ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG
INSTITUT FÜR INFORMATIK
Lehrstuhl für Mustererkennung und Bildverarbeitung

A New Technique For Matching in Large Databases

Internal Report 1/05

Marco Reisert

March 2005

A New Technique For Matching in Large Databases

Marco Reisert
Computer Science Department
Albert-Ludwigs-University Freiburg
79110 Freiburg, Germany
reisert@informatik.uni-freiburg.de

March, 2005

Abstract

This technical report introduces a new matching technique, which is esecially suited for pair by pair matching in large databases. The technique is essentially based on the correlation of two objects. 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 report we propose a method, which circumvents the second transformation, i.e. we are able to obtain a relative pose estimate nearly instantly by the loss of accuracy.

1. Introduction

Fast correlation based matching plays an important role in many fields, ranging from pattern recognition and image processing [3, 7, 1, 5, 8] to biophysics and computational biology [6]. 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 group integrals over representations of the group, where the representations are weighted by the similarity that the current pose is the matching pose. If such probabilities (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 matching is done by computing a special inner-product of those features.

Our technique has a slight similarity to a method proposed by Gavrila and Davis [3] named phase coded filtering. But their ideas deal with linear similarities only, which has negative consequences for the accuracy of the method.

This report is devided in three parts. First we give a general introduction to our approach and give an example in section 3. In section 4 we apply the framework to the 3D-Rotation group and make 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 and give an examplariy application to the 2D cyclic translation group. But at first we have to introduce some useful notation.

2.1. Notation

The objects we want to match live in a real-valued inner-product-space $\mathbb{S}$, also called signalspace. 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 $D(g)$. An element $U \in \mathbb{U}_n$ obeys $U^\dagger U = I_n$, where $U^\dagger$ is the adjungation of U, i.e. $[U^\dagger]_{ij} = [U]^*_{ji}$, where $\cdot^*$ denotes the complex-conjugation. The tensor-product is denoted by $\otimes$, for further explanation have a look into literature about multi-linear algebra.

2.2. The Idea

We try to estimate the relative pose of two objects $\mathbf{x}, \mathbf{y} \in \mathbb{S}$ by computing a similarity weighted integral over all possible relative poses of the two objects, where the current pose is represented by a unitarian representation of the considered group $\mathcal{G}$. The similarity is typically described by a symmetric, positive definite function $K : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}$, also named as Kernel. The relative pose estimate is the projection of

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

on the $\mathbb{U}_n$-manifold. (In the folowing we omit dg for convenience) Before we clarify the notion of a projection on $\mathbb{U}_n$, we want to show that it is possible to factorize the above expression. From the theory of kernels we know that any symmetric, positive definite kernel $K : \mathbb{S} \times \mathbb{S} \mapsto \mathbb{R}$ can be expressed as an inner-product in a feature space $\mathbb{H}$, i.e. $K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle_{\mathbb{H}}$, where $\Phi : \mathbb{S} \mapsto \mathbb{H}$ is the so-called featuremap. In the following we want to assume that the group-action on $\mathbb{S}$ also induces a group-action on $\mathbb{H}$, i.e. $g\Phi(\mathbf{x}) := \Phi(g\mathbf{x})$ is well defined. Using this property, the isometrical action of g on K and the unimodularity of $\mathcal{G}$ it is possible to write (1) as

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{\mu(\mathcal{G})} \langle \int_{\mathcal{G}} \Phi(g\mathbf{x}) \otimes D(g), \int_{\mathcal{G}} \Phi(g\mathbf{y}) \otimes D(g) \rangle_{\mathbb{D}}, \tag{2}$$

where $\otimes$ denotes the tensor-product and $\mathbb{D} = \mathbb{H} \otimes \mathbb{U}_n$ is the corresponding tensor-product space. The induced inner-product $\langle \cdot, \cdot \rangle_{\mathbb{D}}$ is given by the following rule

$$\langle \Phi \otimes D, \Phi' \otimes D' \rangle_{\mathbb{D}} = \langle \Phi, \Phi' \rangle_{\mathbb{H}}\ D^{\dagger} D', \tag{3}$$

where the inner-product is extended by linearity to non-factorized elements of $\mathbb{D}$. We call

$$\Psi(\mathbf{x}) = \frac{1}{\sqrt{\mu(\mathcal{G})}} \int_{\mathcal{G}} \Phi(g\mathbf{x}) \otimes D(g) \tag{4}$$

the pose-feature of $\mathbf{x}$.

