Full-Angle Quaternions for Robustly Matching Vectors of 3D Rotations

Stephan Liwicki[1,*] Minh-Tri Pham[2,*] Stefanos Zafeiriou[1] Maja Pantic[1,3] Björn Stenger[2]

[1]Computer Science, Imperial College London, United Kingdom
[2]Cambridge Research Laboratory, Toshiba Research Europe Ltd, Cambridge, United Kingdom
[3]Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands

Abstract

In this paper we introduce a new distance for robustly matching vectors of 3D rotations. A special representation of 3D rotations, which we coin full-angle quaternion (FAQ)*, allows us to express this distance as Euclidean. We apply the distance to the problems of 3D shape recognition from point clouds and 2D object tracking in color video. For the former, we introduce a hashing scheme for scale and translation which outperforms the previous state-of-the-art approach on a public dataset. For the latter, we incorporate online subspace learning with the proposed FAQ representation to highlight the benefits of the new representation.*

1. Introduction

Outliers and noisy data are common problems when matching feature vectors in applications such as image registration [29], image matching [6], shape matching [8], face recognition [20], object tracking [18], and feature learning [23]. Standard distances (*e.g*. the Euclidean distance between Euclidean points) can be disadvantageous since corruptions may bias the results, *e.g*. [4]. Because identifying outliers may be computationally costly and sometimes difficult, robust distances between vectors that suppress the influence of outliers while preserving the inliers' geometry have been developed [6, 8, 18, 20].

Most commonly, feature vectors are scalar valued. In this case existing methods tackle the problem of outliers mainly by adopting different distances. Early approaches use variants of the Manhattan distance [5, 13], leading to an increase in robustness, but at the cost of reduced efficiency. Recent works [4, 18, 23] achieve both robustness and efficiency by mapping points non-linearly to a space where the distance involving outliers is nearly uniformly distributed, thereby allowing for robust subspace learning.

*Main contributors: sl609@imperial.ac.uk, mtpham@crl.toshiba.co.uk

Non-scalar features have received less attention in the literature. For example, in [8], the matching of unordered sets of 2D points is considered. In [29], a robust comparison of 2D rotations adopting the sum of cosines of angle differences is investigated. The work in [20] extends this approach to match vectors of 3D surface normals, by projecting the 3D normals to 2D rotations.

In this paper we address the problem of matching feature vectors of *3D rotations*, introducing a robust and efficient distance function. We apply our approach to 3D shape recognition by converting the problem of evaluating shape poses into the problem of robustly matching vectors of direct similarities (*i.e*. transformations with a uniform scale, rotation, and translation [3]). In particular, we introduce concurrent hashing of scale and translation to produce vectors of rotations, which we then evaluate using our distance. We also apply the new FAQ representation to 2D object tracking, where we formulate the problem of robustly matching color patches as an online subspace learning task for vectors of rotations. Our contributions are as follows:

1. We propose a closed-form distance between 3D rotations, which allows for robust matching and subspace learning with vectors of 3D rotations.
2. We formulate a new 3D rotation representation, called *full-angle quaternion*, making our distance Euclidean.
3. We introduce a map such that uniformly distributed dilatations (*i.e*. transformations composed of a uniform scaling and a translation [3]) correspond to uniformly distributed coordinates in Euclidean space, facilitating efficient hashing.
4. We evaluate our framework on 3D shape recognition and 2D tracking, showing superior performance in comparison to existing methods.

2. Existing Rotation Distances

We first briefly review existing distances for 2D and 3D rotations in literature.

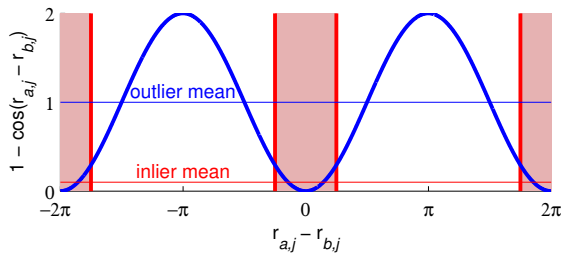


Figure 1: *Cosine-based distance function for different angle differences. Inliers reside in the highlighted areas.*

2.1. Robust Rotation Distance in 2D

In [29], the robust cosine-based distance between gradient orientations is introduced to match vectors of 2D rotations for face recognition. Consider an image $\boldsymbol{I}_i$ with N pixels, the direction of the intensity gradient at each pixel is recorded as rotation angle $\mathtt{r}_{i,j}$, $j = 1, \dots, N$, *i.e.* the j^{th} angle value of the i^{th} image. The squared distance between two images, $\boldsymbol{I}_a$ and $\boldsymbol{I}_b$, is provided by:

$$d(\mathtt{r}_a, \mathtt{r}_b)^2 := 1 - \sum_{j=1}^{N} \frac{\cos(\mathtt{r}_{a,j} - \mathtt{r}_{b,j})}{N}, \qquad (1)$$

based on the robust correlation in [6]. We visualize the distance function and its robust properties in fig. 1. Its advantages stem from the sum of cosines. In particular, for an uncorrelated area $\mathcal{P}$, with random angle directions, the distance values are almost uniformly distributed: $\sum_{j \in \mathcal{P}} \cos(\mathtt{r}_{a,j} - \mathtt{r}_{b,j}) \approx 0$, and the distance tends to 1. For highly correlated vectors of rotations, the distance is near 0. Thus, while inliers achieve the ideal distance of 0, outliers have less effect and only shift the distance towards the mean at 1 (and not the maximum value of 2).

Beside 2D face recognition, the cosine-based distance has been employed in 2D tracking [19] and 3D face recognition [20]. In [19], gradient orientations in intensity images are extracted and matched robustly. The work in [20] takes a very similar approach. First, by using the image and its depth map, the camera facing 3D surface normals of an object are computed. Then, the distance is applied to robustly match vectors of the surface normals' azimuth angles.

