Silhouette-based 6D Object Pose Estimation

Xiao Cui[0009−0006−3166−7483], Nan Li, Chi Zhang, Qian Zhang, Wei Feng, and **Liang Wan**

Tianjin University, Tianjin, China
{cuixiao1998, linan94, zhangchi1736, qianz}@tju.edu.cn,
wfeng@ieee.org, lwan@tju.edu.cn

Abstract. For a long time, deep learning-based 6D object pose estimation networks have lacked the ability to address the problem of pose estimation of the unknown objects beyond the training datasets, due to the closed-set assumption and the expensive cost of high-quality annotation. Conversely, traditional methods struggle to achieve accurate pose estimation for texture-less objects. In this work, we propose a silhouette-based 6D object pose estimation method. being a conventional method As a traditional method, our approach achieves high accuracy without any need of annotation data, demonstrating excellent generalization. Additionally, we employ silhouette to mitigate texture dependency issues, ensuring effectiveness even in the case of textureless objects. In the method, we introduce a dimensionality reduction strategy for SE (3) pose space, accompanied by theoretical proofs, which make it possible to perform pose estimation through search, rendering, and comparison in a reduced-dimensional space efficiently and accurately. Experimental results demonstrate the high precision and generalization of the proposed method. Our code is available at *https://github.com/worldTester/STI-Pose*.

Keywords: Object pose estimation · generalization · silhouette · texture-independent.

1 Introduction

6D object pose estimation from a single image is a classical problem in computer vision. Its objective is to accurately estimate the precise 6D pose of a target object relative to the current camera. This problem plays a crucial role in various tasks, including robotic technology(e.g. automatic manufacturing [22], cooperative assistance [4, 9]), where precise object poses are required to guide grasping, and augmented reality (AR) technology, which relies on determining the real-world object poses for seamless integration with the virtual world [19, 23].

In traditional approaches to object pose estimation, the most common methods are based on correspondence [17]. These methods establish the 2D-3D correspondences between the object in the image and its 3D model and then estimate the object pose through the PnP [15]/RANSAC framework. They typically employ texture-dependent feature point extracting and matching algorithms like SIFT [17] and ORB [24] to establish the 2D-3D correspondences. Consequently, they struggle to handle pose estimation for those objects with weak or no texture. To address this issue, one approach is to adopt template matching-based methods [10, 20]. These methods require generating a collection of images with ground truth object poses from different views prior to usage. Thus, transforming the 6D object pose estimation problem into an image retrieval problem. However, the accuracy of the result obtained through these methods heavily relies on the density of the constructed templates. Another approach to tackle the challenge of pose estimation for weakly-textured or textureless objects involves leveraging depth information [7, 25]. It begins by extracting local shape descriptors of the partial-view point cloud of objects in the image from the current perspective and the complete point cloud on the 3D model. Subsequently, registration is performed to obtain the pose estimation result, thereby circumventing the reliance on object surface texture. However, the limited precision of depth sensors and the constraints of applicable scenarios hinder the widespread usage of these methods.

In recent years, significant progress has been made in the field of object pose estimation, thanks to the advancements in computer vision and deep learning. Numerous deep learning-based approaches have been proposed for pose estimation [26, 31, 30, 21, 29, 6, 13], alleviating many of the challenges encountered by traditional methods. One category of methods involves directly training a single object pose estimation network. These methods employ deep convolutional neural networks to directly regress the position and rotation of the object [29, 31, 6, 13]. Alternatively, an approach is proposed aiming to make the PnP/RANSAC module differentiable [12], enabling end-to-end training of pose estimation networks. Another category of methods achieves higher accuracy in object pose estimation by leveraging neural network outputs that establish sparse or dense 2D-3D correspondence relationships. These methods subsequently estimate the object pose using traditional PnP/RANSAC algorithms [26, 21, 30].

However, deep learning-based approaches inevitably suffer from the closed-set assumption issues, which limit the widespread application of object pose estimation methods. Firstly, such methods require datasets [3, 5, 11] with highly accurate annotations of object poses. The task of annotating 6D poses for individual objects is expensive and the precision is limited. Secondly, training a single object pose estimation network requires a significant amount of time. The state-of-the-art methods that achieve high accuracy often sacrifice the generalization between object instances. They train a network that exclusively serves a single object instance, aiming to maximize the pose estimation capability for that specific object. However, when faced with pose estimation tasks involving multiple objects, it becomes necessary to train multiple network models. Lastly, many downstream tasks of object pose estimation do not prioritize texture but instead focus on the shape information of object, such as robotic arm grasping. These methods heavily rely on texture, requiring the construction of new dataset and retraining networks even for objects with the same shape but different textures.

In response to these issues, we propose a silhouette-based texture-independent object pose estimation method (STI-Pose), which employs an iterative rendering and comparing methodology. By utilizing the 3D model of the object, STI-Pose renders the silhouette in the 6D pose space and compares it with the silhouette of the target object in the reference image, seeking the pose corresponding with the strongest consistency as the estimation result. The method solely relies on silhouettes, which not only avoids the need for object appearance texture but also eliminates the requirement for annotated object poses, exhibiting impressive generalization capabilities.

To address the search problem in the 6D pose space, we introduce a Homography-based Spherical Intersection over Union method (HSIoU), which determines the similarity of camera poses corresponding to two silhouette images while equivalently reducing the search task of the six degrees of freedom in $\mathrm{SE}(3)$ space to three, significantly enhancing the computational efficiency. We provide theoretical derivations to demonstrate the equivalence of this dimensionality reduction strategy. Furthermore, we propose an Optimized Particle Swarm Optimization algorithm, denoted as O-PSO, designed for efficient and robust search within the reduced-dimensional object pose space.

We approach silhouette extraction as an image segmentation task, which is a relatively simpler task compared to object pose estimation. The silhouettes required by our method can be obtained through various means: for cases with a straightforward background, green screen extraction or foreground segmentation methods suffice; for more complex backgrounds, universal segmentation methods such as SAM [14] or SEEM [33] can be employed. None of these segmentation methods impose a closed-set assumption, allowing for segmentation of arbitrary objects.

We validated our method on commonly used datasets [3, 5] for object pose estimation and datasets specifically created for pose estimation of objects with various textures. The results demonstrate that our method is independent of object surface texture, while simultaneously exhibiting high precision and generalizability. Our contributions can be summarized as follows:

1. We propose a silhouette-based object pose estimation method that achieves high accuracy without the requirement of annotated object poses. This breakthrough surpasses the limitation of current networks that can only handle objects in the datasets.

2. Through extensive experiments, we demonstrate the robustness of our method to variations in object appearance, making it highly suitable for real-world applications involving numerous objects with similar geometric structures but different appearances.

2 Related Work

In this section, we will discuss the pose estimation methods most relevant to our work with the input of RGB image, dividing them into two parts: traditional methods and deep learning methods.

2.1 Traditional Methods

Traditional methods often rely on establishing 2D-3D correspondences and utilize the PnP/RANSAC framework to solve object pose estimation, such as [18], achieving high-precision results. However, these methods require the use of feature descriptors such as SIFT [17], SURF [1], or ORB [24], hence struggle to handle textureless objects. To address this issue, a template matching method called LineMod [10] has been proposed. LineMod constructs a large number of templates and utilizes image gradients for template matching, transforming the pose estimation problem into an image retrieval problem. This approach effectively handles textureless objects. However, the accuracy ceiling of template matching methods depends on the density of the templates.

[32] proposed a contour-based pose estimation method for textureless space objects. This method involves extracting the contour of the target object and performing an initial coarse matching with a pre-built library of contour templates. Subsequently, the ORB [24] algorithm is used to establish 2D-2D correspondences between the contours, and the 3D information within the contour templates is used to establish 2D-3D correspondences. The object pose is then computed using the PnP/RANSAC algorithm, thereby improving the accuracy ceiling of template matching methods. However, this method places higher demands on the object's shape, as the contours should not be excessively smooth, as it may cause the 2D-2D correspondences between the contours to fail.

