STAG: Smart Tools and Applications in Graphics (2025)
M. Comino Trinidad, C. Mancinelli, D. Cabiddu, and D. Giorgi (Editors)

Geometric aware local optimization for robust primitive fitting

A. Ferraris[1] and F. Leveni[1] D. Baieri[2] F. Maggioli[3] S. Melzi[2] L. Magri[1]

[1]Politecnico di Milano Piazza Leonardo da Vinci, 32, Milano, Italia
[2]Università degli Studi di Milano-Bicocca, Piazza dell'Ateneo Nuovo, 1, Milano, Italia
[3]Pegaso University, Centro Direzionale - Isola F2, Napoli

Abstract

The decomposition of 3D point clouds into meaningful geometric primitives is a longstanding challenge in Computer Vision and Computer Graphics. While recent advances in data-driven methods and neural representations have achieved significant progress in 3D reconstruction and abstraction, traditional primitive-based representations remain invaluable for tasks requiring interpretability, compactness, and robustness. This work introduces a novel framework for primitive decomposition in 2D and 3D point clouds, designed to cope with noise, outliers, and overlapping structures. Building upon traditional RANSAC-based approaches, the proposed method integrates geometric priors *to enhance its effectiveness in identifying interpretable and meaningful geometric primitives within complex data. Central to our approach is a novel geometric-aware* inlier refinement *step, which incorporates geometric constraints such as* surface completeness and normal consistency. *This refinement step is formulated as an optimization problem solved through the* GRAPH-CUT *algorithm. This optimization process penalizes excessive surface extensions and promotes coherence in normal orientations, ensuring that the refined inlier sets closely match the geometric structures the point cloud represents. Experiments on synthetic and real-world datasets validate the robustness and accuracy of the proposed method, demonstrating its ability to outperform state-of-the-art techniques in terms of both result quality and computational efficiency.*

CCS Concepts

• ***Computing methodologies*** → ***Computer vision;*** *Shape representations;*

1. Introduction

The task of *fitting geometric models to 2D and 3D data* to derive structured interpretations of unorganized spatial points is a key challenge in Computer Vision. Geometric model fitting algorithms aim to represent scenes using simple geometric primitives such as lines, conics, planes, cylinders, or superquadrics, providing a compact and interpretable approximation of complex structures. In 3D, this process is crucial for applications like scene understanding and robotics, where accurate shape modelling underpins tasks such as object grasping [WLL*25], collision avoidance [DKC25], and navigation [ZYZ*23]. In these scenarios, it is essential to balance accuracy and compactness by decomposing scenes into small 3D primitives. Primitive decomposition captures essential geometric structures while avoiding the high memory costs associated with voxel grids, point clouds, or meshes. However, the task is challenging: not only must the method be robust to noise and outliers, which are almost unavoidable in real data, but the simultaneous presence of multiple primitives requires estimating both the segmentation and the model parameters jointly, significantly increasing the challenge.

One of the most popular approaches for robust model fitting is *consensus maximization*, a framework well-represented by RAndom SAmpling Consensus (RANSAC) [FB81] and its numerous variants. RANSAC aims to maximize the number of inliers by iteratively sampling random subsets of the data to identify models with the highest level of consensus among points. This strategy has proven highly effective for single-model fitting, especially in scenarios with significant noise or outliers, making it a cornerstone technique in several vision applications.

Recent advancements, such as Lo-RANSAC [CMK03] and GC-RANSAC [BM22], have improved the original RANSAC framework by integrating global optimization strategies, achieving remarkable performance in image matching tasks for 3D reconstruction. GC-RANSAC, in particular, was designed with a strong focus on two-view geometry, where the objective is to estimate transformations between different images of the same scene. While highly effective in this domain, GC-RANSAC does not directly tailor the challenges of geometric primitive decomposition to scenes that often involve complex structures and diverse orientations. In this work, we aim to bridge this gap by adapting the strengths of these advanced robust fitting frameworks to the primitive fitting domain, bringing their benefits to the task of primitive decomposition.

We carried out primitive decomposition by fitting *superquadrics*. This task is particularly challenging because it violates some of the core assumptions of RANSAC. Unlike simpler primitives, a

DOI: 10.2312/stag.20251321

superquadric cannot be reliably estimated from a minimal sample of points: its parameters must be retrieved through a non-linear optimization problem that does not admit a closed-form solution. We first obtain an initial estimate from a randomly sampled, non-minimal subset of points to address this problem. Building on this initialization, we reformulate the problem, similarly to GC-RANSAC, as a labeling task, classifying points as inliers or outliers. An energy minimization procedure based on graph-cut solves this labeling. Crucially, our energy function integrates geometric priors, emphasizing normal orientation consistency, encouraging neighboring inliers to share coherent surface orientations. This strategy forms the basis of our *inlier refinement step*, which stabilizes the superquadric parameters and significantly reduces the number of iterations required by the RANSAC loop. Finally, we embed this refinement into a novel robust fitting algorithm (dubbed GAIR-RANSAC) that might operate sequentially, enabling effective primitive decomposition in complex 3D scenes.

Contributions: Our main contributions are

- We introduce a novel energy function incorporating 3D geometric priors specifically designed for primitive decomposition. In particular, we define a smoothness term that enforces normal-vector consistency among neighboring points.
- We propose a new *inlier refinement* step, formulated as a labeling problem and solved via energy minimization with GRAPH-CUT, which stabilizes parameter estimation and reduces the number of iterations required by RANSAC.
- We integrate the proposed *inlier refinement* into GAIR-RANSAC, a novel robust fitting framework that extends RANSAC to 2D-3D primitive decomposition.

Our contributions are primarily algorithmic, focusing on formulating the energy function at the method's core. We validate the approach on preliminary experiments with synthetic 2D and 3D data, deliberately restricting our analysis to relatively straightforward scenarios. Although these results already highlight the advantages of our strategy, this study should be regarded as a starting point towards tackling more challenging tasks, such as superquadric decomposition in real-world datasets. Interestingly, the benefits of our method are not limited to superquadrics: we observe improvements also on simpler primitives estimated in closed form, such as circles in 2D and spheres in 3D. This confirms the importance of exploiting the complete geometric information available in the data, including surface normals.

2. Related Work

We review two closely related research directions. On one side, our work addresses the general problem of robust model estimation, which is traditionally studied within the RANSAC paradigm and its numerous extensions. On the other hand, it tackles the task of geometric primitive fitting, where the goal is to approximate complex shapes with parametric structures such as planes, cylinders, or superquadrics.

