Computer Vision and Image Understanding, Volume 115, Issue 10, October 2011, Pages 1427-1448

An Experimental Study of Four Variants of Pose Clustering from Dense Range Data

Ulrich Hillenbrand* and Alexander Fuchs†
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
82234 Wessling, Germany

Abstract

Parameter clustering is a robust estimation technique based on location statistics in a parameter space where parameter samples are computed from data samples. This article investigates parameter clustering as a global estimator of object pose or rigid motion from dense range data without knowing correspondences between data points. Four variants of the algorithm are quantitatively compared regarding estimation accuracy and robustness: sampling poses from data points or from points with surface normals derived from them, each combined with clustering poses in the canonical or consistent parameter space, as defined in [1]. An extensive test data set is employed: synthetic data generated from a public database of three-dimensional object models through various levels of corruption of their geometric representation; real range data from a public database of models and cluttered scenes. It turns out that sampling raw data points and clustering in the consistent parameter space yields the estimator most robust to data corruption. For data of sufficient quality, however, sampling points with normals is more efficient; this is most evident when detecting objects in cluttered scenes. Moreover, the consistent parameter space is always preferable to the canonical parameter space for clustering.

Keywords: robust estimation; pose estimation; range data; parameter density; clustering; performance evaluation.

*Corresponding author: `Ulrich.Hillenbrand@dlr.de`
†Present address: Automatic Control Laboratory, ETH Zurich, 8092 Zurich, Switzerland, `fuchs@control.ee.ethz.ch`

1 Introduction

1.1 The pose estimation problem

Estimation of the pose of known objects in unknown scenes is a prerequisite of many robotic applications, such as bin picking, object manipulation, and self localization. With the advent of fast and affordable range-imaging technologies, such as stereo image processing, laser range scanners, laser stripe profilers, structured-light cameras, and time-of-flight cameras, dense three-dimensional (3D) data points have become available as a geometric scene representation [2, 3, 4, 5]. Likewise, for reverse engineering of surface geometries, a set of regular shapes is often fitted to a 3D data set to obtain a description that can be further processed, e.g., by a CAD system. Also data fusion, e.g., in medical applications, often requires the registration of 3D data sets acquired from different sensors or at different times. Moreover, the problem of pose estimation is mathematically and algorithmically similar to the problem of motion estimation of rigid objects, such as in visual object tracking, or of a sensor relative to its environment, such as in navigation tasks or when registering data sets acquired from different viewpoints. The only difference between pose and motion estimation is that for the former, one of the two data sets that have to be registered is a priori given as the model of the object or the environment, while for the latter both data sets are acquired in the process.

Often there will be just vague or no prior knowledge on object pose or motion. For pose estimation this is the rule, while in a motion sequence the situation occurs when the motion is erratic and fast compared to the processed data frame rate, or when tracked objects get temporarily out of sight. In such cases, a *global* search for the pose or motion parameters has to be performed.

Data acquired from a natural scene do not usually contain just a single object, and two data sets in a motion sequence do not completely overlap. The estimator hence needs to be *robust* in the statistical sense, that is, it must select in the estimation process that part of the data that does match between two sets. In particular, a robust pose estimator has to discard outliers of three kinds. i) Gross errors of the measurement process arise from artefacts of the sensor or from prior data processing. ii) So-called pseudo-outliers represent other scene structures beside the sought object. Often this kind of outliers present the hardest challenge to a robust estimator. iii) For a pose estimator of general shapes, unlike for an estimator of analytic parametric structures (lines, circles, planes, etc.), the hardest challenge often

derives from the correspondence problem. That is, the detailed model-to-scene correspondences, on the level of features or data points, are often not known in advance and must be established during estimation. With growing number of features or data points, the number of possible correspondences then explodes, diminishing the proportion of correct correspondences dramatically, and each false correspondence is effectively an outlier to the rigid motion model. This circumstance makes pose estimation from dense data sets without knowing correspondences a particularly hard parameter estimation problem.

For special objects (often man-made objects) it is possible to rely on higher-level features such as corners, edges, planes, or other geometric primitives to alleviate the correspondence problem. Given sufficient data quality, there are also a number of more generic local shape descriptors available, e.g., [6, 7, 8, 9]. Furthermore, if scenes are constrained as to their variation of lighting and viewpoint, appearance-based image descriptors can be very efficient in guiding correspondence search; see [10] for a recent review.

In this article, we consider alignment of 3D data sets without any hint as to correspondence; indeed, no truly corresponding points are assumed to even exist between the data sets. This constitutes the most generally applicable procedure for pose estimation from range data, as no special features of the objects, quality of the data, or constraints on the scene are required. At the same time, it is the most challenging case for a robust estimator, because of excessive amounts of outliers incurred from false correspondences: almost all possible correspondences are wrong even for moderate numbers of data points. Apart from its practical relevance, we hence consider the present scenario as an interesting test case of robust estimation.

1.2 Robust and global estimators

Practical implementations of robust and global estimators for computer vision problems are usually based upon sampling minimal subsets from the data and computing parameter hypotheses that satisfy constraints posed by each data sample and the underlying model. The most basic distinction between different methods is by how these parameter hypotheses are further processed. In general terms, the hypotheses may be either evaluated in data space or analyzed in parameter space.

Parameter clustering is a technique characterized by computing robust location statistics in a parameter space. The general strategy of parameter clustering has been exploited for a long time in numerous variations [11, 12, 13, 14, 15, 16, 17, 18, 1, 19, 20, 9], although mostly not for pose

estimation but for fitting analytic shape models, which does not suffer the correspondence problem. Common to all these approaches is that data samples are drawn from which parameter samples are computed, often called 'votes' or 'hypotheses'. The intuition is that significant data populations matching an instance of the model will produce many parameter samples that coincide approximately, hence localize in a cluster. The final parameter estimate is the estimated location of that cluster. The popular Hough transform and its generalizations can be regarded as a discrete variant of parameter clustering. In [18, 20], the problem of pose estimation has been considered and solved through mean-shift clustering in continuous parameter spaces. The procedure proposed there avoids using a global parameterization of motions, however, at the cost of transforming between many local parameterizations and, thus, processing only a small number of hypotheses. Accordingly, its application has been to sparse data with feature-guided correspondences, with just a moderate proportion of correspondence outliers. Similarly, clustering in a pose parameter space is used in [9] for object recognition, with correspondences established by a highly descriptive feature. In [1, 19, 21], another formulation of clustering in continuous parameter spaces of transformations has been developed and applied to pose estimation without correspondences.

The alternative to parameter clustering is the evaluation of each parameter hypothesis in data space, that is, in relation to all the available data. The final parameter estimate then is the hypothesis among the sample that reaches the highest score. Various objective functions have been proposed to asses the quality of fit, where the number of supporting data points and the size of the residuals are considered in various ways and to varying degrees. The classic variants are M-estimators [22, 23, 17], a robust generalization of maximum-likelihood estimators, random sample consensus (RANSAC) [24, 25], where traditionally the amount of assumed inlier data is maximized, and least median of squares (LMedS) [26, 23], where the median of squared residuals is minimized. Continuing efforts to achieve higher robustness have led to numerous more recent variants through designing new objective functions or sampling techniques [27, 28, 29, 30, 31, 32, 33, 8, 34]. Like for parameter clustering, these algorithms have mostly not been investigated on pose estimation with a severe correspondence problem, but rather on fitting analytic structures or with correspondences guided by highly descriptive features. It appears, however, that data space methods have obtained more attention in the research community recently, and that the study of parameter space methods has been somewhat neglected.

1.3 Scope of study and article outline

In this article, we focus on studying pose clustering as the parameter space method of global and robust pose estimation. A fair comparison with the various data space methods, while certainly desirable, is problematical because of the different strategies employed. The issue of performance comparison will be discussed in section 5.2. Here we note that when comparing to traditional-style RANSAC – the fastest of highly robust data space methods [32] – on equal run times, parameter clustering achieved superior accuracy and robustness by a large margin [35]. For the problem of pose estimation from dense range data without correspondences, where the proportion of outliers due to false correspondences is close to 100%, it might seem preferable to spend computational resources on accumulating a large sample of hypotheses rather than on evaluating each single one of a much smaller sample. However, this conclusion would need to be consolidated by a careful study involving also the more recent variants of data space methods.

Unlike for the data space methods, for parameter clustering the resulting estimate depends upon the parameterization chosen for the model to be estimated. This fact brings up the question of a proper choice of parameter space for clustering, which was often neglected in the literature. Recently, we have derived a consistency criterion for parameterizations used for clustering [1]. As a special case, we have treated pose or motion estimation in 3D space and have given a consistent parameterization of the Euclidian group of motions.

For all sampling-based methods, the way hypotheses are derived from the data is critical. For pose or motion estimation from 3D data, the sampling procedure is related to the order of geometric surface description employed. Thus, pose hypotheses may be directly computed from subsets of the range data points obtained from a surface, in which case a zeroth-order surface description is effectively used. Alternatively, normal vectors may be estimated from the data points first and pose hypotheses computed from subsets of points with normals, hence using a first-order surface description. If curvature information is also exploited, we rely on a second-order surface description.

In this article, we investigate the relative estimation accuracy and robustness of four variants of the pose clustering algorithm: hypotheses computed from subsets of range data points or from subsets of points with surface normals, each combined with clustering hypotheses in the canonical or consistent pose space.

This study uses data from two public databases. Synthetic range data is

generated from 3D object models from the database of the Princeton Shape Benchmark [36, 37]: the object models are re-sampled with data points and systematically degraded to simulate a range of more or less favorable measurement conditions. In this way, a quantitative study on a very large data set is realized and dependence of relative estimator performance upon some data characteristics made explicit. Moreover, range data from real scenes containing several objects are obtained from [8, 9, 38] and used to complement the study through demonstrating effects of clutter and occlusion.

This work extends current studies on robust estimators in a number of ways. i) Global pose estimation from dense data points without correspondences has rarely been studied systematically. It is a particularly challenging domain for a robust estimator because of the extreme amount of outliers incurred from false correspondences. ii) We are not aware of another quantitative study of continuous parameter clustering or parameter density maximization as a robust estimator on a reasonably sized data set. iii) The concept of consistent clustering, introduced in [1], is evaluated empirically on a large data set. The significance of a consistent parameterization for clustering is thus established. iv) To our knowledge, one of the investigated variants of pose clustering, the one with consistent parameterization and hypotheses sampling from points with normals, has never been published before. This variant turns out to be the most effective estimator on most of the test data. v) We are not aware of a quantitative study of any robust estimator that is based upon a comparably sized data set. The data set used here could serve as a common benchmark for robust pose estimators.

In the next section, we summarize the idea of parameter clustering in general and of pose clustering in particular. Section 3 introduces the four variants of the pose clustering algorithm we investigate. The experiments are described in section 4, including a specification of the algorithmic parameters and of the test data used, a definition of the error statistics, and a compilation of the results. Section 5 discusses the issue of outliers and the breakdown point, as well as some issues not addressed in this study, and summarizes the main results.

2 Parameter clustering

This section outlines a general formulation of the parameter estimation problem and of its solution through parameter clustering and then specializes to the case of pose clustering. For details and derivations the reader is referred to [1].

2.1 Problem formulation

Suppose we want to estimate a transformation T from a model- or data-point set $X \subset \mathbb{R}^m$ to a data-point set $Y \subset \mathbb{R}^n$. The transformation of a point $x \in X$ is assumed to have the general parametric form

$$T(x, \alpha) = F(G_\alpha(x)) = F \circ G_\alpha(x) \ , \tag{1}$$

where $\{G_\alpha \,|\, \alpha \in \mathcal{P}\}$ is a d-dimensional Lie group of transformations

$$G_\alpha \ : \ \mathbb{R}^m \longrightarrow \mathbb{R}^m \ , \tag{2}$$

charted in a parameter space $\mathcal{P} \subset \mathbb{R}^d$, and F is a continuously differentiable function

$$F \ : \ \mathbb{R}^m \longrightarrow \mathbb{R}^n \ . \tag{3}$$

For a set of corresponding point pairs $(x, y) \in C \subset X \times Y$ and a unique parameter value $\alpha \in \mathcal{P}$, we thus have the relation

$$y = T(x, \alpha) + \epsilon_{x,y} \ , \tag{4}$$

where $\epsilon_{x,y} \in \mathbb{R}^n$ are measurement errors. The estimation goal is to uncover the transformation $T = F \circ G_\alpha$ between the point sets X and Y. In motion or pose estimation, $\{G_\alpha \,|\, \alpha \in \mathcal{P}\}$ is the 6D Euclidian group acting on points $x \in \mathbb{R}^3$. The function F is the identity, if the points Y are range data, or a perspective projection with lens distortion, if Y is a set of 2D image points.

Let a unique transformation $T = F \circ G_\alpha$, and hence a unique parameter value $\alpha \in \mathcal{P}$, be determined by posing the ln constraints

$$y_i - T(x_i, \alpha) = 0 \ , \quad i = 1, 2, \ldots, l \ , \tag{5}$$

for any non-degenerate subset of l point pairs $\{(x_i, y_i) \,|\, i = 1, 2, \ldots, l\} \subset X \times Y$. In order to satisfy all the constraints, it may be necessary to enlarge the group $\{G_\alpha \,|\, \alpha \in \mathcal{P}\}$ by inclusion of nuisance parameters. In particular, when estimating rigid motions from range data, $l = 3$ and extra parameters describe deformations needed to match three point pairs. In the final estimate, however, these nuisance parameters are constrained to the values defining the original group of rigid motions.

We write bold Greek symbols $\boldsymbol{\alpha}$ to denote the parameters $\alpha \in \mathcal{P}$ extended with any required nuisance parameters. For the data, we use the bold notations $\boldsymbol{x} = (x_1, x_2, \ldots, x_l) \in \mathbb{R}^{lm}$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_l) \in \mathbb{R}^{ln}$, and

$$\boldsymbol{y} = (T(x_1, \boldsymbol{\alpha}), T(x_2, \boldsymbol{\alpha}), \ldots, T(x_l, \boldsymbol{\alpha})) \equiv T(\boldsymbol{x}, \boldsymbol{\alpha}) \equiv F(G_{\boldsymbol{\alpha}}(\boldsymbol{x})) \ . \tag{6}$$

Let $p(\boldsymbol{x}, \boldsymbol{y})$ be the probability density on $\mathbb{R}^{l(m+n)}$ of measuring l data-point pairs (x_i, y_i), $i = 1, 2, \ldots, l$. When sampling subsets of l point pairs from $X \times Y$ and computing the parameter value $\boldsymbol{\alpha}$ associated with each subset by solving the system (5), we obtain parameter samples from the probability density

$$\rho(\boldsymbol{\alpha}) = \int_{\mathbb{R}^{lm}} d\boldsymbol{x}\, p(\boldsymbol{x}, T(\boldsymbol{x}, \boldsymbol{\alpha}))\ |\det \partial_2 T(\boldsymbol{x}, \boldsymbol{\alpha})| \ . \tag{7}$$

Here $\partial_2 T(\boldsymbol{x}, \boldsymbol{\alpha})$ is the $ln \times ln$ derivative of $T(\boldsymbol{x}, \boldsymbol{\alpha})$ w.r.t. $\boldsymbol{\alpha}$, det denotes the determinant, and $|\cdot|$ the absolute value. The parameter density (7) is the probability density of measuring l data-point pairs related through the transformation $T = F \circ G_{\boldsymbol{\alpha}}$. Of course, the data density p and, hence, the parameter density ρ are not explicitly known.

