Rheinische
Friedrich-Wilhelms-
Universität Bonn

Institute for Computer Science
Department VI
Autonomous Intelligent Systems

Rheinische
Friedrich-Wilhelms-Universität Bonn

Master thesis

Efficient 3D shape co-segmentation from single-view point clouds using appearance and isometry priors

Author:
Nikita Araslanov

First Examiner:
Prof. Dr. Sven Behnke

Second Examiner:
Prof. Dr. Jürgen Gall

Advisor:
Dr. Seongyong Koo

Submitted: March 24, 2016

Declaration of Authorship

I declare that the work presented here is original and the result of my own investigations. Formulations and ideas taken from other sources are cited as such. It has not been submitted, either in part or whole, for a degree at this or any other university.

Location, Date

Signature

Abstract

In our daily life, we seamlessly interact with objects exhibiting a variety of shapes. For example, mugs exist in a wide spectrum of geometrical forms and, yet, it takes us no effort to use a new instance. Conceivably, this ability is a highly desired skill in autonomous robotics: knowledge from one shape, such as grasping a handle of a mug, can be easily used for a new shape if the grasping part is identified.

In this work, we formulate this problem as shape co-segmentation which seeks to establish semantic correspondence between shape parts. Unfortunately, current state-of-the-art approaches to co-segmentation have practical limitations: they heavily rely on the diversity of large training sets, and their algorithms are tailored to handle watertight object meshes.

In accordance with these limitations, we impose stringent conditions. Firstly, we assume that only a single shape is available as a reference, and, secondly, only a partial view of the query shape is provided.

In this thesis, we propose a novel co-segmentation approach that constructs a part-based shape representation. We learn shape appearance of individual parts using feature encoding, such as Bag-of-Words and Fisher vectors, and build an intrinsic prior measuring isometric distortion between the parts with a distribution of diffusion distances. The query shape is pre-segmented by cutting through concave regions and the obtained segmentation graph is transformed into a Conditional Random Field (CRF) using the shape appearance and isometry prior.

We evaluated our approach on a large set of partial views generated from 15 categories of the Labelled Princeton Segmentation Benchmark. We also ran experiments on point cloud data obtained with an RGB-D sensor. The results of the evaluation demonstrated that our approach outperforms the state-of-the-art both in accuracy and efficiency.

Acknowledgements

First and foremost, I thank Dr. Seongyong Koo for prudent supervision and tireless encouragement throughout the work on the thesis. I express deep gratitude to Prof. Sven Behnke for motivating this work, providing excellent working environment and insightful discussions. I am also thankful to Prof. Jürgen Gall for invaluable suggestions that helped shaping this thesis.

I extend my heartfelt thanks to Aura Mũnoz and Rasha Sheikh for interest in this work, proofreading and helpful comments. I also gratefully acknowledge the fruitful discussions with Hannes Schulz that lead to an important refinement in the evaluation procedure.

Last but not least, I am grateful to Sü whose but a fleeting spark gave a source of infinite inspiration.

Contents

List of Figures

List of Algorithms

1 Introduction

1.1 Motivation

As humans, we generally find ourselves comfortable interacting with objects of diverse shapes that can be ascribed to the same semantic category. Mugs, for example, take a variety of shapes, though their function of containing liquid and drinking is not impeded. It takes us little effort to adapt to these shape variations once we see an object and learn its use. This knowledge can then be seamlessly generalised to all other objects of similar type that we encounter later in life.

Conceivably, the ability to generalise, or transfer knowledge between objects, is an important skill in autonomous robotics and is related to a shape matching problem. In manipulation tasks, determining a grasp position of a previously unseen object might only require a retrieval of the corresponding grasp from a known analogue (Li et al., 2007; Saxena et al., 2008). Similarly, usage of articulated objects (e.g. scissors) and items from easily deformable materials can be adapted from their single configuration (Schulman et al., 2013). An important task in human-robot interaction, tracking of human body parts, can be enabled with only one labelled body model (Ye et al., 2011). Information about corresponding parts between compatible bodies, such as a human and a humanoid robot, can also find application in learning from demonstration and robot teleoperation (Du et al., 2012).

Shape matching is a challenging problem in general, owing to the ill-posed nature of deformations. This can be exemplified with an elastic string shown in Figure 1.1. While the precise deformation model is usually unknown, it may include elastic "stretching", "shrinking" (Figure 1.1b) and isometric "bending" (Figure 1.1c) deformations, variations of scale (Figure 1.1d) and topology (Figure 1.1e). Moreover, partial views introduce additional complexity to the problem (Figure 1.1f).

Due to inherent ambiguities created by these deformations, a point-wise correspondence may not have a single solution and, therefore, is not well defined.

In this thesis we define the problem of shape correspondence as an instance of co-segmentation with the goal to establish a semantic correspondence between shape *subparts*. We understand subparts to fulfil a certain subfunction within the working of the whole shape, such as legs of a chair for stability, or a handle of a vase

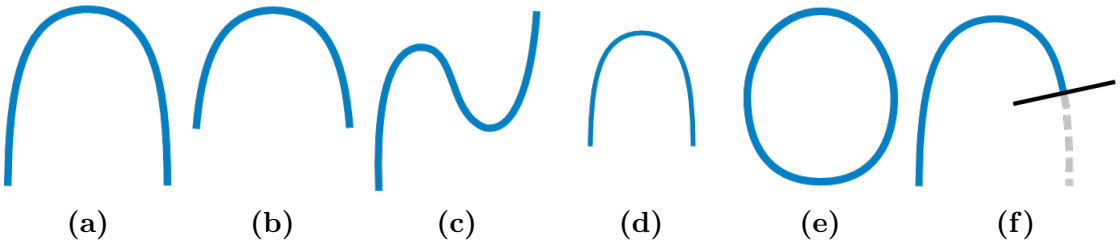


Figure 1.1: Shape variations of a deformable string: **(a)** original model; **(b)** elastic deformation; **(c)** isometric deformation; **(d)** scaling; **(e)** topology change; and, **(f)** partial views.

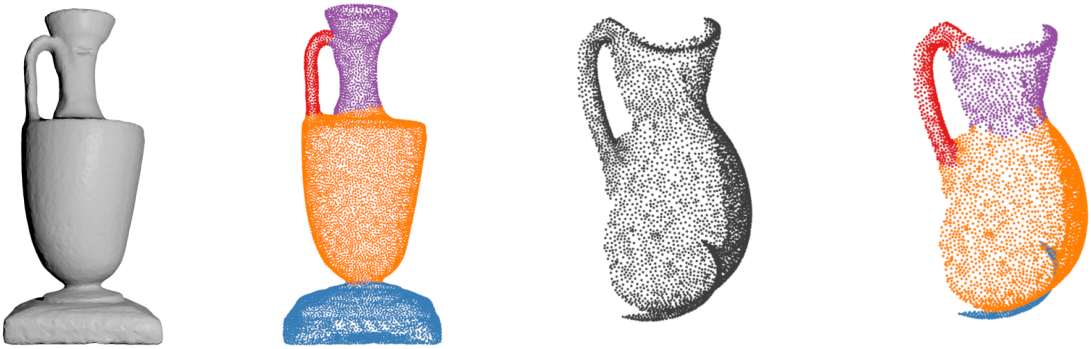
for grasping. In contrast to pointwise correspondence problem, co-segmentation does not seek estimates of the deformation model. Instead, it attempts to model shape structure based on subpart appearance and their topological relation. We argue that this formulation lends itself well for practical applications where high shape discrepancies between same-category objects and partial views are most ubiquitous.

We begin with a specific problem formulation in the next section followed by a review of the related work. At the end of this chapter, we analyse the current state-of-the-art approaches and propose an alternative concept developed in Chapter 3. We give some background information in Chapter 2 that serves as the base to our solution. The experimental setup and evaluation results are the main topics of Chapter 4. In the last Chapter 5 we draw conclusions and outline the path for future work.

1.2 Problem definition and objective

In a real-life scenario our scanty prior knowledge may comprise a single colourless physical object or a CAD model with designated constituent parts marked with labels. It is expected that the observer might require some "learning" time in order to analyse the shape geometry of the given item. We refer to the provided object as the *reference shape* and assume the associated labelling to be given in the form of shape segments. In the ensuing time, the observer is presented with a partially visible object from the same category in an arbitrary pose. We call it the *query shape* and investigate the case when it is represented by a point cloud. The new data is not dissimilar to a setting where a single frame has been obtained from a depth sensor. Our task is to demarcate segments on the query shape which semantically correspond to those on the reference shape.

More formally, let $\mathcal{S} := \bigcup \mathcal{S}_i$ be the union of the reference shape segments



(a) Reference shape **(b)** Query shape **(c)** Ground truth

Figure 1.2: The co-segmentation problem

and the mapping $\ell : \mathcal{S} \to L$, $L \subset \mathbb{Z}$ define segment labels. A query shape $\mathcal{T} := \{t_i \mid t_i \in \mathbb{R}^3\}$ is a partially visible point set of an object which belongs to the category of the reference shape, although it can be geometrically dissimilar to some unknown degree. The task is to find a segmentation $\bigcup \mathcal{T}_j = \mathcal{T}$ with the mapping $\ell^\star : \mathcal{T} \to L$ such that $\ell^\star(\mathcal{T}_j) = \ell(\mathcal{S}_i)$ if and only if segments $\mathcal{S}_i$ and $\mathcal{T}_j$ represent semantically corresponding parts.

As a remark, shape analysis considers only a number of criteria that define the semantic correspondence. Therefore, by finding a geometrically and topologically most compatible configuration of the query shape we can only at best hope to establish a semantic correspondence as well.

The challenges of the problem can be seen from an instance of vase co-segmentation shown in Figure 1.2. The provided labelling of the reference shape distinguishes between four segments (Figure 1.2a): the neck (violet), handle (red), body (orange) and the base (blue). The query shape is a single-view point cloud (Figure 1.2b) of a vase with a different shape which in this case contains the same parts as the reference (Figure 1.2c). In addition to potential pose variation, the geometry of the shape parts can be strikingly different, as can be observed from the bases of the two shapes. Therefore, it cannot be hoped that a feature descriptor alone can be fully relied on. Some context information, however, such as relative location of the parts (e.g. the base is "farther away" from the handle than the neck) and structural constraints (e.g. the base can be connected to the body, but not the handle) can be exploited to offset the biased description of local features.

We summarise the objectives of the present work as follows:

1. Build a shape representation from a limited prior knowledge (a single object);

2. Employ rotation-invariant feature encoding to describe shape appearance;

3. Embed category-specific structural constraints to augment the description of local geometry.

1.3 Related work

In this section, we give a brief overview of the previous work related to the co-segmentation problem. As will become clear from the next subsection, co-segmentation approaches make a careful choice of feature descriptors, a technique to build a shape representation and integrate it into an objective function, usually formulated as an inference problem of a probabilistic graphical model. For this reason, we shortly review these topics in the subsections to come.

1.3.1 Overview

In one of the state-of-the-art approaches Kalogerakis et al. (2010) modelled the co-segmentation problem as a Conditional Random Field (CRF). They trained a JointBoost classifier (Torralba et al., 2007) for unary and pairwise terms represented by contextual features. The latter was obtained by re-training the classifier on the histograms of unary feature predictions in order to capture a global distribution of labels around each mesh face. A similar path was undertaken by Kaick et al. (2011) who trained a GentleBoost classifier on 60% of randomly selected shapes for each class. They added an additional "intra-edge" pairwise term to the objective in order to distinguish between different shape parts based on their feature dissimilarity.

The common idea of these state-of-the-art co-segmentation methods (Kaick et al., 2011; Kalogerakis et al., 2010) is to model each shape represented by a mesh via a Conditional Random Field (CRF). Each node in the CRF is associated with a face of the mesh and the nodes are connected if the respective faces share a common edge. The corresponding energy function is minimised

$$E(\mathbf{x}) = \sum_i \phi(x_i) + \sum_{i,j} \phi(x_i, x_j), \tag{1.1}$$

where the terms $\phi(x_i)$ and $\phi(x_i, x_j)$ are unary and pairwise potentials, respectively. The unary term models geometrical similarity of a single face by means of shape descriptors. In the same vein, the pairwise term models segment boundaries and takes into account a number of geometrical cues, such as dihedral angles between two neighbouring faces.

In the following years, a number of unsupervised approaches were presented

that perform coherent labelling of multiple shapes simultaneously. In one approach, Huang et al. (2011) formulated the problem as a quadratic integer program and applied a linear relaxation to efficiently solve it. Sidi et al. (2011) employed spectral clustering based on diffusion maps measuring similarities of initial shape segments. The statistical model obtained from the clusters was then used to refine boundaries between the segments. More recently, Meng et al. (2013) initialised co-segmentation by clustering similar patches together using normalised cuts. They subsequently refined the result by alternating between energy minimisation defined in terms of the Markov Random Field (MRF) and the parameter update of the Gaussian Mixture Model (GMM) used to describe shape parts.

It is worth stressing that supervised approaches required a significant part of the datasets for training. On the other hand, unsupervised approaches cannot make use of the provided segmentation for the reference shape. For this reason, we do not follow the latter line of research further.

1.3.2 Feature extraction

Characterisation of 3D shape with feature descriptors has been extensively studied. The following is a very short overview of popular approaches to 3D shape description relevant to our work.

Spin images (Johnson and Hebert, 1999) define a cylindrical coordinate system and accumulate projections of the points within the support into bins. The repeatability of a local reference frame is the cornerstone of the SHOT descriptor (Signature of Histograms of OrienTations) proposed by Tombari et al. (2010). It combines local histograms of normals over 3D volumes of a superimposed grid into a signature. Instead of binning the normals, the Point Feature Histogram, PFH (Rusu, Marton, et al., 2008), and its fast successor, FPFH (Rusu, Blodow, et al., 2009), compute various angles from pairs of normals and the vector defining relative location of the point to its neighbour.

A number of comparative studies, such as the one conducted by Guo et al. (2016), assist in making a practical choice of local descriptors with respect to the application domain. In particular, SHOT, PFH and FPFH descriptors have shown best performance in object recognition from random views.

In recent years, the Laplace-Beltrami operator became an important tool for shape analysis, matching and retrieval of non-rigid shapes. The eigenfunctions of Laplace-Beltrami operator are invariant to isometric deformations. Local descriptors based on the eigenfunctions, such as the Heat Kernel signature (Sun et al., 2009) and its scale-invariant counterpart (M. M. Bronstein and Kokkinos, 2010), demonstrated state-of-the-art performance in non-rigid shape retrieval (A. M.

Bronstein, M. M. Bronstein, Guibas, et al., 2011). The diffusion distances computed from the eigenfunctions were also used for non-rigid shape matching (A. M. Bronstein, M. M. Bronstein, Kimmel, et al., 2010). In this work, they were shown to be more resilient than geodesic distances when partial or self-occlusions result in topological changes.

1.3.3 Feature encoding

Feature encodings for 3D shape representation mainly derive from those used in 2D image processing. For a comparative evaluation of these methods on object recognition benchmarks we refer the reader to the survey by Chatfield et al. (2011).

A number of approaches applied the bag of visual words model to 3D shape retrieval. Y. Liu, Zha, et al. (2006) created the vocabulary from local spin images and used Kullback-Leibler divergence as a similarity measure between the quantised vectors. Similarly, Ohbuchi et al. (2008) extracted SIFT descriptors from images constructed from multiple view directions, not dissimilar in spirit to the Light Field Descriptor. Toldo et al. (2009) first pre-segmented the shape using a weighted, convexity sensitive fast marching. One of the features they used was a Geodesic Context that measured the geodesic distance between one region centroid and the centroids of other regions. They subsequently constructed a signature of bag-of-words (BoW) histograms each containing frequencies of one sub-part for a different number of bins. For a non-rigid shape retrieval, Ovsjanikov et al. (2009) used spatial-sensitive BoW encoding of the heat kernel by accumulating co-occurrences of word pairs into a matrix. They used a simple L1-distance for similarity measure. A similar construction was used by Lavoué (2012) to encode spatial relations between BoW vectors. However, they used local features computed from projections of geometry on the eigenfunctions of a locally constructed Laplace-Beltrami operator.

Fisher vectors introduced by Jaakkola, Haussler, et al. (1999) provide additional information beyond the statistics-accumulating nature of the BoW encoding. Perronnin et al. (2010) improved the discriminative properties of the Fisher vector for large-scale image classification by additional vector normalisation. A survey of further approaches using Fisher vectors for image classification can be found in the work by Sánchez et al. (2013).

While Fisher vectors has matured in 2D object processing, it is still an emerging technique for 3D shape analysis. Only recently, **su2015multi** used Fisher vectors extracted from 2D images of multiple views and trained CNN for object recognition which outperformed the state-of-the-art.

1.3.4 Inference techniques

The co-segmentation approaches reviewed earlier rely on optimisation of a non-convex objective. For formulations in terms of a graphical model a number of well-established inference algorithms may apply. Kappes et al. (2014) gave a comparative overview of modern inference techniques. In the following, we take a brief look at some of the most successful.

The graph cut algorithms proposed by Boykov et al. (2001) optimise the energy function by performing two types of moves: an α-expansion, which allows to change the label of a node to α, and an α-β-swap in which the labels of α and β node can be interchanged. The algorithms are guaranteed to converge to a local minimum in which no further allowed moves can be made. However, this guarantee holds only when the energy function is *submodular*. For example, an energy function of the form $E(x_1, ..., x_n) = \sum E^i(x_i) + \sum_{i<j} E^{i,j}(x_i, x_j)$ with binary variables $x_i \in \{0, 1\}$ is submodular if and only if the following inequality is satisfied (Kolmogorov and Zabin, 2004):

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0). \tag{1.2}$$

The family of belief propagation (BP) algorithms is based on message passing introduced by Pearl (1988). Although the original algorithm provides an exact solution only when the corresponding factor graph is a tree, there are no convergence guarantees for graphical models with cycles. Yedidia et al. (2005) proposed partitioning the original graph into regions of nodes and enabling the message passing between the regions instead of separate nodes. Although in practice this generalised belief propagation exhibited better accuracy and convergence properties, the choice of regions, apart from some heuristics proposed in the paper, remained an open problem. In another approach based on a tree-reweighted scheme Wainwright et al. (2003) computed the maximum likelihood (ML) estimate by maximising a concave lower bound on the log likelihood of the data. Similarly for MAP estimation, Wainwright et al. (2005) developed a family of tree-reweighted max-product algorithms which simplify the marginal polytope into a collection of tree-structured distributions. The shared configuration of the trees achieved by a message-passing algorithm was shown to reach a globally MAP-optimal solution. However, the convergence of their tree-reweighted algorithms could not be guaranteed. By contrast, the sequential tree-reweighted algorithm (TRW-S) developed by Kolmogorov (2006) converges to a local maximum of the bound if a weaker form of the tree agreement is satisfied.

