Scale-Invariant Vote-Based 3D Recognition and Registration from Point Clouds

Minh-Tri Pham, Oliver J. Woodford, Frank Perbet, Atsuto Maki,
Riccardo Gherardi, Björn Stenger, and Roberto Cipolla

Abstract. This chapter presents a method for vote-based 3D shape recognition and registration, in particular using mean shift on 3D pose votes in the space of direct similarity transformations for the first time. We introduce a new distance between poses in this space—the SRT distance. It is left-invariant, unlike Euclidean distance, and has a unique, closed-form mean, in contrast to Riemannian distance, so is fast to compute. We demonstrate improved performance over the state of the art in both recognition and registration on a (real and) challenging dataset, by comparing our distance with others in a mean shift framework, as well as with the commonly used Hough voting approach.

1 Introduction

This chapter concerns itself with vote-based pose estimation techniques. These arise in many vision tasks including 2D object detection [21, 28, 37], motion segmentation [39, 40], and 3D shape registration and recognition [11, 20, 41]. These methods all share a common two stage framework: First they generate an empirical distribution of pose through the collation of a set of possible poses, or *votes*. The votes are often computed by matching local features from a test object to those in a library with known pose [11, 20, 21, 28, 37, 39, 40, 41], or by learning a function that maps features to votes [16, 29]. The second step is then to find one or more "best" poses in the distribution (the maxima, in the case of ML/MAP estimation). This curation of

Minh-Tri Pham · Oliver J. Woodford · Frank Perbet · Atsuto Maki · Riccardo Gherardi · Björn Stenger
Toshiba Research Europe Ltd
e-mail: firstname.lastname@crl.toshiba.co.uk

Roberto Cipolla
Department of Engineering, University of Cambridge, UK
e-mail: cipolla@eng.cam.ac.uk

R. Cipolla et al. (Eds.): Machine Learning for Computer Vision, SCI 411, pp. 137–162.
springerlink.com

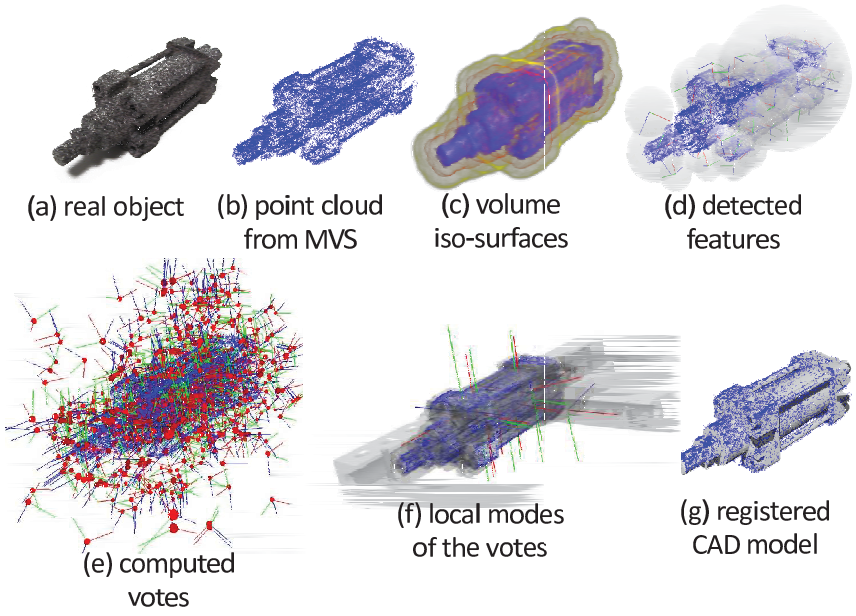


Fig. 1 Our System. for 3D-shape-based object recognition and registration. (a) Real object, fabricated from a CAD model. (b) Point cloud extracted using a multi-view stereo (MVS) system. (c) Iso-surfaces of the scalar volume computed from the points. (d) Features (with full scale, rotation and translation pose) detected in the volume. (e) Votes for the object centre, based on detected features matched with a library of learnt features. (f) Local modes of the votes. (g) The registered CAD model.

data prior to inference makes such vote-based approaches more efficient and robust than competing techniques, *e.g.*, global or appearance-based methods [26].

Two compelling methods in finding best poses are Hough voting and mean shift. In Hough voting the standard approach is to compute the probabilities on a regular grid over the pose parameter space. This discretization leads to loss of accuracy, as well as a complexity exponential in the pose dimensionality, but ensures coverage of the entire space. Mean shift [9] iteratively finds local maxima of probability, resulting in initialization issues but also high accuracy. The complexity of an iteration is usually linear in the pose dimensionality. The two methods are therefore somewhat complementary; indeed they are often used together [21, 28].

While Hough voting can be easily applied to any space (in our case that of all poses), this is not straightforward for mean shift; each iteration requires the computation of a weighted average of input votes, formulated as a least squares minimization of distances from input votes to the mean. In Euclidean space this minimization yields a unique, closed-form solution—the arithmetic mean. When poses lie on a non-linear manifold this mean is typically outside the manifold,

requiring a projection onto it. A more direct approach is to minimize the geodesic arclengths over the manifold, known as the Riemannian distance.

In this chapter we focus on 3D shape recognition and registration, as part of a system (see Fig. 1) for recognizing industrial parts. However, unlike existing approaches, where objects of interest are of either fixed (or omitted) scale [40] or rotation [21, 28, 37], here we recognize and register objects in the direct similarity group: the group of isotropic similarity transformations parameterized by translation, rotation *and* scale [36]. Scale is necessary when the input data's scale is unknown, or when there is high intra-class scale variation. Rotation is necessary for full registration, leading to more accurate recognition. The resulting 7D pose space is currently too large to apply Hough voting to in practice [19]. Here we use mean shift, for which scale and rotation also introduce problems using existing distances: Euclidean distance is scale variant, and the induced mean of poses has a bias in scale. The mean of poses using Riemannian distance has no closed-form solution, even when the poses are pure rotations [25], and is slow to compute [38].

The contribution of this work is to introduce a new distance on the direct similarity group. The distance provides scale, rotation and translation-invariance concomitantly. The weighted mean of this distance is unique, closed-form, and fast to compute, as well as having several key properties discussed in Sect. 2.4.3. We demonstrate the distance's performance in mean shift, in the context of our 3D shape registration and recognition system, comparing it with other distances on the same space, as well as a Hough voting method.

The chapter is laid out as follows: The next section reviews the literature relevant to 3D shape recognition and registration inference, as well as how our method is positioned compared to existing approaches. In the following section, we introduce our new distance on the direct similarity group, and its associated mean. In the final two sections, we present our experiments, before concluding.

2 Background

We start with discussing two main trends in the literature: global approaches versus local approaches, in which our method belongs to the latter. We then review how a local appearance model is learned and used for generating votes from features in local approaches. Our method extracts features using the standard Difference-of-Gaussian (DoG) operator and matches features between the scene and the model to generate votes. In the last part of the section, we review the inference techniques used for vote-based pose estimation, and take a closer look at mean shift applied to this task.

2.1 Global Approaches vs. Local Approaches

Recognizing and registering rigid objects from 3D point clouds is a well-known problem in computer vision [6, 22, 23]. Often, the 3D point clouds obtained by

different sensors such as laser scans, time-of-flight cameras, or stereo systems [43] contain small, irrelevant, neighbouring clutter in addition to the relevant data coming from the objects.In most cases, the relevant data themselves do not capture full shapes. Two main approaches to solve the problem are: global approaches and local approaches. Global approaches recognize objects by relying on global features, *i.e.*, features extracted from the complete 3D geometry of the point cloud. Examples include spherical harmonics [18, 35], shape moments [35], and shape histograms [29]. It is difficult to handle partial shapes using these approaches, since global features are sensitive to both absence of shape parts and occurrence of clutter.

Recent works in 2D object detection and object class categorization [21, 28] have shown the advantage of using local, rather than global, features in dealing with occlusions and clutter. In 3D, similar success stories have been reported with methods using local features [8, 15, 17, 20, 24, 41]. These approaches can integrate information from a large number of object parts. They demonstrate good generalization as they are free to combine parts observed on different training examples. Spin Images by Johnson and Hebert [17] is arguably the most popular early work, in which local 3D descriptions are represented as 2D histograms of points falling within a cylindrical region by means of a plane that "spins" around the normal, and recognition is done by matching spin images, grouping similar matches and verifying each output pose. Many local features and descriptors have been proposed thereafter, with new ones being more discriminative, more repeatable, more robust to noise and more invariant to local rigid transformations. Chen and Bhanu [8] compute histograms of normals and shape indices for describing local regions. The 3D Shape Context of Frome *etal*. [15] extends Spin Images' basic idea to computing 3D histograms of points within a sphere around a feature point. Mian *etal*. [24] accumulate 3D histograms of mesh triangles within a cubic support. Rusu *etal*. [34] propose Point Feature Histograms describing the local geometry of a point and its k nearest neighbours. Knopp *etal*. [20] extend the SURF descriptor from 2D to 3D and show how 3D shape recognition can be improved by a Hough-transform based approach. Petrelli and Di Stefano [33] improve the repeatability of local reference frames via point normals. Surveys of local features and descriptors 3D methods are available in [6, 22, 23].

Many of these approaches share a common vote-based framework. They first learn a local appearance model for the object classes to be recognized and registered, which maps, either directly or indirectly, features to ground truth object identities and poses. During inference, features from the scene point cloud are extracted. Via the local appearance model, each of these features generates one or more votes, representing hypotheses that an object of a given pose exists. The votes can be viewed as points of a kernel density estimator that estimates the joint probability density function of both object identity and pose. Local modes of the kernel density function are found via a suitable mode-seeking approach, and returned as final object identities and poses. Methods such as Iterative Closest Point [5] and variants, are able to further refine the output poses, if necessary.

