf{T}, \delta\mathbf{z}, \boldsymbol{\theta}^*, \boldsymbol{\phi}^*; \{\mathcal{X}_k(\mathbf{p})\})}{\partial \delta\mathbf{z}}\right),$$

where η_1, η_2 are step sizes, $\delta\mathbf{z}^0 = \mathbf{0}$, and $\mathbf{T}^0 = \hat{\mathbf{T}}$ is obtained from the initialization. During optimization, we add regularization $e_r(\delta\mathbf{z}) = \|\delta\mathbf{z}\|_2^2$ to restrict the amount of latent code deformation.

6. Evaluation

6.1. Training Details

The ELLIPSDF decoder model is trained on synthetic CAD models from ShapeNet [9]. Each model's scale is normalized to be inside a unit sphere. We sample points and calculate their SDF values using a uniform distribution in the unit sphere for training the coarse-level shape decoder $g_{\boldsymbol{\phi}}$. Another set of points that are close to the model surface are sampled for training the fine-level shape decoder $f_{\boldsymbol{\theta}}$.

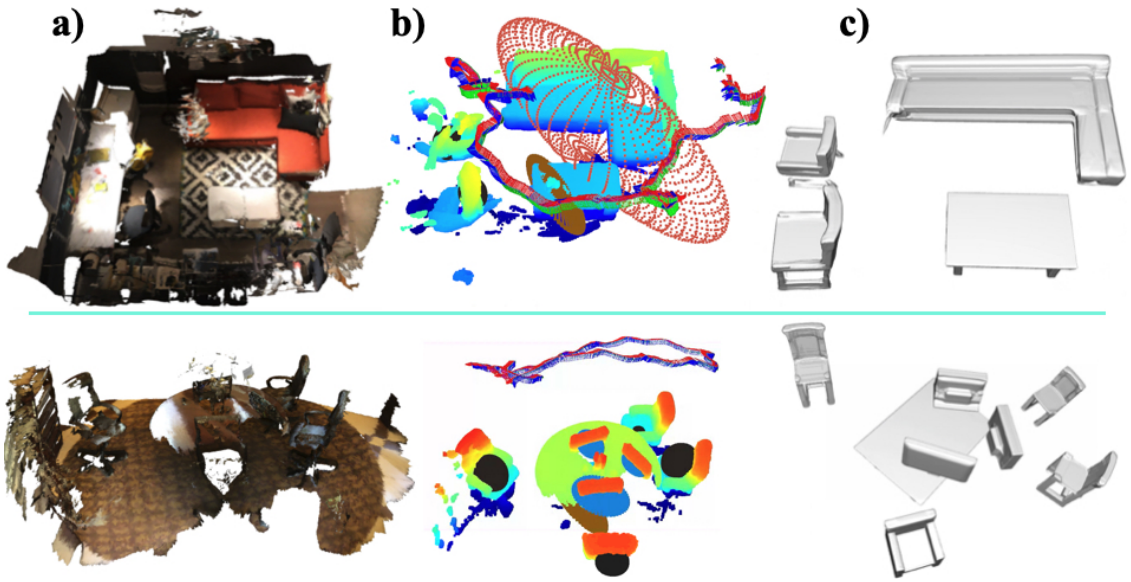

Figure 4. Qualitive results. Column a): Ground-truth scene in ScanNet Sequence 0518 (upper row) and 0314 (lower row). Column b): The RGB axes are the camera trajectory, point clouds are the ones obtained from RGB-D sensor with added pesudo points, and the ellipsoids (black for chair, red for sofa, blue for monitor, brown for table) are the initialized objects. Column c): Reconstructed meshes using ELLIPSDF, rendered from the optimized latent code and pose.

The following setting were used to train the decoder networks and the latent shape code $\mathbf{z}$. We use the Adam optimizer with initial learning rate 5×10^{-4}, 0.5 ratio decay every 300/700 epochs for the coarse and fine level networks separately. The total epoch number is 1500. The latent code dimension is 64, and the network structure follows the model in DualSDF [18].