Let us verify the factorization property. Starting from

$$\begin{aligned}
P(\mathbf{x}, \mathbf{y}) &= \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle_{\mathbb{D}} \\
&\overset{4}{=} \frac{1}{\mu(\mathcal{G})} \langle \int_{\mathcal{G}} \Phi(g\mathbf{x}) \otimes D(g), \int_{\mathcal{G}} \Phi(g\mathbf{y}) \otimes D(g) \rangle_{\mathbb{D}} \\
&\overset{3}{=} \frac{1}{\mu(\mathcal{G})} \int_{\mathcal{G}} \int_{\mathcal{G}} \langle \Phi(g\mathbf{x}), \Phi(g'\mathbf{y}) \rangle_{\mathbb{H}}\ D(g^{-1}g') \\
&= \frac{1}{\mu(\mathcal{G})} \int_{\mathcal{G}} \int_{\mathcal{G}} K(g\mathbf{x}, g'\mathbf{y}) D(g^{-1}g'),
\end{aligned}$$

using the isometry of the kernel and the unimodularity of the group we are able to substitute $g^{-1}g'$ and get

$$\begin{aligned} P(\mathbf{x},\mathbf{y}) &= \frac{1}{\mu(\mathcal{G})}\int_{\mathcal{G}}\int_{\mathcal{G}} K(\mathbf{x},g\mathbf{y})D(g) \\ &= \int_{\mathcal{G}} K(\mathbf{x},g\mathbf{y})D(g), \end{aligned}$$

which verifies the proposed factorization.

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 $P(\mathbf{x},\mathbf{y})$ transforms by

$$P(g\mathbf{x},g'\mathbf{y}) = D(g)P(\mathbf{x},\mathbf{y})D(g')^\dagger, \tag{5}$$

which is a necessary condition for a pose estimate. Secondly, estimating the relative pose of two identical objects in the same pose the result should lead to the 'identity'. Before stating this formally, we have to become clear about the notion of a projection of arbitrary matrices $\mathbb{C}^{n\times n}$ on $\mathbb{U}_n$.

2.3. Projections on $\mathbb{U}_n$

Let A be some fixed, arbitrary matrix. The the goal is to find the the minimum of $||A-U||$ with respect to $U\in\mathbb{U}_n$. If we decide to let $||\cdot||$ be the Frobenius-Norm, it is equivalent to maximize the expression $\mathrm{Tr}(U^\dagger A + A^\dagger U)$. Now let $A = W^\dagger \Sigma V$ be the singular value decomposition of A. One can show that $2\sum_i \sigma_i$ is a upper bound for the objective, where the σ_i are the singular values of A. The choice $U = W^\dagger V$ reaches this upper bound, hence this choice is optimal. In the following we will write $\Pi_{\mathbb{U}}[A]$ for the projection of A on $\mathbb{U}_n$, i.e. $\Pi_{\mathbb{U}}[A] = W^\dagger V$ holds. Note, that $\Pi_{\mathbb{U}}[QAQ'] = Q\Pi_{\mathbb{U}}[A]Q'$ if $Q,Q'\in\mathbb{U}_n$, which is necessary for our purposes.

