Rotation Invariant Kernels and Their Application to Shape Analysis

Onur C. Hamsici and Aleix M. Martinez
Dept. of Electrical and Computer Engineering
The Ohio State University, Columbus, OH 43210

Abstract

Shape analysis requires invariance under translation, scale and rotation. Translation and scale invariance can be realized by normalizing shape vectors with respect to their mean and norm. This maps the shape feature vectors onto the surface of a hypersphere. After normalization, the shape vectors can be made rotational invariant by modelling the resulting data using complex scalar rotation invariant distributions defined on the complex hypersphere, e.g., using the complex Bingham distribution. However, the use of these distributions is hampered by the difficulty in estimating their parameters and the nonlinear nature of their formulation. In the present paper, we show how a set of kernel functions, that we refer to as rotation invariant kernels, can be used to convert the original nonlinear problem into a linear one. As their name implies, these kernels are defined to provide the much needed rotation invariance property allowing one to bypass the difficulty of working with complex spherical distributions. The resulting approach provides an easy, fast mechanism for 2D & 3D shape analysis. Extensive validation using a variety of shape modelling and classification problems demonstrates the accuracy of this proposed approach.

Index terms: Shape analysis, kernel functions, rotation invariance, spherical-homoscedastic distributions, face recognition, object recognition, handshape, LB1.

I. Introduction

In 2D and 3D shape analysis one would ideally like to have a representation that is invariant to translation, scale and rotation. One typical way to address this problem is using least-squares (LS) fitting methods, where the goal is to find that transformation of the original shape feature vectors that best approximates one of the sample shapes, e.g., Procrustes analysis [5]. This approach carries several unwanted disadvantages as, for example, misalignments due to noise and outliers. Efficient solutions exist for 2D shapes and two 3D shapes, but iterative procedures are needed to align multiple 3D shapes.

A much sought after alternative is given by the properties inherent to the complex domain, $\mathbb{C}^p$. In this alternate approach, translation, scale and in-plane rotation invariance can be readily achieved by means of a simple mean-norm-normalization step followed by the modelling of the data using a complex probability density function (pdf) with the property $f(\mathbf{z}) = f(\mathbf{z}e^{i\theta})$ for all θ, with $\mathbf{z} \in \mathbb{C}^p$. We will say that a complex random vector $\mathbf{z}$ has complex symmetry if its distribution is invariant to any scalar rotation, $\mathbf{z}e^{i\theta}$. Among the distributions that provide this property, one of the most used in shape analysis is the complex Bingham $\mathbb{C}B(\mathbf{A})$, where $\mathbf{A}$ is its parameter matrix [14].

In the approach defined in the preceding paragraph, translation invariance is obtained when we subtract the mean from all the feature vectors (i.e., mean-normalization). This is so, because each feature vector contains the 2D coordinates of the object and a mean normalization will move all the feature vectors to the origin. Similarly, a norm-normalization guarantees invariance to scale, since all shapes now share a common size. And, as already shown, complex symmetry permits invariance to rotation.

This approach is related to other recent efforts in the computer vision and machine learning communities. In [16] authors build a rotation invariant representation from a translation invariant bispectrum by projecting the images onto the surface of a sphere. Rotation invariance is given by the spherical harmonics of this representation. Another approach, which is gaining interest, is to construct kernels that are invariant to these transformations. For example, [32] demonstrates that there exist a non-trivial conditionally positive-definite kernel providing scale and rotation invariance. A different type of kernel is defined in [9]. This kernel is based on the distance between samples and all their possible transformations (e.g., scale, translation and rotation). By mapping the data in a kernel space, where the distance between a sample and its transformation is zero, we can derive invariant representations.

These kernels are in contrast to the mean and norm normalizing steps introduced earlier, where scale and translation invariance are inherent to this pre-processing. And, in this alternate representation, rotation invariance

is given by the properties inherent to the complex domain. An advantage of this approach is that shapes do not need to be aligned with respect to their rotation parameters, freeing ourselves from the computations required in the LS formulation and avoiding possible misalignments due to noise and outliers. Unfortunately, the lack of exact solutions for the estimation of the parameters of complex spherical pdf makes these algorithms impractical [21], [10]. Optimization algorithms for finding an estimate of the parameters of the Bingham and similar distributions do exist [21], [17], but have convergence problems (especially when the shape of the object is complex) and are tremendously time consuming. Furthermore, this alternate approach does not apply to 3D shapes, because it is not known how to obtain complex symmetric 3D feature vectors. These issues make this approach impractical and have made the LS formulation and its recent variants the preferred solution [2], [18], [25].

To develop approaches that do not carry the problems associated with the LS formulation, one needs to resolve the issues of parameter estimation in Binghams. In a recent paper [11], we showed under which conditions one can substitute the complex Bingham $\mathbb{C}B(\mathbf{A})$ for the zero-mean complex Normal distribution $\mathbb{C}N(\Sigma)$, where Σ is the well-known parameter matrix of the Gaussian, i.e., the covariance matrix.[1] This exchange can be made whenever the complex Bingham distributions that we need for describing our shape feature vectors are spherical-homoscedastic (SH). In general, two or more distributions are SH if each can be described as a rotated version of the others, yielding a Bayes linear classifier between pairs [10].

In the present paper, we extend on this result. We first show the reason why the SH property allows the interchange of complex Binghams for complex Normals. We then derive a kernel function which can convert any non-SH set of distributions into a SH one. With this kernel trick in place, every shape representation/recognition algorithm is transformed into a simple linear problem.

The *key* concept in this paper is to derive a kernel map which provides invariance to rotation, that is, a kernel function with identical mappings for the feature vectors $\mathbf{z}$ and any rotated version of it, $\mathbf{z}e^{i\theta}$. We refer to these kernel functions as *rotation invariant kernels* (RIK). The advantage of these kernels is that they eliminate the need of the complex symmetric representation. Therefore, the availability of RIK means that we no longer need to utilize complex Bingham distributions. An additional advantage of the RIK approach is that it can be used to represent 2D and 3D shapes, while spherical distributions only work for 2D shapes.

Nonetheless, to be able to use RIK, we require that the original feature vectors be described by SH distributions. This requires that we optimize the parameters of the rotation invariant kernel. In this framework, the most important task is to define a criterion that can find the parameters of our rotation invariant kernel that guarantee SH. A classical way to tune the kernel parameters is to select those that maximize the cross-validation performance of a classifier in classification problems and the mean description error in representation tasks. However this method is not only computationally expensive but, most importantly, does not guarantee that the mapped distributions will be SH. This latter issue could imply that the method will not work for a large number of independent test feature vectors.

To resolve the problem defined in the preceding paragraph, we derive a criterion that is directly proportional to that of SH. That is, when this criterion is maximized, the distributions are SH in the kernel space. Then, we go one step further and derive a criterion that is not only maximized when the distributions are SH, but one that also maximizes between-class distances. This provides a convenient shape representation which can be efficiently used in classification tasks.

We report on extensive experimental results for 2D and 3D shape representation and classification problems. Our results demonstrate that the proposed approach achieves the smallest classification error and the lowest computational cost. A Matlab® implementation of the simplest algorithm can classify faces and objects in less than a second.

The rest of this paper is organized as follows. In section II, we summarize the use of the complex domain as an appropriate tool for shape analysis and present the difficulties associated to estimating the parameters of complex Binghams. In Section III, we introduce the concept of spherical-homoscedastic shapes. Section IV presents the kernel approach. Section V introduces the rotation invariant kernels and the criterion to optimize their parameters. Examples and experimental results are in Section VI. We conclude in Section VII.

II. Background Formulation

One of the most known and used feature representations in 2D shape analysis is that advanced by Kendall [5], [13], [31]. In this representation, the features of the vector $\mathbf{u} = (x_1 + iy_1, \ldots, x_p + iy_p)^T$ correspond to the

[1]Since we are working with a zero-mean Gaussian, the covariance matrix is the same as the autocorrelation matrix.

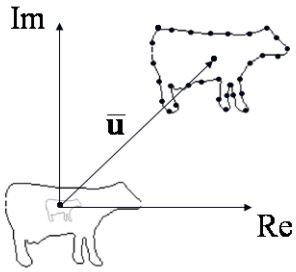


Fig. 1. The shape feature vector of the cow is given by a set of fiducials extracted from the contour shown in the image above. The x and y axes correspond to the real and imaginary components of the representation for each of the fiducials. To obtain a shape vector, we need a representation that is invariant to translation and scale. As seen in the figure, a mean-normalization will center the feature vector around the origin – making the representation invariant to translation. Also shown in the figure is the norm-normalization step, which scales all feature vectors to a common size and thus provides scale invariance.

complex coordinates of a set of points sampled from the shape contour. We illustrate this in Fig. 1, where 29 landmark points are sampled from the contour of an object.

As mentioned earlier, these feature vectors are normalized with respect to their mean and norm, $\mathbf{z} = \mathbf{u}-\bar{\mathbf{u}}/\|\mathbf{u}-\bar{\mathbf{u}}\|$, where $\bar{\mathbf{u}}$ is the mean over all the elements of $\mathbf{u}$. The resulting feature vector $\mathbf{z} \in \mathbb{C}S^{p-2}$ is known as the *preshape*. Note that although $\mathbf{z}$ is a p-dimensional vector, it lies on a $(p-2)$-dimensional manifold. The mean-normalization maps the data onto the null space of the vector of all ones and the norm-normalization maps the data onto the surface of a hypersphere, $\mathbb{C}S^{p-2}$.

