

# Averaging Similarity Weighted Group Representations for Pose Estimation

M.Reisert and H.Burkhardt

Computer Science Department, University of Freiburg, 79110 Freiburg i. Br., Germany

Email: {reisert,burkhardt}@informatik.uni-feiburg.de

## Abstract

Correlation based matching techniques have a long tradition in pattern recognition. It is well known that for the translation group the Fourier Transformation is well suited to compute correlation for all possible relative poses of two objects. For the 3D-rotation group Spherical Harmonics play a similar role. Those techniques have in common that the corresponding transformation has to be computed three times to match the objects, i.e. transform the object into the harmonic domain, multiply the transformations, and transform the result back into the original domain. Only the first transformation can be precomputed, the second has to be done 'online'. In this paper we propose a generic method, which circumvents the second transformation. We present preliminary experimental results for the 3D-rotation group, which prove our theoretical results.

**Keywords:** Pose Registration, 3D Matching, Spherical Harmonics, Fourier Transform, Procrustes Analysis

## 1 Introduction

Fast correlation based registration techniques play an important role in many fields, ranging from pattern recognition and image processing [1, 2, 3, 4] to biophysics and computational biology [5]. It is well known that the harmonic transformations (Fourier FT, Spherical Harmonic SHT) play an important role in this subject. While all methods, which are based on harmonic transformations, need to compute the inverse to extract the relative pose parameters, we want to propose a method which avoids this computational expensive inverse. Roughly the technique can be described as follows: We compute linear combinations of group representations, where the representations are weighted by the similarity of the objects in the corresponding pose. The projection of this linear combination down to the manifold of group representations is the estimate for the unknown transformation. If the similarities (or kernels) can be factorized, one is able to precompute the group integration individually for each object and the matching procedure reduces to plugging this precomputed pose-features together. In other words: we compute an intermediate compact representation of the objects, named pose-feature, which has a certain transformation behavior with respect to the considered group and the matching is done by computing a special inner-product of those features. The computation of such features can be efficiently done in the harmonic domain of the considered group.

Our technique has a slight similarity to a method proposed by Gavrilu and Davis [1] named phase coded filtering. But their ideas deal only with linear similarities, which has negative consequences for the accuracy of the method. Our ideas are also related to Procrustes Analysis [6], which is a famous tool for the alignment of point clouds.

This paper is divided in three parts. First we give a general introduction to our approach and give an illustrative example. In Section 3 we apply the framework to the 3D-Rotation group and perform some preliminary experiments evaluating the robustness and accuracy of the pose estimates. And finally we give a conclusion and an outlook for future work.

## 2 The Framework

In this Section we want to propose the fundamental idea of the new matching technique. But at first we have to introduce some useful notation. The objects we want to match live in a real-valued inner-product space  $\mathbb{S}$ , also called signalspace. The vectors in  $\mathbb{S}$  are denoted in boldface  $\mathbf{x}$ , where we refer to the components of the vector by  $[\ ]$ -brackets. We consider locally compact, unimodular groups  $\mathcal{G}$  acting on  $\mathbf{x} \in \mathbb{S}$ , where we write  $g\mathbf{x}$  for the group-action on  $\mathbf{x}$ . The group acts isometrically with respect to the inner-product of  $\mathbb{S}$ , i.e.  $\langle g\mathbf{x}, g\mathbf{y} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle$ . Further we have to introduce the notion of a unitarian representation of  $\mathcal{G}$ , which

is a group homomorphism  $D : \mathcal{G} \mapsto \mathbb{U}_n$ , where  $\mathbb{U}_n$  is the set of  $n$ -dimensional unitary matrices. For our purposes it is necessary that  $D$  is one-to-one, so that we are able to recover  $g$  from  $D(g)$ . The tensor-product is denoted by  $\otimes$ . For example, if  $[\mathbf{x}]_i = x_i$  are the vector components of  $\mathbf{x}$ , then  $[\Phi]_{ij} = x_i x_j$  are components of the tensor  $\Phi = \mathbf{x} \otimes \mathbf{x}$ . For further explanation we refer to literature about multi-linear algebra.