Robust model estimation In the 1980s, the RANSAC paradigm [FB81] was introduced to match correspondences between stereo images. Since then, RANSAC has become a cornerstone in the estimation of a single parametric model and continues to be a key area of research today [BM22, Bar25]. RANSAC is the de facto standard in several 2D and 3D applications, such as plane detection for autonomous driving, geometric primitives detection in images, and 3D reconstruction. The RANSAC framework is intuitive and straightforward: iteratively, random models are extracted from the input and evaluated according to a quality measurement, called *consensus*. The final output is the model that achieves the best quality. Over the years, several advancements have been proposed to improve the standard RANSAC (see [Bar25] for a recent survey). The LO-RANSAC [CMK03] employs a Local Optimization routine that feeds the inlier points, recognized by the so-far best model, as input sampling pools to an inner RANSAC loop. Another notable extension is GC-RANSAC [BM22], which integrates Graph Cut [BVZ01] optimization to enhance model fitting by leveraging a *locality prior* to refine the models with their neighboring inliers. In GC-RANSAC, model fitting is approached as a *labelling problem* where each data point is classified either as an inlier or an outlier, based on its fitness to the candidate models. The energy function used in GC-RANSAC consists of two terms: a *data fidelity term* E_1 that measures how well points fit the model, and a *smoothness term* E_2 that encourages spatial coherence by promoting the same label for points that are spatially close to each other and near the model, not necessarily inside the fixed threshold. The research on robust fitting primarily focuses on introducing alternative sampling strategies and consensus functions to improve the quality of the retrieved model or to accelerate the computation by decreasing the total number of iterations. However, improvements are usually application-dependent. The feature-matching community drives most of these developments; thus, they develop the majority of robust fitting methods with 2D image-based problems in mind and therefore do not address the challenges of primitive decomposition. In particular, they overlook the geometric consistency inherent to the problem, which we explicitly exploit in our formulation to improve robustness and accuracy.

Robust model estimation must often infer more than one model from the data, and multi-model fitting approaches extend the basic RANSAC framework to handle multiple models. Several techniques have been proposed for multi-model fitting, ranging from clustering-based solutions (*e.g.*, [CNJB03, TF08, MF14, MLB21]) to optimization-based methods (*e.g.*, PEARL [IB12], PROG-X [BM19], and MULTI-X [BM18]). These optimization-driven approaches typically extend the objective function used by RANSAC for single-model fitting by combining two components: a *data fidelity term*, which quantifies how well points fit the models, and a *regularization term* that encourages simpler, more parsimonious solutions.

Additional terms can further regularize the solution. For example, PEARL and its variants include a regularization term that encourages spatial proximity among inliers in the same model. In 3D data, adjacent primitives often touch or overlap, making segmentation especially challenging. Unfortunately, considering only the proximity of inliers can lead to suboptimal results. An illustrative example is shown in Figure 1, where two spheres are close. When inliers are determined solely based on points' distance from the

model and inlier proximity (Figure 1b), points from the neighboring sphere are incorrectly classified as inliers (points depicted in red), leading to an erroneous model. By contrast, the inliers adhere correctly to the target sphere when additional geometric information is also considered, as in our GAIR-RANSAC depicted in Figure 1c).

Primitive fitting Primitive fitting can be considered a specific case of model fitting, where the goal is to fit geometric primitives like planes, cylinders, and parametric surfaces to point cloud data to describe complex objects as a collection of primitive structures. Superquadrics, in particular, are a class of primitives that gained attention for their ability to represent a wide range of shapes with a relatively small number of parameters. Early works introduced the use of superquadrics as a geometric primitive to describe objects in range images employing a least-square solution [SB90], as well as different error of fit measures on single superquadric recovery [GB88]. Notably, in [SWK07], an efficient variant of RANSAC designed for geometric primitive detection exploits surface normals. In their approach, normals are used for inlier validation by enforcing both a distance threshold and an angular deviation threshold between the point normal and the model normal. In contrast, our GAIR-RANSAC integrates normal information directly into the fitting energy through pairwise consistency terms, allowing normals to influence the optimization process rather than being used solely for validation. Later, Solina *et al.* [LSM94] proposed a method for estimating superquadrics through two main steps: firstly, they fit the model via an iterative refinement that gradually adds nearby points, and then the best model is selected according to an objective function, before applying a further refinement.

Liu et al. [LWRC21] proposed a probabilistic framework for superquadric recovery using Expectation Maximization (EM), where they model outliers and noise in the data to enhance the robustness of the fitting process. Monnier *et al.* [MAK*23] proposed Differentiable Blocks World, an approach for scene decomposition and 3D reconstruction that directly optimizes textured superquadric primitives from calibrated multi-view images using differentiable rendering. Although this work balances reconstruction fidelity and model simplicity, the optimization process is computationally heavy and impractical for real-time and interactive applications. Ramamonjisoa *et al.* [RSL22] proposed another promising direction with their stochastic search method. The authors propose an algorithm for fitting primitives to noisy 3D point clouds, robustly handling missing data and noise without requiring training data. However, the method's reliance on pre-defined primitives, such as cuboids, limits its flexibility in representing complex geometries.

Recently, learning-based approaches have been gaining momentum, introducing a different point of view in geometric primitives fitting. A particularly successful research line proposes data-driven approaches that learn to infer a collection of 3D primitives (such as superquadrics [PUG19, PvGG20] or cuboids [TSG*17]) from classes of clean 3D data. While promising, these approaches still require extensive collections of clean data for training. They are challenging to make more general, as they are typically bound to a specific category of scenes or primitives.

3. Method

The proposed method, Geometric Aware Inlier Refinement GAIR, is an Inlier Refinement step enhanced by the geometric local coherency given by the normals of neighboring points. We implement it in the form of GAIR-RANSAC, a GC-RANSAC [BM22] approach that employs a Local Optimization routine formulated as a labeling problem, solved using the Graph-Cut [BVZ01] algorithm. When the outer RANSAC finds a new best model, it triggers the Local Optimization, which consists of a loop where an inner RANSAC optimizes a new model with the points the first model recognizes as inliers. The inner RANSAC will optimize the model found, and the loop will terminate when the optimization converges. Our contribution adapts [BM22] to the complex problem of decomposition of 3D point clouds by employing a new *geometric smoothness term*. Still, it can be leveraged to be used as an Inlier Refinement step for the majority of the multi-model algorithms. Section 3.2 introduces the GAIR-RANSAC approach used, while Section 3.2.1 explains the GAIR algorithm. Our approach to building the method is technical: we choose the well-noted RANSAC method to highlight the benefits gained by the proposed GAIR algorithm and compare it with different RANSAC versions.

3.1. Problem formulation and background

The assumption of Primitive Decomposition is to have a scene with objects, represented by an unordered set of noisy points $\mathcal{D} = \{x_1, \ldots, x_n\}$, with $x_i \in \mathbb{R}^d$. The goal is to find geometric primitives that best fit the original objects in the point cloud, represented as a collection of geometric primitives $\{\theta_1, \ldots, \theta_\kappa\}$. We also assume the scene can be described as a combination of heterogeneous primitives (e.g., superquadrics, planes, circles, etc.). To employ the Graph Cut [BVZ01] algorithm, we need to build the graph $\mathcal{G}$ from the input point cloud. The graph $\mathcal{G}$ identifies, for each point x_i, the set of neighbour points $\mathcal{N}_i$. We implement the graph construction by finding, for each point x_i, the first m neighboring points contained in a sphere of radius r set in function of the point cloud spatial extension. For each point x_i, the corresponding normal vector $n_i \in \mathbb{R}^d$ collected in matrix $\mathcal{V}$ is either provided or can be estimated.