A. Complex Bingham

If one uses the shape description presented above, it is then required to model the resulting shape feature vectors with spherical distributions. Since we wish to have a representation that is invariant to rotations, we also need to describe the preshapes $\mathbf{z}$ using a pdf satisfying $f(\mathbf{z}) = f(\mathbf{z}e^{i\theta})$, $\forall\theta \in [0, 2\pi]$. Note that in such a case, multiplying any feature vector with $e^{i\theta}$ defines all possible planar rotations of the preshape in $\mathbb{C}S^{p-2}$. This property is the bases of several pdf, including that of the complex Bingham distribution [14], given by

$$f(\mathbf{z}) = C^{-1}_{\mathbb{C}B}(\mathbf{A}) \exp(\mathbf{z}^*\mathbf{A}\mathbf{z}), \tag{1}$$

where $C_{\mathbb{C}B}$ is a normalizing constant, which guarantees that $\int_{\mathbb{C}S^{p-2}} f(\mathbf{z})\, d\mathbf{z} = 1$, $\mathbf{z}^*$ is the complex conjugate of the transpose of $\mathbf{z}$, and $\mathbf{A}$ is the Hermitian parameter matrix. Using this notation, it is easily verifiable that $f(\mathbf{z}e^{i\theta}) = C^{-1}_{\mathbb{C}B}(\mathbf{A})\ \exp(e^{-i\theta}\mathbf{z}^*\mathbf{A}\,\mathbf{z}e^{i\theta}) = C^{-1}_{\mathbb{C}B}(\mathbf{A}) \exp(\mathbf{z}^*\mathbf{A}\mathbf{z}) = f(\mathbf{z})$, which provides 2D rotation invariance.

The spectral decomposition of the parameter matrix is $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^*$, where $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_p)$ is a matrix whose columns $\mathbf{q}_i$ correspond to the eigenvectors of $\mathbf{A}$ and $\mathbf{\Lambda} = diag\,(\lambda_1, \ldots, \lambda_p)$ is the diagonal matrix of corresponding eigenvalues.[2] For a random set of unit vectors $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ sampled from the complex Bingham distribution, the log-likelihood of the parameters is written as $\mathcal{L}(\mathbf{Q}, \Lambda) = n\, tr(\mathbf{S}\mathbf{Q}\Lambda\mathbf{Q}^*) - n\log(C_{\mathbb{C}B}(\Lambda))$, where $\mathbf{S} = n^{-1}\,\mathbf{Z}\mathbf{Z}^*$ is the sample autocorrelation matrix. Since the $tr(\mathbf{S}\mathbf{A})$ is maximized when the eigenvectors of $\mathbf{S}$ and $\mathbf{A}$ are the same, the maximum likelihood estimate of $\mathbf{Q}$ (denoted $\widehat{\mathbf{Q}}$) is given by the eigenvector decomposition of the sample autocorrelation matrix $\mathbf{S} = \widehat{\mathbf{Q}}\hat{\Lambda}_S\widehat{\mathbf{Q}}$, where $\hat{\Lambda}_S$ is the eigenvalue matrix of $\mathbf{S}$.

Unfortunately, a similar procedure cannot be used to estimate Λ. This is because to compute Λ, one requires to first estimate the normalizing constant, which includes the generally impossible task of integrating the pdf over the nonlinear complex hypersphere $\mathbb{C}S^{p-2}$. We are thus to content ourself with approximations that are not guarantee to be correct all the time. A saddlepoint approximation to the normalizing constant of the complex Bingham distribution is given in [17]. Even where this algorithm converges, optimization algorithms of this sort require of intensive computation, making these approaches impractical. Note that to compute an *approximate* result for the parameters of complex Binghams, we need to calculate the eigenvectors and eigenvalues of $\mathbf{S}$. An eigenvalue

[2]Recall that at least one of the eigenvalues will be zero. This is the one associated to the eigenvector aligned with the vector of all ones.

decomposition is already of $O(p^3)$. Adding, an optimization routine such as that in [17], which requires solving a quadratic programming subproblem at each iteration, leads to a complexity of order 4 or larger.

B. Zero-mean complex Normal

Contrary to Bingham, the Normal distribution is not defined on the surface of a hypersphere, which facilitate its parameter estimation. The pdf of the complex Normal, $\mathbf{z} \sim \mathbb{C}N(\Sigma)$, is given by

$$f(\mathbf{z}) = C_{\mathbb{C}N}^{-1}(\Sigma)\exp(-\mathbf{z}^*\Sigma^{-1}\mathbf{z}), \quad \mathbf{z} \in \mathbb{C}S^{p-2},$$

where Σ is the covariance matrix and $C_{\mathbb{C}N}(\Sigma) = \pi^{p-1}\det(\Sigma)$ is the normalizing constant. Here, it is important to note that the eigenvectors of the covariance matrix Σ are the same as those of $\mathbf{A}$. This is because the maximum likelihood of $\Sigma = \mathbf{S}$, since the distribution has zero mean.

The parameter estimation of the Normal distribution carries a much lower computational cost than that of Bingham. If we have n samples in a class distribution on a $(p-2)$-dimensional complex hypersphere, $\mathbb{C}S^{p-2}$, the zero-mean class covariance matrix can be estimated with np^2 scalar multiplications and p^2 additions. The normalizing constant of the complex Gaussian can be obtained by calculation of the determinant of the covariance. One of the most efficient algorithms developed to date, has an upper-bound complexity of $p^{2.376}$ [3]. This means that algorithms using the complex Normal distribution will have a polynomial time complexity of degree ≤ 2.376, much smaller than that required to estimate Binghams.

It is clear that the use of complex Gaussians facilitates the computational burden. Furthermore, as we prove in the section to follow, the use of Gaussians can guarantee an optimal (wrt Bayes) representation of the shape vectors, which is rarely the case with approximations.

III. Spherical-Homoscedastic Shapes

In the rest of this paper, we will make the assumption of equal priors. This means that every shape is determined equally probable. It also implies that, whenever we represent shapes corresponding to more than one class, the Bayes decision rule simplifies to a comparison of the likelihoods of the distributions. The likelihood of an observation $\mathbf{z}$ to belong to a class can be computed as

$$d^2_{\mathbb{C}N}(\mathbf{z}) = -\log f(\mathbf{z}) = \mathbf{z}^*\Sigma^{-1}\mathbf{z} + \log(C_{\mathbb{C}N}(\Sigma)) \tag{2}$$

for complex Normals, and

$$d^2_{\mathbb{C}B}(\mathbf{z}) \quad = \quad -\mathbf{z}^*\mathbf{A}\mathbf{z} + \log(C_{\mathbb{C}B}(\mathbf{A})) \tag{3}$$

for complex Bingham distributions.

These likelihoods depend on the type of distribution and their parameter matrices. Several particular cases of interest exist. For example, in planar geometry, we say that a set of r Gaussians $\{N_1(\mu_1, \Sigma_1), \ldots, N_r(\mu_r, \Sigma_r)\}$ are *homoscedastic* if their covariance matrices are the same, $\Sigma_1 = \cdots = \Sigma_r$. Here, $\mu_1, \ldots, \mu_r$ are the means for each of the distributions. Homoscedastic Gaussian pdf are relevant, because their Bayes decision boundaries are given by hyperplanes. This makes the derivation of many diverse computer vision and pattern recognition algorithms possible [8].

However, when all feature vectors are restricted to lay on the surface of a hypersphere, the definition of homoscedasticity given above becomes too restrictive. For example, if we use zero-mean Gaussians to model a data-set that is complex symmetric about its mean, then only those distributions that are exactly identical, can be considered homoscedastic. This is illustrated in Fig. 2. In the example given in this figure, the three class distributions have the same covariance matrix up to a rotation, but only those that are identical (i.e., Class 1 and 3) are said to be homoscedastic. Nonetheless, as shown in [10] the decision boundaries for each pair of classes in Fig. 2 are all hyperplanes. A more general definition exists to include all those distributions in the hypersphere that have the same covariance matrix up to a rotation and yield a linear classifier.

Definition 1: [10] Two pdfs (f_1 and f_2) of the same form are said to be spherical-homoscedastic (SH) if the Bayes decision boundary between f_1 and f_2 is given by one or more hyperplanes and the distribution parameters are the same up to rotation.

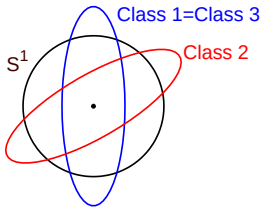


Fig. 2. Assume we model the data of three classes laying on S^1 using three zero-mean Gaussian distributions. In this case, each set of Gaussian distributions can only be homoscedastic when they are the exact same distribution. In this figure, Class 1 and 3 are homoscedastic, but classes 1, 2 and 3 are not. Classes 1, 2 and 3 are spherical-homoscedastic.