Following these approaches, we use the standard vote-based framework in our 3D recognition and registration. However, our approach differs from existing

approaches in that we infer simultaneously scale, rotation and translation in 3D. Due to the introduction of scale, the pose space becomes too large (7D) for existing Hough transform-based approaches to work with, while existing mode-seeking methods like mean shift have bias in scale, as to be seen in the following sections.

2.2 Learning a Local Appearance Model

2.2.1 Feature Extraction

Salient interest regions are extracted over location and scale from a variety of point cloud instances of objects of the same class (in our case 20 point cloud instances per class) using 3D interest point detectors like 3D SURF [20]. For each interest region, *i.e.*, a sphere centered at $\mathbf{c}$ with radius r, a 3D canonical orientation based on the geometry of the points inside the region is computed, for example by finding the principal directions of the points using PCA [24], or by fitting a local surface and then finding most repeatable directions on the surface from the center of the sphere [33]. A local reference frame is created as a result, hereinafter *feature frame*, originating at $\mathbf{c}$, with one unit length equal to r, and with 3D orientation coinciding with the 3D canonical orientation. The feature frame is specified uniquely by a 3D direct similarity transformation $\mathbf{F} \in S^+(3)$ [36],

$$\mathbf{F} = \begin{bmatrix} s(\mathbf{F})\mathbf{R}(\mathbf{F}) & \mathbf{t}(\mathbf{F}) \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}, \tag{1}$$

which converts the coordinates of a 3D point from the global coordinate system to the feature frame. Here, $s(\mathbf{F}) \in \mathbb{R}^+$, $\mathbf{R}(\mathbf{F}) \in SO(3,\mathbb{R})$, and $\mathbf{t}(\mathbf{F}) \in \mathbb{R}^3$ specify the scale, rotation, and translation components of $\mathbf{F}$, respectively. A low-dimensional feature descriptor $\mathbf{d} \in \mathbb{R}^k$ (for some positive integer number k) is extracted based on the distribution of the coordinates of the points inside the region with respect to $\mathbf{F}$.

Each point cloud instance is associated with a local reference frame specifying the ground truth pose of the object captured in the point cloud, hereinafter *object frame*. Unlike most existing 3D approaches where an object pose specifies translation only [41], translation and scale [20], or translation and rotation [13], in our system an object pose specifies scale and rotation and translation altogether, hence dealing with a larger pose space than existing works. Here we treat scale as part of an object pose and choose an object frame originating at the center location of the object, with 3D orientation the same as the object's orientation, and with one unit length equal to the object scale. Analogously to the feature frames, an object frame is specified uniquely by 3D direct similarity transformation $\mathbf{X} \in S^+(3)$,

$$\mathbf{X} = \begin{bmatrix} s(\mathbf{X})\mathbf{R}(\mathbf{X}) & \mathbf{t}(\mathbf{X}) \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix}, \tag{2}$$

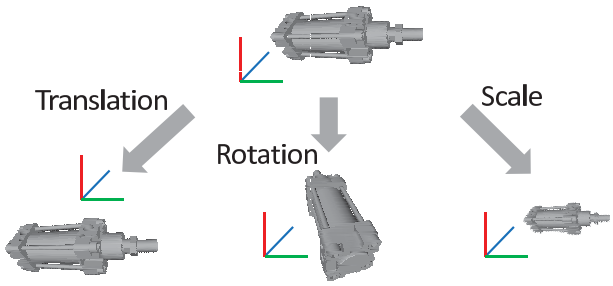


Fig. 2 Effects of translation, rotation, and scale in transforming the pose of an object.

converting the coordinates of a 3D point from the global coordinate system to the object frame. Since object frame and object pose are equivalent terms, from here onwards they are used interchangeably.

An training feature consisting of a feature descriptor $\mathbf{d}$, an object class identity $j \in \{1,..,J\}$ (where J is the number of classes), and a feature-to-object transformation $\mathbf{T} = \mathbf{X}\mathbf{F}^{-1}$ is formed for each detected interest region. Note that although both matrices $\mathbf{F}$ and $\mathbf{X}$ are computed from the global coordinate system, since the feature frame is covariant to the object pose, the resultant feature-to-object transformation $\mathbf{T}$ solely depends on the shape of the object. In other words, it is pose-invariant.

The collection of all training features extracted from every object class, denoted as $e_m = (\mathbf{d}_m, j_m, \mathbf{T}_m = \mathbf{X}_m\mathbf{F}_m^{-1})$ for $m \in \{1,..,M\}$ where M is the number of training features, can be viewed as the local appearance model of the object classes.

2.2.2 Learning the Feature-to-Vote Mapping

Existing approaches differ in how the votes are generated from features extracted from the scene, hereinafter *scene feature*s. There are three main approaches: (1) direct matching of scene features with training features, (2) unsupervised clustering of training features into visual words followed by matching of scene features with visual words, and (3) supervised learning to directly map each scene feature to one or more votes.

In approaches that match scene features with training features directly, all the training features are kept as a library of exemplars. Hence, the cost for training is very low. During inference, each scene feature $f = (\mathbf{d}^{'}, \mathbf{F}^{'})$ (with descriptor $\mathbf{d}^{'}$ and feature frame $\mathbf{F}^{'}$) is matched with every exemplar in the library and each match generates a vote. In Tombari and Di Stefano's work [41], matching of f with an exemplar e_m is done by thresholding the Euclidean distance $\|\mathbf{d}_m - \mathbf{d}'\|$ with a predefined threshold ε. If it is a match, *i.e.*, $\|\mathbf{d}_m - \mathbf{d}'\| < \varepsilon$, a vote for an object of class j_m at pose $\mathbf{T}_m\mathbf{F}^{'}$ is generated. Drost *etal*. [11] use hashing to match a feature descriptor with model descriptors instead. A vote may optionally have a weight to reflect the relative matching score and other prior probabilities, as shown in, for instance, the work of Knopp *etal*. [20]. It makes sense to use $\mathbf{T}_m\mathbf{F}^{'}$ to predict the object pose,

since if we transform the object specified by e_m, albeit unknown, so that the feature used for the construction of e_m aligns perfectly with f, *i.e.*, $\mathbf{F}_m = \mathbf{F}^{'}$, the transformed object pose must be $\mathbf{T}_m\mathbf{F}^{'}$.

Since matching every scene feature with every training feature is a time-costly process, it may be beneficial to group training features of similar descriptors coming from the same class into a visual word using an unsupervised clustering approach [20, 21]. Such a strategy would reduce the number of exemplars in the library, hence increasing the matching efficiency. However, the sizes of the clusters must be chosen carefully, or the false positive rate may increase [21, 41].

In 2D object recognition, the idea of grouping training features of similar appearances is advanced further, by using discriminative and supervised clustering rather than unsupervised clustering, allowing one to optimize the visual words to produce more reliable votes in the vote space. Gall and Lempitsky [16] train a Hough forest that maps a feature directly to multiple votes. However, each node of their Hough tree is trained to either minimize the class uncertainty or the pose uncertainty. Okada [27] instead introduces a combined objective function for training a node. Both methods have shown significant improvements over the unsupervised approach of Leibe *etal*. [21].

It would be tempting to apply this idea to 3D. However, an immediate challenge is how to model uncertainty of a set of 3D poses. In 2D, Gall and Lempitsky work with 2D center points, and Okada works with 2D points plus scale, the variance of which is sufficient to model the uncertainty. In our case, the existence of both 3D rotation and scale makes the pose space a non-linear manifold. Any uncertainty measurement based on the notion of Euclidean distance, including variance, would have a bias in scale, as to be discussed in Sect. 2.4.1.

In our approach, we use a standard feature extraction process, as described in Sect. 4. Similar to Tombari and Di Stefano [41], we use the dataset of training features as the local appearance model without clustering them into visual words. As the scope of this work is to introduce a distance that is efficient and more importantly, unbiased by scale, the task of modeling pose uncertainty, and subsequently supervised learning of visual words, is left for future work.

2.3 Finding Local Modes in the Vote Space

The inference stage involves the computation of local maxima of the joint kernel density function $p(j,\mathbf{X})$ of object identity $j \in \{1,..,J\}$ and pose $\mathbf{X} \in S^{+}(3)$ represented by a set of (weighted) votes generated from an input point cloud. Since j is a discrete variable, we can search for local maxima in each of the J conditional distributions $p(\mathbf{X}|j)$ instead. Without loss of generality, let us assume the form of $p(\mathbf{X}|j)$ as:

$$p(\mathbf{X}|j) = \sum_{i=1}^{N_j} \lambda_i H(\mathbf{X}, \mathbf{X}_i) \tag{3}$$

Table 1 Methods of pose estimation over different transformations. **t**: translation; **R**: rotation; *s*: scale. *Indicates 2D space.

	t	**t**, *s*	**t**, **R**	**t**, **R**, *s*
Hough	[41]	[20]	[13]	[19]
Mean shift	–	[21, 28, 37]*	[40]	[39]*, **This work**

where N_j denotes the number of votes for class j, $H(\cdot,\cdot)$ denotes a kernel function, and with respect to the i^{th} vote for class j, $\lambda_i \geq 0$ denotes the weight and $\mathbf{X}_i \in S^+(3)$ denotes the predicted pose. Here, the weights are normalized, *i.e.*, $\sum_i \lambda_i = 1$, so that $p(\mathbf{X}|j)$ is a proper probability density function. Although we are concerned with 3D transformations, the discussion in the remainder of the chapter assumes n-dimensional transformations for an arbitrary $n > 0$.

Two main techniques for finding modes of $p(\mathbf{X}|j)$ are Hough voting (an extension of the Generalized Hough Transform [4]) and mean shift [9].