6.2. Qualitative Results

We evaluate ELLIPSDF on the ScanNet dataset [13], which provides 3D scans captured by a RGB-D sensor of indoor scenes with chairs, tables, displays, etc. We segment out objects from scene-level mesh using provided instance labels, and sample points from object meshs to generate point observations. Visualizations of shape optimization for a chair are shown in Fig. 5. Optimization step improves the scale and shape estimates notably, e.g. by transforming the four-leg mean shape into an armchair. Larger scale qualitative results are shown in Fig. 4, demonstrating the effectiveness of joint shape and pose optimization. Optimized poses are closer to the ground-truth, and optimized shapes resemble the objects better than simple primitive shapes such as cuboids or quadrics that lacks fine details. For example, the successful reconstruction of an angle sofa is illustrated in the upper row in Fig. 4, which deforms from an initial mean sofa shape that does not have an angle. ELLIPSDF is also able to deal with partial observations as seen in the lower row in Fig. 4. Although the observed point clouds of the displays and the chairs are sparse, our approach still reconstructs those objects successfully. Nevertheless, the reconstruction is a square instead of rounded for the table due to a severe occlusion of the observation that only less than half of the table is observed.

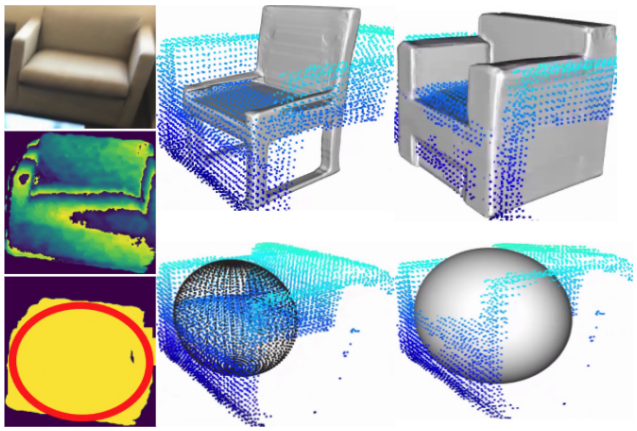

Figure 5. Intermediate ELLIPSDF stages. First column: RGB image, depth image, instance segmentation (yellow), fitted ellipse (red) for a chair in ScanNet scene 0461. Second column: mean shape and ellipsoid with initialized pose. Third column: optimized fine-level and coarse-level shapes with optimized pose.

6.3. Quantitative Results

This section presents quantitative evaluation against other methods regarding both pose and shape estimation accuracy. We also present ablation studies to showcase the improvement of the optimization over initialization-only results, and the bi-level model over a one level model.

Table 1. Quantitative results for pose estimation on ScanNet [13].

Scan2CAD [4]	Vid2CAD [28]	ELLIPSDF (init)	ELLIPSDF (opt)
31.7	38.3	31.5	**39.6**

Table 2. Quantitative results for shape evlaution on ScanNet[13].

Method	cabinet	chair	display	table	avg.
# intances	132	820	209	146	327
ELLIPSDF (fine)	88.4	88.3	90.6	76.2	85.9
ELLIPSDF (coarse+fine)	**91.0**	**90.6**	**96.9**	**77.3**	**89.0**

Evaluation on Object Pose: We obtain the ground-truth object pose annotations from Scan2CAD [4] and follow the pose evaluation metrics it defines, which decomposes a pose $\mathbf{T} \in \mathrm{SIM}(3)$ into rotation $\mathbf{q}$, translation $\mathbf{p}$ and scale $\mathbf{s}$. For an accurate pose estimation, the error thresholds for translation, rotation, and scales are set as 0.2, 20° and 20% respectively with respect to the ground-truth pose. The pose evaluation is presented in Tab. 1, in which ELLIPSDF (init) refers to the initialization-only step in Sec. 5.2, whereas ELLIPSDF (opt) refers using both the initialization and optimization steps in Sec. 5.2. The last two columns in Tab. 1 show that adding optimization step using SDF residuals improves the estimation by the initialization-only variant, due to the additional SDF residuals to help estimate pose. Moreover, ELLIPSDF (opt) outperforms both Scan2CAD and Vid2CAD, which demonstrates the superiority of ELLIPSDF that employs a primitive ellipsoid shape tailored for pose and scale estimation.