Now we are able to formalize the statement given above, namely

$$\Pi_{\mathbb{U}}[P(\mathbf{x},\mathbf{x})] = I_n.$$

Using the symmetry and isometry of K it is easy to show that $P(\mathbf{x},\mathbf{x})$ is hermitian:

$$\begin{aligned} P(\mathbf{x},\mathbf{x})^\dagger &= \int_{\mathcal{G}} K(\mathbf{x},g\mathbf{x})D(g^{-1}) \\ &= \int_{\mathcal{G}} K(\mathbf{x},g^{-1}\mathbf{x})D(g) \\ &= \int_{\mathcal{G}} K(\mathbf{x},g\mathbf{x})D(g) = P(\mathbf{x},\mathbf{x}), \end{aligned}$$

thus for the singular value decomposition of the estimate $W = V$ holds and consequently $\Pi_{\mathbb{U}}[P(\mathbf{x}, \mathbf{x})] = I_n$ is fulfilled, which is a satisfying and necessary result. Another important consequence of this is, that

$$\Pi_{\mathbb{U}}[P(\mathbf{x}, g\mathbf{x})] = D(g),$$

which follows from (5) and the statement above . In Figure 1 we give a rough sketch of the proposed algorithm.

2.4. Relation to Procrustes Analysis

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 $\mathbb{D}$, i.e. the norm is induced by the inner-product defined by $\text{trace}\langle\Psi, \Psi'\rangle_{\mathbb{D}}$. One can show 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. To show this we reformulate the optimization problem. Minimizing $J(g)$ is equivalent to maximizing

$$\begin{aligned} J'(g) &= \text{trace}(\langle\Psi(\mathbf{x}), \Psi(g\mathbf{y})\rangle_{\mathbb{D}} + \langle\Psi(g\mathbf{y}), \Psi(\mathbf{x})\rangle_{\mathbb{D}}) \\ &= \text{trace}(P(\mathbf{x}, g\mathbf{y}) + P(g\mathbf{y}, \mathbf{x})) \end{aligned}$$

Using transformation rule (5), we have

$$J'(g) = \text{trace}(P(\mathbf{x}, \mathbf{y})D(g)^\dagger + D(g)P(\mathbf{y}, \mathbf{x})).$$

Such an expression already occured in the last section about projections to $\mathbb{U}_n$. We sketched that this expression in maximized if $D(g) = W^\dagger V$, where the W and V are given by the SVD of $P(\mathbf{x}, \mathbf{y}) = W^\dagger \Sigma V$. So we already arrived at our goal, $J(g)$ is minimized if g is chosen according to $D(g) = W^\dagger V = \Pi_{\mathbb{U}}[P(\mathbf{x}, \mathbf{y})]$.

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 has to be known. Our method provides a global characterization of the object, such that no correspondences are needed.

2.5. Drawbacks

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.

2.6. 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

$$K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^2. \tag{6}$$

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 $\langle \mathbf{x}_1 \otimes \mathbf{x}_2, \mathbf{y}_1 \otimes \mathbf{y}_2 \rangle = \langle \mathbf{x}_1, \mathbf{y}_1 \rangle \langle \mathbf{x}_1, \mathbf{y}_1 \rangle$ and its linear extensions. In the following experiments we want to restrict on the Kernel given in (6), but generalizations are straight-forward.

3 The 2D Cyclic-Translation Group

We consider 2-dimensional discrete, gray-valued $n \times m$-images as our signalspace $\mathbb{S}$. The pixelvalues of an image $\mathbf{x}$ are denoted by $x_{ij} = \langle \mathbf{b}_{ij}, \mathbf{x} \rangle$, equivalently we are able to represent our image in the discrete fourier space, where we write $\tilde{x}_{kl} = \langle \mathbf{e}_{kl}, \mathbf{x} \rangle$ and the relation between the basissystems is given by $\langle \mathbf{b}_{ij}, \mathbf{e}_{kl} \rangle = e^{\mathbf{i}2\pi(\frac{ki}{n} + \frac{jl}{m})}/\sqrt{nm}$. Elements of the cylic-translation group are denoted by t_{ij}, where t acts by $\langle \mathbf{b}_{ij}, t_{i'j'}\mathbf{x} \rangle = \langle \mathbf{b}_{i-i'j-j'}, \mathbf{x} \rangle$ in spatial domain and by $\langle \mathbf{e}_{kl}, t_{i'j'}\mathbf{x} \rangle = e^{-\mathbf{i}2\pi(\frac{ki'}{n} + \frac{j'l}{m})} \langle \mathbf{e}_{kl}, \mathbf{x} \rangle$ in fourier domain, where all indices have to seen modulo n respectivly m. A unitary representation of the group is given by