Algorithm 1 : *complex Normal*

Normalize the shapes, $\mathbf{z}_{jk} = \mathbf{u}_{jk} - \bar{\mathbf{u}}_{jk}/\|\mathbf{u}_{jk} - \bar{\mathbf{u}}_{jk}\|$, where $\mathbf{u}_{jk}$ is the k^{th} sample of class j.
Let $\mathbf{Z}_j = \left(\mathbf{z}_{j1}, \ldots, \mathbf{z}_{jn_j}\right)$, where n_j is the number of samples in class j.
Compute the sample covariance matrices, $\Sigma_j = n^{-1}\mathbf{Z}_j\mathbf{Z}_j^*$, $j = 1, \ldots, C$, where C is the number of classes.
Calculate the normalizing constants $C_{\mathbb{C}N_j}(\Sigma_j) = \pi^{p-1}det(\Sigma_j)$, $j = 1, \ldots, C$.
The class of a test sample $\mathbf{z}$ is given by $\arg_j \min d^2_{\mathbb{C}N_j}(\mathbf{z}) = \mathbf{z}^*\Sigma_j^{-1}\mathbf{z} + \log(C_{\mathbb{C}N_j}(\Sigma_j))$

Recall that homoscedastic distributions were relevant because their Bayes decision boundaries are linear, facilitating the development of many algorithms. Similarly, SH distributions also result in Bayes linear classifiers, regardless of the form of these distributions.

Another very important property of SH distributions is that they can be substituted by the Gaussian model with no loss in classification. Although the representation will change with such a substitution, the linear classifiers obtained with Binghams and Gaussians will be identical. This result is given, for the particular case of complex Bingham and zero-mean complex Normals in the following theorem, the proof of which is in Supplementary Documentation.

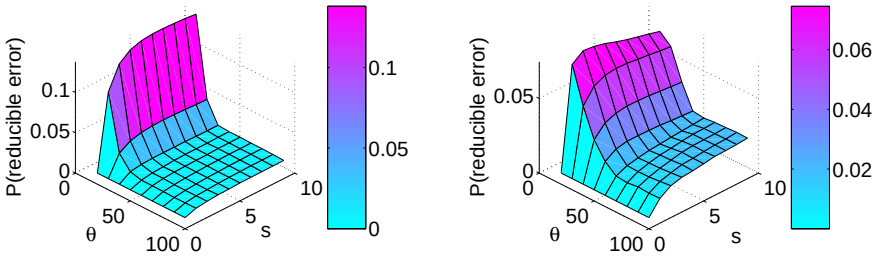
Theorem 2: The Bayes decision boundaries of two spherical-homoscedastic complex Bingham distributions, $\mathbb{C}B_1(\mathbf{A})$ and $\mathbb{C}B_2(\mathbf{R}^*\mathbf{A}\mathbf{R})$, are the same as those obtained when modelling $\mathbb{C}B_1(\mathbf{A})$ and $\mathbb{C}B_2(\mathbf{R}^*\mathbf{A}\mathbf{R})$ with the two zero-mean complex Gaussian distributions $\mathbb{C}N_1(\Sigma)$ and $\mathbb{C}N_2(\mathbf{R}^*\Sigma\mathbf{R})$, where $\Sigma = \mathbf{S}$ and $\mathbf{R}$ defines a planar rotation in the span of (any) two eigenvectors of Σ.

The above result provides the conditions under which one can substitute the complex Bingham for the zero-mean complex Normal. As delineated in the Introduction of this paper, this is a very important result, because it facilitates the computation of the parameters of the distributions describing the shape vectors. Without it, one would need to resort to computational expensive alternatives such as complex optimization procedures or LS fits. Fortunately, if the data can be represented with SH distributions, it can then be modeled with Gaussians. This process is summarized in Algorithm 1. We refer to these shape feature vectors as *spherical-homoscedastic shapes*, or SH shapes for short. Their formal definition is given in the following.

Definition 3: Two shape feature vectors $\mathbf{z}_j \in \mathbb{C}S^{p-2}$ and $\hat{\mathbf{z}}_j \in \mathbb{C}S^{p-2}$ are called *spherical-homoscedastic shapes* (SH shapes) if they are sampled from two SH distributions, i.e., $\mathbf{z}_j \sim \mathbb{C}B(\mathbf{A})$ and $\hat{\mathbf{z}}_j \sim \mathbb{C}B(\mathbf{R}^*\mathbf{A}\mathbf{R})$, where $\mathbf{R}$ defines a planar rotation in the span of any two eigenvectors of $\mathbf{A}$.

A question of interest that follows from Definition 3 is to understand what happens when the shapes deviate from SH. What we know is that, in such cases, the results obtained with the Normal fit will not be the same as those given by Binghams. Nonetheless, we will now show that the more the data distributions deviate from SH, the more different the Normal and Bingham results will become. To see this, we need to calculate the classification error that is added to the original Bayes error when one uses the Normal model in place of the Bingham. Following the classical notation in Bayesian theory, we refer to this added error as the reducible error.

The reducible error can change in two ways. The first case is given when the first eigenvector of a first Bingham distribution deviates from the first eigenvector of a second Bingham. In this case, the reducible error diminishes. This is so because the first eigenvector defines the mean direction of each distribution and the two distributions can be separated more easily as they move away from one another. In the second case, it is easy to prove that more concentrated distributions have a smaller reducible error than those that are spread over a large area on $\mathbb{C}S^{p-2}$, since concentrated pdf need to be separated by a smaller angle (with regard to their first eigenvectors) than those



(a) (b)

Fig. 3. *In (a) we plot the probability of reducible error as a function of θ and s for the high concentrated case. In (b) we do the same for the low concentrated case.*

that are less concentrated. Recall that the concentration of each distribution is determined by the eigenvalues.

To see the effect of the arguments described in the preceding paragraph at work, we construct two examples. In creating these examples, we define the parameter matrices of $\mathbb{C}B_1(\mathbf{A}_1)$ and $\mathbb{C}B_2(\mathbf{A}_2)$ as $\mathbf{A}_1 = diag(\lambda_1, \lambda_2)$ and $\mathbf{A}_2 = \mathbf{R}^T diag(\lambda_1, s\lambda_2)\mathbf{R}$, where $\mathbf{R}$ is an orthonormal planar rotation matrix defined by the angle θ, and s is a scaling parameter. We set $\lambda_1 = -1/2$, and change the value of λ_2, s and θ to simulate the two cases described above – one where the data is highly concentrated and another where the data is more spread out. In the first case, $\lambda_2 = \{-1/(2j)|j = 20, 30, \ldots, 100\}$. In the second case, $\lambda_2 = \{-1/(2j)|j = 2, 3, \ldots, 10\}$. In both cases, $s = \{1, 2, \ldots, 10\}$ and $\theta = \{10^o, 20^o, \ldots, 90^o\}$. The reducible error is estimated by randomly drawing $5,000$ samples from each class and by calculating the ratio between the samples that are incorrectly classified by the Gaussian approximation but correctly classified by the Bingham pdf and the total number of sample vectors.

The results are in Fig. 3. In (a) we plot the reducible error as a function of the rotation angle θ and the scale parameter s for the highly concentrated distributions. In (b), we do the same for the case where the data is less concentrated. Note that, when $s = 1$ the data is SH and, therefore, the reducible error is zero. In both cases, it is apparent that the reducible error is very small for distributions whose parameters are close to those of SH shapes. As we start to deviate from SH, the error increase. This is especially true when the distributions are close to one another.

In summary, the reducible error increases as the distributions deviate from spherical-homoscedasticity, with the increase on the error inverse-proportional to the distance (angle) between distributions. Our goal thus reduces to finding a shape representation where the distributions used to model the shape feature vectors are as close to SH and as far apart from each other as possible. In the sections to follow, we use this result to derive a kernel function which maps the original shape representation to one that best adapts to the SH model while keeping the distributions of different classes apart.

IV. Kernel Spherical-Homoscedastic Shapes

SH shapes are important because they are optimally (wrt Bayes) separated and described with linear methods. We now employ the well-known idea of the kernel trick to define algorithms that are nonlinear in the original space, but linear in the kernel one.

A. Two-dimensional shapes

Key to defining kernel SH shapes is to realize that by using *rotation invariant kernels*, we can drop the requirement of working with complex distributions (with the symmetric property), since rotation invariance will now be directly provided by the kernel map. To see this, let us start with the definition in the complex domain and then see how one can model the data with real pdf.

Our first rotation invariant kernel is given by

$$k(\mathbf{z}_j, \mathbf{z}_k) = \exp\left(-\frac{\|\mathbf{z}_j - \mathbf{z}_k \exp(-i\theta_{jk})\|^2}{2\sigma^2}\right),$$

where $\mathbf{z}_j, \mathbf{z}_k \in \mathbb{C}S^{p-2}$, σ is the kernel parameter to be optimized, and $\theta_{jk} = \angle(\mathbf{z}_j^* \mathbf{z}_k)$ defines the angle between $\mathbf{z}_j$ and $\mathbf{z}_k$.