To describe our input data points, we choose the superquadric model that was first introduced in [Bar81] for applications in the field of computer-aided design. The superquadrics are a family of surfaces derived from the quadric surfaces, and are obtained by raising each trigonometric term to an exponent that rules the features of the surface (*i.e.*, how round or sharp the surface is in each direction). Formally, we can define a canonical superquadric with its implicit formulation:

$$F(\mathbf{x}_c, \Lambda) = \left(\left(\frac{x_c}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y_c}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z_c}{a_3} \right)^{\frac{2}{\varepsilon_1}} = 1 . \quad (1)$$

The canonical parameters are $\Lambda = \{a_1, a_2, a_3, \varepsilon_1, \varepsilon_2\}$; a_1, a_2, a_3 are the size parameters, corresponding to the scale along the x, y, and z axis, respectively. ε_1 and ϵ_2 are the shape parameters; ε_1 governs the shape along the z-axis, called the principal axis, while ε_2 governs the shape orthogonal to the z-axis (*i.e.*, the x and y axes). This means that, by tuning ε_1 and ε_2, we can determine the features of

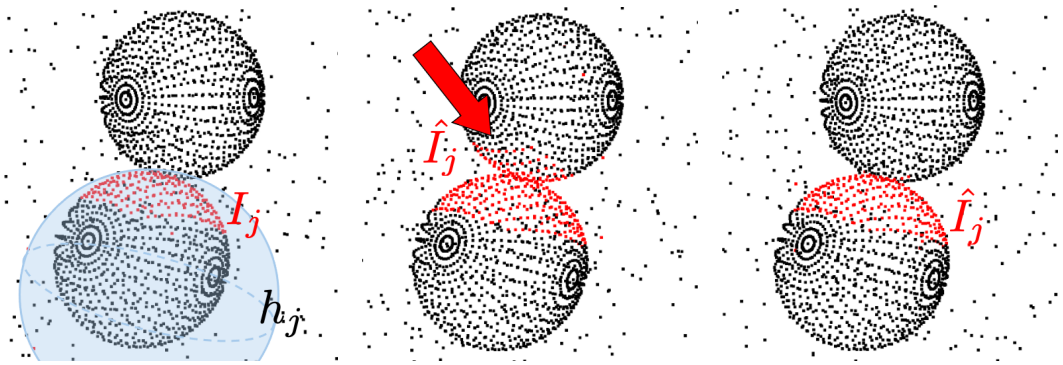


Figure 1: *The images illustrate the different expansion behaviours of GC-RANSAC's spatial coherency (b) and our GAIR-RANSAC (c) method when applied to 3D superquadric spheres. Both algorithms process the same input point cloud and utilize identical points for model fitting. On the left, the expansion produced by GC-RANSAC shows that the inliers are detected even on the other sphere. On the right, our inlier refinement demonstrates how it avoids propagating onto the adjacent sphere.*

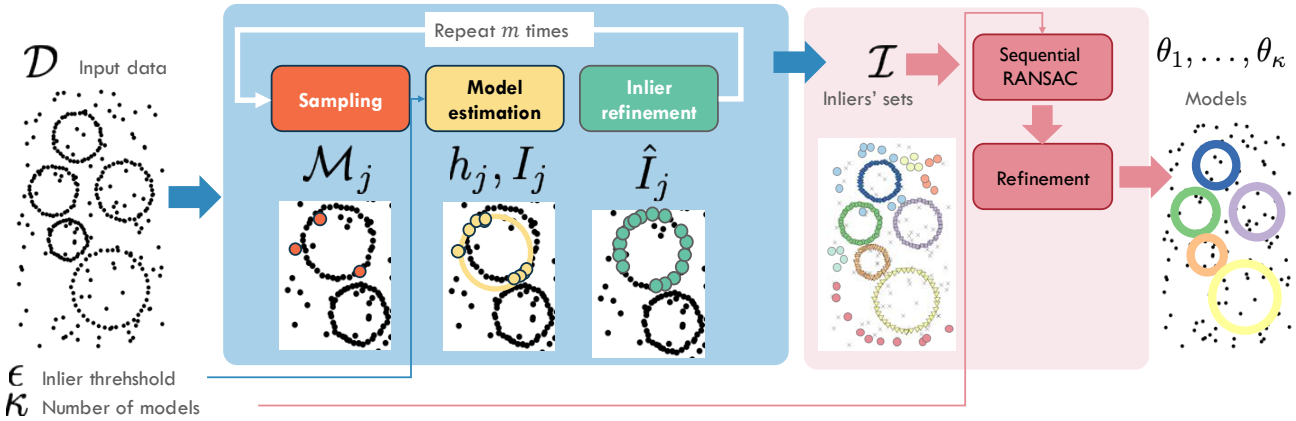


Figure 2: *Our pipeline in a nutshell: the core of our method is our inlier refinement step that, given a model h_j and its inlier I_j (yellow dots), provides a refined set of inliers $\hat{I}_j$ (green dots).*

the surface along the corresponding axes. Those parameters represent the real power of the superquadric, allowing them to describe a wide range of shapes, spanning from spheres and ellipses to cubes and cylinders. A linear transformation $\mathbf{T}$ parametrizes the position and orientation of the geometric model. It controls the rotation and transition from the canonical form to the actual superquadric in the 3D scene. The transformation $\mathbf{T}$ in controlled by the parameter $\Gamma = \{\alpha, \beta, \gamma, p_x, p_y, p_z\}$. Each model θ retrieved as output will be represented by the union of these two sets $\theta = \Lambda \cup \Gamma$.

3.2. GAIR-RANSAC

Figure 2 illustrates our method, and the first phase is outlined in Algorithm 1. The procedure iteratively samples, estimates, and refines a pool of m tentative models, selecting the one that best explains the data. At each iteration j, a random subset $\mathcal{M}_j \subset \mathcal{D}$ is extracted (line 4) and a model h_j is fitted to it (line 5). When the fitting is performed in closed form (e.g., for spheres or planes), $\mathcal{M}_j$ is a minimal sample set as in RANSAC, i.e., the minimum number of points required to instantiate the model (for example, three points for a 2D circle or a plane). For superquadrics, however, parameter estimation requires solving a non-linear optimization problem. Since a minimal set of points is not sufficient to produce a stable initialization, we adopt a bucketing strategy: the dataset $\mathcal{D}$ is partitioned into local patches, from which we randomly select one patch and two points to form $\mathcal{M}_j$. The final MSS (Minimal Subset Selection) is then composed by sampling at least two points from each patch, following the idea of [SWK07] that the probability that two points belong to the same shape is higher the smaller the distance between the points. After estimating the hypothesis h_j, we refine its inlier set (lines 7–16). This refinement step is triggered whenever the number of inliers of the current model exceeds that of the best-so-far model, as detailed in the following section. Finally, when multiple primitives are sought, the procedure is applied sequentially: after a model h^* has been selected, its inliers $\mathcal{I}^*$ are removed from $\mathcal{D}$, and the algorithm is re-run on the remaining data

until no significant structure is left. This sequential strategy allows us to decompose the scene into multiple geometric primitives progressively.