2.2 Methods with Deep Learning

Deep learning methods surpass traditional approaches in terms of both accuracy and computational speed, and they exhibit excellent capability in handling textureless objects.

End-to-end approaches, such as PoseCNN [31], DenseFusion [29], directly regress the pose of the object. [12] attempt to transform PnP/RANSAC into a differentiable module. The end-to-end architecture of these methods enhances their flexibility, enabling them to serve as differentiable pose estimation modules that can be applied to a wider range of tasks.

Non-direct methods, which leverage the powerful regression capability of neural networks, achieve higher prediction accuracy. These methods predict sparse or dense 2D-3D correspondence relationships and subsequently utilize PnP/RANSAC methods to compute the object's pose. Each of these methods employs different approaches to predict the 2D-3D correspondence relationships. PVNet [21] predicts the pixel coordinates of 3D feature points, generating sparse correspondence relationships. GDR-Net [30] divides the object surface into multiple fragments, initially classifying 2D pixel points into a specific fragment and then regressing the offset within that fragment. ZebraPose [26] employs binary encoding for the object's vertices, and the network predicts the corresponding encoding for 2D pixel points, thereby establishing the 2D-3D correspondence relationships. GDR-Net and ZebraPose generate dense correspondence relationships, exhibiting superior performance.

Despite the significant advancements of deep learning methods compared to traditional approaches, they do have certain drawbacks due to their reliance on data. Firstly, these methods necessitate lengthy training on meticulously annotated pose estimation datasets, which can be time-consuming for both dataset creation and training. Secondly, they lack generalizability and can hardly estimate poses for unknown objects not present in the training dataset.

3 The Method

In this section, we propose a silhouette-based object pose estimation method that is texture-independent (STI-Pose). As shown in fig. 1, STI-Pose takes as input the silhouette image of an object in a reference image, along with the corresponding 3D model of the object.

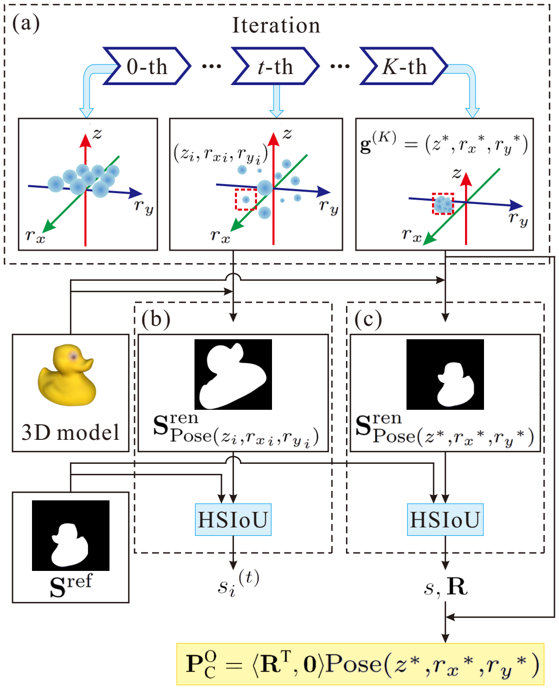

(a)
Iteration
0-th
···
t-th
···
K-th
z
r_y
r_x
(z_i, r_{xi}, r_{y_i})
z
r_y
r_x
$\mathbf{g}^{(K)} = (z^*, r_x^*, r_y^*)$
z
r_y
r_x
3D model
(b)
$\mathbf{S}^{\text{ren}}_{\text{Pose}(z_i, r_{x\,i}, r_{y\,i})}$
(c)
$\mathbf{S}^{\text{ren}}_{\text{Pose}(z^*, r_x^*, r_y^*)}$
HSIoU
HSIoU
$\mathbf{S}^{\text{ref}}$
$s_i^{(t)}$
$s, \mathbf{R}$
$\mathbf{P}^{\text{O}}_{\text{C}} = \langle \mathbf{R}^{\text{T}}, \mathbf{0} \rangle \text{Pose}(z^*, r_x^*, r_y^*)$

Fig. 1. Working flow of the silhouette-based object pose estimation (STI-Pose). The input is silhouette $\mathbf{S}^{\text{ref}}$ and 3D model. (a) is the optimized particle swarm optimization (O-PSO) algorithm to obtain the optimal pose in the reduced space. (b) and (c) are homography-based spherical intersection over union (HSIoU) method to determine the proximity of the camera poses corresponding to the two silhouette images.

We introduce a homography-based spherical intersection over union (HSIoU) method to determine the proximity of the camera poses corresponding to the two silhouette images. By reducing the dimensionality of the object pose space from six dimensions to three, with the help of HSIoU, we use an optimized particle swarm optimization algorithm (O-PSO) in the reduced space to obtain the optimal pose. In the following, we will describe the method in detail.

3.1 Problem Formulation and Notation

In this paper, we use a 3D rotation $\mathbf{R} \in \mathrm{SO}\,(3)$ and a 3D translation $\mathbf{t} \in \mathbb{R}^3$ to indicate the pose $\mathbf{P} \in \mathrm{SE}\,(3)$, i.e., $\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \backsimeq \langle \mathbf{R}, \mathbf{t} \rangle$. We use the uppercase and lowercase subscripts to indicate the relative relationship, specifically, $\mathbf{P}_{\mathrm{A}}^{\mathrm{B}}$ denotes the pose of coordinate system B relative to coordinate system A. We use the capital letter C to indicate the camera while the capital letter O to indicate the object. In addition, we utilize Euler angles to represent the rotation matrix, denoted as r_x, r_y, and r_z, respectively. For the camera model, we denote the intrinsic parameter matrix as $\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$. In addition, the homogeneous coordinates of a 3D point $\mathbf{X} \in \mathbb{R}^3$ are represented as $\tilde{\mathbf{X}} = [x, y, z, 1]^{\mathrm{T}}$, while the homogeneous coordinates of a 2D pixel point $\mathbf{p} \in \mathbb{R}^2$ are represented as $\tilde{\mathbf{p}} = [u, v, 1]^{\mathrm{T}}$.

We transform the pose estimation problem into an optimization problem in SE (3) space, specifically as follows:

$$\mathbf{P}_{\mathrm{C}}^{\mathrm{O}^*} = \underset{\mathbf{P}_{\mathrm{C}}^{\mathrm{O}} \in \mathrm{SE}(3)}{\arg\min} \left\| \mathbf{S}_{\mathbf{P}_{\mathrm{C}}^{\mathrm{O}}}^{\mathrm{ren}} - \mathbf{S}^{\mathrm{ref}} \right\|_2, \tag{1}$$

where $\mathbf{S}^{\mathrm{ref}}$ is the reference silhouette image and $\mathbf{S}_{\mathbf{P}_{\mathrm{C}}^{\mathrm{O}}}^{\mathrm{ren}}$ is the rendered silhouette image using object pose $\mathbf{P}_{\mathrm{C}}^{\mathrm{O}}$. Obtaining the global maximum in the six-dimensional SE (3) space is indeed a hard task. However, STI-Pose allows for efficient and stable identification of the optimal pose. It is worth mentioning that the 3D models used in this paper consist of triangular mesh representations, containing solely geometric shape information and devoid of any texture information.

3.2 Dimensionality Reduction

The key factor to the accurate object pose estimation through iterative search, rendering, and comparison is to determine the similarity of the camera pose corresponding to the reference silhouette image and the rendered silhouette image, thereby deciding the search termination, and obtaining the final estimated object pose. In this section, we propose a homography-based spherical intersection over union (HSIoU) method to determine the proximity of the camera poses corresponding to the two silhouette images. At the same time, we provide detailed instructions on how to effectively reduce the search task of six degrees of freedom in SE (3) space to three degrees.