This kernel is invariant to any arbitrary planar rotation θ_{jk} between two feature vectors $\mathbf{z}_j$ and $\mathbf{z}_k$. However, in principle, this kernel is not very different from the least-squares or alignment algorithms summarized in the Introduction of this paper, because our kernel still requires that we compute θ_{jk} for every pair of shape vectors. The real advantage of this kernel is that it can be reworked as follows

$$\|\mathbf{z}_j - \mathbf{z}_k \exp(-i\theta_{jk})\|^2 = \mathbf{z}_j^*\mathbf{z}_j + \mathbf{z}_k^*\mathbf{z}_k - \mathbf{z}_j^*\mathbf{z}_k \exp(-i\theta_{jk}) - \exp(i\theta_{jk})\mathbf{z}_k^*\mathbf{z}_j = 2 - 2\|\mathbf{z}_j^*\mathbf{z}_k\|. \tag{4}$$

From this result, we have

$$k(\mathbf{z}_j, \mathbf{z}_k) = \exp\left(-\frac{2 - 2\|\mathbf{z}_j^*\mathbf{z}_k\|}{2\sigma^2}\right). \tag{5}$$

This is a very important result, because it shows that *we can build RIKs that do not require of any alignment of the shapes or the calculation of the rotation angle between each pair of feature vectors.*

We now note that our kernel carries an inherent mapping resulting in a kernel space that is still spherical, because $k(\mathbf{z}, \mathbf{z}) = 1$, $\forall \mathbf{z}$. Hence, one requires to model the data defined in the kernel space using spherical distributions such as the real Bingham, $B_j(\mathbf{A})$.

Let two SH Bingham distributions be $B_1(\widetilde{\mathbf{A}})$ and $B_2(\mathbf{R}^T\widetilde{\mathbf{A}}\mathbf{R})$, with $\mathbf{R}$ an arbitrary planar rotation defined by two eigenvectors of $\widetilde{\mathbf{A}}$, $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$. Here, the Bingham distribution is the real counterpart of the complex Bingham with the density function $f(\mathbf{x}) = C_{\mathbb{B}}^{-1}(\mathbf{A}) \exp(\mathbf{x}^T\mathbf{A}\mathbf{x})$. And, let $\mathbf{x} \sim B_1(\widetilde{\mathbf{A}})$, with $\mathbf{x} = (real(\mathbf{z})^T, imag(\mathbf{z})^T)^T \in \mathbb{R}^{2p}$. The Bayes classification boundary between these distributions can be obtained by making the ratio of their log-likelihood equations equal to one. From (3), we have

$$\mathbf{x}^T\widetilde{\mathbf{A}}\mathbf{x} = \mathbf{x}^T\mathbf{R}^T\widetilde{\mathbf{A}}\mathbf{R}\mathbf{x}.$$

The rotation is defined in the subspace spanned by $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$, and we know that $\widetilde{\mathbf{A}} = \widetilde{\mathbf{Q}}\widetilde{\Lambda}\widetilde{\mathbf{Q}}^T$, with $\widetilde{\mathbf{Q}}$ a matrix whose columns $\tilde{\mathbf{q}}_k$ are the eigenvectors of $\widetilde{\mathbf{A}}$ and $\widetilde{\Lambda} = diag\left(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_{2p}\right)$ is the diagonal matrix of corresponding eigenvalues. Thus, the above equation can be written as $\sum_{k=1}^{2p} \tilde{\lambda}_k(\mathbf{x}^T\tilde{\mathbf{q}}_k)^2 = \sum_{k=1}^{2p} \tilde{\lambda}_k(\mathbf{x}^T\mathbf{R}^T\tilde{\mathbf{q}}_k)^2$. In addition, $\mathbf{R}^T\tilde{\mathbf{q}}_k = \tilde{\mathbf{q}}_k$ for $k > 2$, which simplifies our equation to

$$\sum_{k=1}^{2p} \tilde{\lambda}_k(\mathbf{x}^T\tilde{\mathbf{q}}_k)^2 = \sum_{k=1}^{2} \tilde{\lambda}_k(\mathbf{x}^T\mathbf{R}^T\tilde{\mathbf{q}}_k)^2 + \sum_{k=3}^{2p} \tilde{\lambda}_k(\mathbf{x}^T\tilde{\mathbf{q}}_k)^2$$

$$\tilde{\lambda}_1\left((\mathbf{x}^T\tilde{\mathbf{q}}_1)^2 - (\mathbf{x}^T\mathbf{R}^T\tilde{\mathbf{q}}_1)^2\right) + \tilde{\lambda}_2\left((\mathbf{x}^T\tilde{\mathbf{q}}_2)^2 - (\mathbf{x}^T\mathbf{R}^T\tilde{\mathbf{q}}_2)^2\right) = 0.$$

This means that a given sample $\mathbf{x}$ corresponds to the first distribution, B_1, if

$$\tilde{\lambda}_1\left((\mathbf{x}^T\tilde{\mathbf{q}}_1)^2 - (\mathbf{x}^T\mathbf{R}^T\tilde{\mathbf{q}}_1)^2\right) + \tilde{\lambda}_2\left((\mathbf{x}^T\tilde{\mathbf{q}}_2)^2 - (\mathbf{x}^T\mathbf{R}^T\tilde{\mathbf{q}}_2)^2\right) > 0,$$

and to B_2 otherwise. As shown in Fig. 4, $\tilde{\mathbf{q}}_2$ can also be expressed as a function of $\tilde{\mathbf{q}}_1$ as,

$$\tilde{\mathbf{q}}_2 + \mathbf{R}^T\tilde{\mathbf{q}}_2 = (\mathbf{R}^T\tilde{\mathbf{q}}_1 - \tilde{\mathbf{q}}_1)\cot\left(\frac{\theta}{2}\right),$$

$$\tilde{\mathbf{q}}_2 - \mathbf{R}^T\tilde{\mathbf{q}}_2 = (\mathbf{R}^T\tilde{\mathbf{q}}_1 + \tilde{\mathbf{q}}_1)\tan\left(\frac{\theta}{2}\right),$$

where θ is the angular rotation defined by $\mathbf{R}$. Using this result in our equation above yields

$$(\tilde{\lambda}_1 - \tilde{\lambda}_2)\left((\mathbf{x}^T\tilde{\mathbf{q}}_1)^2 - (\mathbf{x}^T\mathbf{R}^T\tilde{\mathbf{q}}_1)^2\right) > 0.$$

If $\tilde{\lambda}_1 > \tilde{\lambda}_2$, $\mathbf{x}$ will be in B_1 when

$$(\mathbf{x}^T\tilde{\mathbf{q}}_1)^2 > (\mathbf{x}^T\mathbf{R}^T\tilde{\mathbf{q}}_1)^2,$$

and in B_2 otherwise. This result can be stated in a more compact form as

$$|\mathbf{x}^T\tilde{\mathbf{q}}_1| > |\mathbf{x}^T\mathbf{R}^T\tilde{\mathbf{q}}_1|. \tag{6}$$

The relevance of (6) is that the class of a test feature vector $\mathbf{x}$ is simply given by the largest inner product. This is a simple computation, providing a robust and fast mechanism for classifying shapes.

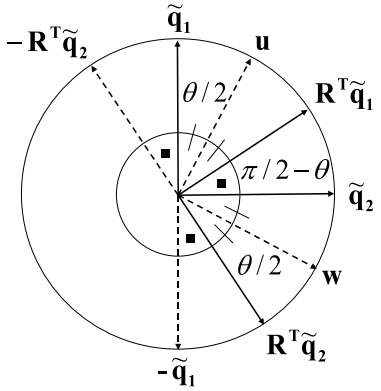


Fig. 4. *Shown here are two orthonormal vectors,* $\tilde{\mathbf{q}}_1$ *and* $\tilde{\mathbf{q}}_2$*, and their rotated versions,* $\mathbf{R}^T\tilde{\mathbf{q}}_1$ *and* $\mathbf{R}^T\tilde{\mathbf{q}}_2$*. We see that* $\mathbf{R}^T\tilde{\mathbf{q}}_1+\tilde{\mathbf{q}}_1=2\mathbf{u}\cos\left(\frac{\theta}{2}\right)$*,* $\mathbf{R}^T\tilde{\mathbf{q}}_2+\tilde{\mathbf{q}}_2=2\mathbf{w}\cos\left(\frac{\theta}{2}\right)$*,* $\mathbf{R}^T\tilde{\mathbf{q}}_1-\tilde{\mathbf{q}}_1=2\mathbf{w}\cos\left(\frac{\pi}{2}-\frac{\theta}{2}\right)$ *and* $\tilde{\mathbf{q}}_2-\mathbf{R}^T\tilde{\mathbf{q}}_2=2\mathbf{u}\cos\left(\frac{\pi}{2}-\frac{\theta}{2}\right)$.

We can readily extend this result to the multi-class problem. For this, we let $B_1(\widetilde{\mathbf{A}})$ be the distribution of the first class and $B_j(\mathbf{R}_j^T\widetilde{\mathbf{A}}\mathbf{R}_j)$ that of the j^{th} class, where $\mathbf{R}_j$ is also defined by any two eigenvectors of $\widetilde{\mathbf{A}}$, $\mathbf{q}_{j_1}$ and $\mathbf{q}_{j_2}$ of corresponding eigenvalues $\tilde{\lambda}_{j_1}$ and $\tilde{\lambda}_{j_2}$, and we have assumed $\tilde{\lambda}_{j_1}>\tilde{\lambda}_{j_2}$. Then, the class of a new test feature vector $\mathbf{x}$ is given by

$$\arg\max_j \ |\mathbf{x}^T\mathbf{q}_{j_1}|. \tag{7}$$

We still need to derive the same classifier in the kernel space. From our discussion above, we should find the eigenvectors of the covariance matrix of the data. The covariance matrix in the kernel space is $\Sigma_j^\Phi=\Phi(\mathbf{X}_j)\Phi(\mathbf{X}_j)^T$, where $\mathbf{X}_j$ is a matrix whose columns are the sample feature vectors of class j, $\mathbf{X}_j=\left(\mathbf{x}_{j_1},\mathbf{x}_{j_2},\ldots,\mathbf{x}_{j_{n_j}}\right)$, and n_j is the number of samples in class j. This allows us to obtain the eigenvectors of the covariance matrix from $\Sigma_j^\Phi\mathbf{V}_j^\Phi=\mathbf{V}_j^\Phi\Lambda_j^\Phi$.