$$D(t_{ij}) = \begin{pmatrix} e^{\mathbf{i}2\pi\frac{i}{n}} & 0 \\ 0 & e^{\mathbf{i}2\pi\frac{j}{m}} \end{pmatrix}$$

Following equation (4) and using the second-order kernel (6) the components of the posefeatures in fourier space (it is much easier to compute the tensor-features in fourier-space and it is of course equivalent) look as follows

$$[\Psi(\mathbf{x})]_{k_1 l_1 k_2 l_2} = \frac{1}{\sqrt{nm}} \sum_{i=1}^{n} \sum_{j=1}^{m} \tilde{x}_{k_1 l_1} \tilde{x}_{k_2 l_2} e^{\mathbf{i}2\pi(\frac{(k_1+k_2)i}{n} + \frac{(l_1+l_2)j}{m})} D(t_{ij}).$$

After performing the summation one yields that there are only $2nm$ components unequal to zero, namely

$$[[\Psi(\mathbf{x})]_{kl}]_{11} = \sqrt{\tfrac{n}{m}}\, \tilde{x}_{(-k)l} \tilde{x}_{(k+1)l} \quad \text{and} \quad [[\Psi(\mathbf{x})]_{kl}]_{22} = \sqrt{\tfrac{m}{n}}\, \tilde{x}_{k(-l)} \tilde{x}_{k(l+1)},$$

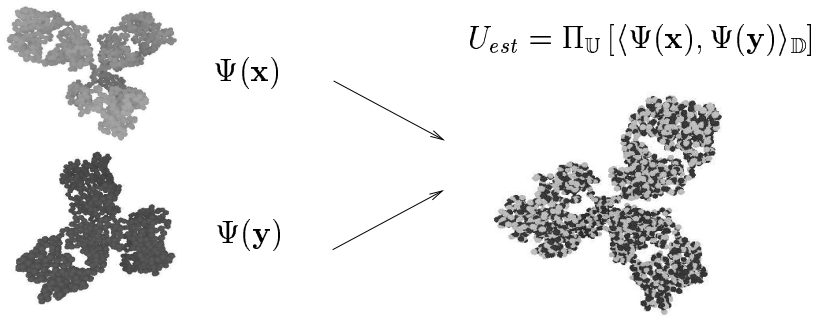

Figure 1: For illustriation of the algorithm. After the determination of the pose-features a simple inner-product gives the estimated pose.

where the subscripts 11 and 22 denote the diagonal-elements of the representation-matrix D. The inner-product in $\mathbb{D}$ according to rule (3) is the usual sesquilinear one known from complex-valued Hilbertspaces, i.e. the pose estimate of two objects is given by

$$[P(\mathbf{x},\mathbf{y})]_{ii} = \sum_{k,l} [[\Psi(\mathbf{x})]_{kl}]^*_{ii} \, [[\Psi(\mathbf{y})]_{kl}]_{ii}$$

where the projection on $\mathbb{U}_2$ is trival, because the representation is already diagonal and thus the complex phases of $[P(\mathbf{x},\mathbf{y})]_{ii}$ are already the estimates for the unknown group parameters.