Algorithm 1 GAIR-RANSAC

Input: point cloud $\mathcal{D}$, threshold distance ϵ,
maximum number of models κ,
maximum number of iterations m
Output: model parameters θ^*

$\mathcal{I}^* = \emptyset, h^* = \emptyset$
$\mathcal{G} = initGraph(\mathcal{D}), \mathcal{V} = initNormals(\mathcal{D})$
for $j = 1 \rightarrow m$ **do**
  $\mathcal{M}_j = estimateMSS(\mathcal{D})$
  $h_j = fitGeometricModel(\mathcal{D}, \mathcal{M}_j)$
  $\mathcal{I}_j = computeConsensus(h_j, \epsilon)$
  if $|\mathcal{I}_j| > |\mathcal{I}^*|$ **then**
    $terminate =$ **False**
    $\mathcal{I} = \mathcal{I}_j, h = h_j$
    while not $terminate$ **do**
      $\widehat{\mathcal{I}} =$ **GAIR**$(\mathcal{D}, \mathcal{G}, \mathcal{V}, h)$
      $\widehat{h}, \widehat{c} = innerRANSAC(\mathcal{D}, \widehat{\mathcal{I}}, \mathcal{I}, \epsilon)$
      if $compareConsensus(\mathcal{I}, \widehat{\mathcal{I}})$ **then**
        $\mathcal{I} = \widehat{\mathcal{I}}, h = \widehat{h}$
      else
        $terminate =$ **True**
    $\mathcal{I}^* = \mathcal{I}, h^* = h$
return h^*

3.2.1. Geometric Aware Inlier Refinement

The purpose of the *Inlier Refinement* step is to improve the quality of the candidate models and their corresponding inlier sets, generated in the *inner RANSAC* phase. This step is essential to make the estimated models more accurate and consistent with the underlying data structures, especially in the presence of noise and outliers.

At a high level, each inlier set $\mathcal{I}_j$ associated with a model h_j can be interpreted as a *labeling* f_j of the input point cloud $\mathcal{D}$. The labeling function $f_j \colon \mathcal{D} \rightarrow \{0,1\}$ assigns each point $p \in \mathcal{D}$ a binary label, where $f_j(p) = 1$ indicates that p is an inlier of the model h_j, and $f_j(p) = 0$ otherwise. In formulas:

$$f_j(p) = \begin{cases} 1, & \text{if } \mathrm{err}(p, h_j) < \epsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Given the initial labeling f_j derived from the residual thresholding criterion, our goal is to compute a refined labeling $\hat{f}_j$ that defines an updated inlier set:

$$\widehat{\mathcal{I}}_j = \{x \in \mathcal{D} \colon \hat{f}_j(x) = 1\}. \tag{3}$$

This refinement leverages additional geometric and spatial priors that are particularly relevant in primitive decomposition tasks. These priors ensure a better alignment of inliers with the underlying primitive and include:

- **Compactness of the Estimated Primitive:** To improve model quality, we encourage the selection of primitives whose surfaces are densely covered by inliers. This criterion discourages fitting primitives that extend significantly beyond the region occupied by the point cloud $\mathcal{D}$. The aim is to ensure that the primitive effectively explains the points within its region of influence while avoiding unused areas on the primitive surface. This approach promotes the identification of tighter, more representative models that closely align with the data distribution.
- **Normal Consistency:** Beyond spatial proximity, inliers are expected to exhibit smooth variations in their normal vectors. This assumption reflects the inherent smoothness of most geometric primitives, where neighboring points typically share similar orientations. Inliers close to each other should therefore have normal vectors with small angular differences. This constraint ensures that the inlier set not only follows the spatial structure of the primitive but also aligns with its geometric properties, enhancing robustness and accuracy.

The refined inlier set $\hat{\mathcal{I}}_j$ is computed by updating the labeling f_j to $\hat{f}_j$ through an energy minimization problem inspired by GC-RANSAC. Specifically, to incorporate the two priors detailed above, our formulation relies on a key component, the **angular distances between normals:** neighboring points with well-aligned normal vectors are encouraged to share the same label. This promotes both spatial coherence and geometric consistency, helping to refine the inlier set to match the smooth structure of the primitive better. The labeling problem is efficiently solved using the *Graph-Cut* algorithm, which provides an optimal and computationally efficient solution for the energy minimization problem. The details of this process, including the energy formulation and the implementation of the *Graph-Cut* algorithm, are described in the following sections.

3.2.2. Energy Formulation

The inlier refinement is formulated as an energy minimization problem defined over a labeling function. For clarity and conciseness, we omit the subscript f_j and consider a general labeling function f. The energy function $E(f)$ quantifies the quality of a labeling configuration and consists of two terms:

$$E(f) = \sum_{p \in \mathcal{D}} E_1(f_p) + \sum_{(p,q) \in \mathcal{N}_p} E_2(f_p, f_q), \tag{4}$$

where:

- E_1 is the *unary term* (data fidelity term) that evaluates how well individual points fit the model.
- E_2 is the *pairwise term* (smoothness term) promoting coherence between neighboring points in the graph $\mathcal{G}$, where the adjacency relationships in the point cloud are represented by the sets N_i.

In traditional RANSAC formulations, only the first term of E_1 is used, promoting the inclusion of as many inliers as possible based on a residual threshold. GC-RANSAC [BM22] extend this formulation by including the *smoothness term* to refine the quality of the inlier set. Our extension to a *geometrical smoothness term* will be completely described in Section 3.2.2.

Unary Term: Data Fidelity The unary term E_1 measures how well the initial binary labelling f given by the thresholding of the residuals fits the input point cloud. The Graph Cut formulation works on penalties, then E_1 assigns the maximum penalty to outlier

points. For inlier points, instead, E_1 assigns a penalty dependent on the computed residual between a point and the model, normalised not to exceed 1. We can formulate it as:

$$E_1(f_p) = \begin{cases} \text{err}(p,h_j), & \text{if } f_p = 1 \\ 1, & \text{if } f_p = 0. \end{cases} \tag{5}$$

Here, $\text{err}(p,h_j)$ represents the normalized residual distance between point p and the model h_j, and ϵ is the inlier threshold. Residuals exceeding ϵ are clamped to 1, ensuring a constant penalty for points far from the model. This thresholding reduces the influence of outliers, aligning with standard robust fitting practices. The minimization of E_1 is the typical task accomplished by the RANSAC algorithm that tries to maximize the number of inliers recognised by a model.

Pairwise Term: Smoothness and Regularity The pairwise term $E_2(f_p, f_q)$ promotes smoothness in the labeling by encouraging neighboring points with similar geometric properties to share the same label. It plays a crucial role in maintaining consistency in the inlier refinement process, particularly in the presence of noise or overlapping structures. We present the formulation of E_2 by incorporating step-by-step additional geometric and spatial constraints while ensuring the regularity conditions required for the Graph-Cut algorithm are met.