In Hough voting, the input space is partitioned into a finite number of L bins, *i.e.*, $S^+(n) = \bigcup_{l=1}^{L} B_l$ where $B_i \cap B_j = \emptyset$ for all $i \neq j$. For each bin B_l, the weights of the votes with poses belonging to B_l are summed up. Modes are found by returning bins with largest sums of weights. Using Hough voting, Khoshelham [19] quantizes the 7D space of 3D translation, rotation and scale for object registration. This creates a trade-off between pose accuracy and computational requirements, the latter proving to be costly. Other methods seek to reduce this complexity by shrinking the pose space and marginalizing over some parameters. Fisher *etal*. [13] quantize translations and rotations in two separate 3D arrays; peak entries in both arrays indicate the pose of the object, but multiple objects create ambiguities. Knopp *etal*. [20] show effective object recognition using Hough voting over 3D translation and scale. Tombari and Di Stefano [41] first compute Hough votes over translation, assuming known scale in their 3D object recognition and registration application, then determine rotation by averaging the rotations at each mode. Geometric hashing [11, 24] is a similar technique to Hough voting which reparameterizes pose in a lower dimensional space before clustering. However, all these dimensionality reduction techniques lead to an increased chance of false positive detections. Another issue with Hough voting is that it returns bins, not poses, as output. One still needs a way to select the best pose, or to compute a representative pose, for each output bin.

Mean shift avoids the trade-off suffered by Hough voting methods, being both accurate and having lower (usually[1] linear) complexity in the pose dimensionality, making it suitable for inference in the full 7D pose space of the direct similarity group in 3D. To date it has been used in 2D applications: object detection over translation and scale [21, 28, 37], and motion segmentation over affine transformations [40], as well as in 3D for motion segmentation over translation and rotation [39]. Mean shift relies on a kernel function typically in the form,

[1] Certain distance computations are not linear, *e.g.*, that of Sect. 2.4.2.

$$H(\mathbf{X},\mathbf{X}_i) = \frac{1}{\zeta} K(d^2(\mathbf{X},\mathbf{X}_i)), \tag{4}$$

where $d(\cdot,\cdot)$ is a distance function and $K(\cdot)$ is a non-negative non-increasing univariate function, and ζ is a normalization factor so that $H(\cdot,\mathbf{X}_i)$ is a proper probability density function. Choosing the distance function $d(\cdot,\cdot)$, is crucial for mean shift as it directly changes the locations and the number of the output modes. On non-Euclidean spaces, even the Euclidean distance yields undesired behaviours as to be shown in Sect. 2.4. This is the first contribution we know of to apply mean shift to a 3D application using translation, rotation and scale simultaneously. A reason this has not done before could be the problems associated with computing means using existing Euclidean and Riemannian distances in the direct similarity group $S^+(n)$. We now review mean shift in more details and discuss distance functions in this space. In what follows, we omit index j since it is clear from the context that j is given.

Algorithm 1. Mean shift [9] (for notation see text)

Require: $\mathcal{X} = \{\mathbf{X}_i, \lambda_i\}_{i=1}^N$, distance function $d(\cdot,\cdot)$
1: Initialize $\mathbf{X}$
2: **repeat**
3: $\quad \mathbf{X}_{\text{old}} := \mathbf{X}$
4: $\quad w_i := \lambda_i K(d^2(\mathbf{X},\mathbf{X}_i)) \qquad \forall i = 1,..,N$
5: $\quad \mathbf{X} := \arg\min_{\mathbf{X}} \sum_i w_i d^2(\mathbf{X},\mathbf{X}_i)$
6: **until** $d(\mathbf{X}_{\text{old}},\mathbf{X}) < \varepsilon$
7: **return** $\mathbf{X}$

2.4 Mean Shift

The mean shift algorithm [9] (Algorithm 1) is a popular algorithm for finding local modes by coordinate ascent in kernel density estimation. Given a distance function $d(\cdot,\cdot)$ on the input space, the kernel density estimator is given by

$$\hat{f}_K(\mathbf{X}) = \sum_{i=1}^{N} \frac{1}{\zeta} \lambda_i K(d^2(\mathbf{X},\mathbf{X}_i)), \tag{5}$$

where $\mathbf{X}$ is the random variable, $\mathcal{X} = \{\mathbf{X}_i, \lambda_i\}_{i=1}^N$ is a set of input points with weights $\lambda_i \geq 0$, $K(\cdot) \geq 0$ is a kernel function, and ζ is a volume density function which normalizes $K(d^2(\cdot,\mathbf{X}_i))$. The most common (and our) choice for $K(\cdot)$ is the Gaussian kernel, $\exp\left(-\frac{\cdot}{2\sigma^2}\right)$, where σ is the bandwidth of the kernel. On Euclidean spaces a natural choice for $d()$ is the Euclidean distance, $d_{\text{E}}()$; *e.g.*, if $\mathbf{X}$ and $\mathbf{Y}$ are matrices, $d_{\text{E}}(\mathbf{X},\mathbf{Y}) = \|\mathbf{X}-\mathbf{Y}\|_{\text{F}}$ where $\|\cdot\|_{\text{F}}$ is the Frobenius norm. Under the Euclidean distance, the solution of step 5 in Algorithm 1,

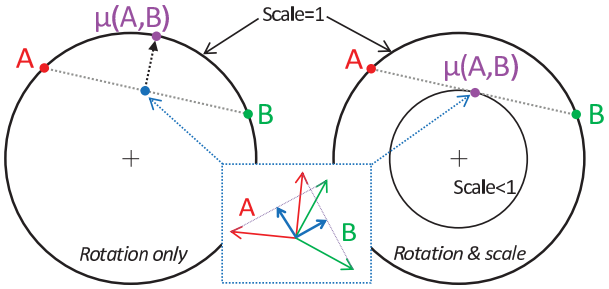


Fig. 3 Scale bias of the extrinsic mean. Let us consider $S^+(2)$ (without translation): on a plane, a rotation can be represented as a point on a circle, the radius being the scale. *Left*: with rotation only, the arithmetic mean of **A** and **B** leads to a smaller scale but the reprojection onto the manifold (*i.e.*, the unit circle) gives a reasonable result. *Right*: with rotation and scale, the mean is already on the manifold, but with a smaller scale.

$$\mu(\mathcal{X}) = \arg\min_{\mathbf{X}} \sum_i w_i d^2(\mathbf{X}, \mathbf{X}_i), \tag{6}$$

also known in the literature as a Fréchet mean [14], becomes an arithmetic mean, *i.e.*, $\mu(\mathcal{X}) = \frac{\sum_i w_i \mathbf{X}_i}{\sum_i w_i}$.

In pose estimation, votes are represented by linear transformations which form a matrix Lie group. This chapter is concerned with the direct similarity group $S^+(n) \subset GL(n+1,\mathbb{R})$, which is the set of all affine transformation matrices $\mathbf{X} \in S^+(n)$ acting on $\mathbb{R}^n$ preserving angles and orientations [36]. When applying mean shift on a matrix Lie group, the choice of $d()$ is crucial since it affects both the computation of weights and the mean (steps 4 & 5 of Algorithm 1). Two well-known distances arise in the literature: Euclidean and Riemannian. We now review how existing methods utilize these distances in mean shift on matrix Lie groups.

2.4.1 Euclidean Distance

Given a matrix Lie group $\mathcal{G} \subset GL(n,\mathbb{R})$, since $GL(n,\mathbb{R}) \subset \mathbb{R}^{n^2}$ (up to an isomorphism), the most straightforward way to apply mean shift on $\mathcal{G}$ is to run Euclidean mean shift on $\mathbb{R}^{n^2}$ instead. However, at each iteration the arithmetic mean may not lie in $\mathcal{G}$. It is therefore projected back to $\mathcal{G}$ via the mapping:

$$\pi : \mathbb{R}^{n^2} \to \mathcal{G} : \pi(\mathbf{X}) = \arg\min_{\mathbf{Y} \in \mathcal{G}} \|\mathbf{Y} - \mathbf{X}\|_{\mathrm{F}}^2. \tag{7}$$

The projected arithmetic mean, $\mu(\mathcal{X}) = \pi\left(\frac{\sum_i w_i \mathbf{X}_i}{\sum_i w_i}\right)$, is referred to in the literature as the *extrinsic* mean [25, 38].

Mean shift using Euclidean distance (extrinsic mean shift) has shown good results on Stiefel and Grassmann manifolds [7]. However, there are two drawbacks with extrinsic mean shift applied to $S^+(n)$. First, $d_{\mathrm{E}}()$ is invariant to rotation and translation but not to scaling, making the weights, w_i, computed by mean shift scale

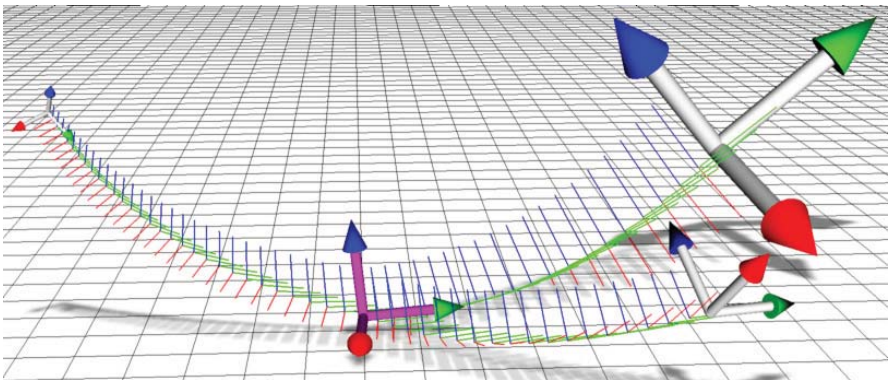


Fig. 4 The intrinsic mean. Three poses in $S^+(3)$ (with different scales, rotations and translations) and their intrinsic mean (pink). The geodesics between the mean and input poses are also drawn. Note that the shortest distance between two transformations is not necessarily a straight line in terms of translation.

variant. Thus, although the extrinsic mean is scale-covariant[2], extrinsic mean shift is *not*. Second, the extrinsic mean of rotation and scale transformations causes a bias towards smaller scales, as illustrated in Fig. 3.