2.2. Rotation Distances in 3D

Numerous distances for 3D rotations have been proposed in the literature [7, 10, 11, 22]. Most of these are Euclidean distances (and variants) under different representations of 3D rotations. The Euler angles distance is the Euclidean distance between Euler angles [11]. The ℓ_2-norm under the half-angle quaternion (HAQ) representation leads to the vectorial/extrinsic quaternion distance [7, 10, 11] and the inverse cosine quaternion distance [11]. Analysis of geodesics on $\mathcal{SO}(3)$ leads to intrinsic distances [10, 11, 22], which are the ℓ_2-norm of rotation vectors (RV). The ℓ_2-norm in the embedding space $\mathbb{R}^9$ of $\mathcal{SO}(3)$ induces the chordal/extrinsic distance [7, 11, 22] between rotation matrices (RM).

In computer vision arguably the most widely used 3D rotation distances are the extrinsic distances based on HAQ or RM, due to their closed forms [7, 22], allowing to efficiently compute rotation means. However, the HAQ representation is not unique, leading to non-global means. The RM representation is unfavorable for general optimization problems due to the extra constraints it imposes – the RV representation is more suitable for such tasks.

3. Robust 3D Rotation Distance

Inspired by the cosine-based 2D rotation distance in (1), we formulate a distance for 3D rotations. This is non-trivial, as the concept of rotation axes is non-existent in 2D.

3.1. Proposed Distance

In 2D, $\mathtt{r}_{i,j}$ is solely defined by an angle $\alpha_{i,j}$. In 3D, let us assume our rotations are given as an angle-axis pair $\mathtt{r}_{i,j} = (\alpha_{i,j}, \boldsymbol{v}_{i,j}) \in \mathcal{SO}(3)$. We propose the following distance function for comparing vectors of 3D rotations:

$$d(\mathtt{r}_a, \mathtt{r}_b)^2 := 1 - \sum_{j=1}^{N} \left(\frac{1 + \boldsymbol{v}_{a,j}^{\mathrm{T}} \boldsymbol{v}_{b,j}}{2} \right) \frac{\cos(\alpha_{a,j} - \alpha_{b,j})}{N} - \sum_{j=1}^{N} \left(\frac{1 - \boldsymbol{v}_{a,j}^{\mathrm{T}} \boldsymbol{v}_{b,j}}{2} \right) \frac{\cos(\alpha_{a,j} + \alpha_{b,j})}{N}. \qquad (2)$$

Note that $\frac{1+\boldsymbol{v}_{a,j}^{\mathrm{T}} \boldsymbol{v}_{b,j}}{2} + \frac{1-\boldsymbol{v}_{a,j}^{\mathrm{T}} \boldsymbol{v}_{b,j}}{2} = 1$, *i.e.* the terms act as weights that depend on the angle between the rotations' unit axes. Fig. 2 visualizes the weights' properties. Let us consider two rotations, $\mathtt{r}_{a,j}$ and $\mathtt{r}_{b,j}$. If both share the same axis $\boldsymbol{v}_{a,j} = \boldsymbol{v}_{b,j}$, then $\boldsymbol{v}_{a,j}^{\mathrm{T}} \boldsymbol{v}_{b,j} = 1$ and our distance turns into its 2D counterpart in (1). In the case of opposing axes, $\boldsymbol{v}_{a,j} = -\boldsymbol{v}_{b,j}$, $\boldsymbol{v}_{a,j}^{\mathrm{T}} \boldsymbol{v}_{b,j} = -1$ and the sign of $\alpha_{b,j}$ is flipped. Notice that $(\alpha_{b,j}, \boldsymbol{v}_{b,j}) = (-\alpha_{b,j}, \boldsymbol{v}_{a,j})$. Hence, again the problem is reduced to (1). A combination of both parts is employed in the case of $-1 < \boldsymbol{v}_{a,j}^{\mathrm{T}} \boldsymbol{v}_{b,j} < 1$.

We compare our 3D cosine distance to the squared Euclidean distance with different 3D rotation representations: HAQ, RM and RV (fig. 3). When similar rotations are compared (fig. 3(a)), the RV representation is sensitive to rotations with angles close to 180°, here the normalized distance may jump from near 0 to near 1. All other methods identify close rotations successfully. When comparing random rotations (fig. 3(b)), RM and RV strongly bias the results either towards small or large distances. The values under HAQ and our distance are more evenly distributed. The proposed distance shows similar properties to the distance under RM when applied to rotations with similar rotation axes (fig. 3(c)). Here HAQ produces overall smaller

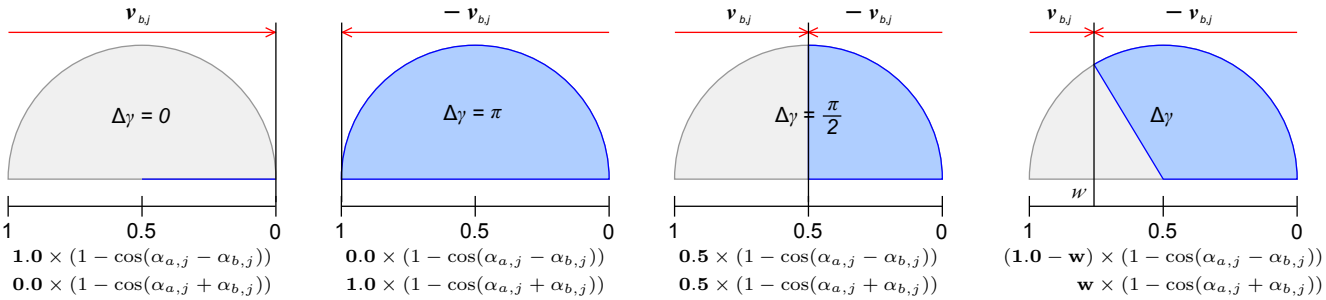
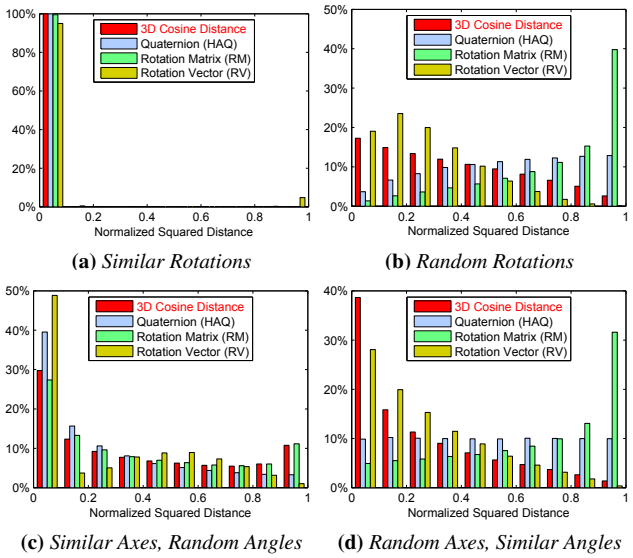