It is the goal of parameter clustering to find the maximum of the parameter density (7) in $\mathcal{P}$. The location of this maximum is returned as the parameter estimate, that is,

$$\hat{\alpha} = \arg \max_{\boldsymbol{\alpha} \in \mathcal{P} \times \{0\}} \rho(\boldsymbol{\alpha}) \ , \tag{8}$$

where $\mathcal{P} \times \{0\}$ is the part of the extended parameter space that defines the original group $\{G_\alpha \mid \alpha \in \mathcal{P}\}$ of transformations, e.g., rigid motions in the present case. The location of the maximal density has to be estimated, in turn, from the parameter samples obtained.

It may seem that the parameter estimate (8) is similar to a maximum-a-posterior estimate. The important difference, however, is that we do not assume here a specific probabilistic observation model, and hence the parameter density (7) is not a true posterior density. The implicit assumption in parameter clustering is that maximizing the parameter density (7) instead of the true posterior density is the best one can do for an estimate, if a good observation model[1] is not available.

2.2 Consistency

Suppose we were faced with a population of data points that is symmetric w.r.t. the group $\{G_\alpha \mid \alpha \in \mathcal{P}\}$, i.e., the data density is invariant under transformation with G_α,

$$p_{\text{inv}}(\boldsymbol{x}, \boldsymbol{y}) = p_{\text{inv}}(G_\alpha(\boldsymbol{x}), \boldsymbol{y})\ |\det \partial G_\alpha(\boldsymbol{x})| \quad \forall\, \alpha \in \mathcal{P} \ . \tag{9}$$

[1] Note that a good observation model must be able to explain both inliers and outliers to the parametric model that is fitted.

Considering, for instance, rotations $R_\alpha \in SO(3)$, an isotropic distribution of points satisfies $p_{\text{inv}}(\boldsymbol{x}, \boldsymbol{y}) = p_{\text{inv}}(R_\alpha(\boldsymbol{x}), \boldsymbol{y})$, for any parameterization of rotations by $\alpha \in \mathcal{P}$. Consistency of parameter clustering requires that such symmetry be reflected in the estimates, which should not be biased towards any particular transformation $T = F \circ G_{\hat{\alpha}}$ with $\hat{\alpha}$ given by (8). Clearly, the only way of insuring this is by making the associated parameter density[2]

$$\rho_{\text{inv}}(\boldsymbol{\alpha}) = \int_{\mathbb{R}^{lm}} d\boldsymbol{x}\, p_{\text{inv}}(\boldsymbol{x}, T(\boldsymbol{x}, \boldsymbol{\alpha}))\ |\det \partial_2 T(\boldsymbol{x}, \boldsymbol{\alpha})| \tag{10}$$

uniform, that is,

$$\rho_{\text{inv}}(\boldsymbol{\alpha}) = \text{const.} \quad \forall\, \boldsymbol{\alpha} \in \mathcal{P} \times \{0\}\ . \tag{11}$$

Hence, clustering of parameters $\alpha \in \mathcal{P}$ is consistent, if and only if the parameter population arising from a group-symmetric data population, satisfying (9), is uniform, i.e., satisfies (11).

Note that condition (9) is independent of the parameterization, while (11) is not. Therefore, requiring the coincidence of the two is a *selection criterion for parameterizations*. We call parameterizations satisfying (11) *consistent* for clustering.

2.3 Pose clustering

In this article, we deal with the special case of $\{G_\alpha \mid \alpha \in \mathcal{P}\}$ being the Euclidian group $SE(3)$ of motions in 3D, that is, for a point $x \in \mathbb{R}^3$,

$$G_\alpha(x) = R_{\alpha_{\text{rot}}}(x) + t_{\alpha_{\text{trans}}}\ , \tag{12}$$

with a rotation $R_{\alpha_{\text{rot}}} \in SO(3)$ and a translation $t_{\alpha_{\text{trans}}} \in \mathbb{R}^3$, parameterized by $\alpha = (\alpha_{\text{rot}}, \alpha_{\text{trans}}) \in \mathcal{P} \subset \mathbb{R}^6$. We want to estimate transformations $T = G_\alpha$ from model points $X \subset \mathbb{R}^3$ to range data points $Y \subset \mathbb{R}^3$, hence F in eq. (1) is the identity.

3 Pose clustering algorithms

The purpose of parameter clustering is to arrive at a parameter estimate through estimating the location of the maximal parameter density; cf. eq. (8). Since the parameter density (7) itself is unknown, its maximum has to be estimated from parameter samples. The algorithm hence consists of the two steps

[2]The density (10) is closely related to the invariant Haar measure of the group $\{G_\alpha | \alpha \in \mathcal{P}\}$.

1. sampling parameters,

2. locating the maximal parameter density.

Both procedures with their variants will now be described for the case of pose estimation.

3.1 Parameter sampling

In order to produce a number of pose parameter samples, data samples are drawn from $X \times Y$ from which pose hypotheses are computed. Two variants of parameter sampling were tested.

3.1.1 Sampling from point triples

A pose hypothesis can be computed from a minimum subset of three X-points matched against a minimum subset of three Y-points. The sampling proceeds thus as follows.

1. Randomly draw a point triple from X.

2. Randomly draw a point triple from Y among all triples that are geometrically consistent with the triple drawn from X.

3. Compute the rigid motion between the two triples.

4. Compute the six parameters of the hypothetical motion.

The parameter samples thus obtained are collected into a spatial array or a tree of bins, from where they can be efficiently retrieved for the subsequent parameter density maximization; cf. section 3.2. The sampling process stops as soon as a significant number of parameter samples has accumulated anywhere in parameter space. This condition is pragmatically taken as fulfilled when one of the bins is full, which depends on the memory allocated for each bin.

Corresponding data points from X and Y can be found only among geometrically consistent groups of points. For drawing triples of potentially corresponding points in sampling step 2, one should hence exploit the constraints that arise from rigid motion. These are i) (approximate) congruence of the triangles defined by the point triples and ii) viewpoint consistency. The latter means that the plane defined by three simultaneously visible points on a non-transparent solid shape generally exposes the same side to

the sensor.[3] For the data used in this study, however, there is no information available on sensor gaze direction. Hence, only congruence of triangles was exploited for sampling.

The congruence constraint is efficiently enforced by using a hash table of point triples sampled from Y that has been built prior to the actual triple-pair-sampling procedure.[4] In sampling step 2, this table is accessed through a key, computed from the drawn X-triple, that encodes the intrinsic geometry of a point triple $\{p_1, p_2, p_3\} \subset \mathbb{R}^3$, like the three point-to-point distances,

$$(\|p_1 - p_2\|, \|p_2 - p_3\|, \|p_3 - p_1\|) \ . \tag{13}$$

In step 3 of the sampling procedure, the least-squares rotation $\bar{R} \in SO(3)$ and translation $\bar{t} \in \mathbb{R}^3$ between two point triples $\{x_1, x_2, x_3\} \subset X$ and $\{y_1, y_2, y_3\} \subset Y$ are computed, i.e.,

$$(\bar{R}, \bar{t}) = \arg \min_{(R,t) \in SE(3)} \sum_{i=1}^{3} \|R(x_i) + t - y_i\|^2 \ . \tag{14}$$

The method in [39] provides a solution, based on quaternions, that is specifically tailored to the three-point case and is hence more efficient than the general ones for point numbers ≥ 3 [40]. If the three point pairs (x_i, y_i) are approximately corresponding between X and Y, the pose hypothesis $(\bar{R}, \bar{t})$ will be close to the true pose.

The sensitivity for noise of a pose hypothesis is larger for smaller distances between the three involved points. However, because larger distances between points are generally more frequent than closer distances[5], unreliable pose hypotheses computed from very close points are largely suppressed by random sampling without taking any special measures.

Note that by sampling approximately congruent point triples and computing least-squares solutions for rigid motion between them, we effectively probe the parameter space of interest, denoted as $\mathcal{P} \times \{0\}$ in eq. (8), and avoid computation of nuisance parameters that would describe deformations.

3.1.2 Sampling from surflet pairs

When given dense range data, it is possible to estimate surface normals from the data points as follows. Let $B_r(x) = \{x' \in \mathbb{R}^3 \mid \|x' - x\| < r\}$ be the ball

[3] Exceptions may occur, e.g., for triples that span holes through a shape.

[4] By swapping the order of sampling from X and Y in steps 1 and 2 of the sampling procedure, building the hash table from the model set X may in fact be done offline as part of the model generation process.

[5] For each point, more other points are found further away than in its vicinity.

around $x \in \mathbb{R}^3$ with radius $r > 0$. The sample mean of the data points in $B_r(x)$ is

$$\bar{x} = \frac{1}{|X \cap B_r(x)|} \sum_{x' \in X \cap B_r(x)} x' \ , \tag{15}$$

where $|\cdot|$ denotes the cardinality of a point set. An estimate $\hat{n}_x$ of a local surface normal at $x \in X$ is given by the eigenvector corresponding to the smallest eigenvalue of the sample covariance matrix

$$C = \frac{1}{|X \cap B_r(x)|} \sum_{x' \in X \cap B_r(x)} (x' - \bar{x}) (x' - \bar{x})^T \ . \tag{16}$$

For a reasonable estimate of surface normals, the radius r needs to be adjusted according to two competing objectives. On one hand, the surflets produced by the model data and the perturbed measured data should be as similar as possible. This suggests to choose r as large as possible, since increasing the relevant data sample reduces the variance of estimated normals. On the other hand, r should be small enough to capture the geometric features that make the object distinguishable. For a too large r these features would be averaged out. The optimal compromise between these two objectives depends upon the curvatures of the object surface, the density of the data points, and the level of noise in the data. We chose a radius of $r = 0.12$ maximum-bounding-box-lengths, which corresponds to a length of around 10% of the objects' longest extension, capturing most of the characteristic geometry of the objects. Clearly, we cannot be sure to have used an optimal value for r.

For some kinds of data, there can be quite different ways of estimating surface normals. For instance, in some volumetric images they may be obtained by computing spatial gradients of image densities.

The outward/inward directions of the surface normals follow usually from the sensor gaze direction, which is not given for our test data set. Instead, the outward normal directions were here simply chosen to agree with those provided for the models from the Princeton Shape Benchmark. We will refer to the set $\tilde{X} = \{(\bar{x}, \hat{n}_x) \,|\, x \in X\}$ of points with their outward normal as the *surflets* associated with the points X. Likewise, $\tilde{Y} = \{(\bar{y}, \hat{n}_y) \,|\, y \in Y\}$ are the surflets associated with the points Y.

A pose hypothesis can now be computed from a minimum subset of two $\tilde{X}$-surflets matched against a minimum subset of two $\tilde{Y}$-surflets. The sampling proceeds thus as follows.

1. Randomly draw a surflet pair from $\tilde{X}$.

2. Randomly draw a surflet pair from $\tilde{Y}$ among all pairs that are geometrically consistent with the pair drawn from $\tilde{X}$.

3. Compute the rigid motion between the two pairs.

4. Compute the six parameters of the hypothetical motion.

There are two differences to the sampling procedure from point triples: regarding the enforcement of geometric consistency and the computation of rigid motion between two surflet pairs.

Geometric consistency of a surflet pair with another requires that the two pairs be (approximately) congruent, as for the point triples. The intrinsic geometry of a surflet pair $\{(p_1, n_1), (p_2, n_2)\}$ sampled from $\tilde{X}$ or $\tilde{Y}$ can be described by four parameters, e.g., by the angle between the two surface normals n_1, n_2 and the three components of the point-to-point difference vector $p_1 - p_2$ along the base vectors

$$\begin{aligned} b_1 &= n_1 \ , \\ b_2 &= \frac{n_1 \times n_2}{\|n_1 \times n_2\|} \ , \\ b_3 &= \frac{(n_1 \times n_2) \times n_1}{\|(n_1 \times n_2) \times n_1\|} \ , \end{aligned} \tag{17}$$

that is, by the parameters

$$(\arccos(n_1 \cdot n_2), (p_1 - p_2) \cdot b_1, (p_1 - p_2) \cdot b_2, (p_1 - p_2) \cdot b_3) \ . \tag{18}$$

As above, the congruence constraint is efficiently enforced in sampling step 2 by indexing into a hash table of surflet pairs previously sampled from $\tilde{Y}$. The table is accessed through the four parameters (18) of the drawn $\tilde{X}$-pair as the key.

Mapping a surflet pair onto another in step 3 requires trading off between positional and directional information, especially for estimating the rotation. Unlike for pure point sets, there is no unique principled formulation of a cost function. In this study, we estimated the rotation from the surface normals alone, while the translation has to be estimated from the surface points. More precisely, the rotation $\bar{R} \in SO(3)$ between the two surflet pairs $\{(x_1, m_1), (x_2, m_2)\} \subset \tilde{X}$ and $\{(y_1, n_1), (y_2, n_2)\} \subset \tilde{Y}$ was computed to minimize the squared angles between the normals, i.e.,

$$\bar{R} = \arg \min_{R \in SO(3)} \sum_{i=1}^{2} \arccos^2(n_i \cdot R(m_i)) \ , \tag{19}$$

and the translation is then the least-squares solution on the points, i.e.,

$$\bar{t} = \arg\min_{t \in \mathbb{R}^3} \sum_{i=1}^{2} \|\bar{R}(x_i) + t - y_i\|^2 \ . \tag{20}$$

The solution to (19) is obtained by first rotating to align the planes spanned by (m_1, m_2) and (n_1, n_2), followed by the appropriate in-plane rotation. The solution to (20) is given by

$$\bar{t} = \frac{1}{2} \sum_{i=1}^{2} \big(y_i - \bar{R}(x_i)\big) \ . \tag{21}$$

Like for sampling from point triples, a pose hypothesis computed from neighboring surflets is more sensitive to noise, and hence less reliable, than one computed from surflets further apart. This is because neighboring surface points usually have similar surface normals, and orientation is poorly conditioned on almost parallel vectors. As for point triples, however, surflet pairs with very small distances are less frequent than those with larger distances, and are hence naturally suppressed through random sampling.