Homography-based spherical intersection over union The process of the HSIoU is illustrated in Fig. 2. Given two camera images of an object silhouette, $\mathbf{S}_1$ and $\mathbf{S}_2$, captured by cameras with the same intrinsic parameters $\mathbf{K}$ but different poses, HSIoU computes the proximity of the translation vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ between the unknown camera poses $\mathbf{P}_{\mathrm{O}}^{\mathrm{C}_1} \simeq \langle \mathbf{R}_1, \mathbf{t}_1 \rangle$ and $\mathbf{P}_{\mathrm{O}}^{\mathrm{C}_2} \simeq \langle \mathbf{R}_2, \mathbf{t}_2 \rangle$. Additionally, the algorithm also outputs the relative rotation $\mathbf{R} = \mathbf{R}_1^{\mathrm{T}} * \mathbf{R}_2$ when the proximity s is high enough.

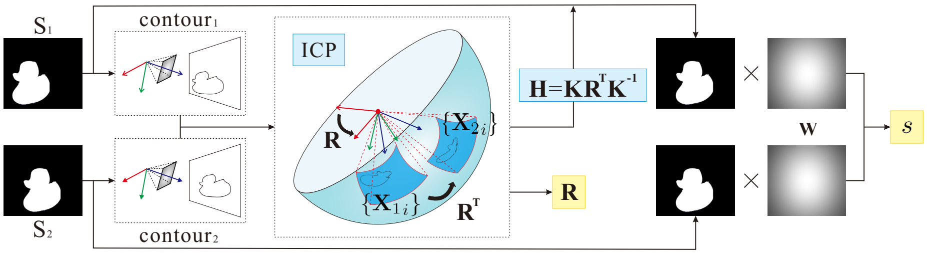


Fig. 2. Illustration of the homography-based spherical intersection over union (HSIoU) method.

Theorem 1. *When $\mathbf{t}_1$ is equal to $\mathbf{t}_2$ and the only difference between $\mathbf{P}_{\mathrm{O}}^{\mathrm{C}_1}$ and $\mathbf{P}_{\mathrm{O}}^{\mathrm{C}_2}$ lies in rotation $\mathbf{R}$, there exists a transformation relationship between $\mathbf{S}_1$ and $\mathbf{S}_2$ through a homography matrix $\mathbf{H}$. Applying $\mathbf{H}$ to $\mathbf{S}_1$ yields $\mathbf{S}_1{}'$, which perfectly aligns with $\mathbf{S}_2$.*

The proof is as follows.

Proof. For the 2D pixel points $\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2$ which are projections of a spatial point $\mathbf{X}$ onto the two image planes of the cameras, we have

$$\begin{aligned} \tilde{\mathbf{p}}_1 &= \mathbf{KX}, \\ \tilde{\mathbf{p}}_2 &= \mathbf{KR}^{\mathrm{T}}\mathbf{X} = \mathbf{KR}^{\mathrm{T}}\mathbf{K}^{-1}\tilde{\mathbf{p}}_1. \end{aligned} \tag{2}$$

Thus, $\tilde{\mathbf{p}}_2 = \mathbf{H}\tilde{\mathbf{p}}_1$ when defining $\mathbf{H} = \mathbf{KR}^{\mathrm{T}}\mathbf{K}^{-1}$. Hence, there exists a homography transformation between the pixel points of $\mathbf{S}_1$ and $\mathbf{S}_2$. Applying this transformation to $\mathbf{S}_1$ yields $\mathbf{S}_1{}'$, the Intersection over Union (IoU) between $\mathbf{S}_1{}'$ and $\mathbf{S}_2$ is always 1.

The $\mathbf{H}$ matrix is easily computable. Firstly, we utilize the contour extraction algorithm [27] to extract the contour points from $\mathbf{S}_1$ and $\mathbf{S}_2$. We then back-project these points onto the unit sphere using the camera intrinsic parameters

K, resulting in point clouds $\{\mathbf{X}_{1i}\}$ and $\{\mathbf{X}_{2i}\}$. Next, we employ a specialized ICP [2] algorithm to align the two point clouds and obtain **R**, from which we derive **H**. The specialized ICP algorithm only applies rotation operations to the point clouds, disregarding translation. It is important to note that the projection onto the unit sphere is necessary to ensure that all points in $\{\mathbf{X}_{1i}\}$ and $\{\mathbf{X}_{2i}\}$ have equal distances from the camera optical center. This requirement satisfies the prerequisites of the specialized ICP algorithm for aligning the point clouds. When $\mathbf{t}_1$ is not equal to $\mathbf{t}_2$, we can still calculate the IoU using the aforementioned process. Note that, in this case, the IoU value will always be less than 1, and we can use it to quantify the proximity between $\mathbf{t}_1$ and $\mathbf{t}_2$.

However, performing IoU calculations on a pixel plane can be susceptible to the influence of the perspective effect, leading to unstable IoU values. Consequently, we have introduced a weight map **W** during the IoU computation to ensure that the results are equivalent to performing IoU calculations on the unit sphere, as represented by Eq. (3), where i and j represent pixel coordinates.

$$\mathrm{IoU}(\mathbf{S}_1, \mathbf{S}_2, \mathbf{W}) = \frac{\sum\limits_{(i,j)\in\mathbf{S}_1\cap\mathbf{S}_2} \mathbf{W}(i,j)}{\sum\limits_{(i,j)\in\mathbf{S}_1\cup\mathbf{S}_2} \mathbf{W}(i,j)} \in (0,1]. \tag{3}$$

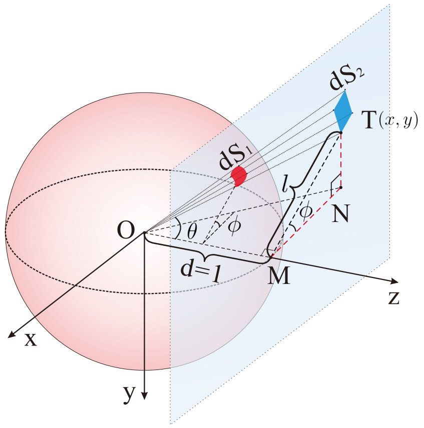


Fig. 3. Derivation illustration of the weight map **W**

The weight map **W** has the same size (width W and height H) as $\mathbf{S}_1$ and $\mathbf{S}_2$, where the value of each pixel represents the ratio of the area occupied by that pixel on the unit sphere to its area on the normalized plane. The creation of **W** is solely dependent on the camera intrinsic parameters **K**. Fig. 3 and Eqs. (4-6) give the derivation process.

According to the back-projection relationship of camera internal parameters, we have the coordinate (x, y) of the point $\mathbf{T}$ in the normalized plane,

$$x = \frac{i - c_x}{f_x}, \qquad y = \frac{j - c_y}{f_y}, \tag{4}$$

where $1 \leq i \leq \mathrm{W}, 1 \leq j \leq \mathrm{H}$. Thus, The distance between $\mathbf{T}$ and the vertical point $\mathbf{M}$ of the optical center $\mathbf{O}$ on the normalized plane is

$$\|\mathbf{MT}\| = l = \tan(\theta) = \sqrt{x^2 + y^2}. \tag{5}$$

Hence, the weight map $\mathbf{W}$ (the ratio of the infinitesimal area element dS_1 on the unit sphere to the corresponding infinitesimal area element dS_2 on the normalized plane) can be calculated by

$$\begin{aligned} \mathbf{W}(i, j) &= \frac{\mathrm{dS}_1}{\mathrm{dS}_2} = \frac{\sin(\theta) \cdot \mathrm{d}\phi \cdot \mathrm{d}\theta}{l \cdot \mathrm{d}\phi \cdot \mathrm{d}l} = \frac{\sin(\theta)}{\tan(\theta) \cdot \frac{\mathrm{d}l}{\mathrm{d}\theta}} \\ &= \cos^3(\theta) \\ &= ((\frac{i - c_x}{f_x})^2 + (\frac{j - c_y}{f_y})^2 + 1)^{-\frac{3}{2}} \in (0, 1]. \end{aligned} \tag{6}$$

By introducing the weight map $\mathbf{W}$ and modifying the IoU calculation, we refer to the algorithmic process described above as HSIoU, which can be represented by Eq. (7).

$$s, \mathbf{R} = \mathrm{HSIoU}(\mathbf{S}_1, \mathbf{S}_2, \mathbf{K}), \tag{7}$$

where the proximity s is the result of Eq. (3), it solely reflects the proximity between $\mathbf{t}_1$ and $\mathbf{t}_2$, with no relation to $\mathbf{R}_1$ and $\mathbf{R}_2$. On the other hand, $\mathbf{R} = {\mathbf{R}_1}^{\mathrm{T}}\mathbf{R_2}$ represents the disparity between the unknown $\mathbf{R}_1$ and $\mathbf{R}_2$, and its value is meaningful only when s approaches 1.