Evaluation on Object Shape: We evaluate ELLIPSDF for shape prediction on ScanNet [13] dataset in Tab. 2. Instead of single object evaluation in FroDO [41], we evaluate on multiple objects, which is harder than the single-object-scene due to clustering and partial observations. The large scale evaluation verifies that our method can generalize across different sequences and objects. The object point cloud sampled from the object mesh from [4] is used as the ground truth $\mathcal{S}_{gt}$, and the estimated point cloud $\mathcal{S}_{est}$ is generated from the optimized latent code $\mathbf{z} + \delta\mathbf{z}$. Given the ground-truth point cloud $\mathcal{S}_{gt}$ and ELLIPSDF point cloud $\mathcal{S}_{est}$ for an object, the fitting rate with inlier ratio is

$$
\begin{aligned}
fit(\mathcal{S}_{est}, \mathcal{S}_{gt}) &= \frac{|\mathcal{S}_{close}|}{|\mathcal{S}_{est}|}, \\
\mathcal{S}_{close} &= \{\mathbf{v} \in \mathcal{S}_{est} : d_f(\mathbf{v}, \mathcal{S}_{gt}) < \lambda\},
\end{aligned}
\tag{15}
$$

where $\lambda = 0.2(m)$. A distance function $d_f(\cdot, \cdot)$ is utilized to measure the distance between a point $\mathbf{v}$ and a point cloud $\mathcal{S}$, which is the distance from the closest point $\mathbf{u} \in \mathcal{S}$ to the point $\mathbf{v}$. In CAD-Deform [22], the distance function is set to be L1 distance, while we use L2 distance.

We run ELLIPSDF (fine) and ELLIPSDF (coarse+fine) on 150 validation sequences on ScanNet [13], where ELLIPSDF (fine) means only the fine level SDF residual is used by setting $\gamma = 0$ in (6), and ELLIPSDF (coarse+fine) means the bi-level SDF residuals are used. For each optimized object, we calculate the fitting rate and then average across all instances. In Tab. 2, we show the number of instances and average fitting rates for 4 object classes. ELLIPSDF (coarse+fine) achieves better results than ELLIPSDF (fine) across all classes, demonstrating an average 3% boost of fitting rate with the assistance of coarse model, reaching nearly 90% accuracy. The results indicate the effectiveness of the coarse level error function for improving the scale estimation.

Table 3. Comparison of 3D detection results on ScanNet [13].

mAP @ IoU=0.5	Chair	Table	Display
FroDO [41]	0.32	0.06	0.04
MOLTR [25]	0.39	0.06	0.10
ELLIPSDF (fine)	0.42	0.26	0.25
ELLIPSDF (coarse+fine)	**0.43**	**0.27**	**0.31**

Evaluation on 3D IoU: For a quantitative evaluation on pose estimation, our approach is compared with FroDO [41] and MOLTR [25] on ScanNet [13]. The ground-truth object poses and shapes are from Scan2CAD [4], whereas the estimated 3D bounding box is generated from the estimated point cloud. The evaluation metric is same as [25], i.e. mean Average Precision (mAP), and the IoU threshold is 0.5. The results are shown in Tab. 3. First, we compare the bi-level model against the one-level model. From the last two rows in Tab. 3, ELLIPSDF (coarse+fine) is superior than ELLIPSDF (fine) in terms of 3D IoU, and thus demonstrates that the bi-level model is beneficial by providing additional cues to constrain the pose and shape. The improvement is more significant for smaller objects, e.g. the displays. This may be explained by the fact that the initialization error is relatively larger for smaller objects, and thus requires a coarse shape residual to confine its pose. Moreover, ELLIPSDF outperforms both FroDO and MOLTR by a large margin for two probably reasons. Firstly, 3D point clouds are used in the observation for ELLIPSDF, while the other two only rely on 2D observations. Secondly, ELLIPSDF computes coarse level SDF residuals using a primitive shape to aid the estimation of pose and shape scale, whereas the other methods use SDF residuals computed from fine shape details.

7. Conclusion

This work proposes ELLIPSDF, which a novel semantic mapping approach for RGB-D sensors using a compact, shared latent representation for a bi-level object model to achieve joint pose and shape optimizaiton. Evaluation results on large-scale dataset demonstrate the superiority of ELLIPSDF compared with other approaches. A future research direction is to integrate ELLIPSDF into the pose graph optimization for key-frame based SLAM.

References

[1] Oladapo Afolabi, Allen Y Yang, and S Shankar Sastry. Extending deepsdf for automatic 3d shape retrieval and similarity transform estimation. *arXiv e-prints*, pages arXiv–2004, 2020.

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[3] N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas. Localization from semantic observations via the matrix permanent. *The International Journal of Robotics Research (IJRR)*, 35:73–99, 2015.