For small graphs heuristic-based approaches can perform better both in accuracy and efficiency. Bergtholdt et al. (2010) used small but complete graphs to

represent their structure for object detection. They applied an $A^\star$-algorithm with admissible heuristics which outperformed the inference techniques based on belief-propagation.

1.4 Discussion

Although the current state-of-the-art co-segmentation methods (Kaick et al., 2011; Kalogerakis et al., 2010) generally performed well on challenging datasets (X. Chen et al., 2009; Sidi et al., 2011) they required a significant portion of the available datasets to be reserved for training. One other practical limitation is the computational cost of these methods. The optimisation complexity depends on the largest clique size present in the mesh. Although inference is NP-hard in general, it can be solved in polynomial time if the tree-width of the underlying graph is bounded (Freuderl, 1990; Robertson and Seymour, 1986) and for a triangulated graph the size of the tree-width is one less than the size of the largest clique (Chandrasekaran et al., 2012). However, complex shape structures may incur tens or hundreds of thousands of nodes and have an appreciable size of the label space. Considering the scale, adding auxiliary constraints (e.g. to introduce informative structural constraints (Mitra et al., 2014)) may considerably slow down the inference. The limitations also apply to point clouds where local information analogous to dihedral angles may be highly unstable.

We propose to revise the state-of-the-art with a two-step approach. In the first step, we designate potential segment boundaries with a cutting plane. Instead of letting the CRF decide on the segment boundaries using only local information, our observation is that segment boundaries are strongly correlated with concave regions of the shape and, hence, can be nominated in a purely unsupervised manner. In the second step, we apply a robust feature encoding scheme to classify the candidate segments by their semantic class. Our choice of the feature encoding will be largely motivated by their good performance on classification tasks in other domains, such as object recognition and retrieval. Overall, our approach considerably reduces the size of the CRF, since the number of candidate segments is usually drastically smaller than the number of faces in the shape. This gives us more freedom to add structural constraints based on diffusion distance that consider spatial locality of object parts.

Before we present out approach in 3, we review the background theory needed to implement our method.

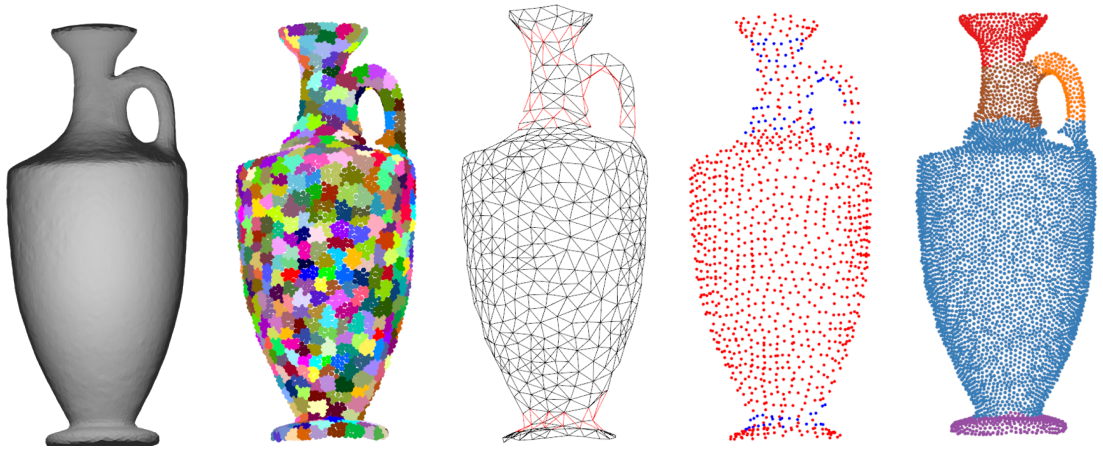
2 Background

In this section we review the necessary background for our approach. We begin with a recent state-of-the-art segmentation method in Section 2.1 which will serve as a baseline for our pre-segmentation step. The section 2.2 focuses on two popular local shape descriptors, SHOT and PFH, which will be used for a low-level feature extraction. We move on next to Sections 2.3 and 2.4 which give details on two widely used methods for feature encoding: Bags-of-Words and Fisher vectors. In section 2.5 we take a look at the Laplace-Beltrami operator and discuss its utility for our domain. In particular, we show how some useful intrinsic properties can be calculated on point clouds. Finally, we recall the TRW-S algorithm in Section 2.6 that provides an efficient inference with strong convergence and optimality guarantees.

2.1 Segmentation with Constrained Planar Cuts

A recently introduced segmentation based on constrained planar cuts for 3D shapes (Schoeler et al., 2015) is an unsupervised bottom-up approach that outperformed existing supervised and unsupervised approaches on the Princeton Object Segmentation Benchmark (X. Chen et al., 2009). The underlying idea of the method is to use local regions with pronounced concavity as candidates for segment boundaries. Although simple Euclidean planes were used in the original work to designate the boundaries, more complex models could be adopted.

The approach proceeds as follows. First, the cloud is over-segmented into supervoxels. The centroids of the supervoxels are then connected with edges to form an adjacency graph. The cornerstone of the procedure is the adoption of a weighted RANSAC to cut though the edges of the adjacency graph. To that end, the adjacency graph is represented by a point cloud where each point is an average of the two points connected by an edge in the original graph. In order to constrain the cutting to regions of local concavities, the set of points $\mathcal{P}_m$ lying within the support region of the candidate model m proposed by RANSAC is also subject to



(a) **(b)** **(c)** **(d)** **(e)**

Figure 2.1: Segmentation with constrained planar cuts. (a) The input model represented by a point cloud. (b) Initial segmentation into supervoxels. (c) The adjacency graph computed from supervoxel centroids with red edges indicating concavities. (d) Point cloud representation of the adjacency graph. Blue points denote concave regions. (e) Final segmentation obtained by cutting with planes.

a euclidean clustering. For each cluster n the score is computed as

$$\mathcal{S}_m^n = \frac{1}{|\mathcal{P}_m^n|} \sum_{i \in \mathcal{P}_m^n} \omega_i t_i, \tag{2.1}$$

where $\mathcal{P}_m^n$ denotes the support region constrained to the cluster n; the parameter t_i favours orthogonal cuts through the edges by taking account of the vector $\mathbf{s_m}$ perpendicular to the model m, and the direction of the connection between the supervoxel centroids $\mathbf{x_1}$ and $\mathbf{x_2}$, $\mathbf{d_i} := \frac{\mathbf{x_1}-\mathbf{x_2}}{\|\mathbf{x_1}-\mathbf{x_2}\|_2}$, and is defined as

$$t_i = \begin{cases} |\mathbf{d_i} \cdot \mathbf{s_m}|, & \text{edge } i \text{ is concave,} \\ 1, & \text{edge } i \text{ is convex;} \end{cases} \tag{2.2}$$

and finally, $\omega_i := \mathcal{H}(\alpha_i - \beta_{thresh})$ is a heaviside step function with $\alpha_i := d_i \cdot (\mathbf{n_{i2}} - \mathbf{n_{i1}}) \cos^{-1}(\mathbf{n_{i1}} \cdot \mathbf{n_{i2}})$, where $\mathbf{n_{i1}}$ and $\mathbf{n_{i2}}$ are normals of adjacent supervoxels. Thus, the parameter ω_i simply encourages cutting through concave edges.

The entire procedure handles newly generated segments recursively until no cuts can be found satisfying the condition $\mathcal{S}_m^n \geq \mathcal{S}_{min}$. Figure 2.1 demonstrates the main steps of the outlined method.

2.2 Local Feature Descriptors

This section briefly reviews the two most successful local shape descriptors relying on shape curvature and normals: SHOT and PFH.

2.2.1 Signature of Histograms of OrienTations (SHOT)

The Signature of Histograms of OrienTations (SHOT) descriptor proposed by Tombari et al., 2010 encodes histograms of first-order differential entities within each cell of a superimposed 3D grid. In addition, a stable reference frame (RF) is defined for each cell which allows for a geometrically meaningful grouping of the histograms, i.e. without significant loss of spatial information. The computation of a repeatable RF is based on the Eigenvalue Decomposition (EVD) of a covariance matrix $\mathbf{M}$ computed for each feature point $\mathbf{p}$ and its neighbours in $\mathcal{N}(\mathbf{p}) := \{\mathbf{p}_i \mid \|\mathbf{p} - \mathbf{p}_i\|_2 \leq R\}$ as a weighted sum:

$$\mathbf{M} = \frac{1}{\sum_{d_i}(R - d_i)} \sum_{d_i \leq R} (R - d_i)(\mathbf{p}_i - \mathbf{p})(\mathbf{p}_i - \mathbf{p})^T, \tag{2.3}$$

where $d_i = \|\mathbf{p} - \mathbf{p}_i\|_2$. The sign disambiguation for EVD is resolved by reorienting each eigenvector in the prevalent direction of the input data. With the RF computed, each local histogram maintains bins with point counts according discretisation of the dot product $\mathbf{n_p} \cdot \mathbf{n}_{v_i}$ between the normal of the feature point $\mathbf{n_p}$ and the corresponding part of the grid $\mathbf{n}_{v_i}$. An isotropic spherical grid that encompasses partitions along the radial, azimuth and elevation axes is used for signature structure.

2.2.2 Point Feature Histogram (PFH)

Like SHOT, the Point Feature Histogram (PFH) Rusu, Marton, et al., 2008 is an affine-invariant local 3D shape descriptor based on point coordinates and surface normals. The PFH accumulates pairwise difference of the angle between normals in the k-neighbourhood of each point $\mathbf{p}$. Concretely, for each pair of points $\mathbf{p}_s$ and $\mathbf{p}_t$ located in the k-neighbourhood of $\mathbf{p}$ such that $\langle \mathbf{n}_s, \mathbf{p}_t - \mathbf{p}_s \rangle \leq \langle \mathbf{n}_s, \mathbf{p}_t - \mathbf{p}_s \rangle$, define $\mathbf{u} = \mathbf{n}_s$, $\mathbf{v} = (\mathbf{p}_t - \mathbf{p}_s) \times \mathbf{u} \|\mathbf{p}_t - \mathbf{p}_s\|$, $\mathbf{w} = \mathbf{u} \times \mathbf{v}$. The following measures of angles, $f_0 = \langle \mathbf{v}, \mathbf{n}_t \rangle$, $f_1 = \|\mathbf{p}_t - \mathbf{p}_s\|$, $f_2 = \langle \mathbf{u}, \mathbf{p}_t - \mathbf{p}_s \rangle / f_1$ and $f_3 = atan(\langle \mathbf{w}, \mathbf{n}_t \rangle, \langle \mathbf{u}, \mathbf{n}_t \rangle)$, are subsequently used as bin coordinates for the histogram.

An efficient implementation of the PFH simplifies its computation by estimating the entities $f_0, ..., f_3$ only for the feature point $\mathbf{p}$ and its neighbours $\mathbf{p}_i$. Furthermore, the resulting values are only concatenated to form the so called Simpli-

fied Point Feature Histogram (SPFH). The final Fast Point Feature Histogram (FPFH) Rusu, Blodow, et al., 2009 is obtained by

$$FPFH(\mathbf{p}) = SPFH(\mathbf{p}) + \sum_{\mathbf{p}_i \in \mathcal{N}(\mathbf{p})} \frac{1}{d_i} SPFH(\mathbf{p_i}), \tag{2.4}$$

where d_i is defined as in the previous subsection and $\mathcal{N}(\mathbf{p})$ is a k-neighbourhood of point $\mathbf{p}$.

2.3 Bag-of-Words

Let $M = \{m_t, t = 1, ..., T\}$ is the set of T shape feature vectors. The so-called "codebook" can be generated by fitting a generative model to the given feature set M (e.g. using K-means). The clusters of the codebook can then be related to a single dimension in the bag-of-words representation. Concretely, let $\boldsymbol{\mu} = \{\mu_i, i = 1, ..., K\}$ denote the K-means clusters fitted to a given feature space $\mathcal{F}$ and M be the feature set as defined previously such that $M \subset \mathcal{F}$. The ith dimension of the resulting bag-of-words histogram is the proportion of features assigned to cluster μ_i. To compare two histograms $S_1 = (u_1, u_2, ..., u_K)$ and $S_2 = (w_1, w_2, ..., w_K)$ the $\mathcal{X}^2$ distance can be used:

$$D(S_1, S_2) = \frac{1}{2} \sum_{i=1}^{T} \frac{(u_i - w_i)^2}{u_i + w_i} \tag{2.5}$$

2.4 Fisher vectors

Instead of encoding data with a limited vocabulary, Fisher vectors encode the likelihood of the data with respect to parameters of the generative model. Similarly to the bag-of-words approach, we first represent data in the Gaussian Mixture model (GMM) using Maximum Likelihood (ML) estimation. The parameters of the GMM are $\lambda = \{\omega_i, \mu_i, \sigma_i, i = 1, ..., K\}$, where ω_i, μ_i and σ_i are weight, mean and covariance of the $i - th$ component and K is the number of components. Fisher vectors encode $\bigtriangledown_\lambda p(M, \lambda)$, where $M = \{m_t, t = 1, ..., T\}$ is the set of T local descriptors extracted from the shape. The computation of Fisher vectors proceeds a follows. Let $\gamma_t(i)$ be the soft assignment if the descriptor m_l to the component i:

$$\gamma_t(i) = \frac{\omega_i u_i(m_t)}{\sum_{j=1}^{K} \omega_j u_j(m_t)} \tag{2.6}$$

The gradients $G^M_{\mu,i} := \frac{\partial \log p(M|\lambda)}{\partial \mu_i}$ and $G^M_{\sigma,i} := \frac{\partial \log p(M|\lambda)}{\partial \sigma_i}$ are computed as:

$$\begin{aligned} G^M_{\mu,i} &= \frac{1}{T\sqrt{\omega_i}} \sum_{t=1}^{T} \gamma_t(i) \left(\frac{m_t - \mu_i}{\sigma_i} \right) \\ G^M_{\sigma,i} &= \frac{1}{T\sqrt{2\omega_i}} \sum_{t=1}^{T} \gamma_t(i) \left(\frac{(m_t - \mu_i)^2}{\sigma_i^2} - 1 \right) \end{aligned} \tag{2.7}$$

with vector division denoting an element-wise operation. The resulting gradient vector is simply a concatenation of the partial derivatives (2.7).

2.5 Laplace-Beltrami operator

In recent years, the Laplace-Beltrami operator has become one of the most widely used tools for shape analysis. In this section we will recall the basic definitions and methods to approximate the operator on real data.

Let $\mathcal{M}$ be a smooth manifold of dimension k that is isometrically embedded in some Euclidean space $\mathbb{R}^d$. Without loss of generality, it is assumed that $\mathcal{M}$ is connected (the following results can be applied component-wise otherwise). Let f be a twice continuously differentiable function $f \in C^2(M)$ (e.g. imposed on some shape $\mathcal{M}_i \in \mathcal{M}$) and $\bigtriangledown_M f$ denote the gradient vector field of f on $\mathcal{M}$. The Laplace-Beltrami operator $\triangle_{\mathcal{M}}$ of f is defined as the divergence of the gradient:

$$\triangle_{\mathcal{M}} f = div \bigtriangledown_{\mathcal{M}} (f). \tag{2.8}$$

Hence, for $\mathcal{M} \subset \mathbb{R}^2$ the operator is simply $\triangle_{\mathbb{R}^2} f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$.

Clearly, in majority of real world cases the function describing a given shape is unknown, and there can be no closed-form solution for computing the Laplace-Beltrami operator. However, a number of approaches exist to approximate the Laplace-Beltrami operator on meshes and point clouds. Some of the approximation schemes on meshes were reviewed by Reuter et al., 2009. In the following we will focus on computing the operator for point clouds.

2.5.1 Laplace operator from point clouds

One of the earliest algorithms for approximating the Laplace-Beltrami operator from an arbitrary point cloud was presented by Belkin et al. (2009). The algorithm, called *PCD Laplacian*, builds a local patch around each data point and estimates the heat kernel on each patch. The results show that these local patches

are sufficient to approximate the manifold Laplacian, even though they do not form a global mesh. However, in order to obtain a provable reconstruction of the LB operator, the method requires strict sampling conditions which limits its application in practice. In a later work, Liang et al. (2012) addressed this issue by local surface reconstruction using moving least squares (MLS), which we later refer to as *MLS LB*. The advantages of their method can be summarised using the following criteria:

- **Accuracy**. The error of the computed eigenvalues was lower by at least one order of magnitude compared to PCD Laplacian and of the same order as the mesh-based method and lower.
- **Efficiency**. Local approximation can be performed simultaneously for multiple points using multi-threading.
- **Flexibility**. Original paper used binomial polynomial of second degree to locally approximate the surface and solved the resulting quadratic problem. However, the model of the manifold can be approximated with polynomials of higher degree for complex shapes.
- **Stability**: Variations of density in the point cloud were shown to have a lesser effect on the accuracy than that of the PCD Laplacian.

In view of these performance points, the approach lends itself well for our problem. Next, we examine the main computation steps the algorithm.

Overview of MLS LB

Let $(\mathcal{M}, g)$ be a smooth surface in $\mathbb{R}^3$ and (s_1, s_2) be its local parametrisation near some point $p \in \mathcal{M}$. For a smooth function $f : \mathcal{M} \to \mathbb{R}$, the LB operator $\Delta_{\mathcal{M}}$ acting on f near p is defined by

$$\Delta_{\mathcal{M}} f = \sum_{i,j=1}^{2} \frac{1}{\sqrt{g}} \frac{\partial}{\partial s_i} \left(\sqrt{g} g^{ij} \frac{\partial f}{\partial g} \right) \tag{2.9}$$

where coefficients g^{ij} are the components of the inverse of the metric tensor $G = [g_{ij}]$ and $g = \det(G)$.

The method (Liang et al., 2012) can be summarised in three steps:

1. At each point p_i define a local coordinate system.
2. Use moving least square (MLS) to calculate a bivariate polynomial which best approximates the surface locally.