The dimensionality reduction by HSIoU HSIoU gives a naive method for object pose estimation, it can effectively reduce the search task of six degrees of freedom in SE (3) space to three. It allows us to first determine the translation $\mathbf{t}$ of the camera pose, and then obtain the rotation R to accomplish pose estimation. To obtain the accurate translation vector $\mathbf{t}$ of the camera pose $\mathbf{P}_{\mathrm{O}}^{\mathrm{C}} \simeq \langle \mathbf{R}, \mathbf{t} \rangle$, we traverse the $\mathbb{R}^3$ space. When $\mathbf{t}$ takes the value $\mathbf{t}_i$, since the rotation $\mathbf{R}$ does not affect the computation of HSIoU, we can set it as an arbitrary rotation matrix $\mathbf{R}_i$. We set the z-axis of camera points towards the origin of the object coordinate system for convenience. This configuration is illustrated in Fig. 4 (a).

We denote these poses as $\{\mathbf{R}_i, \mathbf{t}_i\}$ and use them to render silhouettes $\{\mathbf{S}^{\mathrm{ren}}_{\langle \mathbf{R}_i, \mathbf{t}_i \rangle}\}$. We then compute HSIoU with respect to the reference silhouette $\mathbf{S}^{\mathrm{ref}}$. Then, we can obtain the pose $\langle \mathbf{R}^*, \mathbf{t}^* \rangle$ that yields the maximum $s \approx 1$, along with the corresponding $\mathbf{R}$. This can be expressed using Eq. (8).

$$s, \mathbf{R} = \mathrm{HSIoU}(\mathbf{S}^{\mathrm{ren}}_{\langle \mathbf{R}^*, \mathbf{t}^* \rangle}, \mathbf{S}^{\mathrm{ref}}, \mathbf{K}). \tag{8}$$

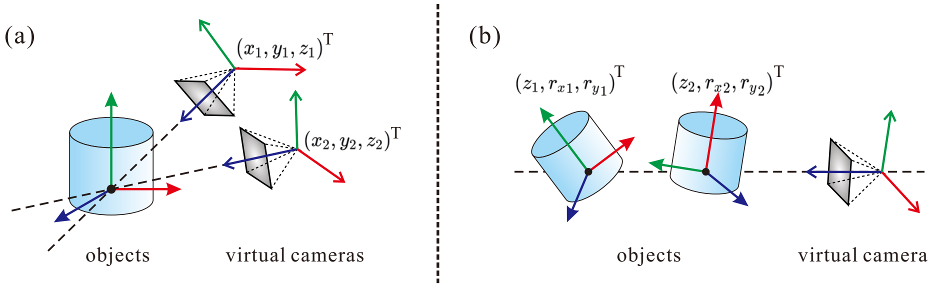

(a)
$(x_1, y_1, z_1)^{\mathrm{T}}$
$(x_2, y_2, z_2)^{\mathrm{T}}$
objects
virtual cameras
(b)
$(z_1, r_{x1}, r_{y1})^{\mathrm{T}}$
$(z_2, r_{x2}, r_{y2})^{\mathrm{T}}$
objects
virtual camera

Fig. 4. Two types of reduced-dimensional pose spaces. (a) is the dimensionality reduction of the **camera pose space** after reduction. (b) is the dimensionality reduction of the **object pose space** after reduction.

In this way, the camera pose can be represented as $\mathbf{P}_{\mathrm{O}}^{\mathrm{C}} \simeq \langle \mathbf{R}^{*}\mathbf{R}, \mathbf{t}^{*} \rangle$, and the object's pose can be obtained as $\mathbf{P}_{\mathrm{C}}^{\mathrm{O}} = {\mathbf{P}_{\mathrm{O}}^{\mathrm{C}}}^{-1}$. During this computation process, we traverse only the translation vector $\mathbf{t} = [x, y, z]^{\mathrm{T}} \in \mathbb{R}^3$, thus, reducing the dimensions that need to be searched in the pose space.

However, this approach has several limitations. Firstly, traversing only the translation vector $\mathbf{t}$ is not the optimal choice. Considering that the individual influence of the three dimensions of $\mathbf{t}$ on the silhouette does not vary significantly, expressing $\mathbf{t}$ in spherical coordinates (r, θ, ϕ) would magnify this difference. Specifically, r primarily influences the size of the area of $\mathbf{S}^{\text{ren}}$, while θ and ϕ have a greater impact on the shape of $\mathbf{S}^{\text{ren}}$. This representation can achieve a certain level of decoupling, providing better properties for exploration within the pose space. Secondly, the approach calculates the camera pose $\mathbf{P}_{\mathrm{O}}^{\mathrm{C}}$ and then converts it into the object pose $\mathbf{P}_{\mathrm{C}}^{\mathrm{O}}$, which may seem less straightforward.

Based on the aforementioned approach and its shortcomings, we reduce the dimensionality of the pose space and provide a more precise definition. This dimensionality reduction method is more concise and rational. As shown in Fig. 4 (b), the coordinate of the reduced-dimensional space is denoted as (z, r_x, r_y), where z represents the z-coordinate of the object in the camera coordinate system, while r_x and r_y denote the object's Euler angles around the x and y axes, respectively, in the camera coordinate system. The dimensions that have been reduced are x, y, and r_z, which are set to 0. $x = y = 0$ signifies that the origin of the object lies on the z-axis of the camera coordinate system, aligning with the earlier approach. Considering that r_z corresponds to the in-plane rotation of the camera, we can indeed set $r_z = 0$.

Thus, to express the mapping relationship from the reduced-dimensional space to the 6D pose space, we employ the notation $\mathbf{P}_{\mathrm{C}}^{\mathrm{O}} = \mathrm{Pose}(z, r_x, r_y)$. Given $\mathbf{S}^{\text{ref}}$, we traverse the (z, r_x, r_y) coordinate space, rendering $\{\mathbf{S}^{\text{ren}}{}_{\mathrm{Pose}(z_i, r_{x\,i}, r_{y\,i})}\}$ with $\{\mathrm{Pose}(z_i, r_{x\,i}, r_{y\,i})\}$ and calculating the HSIoU with respect to $\mathbf{S}^{\text{ref}}$. This process yields the coordinates $(z^*, {r_x}^*, {r_y}^*)$ that maximize the proximity measure s produced by HSIoU.