## 2.1 The Idea

We try to estimate the relative pose of two objects  $\mathbf{x}, \mathbf{y} \in \mathbb{S}$  as follows: We evaluate a linear combination of all possible group representations  $D(g)$  with  $g \in \mathcal{G}$ , where every representation corresponds to exactly one group transformation  $\mathbf{y} \mapsto g\mathbf{y}$  in signalspace  $\mathbb{S}$ . The weight for each group representation is given by a function expressing the quality of the corresponding transformation. The quality is given by the similarity of  $\mathbf{x}$  and  $g\mathbf{y}$ . The similarity is typically described by a symmetric, positive definite function  $K : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}$ , also named as kernel. Since the manifold  $\mathbb{U}_n$  of unitarian matrices is not linear, the linear combination is not necessarily unitarian, hence we have to project the linear combination to the 'closest' representation in  $\mathbb{U}_n$  to get the pose estimate. Formally the proposed pose estimate is the projection of

$$P(\mathbf{x}, \mathbf{y}) = \int_{\mathcal{G}} K(\mathbf{x}, g\mathbf{y}) D(g) dg \quad (1)$$

on the  $\mathbb{U}_n$ -manifold. We write  $U_{est}(\mathbf{x}, \mathbf{y}) = \Pi_{\mathbb{U}}[P(\mathbf{x}, \mathbf{y})]$  for the projection on  $\mathbb{U}_n$ , where the projection minimizes the Frobenius-Norm  $\|P(\mathbf{x}, \mathbf{y}) - U_{est}(\mathbf{x}, \mathbf{y})\|$  (for further explanation see [7]). Using the theory of positive definite kernels one is able to factorize the kernel as  $K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle_{\mathbb{H}}$ , where  $\Phi(\mathbf{x})$  and  $\Phi(\mathbf{y})$  live in high dimensional feature space  $\mathbb{H}$ . This property enables us to precompute the computational expensive integration and obtain  $P(\mathbf{x}, \mathbf{y})$  as a simple inner-product in a new induced feature space  $\mathbb{D}$ , which is the tensor-product space of  $\mathbb{H}$  with the representation space  $\mathbb{U}_n$ . Assuming some conditions the factorization looks as follows

$$\begin{aligned} P(\mathbf{x}, \mathbf{y}) &= \frac{1}{\mu(\mathcal{G})} \langle \int_{\mathcal{G}} \Phi(g\mathbf{x}) \otimes D(g), \int_{\mathcal{G}} \Phi(g\mathbf{y}) \otimes D(g) \rangle_{\mathbb{D}} \\ &= \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle_{\mathbb{D}} \end{aligned} \quad (2)$$

where we introduced the so-called pose-features

$$\Psi(\mathbf{x}) = \frac{1}{\sqrt{\mu(\mathcal{G})}} \int_{\mathcal{G}} \Phi(g\mathbf{x}) \otimes D(g) dg \quad (3)$$

and a new  $\mathbb{U}_n$ -valued inner-product in  $\mathbb{H} \otimes \mathbb{U}_n = \mathbb{D}$ . The expression  $\mu(\mathcal{G})$  denotes the 'volume' of the group  $\mathcal{G}$  with respect to the group measure  $\mu$ . For a detailed derivation of this result and further explanations see [7]. Let us examine some properties of  $\Psi(\mathbf{x})$  and of the inner-product given above. At first it transforms to actions on  $\mathbf{x}$  as follows  $\Psi(g\mathbf{x}) = \Psi(\mathbf{x}) \cdot D(g)^\dagger$  and hence the estimate  $P(\mathbf{x}, \mathbf{y})$  has the following transformation behavior

$$P(g\mathbf{x}, g'\mathbf{y}) = D(g) \cdot P(\mathbf{x}, \mathbf{y}) \cdot D(g')^\dagger,$$

where  $g, g' \in \mathcal{G}$  may be arbitrarily chosen. This is a necessary condition for a pose estimate. Secondly, estimating the relative pose of two identical objects in the same pose should lead to the 'identity' result. Stating this formally:  $U_{est}(\mathbf{x}, \mathbf{x}) = \Pi_{\mathbb{U}}[P(\mathbf{x}, \mathbf{x})] = I_n$ . It is easy to show that  $P(\mathbf{x}, \mathbf{x})$  is hermitian, from which follows that the projection is the identity (for a derivation see [7]). Using this result and the transformation rule from above one also gets that  $U_{est}(\mathbf{x}, g\mathbf{x}) = D(g)$ , which is a necessary result.