Unfortunately, $\mathbf{v}_{j_k}^\Phi$ may be defined in a very high-dimensional space. A usual way to simplify the computation is to employ the kernel trick [29]. To derive this solution recall we only have n_j samples in class j, which means $rank(\Lambda_j^\Phi)\leq n_j$. This allows us to write $\mathbf{V}_j^\Phi=\Phi(\mathbf{X}_j)\Delta_j$, where Δ_j is a $n_j\times n_j$ coefficient matrix, and the above eigenvalue decomposition equation can be rewritten as

$$\begin{aligned}\Phi(\mathbf{X}_j)\Phi(\mathbf{X}_j)^T\Phi(\mathbf{X}_j)\Delta_j &= \Phi(\mathbf{X}_j)\Delta_j\Lambda_j\\ \mathbf{K}_j\Delta_j &= \Delta_j\Lambda_j,\end{aligned}$$

where $\mathbf{K}_j=\Phi(\mathbf{X}_j)^T\Phi(\mathbf{X}_j)$ is the Gram matrix.

From our last result above, we directly see that $\widehat{\mathbf{V}}_j^\Phi=\Phi(\mathbf{X}_j)\Delta_j$. However, the norm of the vectors $\widehat{\mathbf{V}}^\Phi$ thus obtained is not one, but rather $\Lambda_j^\Phi=\Delta_j^T\Phi(\mathbf{X}_j)^T\Phi(\mathbf{X}_j)\Delta_j$. To obtain the (unit-norm) eigenvectors, we need to include a normalization coefficient into this result,

$$\mathbf{V}_j^\Phi=\Phi(\mathbf{X}_j)\Delta_j{\Lambda_j}^{-1/2},$$

where $\mathbf{V}_j^\Phi=\{\mathbf{v}_{j_1}^\phi,\ldots,\mathbf{v}_{j_{n_j}}^\phi\}$, $\mathbf{v}_{j_k}^\phi\in S^d$, and d is the dimensionality of the kernel space.

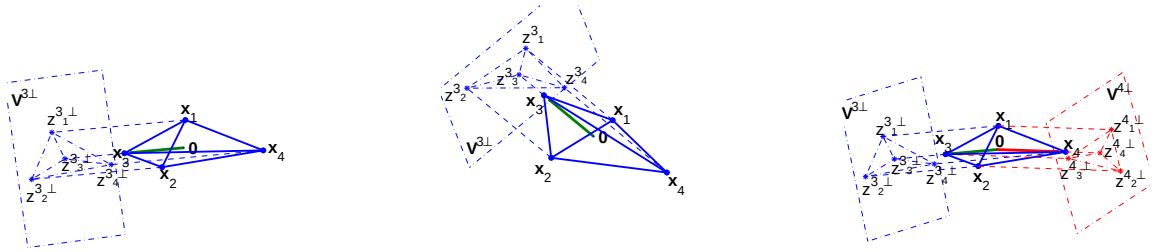
The classification scheme derived in (7) can now be extended to classify $\phi(\mathbf{x})$ as

$$\arg\max_j \ |\phi(\mathbf{x})^T\mathbf{v}_{j_k}^\phi|,$$

where the index $k=\{1,\ldots,n_j-1\}$ defining the eigenvector $\mathbf{v}_{j_k}^\phi$ must be kept constant for all j.

This final result, can be written using a kernel as

$$\arg\max_j \left|\sum_{l=1}^{n_j}\frac{k(\mathbf{x},\mathbf{x}_l)\delta_{j_k}(l)}{\sqrt{\lambda_{j_k}^\phi}}\right|, \tag{8}$$



(a) (b) (c)

Fig. 5. *In (a) we show the projection of* 4 *landmark points representing a 3D tetrahedron shape onto the nullspace of the pivot point* $\mathbf{x}_3$*. The nullspace is spanned by the column vectors of* $\mathbf{V}^{3\perp}$ *and the complex representation in the nullspace is* $\mathbf{z}^{3\perp} = (z^{3_1\perp}, \ldots, z^{3_4\perp})^T$*. As seen in (b) any rotation of the shape around its origin lead to same representation in the nullspace of the pivot point* $\mathbf{x}_3$ *up to a complex scalar rotation in the nullspace. (c) shows the shape representation obtained in the nullspaces of the pivot points* $\mathbf{x}_3$ *and* $\mathbf{x}_4$*.*

where $\Delta_j = \{\delta_{j_1}, \ldots, \delta_{j_{n_j}}\}$, and $\delta_{j_k}(l)$ is the l^{th} coefficient of the vector δ_{j_k}. The simplest algorithm that can be implemented is by assigning $k = 1$, i.e., classification based on the first basis vector of the class distributions in the kernel space. This is the approach we will take in this paper.

B. Three-dimensional shapes

Contrary to 2D shapes, the derivation of a RIK defining any arbitrary 3D rotation $\mathbf{R}$ cannot be directly done in 3D, because this requires that we compute the three-dimensional rotation between every pair of samples or perform alignment. As mentioned in the Introduction section, the LS-like procedures carry a high computational cost and are pruned to local minima and outliers. We resolve this issues by proposing an extension of the above formulation as given by two linearly independent projections of the 3D shapes. Each of this projections is defined as the null space of one of the shape feature points. We call these points *pivotal points*, since they define the 3D rotation of the shape.

Let $\mathbf{X} = \left(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^p\right)^T$ be a $p \times 3$ matrix with each row $\mathbf{x}^l \in \mathbb{R}^3$ corresponding to the mean-norm-normalized 3D location of each of the p feature points. Also, let $\mathbf{x}^q$, for some q, be the first pivot point of the 3D shape. The null space of the vector defined by this first pivot point is given by $\mathbf{V}^{q\perp} = \left(\mathbf{v}^{q1\perp}, \mathbf{v}^{q2\perp}\right)$, with $\mathbf{v}^{ql\perp}$ the l^{th} basis vector of $\mathbf{V}^{q\perp}$. Using this notation, we define the p-dimensional complex shape vector in the nullspace of the pivotal point as

$$\mathbf{z}^{q\perp} = \mathbf{X}\mathbf{v}^{q1\perp} + i\mathbf{X}\mathbf{v}^{q2\perp}, \tag{9}$$

which represents the 2D projection of the 3D shape onto $\mathbf{V}^{q\perp}$.

An example of this process is illustrated in Fig. 5(a). In this example, a 3D tetrahedron has been normalized with regard to its mean and norm. The first pivot point is $\mathbf{x}^3$ ($q = 3$) and the corresponding shape representation is obtained by projecting the 3D feature points onto $\mathbf{V}^{3\perp}$. Note that the representation introduced above already fixes one of the rotation angles of our 3D shape. This is given by the planar rotation defined by the pivotal point $\mathbf{x}^q$ and its rotated version $\mathbf{R}^T\mathbf{x}^q$ shown in Fig. 5(b).

The other rotation angle is given by the null space $\mathbf{V}^{q\perp}$. To demonstrate this effect, we use (4) to compare the representation of $\mathbf{z}^{q\perp}$ with that of its rotated version $\mathbf{z}^{q\perp}\exp(-i\theta^\perp)$, where $\theta^\perp$ is the planar rotation in the null space. Such a comparison shows that the distance between the projections of the 3D shapes is rotation invariant,

$$\|\mathbf{z}^{q\perp} - \mathbf{z}^{q\perp}\exp(-i\theta^\perp)\|^2 = 0.$$

This result demonstrates that the representation in (9) is indeed rotation invariant. However, this representation carries an ambiguity given by the 3D to 2D projection, which is not unique. In other words, different 3D objects can result in identical 2D projections. Since our projections are orthogonal, the 3D object can be uniquely determined by two views [15]. This means we can resolve the ambiguity inherent in the above formulation by simply including a

second pivotal point $\mathbf{x}^r$ that is independent from the first, i.e., $\frac{\mathbf{x}^{r\,T}\mathbf{x}^q}{\|\mathbf{x}^r\|\|\mathbf{x}^q\|} \neq 1$. Let the 2D complex vector representation in its nullspace of this second pivot point be $\mathbf{z}^{r\perp}$. This is illustrated in Fig. 5(c), where $r = 4$.

We can now construct our $2p$-dimensional shape representation by concatenating two p-dimensional 2D complex shape vectors defined in the nullspace of the pivotal points. That is,

$$\mathbf{z}^{\perp} = \begin{pmatrix} \mathbf{z}^{q\perp} \\ \mathbf{z}^{r\perp} \end{pmatrix}.$$

Using this representation we define the 3D RIK for two shape vectors $\mathbf{z}_j^{\perp}$ and $\mathbf{z}_k^{\perp}$ as

$$k(\mathbf{z}_j^{\perp}, \mathbf{z}_k^{\perp}) = \exp\left(-\frac{\|\mathbf{z}_j^{q\perp} - \mathbf{z}_k^{q\perp}\exp(-i\theta_{\mathbf{z}_j^{q\perp}\mathbf{z}_k^{q\perp}})\|^2 + \|\mathbf{z}_j^{r\perp} - \mathbf{z}_k^{r\perp}\exp(-i\theta_{\mathbf{z}_j^{r\perp}\mathbf{z}_k^{r\perp}})\|^2}{2\sigma^2}\right),$$

where $\theta_{\mathbf{z}_j^{q\perp}\mathbf{z}_k^{q\perp}}$ is the rotation angle between $\mathbf{z}_j^{q\perp}$ and $\mathbf{z}_k^{q\perp}$, and σ is the kernel parameter.