The pairwise term begins with a simple penalty based on the angular similarity between the normal vectors of neighboring points p and q. The coherency term $C(n_p, n_q)$ is defined as:

$$C(n_p, n_q) = \frac{1}{2}\left(1 + n_p \cdot n_q\right), \tag{6}$$

where n_p and n_q are the normalized normal vectors of points p and q. This term measures the angular similarity, with $C = 1$ indicating perfectly aligned normals and $C = 0$ indicating opposite normals. By bounding $C(n_p, n_q)$ within $[0,1]$, we ensure that the pairwise penalty remains well-behaved and does not violate submodularity. In the basic formulation, the pairwise term is expressed as:

$$E_2(f_p, f_q) = \begin{cases} C(n_p, n_q), & \text{if } f_p \neq f_q, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

This formulation encourages neighboring points with aligned normals to share the same label, promoting smooth labeling. From now on, we will omit the input normal vectors and call it C.

We further refine the pairwise term to enhance robustness by incorporating a thresholding mechanism for points with large residuals. For points p and q with normalized distances $\overline{d_p}$ and $\overline{d_q}$, the pairwise penalty is clamped when these distances exceed a threshold ϵ. This ensures that points far from the model or poorly aligned with it have a limited influence on the labeling. In conclusion, we can define two distinct pairwise terms as:

$$E_2(f_p, f_q) = \begin{cases} C, & \text{if } f_p \neq f_q, \\ 0, & \text{if } f_p = f_q = 1, \\ C\left(1 - \frac{\overline{d_p} + \overline{d_q}}{2}\right), & \text{if } f_p = f_q = 0. \end{cases} \tag{8}$$

that we will call GAIR. Here, $\overline{d_p}$ and $\overline{d_q}$ represent the normalized distances between points and the model, scaled to $(\frac{3}{2})^2\epsilon$. This normalization ensures the term remains bounded within $[0,1]$.

The pairwise term must satisfy submodularity for the Graph-Cut algorithm to guarantee a globally optimal solution. This requires the following inequality to hold:

$$C\left(1 - \frac{\overline{d_p} + \overline{d_q}}{2}\right) \leq 2C. \tag{9}$$

This condition is satisfied because the term $(1 - \frac{\overline{d_p} + \overline{d_q}}{2})$ is bounded within $[0,1]$, ensuring that the pairwise term remains well-behaved. In practical scenarios, the graph $\mathcal{G}$ may include edges that connect points across discontinuities or sharp corners in the point cloud. We introduce a heuristic to address this: if the coherency term $C(n_p, n_q)$ approaches zero, the corresponding edge weight is nullified. This effectively removes low-coherency edges from the graph, reducing the risk of labeling inconsistencies in regions with abrupt changes in geometry. All in all, our energy formulation combines a unary term that ensures individual points align with the model and a pairwise term that enforces smoothness in the labeling. The inclusion of thresholded residuals enhances robustness by reducing the influence of outliers. Furthermore, the pairwise term ensures that the refined inlier set $\hat{\mathcal{I}}_j$ aligns with the spatial and geometric properties of the underlying primitive. The next section describes how this energy is minimized efficiently using the Graph-Cut algorithm. The visualization example in Figure 1 illustrates the behaviour of our Geometric-Aware Inlier Refinement.

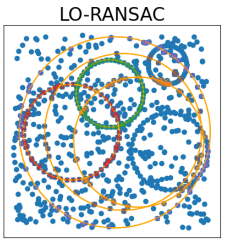
3.2.3. Min-Cut/Max-Flow Optimization

The Graph-Cut [BVZ01] algorithm minimizes the total energy $E(f)$ by solving a min-cut/max-flow problem on the graph G. The algorithm identifies the cut that separates the source from the sink with the minimum total cost, where the cost is the sum of the weights of the edges crossing the cut. The resulting partition of the graph defines the optimal labeling f, where:

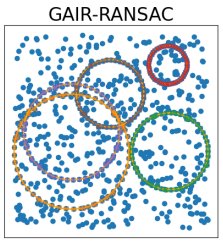
- Nodes connected to the source after the cut correspond to inliers ($f(p) = 1$);
- Nodes connected to the sink correspond to outliers ($f(p) = 0$).

Computational Efficiency The computational efficiency of Graph Cut [BVZ01] is complex to estimate. [BK04] has produced an exhaustive report on the complexity of this method, but has not been able to provide a precise answer. What emerges is that, although in the pessimistic case the complexity is $\mathcal{O}(mn^2|C|)$, where m is the number of edges, n the number of nodes of the graph, and $|C|$ is the cost of the minimum path, this is very rare, especially in the field of Computer Vision. Moreover, the experiments made in [BK04] clearly show that the algorithm runs of magnitude faster in the Computer Vision task, given an empirical argumentation on the fast execution of Graph Cut on grid-like graphs.

The use of the Graph-Cut algorithm enables efficient and robust refinement of the inlier sets $\hat{\mathcal{I}}_j$. By incorporating unary terms (favouring individual point-to-model fits) and pairwise terms (enforcing smoothness and geometric consistency), the algorithm produces high-quality inlier sets that align with the underlying primitives. This refinement is crucial for improving the accuracy and robustness of the final model selection process.



(a) *Sequential LO-RANSAC.*



(b) *Sequential GAIR-RANSAC.*

Figure 3: *The impact of the inlier refinement on multi-model fitting tasks. A simple input point cloud consisting of 5 circles corrupted by outliers. The RANSAC optimization algorithm used in 3a fails in retrieving all models. Our solution, which employs the smoothness term E_2, correctly finds all the structures.*

4. Experiments

We evaluate our framework's effectiveness through preliminary experiments designed to highlight the benefits of GAIR-RANSAC in simple primitive decomposition tasks, both in 2D and 3D. We begin with a qualitative study on 2D circle fitting, and then move to a quantitative analysis on 3D primitive decomposition.

4.1. Circle fitting

We begin our evaluation with a simple circle fitting problem in 2D. Although simplified, this scenario is particularly suitable for illustrating the benefits of our inlier refinement strategy compared to LO-RANSAC, which does not exploit any geometric priors specific to primitive decomposition. Figure 3 shows the impact of inlier refinement on a multi-model fitting task. The input consists of a point cloud with five circles corrupted by outliers. When employing sequential LO-RANSAC with the standard consensus criterion—maximizing the number of inliers—the algorithm fails to recover all circles, missing two of them due to the presence of multiple overlapping structures and significant outlier noise (Figure 3a). In contrast, our method (Figure 3b), which integrates the smoothness term E_2 in the refinement step, correctly retrieves all the structures. This result highlights how incorporating normal consistency into the energy formulation allows us to better separate neighboring models and to maintain robustness in cluttered scenes.

4.2. 3D primitive deocomposition

To further assess the effectiveness of our method, we moved to the problem of primitive decomposition in 3D. For this purpose, we created a synthetic dataset *SqSoup* where point clouds are generated as a mixture of superquadrics, allowing us to control the number of primitives and the amount of noise and outliers in the data. We organize our analysis into two scenarios: single-model and multi-model.