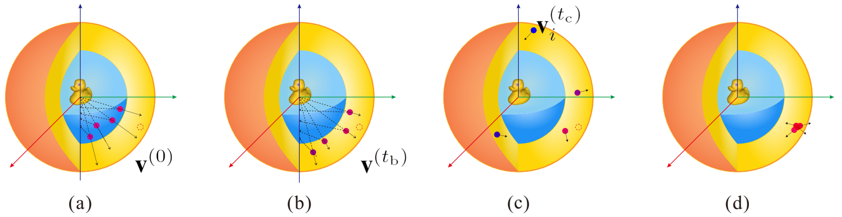
$$s, \mathbf{R} = \mathrm{HSIoU}(\mathbf{S}^{\mathrm{ren}}_{\mathrm{Pose}(z^*, {r_x}^*, {r_y}^*)}, \mathbf{S}^{\mathrm{ref}}, \mathbf{K}). \tag{9}$$

Based on the aforementioned calculations, the 6D pose of the object can be expressed as:

$$\mathbf{P}^{\mathrm{O}}_{\mathrm{C}} = \langle \mathbf{R}^{\mathrm{T}}, \mathbf{0} \rangle \mathrm{Pose}(z^*, {r_x}^*, {r_y}^*). \tag{10}$$

3.3 Optimized Particle Swarm Optimization

In the previous section, we discussed how to reduce the dimensionality of the object pose space but did not provide a detailed explanation of how to search for the global maximum of the variable s in the HSIoU algorithm. In this section, we utilize an optimized particle swarm optimization (O-PSO) algorithm to accurately and reliably accomplish this task, thereby achieving a good object pose estimation.


$\mathbf{v}^{(0)}$
$\mathbf{v}^{(t_\mathrm{b})}$
$\mathbf{v}_i^{(t_\mathrm{c})}$

(a) (b) (c) (d)

Fig. 5. Execution process of the optimized particle swarm optimization (O-PSO) algorithm. The color of the particle reflects its corresponding s of HSIoU, with red indicating a larger value and blue indicating a smaller value. (a) depicts the initialization of the particle swarm, which is on the spherical surface with a radius of z_{near} and moving radially. (b) represents the motion during the first stage (prior to P iterations). The particles gradually accelerate in the radial direction until they reach the radius of z_{far}. (c) showcases the motion during the second stage (after P iterations), where particles start acquiring tangential velocity. (d) illustrates the convergence of O-PSO and the particles gather around the peak of s.

Due to the non-differentiability of HSIoU, it is not possible to compute the gradients with respect to (z, r_x, r_y). Therefore, we choose the Particle Swarm Optimization (PSO) algorithm [8], which is suitable for finding maximum points in spaces with unknown gradients. However, PSO is prone to get trapped in local maxima, and to mitigate this issue, we need a larger number of particles to cover a wider range, which can affect the convergence speed of the algorithm. As shown in Fig. 5, taking into account the characteristics of the object pose space after dimensionality reduction, we propose an initialization and movement strategy for the particle swarm. This strategy effectively addresses the local optima problem without introducing an excessive number of particles.

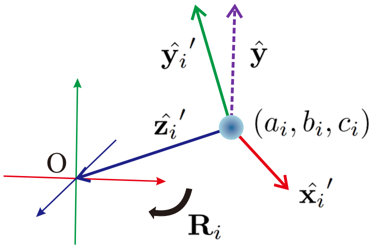


Fig. 6. Initialization for the coordinate of the particle swarm.

Particle Swarm Initialization. Since r_x and r_y in the (z, r_x, r_y) space represent angles, it follows that r_x and r_y are in the interval $(-\pi, \pi]$. For the variable z, we set a search range of $[z_{\text{near}}, z_{\text{far}}]$. Therefore, the search range for the O-PSO algorithm is $[z_{\text{near}}, z_{\text{far}}] \times (-\pi, \pi] \times (-\pi, \pi]$. To initialize the particles, we first assign initial coordinates to each particle. Assuming we have N particles, we utilize the Fibonacci sphere algorithm to uniformly sample N points on a sphere centered at the object with a radius of 1. As shown in Fig. 6, we use (a_i, b_i, c_i) to represent the coordinates of the i-th particle,

$$\begin{cases} b_i &= 1 - \frac{2i}{N-1}, \\ a_i &= \sqrt{1 - {b_i}^2} \cdot \cos(i\phi), \\ c_i &= \sqrt{1 - {b_i}^2} \cdot \sin(i\phi), \\ \phi &= (\sqrt{5} - 1)\pi, \end{cases} \tag{11}$$

where ϕ represents the golden angle in radians. Then, we convert $\{(a_i, b_i, c_i)\}$ to the reduced object pose space, represented as $\{(z_i, r_{xi}, r_{y_i})\}$,

$$\begin{cases} \hat{\mathbf{y}} &= [0, 1, 0]^{\text{T}}, \\ \hat{\mathbf{z}}_i{}' &= -[a_i, b_i, c_i]^{\text{T}}, \\ \hat{\mathbf{x}}_i{}' &= \hat{\mathbf{y}} \times \hat{\mathbf{z}}_i{}', \\ \hat{\mathbf{y}}_i{}' &= \hat{\mathbf{z}}_i{}' \times \hat{\mathbf{x}}_i{}'. \end{cases} \tag{12}$$

Hence, we have the rotation matrix $\mathbf{R}_i$ of an object relative to the coordinate system of the i-th particle,

$$\mathbf{R}_i = [\frac{\hat{\mathbf{x}}_i{}'}{\|\hat{\mathbf{x}}_i{}'\|}, \frac{\hat{\mathbf{y}}_i{}'}{\|\hat{\mathbf{y}}_i{}'\|}, \hat{\mathbf{z}}_i{}']^{\text{T}}. \tag{13}$$

In particular, we use the Euler angle to represent the $\mathbf{R}_i$, that is $\mathrm{R}_i \simeq \langle r_{xi}, r_{y_i}, r_{zi} \rangle$. Subsequently, we replace the z_i values in $\{(z_i, r_{xi}, r_{y_i})\}$ with z_{near}, resulting in $\{(z_{\text{near}}, r_{xi}, r_{y_i})\}$. This signifies that all N particles are located on a spherical surface with a radius of z_{near} from the center of the object, which is shown in Fig. 5(a). Thus, we consider $\{(z_{\text{near}}, r_{xi}, r_{y_i})\}$ as the initial coordinates for the particles.

Particle Swarm Movement Strategy. Firstly, we define a random vector $\mathbf{X}$,

$$\mathbf{X} = (X_1, X_2, X_3) \quad \text{s.t.} \quad \mathbf{X}_i \sim \mathrm{U}(0, 1). \tag{14}$$

Then the the velocity $\mathbf{v}_i^{(t)} = [v_{z\,i}, v_{r_x\,i}, v_{r_y\,i}]^{\mathrm{T}}$ of the i-th particle at the t-th iteration can be represented as

$$\mathbf{v}_i^{(t)}=\begin{cases} [(z_{\text{far}}-z_{\text{near}})\frac{e^{\frac{k}{\mathrm{P}-1}}-1}{e^k-1}e^{\frac{k}{\mathrm{P}-1}(t-1)}, 0, 0]^{\mathrm{T}}, & t \leq P-1 \\ \omega\mathbf{v}_i^{(t-1)} + c_1(\mathbf{p}_i^{(t-1)} - \mathbf{x}_i^{(t-1)}) + & \\ c_2\mathbf{X}(\mathbf{g}^{(t-1)} - \mathbf{x}_i^{(t-1)}), & P \leq t \leq K \end{cases} \tag{15}$$

where $\mathbf{p}_i$ represents the position corresponding to the maximum value of s encountered during the traversal by the i-th particle, while $\mathbf{g}$ denotes the one by all particles, k is a hyperparameter. And the coordinate $\mathbf{x}_i^{(t)} = [z_i, r_{x\,i}, r_{y\,i}]^{\mathrm{T}}$ of the i-th particle at the t-th iteration in Eq.(15) is

$$\mathbf{x}_i^{(t)} = \begin{cases} \mathbf{p}_i^{(t-1)}, & t = P \\ \mathbf{x}_i^{(t-1)} + \mathbf{v}_i^{(t-1)}, & t \neq P \end{cases} \tag{16}$$

In the first P iterations, the algorithm is at the first stage, which is shown in 5 (b). The velocities v_{r_x} and v_{r_y} are set to 0, while v_z increases incrementally. This setting indicates that the particles only accelerate radially in relation to the center of the object. This choice is made because, at larger distances from the center of the object, the impact on silhouette size from the same distance becomes less significant. Hence, we allow v_z to increase with each iteration, resulting in a longer distance. Upon completion of the P iterations, we set the coordinate $\mathbf{x}_i$ of the particle to the maximum point, $\mathbf{p}_i$, it has traversed. After P iterations is the second stage, where we proceed with the standard PSO algorithm. Upon convergence of the particle swarm (as shown in Fig. 5 (d)), we obtain $\mathbf{g} = (z^*, {r_x}^*, {r_y}^*)$, which enables us to compute the object pose $\mathbf{P}_{\mathrm{C}}^{\mathrm{O}}$ using Eqs. (9-10).