3. Modify the classical MLS by introducing a special weight function to locally approximate any function f defined on $\mathcal{M}$.

We will look at these steps in detail.

Building Local Coordinate System

For each point $p_i \in P$ the local coordinate system is defined by the normal and the local coordinates of p_i. The normal is computed by considering k-nearest neighbours (KNN) $N(i)$ and performing PCA of the covariance matrix $P_i = \sum_{k\in N(i)}(p_k - c_i)^T(p_k - c_i)$ in a standard way (Here, $c_i = \frac{1}{K}\sum_{k\in N(i)} p_k$). The eigenvectors (e_1^i, e_2^i, e_3^i) with associated eigenvalues $\lambda_1^i \geq \lambda_2^i \geq \lambda_3^i \geq 0$ form an orthogonal frame. The KNN of p_i are used further for surface and function approximation.

Local Surface Approximation

A local bivariate polynomial of degree two $z_i(x, y)$ is approximated by minimising the following weighted sum:

$$\sum_{k\in N(i)} \omega(\|p_k - p_i\|)\big(z_i(x_k^i, y_k^i) - z_k^i\big)^2 \tag{2.10}$$

where (x_k^i, y_k^i, z_k^i) are local coordinates of point p_k, $w(\cdot)$ is a positive weight function typically chosen as $w(d) = exp(-\frac{d^2}{h^2})$ and $h = max_{k\in N(i)}\|p_k - p_i\|$. With this local parametric approximation the metric tensor and other important quantities are computed. The LB operator is written as a linear combination of deriatives on the surface, given by

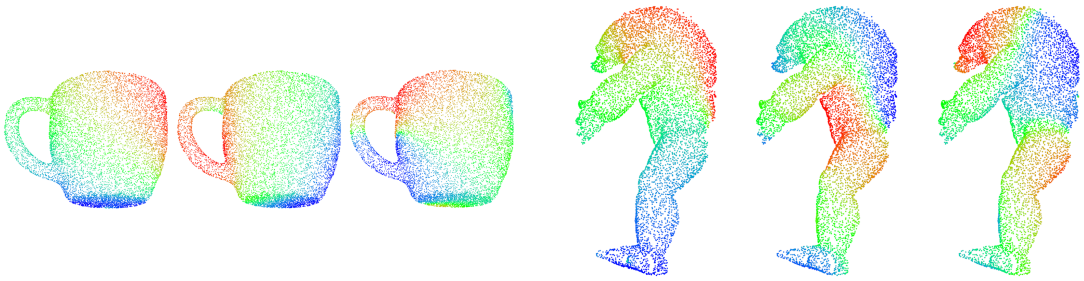
$$\delta_{\mathcal{M}} f = \alpha_1\frac{\partial f}{\partial x} + \alpha_2\frac{\partial f}{\partial y} + \alpha_3\frac{\partial^2 f}{\partial x^2} + \alpha_4\frac{\partial^2 f}{\partial x\partial y} + \alpha_5\frac{\partial^2 f}{\partial x\partial y} \tag{2.11}$$

where α_i's are computed by expanding and simplifying (2.9) (see Appendix 1).

Function Approximation

In order to locally approximate function $F_i(x, y) = c_1^i + c_2^i x + c_3^i y + c_4^i x^2 + c_5^i xy + c_6^i y^2$ locally defined on the manifold $\mathcal{M}$ for each point p_i, the following weighted sum is minimised:

$$\sum_{k\in N(i)} w(\|p_k - p_i\|)\big(F_i(x_k^i, x_k^i) - f_k\big)^2, \tag{2.12}$$



(a) **(b)**

Figure 2.2: Some eigenfunctions of some partial view clouds of a cup (a) and Armadillo (b).

where $f_k = f(p_k)$ and the weight is empirically chosen as $\omega(d) = 1$ if $d = 0$ and $\omega(d) = 1/K$ otherwise. Minimising (2.12) leads to the following linear system:

$$\sum \omega_k V_k^i (V_k^i)^T C^i = \sum \omega_k V_k^i f_k, \tag{2.13}$$

where $\omega_k = \omega(\|p_k - p_i\|)$, $C^i = [c_1^i, c_2^i, c_3^i, c_4^i, c_5^i, c_6^i]^T$ and $V_k^i = [1, x_k^i, y_k^i, (x_k^i)^2, x_k^i y_k^i, (y_k^i)^2]^T$. Solution to this system can be written as $C^i = M^i F$, where M^i is some $6 \times N$ matrix. The partial derivatives of f are proportional to c_j^i's. For example, $\frac{\partial f}{\partial x}(p_i) = c_2^i$. Using equation (2.11), the appoximation of the LB operator is $\Delta_{\mathcal{M}} f(p_i) = L_i F$, where L_i is some row vector. Therefore, the i-th row of the MLS LB operator is simply L_i.

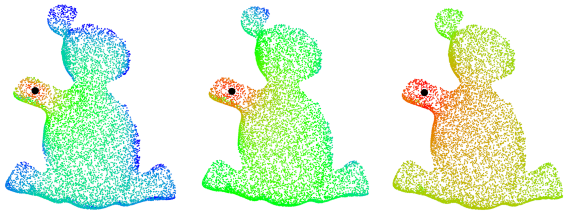
Some of the computed eigenfunctions are shown in Figure 2.2.

2.5.2 Spectral shape distances

As noted previously, eigenfunctions of Laplace-Beltrami operator are invariant under isometric deformations. This quality lies at the core of some informative spectral shape properties, such as diffusion and commute distance (M. M. Bronstein and A. M. Bronstein, 2010). We recall that diffusion distance measures proximity of two points x and y on the surface and can be computed as

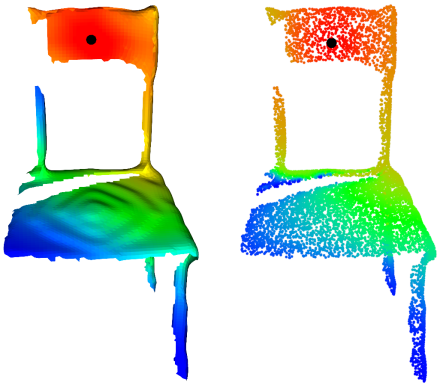
$$d_t^2(x, y) = \sum_i K^{2t}(\lambda_i)(\phi_i(x) - \phi_j(y))^2, \tag{2.14}$$

where λ_i and ϕ_i are ith eigenvalue and eigenfunction of the Laplace-Beltrami operator respectively, and K^t defines a scale space (or, a low-pass filter). A particular choice of K^t is based on the *heat operator* $H^t(\lambda) = e^{-t\lambda}$. In the interpretation of the diffusion distance as a random walk, d_t corresponds to the likelihood of reach-



$t = 0.01$ $t \approx 0.34$ $t = 2$

(a) Diffusion distance for various t



(b) Geodesic (left) and commute time distance (right)

Figure 2.3: Some illustrative properties of diffusion and commute time distance

ing y from x (or vice versa) within walk distance $t \geq 0$. The effect of parameter t on the diffusion distance is illustrated in Figure 2.3a. Unlike geodesic geometry, the diffusion distance was shown to be robust to topological noise (A. M. Bronstein, M. M. Bronstein, and Kimmel, 2009).

For many practical applications, one might be interested in avoiding the choice of t to decrease the impact of scale variation and computing an "average" diffusion distance. This notion is closely connected to the *commute time distance*:

$$d^2_{\mathrm{CT}}(x, y) = \sum_i \frac{1}{\lambda_i}(\phi_i(x) - \phi_j(y))^2, \tag{2.15}$$

which can be related to the diffusion distance by $\frac{1}{2}d^2_{\mathrm{CT}}(x, y) = \int_0^\infty d^2_t(x, y)dt$. We can trace the same resilience of the commute distances to topology changes when compared to geodesic distances as demonstrated in Figure 2.3b.

2.6 Inference with A^* search

Originally designed to efficiently obtain the ground-truth solution in small graphs, the inference based on $A^\star$ search outperformed other approaches used in the comparative study by Bergtholdt et al. (2010) even in runtime. In this section, we take a look at some of the notable features of this algorithm which will motivate its choice for our problem.

First, let us introduce some notation. Define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of n vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$. To each vertex $s \in \mathcal{V}$ we associate a variable x_s from some discrete space $\mathcal{X}_s$. The configuration $\mathbf{x}$ is a concatenation of variables

x_s assigned to each node $s \in \mathcal{V}$ such that $\mathbf{x} \in \mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2 \times ... \times \mathcal{X}_n$. For some index set $\mathcal{I}$ a family of potential functions $\{\phi_\alpha : \mathcal{X} \to \mathbb{R} \mid \alpha \in \mathcal{I}\}$ has a corresponding set of parameters $\{\theta_\alpha \mid \alpha \in \mathcal{I}\}$. The energy function is defined as

$$E(\mathbf{x} \mid \theta) = \sum_{\alpha \in \mathcal{I}} \theta_\alpha \phi_\alpha(\mathbf{x}) \tag{2.16}$$

To simplify notation, we let $\theta_{\alpha;\mathbf{x}_\alpha} := \theta_\alpha \phi_\alpha(\mathbf{x})$, i.e. $\theta_{\alpha;\mathbf{x}_\alpha}$ denotes the score of assigning variables $\mathbf{x}_\alpha$ to nodes α.

The A^* algorithm for inference relies on the following two components:

- a spanning tree T of the graph $\mathcal{G}$ used to compute the lower bound (*);
- and a fixed arbitrary ordering of the graph $\mathcal{G}$ (**).

The A^* induces a tree structure T^* on the configuration space $\mathcal{X}$. Each node of the tree corresponds to a partial configuration of length equal to the node's level in the search tree. The algorithm progressively expands nodes starting from the root node ("zero-length" configuration) and finishing at the leaf node representing a complete configuration. Hence, the nodes in the tree are connected by an edge if their configuration differs by only one variable.

The choice of the next node to expand is governed by a score value assigned to each node. A^*-search algorithm is guaranteed to find the *global* solution if the heuristic is *admissible*, i.e. the score of each node is a lower-bound estimate of the optimal solution. For large graphs, however, the exponential growth of the tree structure results in impractical storage demands. A common remedy to reduce the memory footprint is the tree pruning that excludes some branches of the tree from the search. The global solution cannot be guaranteed once this procedure has been applied.

Let $v^\star, u^\star \in T^*$ denote partial configurations such that $v^\star, u^\star \subset \mathcal{X}$ and their cardinalities given by $|v^\star|$ and $|u^\star|$ respectively. The prerequisite (**) implies a strict total order on $\mathcal{V}$ in a sense that $s < t$ if and only if the level at which node s is included in the configuration node of the tree T^* is less than the level of node t. While considering the expansion of node v^* with parent u^* in the search tree T^*, the search heuristic was defined by Bergtholdt et al. (2010) as follows:

$$\begin{aligned} H(v^* \mid u^*) := \min_{\substack{x \in \mathcal{X} \\ x|_{v^*} = v^*}} \Bigg[& \sum_{\substack{t \in \mathcal{V} \\ t > |v^*|}} \theta_{t;x_t} + \sum_{\substack{st \in \mathcal{E} \\ s \le |u^*|,\, t > |v^*|}} \theta_{s,t;x_s,x_t} + \\ & + \sum_{\substack{s > |u^*| \\ t > |v^*|}} \left(\sum_{st \in \mathcal{E}(T)} \theta_{s,t;x_s,x_t} + \sum_{st \in \mathcal{E} \setminus \mathcal{E}(T)} \min_{x_s \in \mathcal{X}_s} \theta_{s,t;x_s,x_t} \right) \Bigg], \end{aligned} \tag{2.17}$$

Algorithm 1: MAP-inference algorithm with A^*

Data: θ, T
Result: x, *opt*
$v^* \leftarrow []$, $\tau \leftarrow +\infty$;
while $|v^*| < |V|$ **do**
  $u^* \leftarrow v^*$;
  compute $H(v^* \mid u^*)$, $\forall v^*$, $|v^*| = |u^*| + 1$;
  for $i \in \mathcal{X}_{|u^*|+1}$ **do**
    $v^* \leftarrow \{u^*, i\}$;
    NodeQueue.insert($\{v^*, J(v^*) + H(v^*|u^*)\}$);
    if Size = MaxSize **then**
      $\Delta \leftarrow$ lowest value of the 50% worst energy estimates;
      $\tau \leftarrow$ `min` (τ, Δ);
    end
  end
  $v^* \leftarrow$ NodeQueue.pop()
end
$x \leftarrow v^*$, *opt* $\leftarrow$ false;
if $J(x) \leq \tau$ **then**
  opt $\leftarrow$ true;
end

where $\mathcal{E}(T) \subset \mathcal{G}$ are the edges of the preconditioned spanning tree T (*) of graph $\mathcal{G}$. Intuitively, the heuristic (2.17) consists of the terms including the assignment in the current configuration (the first two terms) and projected costs of future assignment (the third compound sum).

The following is given without proves which can be found in the original work by Bergtholdt et al. (2010).

Proposition 1 *Heuristic* (2.17) *is admissible; it is the lower bound corresponding of the energy of any configuration given by a top-down path in* T^*.

Lemma 1 *The lower bound* (2.17) *in the leaf node of the search tree* T^* *is equal to the energy function corresponding to a complete configuration represented by the node.*

For completeness, Algorithm 1 summarises the A^*-search algorithm for inference. The expansion begins with the root node denoted by [] and indicating an empty configuration. The algorithm then consequently adds new variables into a priority queue `NodeQueue` that maintains the ascending order of the nodes by the energy value. If the size of the tree reaches the maximum threshold given by `MaxSize` the pruning is applied. This mechanism keeps the memory demand mentioned earlier

in check. Parameter τ maintains the lowest value of the pruned branches in the tree T. Note that *opt* = false returned if the final energy value $J(x)$ exceeds τ merely implies that the solution is not guaranteed to be the global minimum.

3 Method

3.1 Overview

In this section we present our approach to the co-segmentation problem. The main goal of the proposed method is to build a part-based 3D shape representation from a range of single views. This representation enables the use of more holistic feature encodings that model complete shape subparts instead of accumulating an array of local descriptors. Furthermore, an unsupervised state-of-the-art segmentation can provide meaningful part candidates on the query shape. This allows structure constraints to be incorporated at the level of shape parts and avoid the limitation of modelling only the segment neighbourhood. The resulting model is effectively a moderately-sized complete graph.

The flowchart of our approach is presented in Figure 3.1. Initially, we scan the reference shape from multiple view angles using a virtual scanner. The data gathered from the single views is used to create a part-based representation of the reference shape. Next, we pre-segment the query shape by "cutting" through concave regions and obtain a segmentation graph whose nodes represent separate segments and edges indicate segment neighbourhood. Note that segment boundaries created with the cutting are "fuzzy" in the sense that the divided segments may eventually receive the same label. In the final step, we incorporate the prior knowledge and structural constraints from our part-based shape representation into a Conditional Random Field (CRF) framework to find the optimal labelling.

In contrast to the previous approaches (Kaick et al., 2011; Kalogerakis et al., 2010), the resulting graph is much more compact, since nodes represent whole shape segments rather than single mesh faces. Also, a feature representation of complete shape parts should intuitively be more discriminative than that of local descriptors with a limited support.

Note that our approach does not restrict the type of the underlying shape data structure. In present work we assume that the input shape is a point cloud whereas the reference shape is provided as a mesh. Moreover, implementation of the flowchart may vary depending on the specific choice of the shape representation and the objective function for label optimisation. This work addresses only one feasible configuration.

In the next section, we formulate our objective function by combining the notions of *intrinsic* and *extrinsic* similarities. We then elaborate on each step of the flowchart in Figure 3.1 in the consecutive sections.

3.2 Objective

Concrete models prescribing the spectrum of possible shape variations considerably simplify the task of establishing correspondences between a pair shapes. For example, assumptions of isometric deformations afford reasonable accuracy of pointwise correspondence (Q. Chen and Koltun, 2015) and there are emerging techniques addressing the problem for partially visible shapes (Rodolà et al., 2015).

Although isometry assumption does not hold in general, we can still exploit it in a weaker form by measuring *isometric distortion* (A. M. Bronstein, M. M. Bronstein, Kimmel, et al., 2010) that was originally developed for point-to-point shape matching. Given point sets $\mathcal{S}$, $\mathcal{T}$ and a pointwise correspondence $C = \{(s,t) \mid s \in \mathcal{S}, t \in \mathcal{T}\}$ the distortion is characterised by pairwise difference of the corresponding points,

$$\operatorname{dis}(C) := \sup_{(s,t),(s',t') \in C} \mid d_{\mathcal{S}}(s,s') - d_{\mathcal{T}}(t,t') \mid . \tag{3.1}$$

The objective to embed one surface into another with minimum distortion can be

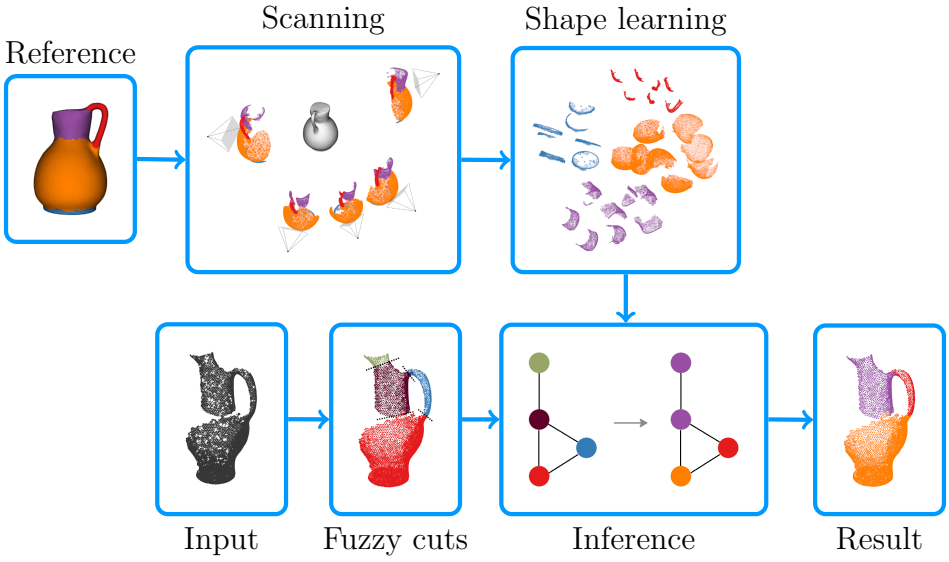


Figure 3.1: Overview of our approach

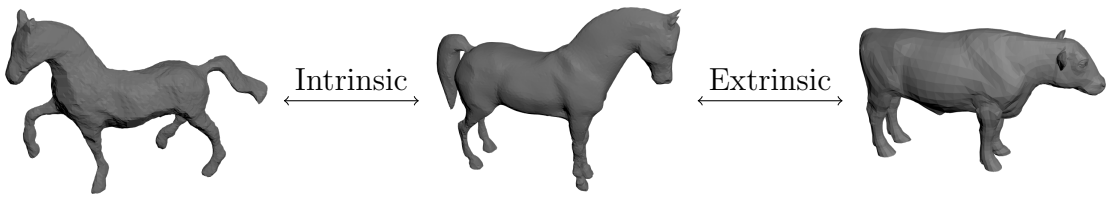


Figure 3.2: Illustration of intrinsic and extrinsic similarity

expressed using *Gromov-Hausdorff* distance,

$$d_{GH}(\mathcal{S}, \mathcal{T}) := \frac{1}{2} \inf_{C} \operatorname{dis}(C), \tag{3.2}$$

where inf stands for the infimum.