It is interesting to note that surflet pairs have previously been shown to be very informative as pose-invariant shape descriptors [41]. There only the intrinsic geometry of surflet pairs is evaluated, while here we exploit both the intrinsic and extrinsic aspects[6].

3.2 Parameter density maximization

Significant populations of points in Y matching a rigid motion of the points X will produce many parameter samples $\alpha = (\alpha_{\text{rot}}, \alpha_{\text{trans}}) \in \mathbb{R}^6$ that coincide approximately. The goal of parameter clustering is hence to estimate the location in the parameter space of the maximum probability density underlying the obtained parameter samples $\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$; cf. eq. (8). A practical realization, derived from kernel density estimation, is through the mean-shift procedure [42, 43]. In brief, a sequence of pose parameters $\alpha^1, \alpha^2, \ldots$ is obtained through iterative weighted averaging

$$\alpha^k = \frac{\sum_{i=1}^{N} w_i^k \, \alpha_i}{\sum_{j=1}^{N} w_j^k} \ , \tag{22}$$

$$w_i^k = u\big(\|\alpha_{\text{rot}}^{k-1} - \alpha_{\text{rot},i}\|/\delta_{\text{rot}}\big) \, u\big(\|\alpha_{\text{trans}}^{k-1} - \alpha_{\text{trans},i}\|/\delta_{\text{trans}}\big) \ . \tag{23}$$

[6] Intrinsic refers to the properties that are independent of the choice of a coordinate system, i.e., the point relations, while extrinsic refers to properties that depend on a coordinate system, i.e., the pose.

Here u is a unit step function,

$$u(\delta) = \begin{cases} 1 & \text{if } \delta < 1, \\ 0 & \text{else,} \end{cases} \tag{24}$$

such that the averaging procedure (22) operates just on a $B_{\delta_{\text{rot}}}(\alpha_{\text{rot}}^{k-1}) \times B_{\delta_{\text{trans}}}(\alpha_{\text{trans}}^{k-1})$-neighborhood of α^{k-1}. The required parameter samples can be efficiently retrieved from the bins indexed by α^{k-1} and its neighbors; cf. section 3.1. The radii δ_{rot} and δ_{trans} of the rotational and translational extensions, respectively, of the averaging procedure derive from the kernel bandwidth of the underlying kernel density estimate. Their choice affects the density peaks that are actually found. There are data-driven techniques for adapting the kernel bandwidth [43]. Here, however, we set all algorithmic parameters to a range of values and find the combination that achieves the lowest estimation error; cf. section 4. This way we keep out any effects of parameter adaptation techniques from the comparison of achievable estimator performance.

The sequence α^k converges to an estimate of the position of a local density maximum [43], even though the density of parameters is not explicitely estimated. By starting with α^0 close to the dominant mode of the density, the sought pose estimate $\hat{\alpha} = \lim_{k\to\infty} \alpha^k$ is thus obtained. The region of the dominant mode, in turn, is estimated through counting of pose parameters in the bins. To cope with quantization effects of the bin counting, the mean-shift procedure is started from several bin centers that have obtained high parameter counts. Thus, first starting from the bin having the highest parameter count, mean shift is repeatedly started from the bin with the next highest parameter count, until the density found in the converged mean-shift window is significantly lower, i.e., more than one standard deviation of a binomial distribution lower, than found in the first run of mean shift. From all the local density maxima found through mean shift, the location $\hat{\alpha}$ in the 6D parameter space of the largest maximum is returned as the pose estimate.

One of the themes of this study is the comparison of the estimation performance achieved through density maximization of canonical pose parameters and density maximization of consistent pose parameters, which both will now be defined.

3.2.1 Canonical rotation space

For pose clustering, we want a parameter space $\mathcal{P}$ that does not have redundant dimensions (such as quaternions have), so it should be six dimensional,

and the region relevant to the estimation problem should be bounded. This will make the best use of the working memory on computers. There should also be, at least locally, a one-to-one relation between parameters and motions, since spreading parameter samples along null spaces (as for Euler angles) would surely spoil the location statistics. From the classical parameterizations of rotations, the only one that meets these requirements is the canonical parameterization[7]; see, e.g., [44]. The canonical parameters $\alpha_{\mathrm{rot}} = (\alpha_1, \alpha_2, \alpha_3) \in \mathbb{R}^3$ are related to rotations through the exponential function of operators,

$$R_{\alpha_{\mathrm{rot}}} = \exp(\alpha_1 \Lambda_1 + \alpha_2 \Lambda_2 + \alpha_3 \Lambda_3) \ , \tag{25}$$

where $\Lambda_1, \Lambda_2, \Lambda_3$ are infinitesimal rotations about three orthogonal axes. In fact, $\|\alpha_{\mathrm{rot}}\| \in [0, \pi]$ is the angle and $\alpha_{\mathrm{rot}}/\|\alpha_{\mathrm{rot}}\|$ the oriented axis of the rotation $R_{\alpha_{\mathrm{rot}}}$. This parameterization is one-to-one within the sphere $\{\alpha_{\mathrm{rot}} \in \mathbb{R}^3 \,|\, \|\alpha_{\mathrm{rot}}\| < \pi\}$. Rotations with parameters $\|\alpha_{\mathrm{rot}}\| = \pi$ are identical to those with parameters $-\alpha_{\mathrm{rot}}$.

The cyclic topology of the rotation group has to be taken care of when locating the pose-density maximum through the mean-shift algorithm. Whenever the mean-shift window defined by eq. (23) crosses the boundary of the parameter sphere, neighboring parameter samples from the antipodal side need to be considered in the computation of the local average in eq. (22). For computing this local average, the relevant antipodal rotations with angle $\|\alpha_{\mathrm{rot}}\| = \pi - \delta$ about axis $\alpha_{\mathrm{rot}}/\|\alpha_{\mathrm{rot}}\|$ need to be flipped to the parameters α'_{rot} with $\|\alpha'_{\mathrm{rot}}\| = \pi + \delta$ and $\alpha'_{\mathrm{rot}}/\|\alpha'_{\mathrm{rot}}\| = -\alpha_{\mathrm{rot}}/\|\alpha_{\mathrm{rot}}\|$.

3.2.2 Consistent rotation space

Like all the other classical, non-redundant parameterizations of rotations, however, the canonical parameterization is not consistent for clustering. In fact, from an isotropic data distribution, cf. eq. (9), one obtains the canonical parameter density

$$\rho_{\mathrm{inv}}(\alpha_{\mathrm{rot}}) \propto \left[\frac{\sin(\|\alpha_{\mathrm{rot}}\|/2)}{\|\alpha_{\mathrm{rot}}\|}\right]^2 \ ; \tag{26}$$

cf. eq. (10); for a plot see the top of fig. 1. Evidently, there is a strong bias towards small rotation angles, and condition (11) is violated. One can expect, and it will below be demonstrated, that this fundamental bias can show up in pose estimates computed through clustering.

[7] Also referred to as exponential coordinates, rotation vector, or Euler parameters.

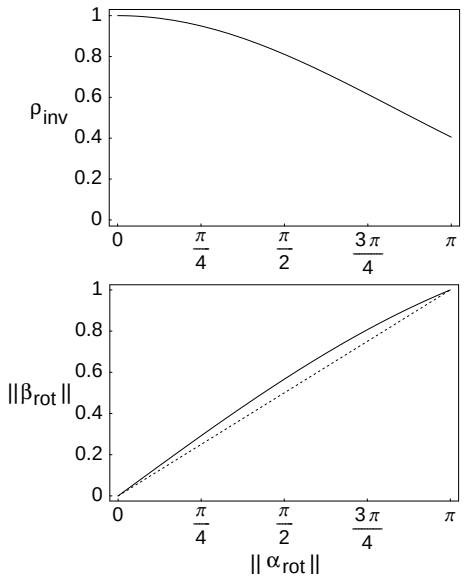

Figure 1: *Top:* angular section of the isotropic parameter density (26); the density decreases with growing rotation angle; the maximum density is assumed at $\alpha_{\mathrm{rot}} = 0$ (corresponding to the identity transform) which is here normalized to one. *Bottom:* consistently mapped angle $\|\beta_{\mathrm{rot}}\|$ vs. rotation angle $\|\alpha_{\mathrm{rot}}\|$ as given by eq. (27); the straight dashed line indicates a pure rescaling of the angle and is drawn for comparison. The isotropic parameter density ρ_{inv} is uniform when expressed as a function of the consistent parameters β_{rot}.

A uniform parameter density ρ_{inv} is obtained from an isotropic data distribution through introducing a non-linear angle mapping. Thus, consistent parameters $\beta_{\text{rot}} = (\beta_1, \beta_2, \beta_3) \in \mathbb{R}^3$ that also retain the desirable properties of the canonical parameters are given by

$$\beta_{\text{rot}} = \left(\frac{\|\alpha_{\text{rot}}\| - \sin \|\alpha_{\text{rot}}\|}{\pi} \right)^{1/3} \frac{\alpha_{\text{rot}}}{\|\alpha_{\text{rot}}\|} \; ; \tag{27}$$

see the bottom of fig. 1 for a plot. As in canonical parameters, $\beta_{\text{rot}}/\|\beta_{\text{rot}}\| = \alpha_{\text{rot}}/\|\alpha_{\text{rot}}\|$ is the oriented axis of the rotation. The consistent parameterization is one-to-one within the unit sphere $\{\beta_{\text{rot}} \in \mathbb{R}^3 \mid \|\beta_{\text{rot}}\| < 1\}$. Rotations with parameters $\|\beta_{\text{rot}}\| = 1$ are identical to those with parameters $-\beta_{\text{rot}}$.

As for the canonical rotation parameters, for computing the local average (22), rotations need to be flipped over from the antipodal side when the mean-shift window crosses the boundary of the parameter sphere.

We note that consistent parameterizations for clustering depend only on the group $\{G_\alpha \mid \alpha \in \mathcal{P}\}$ in the transformation (1) and not on the function F. In particular, if the data set Y were 2D image points (instead of our 3D range data), (27) would still define a consistent parameterization for clustering rotations.

3.2.3 Translation space

The group of translations has a simpler topology than the rotations. In fact, translations $t \in \mathbb{R}^3$ are consistently parameterized simply by their three vector components,

$$\alpha_{\text{trans}} = \beta_{\text{trans}} = t \; . \tag{28}$$

Also the combined parameterization $\beta = (\beta_{\text{rot}}, \beta_{\text{trans}}) \in \mathbb{R}^6$ of the Euclidian group can be shown to be consistent for clustering [1].

4 Experiments

This section describes in detail the critical algorithmic parameters along with the values they took for this study, the test data set, and the error statistics computed from the pose estimates. All relevant results are here compiled and interpreted.

4.1 The test algorithms

Table 1 lists the critical parameters of the algorithms for pose clustering with an explanation and the set of values used in the present study. The parameters are related to quantization of the hash table and the parameter space, and window size for the mean-shift procedure. Length dimensions are given in units of the longest edge of the test objects' bounding box, which was always scaled to one; see section 4.2.

The mean-shift window radii $\delta_{\text{rot}}, \delta_{\text{trans}}$, cf. section 3.2, were always set to the respective size of the bins of the parameter space, such that these parameters were not varied independently. Similarly, for the hash table for surflet pairs, there was a relation between the inter-normal-angle quantization d_{ang} and the point-difference quantization d_{diff}, cf. section 3.1.2, based on an estimate of relative accuracy of the angular and difference measures of surflet pairs [35]. These dependencies had to be chosen to limit the number of runs of the algorithm, and hence computation time, in this study, while focusing on a reasonable parameter regime. Moreover, the value n_{samples} of the maximum filling of a single parameter bin was chosen to assign a one-byte counter to each bin and, hence, limit the memory required for a run of the algorithm. Since sampling stops as soon as any bin has accumulated n_{samples} parameter samples, this parameter determines the amount of samples generated. Increasing n_{samples} can improve the result of estimation, however, at the cost of higher computation time and memory.

The combination of the smallest translational quantization $\delta_{\text{trans}} = 0.1$ with the two smallest rotational quantizations $\delta_{\text{rot}} = 0.04(\pi), 0.08(\pi)$ (factor π for canonical parameterization) were not feasible due to limitation of the working memory to 4 GB. All other combinations of parameters were realized, resulting in 30 parametric variants of each of the four clustering types, that is, sampling point triples or surflet pairs combined with clustering in canonical or consistent parameter space. The parametric variants should represent the working regime of the algorithms. In the experiments, we scan through all the combinations of algorithmic parameters. When comparing clustering types, only the best parameterization for each data type – i.e., synthetic data with specific level of data corruption, real scenes – is taken into account; cf. section 4.3. This way, we are attempting to compare the best achievable performance of the four clustering types. Of course, we cannot be sure how close we actually get to the real optimal algorithmic parameters in our experiments.

Table 1: Parameters of the pose clustering algorithm.

Parameters	**Meaning**	**Values**
c	type of pose clustering	{sampling point triples, sampling surflet pairs} $\times$ {canonical parameterization, consistent parameterization}
d_{dist}	quantization of three point distances in hash table for point triples; cf. eq. (13)	$\{0.01, 0.05, 0.1\}$
$(d_{\text{ang}}, d_{\text{diff}})$	quantization of angle between normals and three difference components between points in hash table for surflet pairs; cf. eq. (18)	$\{(\frac{\pi}{4}, 0.06), (\frac{\pi}{3}, 0.08), (\frac{\pi}{2}, 0.12)\}$ $(\frac{d_{\text{ang}}}{d_{\text{diff}}} = \text{const.})$
δ_{rot}	i) quantization of three rotational dimensions of parameter space for sample binning ii) rotational radius of mean-shift window for density maximization; cf. eq. (23)	canonical parameterization: $\{0.04\pi, 0.08\pi, 0.13\pi, 0.18\pi\}$ consistent parameterization: $\{0.04, 0.08, 0.13, 0.18\}$
δ_{trans}	i) quantization of three translational dimensions of parameter space for sample binning ii) translational radius of mean-shift window for density maximization; cf. eq. (23)	$\{0.1, 0.3, 0.5\}$
n_{samples}	number of samples that must be reached in some parameter bin for sampling to stop	255

4.2 The test data

4.2.1 Synthetic data from single objects

The synthetic data were based on object models from the Princeton Shape Benchmark [36, 37], whose database contains 1814 3D models in polyhedral form, both synthetic and measured. The original benchmark has been designed for shape retrieval algorithms. The model database, however, is a valuable resource for testing various kinds of range-data processing algorithms. The advantages of using synthetic data are

- ground truth for object pose and surface: exact error measures are readily available;
- controllability: the effects of specific disturbances on an algorithm can be studied;
- size of the generated test data set (see below): it is hardly possible for a research team to capture a comparably sized data set in their lab.