Besides, the proposed pose estimate can also be derived by minimizing an optimality criterion. The objective is to minimize the squared norm

$$J(g) = \|\Psi(\mathbf{x}) - \Psi(g\mathbf{y})\|^2,$$

where the norm is the Frobenius-Norm generalized to  $\mathbb{D}$ . One can show (see [7]) that the transformation  $g$  corresponding to our estimate  $U_{est}(\mathbf{x}, \mathbf{y})$  actually minimizes  $J(g)$ , i.e. our pose estimate optimally aligns the pose features in a least square sense. Note, that this shows a close relation of our method to the Procrustes Analysis (PA). A PA of our pose features actually leads to our pose estimate. But in contrast to our method in standard PA point-correspondences between the objects have to be known. Our method provides a global characterization of the object, such that no correspondences are needed.

Finally, we have to mention an inherent problem with our approach. If an object  $\mathbf{x}$  shows some symmetry, i.e. there is a  $g$  unequal to the identity such that  $\mathbf{x} = g\mathbf{x}$ , then there are obviously multiple solutions to the matching problem. In other words, if  $g'$  is the true pose relation of two objects and  $g$  is a symmetry generating element, then also  $gg'g^{-1}$  turns the objects into the same pose. In our approach the pose estimate is somehow the average of the multiple solution, which is of course the wrong answer.

## 2.2 Second-Order Pose-Features

Because we want to prevent the explicit integration of (1) 'online', we are working in the featurespace  $\mathbb{H}$ ,

which is in most cases very high-dimensional. The probably simplest non-linear Kernel is

$$K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^2. \quad (4)$$

We can identify the corresponding featurespace with the space of second-order symmetric tensors, i.e.  $\mathbb{H} = \mathbb{S} \otimes \mathbb{S}$  and the featuremap is given by  $\Phi(\mathbf{x}) = \mathbf{x} \otimes \mathbf{x}$ . The usual inner-product in  $\mathbb{S} \otimes \mathbb{S}$  is defined by the rule  $\langle \mathbf{x}_1 \otimes \mathbf{x}_2, \mathbf{y}_1 \otimes \mathbf{y}_2 \rangle = \langle \mathbf{x}_1, \mathbf{y}_1 \rangle \langle \mathbf{x}_2, \mathbf{y}_2 \rangle$  and its linear extensions. If two second-order tensors  $\Phi$  and  $\Phi'$  are interpreted as a matrix, this inner-product can be written as  $\langle \Phi', \Phi \rangle = \text{trace}(\Phi' \Phi^\dagger)$  and the induced norm  $\|\Phi\| = \sqrt{\text{trace}(\Phi \Phi^\dagger)}$  is the Frobenius-Norm.

In the following experiments we want to restrict on the kernel given in (4), but generalizations are straightforward.

### 2.3 The Cyclic-Translation Group - An Example

We consider the  $n$ -dimensional realspace  $\mathbb{R}^n$  as Signalspace  $\mathbb{S}$ . A cyclic translation  $\tau_i$  acts on a object  $\mathbf{x} \in \mathbb{R}^n$  by the following rule  $\langle \mathbf{b}_j, \tau_i \mathbf{x} \rangle = \langle \mathbf{b}_{j-i}, \mathbf{x} \rangle$ , where  $\mathbf{b}_j$  denotes the standard basis and indices have to be seen modulo  $n$ . In discrete Fourier space the rule translates to  $\langle \mathbf{e}_k, \tau_i \mathbf{x} \rangle = \langle \mathbf{e}_k, \mathbf{x} \rangle e^{i \frac{2\pi}{n} k i}$ . A unitary representation of the group is simply given by  $D(\tau_i) = e^{i \frac{2\pi}{n} i}$ . Following definition (3) of the pose-features and using the second-order kernel (4) the components in Fourier space of the posefeatures look as follows

$$[\Psi(\mathbf{x})]_{k_1 k_2} = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} \tilde{x}_{k_1} \tilde{x}_{k_2} e^{i \frac{2\pi}{n} (k_1 + k_2 + 1) i},$$

thereby we used  $\tilde{x}_k = \langle \mathbf{e}_k, \mathbf{x} \rangle$ . For simplicity and efficiency reasons we compute the posefeatures directly in the Fourier space, this is possible due to the isometry of the Fourier transform. After performing the summation one obtains that there are only  $n$  components unequal to zero, namely

$$[\Psi(\mathbf{x})]_k = \sqrt{n} \tilde{x}_{-k} \tilde{x}_{k+1} = \sqrt{n} \tilde{x}_k^* \tilde{x}_{k+1}$$

The inner-product in  $\mathbb{D}$  is the usual sesquilinear one known from complex-valued Hilbertspaces, i.e. the pose estimate of two objects is given by

$$P(\mathbf{x}, \mathbf{y}) = \sum_{k=0}^{n-1} [\Psi(\mathbf{x})]_k^* [\Psi(\mathbf{y})]_k$$

The projection on  $\mathbb{U}_1$  is trivial, because the representation consists only of one complex number and thus the argument of the complex number  $P(\mathbf{x}, \mathbf{y})$  is already the estimate for the unknown group parameter.