Clearly, measuring isometric discrepancy cannot be used to find part correspondences alone due to intrinsic symmetries and topology changes. A. M. Bronstein, M. M. Bronstein, Kimmel, et al. (2010) introduced the notion of *intrinsic* and *extrinsic* similarity that we illustrate in Figure 3.2. Isometric deformations do not affect intrinsic properties of the shape (Figure 3.2, left), whereas changes in shape appearance are non-isometric, yet bear visual resemblance to the original shape (Figure 3.2, right). The core idea of our contribution is to combine these two notions of the intrinsic and extrinsic similarity for a part-based shape representation. Accordingly, our objective is to minimise the discrepancy in the appearance of the subparts and inter-part isometric distortion.

Let us introduce the notation used throughout the chapter. We define a label function $\ell : \mathcal{S} \to L$ and let the label of segment $\mathcal{S}_i$ be denoted by ℓ_i for short. We relate probability $p(\ell_i \mid \mathcal{T}_j)$ to appearance similarity of the segment $\mathcal{T}_j$ with the segments in $\mathcal{S}$ labelled ℓ_i. Similarly, we model probability $p(\ell_i, \ell_j \mid \mathcal{T}_i, \mathcal{T}_j)$ to measure the degree of isometric distortion between each pairwise assignment. Our objective can be formulated as a maximum likelihood estimate of the form:

$$\underset{\ell}{\text{maximize}} \quad \prod_{i,j} p(\ell_i \mid \mathcal{T}_i) p(\ell_j \mid \mathcal{T}_j) p(\ell_i, \ell_j \mid \mathcal{T}_i, \mathcal{T}_j), \tag{3.3}$$

which is equivalent to a CRF optimisation in the logarithmic scale:

$$\underset{\ell}{\text{minimize}} \quad -\sum_{i} \log p(\ell_i \mid \mathcal{T}_i) - \sum_{i,j} \log p(\ell_i, \ell_j \mid \mathcal{T}_i, \mathcal{T}_j). \tag{3.4}$$

The description of our approach is split into three parts. In the next section,

we develop the segmentation step used to create segment candidates on the query shape. The learning of the shape representation is detailed in Section 3.4. First, we model the shape appearance using feature encoding, and then define the pairwise measure distance between shape segments based on the distribution of diffusion distances. We wrap up the chapter by incorporating the learned model as the unary $p(\ell_i \mid \mathcal{T}_j)$ and pairwise $p(\ell_i, \ell_j \mid \mathcal{T}_i, \mathcal{T}_j)$ terms into our original objective (3.4) and specify the procedure to obtain the CRF hyperparameters in Section 3.5.

3.3 Segmentation

We base the construction of segment candidates on the recently introduced Constrained Planar Cuts (CPC) method (Schoeler et al., 2015) discussed in Section 2.1. While in preliminary experiments the CPC algorithm showed good results, we also observed some shortcomings of the post-processing step. In the original CPC algorithm small segments that resulted from multiple locally concentrated cuts are merged to larger neighbours. The problem with this approach can be illustrated with a simple example shown in Figure 3.3a. Consider an imaginary profile segmented with cuts (1) and (2) into parts **A**, **B** and **C** such that $|\mathbf{B}| < |\mathbf{C}| < |\mathbf{A}|$, where $|\cdot|$ is a segment size measure (e.g. number of points, segment area, etc.). If segment **B** is small enough to be merged, the CPC algorithm will assign it to segment **A** since $|\mathbf{A}| > |\mathbf{C}|$. However, cut (1) exhibits a more pronounced concavity than cut (2) and, hence, merging **B** with **C** will be more visually cohesive.

To mitigate this issue, we refined the original algorithm and devised a replacement for the merging scheme. The resulting procedure is summarised as Algorithm 2. We keep close track on the adjacency of the initially generated supervoxels. First, the edges bisected in the cutting procedure are removed from the graph to find the connected components using the depth-first search in $O(V + E)$ time. Next, we construct a new graph where each node corresponds to a connected component and the edges obtain a score averaged over all edges cut between the subgraphs. Finally, the merging algorithm sequentially considers each edge in the ascending order of the respective scores and merges two segments if either the edge score is lower than a threshold or either of the connected segments is considered "small". The larger segment receives neighbours of the smaller one and a new weighted average of the edge score is computed for already connected neighbours. Since one such iteration is guaranteed to remove at least one edge and the sorting of the edges by their score takes $O(E \log E)$, the running time of this modification is $O(E^2 \log E)$. In practice, however, the number of edges is comparatively small (5–30) which does not lead to a noticeable overhead.

Algorithm 2: Modified CPC algorithm

```
Data: Point cloud Cloud
Result: Labels Labels
Initialise VoxelGraph from Cloud using adjacency-octree structure;
Construct EdgeCloud from VoxelGraph;
EdgesCut ← ∅;
repeat
    NoCutFound ← true;
    Inliers ← WeightedRANSAC(EdgeCloud, MaxIterations);
    if Score(Inliers) > ScoreThreshold then
        NoCutFound ← false;
        EdgesCut ← EdgesCut ∪ Inliers;
        EdgeCloud ← EdgeCloud \ Inliers;
    end
until NoCutFound;
/* Segment Merging                                                    */
VoxelClusters ← FindConnectedComponents(VoxelGraph \ EdgesCut);
Initialise EdgeQueue from VoxelClusters and EdgesCut ;   // See text for
                                                         // details
while EdgeQueue ≠ ∅ do
    (V1, V2) ← EdgeQueue.pop();
    if Score(V1, V2) < ScoreThreshold or
      |V1| < SizeThreshold or |V2| < SizeThreshold then
        MergeNodes (V1, V2) ;                  // See text for details
        update EdgeQueue;
    end
end
Labels ← NearestNeighbourSearch(Cloud, VoxelGraph, VoxelClusters)
```

Some qualitative results of our modification can be seen in Figure 3.3b and Figure 3.3c. The original issue with merging manifested itself through a number of different ways. In the airplane model shown in Figure 3.3b, a cut found with RANSAC initially separated the wing and the tailplane from the fuselage while also cutting off some boundary points of the latter. Subsequent cuts created individual segments from the tailplane and the wing, but the cut-off segment on the fuselage was not subsequently merged due to a sufficient size. In contrast to this result, the condition added in our modification verified that the region between the fuselage and the segment is convex and merged the two. The same explanation applies to the breakaway segment on the wing. For a more complex model such as that of a human shown in Figure 3.3c the modification might have more dramatic

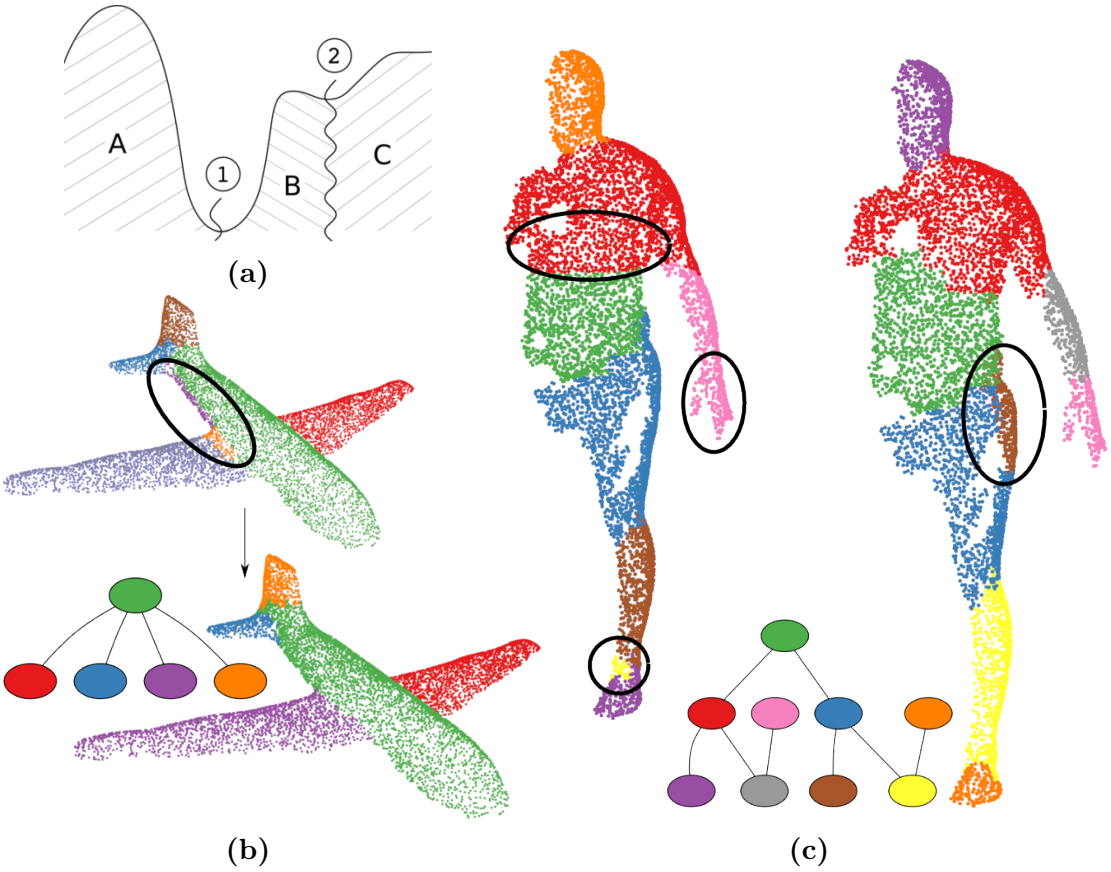


Figure 3.3: Our modification of the CPC segmentation: **(a)** illustration of the problem; **(b)**, **(c)** qualitative comparison between the modified and the original CPC algorithm on the airplane and human models. (Best viewed in colour)

consequences. The subtle concavity which characterises the chest protrusion was partially ignored by the original algorithm which merged small fractions of the solar plexus to the larger "shoulder" segment. Similarly, segment fractions on the hand cut multiple times due to its convoluted structure were merged to the arm. The small segment on the foot, by contrast, happened to be large enough to avoid the merging. These limitations were overcome by our modification with a benign side-effect: identifying a new hip segment which was initially fractured with cuts owing to its narrow size and the resulting distortion of the concavity estimates. Still, our merging appeared to yield a more consistent segmentation throughout qualitative experiments and we leave a more extensive, quantitative comparison with the original method for future work.

In addition to the aforementioned advantages, the new algorithm allows us to maintain a segmentation graph that is crucial to impose topological constraints in our optimisation problem. For instance, the two wings and the tailplane in Fig-

ure 3.3b cannot be neighbours in the segmentation graph. Similarly, there can be no edges connecting the torso segment with a leg on the human model (Figure 3.3c). We incorporate such constraints in the pairwise term of the objective (3.4) as a large penalty value.

Note that the segmentation step is applied only to the query model, since the reference shape is already provided with the target segments. We use these segments to model the shape representation covered next.

3.4 Shape learning

In this section, we show how we extract features and encode them into a feature descriptor. As per our objective (3.4) we distinguish between shape appearance terms and the terms encoding intrinsic spatial relations between shape parts. First, however, we scan the reference shape in order to create viewpoint-sensitive data for feature extraction. We describe this step next.

3.4.1 Object scanning

In real-world scenarios objects can be observed only partially from the view angle of the sensor. We regard this as an important hint and incorporate object scanning into a pre-processing stage of our pipeline since our reference shape is a known CAD model. By analogy, this step simply imitates real object scanning usually performed prior to feature extraction.

We use a virtual scanner placed in a grid of viewpoints to create a subset of partial clouds from the provided reference shape. Each labelled part of the partial cloud extracted with scanning is seen to represent a particular variation of this part within the complete shape. Accumulation of this variation over different viewpoints provides raw data for the shape learning step.

In this work, we investigate applicability of Bag-of-Words (BoW) and Fisher vectors (FV) to 3D part-based shape representation. We outline the details of BoW and FV representations in the following sections and refer the reader to sections Section 2.3 and Section 2.4 for a more general overview of these methods.

3.4.2 Shape learning with Bag-of-Words (BoW)

A number of issues have to be addressed prior to feature encoding. First, we can expect some degree of over-segmentation of the query shape, i.e. the pre-segmentation step does not necessarily divide the shape into its functionally mean-

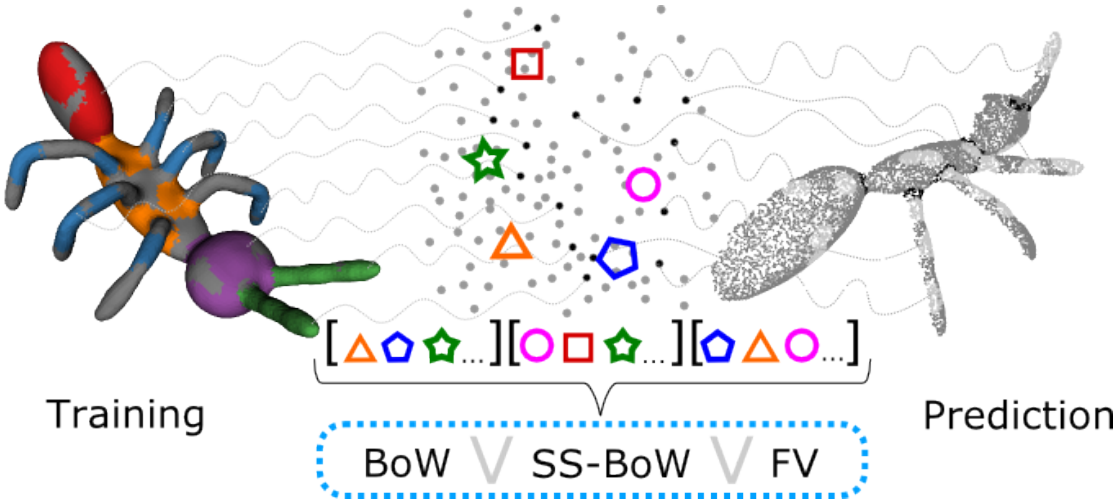


Figure 3.4: Illustration of the feature extraction with subsequent encoding

ingful components. For these reasons, instead of modelling complete subparts of the reference shape we need to reduce the support size of the feature encoding.

Second, we have to consider the difference in relative size of shape subparts. The palm of a hand, for example, has a larger surface area then each finger. We argue that encoding of all available feature vectors extracted from an individual segment might result in misrepresentation of the segment's geometry. One can draw a parallel with text analysis. A news article featuring a new science-fiction film might contain words like "dark energy", "plasma" or "radiation", and a travel agency's advert might include "sun", "stars", "water". Yet despite the apparent difference of the text genre, all of this vocabulary can be subsumed by an astrophysics paper, not least because of a larger volume. This suggests that a discriminative feature encoding should be constructed from an (approximately) equal number of feature descriptors.

Third, since we cannot rely on complete visibility of the shape, features extracted from shape patches in one view might be unavailable in another view. Unfortunately, characterisation of self-occlusion in partial views is largely dependant on the unknown shape geometry itself and, hence, hard to quantify.

In an attempt to address these issues, we propose the following procedure based on random sampling. For each view v and shape part with label ℓ, we extract a set of point clusters $\mathcal{P}_{\ell,v}$ whose centres are uniformly sampled. From each set of point clusters we draw an equal number of randomly sampled fixed-sized subsets $\mathcal{P}_{\ell,v,i} \subset \mathcal{P}_{\ell,v}$. We will call $\mathcal{P}_{\ell,v,i}$ a "feature packet" for short.

The general procedure of the feature encoding is illustrated in Figure 3.4. Let $M_{\ell,v} = \{\mathbf{m}_{\ell,v,t} \mid \mathbf{m}_{\ell,v,t} \in \mathbb{R}^D, \forall t = 1, ..., T\}$ denote the set of T shape feature

vectors with label $\ell \in L$ visible from view angle $v \in V$. We model the union of the feature vectors over all labels and views $\bigcup_{\ell \in L, v \in V} M_{\ell,v}$ by the Gaussian mixture model (GMM):

$$p(\mathbf{m}_{\ell,v,t}) = \sum_{i=1}^{K} w_i \mathcal{N}(\mathbf{m}_{\ell,v,t} | \boldsymbol{\mu}_i, \Sigma_i), \tag{3.5}$$

where $\mathcal{N}(\mathbf{m}_{\ell,v,t} | \boldsymbol{\mu}_i, \Sigma_i)$ is a multinomial normal distribution with the mean $\boldsymbol{\mu}_i$ and the diagonal covariance matrix Σ_i.

A vocabulary representation $\mathbf{f}_{BoW}(p_{\ell,v}) \in \mathbb{R}^K$ for each cluster in the feature packet $\rho_{\ell,v,i} \in \mathcal{P}_{\ell,v,i}$ can be constructed as:

$$f^{(k)}_{\mathrm{BoW}}(\rho_{\ell,v,i}) = \frac{w_k}{|\rho_{\ell,v,i}|} \sum_{t} \mathcal{N}(\mathbf{m}_{\ell,v,t} | \boldsymbol{\mu}_k, \Sigma_k), \tag{3.6}$$

where $\mathbf{m}_{\ell,v,t} \in \rho_{\ell,v,i}$, $|\rho_{\ell,v,i}|$ is the number of low-level feature descriptors and $f^{(k)}_{\mathrm{BoW}}(\cdot)$ is the kth dimension of vector $\mathbf{f}_{\mathrm{BoW}} \in \mathbb{R}^K$.