In order to discard object models inadequate for our evaluation and to limit computation time for this study to a manageable amount, a useful and representative subset of objects was selected from the database as follows. The database was divided into shape classes according to two criteria:

- surface dimensionality: mainly 1D extended (stick like) objects, mainly 2D extended (flat) objects, 3D objects with mainly flat faces (polyhedral with few vertices), 3D objects with mainly curved surfaces (polyhedral with many vertices);
- rotational symmetry: continuously symmetric (such as a cylinder), discretely symmetric (such as a cube), almost symmetric (large fraction of symmetric surface), weakly symmetric (small fraction of symmetric surface), not symmetric.

The number of objects from the database in each of the resulting shape categories is listed in table 2. Rotational symmetry of an object precludes a unique estimate of its orientation. Therefore, only objects without or just weak symmetry were considered in this study. Moreover, very flat, mainly 2D extended objects often have thin parts consisting of just one surface layer, not enclosing any volume. Those parts disappear under occlusion when viewed from the wrong side. Hence, from the 3D extended objects with at most weak symmetry, 10 with flat faces and 10 with curved faces were

Table 2: Number of objects in the Princeton shape database falling in different shape categories.

	mainly 1D	mainly 2D	3D flat	3D curved
not symmetric	0	22	156	500
weakly symmetric	0	48	145	467
almost symmetric	2	47	53	70
discretely symmetric	15	22	107	57
continuously symmetric	14	0	2	87

selected, resulting in a useful and representative set of 20 objects. Figure 2 shows the selected objects from the Princeton shape database. The longest edge of the objects' bounding box was scaled to unit length, so as to make length scales comparable across objects.

For each run of the algorithm, two point sets were generated from an object model, a model point set and a scene point set. The latter is meant to mimic a range measurement of that object. Clearly, different types of range sensors, surface properties, and sensing conditions would produce different point sets [2, 3, 5]. In this study, however, we simulate a generic measurement process that does not capture any specific effects.

A model point set was obtained through centering the object model, such that the origin was at the centroid of the model vertices, followed by uniform sampling of the model surface with a density of 10,000 points per unit area. This resulted in a few 100 to a few 10,000 points, depending on the area of the object surface. For obtaining a scene point set, the same procedure was applied, followed by random motion and subsequent data corruption.

For the random motion, a rotation was drawn uniformly from the quaternion unit sphere[8], while a translation was drawn uniformly from a cuboid[9] just to avoid any artifacts from the translation space quantization; cf. section 3.1. Otherwise the size of translation is irrelevant for this study: adding a translation t to a data set will shift the translation estimate precisely by t, leaving the rotation estimate unchanged. The motion is specified by first rotating followed by translating the data points.

The scene data were corrupted by a combination of three processes:

- adding isotropic Gaussian 3D noise to each point; see fig. 3;

[8]This way no direction or angle was preferred.

[9]Cuboid of twice the size of the object bounding box in each dimension, centered at zero.

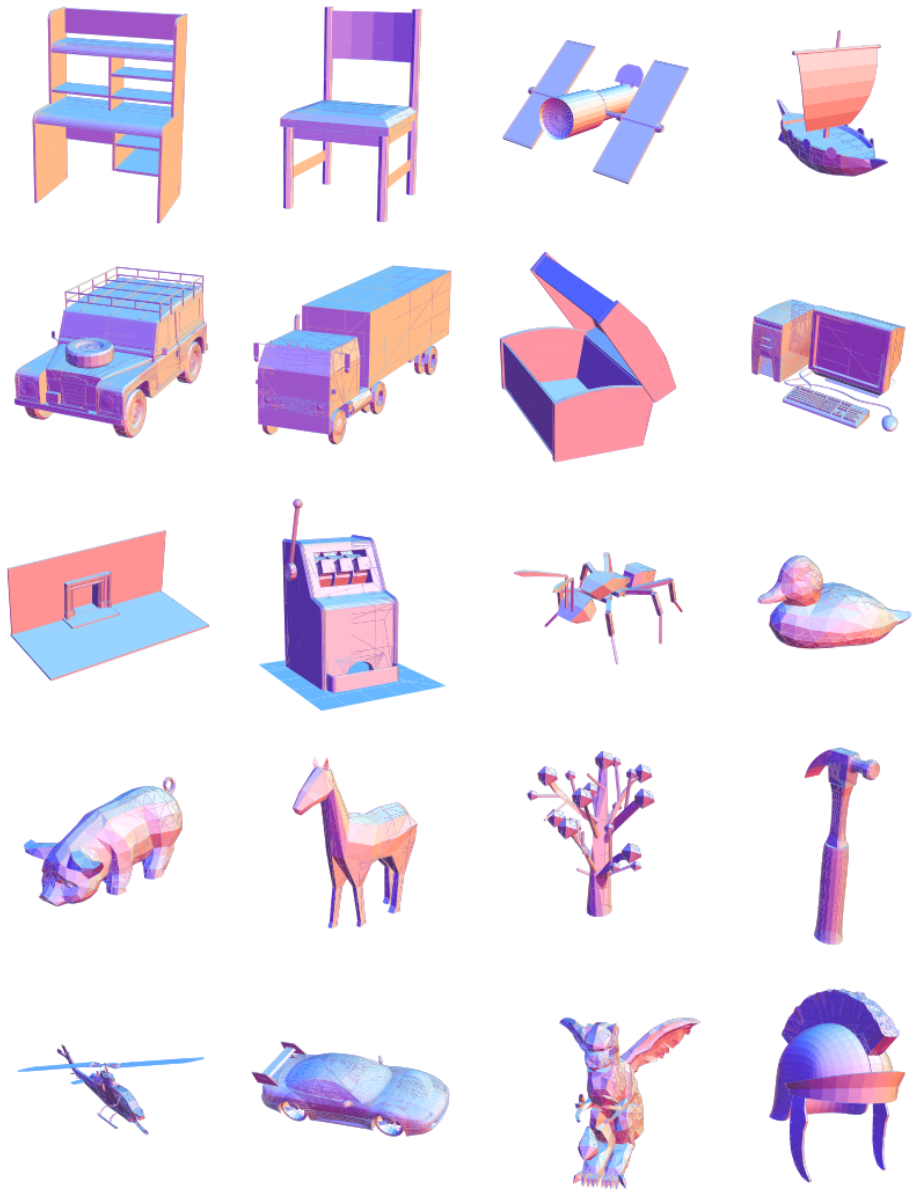

Figure 2: The 20 selected objects from the Princeton shape database [37]. The first 10 objects belong to the shape category with mainly flat faces, the second 10 to the shape category with mainly curved surfaces.

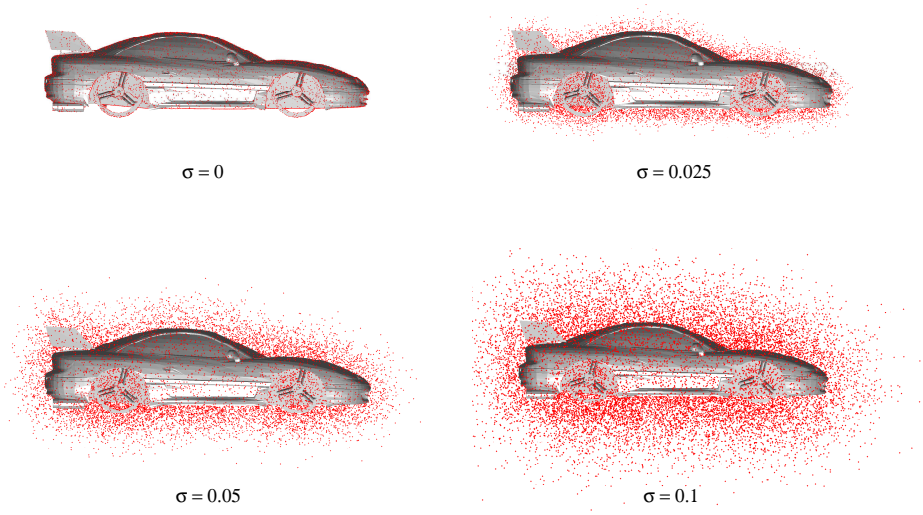

Figure 3: Example object (car, gray surface) from the test set and synthetic data points (red dots) generated with the four levels of additive Gaussian noise used in this study, here without random points or occlusion; cf. table 3.

- adding random points uniformly in a region around the object; this region is defined by the object's bounding box extended in each direction by the mean of the three bounding box extensions;
- randomly choosing a viewing direction and removing occluded object points.

Occlusion is a corruption that results from measuring the object from a single viewpoint only. When fusing multiple such measurements, however, a complete representation of the object surface can be obtained. Both conditions, i.e., with and without occlusion, were included in the test data set. A summary of parameters of the synthetic data set is listed in table 3. The combinations of the data parameters resulted in a test data set of 24,000 scenes.

4.2.2 Real scene data

Another sequence of experiments was run on a public set of real range data captured with a Minolta Vivid 910 scanner [8, 9, 38]. The set comprises 50 2.5D scenes composed of four to five different objects, where complete 3D

Table 3: Parameters of the synthetic data set.

Parameters	**Meaning**	**Values**
m	model number from the Princeton shape database	flat faces: $\{769, 820, 1389, 1454, 1488, 1573, 1703, 1773, 1799, 1803\}$ curved faces: $\{0, 48, 100, 105, 1077, 1110, 1309, 1548, 1622, 1639\}$
o	flag for removal of occluded surface points	$\{\text{no occlusion}, \text{with occlusion}\}$
(R^*, t^*)	i) rotation and translation of the scene object relative to the model ii) ground truth of parameters to be estimated	50 random values, uniformly distributed
ν	fraction of random points	$\{0, 0.2, 0.7\}$
σ	standard deviation of additive isotropic Gaussian 3D noise	$\{0, 0.025, 0.05, 0.1\}$

models and ground truth poses are available for four of them. These four object models were sought in all the scenes where they are present, resulting in 188 cases of pose estimation in cluttered scenes. Figure 4 shows the four object models and an example scene.

The geometric representation of the scenes obtained with the scanner is of a much higher quality than most of our synthetic test data. The challenge in the estimation problems on these real scenes is thus complementary to the one of estimating pose on the synthetic single object data: for the latter, the main difficulty arises from the heavy geometric degradation, while here the problems are clutter and occlusion. Following [8, 9], we characterize each case of pose estimation on the real scenes by the occlusion ratio of the respective object in the respective scene, that is,

$$\text{occlusion} = 1 - \frac{\text{visible object surface area}}{\text{total object surface area}} \ . \tag{29}$$

4.3 Error statistics

4.3.1 Synthetic data from single objects

Estimation of object pose was performed for all combinations of algorithmic variants, cf. section 4.1, and data corruptions, cf. section 4.2, yielding a total

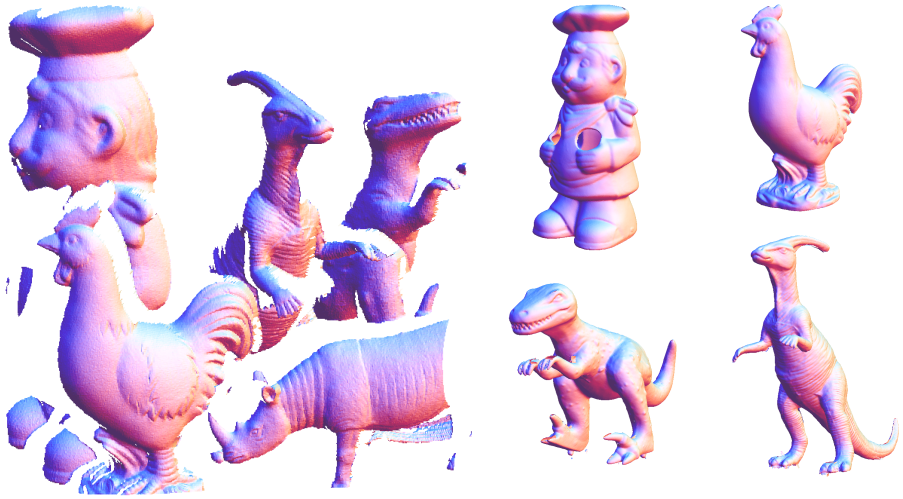

Figure 4: One of the 50 test scenes (left) with real data and the four object models (right) sought in those scenes; data taken from [38].

of 57,600 distinct estimation scenarios. For each of these scenarios, 50 repetitions with random pose parameters were computed, requiring 2,880,000 runs in total of a pose clustering algorithm.

For each run, three error measure of the pose estimate were computed. The *rotational error* was measured as the angle of the difference rotation between the estimate $\hat{R}$ and the ground truth R^*,

$$e_{\text{rot}} = \text{ang}\big(\hat{R}^{-1} \circ R^*\big) \in [0, \pi] \ . \tag{30}$$

Likewise, the *translational error* was measured as the Euclidian norm of the difference between the ground truth t^* and the estimate $\hat{t}$,

$$e_{\text{trans}} = \|\hat{t} - t^*\| \in \mathbb{R}^+ \ . \tag{31}$$

An error measure that combines rotational and translational errors is the square root of the average squared surface distance, to be simply called *distance error*. Let $T(x, R, t) = R(x) + t$ be the motion with rotation R and translation t for a point $x \in S$ on the model surface $S \subset \mathbb{R}^3$. The distance error then is

$$e_{\text{dist}} = \sqrt{\frac{\int_S dA(x)\, \|T(x, \hat{R}, \hat{t}) - T(x, R^*, t^*)\|^2}{\int_S dA(x)}} \in \mathbb{R}^+ \ , \tag{32}$$

where $dA(x)$ is the infinitesimal surface element at $x \in S$. Since the models of the Princeton Shape Benchmark are given as triangular surface meshes, the integrals are computed straightforwardly. The distance error is a measure of the Euclidian deviation between the true and estimated object surfaces. As such, it is often more relevant than rotational and translational errors for applications like robotic manipulation.