Figure 2: *The proposed cosine-based distance for 3D rotations, with varying angles between rotation axes (denoted by $\Delta\gamma$). Our distance is expressed as a weighted sum of the 2D cosine-based distance, and its counterpart with flipped angle $\alpha_{b,j}$.*



(a) *Similar Rotations* **(b)** *Random Rotations*

(c) *Similar Axes, Random Angles* **(d)** *Random Axes, Similar Angles*

Figure 3: *Histograms of squared distances $d(\mathtt{r}_{a,j}, \mathtt{r}_{b,j})^2$ for (a) similar and (b) uniformly distributed random rotations, rotations with (c) similar axes and (d) similar angles, each resulting from 10^5 3D rotation comparisons. Random rotations are taken from the left Haar measure on $\mathcal{SO}(3)$ [9, 24].*

distances. The distance under RV is quite unstable for this set-up, as no trend is observed. However, when exposed to similar rotation angles (fig. 3(d)), it behaves similarly to our proposed distance. RM shows a bias towards large distances, while HAQ exhibits an even distribution of distances.

Our distance and HAQ are least biased for random samples: The maximum count of a single bin is less than 20%, and the mean of outliers, *i.e.* random samples, is near 0.5. The distance for inliers is close to 0. This corresponds to the robust properties of the cosine distance in 2D [29]. Our method differs mainly in its preference when compared to HAQ; both favor similar axes, but HAQ does so more significantly. Our proposed distance rates the strength of rotations more highly, *i.e.* the rotation angle or the amount of displacement when applied to a scene.

3.2. Full-Angle Quaternion

The proposed distance in (2) leads to a new representation for 3D rotations, which allows for efficient comparison. We coin it the *full-angle quaternion* (FAQ) representation.

Let us rewrite the squared distance as follows:

$$
\begin{aligned}
d(\mathtt{r}_a, \mathtt{r}_b)^2 &= 1 - \sum_{j=1}^{N} \frac{\cos\alpha_{a,j}\cos\alpha_{b,j}}{N} \\
&\quad - \sum_{j=1}^{N} \frac{(\boldsymbol{v}_{a,j}^{\mathrm{T}}\boldsymbol{v}_{b,j})\sin\alpha_{a,j}\sin\alpha_{b,j}}{N} \qquad (3) \\
&= \sum_{j=1}^{N} \frac{(\cos\alpha_{a,j} - \cos\alpha_{b,j})^2}{2N} \\
&\quad + \sum_{j=1}^{N} \frac{\|\boldsymbol{v}_{a,j}\sin\alpha_{a,j} - \boldsymbol{v}_{b,j}\sin\alpha_{b,j}\|^2}{2N} \qquad (4) \\
&= \frac{1}{2N}\sum_{j=1}^{N} \|\mathtt{q}_{a,j} - \mathtt{q}_{b,j}\|^2, \qquad (5)
\end{aligned}
$$

where $\mathtt{q}_{i,j}$ is a unit quaternion given by:

$$\mathtt{q}_{i,j} := \cos\alpha_{i,j} + (\mathbf{i}\boldsymbol{v}_{i,j,1} + \mathbf{j}\boldsymbol{v}_{i,j,2} + \mathbf{k}\boldsymbol{v}_{i,j,3})\sin\alpha_{i,j}. \quad (6)$$

Eq. (6) defines our FAQ representation. The key difference to the HAQ representation is that the trigonometric functions $\cos(\cdot)$ and $\sin(\cdot)$ are applied to the full angle $\alpha_{i,j}$ instead of the half angle $\frac{\alpha_{i,j}}{2}$. Hence, it avoids the double covering issue of HAQ, as each 3D rotation corresponds to exactly one unit quaternion under FAQ. In addition, (5) reveals that the proposed distance is *equivalent to the Euclidean distance under our FAQ representation*.

In contrast to HAQ, which returns non-global means [7], the mean of 3D rotations under FAQ is *global* and easy to compute as the normalized sum of FAQs.

The FAQ representation comes with a degenerate case: Every 3D rotation by $180°$ maps to the same unit quaternion, $q = -1$. This, however, does not effect the computation of our distance nor its proposed applications as we do not require an inverse mapping.

4. Matching Ball Features for 3D Recognition

In this section, we apply our distance to the task of 3D recognition from unstructured point clouds. Local features are commonly used for this task [2, 14, 25, 28, 31]. In particular, features in the form of 3D oriented balls together with their descriptions are localized in the input scene using standard multi-scale keypoint detectors like SURF-3D [14] or ISS [32]. At test time, the extracted scene features are matched with features from training data by comparing their descriptions [2, 14, 28], generating an initial set of votes. These votes are hypotheses of the object's identity and pose, consisting of a position and an orientation [14, 28], and additionally a scale if the scale is unknown [25, 31]. The best vote is then selected as an estimate of pose and identity.

Real objects often have repetitive parts which map to the same descriptor, thus generating many votes. To select the best vote, existing methods either (1) group votes according to object identity and find the pose with most neighbors in each group [14, 25, 28, 31], or (2) evaluate the votes by aligning the predicted objects' surface to the scene [2, 12, 21]. While the former approaches are fast, they often fail if the ground truth pose has too few neighbors. The latter frameworks are more robust, but finding the corresponding scene surface for each predicted object surface is computationally costly.

We propose an alternative method for evaluating votes. Instead of aligning surfaces between the scene and the predicted objects, we align *feature locations*. If a predicted object pose is close to its true pose, the scene feature locations and orientations match those of the predicted object's training data when back-projected into the scene.