We vectorise each feature packet by taking the average over the clusters it contains:

$$f_{\mathrm{BoW}}(\mathcal{P}_{\ell,v,i}) = \frac{1}{|\mathcal{P}_{\ell,v,i}|} \sum_{\rho_{\ell,v,i}} f_{\mathrm{BoW}}(\rho_{\ell,v,i}), \quad \rho_{\ell,v,i} \in \mathcal{P}_{\ell,v,i} \tag{3.7}$$

As the underlying feature we use SHOT descriptors discussed in Section 2.2.1.

3.4.3 Shape learning with Fisher vectors (FV)

We use the same routine of fitting the GMM to the data as in the BoW to construct the Fisher vectors. Our arguments favouring random sampling of the feature packets are still valid for the Fisher vector representation and we apply them here as well. Using the Equation (2.7) the gradients are computed for each point cluster $\rho_{\ell,v,i}$ of the feature packet $\mathcal{P}_{\ell,v,i}$ extracted from a segment with label ℓ in view v:

$$\begin{aligned} G_{\boldsymbol{\mu}_k}(\rho_{\ell,v,i}) :=& \frac{\partial \log p(\rho_{\ell,v,i} | \lambda)}{\partial \boldsymbol{\mu}_k} \\ =& \frac{1}{|\rho_{\ell,v,i}| \sqrt{\omega_k}} \sum_{t=1}^{|\rho_{\ell,v,i}|} \gamma_{\ell,v,t}(k) \left(\frac{\mathbf{m}_{\ell,v,t} - \boldsymbol{\mu}_k}{\sigma_k} \right), \end{aligned} \tag{3.8}$$

$$
\begin{aligned}
G_{\boldsymbol{\sigma}_k}(\rho_{\ell,v,i}) :=& \frac{\partial \log p(\rho_{\ell,v,i} \mid \lambda)}{\partial \boldsymbol{\sigma}_k} \\
=& \frac{1}{|\rho_{\ell,v,i}|\sqrt{2\omega_k}} \sum_{t=1}^{|\rho_{\ell,v,i}|} \gamma_{\ell,v,t}(k) \left(\frac{(\mathbf{m}_{\ell,v,t} - \boldsymbol{\mu}_k)^2}{\sigma_k^2} - 1 \right),
\end{aligned}
\tag{3.9}
$$

where vector division is element-wise, $\sigma_i := \mathbf{diag}(\Sigma_i)$ and

$$
\gamma_{\ell,v,t}(k) = \frac{\omega_k u_k(\mathbf{m}_{\ell,v,t})}{\sum_{j=1}^{K} \omega_j u_j(\mathbf{m}_{\ell,v,t})}, \quad \mathbf{m}_{\ell,v,t} \in \rho_{\ell,v,i} \tag{3.10}
$$

is the soft assignment of descriptor $\mathbf{m}_{\ell,v,t}$ to the Gaussian centre k. The Fisher vector is formed by concatenating the gradients (3.8) and (3.9) of each Gaussian centre:

$$
\mathbf{f}_{\mathrm{FV}}(\rho_{\ell,v,i}) = (G^T_{\boldsymbol{\mu}_1}(\rho_{\ell,v,i}), ..., G^T_{\boldsymbol{\mu}_K}(\rho_{\ell,v,i}), G^T_{\boldsymbol{\sigma}_1}(\rho_{\ell,v,i}), ..., G^T_{\boldsymbol{\sigma}_K}(\rho_{\ell,v,i}))^T. \tag{3.11}
$$

The sparsity of the Fisher vectors $\mathbf{f}_{FV}(\rho_{\ell,v}) \in \mathbb{R}^{2KD}$ becomes apparent as the number of Gaussians grows: feature vectors will tend to have a hard-assignment (i.e. $\gamma_{\ell,v,t}(i) \approx 1$) potentially resulting in empty Gaussians with the gradients (3.8) and (3.9) close to null. In this scenario, the commonly used L2-distance might limit the descriptive properties of the Fisher vector representation. Perronnin et al. (2010) proposed power normalisation to uniformly rescale the vector by applying the following function element-wise:

$$
f(z) = sign(z)|z|^{\alpha} \tag{3.12}
$$

We adopt this normalisation here. By contrast, the use of the L2-normalisation proposed in the same work is not well motivated for our problem and we omit it. By our assumption, all points of the shape are potentially relevant (i.e. there is no background to neglect).

In order to keep the size of the Fisher vector moderate and not resort to PCA a the risk of loosing potential information, we use FV in conjunction with the compact FPFH feature descriptors reviewed in Section 2.2.2.

3.4.4 Classifier training

The computed representation vectors, $\mathbf{f}_X(\cdot)$ (where X stands for the type of the feature encoding used: BoW, SS-BoW or FV) along with the corresponding labels $\{(\mathbf{f}_X(\cdot), \ell) \mid \rho_{\ell,v,i} \in \mathcal{P}_{\ell,v,i}, \ell \in L, v \in V\}_{i=1,...,I}$ form the training dataset of the shape appearance model. We use the Support Vector Machine (SVM) with an

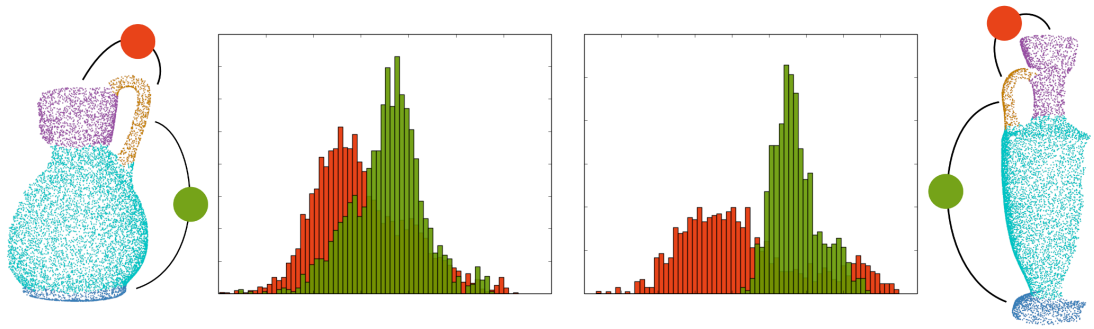


Figure 3.5: Comparison of spectral distances between two pairs of segments

RBF kernel for BoW and SS-BoW vectors, whereas a linear SVM is used for Fisher vectors.

Similarly to the feature extraction from the reference shape, we extract the same number of feature packets on the query shape $\mathcal{T}$ from the segments $\{\mathcal{T}_j \mid \mathcal{T}_j \subset \mathcal{T}\}$ generated in the pre-segmentation step. The scores obtained from predictions of the individual feature packets are averaged over complete segments and the result of the prediction of the label assignment ℓ_i to the segment $\mathcal{T}_j$ is naturally interpreted as $p(\ell_i \mid \mathcal{T}_j)$.

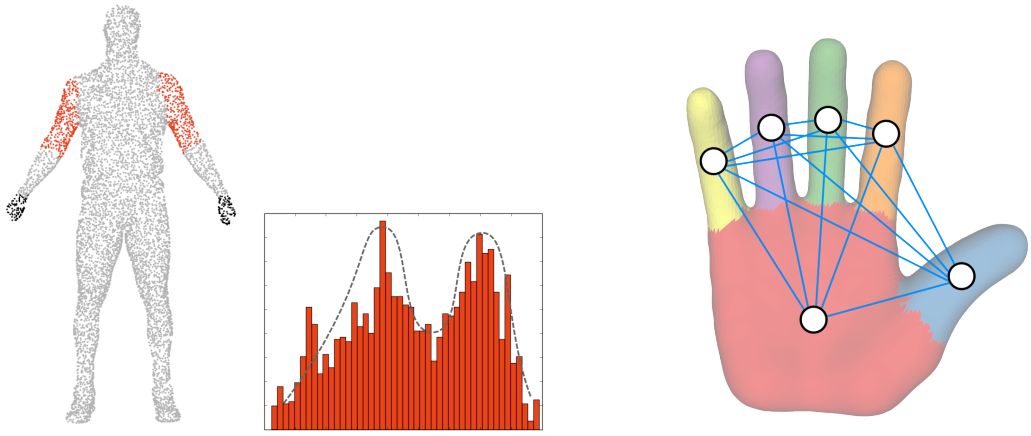
3.4.5 Diffusion distance

In this section, we construct the pairwise term $p(\ell_i, \ell_j \mid \mathcal{T}_i, \mathcal{T}_j)$ from our objective (3.4) based on the spectral distances discussed in Section 2.5.2.

We recall that the distribution of the diffusion and commute time distances can be used as a measure of shape similarity (M. M. Bronstein and A. M. Bronstein, 2010). Intuitively, we could also apply the same principle to pairs of shapes, or as in the context of our problem, to segments of the same shape. More concretely, we can extract diffusion distances between two point sets we want to analyse and compare the resulting distribution to that derived from another pair of point sets.

We illustrate the information contained in the segment distances with an example of two partially visible vases shown in Figure 3.5. The distribution of the commute time distances between pairs of points on the base and the handle (green) and on the handle and the neck (red) are shown in the histograms next to the corresponding vase. Despite shape discrepancy, the histograms still capture the key features of the shape topology: the base is "farther away" to the handle than the neck. Furthermore, the histograms are highly suggestive of a normal distribution for the underlying model where each distance $d_{CT}(s_i, s_j)$ between points of two segments $s_i \in \mathcal{S}_i$, $s_j \in \mathcal{S}_j$ can be seen as sampled from $\mathcal{N}(\mu_{ij}, \sigma_{i,j})$.

Unfortunately, partial views and shapes with symmetries can introduce ambigu-



(a) (b)

Figure 3.6: (a) Cumulative distribution of distances between the hand and the shoulder of a human model (b) An example of the learned graphical model

ity for a single Gaussian model. Somewhat expectedly though, multiple discrepant segments with the same label induce a multinomial distribution as can be seen from a human model shown in Figure 3.6a. Hence, we can still attempt to approximate the distribution with a multinomial Gaussian model.

In view of this observation, we propose to learn a multinomial distribution of distances extracted from the corresponding pairs. Let $D_{CT}(\ell_i, \ell_j) := \{d_{CT}(s_i, s_j)\}$ denote the set of commute time distances between segments with labels i and j. Note, that by construction $D_{CT}(\ell_i, \ell_j) = D_{CT}(\ell_j, \ell_i)$ and we allow $i = j$ since the distance distribution is also informative within a single segment. We fit the GMM to obtain a maximum likelihood estimate $\boldsymbol{\theta}^{\star}_{ij} := \arg\max_{\boldsymbol{\theta}_{ij}} \sum_k \omega_k^{ij} \mathcal{N}(\mu_k^{ij}, D_{CT}(\ell_i, \ell_j) \mid \sigma_k^{ij})$ with parameters $\boldsymbol{\theta}_{ij} = \{(\mu_k^{ij}, \sigma_k^{ij}, \omega_k)\}_k$.

With some abuse of notation, let $D_{CT}(\mathcal{T}_i, \mathcal{T}_j) := \{(d_{CT}(t_{in}, t_{jn})) \mid t_{in} \in \mathcal{T}_i, t_{jn} \in \mathcal{T}_j, n = 1, ..., |\mathcal{T}_i \times \mathcal{T}_j|\}$ be the set of all distances extracted between points on the two segments $\mathcal{T}_i$ and $\mathcal{T}_j$ of the query shape $\mathcal{T}$. Denoting by $\ell_{i\sim i'}$ the assignment of label i' to segment $\mathcal{T}_i$, we can compute the likelihood estimate of the data given any pairwise assignment as follows:

$$p\big(D_{CT}(\mathcal{T}_i, \mathcal{T}_j) \mid \ell_{i\sim i'}, \ell_{j\sim j'}\big) = \sum_n \sum_k \omega_k^{i'j'} \mathcal{N}(\mu_k^{i'j'}, \sigma_k^{i'j'} \mid d_{CT}(t_{in}, t_{jn})) \tag{3.13}$$

Similar to the single labels we assume that any given pairwise assignment is equiprobable. we can compute an estimate of assignment probability using the

Bayes rule:

$$p\big(\ell_{i\sim i'}, \ell_{j\sim j'} \mid D_{CT}(\mathcal{T}_i, \mathcal{T}_j)\big) = \frac{p\big(D_{CT}(\mathcal{T}_i, \mathcal{T}_j) \mid \ell_{i\sim i'}, \ell_{j\sim j'}\big)}{\sum_{i'',j''} p\big(D_{CT}(\mathcal{T}_i, \mathcal{T}_j) \mid \ell_{i\sim i''}, \ell_{j\sim j''}\big)} \tag{3.14}$$

Finally, we let $p(\ell_i, \ell_j \mid \mathcal{T}_i, \mathcal{T}_j) := p\big(\ell_{i\sim i'}, \ell_{j\sim j'} \mid D_{CT}(\mathcal{T}_i, \mathcal{T}_j)\big)$ define our distance measure between the two segments in our objective (3.4).

3.5 Inference

Our model effectively results in a small to medium-sized complete graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, an example of which is demonstrated in Figure 3.6b. Each unary potential $\theta_{i;\ell_i} := -\log p(\ell_i \mid \mathcal{T}_i)$ of the node $i \in \mathcal{V}$ models the shape appearance of the subpart, and the pairwise term $\theta_{ij;\ell_i,\ell_j} := -\log p(\ell_i, \ell_i \mid \mathcal{T}_i, \mathcal{T}_j)$ measures the isometric distortion between each pair of the subparts. Hence, our objective can be solved by the inference of a second-order CRF with the energy given by

$$J(\mathcal{T}) = \sum_i \theta_{i;\ell_i} + \sum_{i,j} \theta_{ij;\ell_i,\ell_j}. \tag{3.15}$$

One issue left for consideration is the trade-off between the unary and pairwise terms. Clearly, unary features can be relied more upon where structure variations dominate shape appearance. For example, legs and the surface of a table should be well-recognisable whereas partial views may convolute the overall structure as perceived through diffusion distance. Likewise, complex models with rich structures, such as animals and human, may benefit from additional structural information. For this reason, we incorporate a trade-off parameter λ in the pairwise term and use $\mathcal{N}(\mu_k^{i'j'}, (\lambda+1)\sigma_k^{i'j'} \mid d_{CT}(t_{in}, t_{jn}))$ in Equation (3.13) instead of the original normal distribution with deviation $\sigma_k^{i'j'}$. Intuitively, large values of λ will "dampen" the effect of the pairwise term while $\lambda = 0$ would leave the distribution unchanged.

To learn the parameter λ, we could attempt to weigh up the expected benefits (or a detrimental effect) of the pairwise term by performing the pre-segmentation on the reference shape. We use a grid search of λ in the range starting from zero ("no change") to a small positive value by running our co-segmentation pipeline on the resulting segments. Note that since the new segmentation might not be identical to the original one and there is a random factor in sampling of the feature descriptors we do not expect a perfect accuracy in segment classification and, hence, some tangible variation of the effect λ has on the overall solution is reasonable to anticipate. For the query shape, we naturally use the value of λ that

returned the best accuracy on the reference shape.

We observe, that our compact model is not dissimilar to the one used by Bergtholdt et al. (2010) in the context of object part detection. One of their main results was a competitive comparison of various established inference techniques, such as the (Loopy) Belief Propagation (Yedidia et al., 2005) and the Tree Reweighted Belief Propagation (Wainwright et al., 2005) with the ground-truth computed using the A^* search. The outcome of the comparison was that for small graphs, A^*-based inference often outperformed the other algorithms in the runtime as well. We could also confirm a better performance of the algorithm in our preliminary tests. In fact, for simple models such as vase and cups, the solution could be found even with a brute-force search in reasonable time. For more complex models, however, A^* provided a faster solution with strong optimality guarantees. This motivated our choice of the $A^\star$ for the inference in our model. For an overview of this algorithm we refer the reader to Section 2.6.

3.6 Implementation details

For initialisation of the Gaussian mixture model we used Gonzalez's algorithm (Gonzalez, 1985) with a fixed number of centres $K = 128$. We used SVM implementation provided by `libsvm` library (Chang and Lin, 2011). Both BoW and Fisher vector representations relied on SHOT as a low-level feature descriptors implemented in the PCL library (Rusu and Cousins, 2011). The feature clusters were obtained from the support size of $|\rho_{\ell,v}| = 128$ points which constituted $\approx 1.5\%$ of each shape surface area. For computation of pairwise distances, we also used a sparse grid to sample a compact subset of points from each segment. For the inference in the mode with the A^* algorithm the node order and the spanning tree were computed randomly. We remark, however, that although a better solution could have been reached with a more meaningful choice, the energy values computed were still lower than those attained by other algorithms, such as TRW-S (Kolmogorov, 2006).

4 Evaluation

We have performed an extensive evaluation of our approach and compared its performance to the state-of-the-art methods in two experiments. Experiment I is based on a subset of the Labelled PSB dataset (Kalogerakis et al., 2010). We selected 15 out 20 representative categories derived from the Princeton Object Segmentation Benchmark (X. Chen et al., 2009). Each category contained 20 object meshes with the ground-truth segmentation. In Experiment II we used point cloud data of two watering cans recorded with an ASUS Xtion sensor. The goal of this experiment is to demonstrate applicability of our approach to the real-world data and compare its efficiency to the state-of-the-art.

In the next section, we describe our experimental setting for Experiment I and the evaluation criteria used for benchmarking. The results are presented in the section afterwards. We introduce Experiment II and analyse the results in Section 4.4. In the last section, we summarise the overall performance of our method based on the two experiments.

4.1 Experimental setup: I

For each selected category in the Labelled PSB dataset (Kalogerakis et al., 2010) we generated a dataset of *valid* random views. In the context of this work, a random view is considered valid if at least 20% of each shape part is visible. This requirement is motivated by the desire to retain object diversity (segmentation results would be only misleading if the dataset contained trivial cases with a single label, or the number of labels was less than in the original model). To generate random views for each shape we created a uniform grid of view points as in Figure 4.1a. Next, we proved each view point for the validity criterion defined above (green points in Figure 4.1a). In order to create a rich subset of random views, we selected randomly only 8 viewpoints with the highest estimate of mutual scatter (in Figure 4.1a discarded viewpoints are red). The flowchart of this process is shown in Figure (4.1b).