The error measures (30), (31), (32) were obtained for 50 repetitions for each of the 10 test objects with flat faces and the 10 test objects with curved faces, yielding 500 samples for each error measure and shape category. These sets of error samples were collected for all data corruption parameters (σ, ν, o), cf. table 3, clustering types c, and algorithmic parameters $a = (d_{\text{dist}}, \delta_{\text{rot}}, \delta_{\text{trans}}, n_{\text{samples}})$ or $a = (d_{\text{ang}}, d_{\text{diff}}, \delta_{\text{rot}}, \delta_{\text{trans}}, n_{\text{samples}})$, depending on the clustering type, cf. table 1. Let the corresponding sets of error samples be denoted by $\mathcal{E}_{\text{rot}}^{(a,c,\sigma,\nu,o)}$, $\mathcal{E}_{\text{trans}}^{(a,c,\sigma,\nu,o)}$, $\mathcal{E}_{\text{dist}}^{(a,c,\sigma,\nu,o)}$, respectively.

Being a stochastic process, pose clustering may fail on individual trials, yielding random estimates of object pose. Clearly, an average rate of failure would be a desirable quantity to compute; however, this would require to define precisely what is a failure, which in turn is quite arbitrary in the absence of a specific application context. So instead, we have computed the mean and the median over the error sets, noting that the former will be strongly affected by individual failures of estimation in the sample, while the latter shows the largest error from the better half of results. The median will hence stay low, as long as at least half of the estimation trials succeeded, quantifying the accuracy achieved among successful trials.

We are here interested in the relative estimation accuracy and robustness of the four clustering types c, depending upon the level of data corruption (σ, ν, o). This is represented by the smallest error statistics achievable over all algorithmic parameters a, hence,

$$E_{\text{rot}}^{(c,\sigma,\nu,o)} = \min_a \left[\text{mean/median}\, \mathcal{E}_{\text{rot}}^{(a,c,\sigma,\nu,o)} \right] , \tag{33}$$

$$E_{\text{trans}}^{(c,\sigma,\nu,o)} = \min_a \left[\text{mean/median}\, \mathcal{E}_{\text{trans}}^{(a,c,\sigma,\nu,o)} \right] , \tag{34}$$

$$E_{\text{dist}}^{(c,\sigma,\nu,o)} = \min_a \left[\text{mean/median}\, \mathcal{E}_{\text{dist}}^{(a,c,\sigma,\nu,o)} \right] . \tag{35}$$

The minimum of the mean or median error is taken over the 30 combinations of algorithmic parameters described in section 4.1 and table 1.

Statistical estimation errors may generally contain a bias and a variance component. In sections 2.2 and 3.2.2 we have argued that with the canonical parameterization of motions one can expect a bias of estimates towards too

small rotation angles, while no such bias should exist with a consistent parameterization.

In order to quantitatively investigate the angle bias, the difference between estimated and true rotation angles was computed, i.e.,

$$\delta = \mathrm{ang}\big(\hat{R}\big) - \mathrm{ang}\big(R^*\big) \in [-\pi, \pi] \ , \tag{36}$$

for all the repetitions and test objects, and for each set of data corruption parameters (σ, ν, o), clustering type c, and set of algorithmic parameters a. Let the corresponding sets of angle-difference samples be denoted by $\mathcal{D}^{(a,c,\sigma,\nu,o)}$. The angle bias for each data and algorithmic variant is the mean of each set, that is,

$$B_{\mathrm{ang}}^{(a,c,\sigma,\nu,o)} = \mathrm{mean}\, \mathcal{D}^{(a,c,\sigma,\nu,o)} \ . \tag{37}$$

For all mean values, we have also computed their standard deviation. For the error medians, a bootstrap estimate of their standard deviation was obtained from 1000 re-sampled error sets.

4.3.2 Real scene data

In the real range data, each sought object is represented in each scene with a unique occlusion ratio that varies strongly between scenes. We therefore cannot provide an error statistics at fixed occlusion ratios, analogous to what we do for the synthetic data at fixed levels of data corruption. Instead, we will provide scatter plots of the individual estimation errors (30), (31), (32), and of the angle differences (36), together with the individual occlusion ratio of the respective object in the respective scene.

For comparing errors and angle differences, we have to select from the 30 parametric variants of all pose clustering algorithms that we run on the data, as described in section 4.1 and table 1. Analogous to the procedure for the synthetic data (cf. eqs. (33), (34), (35)), we determined for each of the four clustering types the parametric variant that performed best in terms of mean and median errors over all 188 cases of pose estimation. It turned out that the same parameterization achieved the minimum mean and median errors for all clustering types, so we have chosen this variant for plotting all results on the real scene data.

It is not possible to compare our results on pose estimation errors to those of [8, 9] on the same data set, as those authors have evaluated their system for object recognition, where alignment is just one of several steps, publishing recognition rates rather than alignment errors.

4.4 Results

4.4.1 Synthetic data from single objects

Figures 5 through 10 show the plots of the mean and median error measures (33), (34), and (35), as functions of the standard deviation σ of additive Gaussian noise. Likewise, fig. 11 shows the plots of the angle bias. In this section, we discuss these error and bias plots, focusing mainly on comparing the four types of pose clustering: sampling point triples or surflet pairs, each combined with clustering in canonical or consistent pose space.

The influence of random points, quantified by their fraction ν, is generally much weaker than the effect of additive Gaussian noise, quantified by its standard deviation σ, or the effect of occluding part of the data. In fact, all four types of clustering are extremely robust to addition of random point outliers. The reason is most probably that the random points do not have any spatial structure that could interfere with the object models, such that they are not very distractive for the sampling process. Hence, unless stated otherwise, the observations below apply regardless of the random-point fraction ν.

The mean estimation errors are often a lot higher than the corresponding median error. A small median error with a large mean error indicates an error distribution that has most of its weight at small errors, but with a long tail of larger errors, resulting from occasional failures of estimation. In fact, if an estimator fails on an individual run, the associated error can take any value from the possible range. As a measure of typical estimation errors, the median error then is of more interest than the mean error. Robustness of the estimators is here understood as the degree of resistance, exhibited in the stability of mean and median errors, to the various kinds of data corruption; see also the discussion in section 5.1.

Pose clustering for flat-faced versus curved objects

The two shape classes, objects with flat and with curved surfaces, exhibit certain differences. While at low noise and without occlusion all errors are similar across the two kinds of shapes, at higher noise or with occlusion the flat-faced objects generally present less problems to pose clustering than the curved objects. A notable exception to this pattern is for sampling point triples and clustering in canonical parameter space: this type of algorithm apparently has unusual problems at low noise and low random-point fraction with flat-faced occluded objects. Indeed, all three mean error measures are

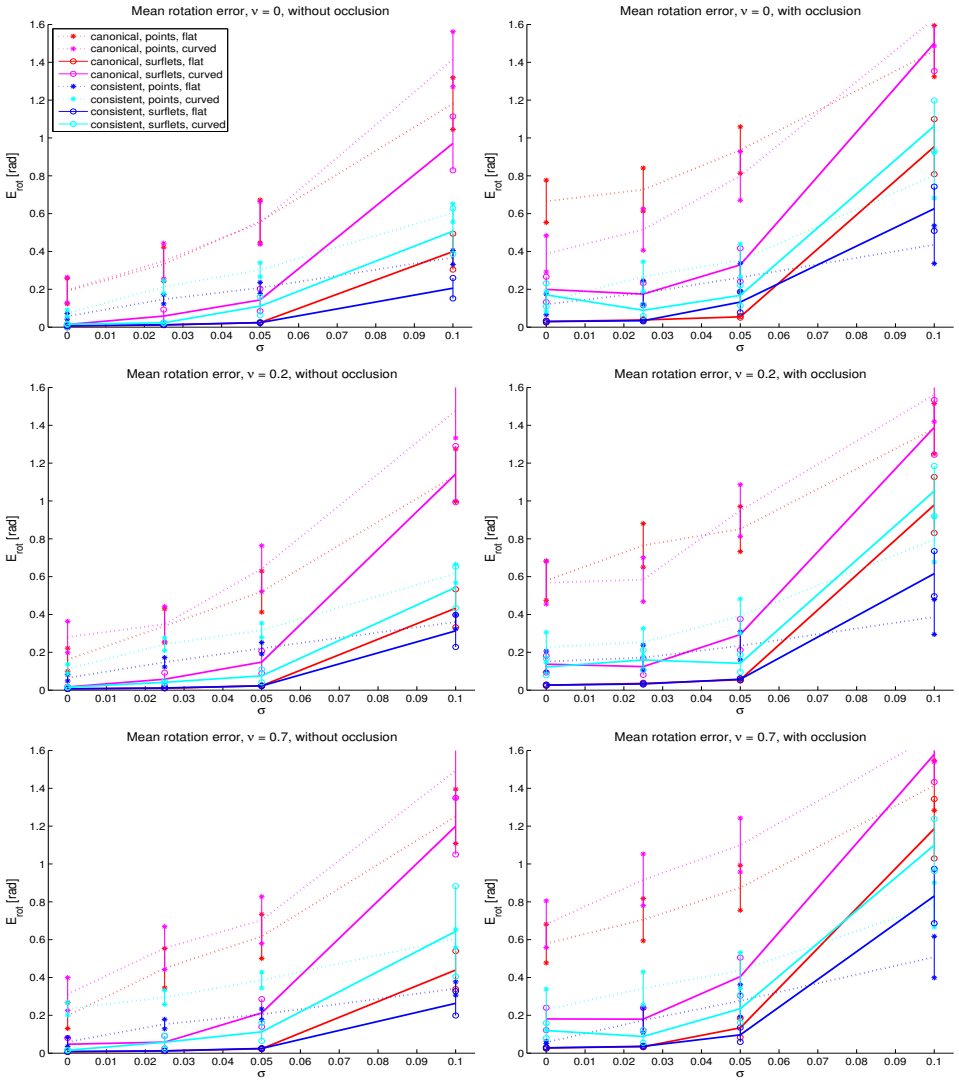

Figure 5: Plots of the mean rotation error measure (33) as function of the standard deviation σ of additive Gaussian noise for three random-point fractions ν, two object shape classes, and four types of pose clustering. Length dimensions are given in units of the longest edge of the test objects' bounding box. Vertical bars indicate the standard deviation of the mean error.

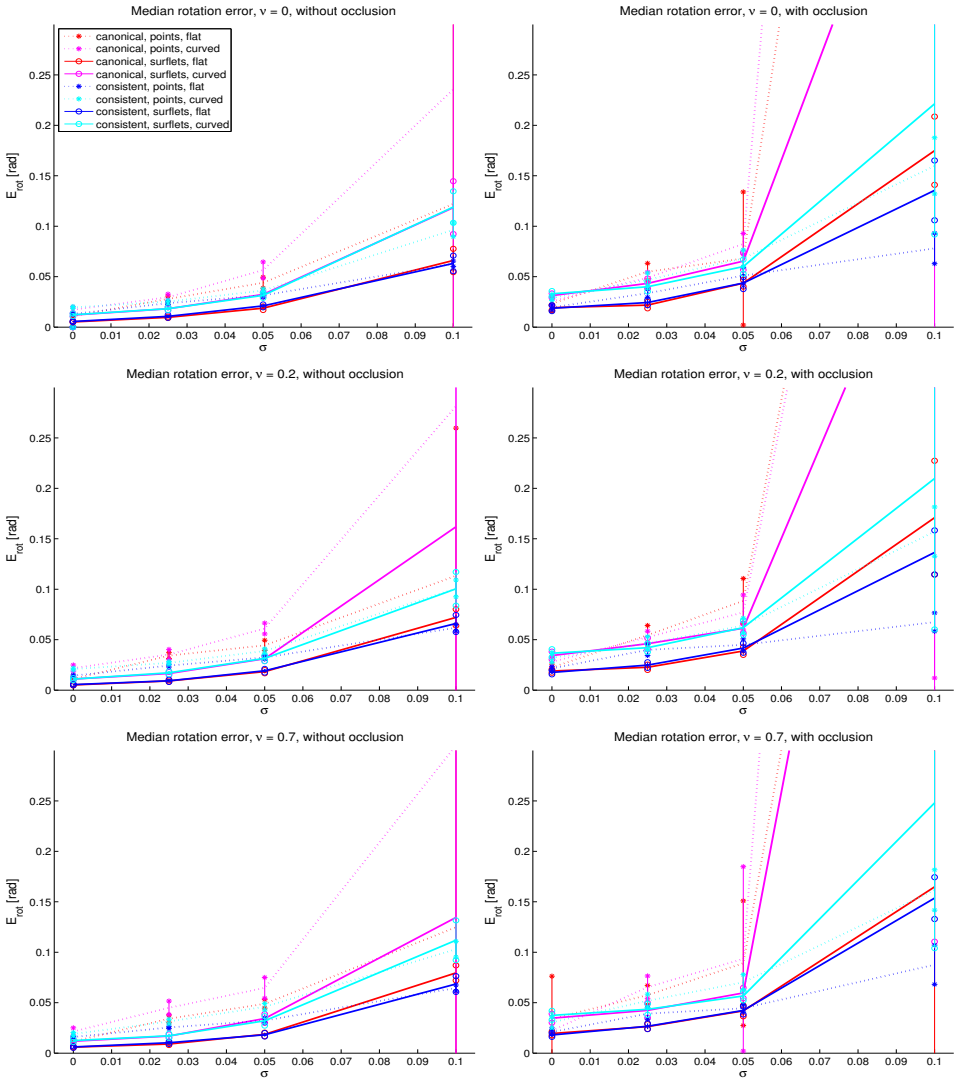

Figure 6: Plots of the median rotation error measure (33) as function of the standard deviation σ of additive Gaussian noise for three random-point fractions ν, two object shape classes, and four types of pose clustering. Length dimensions are given in units of the longest edge of the test objects' bounding box. Vertical bars indicate the standard deviation of the median error.