Furthermore, since $\|\mathbf{z}_1 - \mathbf{z}_2\exp(-i\theta_{\mathbf{z}_1\mathbf{z}_2})\|^2 = \mathbf{z}_1^*\mathbf{z}_1 + \mathbf{z}_2^*\mathbf{z}_2 - 2\|\mathbf{z}_1^*\mathbf{z}_2\|$, the 3D RIK can be written as

$$k(\mathbf{z}_j^{\perp}, \mathbf{z}_k^{\perp}) = \exp\left(-\frac{\mathbf{z}_j^{q\perp *}\mathbf{z}_j^{q\perp} + \mathbf{z}_k^{q\perp *}\mathbf{z}_k^{q\perp} + \mathbf{z}_j^{r\perp *}\mathbf{z}_j^{r\perp} + \mathbf{z}_k^{r\perp *}\mathbf{z}_k^{r\perp} - 2\|\mathbf{z}_j^{q\perp *}\mathbf{z}_k^{q\perp}\| - 2\|\mathbf{z}_j^{r\perp *}\mathbf{z}_k^{r\perp}\|}{2\sigma^2}\right). \tag{10}$$

Once again, we see that no pre-alignment of LS fit is needed. For 3D shapes only 6 inner products are required. Nonetheless, the kernel parameter σ in (10) or (5) still needs to be optimized. We turn to this point next.

V. Rotation Invariant Kernels

We are now concerned with the definition of a criterion that can guarantee spherical-homoscedasticity in the kernel space. The simplest and most classical approach would be to employ cross-validation (CV) as a tool to optimize the kernel parameter σ. While this is a generally affective technique when the task is classification and the training set is representative of the testing one, the approach carries several disadvantages.

First, this approach does not guarantee that the distributions will be SH in the kernel space. As shown in Section III and Fig. 3, this can result in poor classification results for *independent* testing samples. In theory, this could be resolved if we had a large representative training set. However, whereas this is possible in some applications, it is not in others. Moreover, even where this is possible, the major problem with CV is its high computational cost. Note that to tune the parameters of the kernel of our SH classifier, we need to run the CV procedure over a large range of possible values. If we search m parameters, each over n possible values, we have mn points on the solution grid. This means that for a k-fold CV procedure, we need to run our algorithm nmk times, a complexity that is impractical in most applications.

To resolve these problems, we now define an easy to optimize criterion which guarantees spherical-homoscedasticity *as well as* separability of classes. This yields comparable or superior results to CV in a fraction of the computational time.

A. SH criterion

We start by defining the criterion in the Euclidean space. The results is then extended to the kernel space using the kernel trick.

We consider two positive semi-definite parameter matrices, $\mathbf{A}$ and $\mathbf{B}$. We work with the parameter matrices, rather than their distributions, because these can be either zero-mean Normals or Binghams. Let the corresponding eigenvalues and eigenvectors of these parameter matrices be given by the spectral decompositions, $\mathbf{A} = \mathbf{V}_A\Lambda_A\mathbf{V}_A^T$ and $\mathbf{B} = \mathbf{V}_B\Lambda_B\mathbf{V}_B^T$, with $\mathbf{V}_A = (\mathbf{v}_{A_1}, \ldots, \mathbf{v}_{A_d})$, $\Lambda_A = diag(\lambda_{A_1}, \ldots, \lambda_{A_d})$, $\mathbf{V}_B = (\mathbf{v}_{B_1}, \ldots, \mathbf{v}_{B_d})$, and $\Lambda_B = diag(\lambda_{B_1}, \ldots, \lambda_{B_d})$. As shown by Theobald [30], $trace(\mathbf{AB}) \leq trace(\Lambda_A\Lambda_B)$, with the equality holding when $\mathbf{V}_A^T\mathbf{V}_B = \mathbf{I}$, i.e., the eigenvectors of $\mathbf{A}$ and $\mathbf{B}$ are not only the same but are in the same order.

Next, we divide the parameter matrices into two parts, one representing the mean direction and the other the within-class variations,

$$\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_W \quad and \quad \mathbf{B} = \mathbf{B}_1 + \mathbf{B}_W,$$

where $\mathbf{A}_1 = \mathbf{v}_{A_1}\lambda_{A_1}\mathbf{v}_{A_1}^T$, $\mathbf{B}_1 = \mathbf{v}_{B_1}\lambda_{B_1}\mathbf{v}_{B_1}^T$, $\mathbf{A}_W = \sum_{k=2}^{d}\mathbf{v}_{A_k}\lambda_{A_k}\mathbf{v}_{A_k}^T$, and $\mathbf{B}_W = \sum_{k=2}^{d}\mathbf{v}_{B_k}\lambda_{B_k}\mathbf{v}_{B_k}^T$. We note that, in this notation, $\mathbf{A}_W$ and $\mathbf{B}_W$ are the within-class scatter matrices defined in the null-space of $\mathbf{v}_{A_1}$ and $\mathbf{v}_{B_1}$. Recall that for shape distributions the first eigenvector is also know to be the Procrustes' mean.

From this notation, it follows that minimizing $trace(\mathbf{A}_1\mathbf{B}_1) = \lambda_{\mathbf{A}_1}\lambda_{\mathbf{B}_1}(\mathbf{v}_{\mathbf{A}_1}^T\mathbf{v}_{\mathbf{B}_1})^2$ will maximize the separability between classes (distributions).[3] This is thus the approach we take, since the farther apart the class distributions are, the smaller the reducible error is, as previously shown in Fig. 3.

Nonetheless, maximizing spherical-homoscedasticity will also minimize the reducible error. Therefore, our second goal is to make the class distributions as SH as possible. This we can do with the help of $trace(\mathbf{A}_W\mathbf{B}_W)$, since this is maximized when the class distributions are SH. This result is formally proven next.

Theorem 4: Let $\mathbf{A}$ and $\mathbf{B}$ be two symmetric positive semi-definite parameter matrices defining two distributions in S^{d-1}, $d > 2$. Let their within-class scatter matrices be $\mathbf{A}_W$ and $\mathbf{B}_W$ and their mean direction vectors be $\mathbf{v}_{A_1}$ and $\mathbf{v}_{B_1}$, and assume $\mathbf{v}_{A_1} \neq \mathbf{v}_{B_1}$. Then, the $trace(\mathbf{A}_W\mathbf{B}_W)$ is maximized when $\mathbf{A}$ and $\mathbf{B}$ define two SH distributions with $\Lambda_A = \Lambda_B = diag(.5, .5, 0, \ldots, 0)$ and $\mathbf{B}$ a rotated version of $\mathbf{A}$, $\mathbf{B} = \mathbf{R}^T\mathbf{A}\mathbf{R}$, where $\mathbf{R}$ defines a one-dimensional rotation along one of the eigenvectors associated to a zero eigenvalue.

Proof: As mentioned above, $trace(\mathbf{A}_W\mathbf{B}_W)$ is maximized when all the eigenvectors of $\mathbf{A}_W$ and $\mathbf{B}_W$ are identical [30]. Since

$$trace(\mathbf{A}_W\mathbf{B}_W) = \sum_{j=2}^{d}\sum_{k=2}^{d}(\mathbf{v}_{A_j}^T\mathbf{v}_{B_k})^2\lambda_{A_j}\lambda_{B_k}, \tag{11}$$

the maximum is given by $m = \sum_{j=2}^{d}\lambda_{A_j}\lambda_{B_j}$, which is achieved when the eigenvectors are the same. Recall that the summing terms start at the second eigenvector, since the first eigenvectors are the mean directions used to define $\mathbf{A}_1$ and $\mathbf{B}_1$.