4 Experiments

In this section, we will substantiate the high precision, texture independence, excellent generalization, and numerical stability of STI-Pose through a series of experiments. In this context, the distinctive characteristic of STI-Pose will be emphasized.

4.1 Experiments Setup

Implementation Details. Our approach involves rendering silhouette images based on object poses and comparing them with reference silhouette images. We utilize OpenGL for image rendering and use a fragment shader to output white color, enabling direct rendering of silhouettes. For contour extraction, we employ the "findContours" function from OpenCV. Additionally, if multiple contours are detected, we select the one with the maximum length to eliminate noise interference. The pure rotational ICP algorithm is implemented by adapting the source code of the point cloud registration algorithm from Open3D. We perform dense interpolation on the back-projected point cloud of the reference silhouette image contours, while no processing is applied to the back-projected point cloud of the rendered silhouette image contours. This minimizes point cloud registration errors as much as possible. The scale map can be computed offline and stored since it only depends on the image size and the camera intrinsic $\mathbf{K}$.

We have implemented the particle swarm optimization algorithm ourselves, with the following settings in Eqs. (15,16): $k = 2$, inertia weight $\omega = 0.8$, acceleration coefficients $c_1 = c_2 = 0.5$, and a maximum iteration limit of $K = 200$. For all datasets, the search space for the particle swarm optimization algorithm is constrained with $z_{\text{near}} = 400$mm, $z_{\text{far}} = 1400$mm. P is set to 20, and the number of particles N is set to 50.

Datasets. Currently, the commonly used object pose estimation datasets includes LM-O [3] and YCB-V [5] . LM-O consists of 8 objects, with a higher proportion of textureless objects. On the other hand, YCB-V comprises 21 objects, most of which are symmetrical and have textured surfaces. Since our proposed method requires complete silhouettes, we filtered the test sets of these two datasets based on occlusion conditions and conducted experiments only on datasets with occlusion rates below 10%. In our experiments, we employ the combination of bounding box detector FCOS [28] and SAM [14] to obtain the object silhouettes. The FCOS detector is provided by CDPNv2 [16].

To demonstrate the STI-Pose is texture-independent, we created two dataset of objects with various surface textures. The first dataset based on the YCB-V dataset called YCB-V-NT(YCB-V with new texture), which is a virtual rendering dataset. Specifically, we replaced the texture maps of the original objects in YCB-V with three different texture images. We then rendered the objects using the ground truth object poses and synthesized the rendered images with the original ones. The synthesized data is illustrated in Fig. 7. Additionally, we has curated an dataset comprising 155 images collected from the real world, called texture replacement dataset from real world(TR-RW). The dataset includes three variations of identical-shaped cans and two types of industrially molded components with distinct textures. Pose annotations were manually obtained for accurate positioning. The data is illustrated in Fig.. 8.

Error Metrics. We employ the commonly utilized ADD(-S) metric for the task of object pose estimation. The ADD metric assesses whether the average deviation of the transformed model points falls below 10% of the object's diameter. In the case of symmetric objects, the ADD-S metric is utilized to measure the error as the average distance to the nearest model point. Additionally, we utilize the Area Under the Curve (AUC) of the ADD(-S) with a maximum threshold of 10 cm.

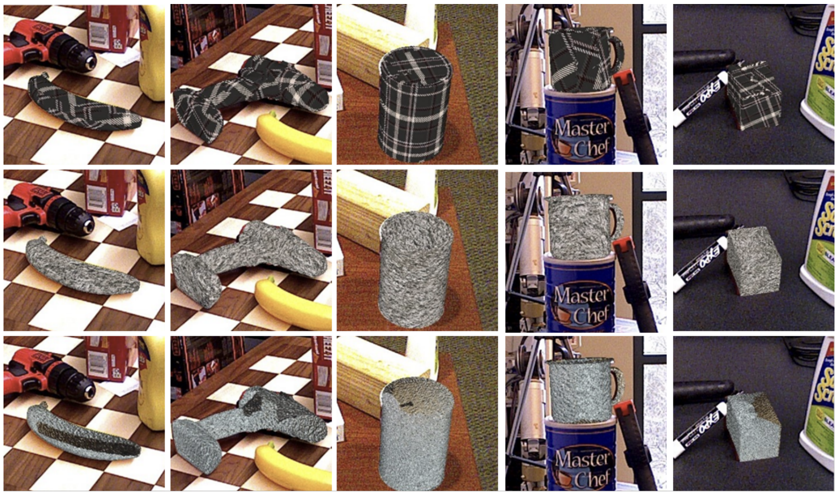


Fig. 7. Samples from the dataset YCB-V-NT rendered using three different textures. The first, second, and third rows correspond to the textures of grid, stone, and metal, respectively.

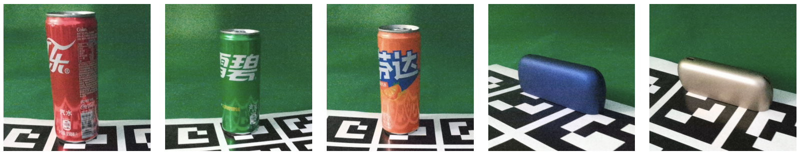


Fig. 8. Samples from the TR-RW dataset.

It is worth noting that, as our method is texture-independent, for objects with shape symmetry, we will consistently use the ADD-S metric for comparison with other methods, without considering symmetry in texture.

4.2 Comparison to State of the Art

We compared our STI-Pose with the state-of-the-art methods on the unoccluded data from YCB-V and LM-O datasets to demonstrate the high accuracy.

Results on YCB-V. We present the results of ADD(-S) and its corresponding AUC in Table 1. Both ZebraPose [26] and GDR-Net [30] are deep learning approaches based on 2D-3D correspondences, exhibiting exceptional accuracy. Since our method relies solely on silhouettes and does not consider the internal textures of objects, we solely consider the object shape symmetry. In the experimental process, for shape-symmetric objects, Zebrapose, GDR-Net, and the proposed STI-Pose, all employ ADD-S for evaluation. To ensure fair comparisons, Zebrapose and GDR-Net utilize pre-trained models provided by their