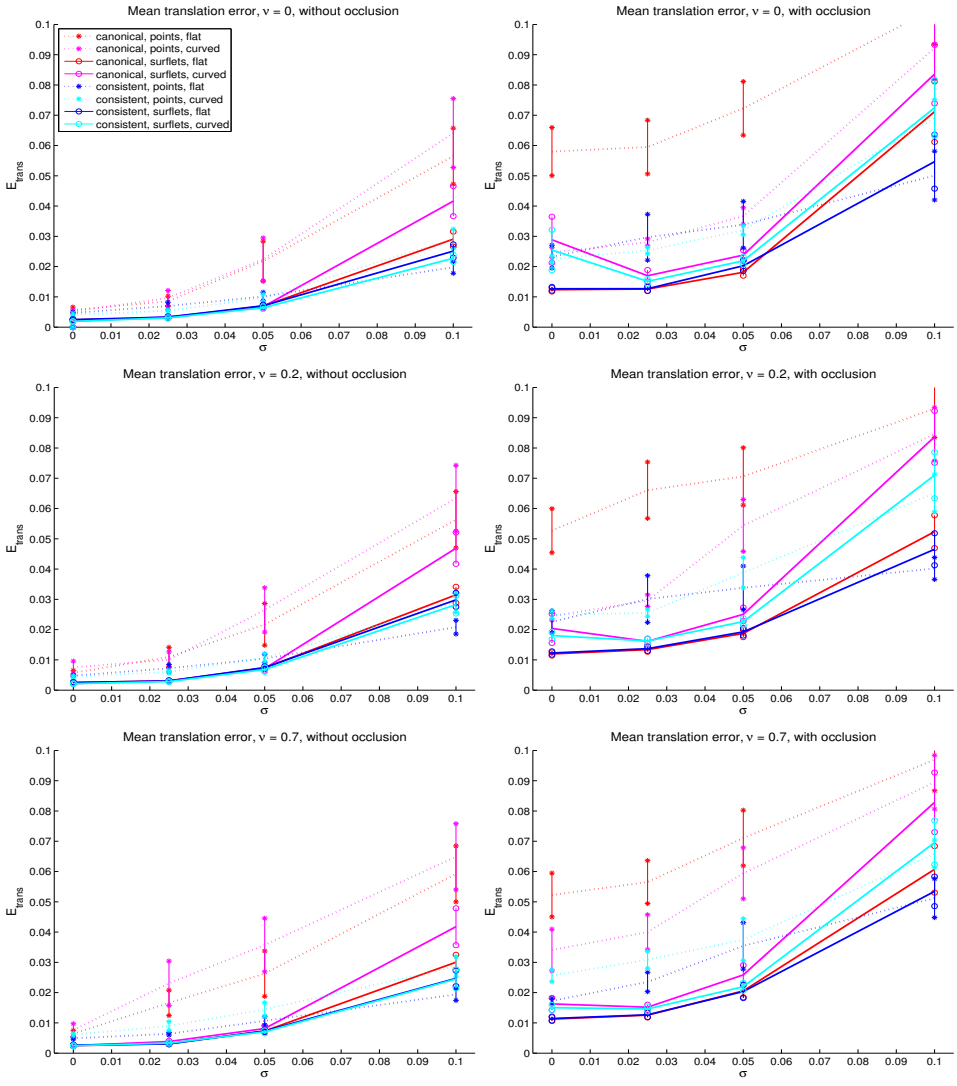

Figure 7: Plots of the mean translation error measure (34) as function of the standard deviation σ of additive Gaussian noise for three random-point fractions ν, two object shape classes, and four types of pose clustering. Length dimensions are given in units of the longest edge of the test objects' bounding box. Vertical bars indicate the standard deviation of the mean error.

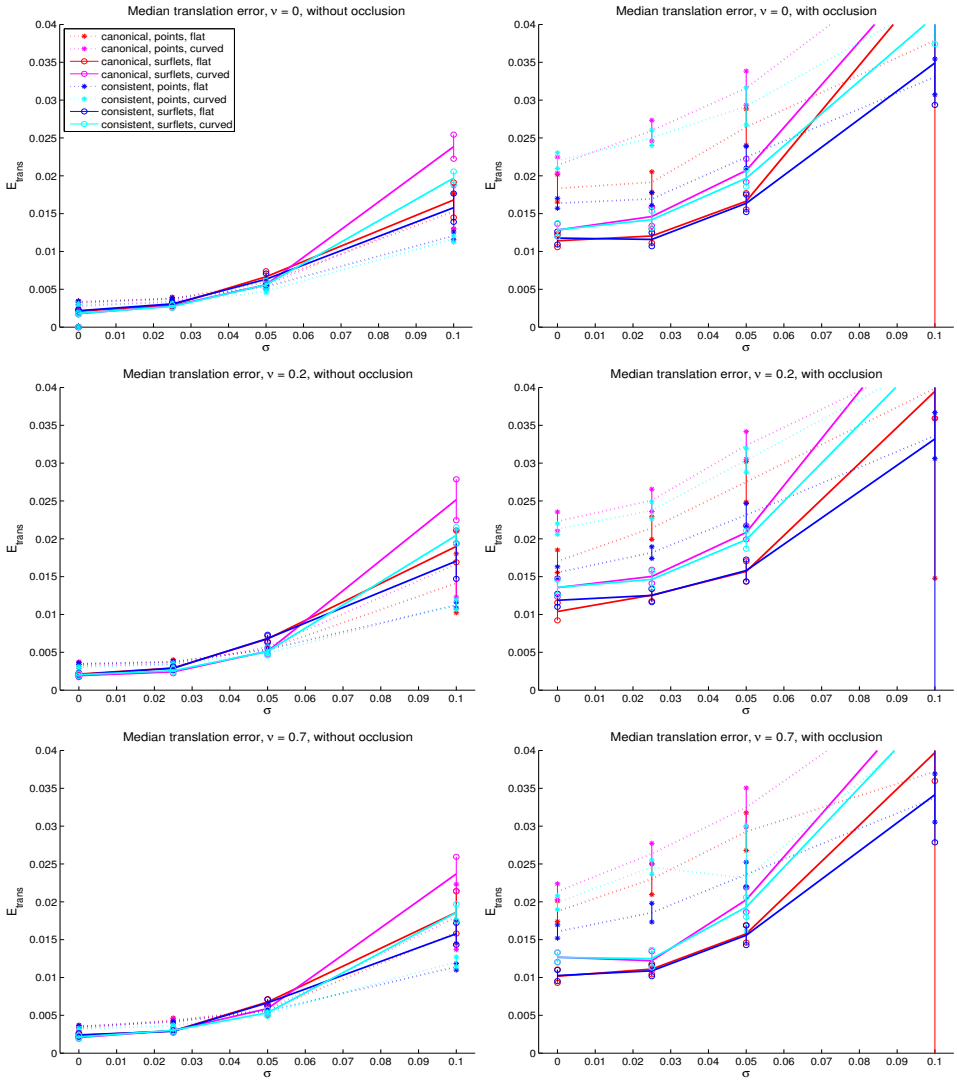

Figure 8: Plots of the median translation error measure (34) as function of the standard deviation σ of additive Gaussian noise for three random-point fractions ν, two object shape classes, and four types of pose clustering. Length dimensions are given in units of the longest edge of the test objects' bounding box. Vertical bars indicate the standard deviation of the median error.

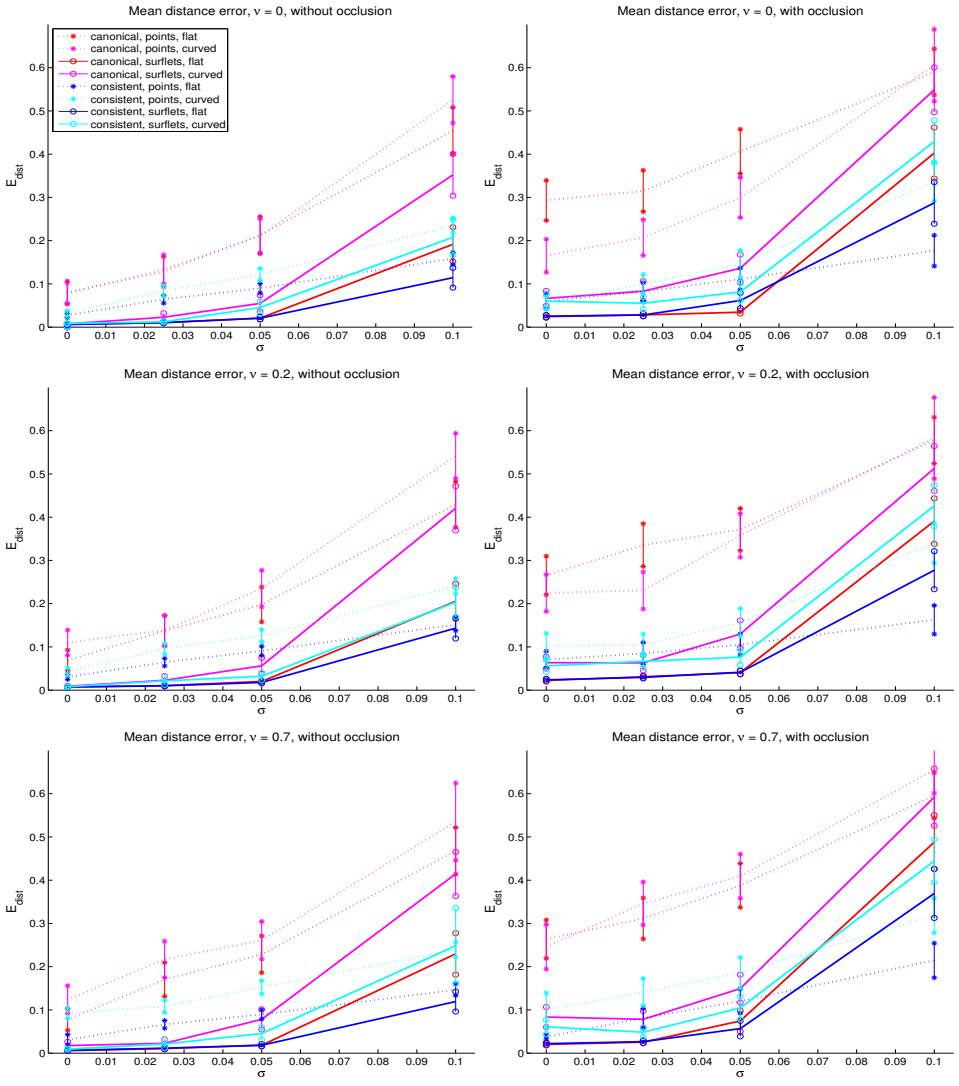

Figure 9: Plots of the mean distance error measure (35) as function of the standard deviation σ of additive Gaussian noise for three random-point fractions ν, two object shape classes, and four types of pose clustering. Length dimensions are given in units of the longest edge of the test objects' bounding box. Vertical bars indicate the standard deviation of the mean error.

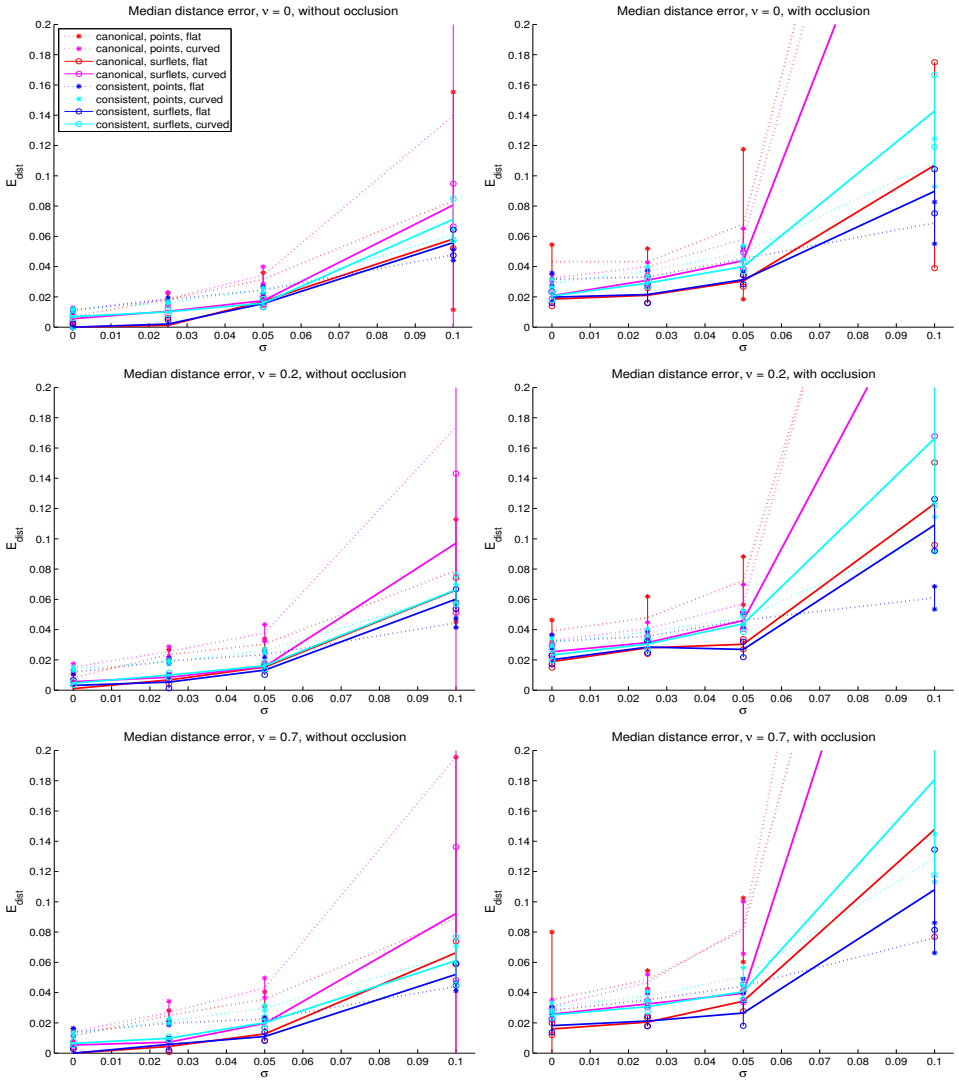

Figure 10: Plots of the median distance error measure (35) as function of the standard deviation σ of additive Gaussian noise for three random-point fractions ν, two object shape classes, and four types of pose clustering. Length dimensions are given in units of the longest edge of the test objects' bounding box. Vertical bars indicate the standard deviation of the median error.