[4] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019.

[5] Armen Avetisyan, Angela Dai, and Matthias Nießner. End-to-end cad model retrieval and 9dof alignment in 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2551–2560, 2019.

[6] Timothy D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, Sep 2016.

[8] Z. Cai and N. Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[9] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] David Crandall, Andrew Owens, Noah Snavely, and Dan Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR 2011*, pages 3001–3008. IEEE, 2011.

[12] Marco Crocco, Cosimo Rubino, and Alessio Del Bue. Structure from motion with objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4141–4149, 2016.

[13] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[14] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. In *Icra*, volume 3, pages 1691–1696, 2012.

[15] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an rgb-d camera. *IEEE transactions on robotics*, 30(1):177–187, 2013.

[16] Xiang Gao, Tao Zhang, Yi Liu, and Qinrui Yan. *14 Lectures on Visual SLAM: From Theory to Practice*. Publishing House of Electronics Industry, 2017.

[17] Paul Gay, James Stuart, and Alessio Del Bue. Visual graphs from motion (vgfm): Scene understanding with object geometry reasoning. In *Asian Conference on Computer Vision*, pages 330–346. Springer, 2018.

[18] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. Dualsdf: Semantic shape manipulation using a two-level representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7631–7641, 2020.

[19] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[20] Lan Hu, Wanting Xu, Kun Huang, and Laurent Kneip. Deepslam++: Object-level rgbd slam based on class-specific deep shape priors. *arXiv preprint arXiv:1907.09691*, 2019.

[21] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

[22] Vladislav Ishimtsev, Alexey Bokhovkin, Alexey Artemov, Savva Ignatyev, Matthias Niessner, Denis Zorin, and Evgeny Burnaev. Cad-deform: Deformable fitting of cad models to 3d scans. *arXiv preprint arXiv:2007.11965*, 2020.

[23] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Kejie Li, Hamid Rezatofighi, and Ian Reid. Moltr: Multiple object localisation, tracking and reconstruction from monocular rgb videos. *IEEE Robotics and Automation Letters*, 2021.

[26] Cewu Lu, Hao Su, Yonglu Li, Yongyi Lu, Li Yi, Chi-Keung Tang, and Leonidas J Guibas. Beyond holistic object recognition: Enriching image understanding with part states. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6955–6963, 2018.

[27] Tiange Luo, Kaichun Mo, Zhiao Huang, Siyu Hu, Jiarui Xu, Liwei Wang, and Hao Su. Learning to group: A bottom-up framework for 3d part discovery in unseen categories. In *International Conference on Learning Representations (ICLR)*, 2020.

[28] Kevis-Kokitsi Maninis, Stefan Popov, Matthias Nießner, and Vittorio Ferrari. Vid2cad: Cad model alignment using multi-view constraints from videos. *arXiv preprint arXiv:2012.04641*, 2020.

[29] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level slam. In *2018 International Conference on 3D Vision (3DV)*, pages 32–41. IEEE, 2018.

[30] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[31] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2019.

[32] Beipeng Mu, Shih-Yuan Liu, Liam Paull, John Leonard, and Jonathan P How. Slam with objects using a nonparametric pose graph. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4602–4609. IEEE, 2016.

[33] Richard A Newcombe, Shahram Izadi, and Otmar Hilliges. Kinectfusion: Real-time dense surface mapping and tracking. 2011.

[34] Lachlan Nicholson, Michael Milford, and Niko Sünderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 4(1):1–8, 2018.

[35] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguère, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. *arXiv preprint arXiv:1905.02082*, 2019.

[36] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] Andrzej Pronobis. *Semantic mapping with mobile robots*. PhD thesis, KTH Royal Institute of Technology, 2011.

[38] Kejie Qiu, Tong Qin, Wenliang Gao, and Shaojie Shen. Tracking 3-d motion of dynamic objects using monocular visual-inertial sensing. *IEEE Transactions on Robotics*, 2019.

[39] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696. IEEE, 2020.

[40] Cosimo Rubino, Marco Crocco, and Alessio Del Bue. 3d object localisation from multi-view image detections. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1281–1294, 2017.

[41] Martin Runz, Kejie Li, Meng Tang, Lingni Ma, Chen Kong, Tanner Schmidt, Ian Reid, Lourdes Agapito, Julian Straub, Steven Lovegrove, et al. Frodo: From detections to 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14720–14729, 2020.