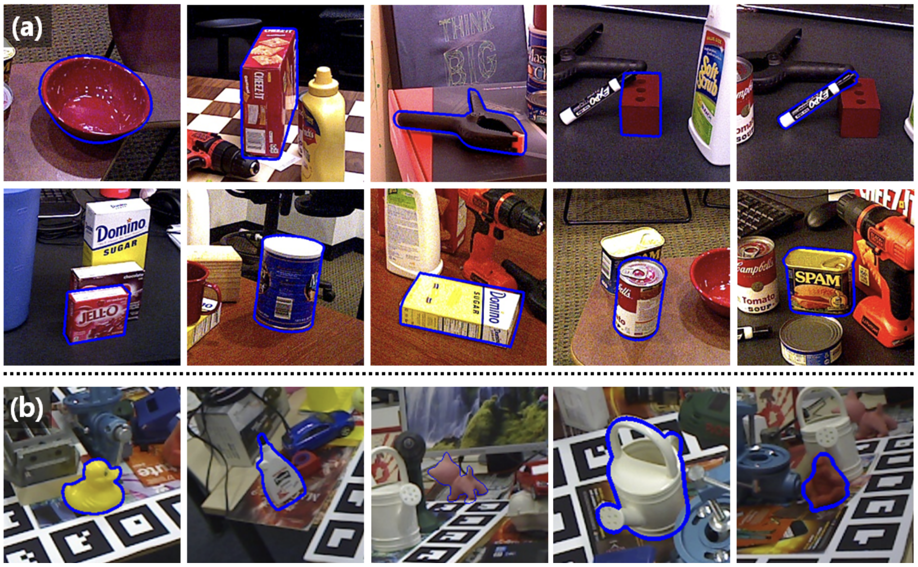


Fig. 9. Visualization of pose estimation results using STI-Pose on the YCB-V and LM-O datasets. The estimated poses are represented by blue contours overlaid on the reference images. (a) displays visualizations from the YCB-V dataset, and (b) shows visualizations from the LM-O dataset.

authors. Additionally, since both methods require RoI as input, we employ the same detector, FCOS [28], to obtain the RoI images.

Experimental results demonstrate that STI-Pose performs the best. Moreover, STI-Pose extremely accurately estimates the poses of textureless objects such as the bowl and banana. Our method outperforms others in both metrics for the bowl object, owing to its symmetry and lack of texture. Thus, deep learning methods struggle to learn pose-related features based on texture or shape, whereas our approach solely utilizes silhouette, eliminating such limitations.

Results on LM-O. We compared the STI-Pose with the methods presented in Table 2. The experiment shows that although STI-Pose does not achieve the highest accuracy, it exhibits performance comparable to state-of-the-art methods. It is noteworthy that in LM-O, a significant portion of the object lacks surfaces texture or have no texture, yet STI-Pose, a conventional approach, achieves sufficiently high precision.

It is noteworthy that our methods were directly tested on the evaluation dataset, confirming the generalization capability of STI-Pose for object pose estimation on various objects.

Method	ZebraPose		GDR-Net		STI-Pose	
	ADD(-S)	AUC-ADD(-S)	ADD(-S)	AUC-ADD(-S)	ADD(-S)	AUC-ADD(-S)
master_chef_can	100	94.4	98.3	93.4	100	96.5
cracker_box	100	85.5	100	97.2	100	97
sugar_box	100	94.5	100	95.9	97.3	93.1
tomato_soup_can	100	96.2	100	94.2	100	95
mustard_bottle	100	96.4	100	95.3	100	96
tuna_fish_can	97.3	95.3	94.6	95.4	85	94.1
gelatin_box	86.8	94.7	88.9	94.1	100	94.4
potted_meat_can	100	95.2	100	90.3	100	94.8
banana	100	90.0	100	92.8	100	89.2
pitcher_base	100	92.9	100	90.3	89.7	85.5
bleach_cleanser	100	91.1	97.8	89.7	100	92.4
bowl	62.5	78.5	74.9	81.8	100	95.4
mug	76.0	89.7	72.1	90.5	60	80.8
power_drill	98.8	90.5	100	92.3	100	93.9
large_clamp	98.1	91.0	92.4	83.3	95.2	93.2
extra_large_clamp	100	94.6	100	90.3	93.8	97.6
foam_brick	100	95.2	100	94.6	100	95.8
mean	95.3	92.7	95.2	91.8	**95.4**	**93.2**

Table 1. Comparison results between STI-Pose and other state-of-the-art methods on the YCB-V dataset. The table showcases the ADD(-S) and AUC-ADD(-S) metrics for each object in %.

4.3 Performance on YCB-V-NT and TR-RW

Although it is evident that utilizing silhouettes allows our method to be texture-agnostic, we still conducted experiments on two self-constructed datasets YCB-V-NT and TR-RW, which fully illustrate that existing methods lack texture generalization.

We conducted comparative experiments on the YCB-V-NT dataset, comparing it with the state-of-the-art ZebraPose, which has shown excellent performance in deep learning approaches. The experimental results are presented in Table 3, where it can be observed that STI-Pose maintains high accuracy even on the texture-replaced dataset, while the deep learning methods struggle to achieve correct pose estimation. This indicates that deep learning approaches fundamentally rely heavily on extracting features from the surface texture of objects, and their training on data with a specific texture does not generalize well to objects with different textures. In contrast, our method only requires the input of silhouettes and is completely independent of object surface textures. As a result, it naturally possesses texture generalization capabilities.

Furthermore, we also tested STI-Pose on our self-constructed TR-RW dataset, as shown in Table 4 The experimental results on cans and injection-molded samples demonstrate that variations in object surface textures, when the shapes are

Method	ADD-S	AUC of ADD-S
ZebraPose	**91.2**	88.1
GDR-Net	78.1	89.6
RePose	80.4	86.5
SO-Pose	74.3	88.9
Ours	90.4	**90.0**

Table 2. Comparison with state-of-the-art methods on LM-O. We compare our STI-Pose with these methods using metrics of ADD-S, AUC of ADD-S in %.

Method	ZebraPose		STI-Pose	
	ADD(-S)	AUC-ADD(-S)	ADD(-S)	AUC-ADD(-S)
YCB-V	95.3	92.7	**95.4**	**93.2**
YCB-V-NT	4.6	13.3	**91.3**	**92.9**

Table 3. Comparison with ZebraPose on YCB-V and YCB-V-NT. We compared our STI-Pose with the state-of-the-art deep learning pose estimation method, ZebraPose, using the metrics of ADD(-S) and AUC-ADD(-S) in %.

the same, have negligible impact on the accuracy of our approach. This further validates the texture-agnostic nature of STI-Pose.

Objects/Metrics	ADD(-S)	AUC-ADD(-S)
Coca-Cola can	96.8	97.6
Sprite can	100	98.1
Fanta can	100	97.8
Blue injection-molded part	96.8	90.3
Gray injection-molded part	93.5	88.5

Table 4. The experimental results of STI-Pose on the TR-RW dataset.

4.4 Silhouette stability experiments

Due to the utilization of silhouettes as input in STI-Pose, it is essential to investigate the stability of the algorithm with respect to silhouette extraction accuracy. In this paper, based on the YCB-V and LM-O datasets, we introduce various degrees of perturbations to the silhouette images segmented by SAM and evaluate the accuracy of STI-Pose on perturbed data. We employ the function $\delta = A\cos(x)$ as a random perturbation method. For each silhouette edge point, a random value x is chosen to calculate the perturbation value δ, and the point is displaced along the normal direction by δ to obtain the new silhouette image. In this perturbation data generation method, the amplitude A determines the perturbation magnitude. During actual generation, different values ranging from 0 to 2.5 are used to simulate various silhouette extraction accuracies. The visual effects of silhouettes under different perturbation amplitudes are illustrated in Fig. 10.

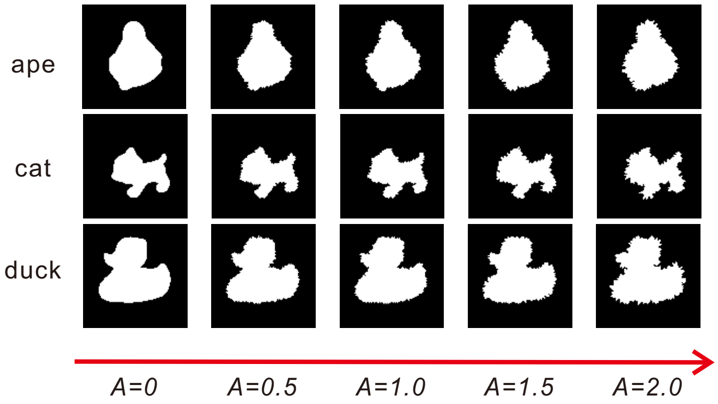


Fig. 10. Silhouette images under different perturbation amplitudes.