We performed a comparison of our approach with two baseline methods derived from the state-of-the-art co-segmentation developed by Kalogerakis et al. (2010)

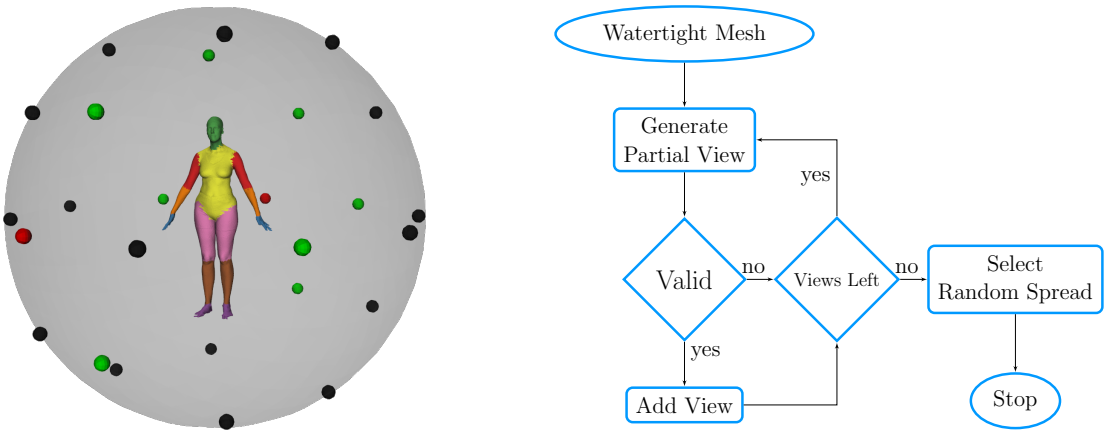


Figure 4.1: Generation of partial clouds: **(a)** a uniform grid on the sphere; **(b)** the flowchart of the process (see text for details).

and Kaick et al. (2011). Recall that the number of shapes allocated for training in Kalogerakis's work ranged from 3 to 19 out of total 20 in a category, and the training set of van Kaick constituted 60% of the dataset. In line with our problem formulation, we provided only one reference shape for training.

Random views necessitate further alteration to the choice of shape descriptors used in the original methods. Concretely, volumetric feature descriptors, such as shape diameter (Shapira et al., 2008) and distances from medial surface points (R. Liu et al., 2009) cannot be applied to partial views. Likewise, the Average Geodesic Distance (Hilaga et al., 2001) relies on holistic shape representation and will only be misleading in a partial setting. Hence, these feature descriptors were omitted in the implementation of the original methods.

For computation of curvature features used by Kalogerakis et al. (2010) and Kaick et al. (2011) we used bivariate polynomial fitting and estimated the principal curvatures according to (Cazals and Pouget, 2005). In implementation of the approach by Kalogerakis et al. (2010) the context features were restricted to a local scale with maximum distance to the neighbour 30% of the shape diameter. The pairs of features used for training the inter-mesh term in implementation of van Kaick's approach were extracted from the same shape.

We sampled the points with uniform density from the surface of the generated mesh parts to create point clouds for our approach. In order to make our results comparable with those obtained from meshes, we projected the estimated labels of points to the mesh surface and selected the dominant label for each face.

4.2 Evaluation criteria

For benchmarking our results, we adopted the evaluation metrics proposed by X. Chen et al. (2009), namely the Hamming distance, Rand index, the Global and Local Consistency Error (GCE and LCE, respectively) and accuracy. Another measure proposed by X. Chen et al. (2009), the Cut Discrepancy, was not used in the comparison, due to ill-defined non-informative notion of cuts and geodesic distances between them in partial views of point clouds. Since we projected the labels of the points back to the mesh surface, we used the standard measure on segment $\|\mathcal{X}\|$ equal to its area.

Next, we give a brief overview of the above-mentioned evaluation metrics used in the experiments.

Intuitively, the **Hamming Distance** is a sum of a set difference between pairs of corresponding segments. Let $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_n\}$ and $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_m\}$ denote the results of segmentation of two shapes. Let $\mathcal{S}_i \sim \mathcal{T}_j$ iff $i = \arg\max_k \|\mathcal{S}_k \cap \mathcal{T}_j\|$. The Directional Hamming Distance is defined as $D_H(\mathcal{S} \Rightarrow \mathcal{T}) := \sum_{\mathcal{S}_i \sim \mathcal{T}_j} \|\mathcal{T}_j \setminus \mathcal{S}_i\|$. Considering $\mathcal{T}$ to be the ground truth, the missing rate R_m and false alarm rate R_f are defined as follows:

$$R_m(\mathcal{S}, \mathcal{T}) = \frac{D_H(\mathcal{S} \Rightarrow \mathcal{T})}{\|\mathcal{T}\|} \quad R_f(\mathcal{S}, \mathcal{T}) = \frac{D_H(\mathcal{T} \Rightarrow \mathcal{S})}{\|\mathcal{S}\|}, \tag{4.1}$$

where the measure $\|\mathcal{A}\|$ is applied to a collection $\mathcal{A}$ in a natural way: $\|\mathcal{A}\| = \sum_i \|\mathcal{A}_i\|$, $\mathcal{A}_i \in \mathcal{A}$. Note also that in our case $\|\mathcal{T}\| = \|\mathcal{S}\|$. The Hamming Distance is the average of the missing rate and false alarm rate.

Rand index Rand, 1971 corresponds to the likelihood that a pair of faces (points) is either in the same or different segments in two segmentations. It is computed by first counting

- the number of pairs of faces a in the same segment;
- the number of pairs of faces b in different segments.

The final score for a shape of size N is computed as $R = \binom{N}{2}^{-1}(a + b)$.

Consistency Error measures region-based difference between two segmentation results $\mathcal{S}$ and $\mathcal{T}$. Let $\mathcal{S}_i$ denote a segment in $\mathcal{S}$ containing face i (equivalently for segmentation $\mathcal{T}$). The local refinement error is defined by $E_i(\mathcal{S}, \mathcal{T}) := \frac{\|s_i \setminus \mathcal{T}_i\|}{\|\mathcal{S}_i\|}$. For a shape with N faces, the global and local versions of the error measure, GCE

and LCE, are defined as:

$$\begin{aligned} GCE(\mathcal{S},\mathcal{T}) &= \frac{1}{N}\texttt{min}\Big\{\sum_i E_i(\mathcal{S},T), \sum_i E_i(\mathcal{T},\mathcal{S})\Big\} \\ LCE(\mathcal{S},\mathcal{T}) &= \frac{1}{N}\sum_i \texttt{min}\big\{E_i(\mathcal{S},T), E_i(\mathcal{T},\mathcal{S})\big\}. \end{aligned} \tag{4.2}$$

Finally, given the ground truth segmentation, **accuracy** is naturally defined as the ratio of correctly labelled faces to the total number of faces.

In addition to these standard metrics, we also report average recall of shape parts. This metric can be especially useful for shapes with only a few subparts, or shapes with a dominant part. The body of the vase, for example, or the torso of the human model might make up a significant share of the total surface area. Predictors that simply return the dominant label on every query could be falsely regarded as well performing.

4.3 Results: I

For each selected category of shapes, we consecutively chose one object as the reference shape and ran the co-segmentation pipeline against all *compatible* partial shapes in the category. We refer to a query shape as compatible if it doesn't contain labels not present on the reference shape. Hence, with 20 objects in the category, we run at most $20 \times 8 \times 20 = 3200$ co-segmentation instances for each object group. The results are averaged for each category. Comparison results of the reference object with its own partial views were excluded from quantitative results.

In the first part of Experiment I we compared four configurations of our approach with the state-of-the-art using 15 categories of the Labelled PSB dataset. Each configuration is built on either BoW or Fisher vector representation, with or without pairwise features. To simplify notation, we use BoW for the plain Bag-of-Words classification and BoW+ISO for the Bag-of-Words with the isometry prior. The same notation applies to the Fisher Vectors referred to as FV for short. Recall also that the Bag-of-Words implementation relied on SHOT feature descriptors, whereas the Fisher vector used FPFH. The main results on accuracy are summarised in Figure 4.2.

As can be seen from the accuracy values, the plain Fisher vector classification performs best on average while BoW and FV+ISO already outperform the state-of-the-art. The state-of-the-art still shows higher accuracy on three categories: "Fish", "FourLeg" and "Octopus". We believe that these results stem from mediocre performance on the pre-segmentation step which fails to identify con-

Category	van Kaick et al.	Kalogerakis et al.	BoW	BoW+ISO	FV	FV+ISO
Ant	58.8	58.9	66.2	65.6	**77.7**	74.1
Airplane	62.7	62.0	59.2	57.0	**64.0**	60.0
Bird	58.1	57.0	57.4	52.0	**58.5**	53.6
Chair	59.6	59.6	**60.6**	56.7	60.2	55.5
Cup	81.6	81.8	**90.0**	87.6	88.7	87.5
Fish	84.2	**84.4**	72.1	71.7	78.4	77.7
Fourleg	**60.1**	59.4	51.1	48.1	54.9	50.6
Hand	52.2	52.7	53.4	46.8	**56.0**	49.6
Human	41.3	41.6	35.8	34.2	**43.7**	40.4
Mech	81.3	81.7	82.4	84.4	84.1	**84.6**
Octopus	82.0	**82.8**	76.5	75.0	69.6	69.8
Plier	33.7	32.5	70.5	57.3	**71.9**	58.8
Table	71.6	70.9	**88.9**	87.5	85.4	84.1
Teddy	71.9	71.1	64.5	69.4	76.4	**77.0**
Vase	64.3	65.5	**70.6**	65.3	70.3	63.8
Average	64.2	64.1	66.6	63.9	**69.3**	65.8

Figure 4.2: Average accuracy on the LPSB dataset used in Experiment I (in percent)

cave regions on partial views of smooth-surfaced ("Fish") and complex-structured ("FourLegt" and "Octopus") objects. Another observation from the obtain results is that the isometry prior seems to decrease the overall accuracy. One explanation might be high sensitivity of the Laplace-Beltrami eigenfunctions to partial views in the current implementation. Still, the use of the isometry prior in conjunction with Fisher vectors achieved best accuracy among all evaluated methods in two categories: "Teddy" and "Mech".

In the second part of the Experiment I we used the evaluation criteria reviewed in Section 4.2 to look at the quality of segmentation. We present the results for each category in Figure 4.3 with the last graph showing the overall average. Note that the rand index shown is actually subtracted from one by convention.

From the graphs in Figure 4.3 we observe that all our methods exhibit a sharp decrease of GCE and LCE. We explain this phenomenon by the "greedy" property of our approach: whereas the state-of-the-art methods tend to produce segmentations with many local inconsistencies, our tactics of "cutting and classifying" assigns labels to large segments.

The results of the evaluation criteria in Figure 4.3 also agree well with those in Figure 4.2: the configuration based on the Fisher vectors achieves best scores overall while the isometry prior generally seems to exacerbate the performance.

A selection of qualitative examples computed with FV from the Experiment I is shown in Figure 4.4. It can be seen, that our best-performing configuration fails to identify small segments, such as the tail of the airplane, of the centre of the pliers. In other instances, some parts are misclassified (e.g. the leg of a chair) which might also be the result of the pre-segmentation failing to separate them from the main body. Our approach also shows decent results on some challenging problems, such as the vase, table and octopus.

As a concluding remark of Experiment I, all co-segmentation approaches showed a particularly weak performance on the "Hand" and "Human" datasets. This categories are also most challenging, because they include variations of scale, shape appearance and isometric deformations.

4.4 Experiment II

In Experiment II we evaluated our best performing approach (FV) on real data and qualitatively compared its efficiency and accuracy with the state-of-the-art (Kaick et al., 2011).

We supplied both algorithms with a reference shape obtained from a sensor, and manually labelled segmentation. Since this data was initially obtained as

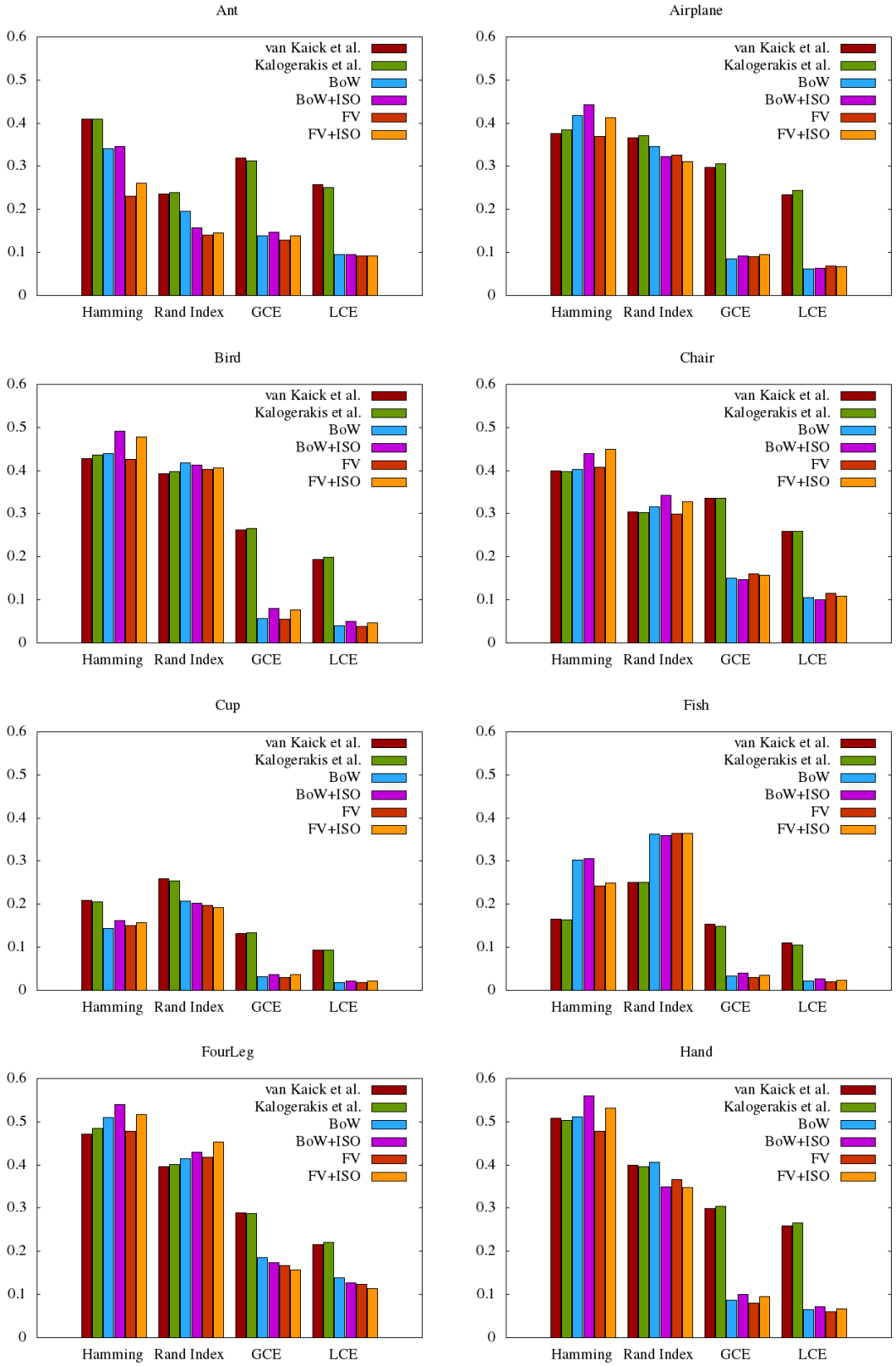


Figure 4.3: The average performance of different co-segmentation algorithms per category used in Experiment I (continue)

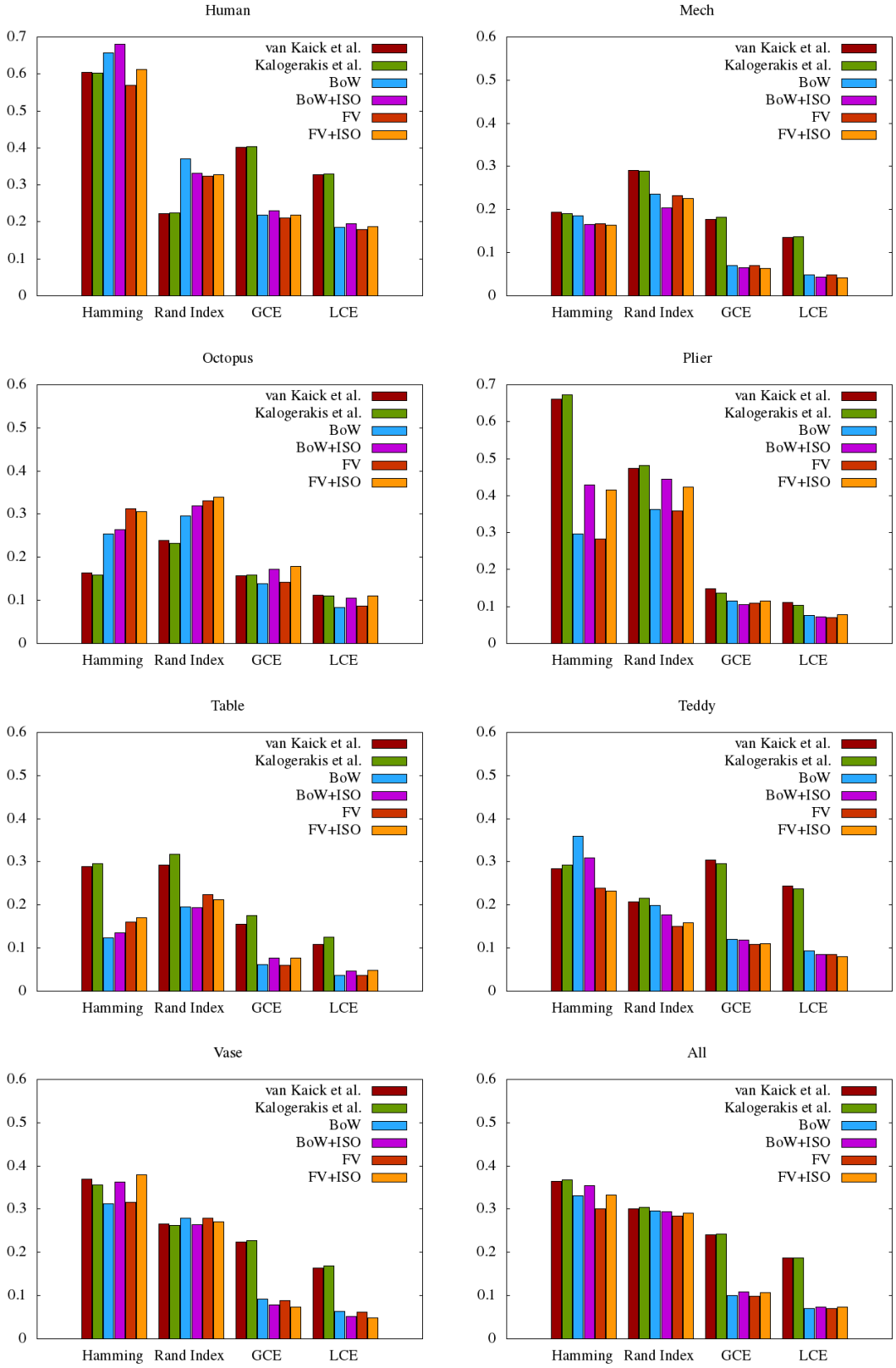


Figure 4.3 (cont.)**:** The average performance of different co-segmentation algorithms per category used in Experiment I