For notational convenience, we denote by $\mathbf{X}_s$, $\mathbf{X}_\mathbf{R}$ and $\mathbf{X}_\mathbf{t}$ the scale, rotation and translation part respectively of a direct similarity $\mathbf{X}$. Our method consists of two phases:

In the offline phase (alg. 1), for each object we collect all feature locations that occur in the training data, and normalize these *via* left-multiplication with their corresponding object pose's inverse. In alg. 1, $\mathfrak{F}$ and $\mathcal{C}$ are multi-index lists such that $\mathfrak{F}_{i,j,k}$ denotes the i^{th} object's j^{th} training instance's k^{th} feature location, and $\mathcal{C}_{i,j}$ denotes the the i^{th} object's j^{th} training instance's pose. All normalized locations of object i are then stored in a single hash table $\mathcal{H}_i$ in which hash keys are computed based on the scale and translation components. The design of function $\mathbf{h}(\cdot)$ is detailed in §4.1. The value of a hash entry is the set of rotations of all normalized locations hashed to it.

In the online phase (alg. 2), we first restrict the search space to the 3D ball features observed in the scene and produce a vector of 3D rotations. For each vote, we then left-multiply all the scene feature locations, denoted by $\boldsymbol{S}$, with the inverse of the vote's predicted pose. Finally, we look into the hash entry (*via* $\mathcal{H}_i$ - the hash table of the predicted object) of each transformed feature location and find the nearest rotation. Thus, we compare the vector of scene features, in particular their rotations, to the training data. Note that our method does not involve any feature descriptions, as only pose is required. Therefore, it exploits the geometry of an object as a whole, not the geometry of local features. Section §4.2 raises further points regarding the comparison.

Algorithm 1 Offline phase: creating hash tables

Input: training feature locations $\mathfrak{F}$ and poses $\mathcal{C}$

1: **for all** object i:
2: Create hash table $\mathcal{H}_i$.
3: **for all** training instance j of the object:
4: **for all** feature k of the training instance:
5: $\mathbf{X} \leftarrow \mathcal{C}_{i,j}^{-1}\mathfrak{F}_{i,j,k}$.
6: Find/insert hash entry $\boldsymbol{V} \leftarrow \mathcal{H}_i(\mathbf{h}(\mathbf{X}))$.
7: $\boldsymbol{V} \leftarrow \boldsymbol{V} \cup \{\mathbf{X}_\mathbf{R}\}$.
8: Return $\mathcal{H}$.

Algorithm 2 Online phase: vote evaluation

Parameters: hash tables $\mathcal{H}$ and scene feature locations $\boldsymbol{S}$
Input: vote = (object identity i, pose $\mathbf{Y}$)

1: $w \leftarrow 0$.
2: **for all** scene feature j:
3: $\mathbf{X} \leftarrow \mathbf{Y}^{-1}\boldsymbol{S}_j$.
4: Find hash entry $\boldsymbol{V} \leftarrow \mathcal{H}_i(\mathbf{h}(\mathbf{X}))$.
5: **if** found:
6: $w \leftarrow w + 4 - \min_{\mathbf{R}\in\boldsymbol{V}} d(\mathbf{R}, \mathbf{X}_\mathbf{R})^2$.
7: Return w.

4.1. Hashing Dilatations

The hash keys are computed as follows. The scale and translation parts of a direct similarity form a transformation called (direct) dilatation [3] in the space

$$\mathcal{DT}(3) := \left\{ \begin{bmatrix} s\mathbf{I} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, s \in \mathbb{R}_+, \mathbf{t} \in \mathbb{R}^3 \right\}, \quad (7)$$

where $\mathbf{I}$ is the 3×3 identity matrix. Given a direct similarity $\mathbf{X}$, we first map its dilatation part, $\mathbf{X}_\mathbf{D} := \begin{bmatrix} \mathbf{X}_s\mathbf{I} & \mathbf{X}_\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$, to a 4D point *via* $\phi : \mathcal{DT}(3) \to \mathbb{R}^4$:

$$\phi(\mathbf{X}_\mathbf{D}) := (\ln \mathbf{X}_s, \mathbf{X}_\mathbf{t}^\mathrm{T}/\mathbf{X}_s)^\mathrm{T}. \quad (8)$$

We quantize the 4D point to a 4D integer vector, *i.e.* a hash key, via a quantizer $\eta : \mathbb{R}^4 \to \mathbb{Z}^4$:

$$\eta(\mathbf{x}) := \left(\left\lfloor \frac{\mathbf{x}_1}{\sigma_s} \right\rfloor, \left\lfloor \frac{\mathbf{x}_2}{\sigma_t} \right\rfloor, \left\lfloor \frac{\mathbf{x}_3}{\sigma_t} \right\rfloor, \left\lfloor \frac{\mathbf{x}_4}{\sigma_t} \right\rfloor \right)^\mathrm{T}, \quad (9)$$

where σ_s and σ_t are parameters that enable making trade-offs between scale and translation, and operator $\lfloor\cdot\rfloor$ finds the integer value of a real number. Thus, the function $\mathbf{h}(\cdot)$ in alg. 1 and alg. 2 is defined as $\mathbf{h}(\mathbf{X}) := \eta \circ \phi(\mathbf{X}_\mathbf{D})$.

An efficient hash table should ensure that every hash entry is accessed with similar probability so that collisions are minimized. To achieve this, we have designed $\phi(\cdot)$ so that the following lemma holds.

Lemma 1. *The Euclidean volume element of $\mathbb{R}^4$ is pulled back via $\phi(\cdot)$ to a left-invariant 4-form on $\mathcal{DT}(3)$.*

Proof. Denote by $D(\mathbf{x}) := d\mathbf{x}_1 d\mathbf{x}_2 d\mathbf{x}_3 d\mathbf{x}_4$ the Euclidean volume element at $\mathbf{X} := \phi^{-1}(\mathbf{x})$. To prove the lemma, it is sufficient to show that for all $\mathbf{Y} \in \mathcal{DT}(3)$ and $\mathbf{x} \in \mathbb{R}^4$:

$$D(\mathbf{x}) = D(\phi(\mathbf{Y}\phi^{-1}(\mathbf{x}))). \tag{10}$$

Let $\mathbf{y} := \phi(\mathbf{Y})$. Substituting (8) to (10) yields:

$$\phi(\mathbf{Y}\phi^{-1}(\mathbf{x})) \tag{11}$$

$$= \phi\left(\begin{bmatrix} e^{\mathbf{y}_1+\mathbf{x}_1} & e^{\mathbf{y}_1+\mathbf{x}_1}\mathbf{x}_{2:4} + e^{\mathbf{y}_1}\mathbf{y}_{2:4} \\ \mathbf{0} & 1 \end{bmatrix}\right) \tag{12}$$

$$= (\mathbf{y}_1 + \mathbf{x}_1, \mathbf{x}_{2:4}^{\mathrm{T}} + e^{-\mathbf{x}_1}\mathbf{y}_{2:4}^{\mathrm{T}})^{\mathrm{T}}. \tag{13}$$