4.2.1. 3D dataset: SqSoup

To evaluate our method, we introduce SQSOUP (Figure 4), a synthetic dataset designed for 3D primitive decomposition. Each point cloud is constructed as a soup of superquadrics, i.e., a mixture of multiple superquadric primitives, providing flexibility in shape generation and complete control over the experimental conditions. Crucially, the dataset is annotated with ground-truth labels identifying the primitive to which each point belongs, enabling the evaluation of segmentation accuracy through misclassification error.

Single-model configuration We generate a small set of 10 point clouds, each sampled from a single superquadric. This scenario is intended to isolate the fitting problem and assess the accuracy of the algorithms in the absence of segmentation. To increase difficulty, we augment these point clouds by injecting Gaussian noise with fixed standard deviation $\sigma = 0.2$ and by adding varying percentages of outlier points in the range $\{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$. This procedure yields a total of 80 test point clouds.

Multi-model configuration. To test primitive decomposition, we generate point clouds of multiple superquadrics arranged to represent simple objects, animals, or conceptual shapes, selected from publicly available repositories. The data are augmented with Gaussian perturbations ($\sigma = 0.1$) and with outlier injections at the same range of levels as in the single-model case.

The resulting dataset provides a controlled yet challenging benchmark for evaluating both the fitting accuracy of single primitives and the robustness of multi-model decomposition in noisy and cluttered scenarios.

4.2.2. Figures of Merit

We evaluate the performance of our method in terms of:

1. *Misclassification error*: measures the ability to correctly segment the data into groups corresponding to the κ underlying models.
2. *Execution time*: measures the time needed to run the algorithm.
3. *Convergence*: number of optimization steps occurred to find the best models.

4.3. Competitors

We compare our method against three well-established variants of the RANSAC paradigm:

Vanilla RANSAC [FB81] the original formulation, used here as a baseline to assess the benefits of more advanced strategies.

LO-RANSAC [CMK03] which introduces a local optimization step based on inlier refinement. The comparison with LO-RANSAC highlights the advantages of our formulation, which integrates explicit geometric priors into the refinement process.

GC-RANSAC [BM22] which frames inlier refinement as an energy minimization problem solved via Graph-Cut. By comparing against GC-RANSAC, we demonstrate the benefits of our novel energy function, tailored to 3D primitive decomposition.

For the multi-model case, we adopt a sequential strategy: all methods are embedded into a Sequential-RANSAC framework [VL01], where models are extracted one at a time and their inliers removed before reapplying the algorithm to the residual data. This procedure allows us to evaluate the robustness of each variant in primitive decomposition tasks.

To ensure a fair comparison, we configure all methods under the

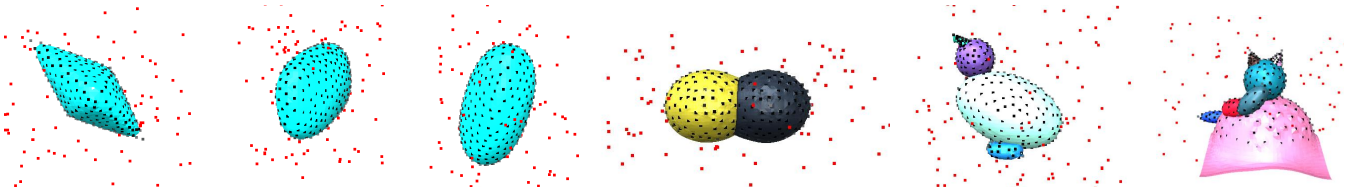
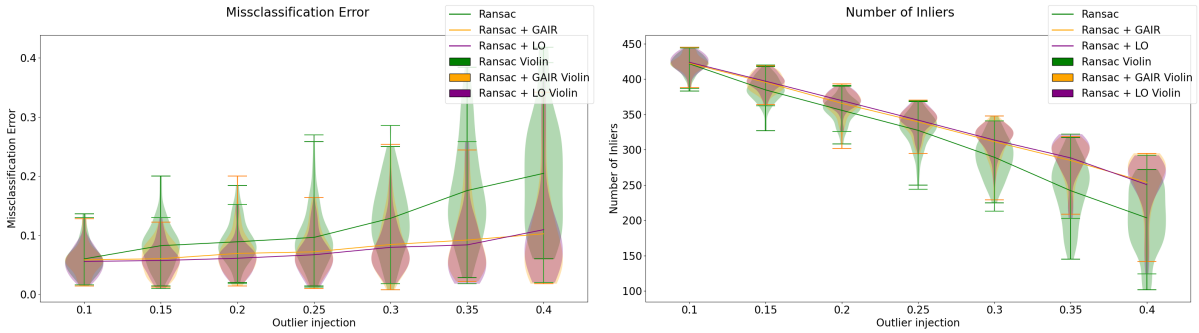


Figure 4: *Sample point clouds from the* SQSOUP *dataset.*



(a) *Misclassification error vs outlier (%)* **(b)** *Number of inliers vs outliers (%)*

Figure 5: *Results on SqSoup – single model.*

same conditions: The maximum number of iterations m is fixed to 500 for all algorithms. We also considered $m = 5000$ iterations. A larger budget would eventually allow all methods to saturate, thus hiding meaningful performance differences. The consensus function is the standard count of inliers. The inlier threshold ϵ is set as $\epsilon = 3 \cdot \sigma$, where σ is the amount of noise in the data following a common heuristic. The size of the sample set $\mathcal{M}_j$ is fixed to 20. Non-linear model fitting (for superquadrics) is performed using the `scipy` solver. For algorithms involving an inner RANSAC loop (LO-RANSAC and GAIR-RANSAC), we fix the number of inner iterations to 50. This setup allows us to isolate the contribution of the refinement strategies and proposed energy formulation, while keeping all other factors consistent across the baselines.

4.3.1. Results on SqSoup-single model

We evaluate the algorithms on a subset of SqSoup containing 10 point clouds, each representing a single superquadric. Each algorithm is run 20 times with $m = 500$ iterations, recording misclassification errors and the number of inliers for the best-fitting model. Figure 5 presents the results as violin plots across varying outlier ratios.

Vanilla RANSAC performs worst across all metrics. GC-RANSAC and GAIR-RANSAC achieve comparable misclassification errors, which is expected in single-model scenarios where the challenge of distinguishing between adjacent overlapping primitives does not arise. However, the limited iteration budget ($m = 500$) hinders all methods as outlier ratios increase, preventing convergence to accurate superquadric fits. Increasing the iteration count addresses this limitation.