[42] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013.

[43] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[44] Edgar Sucar, Kentaro Wada, and Andrew Davison. Nodeslam: Neural object descriptors for multi-view shape reconstruction. In *2020 International Conference on 3D Vision (3DV)*, pages 949–958. IEEE, 2020.

[45] Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid. Meaningful maps with object-oriented semantic mapping. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5079–5085. IEEE, 2017.

[46] Kaixuan Wang, Fei Gao, and Shaojie Shen. Real-time scalable dense surfel mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6919–6925. IEEE, 2019.

[47] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[48] Yanmin Wu, Yunzhou Zhang, Delong Zhu, Yonghui Feng, Sonya Coleman, and Dermot Kerr. Eao-slam: Monocular semi-dense object slam based on ensemble data association. *arXiv preprint arXiv:2004.12730*, 2020.

[49] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019.

Supplementary Material

Trained Object Models

This section provides additional visualizations for the trained object models. Training loss for the chair category is visualize in Fig. 6, which shows the loss is decreasing and stabilizes around 40,000 epochs.

Fig. 7 visualizes the rendering results for some chairs in the training set. It shows that the scale of the primitive-based representation varies proportionally with the high-resolution representation.

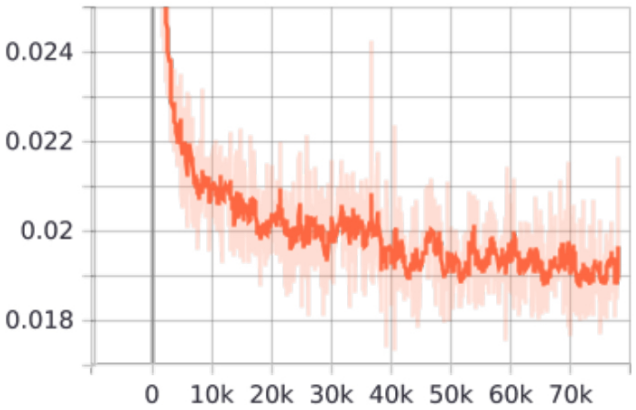

Figure 6. Visualization of the training loss for chairs.

Fig. 8 visualizes the rendering results for sofas in the training set. There is a lack of shape variation since the majority of sofas have similar structure. Nevertheless, the ellispoid for the angle sofa is still different with that of other sofas.

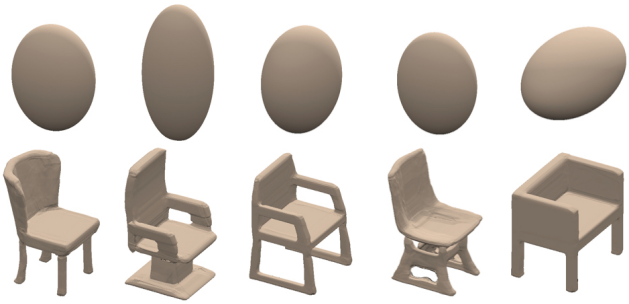

Figure 7. Visualization of the trained object model for chairs. Upper row: coarse ellipsoid shapes regressed from g_{ϕ} and $\mathbf{z}$. Lower row: SDF object model from f_{θ} and $\mathbf{z}$.

Fig. 9 visualizes the rendering results for tables in the training set. Similar to sofas, the variation is limited due to similar table shapes. Nonetheless, the ellipsoid for the rounded table is different from the rest.

Fig. 10 visualizes the rendering results for trashbins in the training set. It could be observed that the ellipsoid shape varies based on the object shape, for instance, the ellipsoid is enlongated for a tall trashbin.

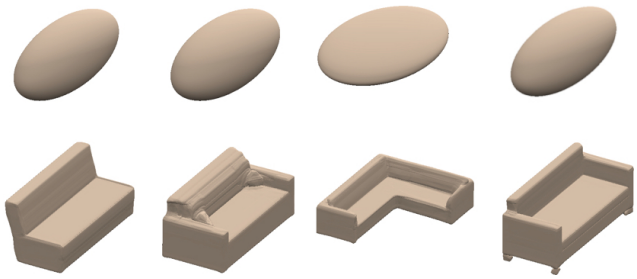

Figure 8. Visualization of the trained object model for sofas. Upper row: coarse ellipsoid shapes regressed from g_{ϕ} and $\mathbf{z}$. Lower row: SDF object model from f_{θ} and $\mathbf{z}$.