Here, the values of m cannot be attained, because $\mathbf{v}_{A_1} \neq \mathbf{v}_{B_1}$, which implies that there is a rotation between $\mathbf{A}_W$ and $\mathbf{B}_W$. Therefore, at least one eigenvector of $\mathbf{A}_W$ must be different from one eigenvector of $\mathbf{B}_W$. If we are to maximize (11), only one eigenvector from each within-class scatter matrix can be different from the other, and these need to be the ones associated to the smallest eigenvalue. Without loss of generality, let us assume these are the last eigenvectors of $\mathbf{A}_W$ and $\mathbf{B}_W$, $\mathbf{v}_{A_d}$ and $\mathbf{v}_{B_d}$. Then, we can write

$$trace(\mathbf{A}_W\mathbf{B}_W) = \sum_{j=2}^{d-1}\lambda_{A_j}\lambda_{B_j} + (\mathbf{v}_{A_d}^T\mathbf{v}_{B_d})^2\lambda_{A_d}\lambda_{B_d}. \tag{12}$$

The spherical property of our distributions tells us that $trace(\mathbf{A}) = trace(\mathbf{B}) = 1$. This property allows us to write the eigenvalues as $\lambda_{A_j} = (1-\lambda_{A_1})c_{A_j}/\sum_{j=2}^{d}(c_{A_j})$, with $c_{A_2} \geq c_{A_3} \geq \cdots \geq c_{A_d} \geq 0$. Similarly, $\lambda_{B_j} = (1-\lambda_{B_1})c_{B_j}/\sum_{j=2}^{d}(c_{B_j})$ with $c_{B_2} \geq c_{B_3} \geq \cdots \geq c_{B_d} \geq 0$. Substituting this in (12), we get

$$\begin{aligned}\frac{\sum_{j=2}^{d-1}(1-\lambda_{A_1})(1-\lambda_{B_1})c_{A_j}c_{B_j}}{(\sum_{j=2}^{d}c_{A_j})(\sum_{j=2}^{d}c_{B_j})} &+ (\mathbf{v}_{A_d}^T\mathbf{v}_{B_d})^2(1-\lambda_{A_1})(1-\lambda_{B_1})\frac{c_{A_d}c_{B_d}}{(\sum_{j=2}^{d}c_{A_j})(\sum_{j=2}^{d}c_{B_j})} \\ &\leq (1-\lambda_{A_1})(1-\lambda_{B_1}),\end{aligned}$$

with the equality holding when $c_{A_2} > 0$, $c_{B_2} > 0$ and $c_{A_j} = c_{B_j} = 0$, $\forall j > 2$. This means that the maximum of the $trace(\mathbf{A}_W\mathbf{B}_W) = (1-\lambda_{A_1})(1-\lambda_{B_1})$. The maximum is thus given by the smallest possible λ_{A_1} and λ_{B_1}. Since $\lambda_{A_1} \geq 1-\lambda_{A_1}$ and $\lambda_{B_1} \geq 1-\lambda_{B_1}$, the maximum of $trace(\mathbf{A}_W\mathbf{B}_W) = .25$, which is given when $\lambda_{A_2} = \lambda_{B_2} = \lambda_{A_1} = \lambda_{B_1} = .5$. ■

The result presented above implies that the two distributions that maximize the trace of their within-class scatter matrices are uniformly distributed along two eigenvectors (i.e., they have the same variance along these two dimensions). The important point is to realize that if $\mathbf{v}_{A_1} \neq \mathbf{v}_{B_1}$, the two dimensions in S^{d-1} where the first distribution is described are different to those of the second distribution. Hence, the two distributions are readily separated by a single hyperplane, which means there will not be any classification error.

This result is illustrated in Fig. 6. In this example, we have two distributions on S^2 with identical eigenvalue matrices, $\Lambda_A = \Lambda_B = diag\,(.5, .5, 0)$. Here, the within-class scatter matrix of the second distribution is a rotated

[3]This argument was derived for the generalized eigenvalue decomposition in Theorem 2 in [24].

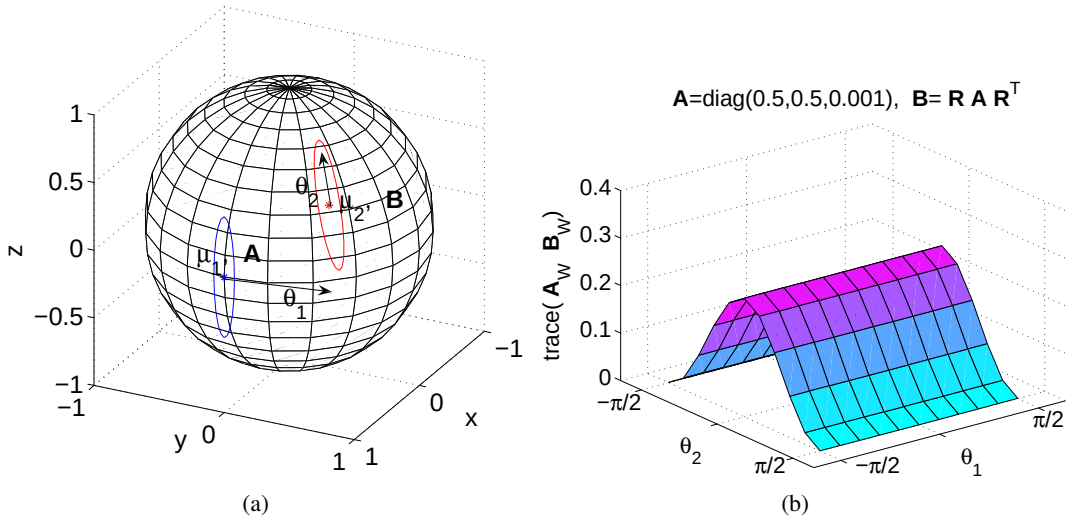


Fig. 6. (a) Shown here are two distributions with identical parameter matrices, $\mathbf{A}$ and $\mathbf{B}$, up to rotation. This means that the eigenvalues of the parameter matrices are the same. In this example, we have set them to $\Lambda_A = \Lambda_B = diag(.5, .5, 0)$. The rotation between the two parameter matrices shown in this illustration is given by $\theta_1 = \pi/4$ along the third eigenvector and $\theta_2 = \pi/6$ along the second. In (b) we show the value of the $trace(\mathbf{A}_W\mathbf{B}_W)$ for each of the possible values of θ_1 and θ_2.

version of the first one, $\mathbf{B}_W = \mathbf{R}^T\mathbf{A}_W\mathbf{R}$, where $\mathbf{R}$ is defined by two angles, θ_1 and θ_2, which specify the rotation about $\mathbf{v}_{A_3}$ and $\mathbf{v}_{A_2}$, respectively, Fig. 6(a). Fig. 6(b) shows the value of the $trace(\mathbf{A}_W\mathbf{B}_W)$ for the possible range in θ_1 and θ_2. As demonstrated in Theorem 4, rotating the second distribution along the eigenvector associated to the smallest eigenvalue maintains the maximum value for the trace, .25. Whereas rotating the second distribution along the other eigenvector of $\mathbf{A}_W$, results in a decrease of the trace.

In the above example, it is important to note that the $trace(\mathbf{A}_W\mathbf{B}_W) = .25$ for any separation of the two distributions – from 0^o to 90^o. As already mentioned, in practice, one wishes to keep the two distributions as apart as possible, since these represent different classes. This can be achieved by minimizing the $trace(\mathbf{A}_1\mathbf{B}_1)$. Our criterion thus simplifies to maximizing the $trace(\mathbf{A}_W\mathbf{B}_W)$ for the range of values where the $trace(\mathbf{A}_1\mathbf{B}_1)$ is minimum. We refer to this approach as the SH criterion.

B. A criterion for the kernel space

If we are to use the result presented in the preceding section in our problem, we need to redefine our criteria, $trace(\mathbf{A}_W\mathbf{B}_W)$ and $trace(\mathbf{A}_1\mathbf{B}_1)$, in the kernel space. Let these be $trace(\mathbf{A}_W^\phi\mathbf{B}_W^\phi)$ and $trace(\mathbf{A}_1^\phi\mathbf{B}_1^\phi)$.

To make our point clearer, we start with a simple illustrative example, Fig. 7. In this example, we have data sampled from two von Mises Fisher (vMF) distributions $M((0,1),20)$ and $M((cos(\theta), sin(\theta)),10)$. [4] The mean direction vector of the second distribution is given by $\theta = \pi/4$ in Fig. 7(a), $\theta = \pi/2$ in Fig. 7(c), and $\theta = \pi$ in Fig. 7(e). Fig. 7(b,d,f) shows the corresponding plots of the criteria $trace(\mathbf{A}_W^\phi\mathbf{B}_W^\phi)$ and $trace(\mathbf{A}_1^\phi\mathbf{B}_1^\phi)$ for different values of σ when using the RIK defined as in Eq. (5).

As argued in the previous section, the behavior of the $trace(\mathbf{A}_W^\phi\mathbf{B}_W^\phi)$ is (approximately) concave. In contrast, the criterion $trace(\mathbf{A}_1^\phi\mathbf{B}_1^\phi)$ results in a monotonically increasing function. This is also to be expected, because the more apart the two distributions are, the better.

The close to concave behavior of the $trace(\mathbf{A}_W^\phi\mathbf{B}_W^\phi)$ can be formally justified as follows. Let $\mathbf{A}_W^\phi = \sum_{j=2}^{r_A}\mathbf{v}_j^\phi\lambda_{A_j}^\phi\mathbf{v}_j^{\phi^T}$ and $\mathbf{B}_W^\phi = \sum_{j=2}^{r_B}\mathbf{u}_j^\phi\lambda_{B_j}^\phi\mathbf{u}_j^{\phi^T}$, where r_A and r_B are the corresponding ranks of the two within-class scatter matrices. Using this notation, we have

$$trace(\mathbf{A}_W^\phi\mathbf{B}_W^\phi) = \sum_{j=2}^{r_A}\sum_{k=2}^{r_B}(\mathbf{v}_j^{\phi^T}\mathbf{u}_k^\phi)^2\lambda_{A_j}^\phi\lambda_{B_k}^\phi.$$

[4]The density function for a vMF distribution $M(\mu,\kappa)$ is $f(\mathbf{x}) = C_M\exp(\kappa\mu^T\mathbf{x})$ where μ is the mean direction vector and κ is the concentration parameter.