4. 3D Rotations

Now we have a closer look what happens if we calculate posefeatures 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 are introduction for the theory of spherical harmonics see for example [4]. In polar coordinates the function $x : S^2 \mapsto \mathbb{R}$ representing the signal $\mathbf{x}$ is written by

$$x(\phi,\theta) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a^l_m Y^l_m(\phi,\theta),$$

where the $Y^l_m(\phi,\theta)$ are the usual spherical harmonics in polar representation and the coefficents are obtained by $a^l_m = \langle \mathbf{Y}^l_m, \mathbf{x}\rangle$ or as a shorthand we write $a^l = \langle \mathbf{Y}^l, \mathbf{x}\rangle$. 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 reps of the rotation group, but only D^1 is one-to-one. According to equation

(4) and (6) and using D^1 as unitary representation we arrive at

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

where we used the complex-conjugate of $D^1(r)$ for calculational reasons, which will be explained now. Pulling the integral inward we have to compute third-order products of D-Wigner matrix-components, that can be done by the following formula (see e.g. [9])

$$\begin{aligned} \int\limits_{SO(3)} D_{m_1 m'}^{l_1}(r) D_{m_2 m''}^{l_2} D_{pq}^{l}(r)^* &= \frac{8\pi^2}{2l+1} \langle lp | l_1 m_1, l_2 m_2 \rangle \langle lq | l_1 m', l_2 m'' \rangle \\ &= C^{l l_1 l_2}_{p m_1 m_2,\ q m' m''}, \end{aligned} \tag{7}$$

where $\langle lp | l_1 m_1, l_2 m_2 \rangle$ are the so called Clebsch-Gordan-coefficents. The CG-coefficents fulfill several selection rules, whether they give a contribution or not. The most important thing 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

$$\left[[\Psi(\mathbf{x})]_{l_1 m_1 \Delta} \right]_{pq} = \sum_{m'=-l_1}^{l_1} a_{m'}^{l_1} a_{q-m''}^{l_1+\Delta}\ C^{1 l_1 (l_1+\Delta)}_{p m_1 (p-m_1),\ q m' (q-m')}, \tag{8}$$

thereby p, q are the matrix-indices of the representation D_1 and the Δ-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 by the lack of space. But we can conclude that if we need N coefficents to describe an object by spherical harmonics, then our pose feature will be of size $\mathcal{O}(N)$. The inner-product according to (3) is given by

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

Finally we have to compute the projection $U_{est} = \Pi_{\mathbb{U}} [P(\mathbf{x}, \mathbf{y})^*]$ and transform the unitary representation to the more custom orthogonal, real representation $R_{est} = V U_{est} V^{\dagger}$ of the rotation group, where V is special unitary transform.

4.1. 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 coefficents and the Legendre-Polynomials for the SH-transform the C++ library *Matpack* release 1.7.3 is used.

4.2. Experiments

We consider two types of objects, pointclouds given by a set of 3-dimensional points and surfacemodels given by a triangulated surface. In both cases we have to present the objects on a single sphere, what is done by projecting the points onto the sphere. Before projecting the coordinates are normalized by shifting the center of gravity into the origin. For a pointcloud we compute $a_m^l(\mathbf{P}) = \sum_{p \in \mathbf{P}} Y_m^l(\frac{p}{||p||})$, where $\mathbf{P}$ denotes the normalized set of points. For the surfacemodels we use a Monte-Carlo integration scheme. We generate equidistributed points on the surface of the model and process them like it is done for the pointclouds, i.e. $a_m^l(\mathbf{S}) = \sum_{p \in Rnd(\mathbf{S})} Y_m^l(\frac{p}{||p||})$, where $\mathbf{S}$ denotes the surface and $Rnd(\mathbf{S})$ a random point set drawn from the surface, where $c_{rnd} = |Rnd(\mathbf{S})|$ denotes the number of random events. Note, that for every $a_m^l(\mathbf{S})$ a different set of random points is chosen.

For the examination concerning pointclouds we use the C_α-atoms of mamal antibodies retrieved from the RCSB Protein Database. We decided to analyze objects with over thousand atoms, i.e. a complete antibody, and only the active parts of an antibody with under hundred atoms. The surfacemodels were retrieved from [2] and consist of 40 fish-models.