2.4.2 Riemannian Distance

An alternative choice for $d()$ is the Riemannian distance. $d_{\mathrm{R}}()$. Given $\mathbf{X}, \mathbf{Y} \in S^+(n)$, $d_{\mathrm{R}}(\mathbf{X}, \mathbf{Y})$ is defined as the arclength of the geodesic between $\mathbf{X}$ and $\mathbf{Y}$, *i.e.*, the shortest curve along the manifold connecting $\mathbf{X}$ and $\mathbf{Y}$ (see Fig. 4). In general $d_{\mathrm{R}}(\mathbf{X}, \mathbf{Y})$ is difficult to compute, but if $\mathbf{Y}$ is located within the open neighbourhood bounded by the cut locus of $\mathbf{X}$ in $S^+(n)$ (defined in [31]) then $d_{\mathrm{R}}(\mathbf{X}, \mathbf{Y}) = \left\|\mathrm{logm}\left(\mathbf{X}^{-1}\mathbf{Y}\right)\right\|_{\mathrm{F}}$, where $\mathrm{logm}(\cdot)$ is the matrix logarithm. This requirement is not too restrictive in practice; in $S^+(n)$ the rotation angle should not reach π radians [45]. For example, in $SO(2, \mathbb{R})$ the cut locus of $\mathbf{X}$ is just a single point: $-\mathbf{X}$ [25].

Since $d_{\mathrm{R}}()$ depends only on the *intrinsic* geometry of $\mathcal{G}$, the Fréchet mean (*i.e.*, mean defined as the solution of Eq. (6)) using $d_{\mathrm{R}}()$ is called the intrinsic mean [25, 31]. Efficient formulations of $d_{\mathrm{R}}()$ exist for some $\mathcal{G}$, notably $SE(3)$ [2], which can be adapted to $S^+(3)$. However, in $S^+(n)$ for $n > 3$, $d_{\mathrm{R}}()$ generally has no efficient formulation, taking $O(n^4)$ time to compute [10].

Intrinsic mean shift methods have been proposed [7, 40]. The intrinsic mean itself has multiple non-closed-form solutions [25]; in our experiments we compute an approximation using a single step[3] of the iterative method of [40].

[2] Scale-covariant means a scale transformation of input data produces the same transformation on the output.

[3] This is equivalent to computing a mean using the *log-Euclidean* distance [3], $d(\mathbf{X}, \mathbf{Y}) = \|\mathrm{logm}(\mathbf{X}) - \mathrm{logm}(\mathbf{Y})\|_{\mathrm{F}}$.

2.4.3 Properties of a Good Distance in $S^+(n)$

In the context of mean shift, and subsequent to our overview of Euclidean and Riemannian distances, we propose the following list of desirable properties for a distance in $S^+(n)$ and its associated mean:

1. *Unique:* The mean should have a unique solution.
2. *Closed-form:* For efficient computation, the mean should have a closed-form solution.
3. *Scale-compatible:* If all rotations and translations are equal, the mean should behave as an average of the scales. Mathematically, if $\forall \mathbf{X}_i \in \mathcal{X} : \mathbf{R}(\mathbf{X}_i) = \mathbf{R}', \mathbf{t}(\mathbf{X}_i) = \mathbf{t}'$ for some $\mathbf{R}'$ and $\mathbf{t}'$, then we would like $\mathbf{R}(\mu(\mathcal{X})) = \mathbf{R}'$, $\mathbf{t}(\mu(\mathcal{X})) = \mathbf{t}'$, and $s(\mu(\mathcal{X}))$ to be an average of $s(\mathbf{X}_i)$'s. In this case, we say that μ is scale-compatible.
4. *Rotation-compatible:* If $\forall \mathbf{X}_i \in \mathcal{X} : s(\mathbf{X}_i) = s'$, $\mathbf{t}(\mathbf{X}_i) = \mathbf{t}'$, then $s(\mu(\mathcal{X})) = s'$, $\mathbf{t}(\mu(\mathcal{X})) = \mathbf{t}'$ and $\mathbf{R}(\mu(\mathcal{X}))$ is an average of $\mathbf{R}(\mathbf{X}_i)$'s.
5. *Translation-compatible:* If $\forall \mathbf{X}_i \in \mathcal{X} : s(\mathbf{X}_i) = s'$, $\mathbf{R}(\mathbf{X}_i) = \mathbf{R}'$, then $s(\mu(\mathcal{X})) = s'$, $\mathbf{R}(\mu(\mathcal{X})) = \mathbf{R}'$ and $\mathbf{t}(\mu(\mathcal{X}))$ is an average of $\mathbf{t}(\mathbf{X}_i)$'s.
6. *Left-invariant:* A left-invariant distance is one that is unchanged by any post-transformation, *i.e.*, $d(\mathbf{ZX}, \mathbf{ZY}) = d(\mathbf{X}, \mathbf{Y})$ $\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in S^+(n)$. This property is crucial for two reasons: (a) it leads to a left-covariant mean: $\mu(\mathbf{Z}\mathcal{X}) = \mathbf{Z}\mu(\mathcal{X})$[4]; *i.e.*, if all poses $\mathbf{X}_i$ are transformed by $\mathbf{Z}$, the mean is transformed by $\mathbf{Z}$ as well, and (b) it ensures that the weights w_i computed in mean shift are invariant to any post-transformation $\mathbf{Z}$, leading to left-covariant mean shift.

A symmetric distance, *s.t.* $d(\mathbf{X}, \mathbf{Y}) = d(\mathbf{Y}, \mathbf{X})$ $\forall \mathbf{X}, \mathbf{Y} \in S^+(n)$, intuitively seems desirable, but its absence does not prevent a distance from being used in mean shift and furthermore, given the properties listed, it is not necessary. In other words, we do not require the distance function to be a metric. Right-invariance might also be considered a desirable property, but in the context of 3D recognition this occurrence does not relate to any meaningful behaviour.

3 The SRT Distance and Its Mean

In this section, we describe our new distance on $S^+(n)$, which fulfills all the desirable properties defined in Sect. 2.4.3. We call it the SRT distance, with corresponding mean μ_{SRT}.

3.1 Distance Definition

We first define the following component-wise distances:

$$d_s(\mathbf{X}, \mathbf{Y}) = \left| \log \left(\frac{s(\mathbf{X})}{s(\mathbf{Y})} \right) \right|, \tag{8}$$

[4] $\mathbf{Z}\mathcal{X} = \{\mathbf{ZX} : \mathbf{X} \in \mathcal{X}\}$ is a left coset of $\mathcal{X}$. Proof in [32, Append. A.4].

Table 2 Properties of distances and associated means in $S^+(n)$. †The approximation of [40] is, however, unique and translation compatible.

Properties	Extrinsic	Intrinsic	SRT
Distance:			
Symmetric	✓	✓	✗
Left-invariant	✗	✓	✓
Mean:			
Unique	✓	✗†	✓
Closed-form	✓	✗	✓
Scale-compatible	✓	✓	✓
Rotation-compatible	✗	✓	✓
Translation-compatible	✓	✗†	✓

$$d_r(\mathbf{X},\mathbf{Y}) = \|\mathbf{R}(\mathbf{X}) - \mathbf{R}(\mathbf{Y})\|_{\mathrm{F}}, \tag{9}$$

$$d_t(\mathbf{X},\mathbf{Y}) = \frac{\|\mathbf{t}(\mathbf{X}) - \mathbf{t}(\mathbf{Y})\|}{s(\mathbf{Y})}, \tag{10}$$

in which $d_s()$, $d_r()$ and $d_t()$ measure scale, rotation and translation distances respectively, with $\mathbf{X}$ and $\mathbf{Y}$ in $S^+(n)$. Given some bandwidth coefficients $\sigma_s, \sigma_r, \sigma_t > 0$, the SRT distance is defined as:

$$d_{\mathrm{SRT}}(\mathbf{X},\mathbf{Y}) = \sqrt{\frac{d_s^2(\mathbf{X},\mathbf{Y})}{\sigma_s^2} + \frac{d_r^2(\mathbf{X},\mathbf{Y})}{\sigma_r^2} + \frac{d_t^2(\mathbf{X},\mathbf{Y})}{\sigma_t^2}}. \tag{11}$$

By controlling $\sigma_s, \sigma_r, \sigma_t$, it is possible to create an SRT distance that is more sensitive to one type of transformations among scale, rotation, and translation than the others. In this sense, the SRT distance is more flexible than the Euclidean and Riemannian distances.

We now prove that the SRT distance possesses the most crucial property, the 6th property in the list.

Theorem 1. *$d_{SRT}()$ is left-invariant.*

Proof. The main idea involves showing that $d_{\mathrm{SRT}}()$ is related to a pseudo-seminorm on $S^+(n)$, *i.e.*, $d_{\mathrm{SRT}}(\mathbf{X},\mathbf{Y}) = \left\|\mathbf{Y}^{-1}\mathbf{X}\right\|_{\mathrm{SRT}}$, where

$$\|\cdot\|_{\mathrm{SRT}} = \sqrt{\frac{\log^2(s(\cdot))}{\sigma_s^2} + \frac{\|\mathbf{R}(\cdot) - \mathbf{I}\|_{\mathrm{F}}^2}{\sigma_r^2} + \frac{\|\mathbf{t}(\cdot)\|^2}{\sigma_t^2}}. \tag{12}$$

Indeed, for all $\mathbf{X}, \mathbf{Y} \in S^+(n)$, the transformation $\mathbf{Y}^{-1}\mathbf{X}$ consists of:

$$s\left(\mathbf{Y}^{-1}\mathbf{X}\right) = \frac{s(\mathbf{X})}{s(\mathbf{Y})}, \tag{13}$$

$$\mathbf{R}\left(\mathbf{Y}^{-1}\mathbf{X}\right) = \mathbf{R}^T(\mathbf{Y})\mathbf{R}(\mathbf{X}), \tag{14}$$

$$\mathbf{t}\left(\mathbf{Y}^{-1}\mathbf{X}\right) = \frac{\mathbf{R}^T(\mathbf{Y})(\mathbf{t}(\mathbf{X}) - \mathbf{t}(\mathbf{Y}))}{s(\mathbf{Y})}. \tag{15}$$