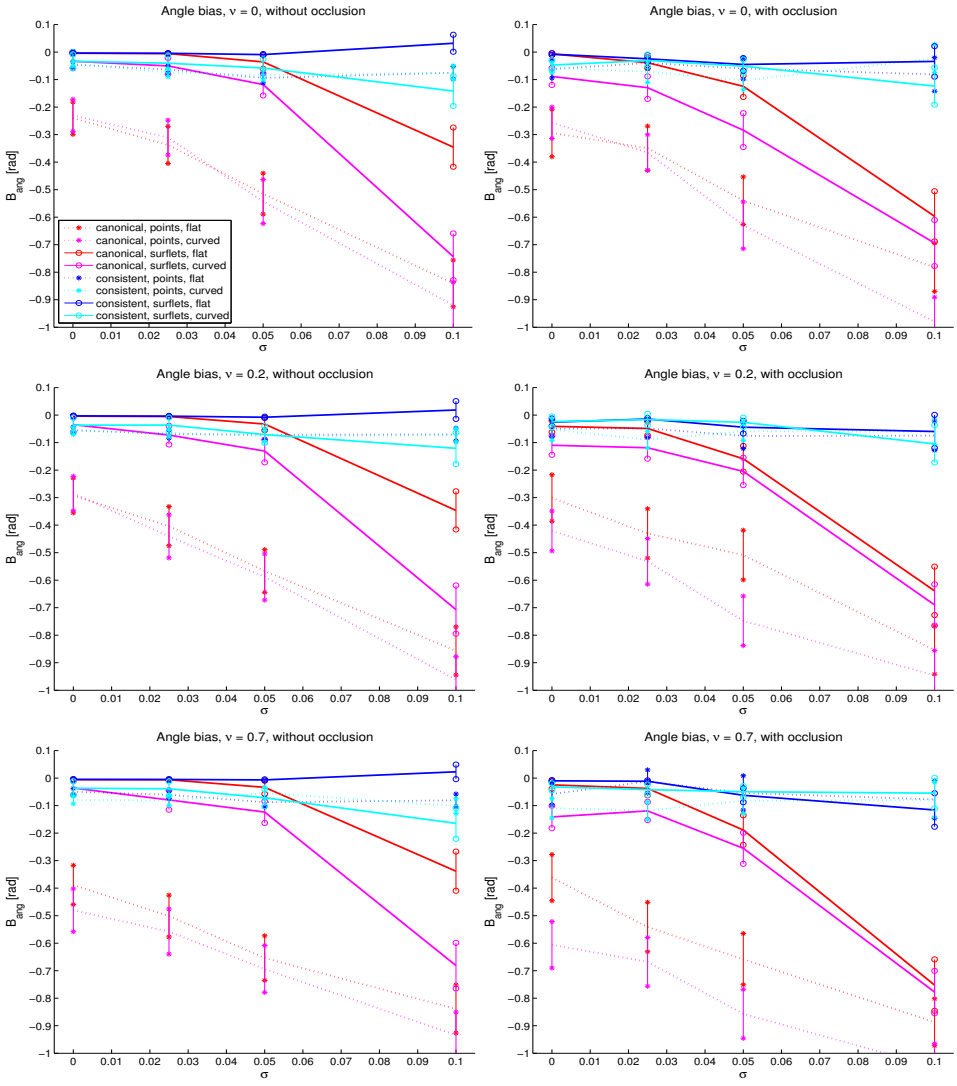

Figure 11: Plots of the angle bias (37) as function of the standard deviation σ of additive Gaussian noise for three random-point fractions ν, two object shape classes, and four types of pose clustering. For these examples, algorithmic parameters were set to intermediate values: $d_{\text{dist}} = 0.05, d_{\text{ang}} = \pi/3, d_{\text{diff}} = 0.08, \delta_{\text{rot}} = 0.08(\pi), \delta_{\text{trans}} = 0.3$; cf. table 1. Length dimensions are given in units of the longest edge of the test objects' bounding box. Vertical bars indicate the standard deviation of the angle bias.

outstandingly high for this constellation, indicating a high rate of failure of estimation.

The generally higher difficulty of estimating pose for curved objects is probably due to their greater similarity under slight changes of orientation. However, if an estimator gets unstable and often fails completely, a flat-faced object with a lot of orthogonal faces may divert it to false matches that are an angle of $\pi/2$ or even π away from the true orientation. This could explain the particular problem of canonical clustering from sampled point triples when processing occluded data of flat-faced objects.

Sampling point triples versus sampling surflet pairs

Sampling surflet pairs is more effective in producing good hypotheses than sampling point triples, as long as the data are of sufficient quality. Thus, for small to moderate additive noise, mean and median estimation errors are lower for clustering of hypotheses generated from surflet pairs. At high levels of noise, however, hypothesis generation from point triples can outperform hypothesis generation from surflet pairs, when using the *consistent* parameterization; the same is *not* true for sampling point triples with canonical parameterization. Regarding rotation and distance errors, an advantage of sampling point triples with consistent parameterization over sampling surflet pairs is most evident for very noisy data with occlusion. Regarding translation errors, sampling point triples with consistent parameterization outperforms sampling surflet pairs at very high noise more clearly without occlusion.

The relative sensitivity of surflets to noise can be explained by the fact that local surface normal directions become undefined when the surface representation is heavily corrupted by uncorrelated additive noise. Estimated surface normals will then not contain a lot of information on object shape, and hence orientation. On the other hand, relations between more distant surface points, as exploited when sampling point triples, are more robust to additive noise. It is an interesting result, however, that this potential advantage of point triples is only effective with the consistent parameterization.

Clustering in canonical versus consistent pose space

Pose clustering with consistent parameterization produces the same or lower mean and median estimation errors than with canonical parameterization. The advantage of the consistent parameterization is more significant

for more heavily corrupted data, that is, with higher noise and with occlusion. It is also more pronounced when sampling point triples than when sampling surflet pairs.

Noise and occlusion have their most destructive effect on canonical clustering from point triples: its mean errors at low noise are significantly increased by occlusion, indicating a higher probability of failure; while the mean and median errors for rotation and distance rise dramatically when going to the highest noise level, indicating a systematic brake down of the algorithm. In sharp contrast to this, there is no strong effect of noise on consistent clustering from point triples, and a destructive effect of occlusion is only apparent in the translation errors. Therefore, this type of pose clustering turns out to be the most robust to all tested kinds of data corruption. At very high noise with occlusion, it has the lowest mean and median estimation errors of the four clustering types.

Also for sampling surflet pairs, the advantage of clustering in the consistent parameter space becomes obvious with increasing noise. Consistent clustering performs better than canonical clustering on both shape classes; however, the difference is most dramatic for the objects with curved surface.

The generally lower accuracy and robustness of clustering in canonical pose space, apparent in its higher and less stable mean and median errors, are likely due to an estimation bias towards too small rotation angles, as discussed in sections 2.2 and 3.2.2, and as seen in fig. 11. The figure also shows that there is no angle bias of clustering in consistent pose space.

These results confirm and extend what has been found in [1] for pure rotation estimation, i.e., without translation, from spatially uniform random data. There the estimation procedure was through sampling point triples followed by various methods for locating a cluster center in parameter space, including the mean-shift procedure used in this study; cf. section 3.2.

Computation times

To enable comparison with other stochastic pose estimators, which like pose clustering depend heavily on the allowed run time (see section 5.2), we here include the mean computation times taken by the four clustering types at all levels of data corruption. The tables 4 and 5 show the mean run times for the algorithmic variants that yielded the mean and median estimation errors, respectively, plotted in figs. 5 through 10. The run times include the sampling procedure from the data (with building the used hash tables), computation of parameter hypotheses, and finding the maximum parameter

Table 4: Mean computation times in seconds for algorithmic variants minimizing the mean of the estimation errors.

			no occlusion				**with occlusion**			
		ν	σ				σ			
			0	0.025	0.05	0.1	0	0.025	0.05	0.1
consistent	**point triples**	0	4.31	2.31	4.46	16.60	2.60	2.76	4.63	11.96
		0.2	5.82	3.41	5.61	16.66	2.92	3.30	5.58	14.71
		0.7	5.92	6.40	10.13	22.24	5.70	6.59	8.94	18.52
	surflet pairs	0	1.04	1.18	1.46	5.64	1.14	1.23	1.23	6.82
		0.2	1.21	1.30	1.60	5.97	1.10	1.03	1.12	7.01
		0.7	2.00	2.21	2.46	6.23	1.47	2.05	1.72	7.19
canonical	**point triples**	0	1.71	2.25	4.23	12.75	1.90	2.13	3.15	13.47
		0.2	2.53	3.15	4.86	12.79	2.37	2.41	4.90	12.26
		0.7	4.96	5.10	9.32	18.09	5.53	6.41	6.96	17.02
	surflet pairs	0	1.02	0.97	1.07	3.90	0.95	0.75	0.89	3.07
		0.2	1.18	1.16	1.13	2.93	0.86	0.94	1.06	6.69
		0.7	1.96	1.87	1.90	5.93	1.31	1.63	1.59	6.00

density. It does not include generation of test data sets or estimation of the surface normals for the surflets. The algorithms were implemented in C++ without making use of special features of a hardware architecture. Computation was carried out single-threaded on a Xeon 5160 CPU running at 3.0 GHz.

Note that a finer quantization of the pose parameter space (described by parameters δ_{rot}, δ_{trans}) can achieve lower pose errors with high-quality data, but usually at the cost of more time spent to fill a bin in parameter space.[10] This effect is reflected in tables 4 and 5 by the occasionally longer run times at lower levels of data corruption.

4.4.2 Real scene data

Since for the real data, we have just one case of pose estimating each object in each scene, and the level of occlusion is very different across objects and scenes, we cannot present an error statistics like for the synthetic data. Instead, fig. 12 shows the individual errors (30), (31), (32), and the angle differences (36), plotted against the occlusion ratio of the respective object in the respective scene. To facilitate comparison between clustering types, the

[10] Put differently, more time spent to locate the maximum parameter density at a finer scale.

Table 5: Mean computation times in seconds for algorithmic variants minimizing the median of the estimation errors.

			no occlusion				**with occlusion**			
		ν	σ				σ			
			0	0.025	0.05	0.1	0	0.025	0.05	0.1
consistent	**point triples**	0	4.62	2.44	4.25	17.16	3.35	2.13	3.93	11.72
		0.2	5.82	3.04	5.00	14.93	3.60	3.07	4.62	15.39
		0.7	7.93	5.65	8.33	20.96	7.45	6.49	8.97	20.77
	surflet pairs	0	1.01	1.23	1.18	3.43	0.91	0.88	1.26	2.79
		0.2	1.17	1.41	1.29	3.29	1.09	1.17	1.08	5.09
		0.7	1.94	2.12	1.98	5.19	1.73	2.11	1.66	7.16
canonical	**point triples**	0	4.53	3.23	4.89	16.24	3.89	3.04	3.52	13.64
		0.2	6.24	3.15	4.86	15.82	5.23	2.94	4.97	13.06
		0.7	9.51	7.03	8.28	22.74	8.88	5.97	8.34	15.59
	surflet pairs	0	0.99	1.18	1.44	3.28	0.94	0.83	0.92	5.52
		0.2	1.19	1.46	1.30	3.68	1.03	1.05	1.06	6.42
		0.7	1.96	2.18	2.85	4.74	1.77	1.86	1.59	6.25

errors for the same object in the same scene are joined by a line with coloring according to the type producing the lower error. For the angle differences, the joining line is colored according to the larger estimated rotation angle. We note that the purpose of the angle difference plot is to give a sense of the relative angle bias, analogous to fig. 11; angle difference is not an error measure.

Only results for sampling surflet pairs are shown; point triple sampling, both with clustering in canonical and in consistent pose space, usually failed to return an acceptable pose estimate. The reason was, quite evidently, that the large amount of clutter in the scenes made it too unlikely to randomly draw corresponding point triples, thus spoiling the statistics for locating good clusters. Drawing surflet pairs, on the other hand, has a great combinatorial advantage over drawing point triples.

All types of clustering could be enhanced by drawing more samples, i.e., beyond the stopping criterion used throughout this paper (cf. section 4.1), or by constraining the sampling to surface patches obtained from prior segmentation of the scene. However, both these measures fall outside the scope of the present study. The comparison between algorithms here is strictly based on the same procedures for all the test data.

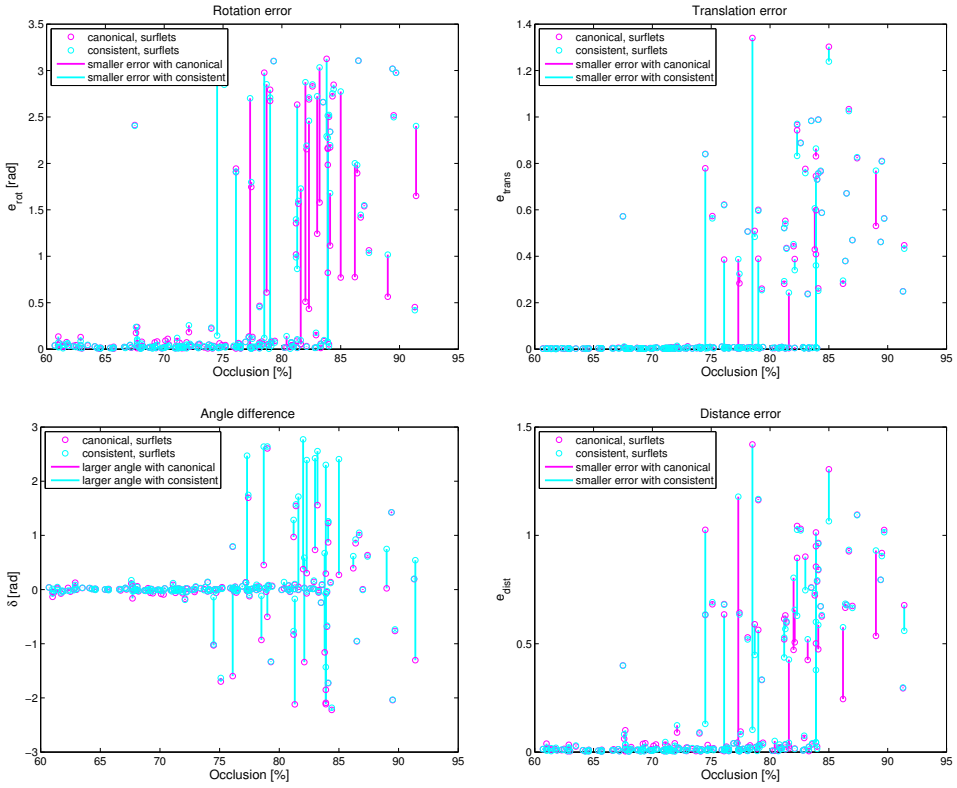

Figure 12: Plots of the estimation errors (30), (31), (32), and the angle differences (36) against the object occlusion ratio for each case. Length dimensions are given in units of the longest edge of the test objects' bounding box. Vertical lines connect estimates for the same object and scene by the two clustering types with surflet sampling.

Estimation errors

The difference between the two clustering types based on sampling surflet pairs is best discussed in terms of their gross failures on individual cases of estimating object pose in a scene. These failures show up in all three error measures simultaneously. It is seen that up to roughly 75% occlusion there is just one failure of pose estimation, where both canonical and consistent parameter clustering fail on the same object and scene. Around 75% occlusion, clustering in canonical parameter space has two more cases of failure, while clustering in consistent parameter space fails around 77% occlusion for the second time. Both types fail more often at higher occlusion. However, there are altogether five cases of failure in canonical parameter space that do not occur in consistent parameter space, while there are only two cases with the reverse situation.

Other differences between the errors of canonical and consistent parameter clustering are either much smaller, such that their statistical significance cannot be established with this data set; or they lie within an error range that disqualifies both estimates as failures.