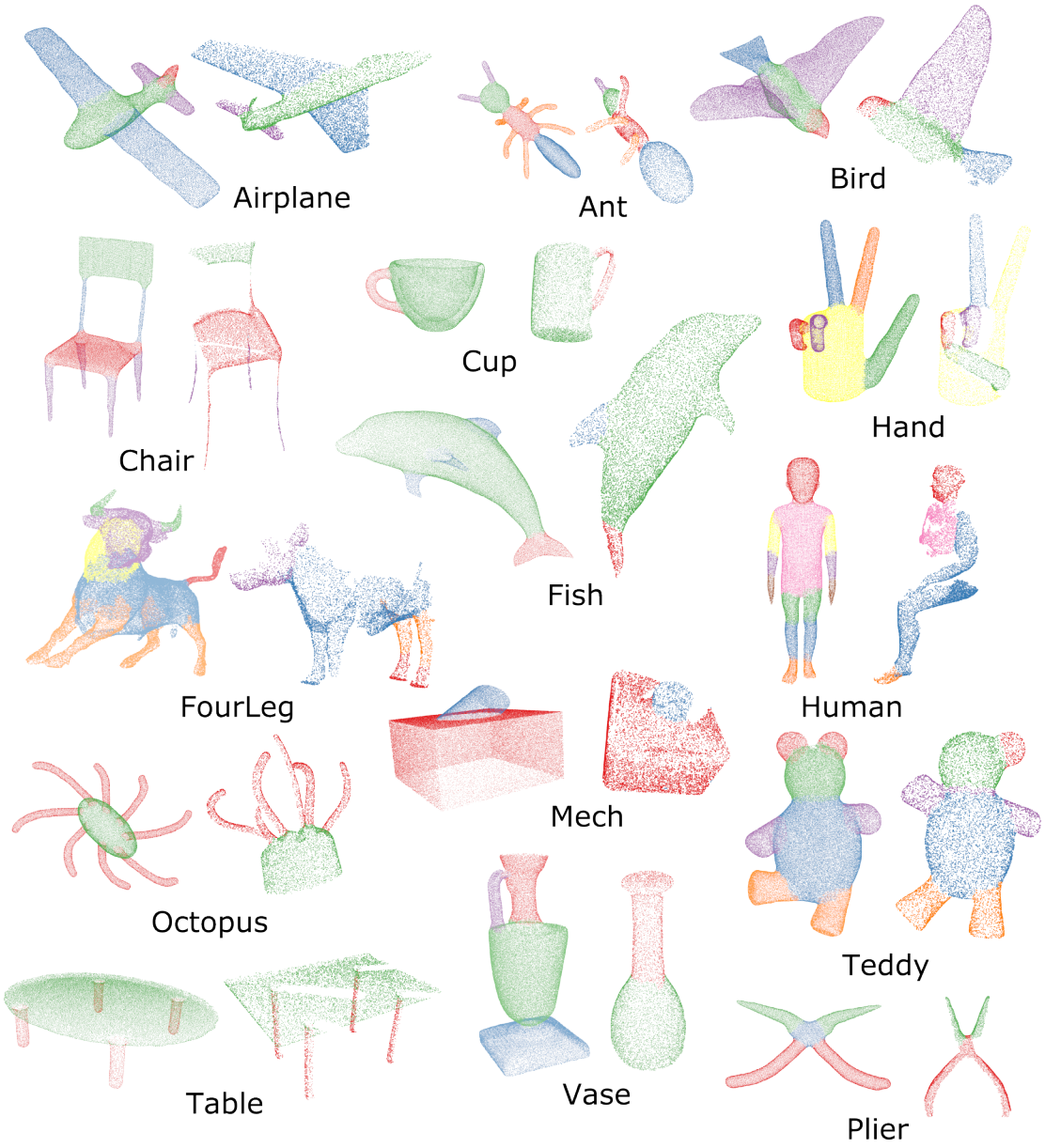


Figure 4.4: Selected results obtained from our co-segmentation approach. **Left:** Reference shape. **Right:** Query shape

(a) Reference shape **(b)** Query shapes

Figure 4.5: Two types of watering cans used in Experiment II

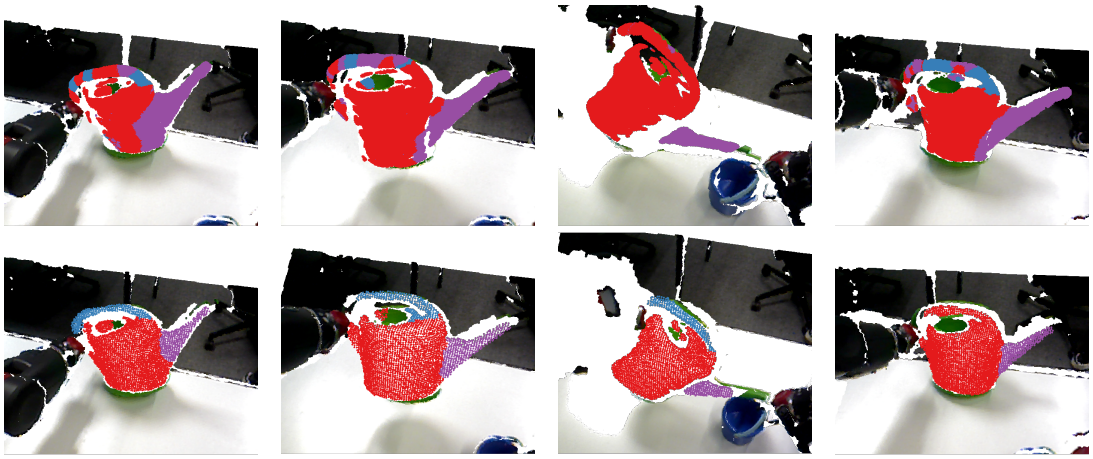
a point cloud, we computed a mesh model from a representative partial view using fast triangulation (Marton et al., 2009) to make it compatible with the mesh-based state-of-the-art. The computed mesh of the reference shape and the corresponding labelling is shown in Figure 4.5a. Our approach, however, does not require a complete model; it was sufficient to provide a small number of labelled partial-view point clouds which were provided using a separate train sequence of frames.

We proceed with the experiment by obtaining two frame sequences. The first is a recording of the original model (see Figure 4.5, left) but previously unseen action. A representative set of frames demonstrating the resulting segmentation of the evaluated algorithms is shown in Figure 4.6. We note that although our approach failed to detect the handle in frame 99, it still performed well in other sequences. By comparison, the state-of-the-art method (Kaick et al., 2011) identified only patches of the handle throughout the sequence.

In a more challenging experiment, we presented both algorithms with a sequence containing a novel object (see Figure 4.5, right). We used exactly the same reference shape as in the first part. As can be seen from Figure 4.7 our approach misclassified the spout as a handle in frame 2 and detected only part of the handle in frame 22. The state-of-the-art misclassified a large fraction of the container in the first two frames while showing relatively good segmentation in the last ones.

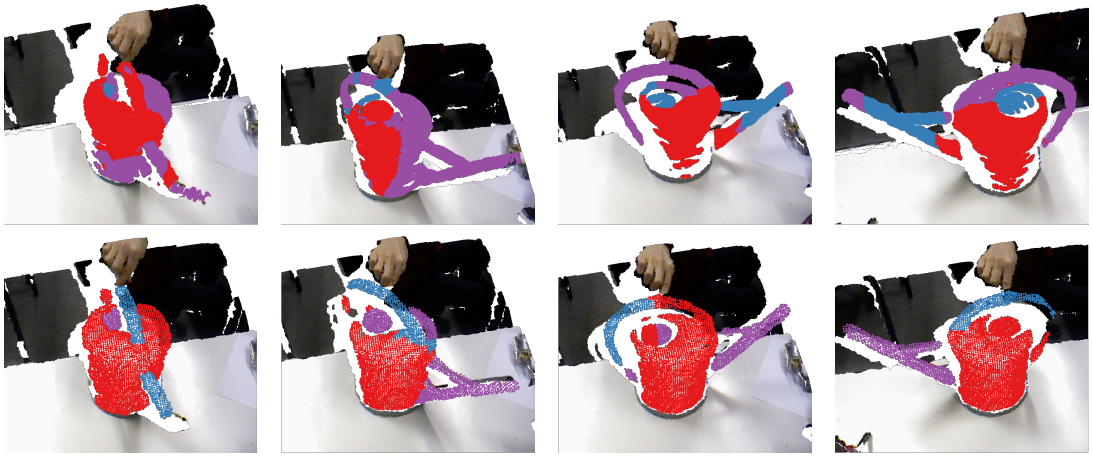
As a final part of Evaluation II we compared the computational time required for both methods. The ran the first part of the Experiment II on a laptop with Intel Core i7 CPU and 8GB RAM. The code was parallelised for point- and face-wise operations, such as computing the normals and curvatures.

The results of the benchmarking are summarised in Figure 4.8. While the training time of the state-of-the-art required both the classifier training and the learning of the CRF parameters, our method FV needs only the latter. Notwithstanding an additional pre-segmentation step, our approach was almost six times faster than



Frame 6 Frame 34 Frame 56 Frame 99

Figure 4.6: Test sequence with the same query shape as the reference. **Top row:** Kaick et al. (2011); **Bottom row:** Ours (FV).



Frame 2 Frame 17 Frame 22 Frame 28

Figure 4.7: Test sequence with a novel query shape. **Top row:** Kaick et al. (2011); **Bottom row:** Ours (FV).

	van Kaick et al.	FV
Training	259.6	581.0
Learning CRF	506.5	-
Total	766.1	**581.0**
Pre-segmentation	-	34.2
Inference	290.15	16.1
Total	290.15	**50.3**

Figure 4.8: Average time per object pair in Experiment II (in seconds)

the state-of-the-art implementation.

4.5 Summary

In this section, we performed quantitative and qualitative comparison of our method the state-of-the-art approaches. In the first experiment, we established that three out of four variations of the proposed algorithm showed higher average accuracy on a challenging Labelled PSB dataset. In the second experiment, our method demonstrated good performance on training data and almost six-fold improvement over the state-of-the-art in computational time.

5 Conclusions

In the concluding part of the thesis we highlight some of the limitations of our approach and outline the direction for future work. We summarise our contribution in the final section.

5.1 Limitations

There are some limitations of our approach. The first obvious one, is the weak link between the segmentation and inference. In a way, the pre-segmentation step imposes an upper bound on the segmentation accuracy.

Another issue is that the heavy reliance of the pre-segmentation step on concave regions can easily fall prey to partially observed point clouds. The boundaries between parts which exhibit concavity might be occluded from the view in the first place. Another consideration offer nearly flat objects with the concavity expressed only in profile views. Imagine a human hand with the viewpoint of the observer directed towards the palm. The concavity between the fingers is apparent, yet the normals expressing it would fall short of the support region due to little depth information.

What is more, the representation of shape appearance can be closely linked to the over-fitting problem in machine learning. In general, one cannot expect a robust generalisation of a feature encoding obtained from a single shape to all other shapes of the same kind.

5.2 Future work

We plan to undertake a number of steps in future work to mitigate the limitations discussed. First, incorporating more sophisticated concavity cues, such as ones observed from profile views, might potentially improve the quality of the pre-segmentation. We also intend to investigate other feature encoding schemes, such as spatially sensitive Bag-of-Words (Ovsjanikov et al., 2009). We still believe that contextual information based on isometric distortion can bring in a significant boost to the segmentation accuracy. The diffusion distances computed either from

different approximations of the Laplace-Beltrami operator (Y. Liu, Prabhakaran, et al., 2012) or directly from the Euclidean distance with a Gaussian kernel (A. M. Bronstein, M. M. Bronstein, Kimmel, et al., 2010) are reasonable candidates to study.

5.3 Summary

In this work, we proposed a new approach to the co-segmentation problem that takes into account practical limitations of the existing state-of-the-art methods. Our algorithm is readily applicable to point clouds captured from real sensors and does not require a complete object model both for the reference and the query shape. The generality of our pipeline allows a number of variations and we have investigated only a subset of plausible configurations. However, our results already demonstrated a superior performance compared to the state-of-the art methods. This makes us believe that the introduced concept opens promising directions for future research of shape understanding.

Appendix

1 Expansion of Laplace-Beltrami operator

We can expand and simplify the local Laplace-Beltrami operator as follows:

$$\begin{aligned}
\Delta_{\mathcal{M}} f &= \sum_{i,j=1}^{2} \frac{1}{\sqrt{g}} \frac{\partial}{\partial s_i} \left(\sqrt{g} g^{ij} \frac{\partial f}{\partial g} \right) \\
&= \frac{1}{\sqrt{g}} \frac{\partial}{\partial x} \left(\sqrt{g} g^{11} \frac{\partial f}{\partial x} + \sqrt{g} g^{12} \frac{\partial f}{\partial y} \right) + \frac{1}{\sqrt{g}} \frac{\partial}{\partial y} \left(\sqrt{g} g^{21} \frac{\partial f}{\partial x} + \sqrt{g} g^{22} \frac{\partial f}{\partial y} \right) \\
&= \frac{1}{\sqrt{g}} \Bigg[\sqrt{g} g^{11} \frac{\partial f^2}{\partial^2 x} + (\frac{\partial}{\partial x} \sqrt{g} g^{11}) \frac{\partial f}{\partial x} + \sqrt{g} g^{12} \frac{\partial f^2}{\partial x \partial y} \\
&\quad + (\frac{\partial}{\partial x} \sqrt{g} g^{12}) \frac{\partial f}{\partial y} + \sqrt{g} g^{21} \frac{\partial f^2}{\partial x \partial y} + (\frac{\partial}{\partial y} \sqrt{g} g^{21}) \frac{\partial f}{\partial x} + \sqrt{g} g^{22} \frac{\partial f^2}{\partial^2 y} + \left(\frac{\partial}{\partial y} \sqrt{g} g^{22} \right) \frac{\partial f}{\partial y} \Bigg] \\
&= g^{11} \frac{\partial f^2}{\partial^2 x} + (g^{12} + g^{21}) \frac{\partial f^2}{\partial x \partial y} + g^{22} \frac{\partial f^2}{\partial^2 y} \\
&\quad + \frac{1}{\sqrt{g}} \Bigg[(\frac{\partial}{\partial x} \sqrt{g} g^{11} + \frac{\partial}{\partial y} \sqrt{g} g^{21}) \frac{\partial f}{\partial x} + (\frac{\partial}{\partial x} \sqrt{g} g^{12} + \frac{\partial}{\partial y} \sqrt{g} g^{22}) \frac{\partial f}{\partial y} \Bigg]
\end{aligned} \tag{1}$$

Therefore, the coefficients α_i are defined as:

$$\begin{aligned}
\alpha_1 &= \frac{1}{\sqrt{g}} \left(\frac{\partial}{\partial x} (\sqrt{g} g^{11}) + \frac{\partial}{\partial y} (\sqrt{g} g^{21}) \right) \\
\alpha_2 &= \frac{1}{\sqrt{g}} \left(\frac{\partial}{\partial x} (\sqrt{g} g^{12}) + \frac{\partial}{\partial y} (\sqrt{g} g^{22}) \right) \\
\alpha_3 &= g^{11} \\
\alpha_4 &= g^{12} + g^{21} \\
\alpha_5 &= g^{22}
\end{aligned} \tag{2}$$

In order to obtain the values g^{ij} we'll need to consider the inverse of the metric tensor G. Assume $z_i(x, y) = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2$. The two tangent vector basis are given by $\Gamma_x(p_i) = (1, 0, \frac{\partial z_i}{\partial x})$, $\Gamma_y(p_i) = (0, 1, \frac{\partial z_i}{\partial y})$. By definition,

$g_{ij} = \langle \Gamma_i, \Gamma_j \rangle$, so

$$G = \begin{pmatrix} \langle \Gamma_x, \Gamma_x \rangle & \langle \Gamma_x, \Gamma_y \rangle \\ \langle \Gamma_x, \Gamma_y \rangle & \langle \Gamma_y, \Gamma_y \rangle \end{pmatrix} = \begin{pmatrix} 1 + \left(\frac{\partial z_i}{\partial x}\right)^2 & \frac{\partial z_i}{\partial x}\frac{\partial z_i}{\partial y} \\ \frac{\partial z_i}{\partial x}\frac{\partial z_i}{\partial y} & 1 + \left(\frac{\partial z_i}{\partial y}\right)^2 \end{pmatrix} \tag{3}$$

Let $\gamma(x,y) := \frac{\partial z_i}{\partial x} = a_2 + 2a_4 x + a_5 y$ and $\beta(x,y) := \frac{\partial z_i}{\partial y} = a_3 + a_5 x + 2a_6 y$. Since we take p_i as the origin of the coordinate system, the values of interest are $\gamma := \gamma(0,0) = a_2$ and $\beta := \beta(0,0) = a_3$. It's also easy to see that $\frac{\partial}{\partial x}\gamma = 2a_4$, $\frac{\partial}{\partial y}\gamma = a_5$, $\frac{\partial}{\partial x}\beta = a_5$, $\frac{\partial}{\partial y}\beta = 2a_6$. Then (3) can be rewritten accordingly:

$$G = \begin{pmatrix} 1 + \gamma^2 & \gamma\beta \\ \gamma\beta & 1 + \beta^2 \end{pmatrix} \tag{4}$$