It can be seen from (13) that the Jacobian determinant of (11) is equal to 1. Therefore, $D(\phi(\mathbf{Y}\phi^{-1}(\mathbf{x}))) = |1| d\mathbf{x}_1 d\mathbf{x}_2 d\mathbf{x}_3 d\mathbf{x}_4 = D(\mathbf{x})$. □

Lemma 1 implies that if the dilatations are uniformly distributed in $\mathcal{DT}(3)$, *i.e.* distributed by a (left-)Haar measure [9, 24], their coordinates *via* $\phi(\cdot)$ are uniformly distributed in $\mathbb{R}^4$, and vice versa. Combining this fact with the fact that the quantizer η partitions $\mathbb{R}^4$ into cells with equal volumes, we conclude that *if the dilatations are uniformly distributed, their hash keys are uniformly distributed.*

4.2. Comparing 3D Rotations

Unlike the general case of robustly matching rotations where both inputs can be noisy, we argue that the rotation of a training feature is usually an inlier since the training data is often clean, *e.g.* when generated from a 3D object model. Thus, our method mostly compares rotations from the scene with inliers. To exploit this fact, apart from using (2), we propose to use its left-invariant version, given by (see *e.g.* [24]):

$$d'(\mathbf{R}, \mathbf{X}_\mathbf{R})^2 := d(\mathbf{I}, \mathbf{R}^{-1}\mathbf{X}_\mathbf{R})^2, \tag{14}$$

where $\mathbf{R}$ is the rotation of a training feature and $\mathbf{X}_\mathbf{R}$ is a rotation from the scene.

Interestingly, the distance is *equivalent to the Euclidean distance under RM*, since:

$$\frac{1}{2}\left\|\mathbf{R} - \mathbf{X}_\mathbf{R}\right\|_{\mathrm{F}}^2 = (1-\cos\alpha)^2 + (\sin\alpha)^2 \tag{15}$$

$$= (1-\cos\alpha)^2 + \left\|\mathbf{0} - \mathbf{v}\sin\alpha\right\|^2 \tag{16}$$

$$= \left\|\mathrm{faq}(\mathbf{I}) - \mathrm{faq}(\mathbf{R}^{-1}\mathbf{X}_\mathbf{R})\right\|^2 = d'(\mathbf{R}, \mathbf{X}_\mathbf{R})^2, \tag{17}$$

where α and $\mathbf{v}$ are respectively the angle and axis of $\mathbf{R}^{-1}\mathbf{X}_\mathbf{R}$, (15) is taken from [10], and $\mathrm{faq}(\cdot)$ denotes the FAQ representation of a rotation matrix.

5. Object Tracking in Color Video

In this section we apply the FAQ representation to the task of object tracking. In the following, we discuss our feature representation, the appearance model acquisition and our tracking framework.

5.1. 3D Rotation Features for 2D Color Values

With robustness in mind we design a lighting-invariant color representation for an image region. Due to their simplicity quaternions have been employed to represent RGB vectors. For example, in [16] a pure quaternion is used:

$$q_{\texttt{RGB}} := \mathbf{i}r + \mathbf{j}g + \mathbf{k}b, \tag{18}$$

where r, g and b are color values respectively. Alternatively, a normalized quaternion can be adopted:

$$q_{\texttt{NRGB}} := \frac{\mathbf{i}r + \mathbf{j}g + \mathbf{k}b}{\sqrt{r^2+g^2+b^2}}. \tag{19}$$

Normalization introduces some robustness towards lighting and cast shadows [30], and is particularly suitable for human skin color under changing illumination [27].

While color presents valuable cues, image gradient orientations were found to be particularly useful in the literature [19, 29]. Thus, we formulate the 3D rotation features of color images as the angle of the gradient orientations around the unit axis, given by the normalized RGB color values. In particular, a feature at a pixel location is given as

$$q_{\texttt{FAQ}} := \cos\alpha + \frac{\mathbf{i}r + \mathbf{j}g + \mathbf{k}b}{\sqrt{r^2+g^2+b^2}}\sin\alpha \tag{20}$$

with gradient orientation $\alpha \in (-\pi, \pi)$.

5.2. Tracking with an Adaptive Appearance Model

The acquisition of online-learned appearance models in contrast to *a priori* learned models is considered advantageous for robust tracking in unknown scenes [19, 26]. The direct FAQ representation allows us to adopt an incremental PCA strategy for online learning while utilizing our robust distance for matching 3D rotations. In particular, we replace the scalar values with quaternions and formulate the update of the appearance model using the tools of quaternion PCA [15] and incremental PCA [17, 26].

Finally, we combine the incremental PCA of quaternions with a particle filter to calculates the posterior of the system's state based on a transition model and an observation model. This framework is commonly applied, and we base our work on [26] and [19]. In short, our transition model is described as a Gaussian mixture model around an approximation of the state posterior distribution of the previous time-step. Given an independent covariance matrix, which represents the variance in horizontal and vertical displacement, rotation, scale, ratio and skew, we extract a certain number of *particles*, *i.e.* image patches, and find their feature vectors of rotations, given by (20). We then apply our observation model to the extracted image features to find the best match, and to initialize the next Gaussian mixture model for the next frame of the video sequence. Our observation model computes the probability of a sample being generated by the learned eigenspace in the appearance