4.3.2. Results on SqSoup-multi model

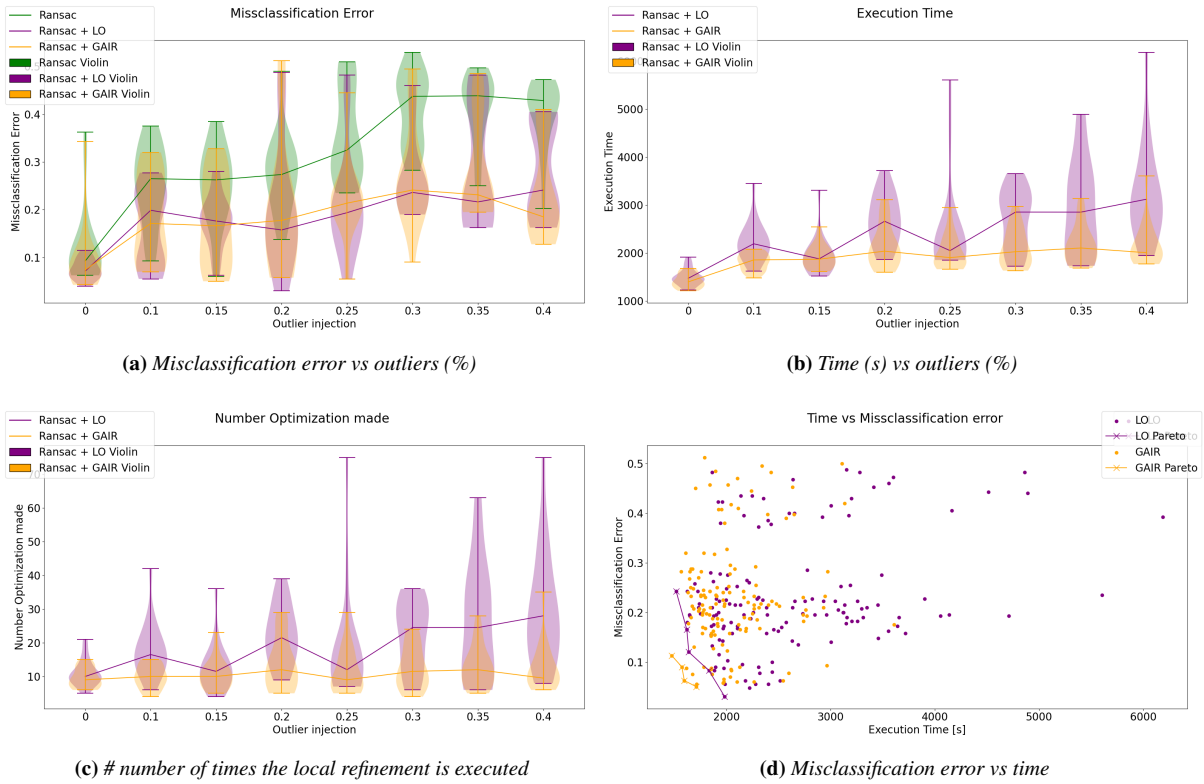
We repeat the experiment on the multi-model portion of SqSoup, where each scene contains multiple superquadrics. All algorithms operate sequentially, removing the inliers of each fitted superquadric until κ models are extracted. We perform 20 repetitions per dataset with $m = 500$ iterations for each sequential step.

Figure 6 summarizes the results across three metrics: misclassification error (Figure 6a), execution time (Figure 6b), and the number of local optimization steps performed by LO-RANSAC and GAIR-RANSAC (Figure 6c).

Vanilla RANSAC performs worst, confirming that minimizing the data fidelity term E_1 alone is insufficient for superquadric decomposition. While GAIR-RANSAC and LO-RANSAC achieve similar misclassification errors, LO-RANSAC requires approximately twice as many optimization steps to converge, resulting in longer execution times (Figure 6b). This demonstrates the effectiveness of GAIR's inlier refinement: although each refinement step is more computationally expensive than LO-RANSAC's approach, it produces higher-quality models that require fewer iterations overall, reducing total computational cost.

This advantage is further illustrated in Figure 6d, which plots misclassification error against execution time for all runs. GAIR-RANSAC exhibits more consistent execution times and Pareto-dominates LO-RANSAC, striking a better tradeoff between accuracy and computational efficiency.

We evaluate LO-RANSAC and GAIR-RANSAC with an increased iteration budget of $m = 5000$ to provide a more complete analysis. As shown in Figure 7, both methods achieve lower misclassification errors, confirming that additional itera-



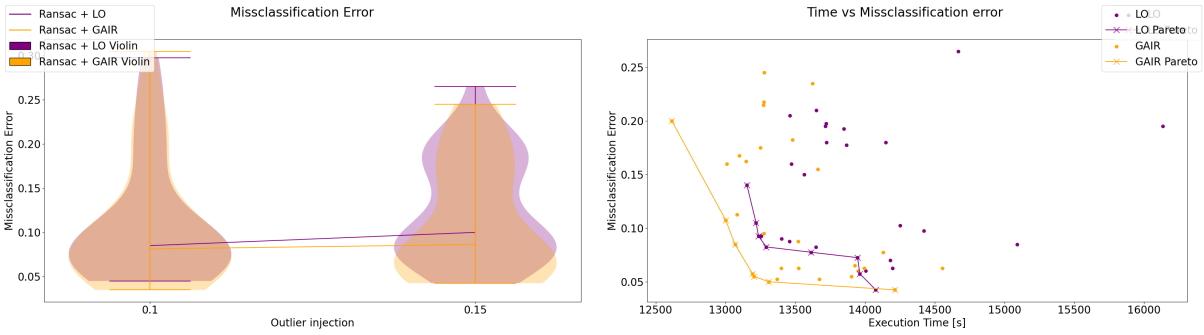
(a) *Misclassification error vs outliers (%)*

(b) *Time (s) vs outliers (%)*

(c) *# number of times the local refinement is executed*

(d) *Misclassification error vs time*

Figure 6: *Results on Sq-soup – multi-model.*



(a)

(b)

Figure 7: *Results of the supplementary experiments on the synthetic multi-model dataset with 5000 iterations.*

tions improve performance. However, GAIR-RANSAC continues to Pareto-dominate LO-RANSAC, as evidenced by the Pareto frontier in Figure 7b.

Furthermore, the scatter plot reveals that GAIR-RANSAC exhibits more consistent execution times across runs. This stability further demonstrates the advantage of incorporating geometric priors into the refinement process: while individual refinement steps are more expensive, they consistently produce higher-quality models that require fewer overall iterations, resulting in both better accuracy and more predictable runtime.

5. Conclusions and Future Work

This work presents an algorithmic contribution to the problem of primitive decomposition in 2D and 3D point clouds. At its core lies a novel energy formulation that extends the GC-RANSAC framework by incorporating geometric priors specifically tailored for primitive fitting. While we focus primarily on normal consistency

in this work, the proposed formulation is general and provides a sound foundation for integrating additional geometric constraints.

Our preliminary experiments on synthetic data demonstrate the effectiveness of the approach. In particular, the geometric-aware inlier refinement successfully mitigates some inherent drawbacks of sequential fitting strategies, as illustrated in the circle fitting experiment. Although Graph-Cut optimization is computationally more expensive per iteration than the refinement step in LO-RANSAC, our experiments on superquadric decomposition reveal a favourable tradeoff: the higher cost of individual refinement steps is offset by the improved model quality, which significantly reduces the total number of optimization iterations required. This results in both better accuracy and more consistent execution times overall.

Several promising directions remain open for future investigation:

Additional geometric priors. The energy formulation can be naturally extended to incorporate other geometric constraints. An exciting direction would be to integrate a term based on Chamfer distance, which could further regularize the solution by promoting the selection of compact superquadrics that provide better coverage of the point data while discouraging excessive surface extensions into empty regions.