Applying the $\|\cdot\|_{\mathrm{SRT}}$ norm on $\mathbf{Y}^{-1}\mathbf{X}$ yields:

$$\begin{aligned}\left\|\mathbf{Y}^{-1}\mathbf{X}\right\|_{\mathrm{SRT}}^2 = \frac{1}{\sigma_s^2}\log^2\left(\frac{s(\mathbf{X})}{s(\mathbf{Y})}\right) + \frac{1}{\sigma_r^2}\left\|\mathbf{R}^T(\mathbf{Y})\mathbf{R}(\mathbf{X}) - \mathbf{I}\right\|_{\mathrm{F}}^2, \\ + \frac{1}{\sigma_t^2}\left\|\frac{\mathbf{R}^T(\mathbf{Y})(\mathbf{t}(\mathbf{X}) - \mathbf{t}(\mathbf{Y}))}{s(\mathbf{Y})}\right\|^2.\end{aligned} \tag{16}$$

Since the Frobenius norm is rotation invariant, the second and third terms of the right-hand side of Eq. (16) may be rewritten as:

$$\frac{1}{\sigma_r^2}\left\|\mathbf{R}^T(\mathbf{Y})\mathbf{R}(\mathbf{X}) - \mathbf{I}\right\|_{\mathrm{F}}^2 = \frac{1}{\sigma_r^2}\left\|\mathbf{R}(\mathbf{X}) - \mathbf{R}(\mathbf{Y})\right\|_{\mathrm{F}}^2, \tag{17}$$

$$\frac{1}{\sigma_t^2}\left\|\frac{\mathbf{R}^T(\mathbf{Y})(\mathbf{t}(\mathbf{X}) - \mathbf{t}(\mathbf{Y}))}{s(\mathbf{Y})}\right\|^2 = \frac{1}{\sigma_t^2}\left\|\frac{\mathbf{t}(\mathbf{X}) - \mathbf{t}(\mathbf{Y})}{s(\mathbf{Y})}\right\|^2. \tag{18}$$

proving $d_{\mathrm{SRT}}(\mathbf{X}, \mathbf{Y}) = \left\|\mathbf{Y}^{-1}\mathbf{X}\right\|_{\mathrm{SRT}}$. It follows that:

$$d_{\mathrm{SRT}}(\mathbf{X}, \mathbf{Y}) = \left\|\mathbf{Y}^{-1}\mathbf{X}\right\|_{\mathrm{SRT}} = \left\|\left(\mathbf{X}_i^{-1}\mathbf{Z}^{-1}\right)(\mathbf{Z}\mathbf{X})\right\|_{\mathrm{SRT}} = d_{\mathrm{SRT}}(\mathbf{Z}\mathbf{X}, \mathbf{Z}\mathbf{Y}), \tag{19}$$

proving $d_{\mathrm{SRT}}()$ is left invariant. □

Note that, unlike $d_{\mathrm{E}}()$ and $d_{\mathrm{R}}()$, $d_{\mathrm{SRT}}()$ is not symmetric; it could be made symmetric by a slight modification of the translation component, but at the expense of the translation-compatibility of the corresponding mean.

3.2 Mean Computation

Having defined $d_{\mathrm{SRT}}()$, we now derive the Fréchet mean μ_{SRT} using $d_{\mathrm{SRT}}()$, which is:

$$\mu_{\mathrm{SRT}}(\mathcal{X}) = \underset{\mathbf{X} \in S^+(n)}{\arg\min} \sum_i w_i d_{\mathrm{SRT}}^2(\mathbf{X}, \mathbf{X}_i). \tag{20}$$

and show that it is closed-form[5] and generally unique.

[5] Our close-form notion includes matrix singular-value decomposition.

Theorem 2. *The solution of Eq. (20), the SRT mean, is given as:*

$$s(\mu_{SRT}(\mathcal{X})) = \exp\left(\frac{\sum_i w_i \log s(\mathbf{X}_i)}{\sum_i w_i}\right), \tag{21}$$

$$\mathbf{R}(\mu_{SRT}(\mathcal{X})) = \mathrm{sop}\left(\frac{\sum_i w_i \mathbf{R}(\mathbf{X}_i)}{\sum_i w_i}\right), \tag{22}$$

$$\mathbf{t}(\mu_{SRT}(\mathcal{X})) = \sum_i \frac{w_i \mathbf{t}(\mathbf{X}_i)}{s^2(\mathbf{X}_i)} \Bigg/ \sum_i \frac{w_i}{s^2(\mathbf{X}_i)} \tag{23}$$

where $\mathrm{sop}(\mathbf{X}) = \arg\min_{\mathbf{Y}\in SO(n,\mathbb{R})} \|\mathbf{Y}-\mathbf{X}\|_{\mathrm{F}}$ *is the orthogonal projection of matrix* $\mathbf{X}$ *onto* $SO(n,\mathbb{R})$. *Additionally if* $\mathbf{X}$ *is singular-value decomposed into* $\mathbf{X} = \mathbf{U}diag(\lambda_1 \ldots, \lambda_n)\mathbf{V}^{\mathsf{T}}$ *for some orthogonal matrices* $\mathbf{U},\mathbf{V} \in O(n,\mathbb{R})$ *and singular values* $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$. *the function* $\mathrm{sop}(\mathbf{X})$ *computes*

$$\mathrm{sop}(\mathbf{X}) = \mathbf{U}diag(1,..,1,\det(\mathbf{U}\mathbf{V}))\,\mathbf{V}^{\mathsf{T}}. \tag{24}$$

The SRT mean is unique if and only if all the singular values are distinct.

Proof. The sum in Eq. (20) can be rewritten as

$$\sum_i w_i d^2_{\mathrm{SRT}}(\mathbf{X},\mathbf{X}_i) = \frac{F_s(\mathbf{X})}{\sigma_s^2} + \frac{F_r(\mathbf{X})}{\sigma_r^2} + \frac{F_t(\mathbf{X})}{\sigma_t^2}, \tag{25}$$

where[6] $F_\star(\mathbf{X}) = \sum_{i=1}^N w_i d^2_\star(\mathbf{X},\mathbf{X}_i)$. Since $s(\mathbf{X})$ only appears in $F_s(\mathbf{X})$, we can reformulate

$$s(\mu_{\mathrm{SRT}}(\mathcal{X})) = \underset{s(\mathbf{X})\in\mathbb{R}^+}{\arg\min} \sum_i w_i \log^2\left(\frac{s(\mathbf{X})}{s(\mathbf{X}_i)}\right), \tag{26}$$

yielding the solution (21). Similarly, since $\mathbf{t}(\mathbf{X})$ only appears in $F_r(\mathbf{X})$, after rewriting

$$\mathbf{t}(\mu_{\mathrm{SRT}}(\mathcal{X})) = \underset{\mathbf{t}(\mathbf{X})\in\mathbb{R}^n}{\arg\min} \sum_i w_i \frac{\|\mathbf{t}(\mathbf{X}) - \mathbf{t}(\mathbf{X}_i)\|^2}{s^2(\mathbf{X}_i)}, \tag{27}$$

we get Eq. (23) as the solution. Finally, since $\mathbf{R}(\mathbf{X})$ only appears in $F_r(\mathbf{X})$, we rewrite

$$\mathbf{R}(\mu_{\mathrm{SRT}}(\mathcal{X})) = \underset{\mathbf{R}(\mathbf{X})\in SO(n,\mathbb{R})}{\arg\min} \sum_i w_i \|\mathbf{R}(\mathbf{X}) - \mathbf{R}(\mathbf{X}_i)\|^2_{\mathrm{F}}. \tag{28}$$

This is precisely Moakher's definition of Euclidean (extrinsic) mean of 3D rotation matrices [25, def. 5.1] generalized to n-dimensional rotation matrices. Moakher proves that for the case of $n = 3$ [25, Sect. 3.1],

$$\mathbf{R}(\mu_{\mathrm{SRT}}(\mathcal{X})) = \mathrm{sop}(\overline{\mathbf{R}}) = \underset{\mathbf{Y}\in SO(n,\mathbb{R})}{\arg\min} \left\|\mathbf{Y} - \overline{\mathbf{R}}\right\|_{\mathrm{F}}, \tag{29}$$

[6] $\star$ should be replaced with s, r or t.

where $\overline{\mathbf{R}} = \sum_i w_i \mathbf{R}(\mathbf{X}_i)$, by showing that

$$\sum_i w_i \left\|\mathbf{R}(\mathbf{X}) - \mathbf{R}(\mathbf{X}_i)\right\|_{\mathrm{F}}^2 = \left(\sum_i w_i\right) \left\|\mathbf{R}(\mathbf{X}) - \overline{\mathbf{R}}\right\|_{\mathrm{F}}^2 + \sum_i w_i \left\|\overline{\mathbf{R}} - \mathbf{R}(\mathbf{X}_i)\right\|_{\mathrm{F}}^2. \quad (30)$$

which is straightforwardly generalized to the case of $n \neq 3$.