The inverse of the 2×2 matrix is well defined:

$$G^{-1} = \frac{1}{g}\begin{pmatrix} 1 + \beta^2 & -\gamma\beta \\ -\gamma\beta & 1 + \gamma^2 \end{pmatrix} = \frac{1}{g}\begin{pmatrix} 1 + \beta^2 & -\gamma\beta \\ -\gamma\beta & 1 + \gamma^2 \end{pmatrix}, \tag{5}$$

where we used notation $g = \det G$. Using the expressions for g^{ij}, we can plug them into (2) to find partial derivatives for $\alpha_{1..5}$. Making use of the fact $\frac{\partial}{\partial x}\frac{1}{\sqrt{g}} = -\frac{1}{g\sqrt{g}}\Big(2a_2a_4 + a_3a_5\Big)$ and $\frac{\partial}{\partial y}\frac{1}{\sqrt{g}} = -\frac{1}{g\sqrt{g}}\Big(a_2a_5 + 2a_3a_6\Big)$ we can simplify the derivation of α_1 and α_2 by breaking it up into parts:

$$\begin{aligned} \frac{\partial}{\partial x}\Big(\sqrt{g}g^{11}\Big) &= \frac{\partial}{\partial x}\frac{1}{\sqrt{g}}(1+\beta^2) = (1+\beta^2)\frac{\partial}{\partial x}\frac{1}{\sqrt{g}} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}(1+\beta^2) \\ &= -\frac{1+a_3^2}{g\sqrt{g}}(2a_2a_4 + a_3 \cdot a_5) + \frac{1}{\sqrt{g}}2a_3a_5 \\ &= \frac{1}{\sqrt{g}}\Big(2a_3a_6 - \frac{1+a_3^2}{g}(2a_2a_4 + a_3a_5)\Big) \end{aligned} \tag{6}$$

$$\begin{aligned} \frac{\partial}{\partial y}\Big(\sqrt{g}g^{21}\Big) &= -\frac{\partial}{\partial y}\frac{1}{\sqrt{g}}(\gamma\beta) = -\Big[\gamma\beta\frac{\partial}{\partial y}\frac{1}{\sqrt{g}} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}\gamma\beta\Big] \\ &= -\Big[\Big(a_2a_3\Big)\Big(-\frac{1}{g\sqrt{g}}\Big)\Big(2a_3a_6 + a_2a_5\Big) + \frac{1}{\sqrt{g}}(a_3a_5 + 2a_2a_6)\Big] \\ &= \frac{1}{\sqrt{g}}\Big[\frac{a_2a_3}{g}(2a_3a_6 + a_2a_5) - 2a_2a_6 - a_3a_5\Big] \end{aligned} \tag{7}$$

$$\begin{aligned}\frac{\partial}{\partial x}\left(\sqrt{g}g^{12}\right) &= -\frac{\partial}{\partial x}\frac{1}{\sqrt{g}}(\gamma\beta) = -\left[\gamma\beta\frac{\partial}{\partial x}\frac{1}{\sqrt{g}} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial x}\gamma\beta\right] \\ &= -\left[\left(a_2a_3\right)\left(-\frac{1}{g\sqrt{g}}\right)\left(2a_2a_4 + a_3a_5\right) + \frac{1}{\sqrt{g}}(2a_3a_4 + a_2a_5)\right] \\ &= \frac{1}{\sqrt{g}}\left[\frac{a_2a_3}{g}(2a_2a_4 + a_3a_5) - 2a_3a_4 - a_2a_5\right]\end{aligned} \tag{8}$$

$$\begin{aligned}\frac{\partial}{\partial y}\left(\sqrt{g}g^{22}\right) &= \frac{\partial}{\partial y}\frac{1}{\sqrt{g}}(1+\gamma^2) = (1+\gamma^2)\frac{\partial}{\partial y}\frac{1}{\sqrt{g}} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial y}(1+\gamma^2) \\ &= -\frac{1+a_2^2}{g\sqrt{g}}(a_2a_5 + 2a_3a_6) + \frac{1}{\sqrt{g}}2a_2a_5 \\ &= \frac{1}{\sqrt{g}}\left(2a_2a_5 - \frac{1+a_2^2}{g}(a_2a_5 + 2a_3a_6)\right)\end{aligned} \tag{9}$$

Combining the derivations (6) - (9) with (2), we obtain the final formulas for the α's:

$$\begin{aligned}\alpha_1 &= \frac{1}{g}\left(2a_3a_6 - \frac{1+a_3^2}{g}(2a_2a_4 + a_3a_5) + \frac{a_2a_3}{g}(2a_3a_6 + a_2a_5) - 2a_2a_6 - a_3a_5\right) \\ \alpha_2 &= \frac{1}{g}\left(2a_2a_5 - \frac{1+a_2^2}{g}(a_2a_5 + 2a_3a_6) + \frac{a_2a_3}{g}(2a_2a_4 + a_3a_5) - 2a_3a_4 - a_2a_5\right) \\ \alpha_3 &= 1 + a_2^2 \\ \alpha_4 &= 2a_2a_3 \\ \alpha_5 &= 1 + a_3^2\end{aligned} \tag{10}$$

Bibliography

Belkin, M., J. Sun, and Y. Wang (2009). "Constructing Laplace Operator from Point Clouds in Rd". In: Symp. Discrete Algorithms, pp. 1031–1040 (cit. on p. 13).

Bergtholdt, M., J. Kappes, S. Schmidt, and C. Schnörr (2010). "A study of parts-based object class detection using complete graphs". In: *International Journal of Computer Vision* 87.1-2, pp. 93–117 (cit. on pp. 7, 17–19, 34).

Boykov, Y., O. Veksler, and R. Zabih (2001). "Fast approximate energy minimization via graph cuts". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.11, pp. 1222–1239 (cit. on p. 7).

Bronstein, A. M., M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov (2011). "Shape google: Geometric words and expressions for invariant shape retrieval". In: *ACM Transactions on Graphics (TOG)* 30.1, p. 1 (cit. on p. 5).

Bronstein, A. M., M. M. Bronstein, and R. Kimmel (2009). "Topology-invariant similarity of nonrigid shapes". In: *International journal of computer vision* 81.3, pp. 281–301 (cit. on p. 17).

Bronstein, A. M., M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro (2010). "A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching". In: *International Journal of Computer Vision* 89.2-3, pp. 266–286 (cit. on pp. 6, 22, 23, 48).

Bronstein, M. M. and A. M. Bronstein (2010). "Shape recognition with spectral distances". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5, pp. 1065–1071 (cit. on pp. 16, 31).

Bronstein, M. M. and I. Kokkinos (2010). "Scale-invariant heat kernel signatures for non-rigid shape recognition". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, pp. 1704–1711 (cit. on p. 5).

Cazals, F. and M. Pouget (2005). "Estimating differential quantities using polynomial fitting of osculating jets". In: *Computer Aided Geometric Design* 22.2, pp. 121–146 (cit. on p. 36).

Chandrasekaran, V., N. Srebro, and P. Harsha (2012). "Complexity of inference in graphical models". In: *arXiv preprint arXiv:1206.3240* (cit. on p. 8).

Chang, C.-C. and C.-J. Lin (2011). "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2 (3). Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 27:1–27:27 (cit. on p. 34).

Chatfield, K., V. S. Lempitsky, A. Vedaldi, and A. Zisserman (2011). "The devil is in the details: an evaluation of recent feature encoding methods." In: *BMVC*. Vol. 2. 4, p. 8 (cit. on p. 6).

Chen, Q. and V. Koltun (2015). "Robust Nonrigid Registration by Convex Optimization". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2039–2047 (cit. on p. 22).

Chen, X., A. Golovinskiy, and T. Funkhouser (2009). "A benchmark for 3D mesh segmentation". In: *ACM Transactions on Graphics (TOG)*. Vol. 28. 3. ACM, p. 73 (cit. on pp. 8, 9, 35, 37).

Du, G., P. Zhang, J. Mai, and Z. Li (2012). "Markerless kinect-based hand tracking for robot teleoperation". In: *International Journal of Advanced Robotic Systems* 9 (cit. on p. 1).

Freuderl, E. C. (1990). "Complexity of K-Tree Structurecl Constraint Satisfaction Problems". In: *Proc. 8th National Conference on Artificial Intelligence*, pp. 4–9 (cit. on p. 8).

Gonzalez, T. F. (1985). "Clustering to minimize the maximum intercluster distance". In: *Theoretical Computer Science* 38, pp. 293–306 (cit. on p. 34).

Guo, Y., M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok (2016). "A comprehensive performance evaluation of 3D local feature descriptors". In: *International Journal of Computer Vision* 116.1, pp. 66–89 (cit. on p. 5).

Hilaga, M., Y. Shinagawa, T. Kohmura, and T. L. Kunii (2001). "Topology matching for fully automatic similarity estimation of 3D shapes". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, pp. 203–212 (cit. on p. 36).

Huang, Q., V. Koltun, and L. Guibas (2011). "Joint shape segmentation with linear programming". In: *ACM Transactions on Graphics (TOG)*. Vol. 30. 6. ACM, p. 125 (cit. on p. 5).

Jaakkola, T., D. Haussler, et al. (1999). "Exploiting generative models in discriminative classifiers". In: *Advances in neural information processing systems*, pp. 487–493 (cit. on p. 6).

Johnson, A. E. and M. Hebert (1999). "Using spin images for efficient object recognition in cluttered 3D scenes". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21.5, pp. 433–449 (cit. on p. 5).

Kaick, O. van, A. Tagliasacchi, O. Sidi, H. Zhang, D. Cohen-Or, L. Wolf, and G. Hamarneh (2011). "Prior knowledge for part correspondence". In: *Computer Graphics Forum*. Vol. 30. 2. Wiley Online Library, pp. 553–562 (cit. on pp. 4, 8, 21, 36, 40, 44, 45).

Kalogerakis, E., A. Hertzmann, and K. Singh (2010). "Learning 3D mesh segmentation and labeling". In: *ACM Transactions on Graphics (TOG)*. Vol. 29. 4. ACM, p. 102 (cit. on pp. 4, 8, 21, 35, 36).

Kappes, J. H., B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, et al. (2014). "A comparative study of modern inference techniques for structured discrete energy minimiza-

tion problems". In: *International Journal of Computer Vision*, pp. 1–30 (cit. on p. 7).

Kolmogorov, V. (2006). "Convergent tree-reweighted message passing for energy minimization". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28.10, pp. 1568–1583 (cit. on pp. 7, 34).

Kolmogorov, V. and R. Zabin (2004). "What energy functions can be minimized via graph cuts?" In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26.2, pp. 147–159 (cit. on p. 7).

Lavoué, G. (2012). "Combination of bag-of-words descriptors for robust partial shape retrieval". In: *The Visual Computer* 28.9, pp. 931–942 (cit. on p. 6).

Li, Y., J. L. Fu, and N. S. Pollard (2007). "Data-driven grasp synthesis using shape matching and task-based pruning". In: *Visualization and Computer Graphics, IEEE Transactions on* 13.4, pp. 732–747 (cit. on p. 1).

Liang, J., R. Lai, T. W. Wong, and H. Zhao (2012). "Geometric understanding of point clouds using laplace-beltrami operator". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, pp. 214–221 (cit. on p. 14).

Liu, R., H. Zhang, A. Shamir, and D. Cohen-Or (2009). "A Part-aware Surface Metric for Shape Analysis". In: *Computer Graphics Forum*. Vol. 28. 2. Wiley Online Library, pp. 397–406 (cit. on p. 36).

Liu, Y., B. Prabhakaran, and X. Guo (2012). "Point-based manifold harmonics". In: *Visualization and Computer Graphics, IEEE Transactions on* 18.10, pp. 1693–1703 (cit. on p. 48).

Liu, Y., H. Zha, and H. Qin (2006). "Shape topics: A compact representation and new algorithms for 3d partial shape retrieval". In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. IEEE, pp. 2025–2032 (cit. on p. 6).

Marton, Z. C., R. B. Rusu, and M. Beetz (May 2009). "On Fast Surface Reconstruction Methods for Large and Noisy Datasets". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Kobe, Japan (cit. on p. 44).

Meng, M., J. Xia, J. Luo, and Y. He (2013). "Unsupervised co-segmentation for 3D shapes using iterative multi-label optimization". In: *Computer-Aided Design* 45.2, pp. 312–320 (cit. on p. 5).

Mitra, N. J., M. Wand, H. Zhang, D. Cohen-Or, V. Kim, and Q.-X. Huang (2014). "Structure-aware shape processing". In: *ACM SIGGRAPH 2014 Courses*. ACM, p. 13 (cit. on p. 8).

Ohbuchi, R., K. Osada, T. Furuya, and T. Banno (2008). "Salient local visual features for shape-based 3D model retrieval". In: *Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on*. IEEE, pp. 93–102 (cit. on p. 6).

Ovsjanikov, M., A. M. Bronstein, M. M. Bronstein, and L. J. Guibas (2009). "Shape Google: a computer vision approach to invariant shape retrieval". In: *Proc. NORDIA* 1.2 (cit. on pp. 6, 47).

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann (cit. on p. 7).

Perronnin, F., J. Sánchez, and T. Mensink (2010). "Improving the fisher kernel for large-scale image classification". In: *Computer Vision–ECCV 2010*. Springer, pp. 143–156 (cit. on pp. 6, 30).

Rand, W. M. (1971). "Objective criteria for the evaluation of clustering methods". In: *Journal of the American Statistical association* 66.336, pp. 846–850 (cit. on p. 37).

Reuter, M., S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo (2009). "Discrete Laplace–Beltrami operators for shape analysis and segmentation". In: *Computers & Graphics* 33.3. {IEEE} International Conference on Shape Modelling and Applications 2009, pp. 381–390. ISSN: 0097-8493. URL: `http://www.sciencedirect.com/science/article/pii/S0097849309000272` (cit. on p. 13).

Robertson, N. and P. D. Seymour (1986). "Graph minors. II. Algorithmic aspects of tree-width". In: *Journal of algorithms* 7.3, pp. 309–322 (cit. on p. 8).

Rodolà, E., L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers (2015). "Partial functional correspondence". In: *arXiv preprint arXiv:1506.05274* (cit. on p. 22).

Rusu, R. B., N. Blodow, and M. Beetz (2009). "Fast point feature histograms (FPFH) for 3D registration". In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, pp. 3212–3217 (cit. on pp. 5, 12).

Rusu, R. B. and S. Cousins (May 2011). "3D is here: Point Cloud Library (PCL)". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China (cit. on p. 34).

Rusu, R. B., Z. C. Marton, N. Blodow, and M. Beetz (2008). "Learning informative point classes for the acquisition of object model maps". In: *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*. IEEE, pp. 643–650 (cit. on pp. 5, 11).

Sánchez, J., F. Perronnin, T. Mensink, and J. Verbeek (2013). "Image classification with the fisher vector: Theory and practice". In: *International journal of computer vision* 105.3, pp. 222–245 (cit. on p. 6).

Saxena, A., J. Driemeyer, and A. Y. Ng (2008). "Robotic grasping of novel objects using vision". In: *The International Journal of Robotics Research* 27.2, pp. 157–173 (cit. on p. 1).

Schoeler, M., J. Papon, and F. Wörgötter (2015). "Constrained Planar Cuts-Object Partitioning for Point Clouds". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5207–5215 (cit. on pp. 9, 24).

Schulman, J., J. Ho, C. Lee, and P. Abbeel (2013). "Learning from demonstrations through the use of non-rigid registration". In: *Proceedings of the 16th international symposium on robotics research (ISRR)* (cit. on p. 1).

Shapira, L., A. Shamir, and D. Cohen-Or (2008). "Consistent mesh partitioning and skeletonisation using the shape diameter function". In: *The Visual Computer* 24.4, pp. 249–259 (cit. on p. 36).

Sidi, O., O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or (2011). *Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering*. Vol. 30. 6. ACM (cit. on pp. 5, 8).

Sun, J., M. Ovsjanikov, and L. Guibas (2009). "A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion". In: *Proceedings of the Symposium on Geometry Processing*. SGP '09. Berlin, Germany: Eurographics Association, pp. 1383–1392. URL: `http://dl.acm.org/citation.cfm?id=1735603.1735621` (cit. on p. 5).

Toldo, R., U. Castellani, and A. Fusiello (2009). "Visual Vocabulary Signature for 3D Object Retrieval and Partial Matching". In: *Proceedings of the 2Nd Eurographics Conference on 3D Object Retrieval*. 3DOR '09. Munich, Germany: Eurographics Association, pp. 21–28. ISBN: 978-3-905674-16-3 (cit. on p. 6).

Tombari, F., S. Salti, and L. Di Stefano (2010). "Unique Signatures of Histograms for Local Surface Description". In: *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*. ECCV'10. Heraklion, Crete, Greece: Springer-Verlag, pp. 356–369. ISBN: 3-642-15557-X, 978-3-642-15557-4. URL: `http://dl.acm.org/citation.cfm?id=1927006.1927035` (cit. on pp. 5, 11).

Torralba, A., K. P. Murphy, and W. T. Freeman (2007). "Sharing visual features for multiclass and multiview object detection". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29.5, pp. 854–869 (cit. on p. 4).

Wainwright, M. J., T. S. Jaakkola, and A. S. Willsky (2003). "Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching". In: *Workshop on Artificial Intelligence and Statistics*. Vol. 21. Society for Artificial Intelligence and Statistics Np, p. 97 (cit. on p. 7).

— (2005). "MAP estimation via agreement on trees: message-passing and linear programming". In: *Information Theory, IEEE Transactions on* 51.11, pp. 3697–3717 (cit. on pp. 7, 34).

Ye, M., X. Wang, R. Yang, L. Ren, and M. Pollefeys (2011). "Accurate 3d pose estimation from a single depth image". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, pp. 731–738 (cit. on p. 1).

Yedidia, J. S., W. T. Freeman, and Y. Weiss (2005). "Constructing free-energy approximations and generalized belief propagation algorithms". In: *Information Theory, IEEE Transactions on* 51.7, pp. 2282–2312 (cit. on pp. 7, 34).