For the first set of experiments we apply different noise models and a random rotation R on individual objects and then match each object with the undistorted and unrotated version of itself. We measure the deviation of the estimated pose R_{est} to the former applied rotation R. The error is measured 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. The error is averaged over the whole dataset. For reference we use the simple well known PCA alignment technique.

For the pointclouds we consider two types of noise: additive normal-distributed noise on the coordinates of the points and random point removal. The noise-levels are given coordinate units, where the objects' extents are always about 100 coordinate units. In Table 1 and 2 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. Here for both objects different randomly chosen sets of points are removed. For illustration some exemplary results are visualized in Figure 2.

For the surfacemodels we consider additive noise on the coordinates and affine distortions. In table 3 the results are presented. 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. As we are using a Monte-Carlo scheme, the integration error plays an important role, it generates a somehow intrinsic noise, i.e. we cannot expect perfect matches for zero noise levels. In table 3 we give additionally the running-

$l_{max}\backslash$ noise	1.0	4.0	7.0	10.0
10	1	5	10	25
20	1	4	9	20
PCA	0	2	5	10

gaussian noise

$l_{max}\backslash$ del pts	10	20	30	40
10	4	6	6	13
20	3	4	6	10
PCA	2	3	4	7

point removal

Table 1: Matching errors E_R for the antibodies (> 1000 points).

$l_{max}\backslash$ noise	1.0	4.0	7.0	10.0
10	2	9	18	31
20	2	7	17	36
PCA	1	4	6	14

gaussian noise

$l_{max}\backslash$ del pts	10	20	30	40
10	19	36	51	60
20	16	35	47	62
PCA	7	18	23	45

point removal

Table 2: Matching errors E_R for the active centers (< 100 points).

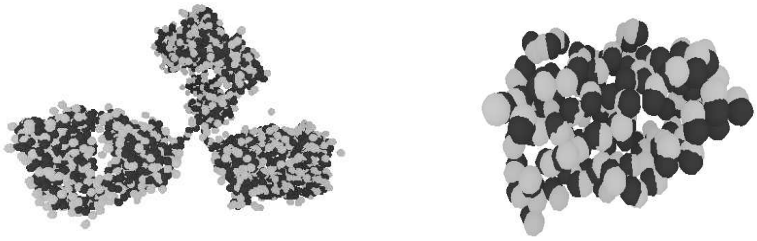

Figure 2: For illustriation of the matching error and the noise-level. Both objects with noise-level of 5.0. Left: (1igt) $E_R = 1$, Right: (active center) $E_R = 5$

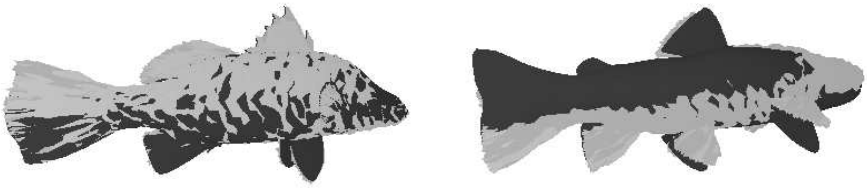

Figure 3: For illustriation of the matching error and the noise-level. Both objects with noise-level of 1.0. Left: $E_R = 3$, Right: $E_R = 17$

times of the SH-transform, because it is the limiting factor, its complexity is of order $\mathcal{O}(l_{max}^2 c_{rnd} \log(n))$, where n is the number of triangles of the objects (in the experiments $n \approx 6000$). In comparison the feature computation and the matching procedure is of order $\mathcal{O}(l_{max}^2)$ and for $l_{max} = 20$ it is about $0.1s$ in our experiments.

Let us have a closer look at the results. The simple PCA alignment technique outperforms the proposed approach in all setups, which is a little bit demoralizing. One can see that a higher cutoff frequency l_{max} do not necessarily lead to better results, because the higher frequency are the noise containing elements. For the objects with a high number of points the matching procedure is very robust against point removal, because the overlap of the objects is high. For objects with a low number of points the results for the point removal are rather bad. In general we can say, that for objects with a high number of points the uncorrelated noise is annihilating itself, which is not so much the case for the low point objects.