Finding $\mathrm{sop}(\overline{\mathbf{R}})$ when $n = 3$ is studied in [12, 25]. Here, we generalize the results to $SO(n, \mathbb{R})$. First, let the singular value decomposition of $\overline{\mathbf{R}}$ be

$$\overline{\mathbf{R}} = \mathbf{U}\mathrm{diag}\left(\lambda_1 .., \lambda_n\right) \mathbf{V}^{\mathsf{T}}, \quad (31)$$

for some orthogonal matrices $\mathbf{U}, \mathbf{V} \in O(n, \mathbb{R})$ and unique (but not necessarily distinct) singular values $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$. Considering a change of variable $\mathbf{R}^{'} = \mathbf{U}^{\mathsf{T}}\mathbf{R}(\mathbf{X})\mathbf{V}$, we get:

$$\mathbf{U}^{\mathsf{T}}\overline{\mathbf{R}}\mathbf{V} = \mathrm{diag}\left(\lambda_1 .., \lambda_n\right), \quad (32)$$

$$\left\|\mathbf{R}(\mathbf{X}) - \overline{\mathbf{R}}\right\|_{\mathrm{F}}^2 = \left\|\mathbf{U}^{\mathsf{T}}\left(\mathbf{R}(\mathbf{X}) - \overline{\mathbf{R}}\right)\mathbf{V}\right\|_{\mathrm{F}}^2 = \left\|\mathbf{R}^{'} - \mathrm{diag}\left(\lambda_1 \ldots, \lambda_n\right)\right\|_{\mathrm{F}}^2. \quad (33)$$

Thus, minimizing $\left\|\mathbf{R}(\mathbf{X}) - \overline{\mathbf{R}}\right\|_{\mathrm{F}}^2$ with respect to $\mathbf{R}(\mathbf{X})$ is equivalent to minimizing $f(\mathbf{R}^{'}) = \left\|\mathbf{R}^{'} - \mathrm{diag}\left(\lambda_1 .., \lambda_n\right)\right\|_{\mathrm{F}}^2$ with respect to $\mathbf{R}^{'}$. Here, $\mathbf{R}^{'} \in O(n, \mathbb{R})$ and $\det(\mathbf{R}^{'}) = \det(\mathbf{UV})$. Rewriting function $f(\mathbf{R}^{'})$:

$$f(\mathbf{R}^{'}) = \mathrm{trace}\left(\mathbf{I} - 2\mathbf{R}^{'\mathsf{T}}\mathrm{diag}\left(\lambda_1 .., \lambda_n\right) + \mathrm{diag}^2\left(\lambda_1 \ldots, \lambda_n\right)\right) \quad (34)$$

$$= \sum_{i=1}^{n} \left(1 - 2\mathbf{R}^{'}_{i,i}\lambda_i + \lambda_i^2\right), \quad (35)$$

we can see that only the diagonal elements of $\mathbf{R}^{'}$ are involved in $f(\mathbf{R}^{'})$. Therefore, the optimal $\mathbf{R}^{'}$ must be a diagonal orthogonal matrix. Among all the diagonal orthogonal matrices available in $O(n, \mathbb{R})$ (there are 2^n in total), the one that minimizes $f(\mathbf{R}^{'})$ and has $\det(\mathbf{R}^{'}) = \det(\mathbf{UV})$, considering that λ_n is the smallest singular value, is given by

$$\mathbf{R}' = \mathrm{diag}\left(1, .., 1, \det(\mathbf{UV})\right). \quad (36)$$

In other words,

$$\mathrm{sop}(\overline{\mathbf{R}}) = \mathbf{U}\mathrm{diag}\left(1, .., 1, \det(\mathbf{UV})\right)\mathbf{V}^{\mathsf{T}}. \quad (37)$$

We now analyze the uniqueness of $\mathrm{sop}(\overline{\mathbf{R}})$. First, if some singular values are not distinct, *i.e.*, $\lambda_k = \lambda_{k+1}$, then the k^{th} and $(k+1)^{\text{th}}$ columns of matrices $\mathbf{U}$ and $\mathbf{V}$ of Eq. (31) become non-unique, making $\mathrm{sop}(\overline{\mathbf{R}})$ non-unique. Second, in the case that all the singular values are distinct, if $\lambda_n > 0$ then both $\mathbf{U}$ and $\mathbf{V}$ are unique, making $\mathrm{sop}(\overline{\mathbf{R}})$ unique. If $\lambda_n = 0$, the n^{th} column of $\mathbf{U}$ and the n^{th} column of $\mathbf{V}$ can both be negated while still satisfying Eq. (31) (their directions are fixed by other columns). However, $\mathrm{sop}(\overline{\mathbf{R}})$ remains unchanged due to the simultanous negation of

both singular vectors. Therefore, $\text{sop}(\overline{\mathbf{R}})$ is unique if and only if all the singular values of $\overline{\mathbf{R}}$ are distinct. □

It can be further verified that $\mu_{\text{SRT}}(\mathcal{X})$ is:

1. *Scale-compatible*: The scale component $s(\mu_{\text{SRT}}(\mathcal{X}))$ is a **geometric mean** of $s(\mathbf{X}_i)$'s.
2. *Rotation-compatible*: The rotation component $\mathbf{R}(\mu_{\text{SRT}}(\mathcal{X}))$ is an **extrinsic mean** of $\mathbf{R}(\mathbf{X}_i)$'s.
3. *Translation-compatible*: The translation component $\mathbf{t}(\mu_{\text{SRT}}(\mathcal{X}))$ is an **arithmetic mean** of $\mathbf{t}(\mathbf{X}_i)$'s.

Table 2 summarizes the desirable properties of the SRT distance and mean, and contrasts them with those of the Euclidean and Riemannian distances.

3.3 SRT Mean Shift

We form our mean shift algorithm on $S^+(n)$ using $d_{\text{SRT}}()$ and $\mu_{\text{SRT}}(\mathcal{X})$ in steps 4 & 5 of Algorithm 1 respectively. It follows from the left-invariance of d_{SRT} that SRT mean shift is left-covariant.

The coefficients $\sigma_s, \sigma_t, \sigma_r$ act in place of the kernel bandwidth σ in Eq. (5). Also note that, while the coefficient ζ is constant in Euclidean space, it is *not* constant in a non-Euclidean space, in which case $\zeta = \zeta(\mathbf{X}_i)$ [30, 40] cannot be factored out of the kernel density estimate. Since $\zeta(\mathbf{X}_i)$ can be costly to compute (sometimes non-closed-form), existing mean shift algorithms on Lie groups [7, 40] replace $\zeta(\mathbf{X}_i)$ with a constant. However, in the case of $d_{\text{SRT}}()$, indeed any left-invariant distance, it can be shown that $\zeta(\mathbf{X}_i)$ is constant:

Lemma 1. *Using d_{SRT}, the volume densities are constant:* $\forall \mathbf{X}, \mathbf{Y} \in S^+(n) : \zeta(\mathbf{X}) = \zeta(\mathbf{Y})$.

Proof. The volume density function $\zeta(\mathbf{Y})$ with respect to the SRT distance and kernel $K(\cdot)$ at transformation $\mathbf{Y}$ is given by:

$$\zeta(\mathbf{Y}) = \int_{S^+(n)} K\left(d^2_{\text{SRT}}(\mathbf{U}, \mathbf{Y})\right) d\nu(\mathbf{U}), \tag{38}$$

where $\nu(\mathbf{U})$ is a (left-)Haar measure on $S^+(n)$ [30]. $\nu(\mathbf{U})$ has a property that $d\nu(\mathbf{U}) = d\nu(\mathbf{ZU})$ for all $\mathbf{Z} \in S^+(n)$. Let us fix $\mathbf{Z} = \mathbf{XY}^{-1}$. Since $d_{\text{SRT}}()$ is left-invariant, using the substitute $\mathbf{V} = \mathbf{ZU}$ and left-multiplying both input arguments of $d_{\text{SRT}}()$ with $\mathbf{Z}$, we obtain:

$$\begin{aligned} \zeta(\mathbf{Y}) &= \int_{S^+(n)} K\left(d^2_{\text{SRT}}(\mathbf{Z}^{-1}\mathbf{V}, \mathbf{Y})\right) d\nu(\mathbf{V}) \\ &= \int_{S^+(n)} K\left(d^2_{\text{SRT}}(\mathbf{V}, \mathbf{ZY})\right) d\nu(\mathbf{V}) = \zeta(\mathbf{X}). \end{aligned} \tag{39}$$

Therefore, $\zeta(\mathbf{X})$ is constant for all $\forall \mathbf{X} \in S^+(n)$. □

4 Experiments

4.1 Experimental Setup

Our experimental data consists of 12 shape classes, for which we have both a physical object and matching CAD model. We captured the geometry of each object using Vogiatzis and Hernández's multi-view stereo method [43] in the form of point clouds (Fig. 1(b)), 20 times from a variety of poses. Along with the class label, every shape instance has an associated ground truth pose, computed by first approximately registering the relevant CAD model to the point cloud manually, then using the Iterative Closest Point algorithm [5] to refine the registration.

4.1.1 Pose Vote Computation

Given a test point cloud and set of training point clouds (with known class and pose), the computation of input pose votes $\mathcal{X}$ is a two stage process similar to [20, 41]. In the first stage, local shape features, consisting of a descriptor and a scale, translation and rotation relative to the object, are computed on all the point clouds (Fig. 1(c)). In the second stage each test feature is matched to the m (we use 20) nearest training features, in terms of Euclidean distance between descriptors, to generate m pose votes.[7]

4.2 Inference

4.2.1 Mean Shift

Mean shift finds a local mode, and its weight, in the output pose distribution for a given object class. Since there may be many such modes we start mean shift from 100 random input poses for each class. Each mode, duplicates excepted, is then added to a list of candidate poses across all classes.

In $S^+(3)$ it is possible to use the quaternion representation of rotation, $\mathbf{q}(\mathbf{X})$, which we do. For efficiency, we therefore alternatively define the rotation component of $d_{\mathrm{SRT}}()$ as

$$d_r(\mathbf{X},\mathbf{Y}) = \sqrt{1 - |\mathbf{q}(\mathbf{X})^{\mathsf{T}}\mathbf{q}(\mathbf{Y})|}, \tag{40}$$

where $|\cdot|$ is needed to account for the fact that $\mathbf{q}(\mathbf{X})$ and $-\mathbf{q}(\mathbf{X})$ represent the same rotation. This formulation confers a small computational advantage over other, non-component-wise distances in this space.

[7] Since all inference methods will use the same set of input pose votes, the method by which these are computed is not central to the evaluation of relative performance.

4.2.2 Hough Voting

We implemented a published Hough voting scheme [20] to compare with the mean shift inference approaches. This computes sums of weights of the pose votes which fall into each bin of a 4D histogram over translation and scale, effectively marginalizing over rotation. The highest bin sum for each class defines a pose mode. Note that we used our own pose votes and weights, and not those computed using the method described in [20].