Datasets/A	0	0.25	0.5	1	1.5	2	2.5
YCB-V	1	1.06	1.14	1.23	1.33	1.57	1.85
LM-O	1	1.04	1.11	1.18	1.35	1.52	1.77

Table 5. The relative change in estimation error of STI-Pose as the silhouette perturbation amplitude varies.

Table 5 presents the relative changes in the ADD(-S) values of STI-Pose under both unperturbed and perturbed data at different amplitude levels. The data in the table represent the relative change rates of the ADD(-S) distance values. It can be observed that as the perturbation level increases, STI-Pose initially maintains stability until a significant decrease in accuracy occurs when the perturbation becomes excessive. This may be attributed to the calculation of Intersection over Union (IoU). The results indicate that the proposed method exhibits a certain tolerance to silhouette extraction accuracy, effectively addressing potential issues of inaccurate object silhouette extraction in practical applications.

4.5 Ablation Study on YCB-V

Exp.	Module Selection		Evaluation Metrics	
	W	O-PSO	ADD(-S)	AUC-ADD(-S)
1			63.7	73.2
2	✓		65.6	75.3
3		✓	85.6	90.4
4	✓	✓	**95.4**	**93.2**

Table 6. Ablation Study on YCB-V. We conducted ablation experiments on the weight map and O-PSO in STI-Pose, and the results are represented in % using ADD(-S) and AUC-ADD(-S).

We conducted ablation experiments on the YCB-V dataset to examine the effects of the weight map $\mathbf{W}$ and O-PSO. Specifically, the weight map $\mathbf{W}$ was used to calculate the IoU on a spherical surface, while the calculation was performed on a planar surface otherwise. In the absence of the proposed O-PSO algorithm, we employed a regular PSO algorithm with parameters aligned with those of O-PSO.

The experimental results, as depicted in Table 6, clearly demonstrate the significant performance improvement achieved with O-PSO. This improvement indicates its ability to assist STI-Pose in reliably locating the global maximum. Furthermore, the inclusion of the weight map $\mathbf{W}$ leads to further precision enhancement when using O-PSO. However, without O-PSO, the impact of the weight map is less pronounced. This is because, in scenarios near non-global maximum points, the improvement in HSIoU precision brought about by the weight map does not directly translate into improved pose estimation accuracy.

5 Conclusion and Outlook

In summary, we propose a silhouette-based 6D object pose estimation method, achieving high accuracy in the experiments. This method eliminates the need for annotated data, overcoming the limitations of deep learning-based pose estimation methods that can only handle several objects in the datasets. Furthermore, this method does not rely on object surface characteristics, exhibiting excellent generalization on objects with similar structures but different appearances, and demonstrated that achieving reasonably accurate object pose estimation is possible solely through silhouette information.

The use of silhouettes is the key factor in achieving generalization in our approach. However, relying solely on silhouettes comes with several limitations. STI-Pose requires precise and complete silhouette as input, and when an object is occluded, silhouettes may not be effectively extracted, making the method ineffective. In cases where silhouette ambiguity arises due to symmetry, STI-Pose can align silhouettes but may not provide correct pose values. Therefore, we plan to improve the method for assessing silhouette overlap in future work, enhancing the occlusion tolerance of HSIoU. Furthermore, while we have demonstrated that using only silhouette information can achieve satisfactory object pose estimation, completely disregarding texture information is not an optimal choice. We plan to consider object texture as an optional attribute and incorporate it into the HSIoU calculation. This integration aims to address potential silhouette ambiguity issues by leveraging texture information when needed.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Proceedings of the 9th European Conference on Computer Vision. vol. Part I, pp. 404–417 (2006)
2. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes. In: Proceedings of the International Society for Optical Engineering. vol. 14, pp. 239–256 (1992)
3. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6D object pose estimation using 3D object coordinates. In: Proceedings of the European Conference on Computer Vision. vol. Part II, pp. 536–551 (2014)
4. Busam, B., Esposito, M., Che'Rose, S., Navab, N., Frisch, B.: A stereo vision approach for cooperative robotic movement therapy. In: Proceedings of the IEEE International Conference on Computer Vision workshops. pp. 127–135 (2015)
5. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: The ycb object and model set: Towards common benchmarks for manipulation research. In: Proceedings of thr IEEE International Conference on Advanced Robotics. pp. 510–517 (2015)
6. Di, Y., Manhardt, F., Wang, G., Ji, X., Navab, N., Tombari, F.: SO-Pose: Exploiting self-occlusion for direct 6D pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12396–12405 (2021)
7. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3D object recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 998–1005 (2010)
8. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the IEEE International Symposium on Micro Machine and Human Science. pp. 39–43 (1995)
9. Ghazaei, G., Laina, I., Rupprecht, C., Tombari, F., Navab, N., Nazarpour, K.: Dealing with ambiguity in robotic grasping via multiple predictions. In: Proceedings of the Asian Conference on Computer Vision. pp. 38–55 (2019)
10. Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 858–865 (2011)

11. Hodan, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision. pp. 880–888. IEEE (2017)
12. Hu, Y., Fua, P., Wang, W., Salzmann, M.: Single-stage 6D object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2930–2939 (2020)
13. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2938–2946 (2015)
14. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment Anything. arXiv:2304.02643 (2023)
15. Lepetit, V., Moreno-Noguer, F., Fua, P.: Ep*n*p: An accurate o (n) solution to the p*n*p problem. International Journal of Computer Vision **81**, 155–166 (2009)
16. Li, Z., Wang, G., Ji, X.: CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7678–7687 (2019)
17. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE International Conference on Computer Vision. vol. 2, pp. 1150–1157. Ieee (1999)
18. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**, 91–110 (2004)
19. Marchand, E., Uchiyama, H., Spindler, F.: Pose estimation for augmented reality: a hands-on survey. IEEE Transactions on Visualization and Computer Graphics **22**(12), 2633–2651 (2015)
20. Olson, C.F., Huttenlocher, D.P.: Automatic target recognition by matching oriented edge pixels. IEEE Transactions on Image Processing **6**(1), 103–113 (1997)
21. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: PVNet: Pixel-wise voting network for 6DoF pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4561–4570 (2019)
22. Pérez, L., Rodríguez, Í., Rodríguez, N., Usamentiaga, R., García, D.F.: Robot guidance using machine vision techniques in industrial environments: A comparative review. Sensors **16**(3), 335 (2016)
23. Rambach, J., Pagani, A., Schneider, M., Artemenko, O., Stricker, D.: 6DoF object tracking based on 3D scans for augmented reality remote live support. Computers **7**(1), 6 (2018)
24. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2564–2571 (2011)
25. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3384–3391 (2008)
26. Su, Y., Saleh, M., Fetzer, T., Rambach, J., Navab, N., Busam, B., Stricker, D., Tombari, F.: ZebraPose: Coarse to fine surface encoding for 6DoF object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6738–6748 (2022)
27. Suzuki, S., et al.: Topological structural analysis of digitized binary images by border following. Computer vision, Graphics, and Image processing **30**(1), 32–46 (1985)

28. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9627–9636 (2019)
29. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: DenseFusion: 6D object pose estimation by iterative dense fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3343–3352 (2019)
30. Wang, G., Manhardt, F., Tombari, F., Ji, X.: GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16611–16621 (2021)
31. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In: Robotics: Science and Systems XIV (2018)
32. Zhang, X., Jiang, Z., Zhang, H., Wei, Q.: Vision-based pose estimation for textureless space objects by contour points matching. IEEE Transactions on Aerospace and Electronic Systems **54**(5), 2342–2355 (2018)
33. Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Gao, J., Lee, Y.J.: Segment everything everywhere all at once. arXiv preprint arXiv:2304.06718 (2023)