The results on estimation errors on real scenes are thus consistent with those on synthetic single objects, albeit not as expressive. A qualitative difference between the results is in the way the estimators fail: for single objects, failure is often more gradual with increasing data corruption; while in a scene the outcome of a failed estimation is determined by the other objects that 'capture' the model, producing a larger gap between successful and failed cases.

Angle differences

The size of rotation from object model to scene frame, as measured by the rotation angle, is estimated systematically lower through clustering in canonical parameter space than through clustering in consistent parameter space. This parallels the finding on the synthetic data of single objects; cf. fig. 11. On the other hand, unlike for the synthetic data, there is no indication of a negative angle bias for the canonical parameter space. Rather, a small positive bias is apparent for the consistent parameter space. This bias, however, arises only among the cases of failed estimation with a large orientation error.

As argued above, it is likely that the outcomes of failed estimation are largely determined by the other objects in a scene. We believe that this

explains the difference in angle bias found between the single object and scene data.

5 Discussion

5.1 Outliers and the breakdown point

Throughout this article, we have used the term 'robustness' in the somewhat informal sense of a degree of resistance to various kinds of data corruption – additive noise, addition of random points, and occlusion – that all lead to deviation from the rigid motion model to be estimated. This terminology is common in the computer vision literature and has even been explicitly advocated [45]. However, often in the statistics literature and sometimes in the computer vision literature, the notion of robustness is connected more narrowly with the tolerance to outliers to the fitted parametric model. Therefore, we shall now briefly discuss the issue of outliers in the present context of pose estimation.

We have estimated the parameters of the rigid motion model from object-model points X and scene points Y without information on their correspondence. The estimate is hence based upon the set of all model/scene pairs $X \times Y$. The outliers to the motion model are constituted by all false correspondences, i.e., all pairs in $X \times Y$ that do not actually correspond through rigid motion of X onto Y. Because of occlusion and additional scene points, not all points in X have a corresponding counterpart in Y, and vice versa. In a sense, the proportion of outliers in relation to all point pairs is hence greater than

$$\frac{|X|\,|Y| - \min(|X|, |Y|)}{|X|\,|Y|} = 1 - \frac{1}{\max(|X|, |Y|)} \;, \tag{38}$$

by assuming that each point in X corresponds at most to its nearest neighbor in Y after correct alignment, and vice versa. For dense range data, the resulting outlier ratio is then amazingly close to the value one: in our case, $\max(|X|, |Y|)$ was between a few 100 and a few 10,000 points. However, a correspondence between dense range data points is never exactly correct and often not completely wrong, as points are distributed quasi-continuously over the object surfaces. Indeed, a point in X may be corresponded to a local subset of points from Y such that acceptable pose hypotheses are produced. The acceptable correspondence region of a model point in the scene data space depends on the constellation of *all* points used to compute a pose hypothesis and, of course, on the application context. The acceptable

correspondence subset of Y depends also on the local distribution of points. The effective outlier ratio may hence be smaller than given in (38) by an amount that is very hard to quantify.

Having no definite outlier ratios available, we can also not quantify a breakdown point as a critical proportion of outliers. It should be noted, however, that the actual breakdown of an estimator is not only dependent on the proportion of outliers, but also on the specific distribution of both outliers and inliers [32, 45]. Any empirical value of a breakdown point is hence questionable in principle. On the other hand, as usual for robust estimators, a theoretical derivation of a breakdown point for parameter clustering is unfortunately not known.[11]

5.2 Open issues

Extension of our study to other problem domains and other classes of estimators would be desirable. Regarding other problem domains, one could be interested in the behavior of pose clustering algorithms for estimating motion from sparse data points, each one attributed with additional features. This kind of estimation problem commonly arises when computing motion of a camera from image sequences. In this situation, the amount of outliers due to false correspondences is a lot less and other effects may be dominant. Likewise, when estimating analytic shape models (planes, cylinders, quadratic surfaces, etc.), there is no problem of detailed correspondences between data points and model points to be solved. Only the inlier data segment as a whole needs to be identified.

Regarding other classes of estimators, it would be desirable to compare, in a similarly extensive study, parameter clustering as a parameter space method with the various data space methods mentioned in section 1.2. Because of their different algorithmic strategies, however, such a comparison demands special care. Both types of method can improve on their result simply by drawing and processing more samples from the data. As data space methods evaluate each parameter hypothesis on the data set in a rather expensive procedure, they must rely on relatively few hypotheses; but having processed just one acceptable hypothesis in the sequence may be sufficient. Parameter clustering, on the other hand, never evaluates a hypothesis on the data set and, hence, can process some orders of magnitude more parameter samples within the same amount of time; but it needs to sample many acceptable hypotheses for detecting a cluster within the background of false

[11] Even the respective derivations for estimates based on kth-order residual statistics do not apply generally to realistic situations in computer vision [46, 32].

parameter samples. Data space and parameter space methods hence spend their run time with different kinds of computation. A fair comparison of performance thus needs to consider the trade-off between performance and run time, which in turn requires a careful optimization of all implementations.

In general, the best framework for realizing a fair comparison of different methods would be through the quantitative evaluation by the respective authors on a range of relevant public data sets or within a regular open challenge. Regarding evaluation standards, the field of robust estimators lags behind some other fields, e.g., object category recognition, that have established such procedures. Having used public databases in our study should enable other authors to compare their robust, global pose estimators to pose clustering.

Finally, a lot of recent efforts in improving data space methods of robust estimation has been directed towards data-driven scale selection, that is, automatically adapting the error bounds of the inlier data around a correct solution [28, 29, 30, 23, 33, 34]. For parameter clustering, since we effectively rely on a kernel density estimate of the parameters (see section 3.2), instead of the scale one has to adapt the kernel bandwidth. In this study, however, we have avoided a data-driven selection of kernel bandwidth. Instead, we have systematically varied the bandwidth across different runs and selected the best value for each level of data corruption and for the real scene data. The results of this study hence show the estimator performance achievable with a proper adjustment of the bandwidth (along with the other algorithmic parameters), either a priori or through online adaptation. When comparing to data space methods with data-driven scale selection, an analogous data-driven procedure for kernel bandwidth selection has to be implemented as well, such as those suggested in [43]. We note, however, that in all our practical applications of pose clustering, the a priori adjustment of all algorithmic parameters to a specific source of data was never a problem.

5.3 Summary

We have presented an extensive quantitative study of four variants of the pose clustering algorithm: sampling point triples or surflet pairs, each combined with clustering in canonical or consistent pose space. The focus has been on the relative pose estimation accuracy and robustness achievable by these different sampling strategies and parameter spaces. Synthetic test data were generated from a public database of 3D object models through systematic degradation of the geometric object representation. Real scene data were taken from another public database.

The main conclusions from this study are as follows.

- Explicit usage of local surface normal information, extracted from measured surface points, is more efficient than relying directly on the same surface points for computing pose hypotheses, as long as the surface representation is not heavily corrupted by measurement noise. It is to be expected that the same is true for the extraction and usage of surface curvature information, where the requirements on data quality would be even stronger. The advantage of sampling surflet pairs over sampling point triples is most evident when analyzing (non-segmented) cluttered scenes. Conversely, if the surface representation is severely degraded by noise, direct usage of data points is preferable, but only with consistent parameterization.

- Pose clustering in the consistent parameter space is always preferable to clustering in the canonical parameter space. The latter may produce strong angle bias; estimation errors are generally larger and lead to failure more often. It is not to be expected that the other classic parameterizations of rotation would perform better than the canonical, because of similar inherent problems with bias, in addition to lack of compactness or presence of singularities.

- The pose estimator most robust to data corruption among the four variants investigated is through sampling point triples and clustering in the consistent parameter space.

We are here making no claims as to the relative performance of pose clustering and any other type of pose estimator. It is, in fact, quite likely that the most efficient global pose estimator for any particular kind of data will be some specific blend of different estimation strategies, from clustering, RANSAC, and local estimators (ICP, M-estimators, etc.). For component as well as stand-alone estimators, the results of this study give advice on how to utilize parameter density estimation and clustering techniques for global pose estimation from dense range data without correspondences.

Acknowledgments

This work has been partially funded by the German Federal Ministry of Education and Research as part of the integrated project Lynkeus (www.lynkeus-3d.de).

References

[1] U. Hillenbrand, "Consistent parameter clustering: definition and analysis," *Patt. Recogn. Let.*, vol. 28, pp. 1112–1122, 2007.

[2] T. Bodenmüller, W. Sepp, M. Suppa, and G. Hirzinger, "Tackling multisensory 3D data acquisition and fusion," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007.

[3] M. Suppa, S. Kielhöfer, J. Langwald, F. Hacker, K. H. Strobl, and G. Hirzinger, "The 3D-Modeller: a multi-purpose vision platform," in *Proc. IEEE Int. Conf. on Robotics & Automation*, 2007.

[4] Z. Zalevsky, A. Shpunt, A. Maizels, and J. Garcia, "Method and system for object reconstruction," 2007. WIPO Patent Application WO/2007/043036.

[5] S. Fuchs and G. Hirzinger, "Extrinsic and depth calibration of ToF-cameras," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2008.

[6] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 21, pp. 433–449, 1999.

[7] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 24, pp. 509–522, 2002.

[8] A. S. Mian, M. Bennamoun, and R. A. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 28, pp. 1584–1601, 2006.

[9] A. S. Mian, M. Bennamoun, and R. A. Owens, "On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes," *Int. J. Computer Vision*, vol. 89, pp. 348–361, 2010.

[10] J. Li and N. M. Allinson, "A comprehensive review of current local features for computer vision," *Neurocomputing*, vol. 71, pp. 1771–1787, 2008.

[11] P. V. C. Hough, "Method and means for recognizing complex patterns," 1962. U.S. Patent 3069654.

[12] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Patt. Recogn.*, vol. 13, pp. 111–122, 1981.

[13] G. Stockmann, S. Kopstein, and S. Benett, "Matching images to models for registration and object detection via clustering," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 4, pp. 229–241, 1982.

[14] G. Stockmann, "Object recognition and localization via pose clustering," *CVGIP*, vol. 40, pp. 361–387, 1987.

[15] J. Illingworth and J. Kittler, "A survey of the Hough transform," *CVGIP*, vol. 44, pp. 87–116, 1988.

[16] S. Moss, R. C. Wilson, and E. R. Hancock, "A mixture model for pose clustering," *Patt. Recogn. Let.*, vol. 20, pp. 1093–1101, 1999.

[17] H. Chen and P. Meer, "Robust compter vision through kernel density estimation," in *Proc. Europ. Conf. Computer Vision*, vol. 2350 of *Lecture Notes in Computer Science*, (Berlin, Heidelberg), pp. 236–250, Springer, 2002.

[18] O. Tuzel, R. Subbarao, and P. Meer, "Simultaneous multiple 3D motion estimation via mode finding on Lie groups," in *Proc. Int. Conf. Computer Vision*, pp. 18–25, 2005.

[19] U. Hillenbrand, "Pose clustering from stereo data," in *Proc. VISAPP Int. Workshop on Robotic Perception*, pp. 23–32, 2008.

[20] R. Subbarao and P. Meer, "Nonlinear mean shift over riemannian manifolds," *Int. J. Computer Vision*, vol. 84, pp. 1–20, 2009.

[21] G. Stillfried, U. Hillenbrand, M. Settles, and P. van der Smagt, "MRI based skeletal hand movement model," in *The Human Hand - A Source of Inspiration for Robotic Hands*, Springer Tracts on Advanced Robotics. In press.

[22] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[23] C. V. Stewart, "Robust parameter estimation in computer vision," *SIAM Review*, vol. 41, pp. 513–537, 1999.

[24] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography," *Comm. ACM*, vol. 24, pp. 381–395, 1981.

[25] "25 years of RANSAC: workshop in conjunction with CVPR 2006." http://cmp.felk.cvut.cz/ransac-cvpr2006/.

[26] P. J. Rousseeuw, "Least median of squares regression," *J. Amer. Stat. Assoc.*, vol. 79, pp. 871–880, 1984.

[27] X. Yu, T. D. Bui, and A. Krzyzak, "Robust estimation for range image segmentation and reconstruction," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, pp. 530–538, 1994.

[28] C. V. Stewart, "MINPRAN: a new robust estimator for computer vision," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, pp. 925–938, 1995.

[29] J. V. Miller and C. V. Stewart, "MUSE: robust surface fitting using unbiased scale estimates," in *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 300–306, 1996.

[30] K.-M. Lee, P. Meer, and R.-H. Park, "Robust adaptive segmentation of range images," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 20, pp. 200–205, 1998.

[31] U. Hillenbrand and G. Hirzinger, "Probabilistic search for object segmentation and recognition," in *Proc. Europ. Conf. Computer Vision*, vol. 2352 of *Lecture Notes in Computer Science*, (Berlin, Heidelberg), pp. 791–806, Springer, 2002.

[32] H. Wang and D. Suter, "MDPE: a very robust estimator for model fitting and range image segmentation," *Int. J. Computer Vision*, vol. 59, pp. 139–166, 2004.

[33] H. Wang and D. Suter, "Robust adaptive-scale parametric model estimation for computer vision," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 26, pp. 1459–1474, 2004.

[34] C.-M. Cheng and S.-H. Lai, "A consensus sampling technique for fast and robust model fitting," *Patt. Recogn.*, vol. 42, pp. 1318–1329, 2009.

[35] A. Nölle, "Untersuchung der Genauigkeit von Objektlageschätzern," Diploma thesis, TU Dresden, 2008.

[36] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *Proc. Shape Modeling International*, 2004.

[37] Princeton Shape Benchmark. http://shape.cs.princeton.edu/benchmark/.

[38] Website of A. S. Mian. http://www.csse.uwa.edu.au/~ajmal/recognition.html.

[39] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, pp. 629–642, 1987.

[40] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-D rigid body transformations: a comparison of four major algorithms," *Mach. Vision App.*, vol. 9, pp. 272–290, 1997.

[41] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification," in *Proc. Int. Conf. 3-D Digital Imaging and Modeling*, pp. 474–481, IEEE Computer Society Press, 2003.

[42] K. Fukunaga and L. D. Hostetler, "The estimtion of a gradient of a density function, with applications in pattern recognition," *IEEE Trans. Info. Theory*, vol. 21, pp. 32–40, 1975.

[43] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 24, pp. 603–619, 2002.

[44] M. Shuster, "A survey of attitude representations," *J. Astronaut. Sci.*, vol. 41, pp. 439–517, 1993.

[45] P. Meer, "Robust techniques for computer vision," in *Emerging Topics in Computer Vision*, ch. 4, Prentice Hall, 2004.

[46] H. Wang and D. Suter, "Using symmetry in robust model fitting," *Patt. Recogn. Let.*, vol. 24, pp. 2953–2966, 2003.