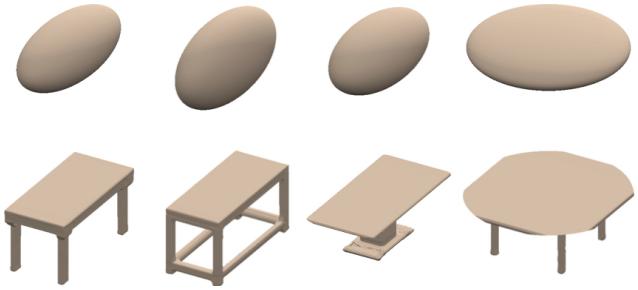

Figure 9. Visualization of the trained object model for tables. Upper row: coarse ellipsoid shapes regressed from g_{ϕ} and $\mathbf{z}$. Lower row: SDF object model from f_{θ} and $\mathbf{z}$.

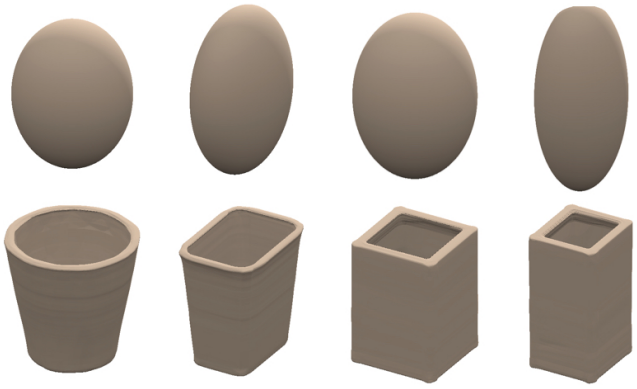

Figure 10. Visualization of the trained object model for trashbins. Upper row: coarse ellipsoid shapes regressed from g_{ϕ} and $\mathbf{z}$. Lower row: SDF object model from f_{θ} and $\mathbf{z}$.

Fig. 11 visualizes the rendering results for displays in the training set. The ellipsoid is rounded for the thicker display and is very thin for the rest.

Fig. 12 visualizes the rendering results for cabinets in the training set. The ellipsoid varies according to the different cabinet shapes.

More Qualitative Results on ScanNet

This section presents more qualitative results on ScanNet [13]. Fig. 13 shows a reconstruction with table, trash-

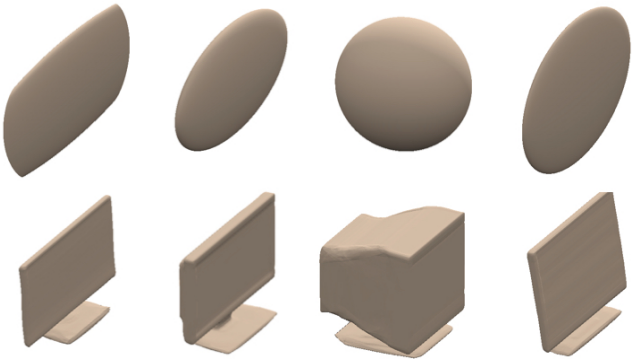

Figure 11. Visualization of the trained object model for displays. Upper row: coarse ellipsoid shapes regressed from g_{ϕ} and $\mathbf{z}$. Lower row: SDF object model from f_{θ} and $\mathbf{z}$.