## 3 3D Rotations

Now we have a closer look what happens if we calculate pose-features for the 3-dimensional rotation-group  $SO(3)$  acting on real-valued functions defined on the sphere  $S^2$ . It is well known that functions defined on  $S^2$  can be expanded in terms of spherical harmonics. For an introduction for the theory of spherical harmonics see for example [8]. In polar coordinates the function  $x : S^2 \mapsto \mathbb{R}$  representing the signal  $\mathbf{x}$  is written by

$$x(\phi, \theta) = \sum_{l=0}^{\infty} \sum_{m=-l}^l a_m^l Y_m^l(\phi, \theta), \quad (5)$$

where the  $Y_m^l(\phi, \theta)$  are the usual spherical harmonics in polar representation and the coefficients are obtained by  $a_m^l = \langle \mathbf{Y}_m^l, \mathbf{x} \rangle$  or as a shorthand we write  $a^l = \langle \mathbf{Y}^l, \mathbf{x} \rangle$ . In practice, the outer infinite sum of (5) is truncated at some finite cutoff index  $l_{max}$ . A rotation, denoted by  $r$ , acts on the  $a^l$  by  $\langle \mathbf{Y}^l, r \mathbf{x} \rangle = D^l(r) a^l$ , thereby the  $D^l(r) \in \mathbb{U}_{(2l+1)}$  are the so called Wigner D-matrices, which are all representations of the rotation group, but only  $D^1$  is one-to-one. According to the definition of the posefeatures (3), using the second-order kernel and  $D^1$  as unitary representation we arrive at

$$[\Psi(\mathbf{x})]_{l_1 m_1}^{l_2 m_2} = \int_{SO(3)} \left( \sum_{m'=-l_1}^{m'=l_1} a_{m'}^{l_1} D_{m_1 m'}^{l_1}(r) \right) \cdot \left( \sum_{m''=-l_2}^{m''=l_2} a_{m''}^{l_2} D_{m_2 m''}^{l_2}(r) \right) D^1(r)^*$$

Again we compute the posefeatures directly in the Harmonic domain of the transformation. Above we use the complex-conjugate of  $D^1(r)$  for calculational reasons: Pulling the integral inward we have to compute third-order products of D-Wigner matrix-components, which can be done by the following formula (see e.g. [9])

$$\begin{aligned} \int_{SO(3)} D_{m_1 m'}^{l_1}(r) D_{m_2 m''}^{l_2} D_{p q}^l(r)^* & \\ &= \frac{8\pi^2}{2l+1} \langle l p | l_1 m_1, l_2 m_2 \rangle \langle l q | l_1 m', l_2 m'' \rangle \\ &= C_{p m_1 m_2, q m' m''}^{l l_1 l_2} \end{aligned} \quad (6)$$

where  $\langle l p | l_1 m_1, l_2 m_2 \rangle$  are the so called Clebsch-Gordan-coefficients. The CG-coefficients fulfill several selection rules, whether they give a contribution or not. The most important requirement is that  $l, l_1, l_2$  have to fulfill the 'triangle relation', formally  $|l - l_1| \leq l_2 \leq |l + l_1|$ , and secondly it must hold