If we look at the results of the surfacemodels, they are much worse than for the point clouds. This is obviously caused by the Monte-Carlo integration scheme. But looking at figure 3 errors under $E_R = 10$ are visually acceptable. Interpreting the fishes as pointclouds, one would get nearly the same results like for the antibodies. For the affine noise the results are not so bad in comparsion to the PCA approach like. Of course the running-times for the SHT are high, but remembering that the SHT is done 'offline', this is

$l_{max}\backslash c_{rnd}$	5T	10T	15T	20T	30T	40T
10	14	9	7	5	4	4
20	10	7	6	4	4	3
PCA	5	4	3	2	2	1

without noise

$l_{max}\backslash c_{rnd}$	5T	10T	15T	20T	30T	40T
10	15	11	8	7	5	5
20	11	10	7	7	5	4
PCA	5	5	4	3	2	2

gaussian noise / width 1.0

$l_{max}\backslash c_{rnd}$	5T	10T	15T	20T	30T	40T
10	21	19	18	14	13	13
20	19	18	16	14	12	14
PCA	15	11	10	8	6	5

affine noise / width 0.1

$l_{max}\backslash c_{rnd}$	5T	10T	15T	20T	30T	40T
10	0.3	0.6	0.9	1.2	1.8	2.4
15	0.7	1.3	2.0	2.7	4.0	5.4
20	1.2	2.5	3.6	5.0	7.2	10.0

running times of SHT in seconds

Table 3: Matching errors E_R and running-times for the fish-database.

acceptable.

Finally we want to mention that the reflective symmetry of the fish corpus does not bother our matching technique as one tends to think. A reflection is actually not a member of $SO(3)$, and hence there is no problem for our approach.

In figure 4 we give two examples to show what can happen if we match different objects. The two fishes are matched upside down, but that may also happen to an accurate matcher. Looking at the two different butterflies the result is satisfactory, they are matched exactly into the pose one would expect.

5. Conclusion and Future Work

We introduced a general framework for fast correlation based matching. We demonstrated effectivness of the approach for 3D-rotations. Unfortunately, for the 3D-rotation group it is shown that PCA-alignment would be the better choice. But our approach gives the estimate directly, PCA-based approaches have to determine one of the eight axial flips as a postprocessing step. Due to

Figure 4: Examples. Both matches were performed with $l_{max} = 20$ and $c_{rnd} = 20000$.

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 volumetric data. As the approach is applicable to every unimodular, compact group, there are possibly other application domains, which are interesting.

References

[1] L. Sorgi A. Makadia and K. Daniilidis. Rotation esitimation from spherical images. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 3, pages 590–593, Cambridge, UK, Aug 2004.

[2] Toucan Corporation. 3ds fish models *http://toucan.web.infoseek.co.jp/3DCG/3ds/FishModelsE.html.*

[3] 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.

[4] H. Groemer. *Applications of Fourier Series and Spherical Harmonics.* New York: Cambridge University Press, 1996.

[5] P. Kostelec and D. Rockmore. Ffts on the rotation group. *Santa Fe Institute Working Papers Series*, 03-11-060, Dec 2003.

[6] J. Kovacs and W. Wriggers. Fast rotational matching. *Acta Crystallographica Section D*, 58:1282–1286, 2002.

[7] A. Makadia and K. Daniilidis. Direct 3d-rotation estimation from spherical images via a generalized shift theorem. In *Proceedings of the CVPR'03*, volume 2, pages 217–225, 2003.

[8] W. Miller R. Blahut and C. Wilcox. Topics in harmonic analysis with applications to radar and sonar. *IMA Volumes in Mathematics and its Applications*, 1991.

[9] P. Wormer. Angular momentum theory. *Lecture Notes - University of Nijmegen Toernooiveld, 6525 ED Nijmegen, The Netherlands.*