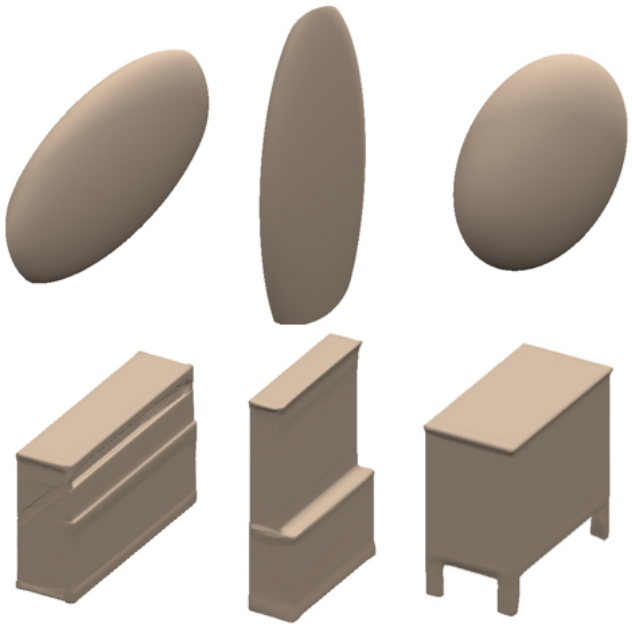

Figure 12. Visualization of the trained object model for cabinets. Upper row: coarse ellipsoid shapes regressed from g_{ϕ} and $\mathbf{z}$. Lower row: SDF object model from f_{θ} and $\mathbf{z}$.

bins, and cabinet. The cabinet and trashbins are reconstructed well, as can be seen from the resulting meshes which resemble the original object shapes. However, the table is poorly reconstructed, since the shape is quite different and the pose is inaccurate. This is because the available observation in the scene for the table is very limited, as can be seen in the segmented mesh, which is insufficient for optimization.

A ScanNet scene with bookshelves and tables are shown in Fig. 14, to demonstrate the usefulness of the coarse and fine level residuals. The figure illustrates that the initialized object pose and shape are different from the actual scene, since the two bookshelves in the center are not parallel and are too small compared to the observation. In contrast, the bookshelves become larger after applying the fine level residual, which is more consistent with the observations. The reconstructions are further improved with both the coarse and fine level residuals, where the bookshelves become parallel. Moreover, the bottom bookshelf and the top right table also become thinner, which agrees more with the observation. This example clearly shows the effectiveness of the proposed bi-level model for joint object pose and shape optimization.

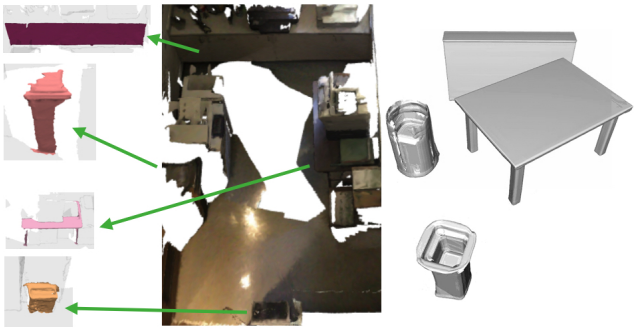

Figure 13. Visualization of the original scene and reconstructed objects for ScanNet scene 0077. The green arrows point to the segmented mesh of the objects.

Pose Estimation Metric

This section presents the metric used to evaluate the object pose, which follows Scan2CAD [4]. We introduce the details on how to decompose a pose $\mathbf{T} \in \mathrm{SIM}(3)$ into rotation $\mathbf{q}$, translation $\mathbf{p}$ and scale $\mathbf{s}$ and the error functions for each element separately. For rotation and scale, $\mathbf{R}_s = \mathbf{PTP}^{\top}$:

$$s_1 = \|\mathbf{R}_s\mathbf{e}_1\|_2 \quad s_2 = \|\mathbf{R}_s\mathbf{e}_2\|_2 \quad s_3 = \|\mathbf{R}_s\mathbf{e}_3\|_2,$$
$$\mathbf{Re}_1 = \frac{\mathbf{R}_s\mathbf{e}_1}{s_1} \quad \mathbf{Re}_2 = \frac{\mathbf{R}_s\mathbf{e}_2}{s_2} \quad \mathbf{Re}_3 = \frac{\mathbf{R}_s\mathbf{e}_3}{s_3}. \tag{16}$$

Suppose $\boldsymbol{R} = \{m_{ij}\}, i, j \in [1, 2, 3]$, we transform it to quaternion $\mathbf{q}$ by

$$q_0 = \frac{\sqrt{\mathrm{tr}(R)+1}}{2}, q_1 = \frac{m_{23}-m_{32}}{4q_0}, q_2 = \frac{m_{31}-m_{13}}{4q_0}, q_3 = \frac{m_{12}-m_{21}}{4q_0}. \tag{17}$$

Suppose the prediction and groundtruth are $\mathbf{q}_{pred}, \mathbf{q}_{gt}$, we compute the difference by

$$e_{\mathrm{SO(3)}}(\mathbf{q}, \hat{\mathbf{q}}) := 2\arccos(|\mathbf{q}_{gt}^{\top}\mathbf{q}_{pred}|). \tag{18}$$