$m_1 + m_2 = p$ . Applying this relations to our special case we have

$$[[\Psi(\mathbf{x})]_{l_1 m_1 \Delta}]_{pq} = \quad (8)$$

$$\sum_{m'=-l_1}^{l_1} a_{m'}^{l_1} a_{q-m'}^{l_1+\Delta} C_{pm_1(p-m_1), qm'(q-m')}^{ll_1(l_1+\Delta)} \quad (9)$$

thereby  $p, q$  are the matrix-indices of the representation  $D_1$  and the  $\Delta$ -index takes values in  $\{-1, 1\}$ . Actually, the above expression looks a little bit cumbersome and intricate, but for a machine it is easy to evaluate, which is most important. There are still some symmetries and redundancies left in the above expression which we do not mention. But we can conclude that if we need  $N$  coefficients to describe an object by spherical harmonics, then our pose feature is also of size  $\mathcal{O}(N)$ . The inner-product is given by

$$P(\mathbf{x}, \mathbf{y}) = \sum_{l_1 m_1 \Delta} [\Psi(\mathbf{x})]_{l_1 m_1 \Delta}^\dagger [\Psi(\mathbf{y})]_{l_1 m_1 \Delta}$$

Finally, we have to compute the projection  $U_{est}(\mathbf{x}, \mathbf{y}) = \Pi_{\mathbb{U}}[P(\mathbf{x}, \mathbf{y})^*]$  and transform the unitary representation to the more custom orthogonal, real representation  $R_{est}(\mathbf{x}, \mathbf{y}) = VU_{est}(\mathbf{x}, \mathbf{y})V^\dagger$  of the rotation group, where  $V$  is a special unitary transform. Besides, projecting a rotation matrix using the Frobenius-Metric was analyzed in [10] and it was shown that the results of such a projection is very close to the one obtained using the Riemannian metric, which is in some sense more optimal.

### 3.1 Implementation

Everything is implemented in  $C++$  using a Linux system with an *Intel P4 2.8G*, i.e. the running times mentioned below are relative to this machine. For the computation of the Clebsch-Gordan coefficients and the Legendre-Polynomials for the SHT the  $C++$  library *Matpack* release 1.7.3 is used.

### 3.2 Experiments

The following experiment are rather artificial, they should only show that the presented approach is also practical and lead to reasonable results. We consider point clouds given by a set of 3-dimensional points  $\mathbf{P} = \{p_1, \dots, p_n\}$ . For simplicity we present the objects on a single sphere, which is done by projecting the points onto the sphere. Before projecting the coordinates are normalized by shifting the center of gravity into the origin. We compute

$$a_m^l(\mathbf{P}) = \sum_{i=1}^n Y_m^l \left( \frac{p_i - p}{\|p_i - p\|} \right),$$

where  $p = \frac{1}{n} \sum_{i=1}^n p_i$  is the center of gravity of the object.

For the examination we use the  $C_\alpha$ -atoms of mammal antibodies retrieved from the RCSB Protein Database. The antibodies consist of round about 1000 atoms. The considered antibodies have the typical Y-shaped form.

For the first set of experiments we apply different noise models and a random rotation  $R$  on the individual objects and then match each object with the undistorted and unrotated version of itself. We use two performance measures. First, we measure the deviation of the estimated pose  $R_{est}$  to the former applied rotation  $R$ . The error is given by  $E_R = 100 * \|R - R_{est}\|/\sqrt{6}$ , where  $\|\cdot\|$  is the Frobenius-norm and  $\sqrt{6}$  is the upper-bound for the normed difference. Secondly, we compute the deviation of corresponding points in terms of the euclidean distance, i.e. we compute  $E_d = \sqrt{\sum_{i=1}^n \|Rp_i - R_{est}p_i\|^2}$ . Both errors are always averaged over the whole dataset.

For reference we use the well known PCA alignment technique. The objects are aligned by rotating the objects such that the three eigenvectors of their inertia tensors agree in a sorted manner. One of the 8 axial flips is chosen by selecting the one with the lowest error  $E_d$ .

We consider three types of noise: additive normal-distributed noise on the coordinates of the points, random point removal and affine distortions. The noise-levels are given in coordinate units, where the objects' extents are always about 100 coordinate units. In Table 1 and 1 the results for different noise levels and different cutoff parameters  $l_{max}$  are shown. The gaussian noise is simply generated by adding normal distributed vectors with zero mean and given width on the coordinate vectors. For the point-removal noise the percentage of randomly removed points is given. For the affine distortion we applied a matrix  $I_3 + A$  to the object, where the components of  $A$  are normal distributed with zero mean and deviation given in the table.