Method name	Weight
Hashing-CNT	1
Hashing-HAQ	$4 - \min_{\mathbf{R}\in\boldsymbol{V}} \|\mathrm{haq}(\mathbf{R}) - \mathrm{haq}(\mathbf{X_R})\|^2$
Hashing-RV	$4\pi^2 - \min_{\mathbf{R}\in\boldsymbol{V}} \|\mathrm{rv}(\mathbf{R}) - \mathrm{rv}(\mathbf{X_R})\|^2$
Hashing-LI-RV	$\pi^2 - \min_{\mathbf{R}\in\boldsymbol{V}} \|\mathrm{rv}(\mathbf{R}^{-1}\mathbf{X_R})\|^2$
Hashing-FAQ	$4 - \min_{\mathbf{R}\in\boldsymbol{V}} \|\mathrm{faq}(\mathbf{R}) - \mathrm{faq}(\mathbf{X_R})\|^2$
Hashing-LI-FAQ	$4 - \min_{\mathbf{R}\in\boldsymbol{V}} \|\mathrm{faq}(\mathbf{I}) - \mathrm{faq}(\mathbf{R}^{-1}\mathbf{X_R})\|^2$

Table 1: *Weighting strategies for different methods. Functions* $\mathrm{haq}(\cdot)$, $\mathrm{rv}(\cdot)$, $\mathrm{faq}(\cdot)$ *are representations of a 3D rotation matrix.*

model, which we assume to be proportional to

$$e^{-\gamma\left\|\mathbf{q}_i - \boldsymbol{U}_t\boldsymbol{U}_t^{\mathrm{H}}\mathbf{q}_i\right\|_{\mathrm{F}}^2}, \tag{21}$$

where $\mathbf{q}_i$ is the FAQ representation vector of the tested particle, $\boldsymbol{U}_t$ is the current subspace, $(\cdot)^{\mathrm{H}}$ computes the Hermitian transpose, and γ is a parameter that controls the spread. We update the appearance model with the best matching particle. We refer to [19, 26] for details of this approach.

6. Experiments

Our representation of 3D rotations is evaluated on 3D recognition and object tracking in color video.

6.1. 3D Recognition

We evaluate our hash map with FAQs from §4 using the public Toshiba CAD model point clouds dataset [25]. The dataset consists of 1000 test sets of votes, each computed from a point cloud containing a single rigid object – one of 10 test objects. Training and test features are provided.

We compare the proposed method and five variants. These methods differ in line 6 of alg. 2, where different weighting strategies corresponding to different distances are adopted (see tab. 1). We use `hashing-CNT` as the baseline method for finding σ_s and σ_t, as this variant is purely based on the count of matching dilatations.

To find the best values for σ_s and σ_t, we adopt a grid search using leave-one-out cross validation, similar to [25, 31]. We maximize the recognition rate, followed by the registration rate – each registration is evaluated as a binary score on the displacement error (see [25] for details). The best result for `hashing-CNT` is found at $(\sigma_s, \sigma_t) = (0.111, 0.92)$ where the recognition rate is 100% and the registration rate is 86.7% (tab. 2, row 2). This method alone outperforms the best-performing method on this dataset, the minimum-entropy Hough transform [31] (tab. 2, row 1). It also produces a perfect recognition rate, leaving little room for improvement of the registration score.

We perform cross validation over the other five variants using the same values for (σ_s, σ_t), so that their results are comparable (see tab. 2). Generally, it is difficult to improve significantly upon the performance of the proposed `hashing-CNT`. We obtain a 100% recognition rate and a slightly higher registration rate than that of `hashing-CNT` in all cases. `Hashing-LI-FAQ` gives the best registration rate, followed by `hashing-HAQ`, `hashing-LI-RV`, and `hashing-FAQ`, and `hashing-RV`. The left-invariant distances of RV and FAQ outperform their non-invariant counterparts respectively.

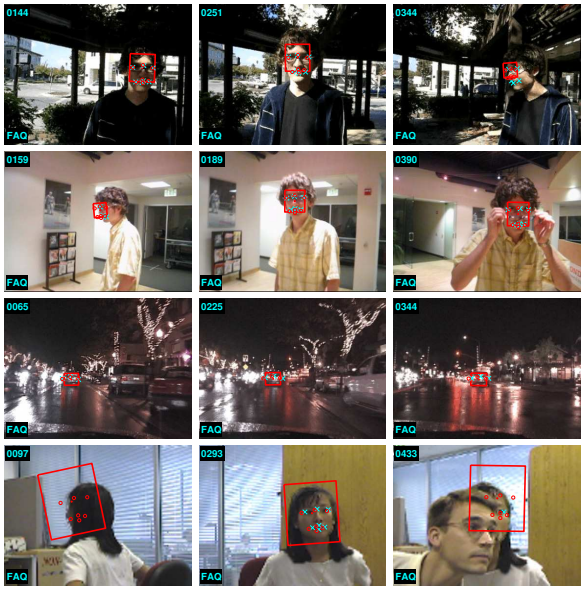


Figure 4: *Example frames from the tracked video sequences with results of the proposed FAQ tracking overlayed.*

Unlike existing vote-evaluation methods [2, 12, 21], the evaluation time per test set (10,000 votes on average) of our methods (last column of tab. 2) are even faster than the fast minimum entropy Hough transform approach [31]. Among them, distances based on FAQ have a slight advantage.

6.2. Object Tracking in Color Video

We now evaluate our 3D rotation representation in the set-up of §5 on data with outliers caused by varying lighting, occlusions and appearance changes. Our system (`FAQ`) with quaternions as in (20) is compared to the competitive tracker of `Euler-PCA` [18], and the original 2D distance [29] (`Gradients`), both with gray scale images. We also include the results of the RGB quaternion (`RGB-Quat`) in (18) and its normalized version (`N-RGB-Quat`) in (19). Finally, we replace our `FAQ` with the same rotation, but represented by `HAQ` and `RM` (tab. 3). Tests are performed on four color videos, Vid1 to Vid4, taken from [1] and [26] (fig. 4), which gray scale versions are commonly used in tracking [18, 26]. In our particle filter, we fix the covariance as follows: translation x = 0.11, y = 0.09, scale = 1.4, rotation = 0.02, ratio = 0.002 and skew = 0.001. Online learning is performed with optimal component numbers for each set-up (see tab. 3). Analogously to [18], we extract 800 particles at each frame and update with a batch of 5