4.3 Evaluation

We use cross validation on our training data for evaluation—a training set is created from 19 of the 20 shape instances in each class, and the remaining instance in each class becomes a test shape. Each test shape undergoes 5 random transformations (over translation, rotation and scale in the range 0.5–2), and this process is repeated with each training shape being the test shape, creating 100 test instances per class. We use 10 classes in our evaluation (shown in Fig. 5), so 1000 tests in all. The remaining 2 classes are used to learn the optimal kernel bandwidth, σ, for each inference method. We have made the data used in this evaluation publicly available [1].

We evaluate each inference method on two criteria: Recognition rate and registration rate.

4.3.1 Recognition Rate

As described above, each inference method generates a list of modes across pose and class for a given test instance, each with an associated weight. The output class is that of the mode of highest weight. A confusion matrix logs the output class versus ground truth class across all tests. The recognition rate is given by the trace of this matrix, *i.e.*, the number of correct classifications.

4.3.2 Registration Rate

The output pose for a given test instance is given by that of the weightiest mode whose class matches the ground truth class. We choose to consider a pose $\mathbf{X}$ to be correct if its scale is within 5%, orientation is within 15° and translation is within 10% (of the object size) of the ground truth's. Explicitly, the criteria to be met are

$$\left|\log\left(\frac{s(\mathbf{X})}{s(\mathbf{Y})}\right)\right| < 0.05, \tag{41}$$

$$\operatorname{acos}\left(\frac{\operatorname{trace}(\mathbf{R}(\mathbf{X})^{-1}\mathbf{R}(\mathbf{Y})) - 1}{2}\right) < \pi/12, \tag{42}$$

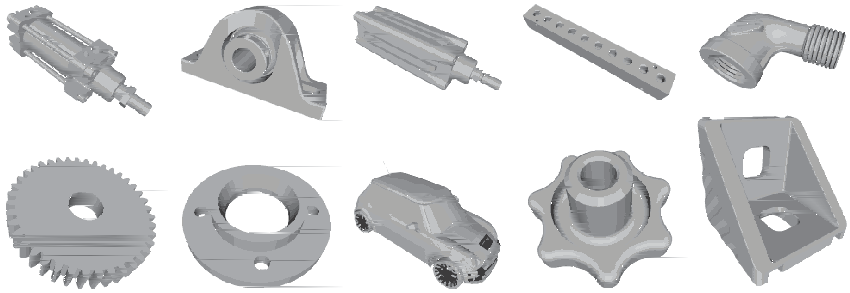


Fig. 5 Test objects. CAD models of the 10 real objects used for evaluation. *Top*: piston2, bearing, piston1, block, and pipe. *Bottom*: cog, flange, car, knob, and bracket.

$$\frac{\|\mathbf{t}(\mathbf{X}) - \mathbf{t}(\mathbf{Y})\|}{\sqrt{s(\mathbf{X})s(\mathbf{Y})}} < 0.1, \tag{43}$$

with $\mathbf{Y}$ being the ground truth pose. In the case of an object having symmetries there are multiple $\mathbf{Y}$'s, and distance to the closest is used.

4.3.3 Learning σ

We learn the mean shift kernel bandwidth, σ (or in the case of SRT, σ_s, σ_r and σ_t), used for each mean shift algorithm by maximizing the registration rate from cross-validation on two training classes (which are not used in the final evaluation). Registration rate is maximized using local search: an initial bandwidth is chosen, then the registration rate computed for this value and the values 1.2 and $1/1.2$ times this value. That value with the highest score is chosen, and the process is repeated until convergence. With 3 parameters to learn, the local search is computed over a 3D grid.

4.4 Results

Table 3 summarizes the quantitative results for the four inference methods tested. It shows that SRT mean shift performs best at both recognition and registration. The third row gives registration rate taking into account scale and translation only (as the Hough method only provides these), indicating that mean shift performs considerably better than Hough voting at registration. Also given (row 5) is the mean of output scales (each as a ratio of the output scale over the ground truth scale) of the registration result, which shows a marked bias towards a smaller scale when using extrinsic mean shift. Whilst better than extrinsic mean shift at registration, intrinsic mean shift is the slowest[8] method by an order of magnitude.

[8] We used optimized implementations for all methods.

Table 3 Quantitative results for the four inference methods tested. The SRT mean shift method is best in all respects except speed, for which it is better than the other mean shift methods.

	SRT	Extrinsic	Intrinsic	Hough
Recognition	**64.9%**	49.6%	45.5%	56.1%
Registration	**68.3%**	52.0%	62.0%	–
Registration ($\mathbf{t}$,s)	**79.8%**	62.0%	75.7%	57.3%[9]
Processing time	1.6s	9.7s	127s	**0.043s**
Mean scale	**0.995**	0.959	0.987	–

Table 4 Registration rate per class (%). SRT mean shift performs best on 7/10 classes.

	bearing	block	bracket	car	cog	flange	knob	pipe	piston1	piston2
SRT	**77**	13	**95**	75	**100**	**41**	**88**	**86**	**44**	63
Extrinsic	36	12	90	50	80	32	53	63	37	**67**
Intrinsic	54	**19**	83	**90**	90	36	65	82	34	**67**

The per-class registration rates of the mean shift methods are given in Table 4, showing that SRT out-performs extrinsic mean shift in 9 out of 10 classes, and intrinsic mean shift in 7 out of 10. The scale-invariance of registration rate, and hence, by implication, recognition rate, using SRT and intrinsic mean shift, and the contrasting scale-variance of extrinsic mean shift (as discussed in Sect. 2.4.1), is shown empirically in Fig. 6.

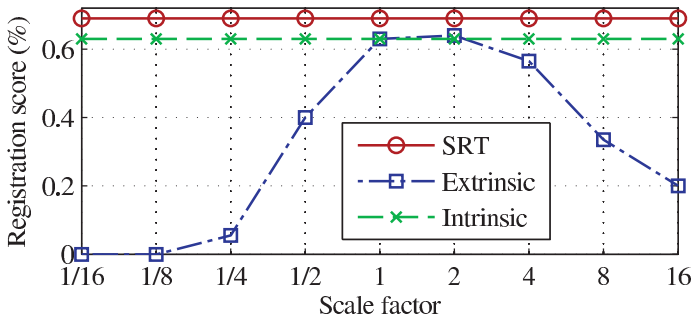
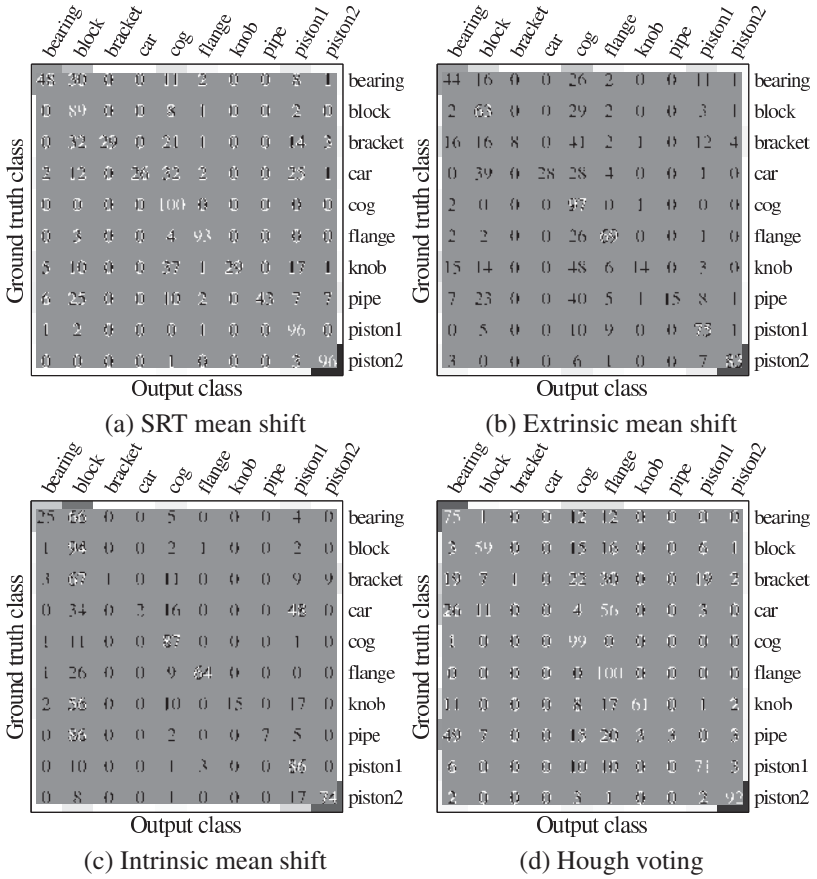


Fig. 6 Scale-invariance. Registration rate over scale, showing that only extrinsic mean shift varies with scale.

[9] This score is the percentage of ground truth poses that were in the same bin as the output pose.



(a) SRT mean shift (b) Extrinsic mean shift

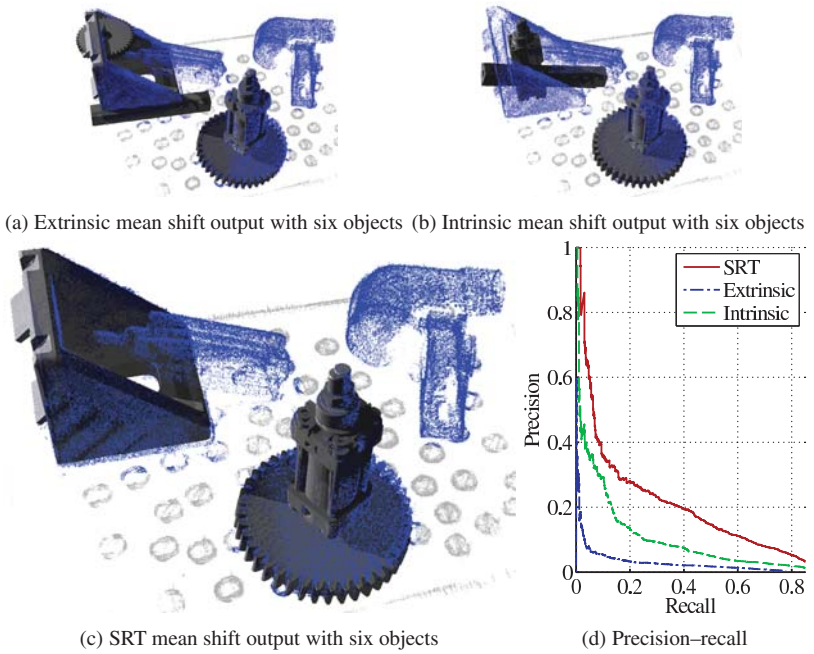
(c) Intrinsic mean shift (d) Hough voting

Fig. 7 Confusion matrices for the four inference methods tested. The Hough voting method performs poorly on objects with low rotational symmetry, while mean shift methods, and in particular SRT, perform better.