Let us have a closer look at the results. The simple PCA alignment technique has always better  $E_R$ -values. But by definition  $E_R$  measures the deviation to the original pose and not to the optimal pose. With respect to the  $E_d$ -measure our method outperforms the standard PCA method for affine distortions and point removal. For the gaussian noise they perform nearly equal. Since PCA only incorporates second-

$l_{max} \setminus$ noise	1.0	4.0	7.0	10.0
10	1.1	5.0	9.3	24.9
20	0.6	3.7	8.6	19.3
PCA	0.1	1.7	5.4	10.2

gaussian noise

$l_{max} \setminus$ del pts	10	20	30	40
10	3.9	5.7	6.1	12.8
20	3.1	4.1	5.9	9.6
PCA	2.2	3.3	3.9	7.0

point removal

$l_{max} \setminus$ noise	0.001	0.005	0.01	0.02
10	0.1	0.6	2.3	5.7
20	0.1	0.6	1.8	5.6
PCA	0.2	0.7	1.3	3.4

affine noise

Table 1: Matching error  $E_R$

$l_{max} \setminus$ noise	1.0	4.0	7.0	10.0
10	1.7	6.5	11.3	16.9
20	1.7	6.7	11.7	16.8
PCA	1.6	6.5	11.7	16.0

gaussian noise

$l_{max} \setminus$ del pts	10	20	30	40
10	0.8	2.4	6.9	10.1
20	0.5	2.0	6.7	9.0
PCA	1.2	3.5	8.6	11.4

point removal

$l_{max} \setminus$ noise	0.001	0.005	0.01	0.02
10	0.2	0.8	2.1	7.3
20	0.1	0.5	1.3	6.2
PCA	0.3	1.7	2.9	8.1

affine noise

Table 2: Matching errors  $E_d$ .

order information, meaning the inertia tensor, it is a little more robust to uncorrelated gaussian noise. One can also see that a higher cutoff frequency  $l_{max}$  does not necessarily lead to better results, because the higher frequencies are the noise containing elements.

The running times on our system for the SHT with  $l_{max} = 20$  are about  $180ms$  per object. The computation of the posefeatures take about  $100ms$  per object and the computation time for the inner-product is less than  $1ms$ .

## 4 Conclusion and Future Work

We introduced a general framework for correlation based registration based on group integration. For 3D-Rotations and cyclic Translations the method is efficient since its time complexity is linear in the number of describing coefficients. No correspondences has to be known a priori like in Procrustes Analysis. We gave preliminary experimental results and demonstrated the effectiveness of the approach for 3D-rotations. The new approach was able to outperform the standard PCA approach in two of three cases. For future work it would be interesting to compare our method to schemes like ICP or RANSAC. Due to the generality of our approach there are many extension, which could improve the results, for example the use of higher-order kernels, proper weightings of the features or extension to volume data, instead of simple projections onto the sphere. As our approach is applicable to every unimodular, compact group, there are possibly other application domains, which are interesting.

## References

- [1] D. Gavrila and L. Davis, "Fast correlation matching in large (edge) image databases," in *Proceedings of the 23rd AIPR Workshop, Washington D.C., USA, 1994*.
- [2] P. Kostelec and D. Rockmore, "Ffts on the rotation group," *Santa Fe Institute Working Papers Series*, vol. 03-11-060, Dec 2003.
- [3] A. Makadia and K. Daniilidis, "Direct 3d-rotation estimation from spherical images via a generalized shift theorem," in *Proceedings of the CVPR'03*, vol. 2, pp. 217–225, 2003.
- [4] W. M. R. Blahut and C. Wilcox, "Topics in harmonic analysis with applications to radar and sonar," *IMA Volumes in Mathematics and its Applications*, 1991.

- [5] J. Kovacs and W. Wriggers, “Fast rotational matching,” *Acta Crystallographica Section D*, vol. 58, pp. 1282–1286, 2002.
- [6] I. Dryden and K. Mardia, *Statistical shape analysis*. John Wiley and Sons, 1998.
- [7] M. Reisert, “A new technique for matching in large databases,” technical report <http://lmb.informatik.uni-freiburg.de/people/reisert/pubs/poseest05.pdf>, Albert-Ludwigs-Universität, Freiburg, Institut für Informatik, 2005.
- [8] H. Groemer, *Applications of Fourier Series and Spherical Harmonics*. New York: Cambridge University Press, 1996.
- [9] P. Wormer, “Angular momentum theory,” *Lecture Notes - University of Nijmegen Toernooiveld, 6525 ED Nijmegen, The Netherlands*.
- [10] C. Gramkow, “On averaging rotations,” *International Journal of Computer Vision*, vol. 42(1/2), pp. 7–16, 2001.