Method name	Registration rate per object (%)										Registration rate (%)	Recognition rate (%)	Time (s)
	Bearing	Block	Bracket	Car	Cog	Flange	Knob	Pipe	Piston1	Piston2			
Min-entropy [31]	83	20	98	91	**100**	86	91	89	54	84	79.6	98.5	0.214
Hashing-CNT	85	**31**	**100**	**97**	**100**	95	99	**92**	71	97	86.7	**100**	**0.092**
Hashing-HAQ	91	29	**100**	95	**100**	94	99	90	83	96	87.7	**100**	0.103
Hashing-RV	92	23	**100**	94	**100**	89	**100**	89	81	94	87.3	**100**	0.117
Hashing-LI-RV	92	28	**100**	95	**100**	94	99	90	83	96	87.7	**100**	0.106
Hashing-FAQ	93	27	**100**	95	**100**	92	99	89	**84**	**98**	87.7	**100**	0.097
Hashing-LI-FAQ	**94**	26	**100**	95	**100**	**97**	99	90	82	96	**87.9**	**100**	0.095

Table 2: ***Qualitative results*** *for all methods. Bold values indicate the best achieved results across all methods.*

Method name	Feature representation	Components
Euler-PCA	$\cos(1.9\pi i) + \mathbf{i}\sin(1.9\pi i)$	15
Gradient	$\cos(\alpha) + \mathbf{i}\sin(\alpha)$	35
RGB-Quat	$\mathbf{i}r + \mathbf{i}g + \mathbf{i}b$	20
N-RGB-Quat	$\frac{\mathbf{i}r+\mathbf{i}g+\mathbf{i}b}{\sqrt{r^2+g^2+b^2}}$	10
RM	vectorised rotation matrix	40
HAQ	$\cos\frac{\alpha}{2} + \frac{\mathbf{i}r+\mathbf{i}g+\mathbf{i}b}{\sqrt{r^2+g^2+b^2}}\sin\frac{\alpha}{2}$	20
FAQ	$\cos\alpha + \frac{\mathbf{i}r+\mathbf{i}g+\mathbf{i}b}{\sqrt{r^2+g^2+b^2}}\sin\alpha$	40

Table 3: *The different tracking set-ups. The image intensity is given by i, color is given as r, g, b. The gradient angle is* α.

samples. We evaluate the tracking performance based on accuracy (fig. 5), *i.e.* root mean square (RMS) error between predicted and true landmarks, and precision (fig. 6).

`Gradient`, `RM`, `HAQ` and `FAQ` track the target in Vid1 successfully, while all other set-ups, which build upon color and intensity alone, fail to track the video due to cast shadows and large pose variations. In Vid2, the 3D rotation feature-based frameworks (`RM`, `HAQ` and `FAQ`) again outperform the other systems. `N-RGB-Quat` is able to track most of the sequence, but fails during the appearance change in frame 390. All other systems struggle during the motion blur and large pose variation around frame 159. The penultimate video Vid3 is a night-time recoding of a rigid object in low light intensity. Such scene is considered easy to track, and all systems succeed. The low light however slightly reduces the performance of `RM`, `HAQ` and `FAQ`, as the color values for the rotation axes are less reliable. The most challenging video is Vid4, in which the target performs two 360° turns. The `FAQ` method is the only system among the ones compared to succeed in tracking these appearance changes. The combination of robustness to varying appearance and the color and gradient cues make the tracking for `FAQ` possible.

With regards to precision, the group of gradient-based methods (`Gradient`, `RM`, `HAQ`, `FAQ`) performs generally well, while non-gradient-based methods (`Euler-PCA`, `RGB-Quat`, `N-RGB-Quat`) are less precise. Furthermore, systems with color and gradient outperform other methods for most videos – Vid3 is an exception, as the target is less difficult to track and the low lighting causes reduced performance with 3D rotation features.

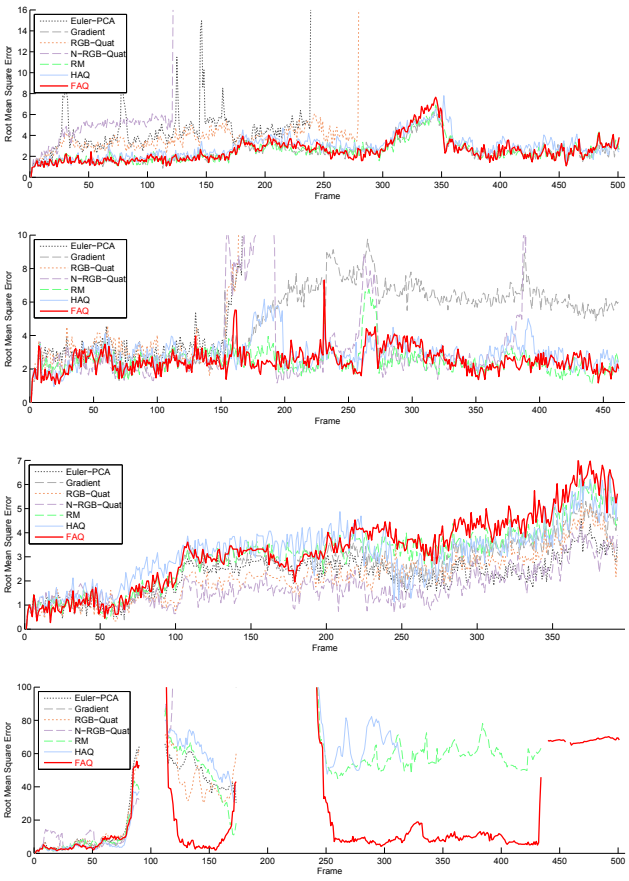


Figure 5: *RMS error of each frame for Vid1, Vid2, Vid3 and Vid4 (top to bottom). Gaps indicate occlusions or tracking failures in cases were the tracked object is empty.*

In general, `FAQ` largely improves upon its 2D version, *i.e.* `Gradient`, and is among the best methods for Vid1, Vid2 and especially Vid4. In comparison to other 3D rotation features, only `RM` performs similarly well in Vid1 and Vid3 – note however, `RM` is slower as the dimensionality is increased by a factor of 9. Finally, we emphasize that `FAQ`'s unique representation of rotations is advantageous to `HAQ`'s dual mapping, as `FAQ` achieves higher precision. We conclude that the FAQ representation can be employed for fast and robust online subspace learning for object tracking.