Translation is $\mathbf{p} = \mathbf{T}[1:3, 4]$, and we compare the difference between prediction and groundtruth by

$$\|\mathbf{p}_{pred} - \mathbf{p}_{gt}\|_2. \tag{19}$$

For scale percentage error, we compute it by

$$100 \times |\frac{1}{3}\sum_{i=1}^{3}\bar{s}_i - 1|, \tag{20}$$

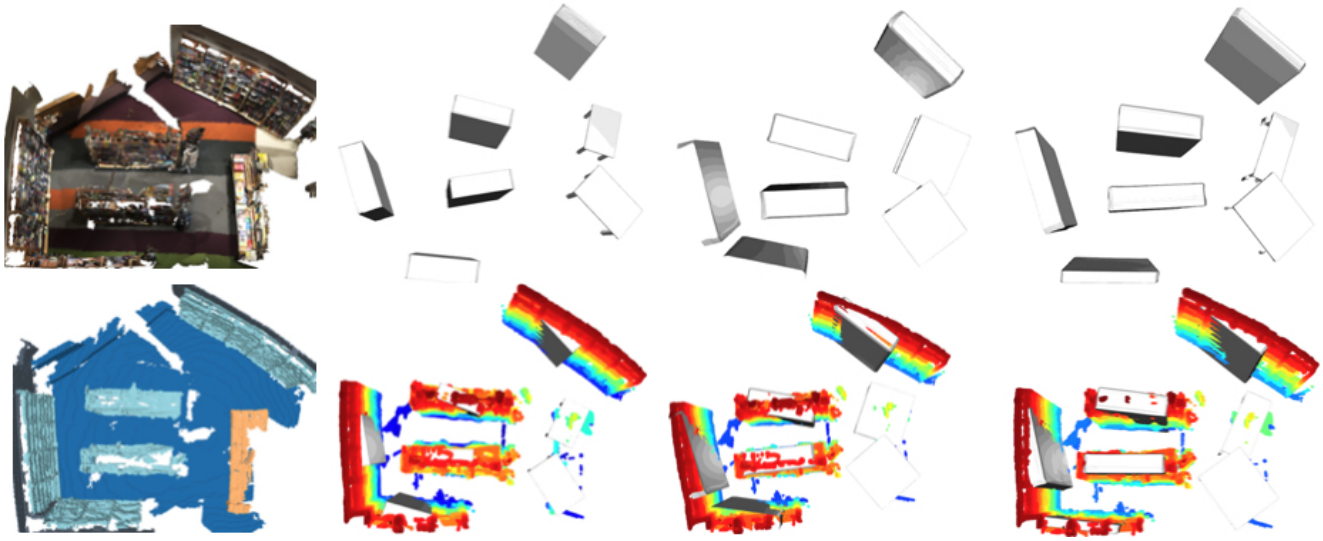

Figure 14. Visualization of the original scene and reconstructed objects for ScanNet scene 0208. First row from left to right: original scene, reconstruction using initialized pose and mean categorical object shape, reconstruction using optimized pose and shape with fine level residual only, reconstruction using optimized pose and shape with both coarse and fine level residuals. Second row from left to right: original scene with bookshelves and tables highlighted in light blue and beige, the rest are reconstructions overlaid with object point clouds and added pseudo points.

where $\bar{s}_i = \frac{s_{pred}}{s_{gt}}$ for each of s_1, s_2, s_3 recovered from the SIM(3) matrix.

Timing

Table 4. ELLIPSDF timing breakdown (sec)

Init	Latent Code Opt	SIM(3) Opt	SDF Decoding	Meshing
0.04	0.13	0.58	1.38	2.34

Timing for one instance is provided in Table 4. *Init* is the pose initialization in (14) for 100 views. *Latent Code Opt* and *SIM(3) Opt* are a single SGD step with respect to $\delta\mathbf{z}$ and $\mathbf{T}$ respectively using 10000 points as batch size. *SDF Decoding* and *Meshing* are optional steps that generate SDF predictions over 256^3 points and apply Marching Cubes to generate a mesh. Our approach does not currently operate in real-time but it is more efficient than existing work. We will investigate how to accelerate the current slow python SIM(3) optimization.

Acknowledgments

The first author would like to thank Kejie Li at University of Adelaide for helpful discussions.