The confusion matrices for the four inference methods are shown in Fig. 7. Hough voting performs very poorly on bracket, car and pipe, getting a recognition rate of just 1.3% on average for these classes, which all have low rotational symmetry; in particular it prefers cog and flange (which both have high rotational symmetry), no doubt due to the marginalization this method performs over rotation. Intrinsic mean shift shows a tendency to confuse block, and cog and piston1 to a lesser degree, for other classes, whilst extrinsic and SRT mean shift confuse cog, and block and piston1 to a lesser degree for other classes.

Finally, Fig. 8a–c demonstrates that SRT mean shift applied to a real scene containing multiple objects yield more accurate results than extrinsic mean shift and intrinsic mean shift. Given a threshold weight above which modes are accepted, mean shift on the votes can produce many false positive detections, as shown by the low precision at high recall rates in Fig. 8d. This issue is addressed in another

work [44]. Our system can additionally (though not used here) filter the list of output poses using physical constraints such as the position of the ground plane and collision detection, which we found removed the majority of false positive results, including those shown in Fig. 8a–c.



(a) Extrinsic mean shift output with six objects (b) Intrinsic mean shift output with six objects

(c) SRT mean shift output with six objects (d) Precision–recall

Fig. 8 Performance with multiple objects. Given a point cloud with 6 objects, (a) Extrinsic mean shift finds 3 of them with 2 false alarms, (b) Intrinsic mean shift finds 2 of them with 2 false alarms, (c) SRT mean shift find 3 of them with no false alarms. (d) Precision-recall curves of the mean shift methods for correct registration and recognition jointly.

5 Conclusion

We have introduced the SRT distance for use in mean shift on poses in the space of direct similarity transformations, $S^+(n)$. We have proven the distance to be left-invariant, and have a unique, closed-form mean with the desirable properties of scale, rotation and translation compatibilities. We have demonstrated the use of this distance for registration and recognition tasks on a challenging and realistic 3D dataset which combines real-world objects, with and without rotational symmetries, together with a vision-based geometry capture system and basic features.

Our results show that SRT mean shift has better recognition and registration rates than both intrinsic and extrinsic mean shift, as well as Hough voting. We also show that extrinsic mean shift not only is scale-variant but also biases output scale, and

that intrinsic mean shift is slower to compute. In addition to the performance increase over Hough voting, especially in the presence of rotationally symmetric objects, we demonstrate for the first time that mean shift on the full 7D pose space of $S^+(3)$ is not only possible, but that it also provides accurate 7D registration, including rotation. This is not practical using Hough-based approaches, due to their exponential memory requirements.

Potential future research includes creating efficient probability density functions on $S^+(n)$, which will serve as building blocks for statistical learning and inference on this non-Euclidean space.

References

1. Toshiba CAD model point clouds dataset
2. Agrawal, M.: A Lie algebraic approach for consistent pose registration for general euclidean motion. In: Proc. Int. Conf. on Intelligent Robot and Systems, pp. 1891–1897 (2006)
3. Arsigny, V., Commowick, O., Pennec, X., Ayache, N.: A Log-Euclidean Polyaffine Framework for Locally Rigid or Affine Registration. In: Pluim, J.P.W., Likar, B., Gerritsen, F.A. (eds.) WBIR 2006. LNCS, vol. 4057, pp. 120–127. Springer, Heidelberg (2006)
4. Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition 13(2), 111–122 (1981)
5. Besl, P., McKay, N.: A method for registration of 3D shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(2) (1992)
6. Campbell, R.J., Flynn, P.J.: A survey of free-form object representation and recognition techniques. Computer Vision and Image Understanding 81, 166–210 (2001)
7. Cetingul, H.E., Vidal, R.: Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1896–1902 (2009)
8. Chen, H., Bhanu, B.: 3d free-form object recognition in range images using local surface patches. J. Pattern Recognition Letters 28, 1252–1262 (2007)
9. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 17, 790–799 (1995)
10. Davies, P.I., Higham, N.J.: A Schur-Parlett algorithm for computing matrix functions. SIAM J. Matrix Anal. Appl. 25, 464–485 (2003)
11. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3D object recognition. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 998–1005 (2010)
12. Eggert, D.W., Lorusso, A., Fisher, R.B.: Estimating 3-d rigid body transformations: a comparison of four major algorithms. Machine Vision Application 9, 272–290 (1997)
13. Ashbrook, A.P., Fisher, R.B., Robertson, C., Werghi, N.: Finding Surface Correspondence for Object Recognition and Registration Using Pairwise Geometric Histograms. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, p. 674. Springer, Heidelberg (1998)
14. Fréchet, M.: Les éléments aléatoires de nature quelconque dans un espace distancié. Ann. Inst. H. Poincaré 10, 215–310 (1948)
15. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing Objects in Range Data Using Regional Point Descriptors. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 224–237. Springer, Heidelberg (2004)
16. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1022–1029 (June 2009)

17. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(5), 433–449 (1999)
18. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. In: Proc. Eurographics/ACM SIGGRAPH Symp. on Geometry Processing, pp. 156–164 (2003)
19. Khoshelham, K.: Extending generalized Hough transform to detect 3D objects in laser range data. In: Workshop on Laser Scanning, vol. XXXVI, pp. 206–210 (2007)
20. Knopp, J., Prasad, M., Willems, G., Timofte, R., Van Gool, L.: Hough Transform and 3D SURF for Robust Three Dimensional Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 589–602. Springer, Heidelberg (2010)
21. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. Int. J. Computer Vision 77(1-3), 259–289 (2008)
22. Mamic, G., Bennamoun, M.: Representation and recognition of 3d free-form objects. Digital Signal Processing 12(1), 47–76 (2002)
23. Mian, A.S., Bennamoun, M., Owens, R.A.: Automatic correspondence for 3D modeling: an extensive review. Int. J. Shape Modeling 11(2), 253–291 (2005)
24. Mian, A.S., Bennamoun, M., Owens, R.: Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(10), 1584–1601 (2006)
25. Moakher, M.: Means and averaging in the group of rotations. SIAM J. Matrix Anal. Appl. 24, 1–16 (2002)
26. Mundy, J.L.: Object Recognition in the Geometric Era: A Retrospective. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) Toward Category-Level Object Recognition. LNCS, vol. 4170, pp. 3–28. Springer, Heidelberg (2006)
27. Okada, R.: Discriminative generalized hough transform for object dectection. In: Proc. Int. Conf. on Computer Vision, pp. 2000–2005 (October 2009)
28. Opelt, A., Pinz, A., Zisserman, A.: Learning an alphabet of shape and appearance for multi-class object detection. Int. J. Computer Vision 80(1) (2008)
29. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Trans. Graph. 21, 807–832 (2002)
30. Pelletier, B.: Kernel density estimation on Riemannian manifolds. Statistics Probability Letters 73(3), 297–304 (2005)
31. Pennec, X.: Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. JMIV 25(1), 127–154 (2006)
32. Pennec, X., Ayache, N.: Uniform distribution, distance and expectation problems for geometric features processing. J. Math. Imaging Vis. 9, 49–67 (1998)
33. Petrelli, A., Di Stefano, L.: On the repreatability of the local reference frame for partial shape matching. In: Proc. Int. Conf. on Computer Vision (2011)
34. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: Proc. Int. Conf. Robotics and Automation, pp. 3212–3217 (2009)
35. Saupe, D., Vranic, D.V.: 3D Model Retrieval with Spherical Harmonics and Moments. In: Radig, B., Florczyk, S. (eds.) DAGM 2001. LNCS, vol. 2191, p. 392. Springer, Heidelberg (2001)
36. Schramm, É., Schreck, P.: Solving geometric constraints invariant modulo the similarity group. In: Int. Conf. on Computational Science and Applications, pp. 356–365 (2003)
37. Shotton, J.D.J., Blake, A., Cipolla, R.: Multiscale categorical object recognition using contour fragments. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(7), 1270–1281 (2008)
38. Srivastava, A., Klassen, E.: Monte Carlo extrinsic estimators of manifold-valued parameters. IEEE Trans. on Signal Processing 50(2), 299–308 (2002)
39. Subbarao, R., Meer, P.: Nonlinear mean shift for clustering over analytic manifolds. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, vol. I, pp. 1168–1175 (2006)

40. Subbarao, R., Meer, P.: Nonlinear mean shift over Riemannian manifolds. Int. J. Computer Vision 84(1) (2009)
41. Tombari, F., Di Stefano, L.: Object recognition in 3D scenes with occlusions and clutter by Hough voting. In: Proc. Pacifc-Rim Symp. on Image and Video Technology, pp. 349–355 (2010)
42. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Proc. European Conf. on Computer Vision (2010)
43. Vogiatzis, G., Hernández, C.: Video-based, real-time multi view stereo. Image and Vision Computing 29(7), 434–441 (2011)
44. Woodford, O.J., Pham, M.-T., Maki, A., Perbet, F., Stenger, B.: Demisting the Hough transform for 3D shape recognition and registration. In: British Machine Vision Conference (2011)
45. Roger, P.: Woods. Characterizing volume and surface deformations in an atlas framework: theory, applications, and implementation. NeuroImage, 18(3):769–788 (2003)