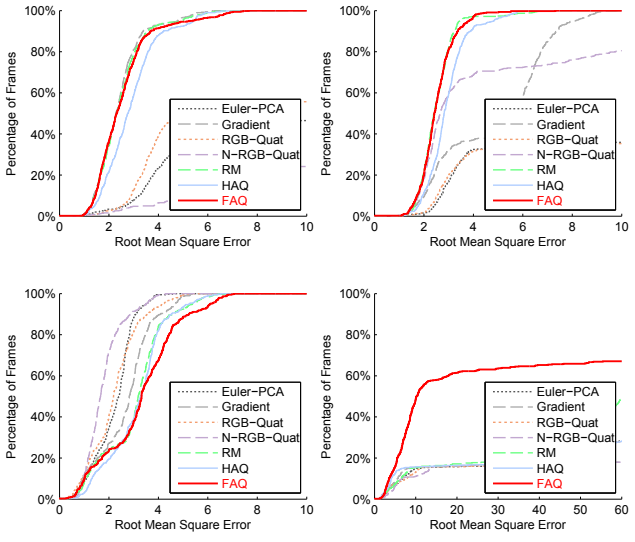


Figure 6: Object tracking results. *Precision is shown for Vid1, Vid2, Vid3 and Vid4 (left to right, top to bottom). Plots show the percentage of frames which have smaller errors than the given RMS error.*

7. Conclusion

We have introduced a new distance for robustly matching vectors of 3D rotations and have shown that this distance leads to an efficient representation in which any 3D rotation maps uniquely to a unit quaternion. We have applied the distance to 3D shape recognition where we introduced efficient hashing of dilatations and obtained state-of-the-art recognition and registration results on a public dataset. In the application to 2D object tracking we have combined online subspace learning with our proposed FAQ representation to facilitate fast and robust updates.

Acknowledgements. The work was carried out during S. Liwicki's internship at Toshiba Research Europe. The team of Imperial College gratefully acknowledges funding by the Engineering and Physical Sciences Research Council (EPSRC) project EP/J017787/1 (4D-FAB), and S. Liwicki's EPSRC Studentship.

References

[1] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *CVPR*, 1998. 6

[2] H. Chen and B. Bhanu. 3d free-form object recognition in range images using local surface patches. *PRL*, 2007. 4, 6

[3] H. S. M. Coxeter. *Introduction to Geometry*. 1961. 1, 4

[4] F. de la Torre and M. J. Black. A framework for robust subspace learning. *IJCV*, 54(1):117–142, 2003. 1

[5] D. Ding, D. Zhou, X. He, and H. Zha. R1-PCA: Rotational Invariant L1-norm Principal Component Analysis for Robust Subspace Factorization. In *ACM*, pages 281 – 288, 2006. 1

[6] A. Fitch, A. Kadyrov, W. Christmas, and J. Kittler. Fast Robust Correlation. *TIP*, 14(8):1063–1073, 2005. 1, 2

[7] C. Gramkow. On averaging rotations. *IJCV*, 2001. 2, 3

[8] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, 2005. 1

[9] A. Haar. Der massbegriff in der theorie der kontinuierlichen gruppen. *Ann. Math.*, 34(1):147–169, 1933. 3, 5

[10] R. Hartley, J. Trumpf, Y. Dai, and H. Li. Rotation averaging. *IJCV*, 103(3):267–305, 2013. 2, 5

[11] D. Q. Huynh. Metrics for 3d rotations: Comparison and analysis. *J. Math. Imaging Vis.*, 35(2):155–164, 2009. 2

[12] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *TPAMI*, 1999. 4, 6

[13] Q. Ke and T. Kanade. Robust L1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming. In *CVPR*, pages 739 – 746, 2005. 1

[14] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool. Hough transform and 3D SURF for robust three dimensional classification. In *ECCV*, 2010. 4

[15] N. Le Bihan and J. Mars. Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing. *Signal Processing*, 84(7):1177 – 1199, 2004. 5

[16] N. Le Bihan and S. Sangwine. Quaternion principal component analysis of color images. In *ICIP*, 2003. 5

[17] A. Levy and M. Lindenbaum. Sequential Karhunen-Loeve Basis Extraction and its Application to Images. *TIP*, 8(8):1371 – 1374, 2000. 5

[18] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Euler Principal Component Analysis. *IJCV*, 2012. 1, 6

[19] S. Liwicki, S. Zafeiriou, G. Tzimiropoulos, and M. Pantic. Efficient Online Subspace Learning with an Indefinite Kernel for Visual Tracking and Recognition. *TNNLS*, 2012. 2, 5, 6

[20] I. Marras, S. Zafeiriou, and G. Tzimiropoulos. Robust Learning from Normals for 3D face recognition. In *ECCV*, 2012. 1, 2

[21] A. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *TPAMI*, 28(10):1584–1601, 2006. 4, 6

[22] M. Moakher. Means and averaging in the group of rotations. *SIAM J. Matrix Anal. Appl.*, 24:1–16, 2002. 2

[23] M. H. Nguyen and F. De la Torre. Robust kernel principal component analysis. In *NIPS*. 2009. 1

[24] X. Pennec and N. Ayache. Uniform distribution, distance and expectation problems for geometric features processing. *J. Math. Imaging Vis.*, 9:49–67, 1998. 3, 5

[25] M.-T. Pham, O. J. Woodford, F. Perbet, A. Maki, B. Stenger, and R. Cipolla. A new distance for scale-invariant 3D shape recognition and registration. In *ICCV*, 2011. 4, 6

[26] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental Learning for Robust Visual Tracking. *IJCV*, 2008. 5, 6

[27] M. Soriano, B. Martinkauppi, S. Houvinen, and M. Laaksonen. Skin detection in video under changing illumination conditions. In *ICPR*, pages 839 – 842, 2000. 5

[28] F. Tombari and L. Di Stefano. Object recognition in 3D scenes with occlusions and clutter by Hough voting. In *PSIVT*, pages 349–355, 2010. 4

[29] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Subspace Learning from Image Gradient Orientations. *TPAMI*, 34(12):2454–2466, 2012. 1, 2, 3, 5, 6

[30] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *TPAMI*, 32(9):1582 – 1596, Sep 2010. 5

[31] O. J. Woodford, M.-T. Pham, A. Maki, F. Perbet, and B. Stenger. Demisting the Hough transform for 3D shape recognition and registration. In *IJCV*, 2013. 4, 6, 7

[32] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *3DRR*, pages 689–696, 2009. 4