Validation on real-world data. While synthetic experiments provide controlled conditions for evaluating algorithmic performance, extensive validation on real-world datasets is essential. Superquadric models often provide only local approximations of complex shapes in practical scenarios. We anticipate that the choice of the inlier threshold ϵ will become critical in such settings, and adaptive strategies for setting this parameter may be necessary.

Integration with global multi-model fitting. An important direction is to explore how the proposed inlier refinement can be embedded within more sophisticated multi-model fitting frameworks. Rather than extracting primitives sequentially through a greedy procedure, a promising approach would reformulate the problem as a global multi-label optimization, where each label corresponds to a different model (plus one for outliers). This would reduce the dependency on the order of model extraction and potentially lead to more robust decompositions in complex scenes.

Automatic model complexity selection. Finally, integrating a model complexity term directly into the RANSAC loop could automatically determine the number of primitives κ, moving beyond the current requirement of specifying it in advance. This would bring the method closer to a fully automatic primitive decomposition pipeline.

References

[Bar81] BARR: Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications 1*, 1 (1981), 11–23. doi:10.1109/MCG.1981.1673799. 3

[Bar25] BARATH D.: Superansac: One ransac to rule them all, 2025. URL: https://arxiv.org/abs/2506.04803, arXiv:2506.04803. 2

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 9 (2004), 1124–1137. doi:10.1109/TPAMI.2004.60. 6

[BM18] BARATH D., MATAS J.: Multi-class model fitting by energy minimization and mode-seeking. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI* (Berlin, Heidelberg, 2018), Springer-Verlag, p. 229–245. URL: https://doi.org/10.1007/978-3-030-01270-0_14, doi:10.1007/978-3-030-01270-0_14. 2

[BM19] BARÁTH D., MATAS J.: Progressive-x: Efficient, anytime, multi-model fitting algorithm. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), 3779–3787. URL: https://api.semanticscholar.org/CorpusID:174802993. 2

[BM22] BARATH D., MATAS J.: Graph-cut ransac: Local optimization on spatially coherent structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence 44*, 9 (2022), 4961–4974. doi:10.1109/TPAMI.2021.3071812. 1, 2, 3, 5, 7

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (2001), 1222–1239. doi:10.1109/34.969114. 2, 3, 6

[CMK03] CHUM O., MATAS J., KITTLER J.: Locally optimized ransac. In *Pattern Recognition* (Berlin, Heidelberg, 2003), Michaelis B., Krell G., (Eds.), Springer Berlin Heidelberg, pp. 236–243. 1, 2, 7

[CNJB03] CHEVALIER L., NAUTIBUS B., JAILLET F., BASKURT A.: Segmentation and superquadric modeling of 3d objects. *Journal of Winter School of Computer Graphics* (02 2003). 2

[DKC25] DHAL K., KASHYAP A., CHAKRAVARTHY A.: Collision avoidance of moving 3-d objects in dynamic environments. *IEEE Transactions on Automation Science and Engineering 22* (2025), 17914–17930. doi:10.1109/TASE.2025.3585450. 1

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (jun 1981), 381–395. URL: https://doi.org/10.1145/358669.358692, doi:10.1145/358669.358692. 1, 2, 7

[GB88] GROSS A., BOULT T.: Error of fit measures for recovering parametric solids. In *[1988 Proceedings] Second International Conference on Computer Vision* (1988), pp. 690–694. doi:10.1109/CCV.1988.590052. 3

[IB12] ISACK H., BOYKOV Y.: Energy-based geometric multi-model fitting. *International Journal of Computer Vision 97* (04 2012), 123–147. doi:10.1007/s11263-011-0474-7. 2

[LSM94] LEONARDIS A., SOLINA F., MACERL A.: A direct recovery of superquadric models in range images using recover-and-select paradigm. In *Computer Vision — ECCV '94* (Berlin, Heidelberg, 1994), Eklundh J.-O., (Ed.), Springer Berlin Heidelberg, pp. 309–318. 3

[LWRC21] LIU W., WU Y., RUAN S., CHIRIKJIAN G. S.: Robust and accurate superquadric recovery: a probabilistic approach. *CoRR abs/2111.14517* (2021). URL: https://arxiv.org/abs/2111.14517, arXiv:2111.14517. 3

[MAK*23] MONNIER T., AUSTIN J., KANAZAWA A., EFROS A., AUBRY M.: Differentiable blocks world: Qualitative 3d decomposition by rendering primitives, 07 2023. doi:10.48550/arXiv.2307.05473. 3

[MF14] MAGRI L., FUSIELLO A.: T-linkage: A continuous relaxation of j-linkage for multi-model fitting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2014). 2

[MLB21] MAGRI L., LEVENI F., BORACCHI G.: Multilink: Multi-class structure recovery via agglomerative clustering and model selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 1853–1862. 2

[PUG19] PASCHALIDOU D., ULUSOY A. O., GEIGER A.: Superquadrics revisited: Learning 3d shape parsing beyond cuboids. *CoRR abs/1904.09970* (2019). URL: http://arxiv.org/abs/1904.09970, arXiv:1904.09970. 3

[PvGG20] PASCHALIDOU D., VAN GOOL L., GEIGER A.: Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (June 2020). 3

[RSL22] RAMAMONJISOA M., STEKOVIC S., LEPETIT V.: Monteboxfinder: Detecting and filtering primitives to fit a noisy point cloud, 07 2022. doi:10.48550/arXiv.2207.14268. 3

[SB90] SOLINA F., BAJCSY R.: Recovery of parametric models from range images: the case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 2 (1990), 131–147. doi:10.1109/34.44401. 3

[SWK07] SCHNABEL R., WAHL R., KLEIN R.: Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum 26*, 2 (6 2007), 214–226. 3, 4

[TF08] TOLDO R., FUSIELLO A.: Robust multiple structures estimation with j-linkage. In *Computer Vision – ECCV 2008* (Berlin, Heidelberg, 2008), Forsyth D., Torr P., Zisserman A., (Eds.), Springer Berlin Heidelberg, pp. 537–547. 2

[TSG*17] TULSIANI S., SU H., GUIBAS L. J., EFROS A. A., MALIK J.: Learning shape abstractions by assembling volumetric primitives. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 1466–1474. doi:10.1109/CVPR.2017.160. 3

[VL01] VINCENT E., LAGANIERE R.: Detecting planar homographies in an image pair. pp. 182 – 187. doi:10.1109/ISPA.2001.938625. 7

[WLL*25] WU Y., LI W., LIU Z., LIU W., CHIRIKJIAN G. S.: Autonomous learning-free grasping and robot-to-robot handover of unknown objects. *Autonomous Robots 49*, 3 (2025), 1–16. 1

[ZYZ*23] ZHANG C., YANG Z., ZHUO H., LIAO L., YANG X., ZHU T., LI G.: A lightweight and drift-free fusion strategy for drone autonomous and safe navigation. *Drones* (2023). URL: https://api.semanticscholar.org/CorpusID:255697797. 1