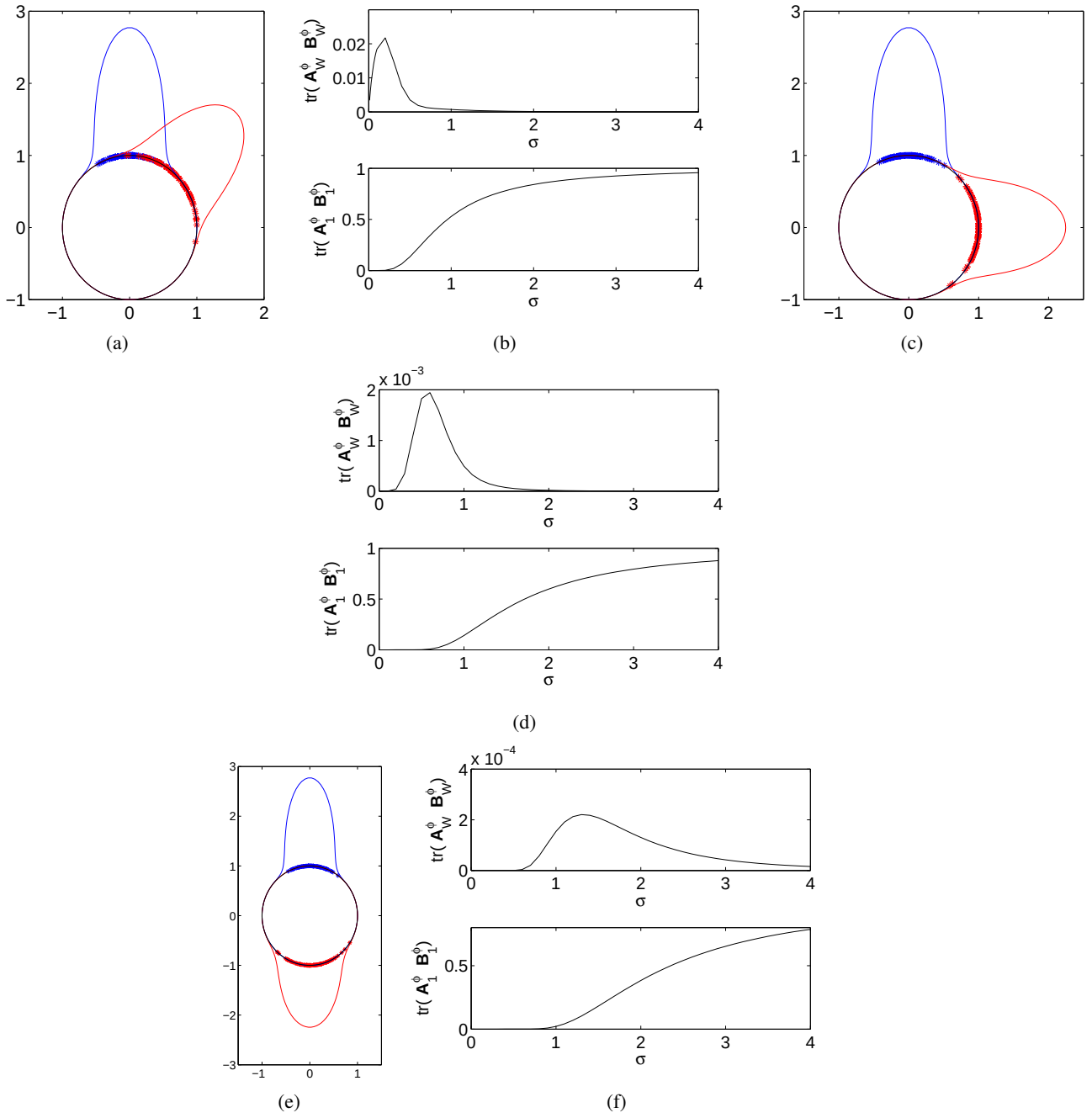


Fig. 7. In (a,c,e) we show two vMF distributions $M((0,1),20)$ and $M(cos(\theta),sin(\theta),10)$, with $\theta = \{\pi/4, \pi/2, \pi\}$. In (b,d,f) we plot the corresponding values of the criteria $trace(\mathbf{A}_W^\phi \mathbf{B}_W^\phi)$ and $trace(\mathbf{A}_1^\phi \mathbf{B}_1^\phi)$ for different values of σ when using the RIK given in (5).

From (5), we know that as σ decrease, the distance between different samples goes to zero, $k(\mathbf{x},\mathbf{y}) = 0$. This means that, in the kernel space, there is no correlation between the eigenvectors of $\mathbf{A}_W^\phi$ and $\mathbf{B}_W^\phi$, and $trace(\mathbf{A}_W^\phi \mathbf{B}_W^\phi) = 0$. Conversely, as σ approaches ∞, the distributions become concentrated along the first eigenvector, i.e., the eigenvalues $\lambda_{A_j}^\phi$ and $\lambda_{B_j}^\phi$ approach 0 for all $j > 1$. This directly implies $trace(\mathbf{A}_W^\phi \mathbf{B}_W^\phi) \to 0$. What interests us is the cases in between, where the $trace(\mathbf{A}_W^\phi \mathbf{B}_W^\phi)$ takes positive values.

Similarly, as σ approaches zero, the distributions become distant in the kernel space and, hence, the $trace(\mathbf{A}_1^\phi \mathbf{B}_1^\phi) = 0$. On the other hand, as σ increases the distance between the first eigenvectors decreases, resulting in larger $trace(\mathbf{A}_1^\phi \mathbf{B}_1^\phi)$. Here, we are interested in the former case, where the $trace(\mathbf{A}_1^\phi \mathbf{B}_1^\phi) = 0$. Hence, our approach is

to select that value of σ that maximizes the $trace(\mathbf{A}_W^\phi \mathbf{B}_W^\phi)$ within the range of σ yielding $trace(\mathbf{A}_1^\phi \mathbf{B}_1^\phi) = 0$.

The above defined procedure is only applicable to the two-class problem. A natural way to extend this type of algorithms to the multiclass setting is to use the 1-versus-1 approach, which is known to be a robust and efficient technique [7]. In our algorithm, this means that a different σ will be selected for each class pair. Each of these $C(C-1)/2$ possible σ will define a classifier that separates class j from class k. Then, to classify a test sample, we use all the $C(C-1)/2$ classifiers. Each of these classifiers will classify the test sample in a class, which is interpreted as a vote for that class. At the end, the class with most votes is selected as the classification result. This algorithm is summarized in Algorithm 2.

Algorithm 2 : *RIK*

Normalize the shapes, $\mathbf{z}_{jk} = \mathbf{u}_{jk} - \bar{\mathbf{u}}_{jk}/\|\mathbf{u}_{jk} - \bar{\mathbf{u}}_{jk}\|$, where $\mathbf{u}_{jk}$ is the k^{th} sample of class j.

Let $\mathbf{Z}_j = \left(\mathbf{z}_{j1}, \ldots, \mathbf{z}_{jn_j}\right)$, where n_j is the number of samples in class j.

Select the σ_{ab} maximizing $trace(\mathbf{A}_W^\phi \mathbf{B}_W^\phi)$ and satisfying $trace(\mathbf{A}_1^\phi \mathbf{B}_1^\phi) = 0$ for each class pair a, b. Here, $\mathbf{A}^\phi$ is the sample covariance matrix of the a^{th} class and $\mathbf{B}^\phi$ that of the b^{th} class.

The kernel matrix $\mathbf{K}_j = \Phi(\mathbf{Z}_j)^T \Phi(\mathbf{Z}_j)$ for 2D shapes is given by

$k(\mathbf{z}_{jk}, \mathbf{z}_{jl}) = \exp\left(-\frac{2-2\|\mathbf{z}_{jk}^* \mathbf{z}_{jl}\|}{2\sigma^2}\right)$, and for 3D shapes is given by

$k(\mathbf{z}_{jk}^\perp, \mathbf{z}_{jl}^\perp) = \exp\left(-\frac{\mathbf{z}_{jk}^{q\perp *}\mathbf{z}_{jk}^{q\perp} + \mathbf{z}_{jl}^{q\perp *}\mathbf{z}_{jl}^{q\perp} + \mathbf{z}_{jk}^{r\perp *}\mathbf{z}_{jk}^{r\perp} + \mathbf{z}_{jl}^{r\perp *}\mathbf{z}_{jl}^{r\perp} - 2\|\mathbf{z}_{jk}^{q\perp *}\mathbf{z}_{jl}^{q\perp}\| - 2\|\mathbf{z}_{jk}^{r\perp *}\mathbf{z}_{jl}^{r\perp}\|}{2\sigma^2}\right)$.

Let $\mathbf{K}_j \delta_{j_1} = \delta_{j_1} \lambda_{j_1}^\phi$, where δ_{j_1} is the eigenvector associated to the largest eigenvalue $\lambda_{j_1}^\phi$.

Let $\arg_{j\in\{a,b\}} \max \left| \sum_{l=1}^{n_j} \frac{k(\mathbf{z},\mathbf{z}_l)\delta_{j_1}(l)}{\sqrt{\lambda_{j_1}^\phi}} \right|$ provide the vote for each class pair when classifying a test sample $\mathbf{z}$.

Classify the test sample $\mathbf{z}$ to the class with most votes.

VI. Experimental results

In the following we provide extensive comparative results using a variety of 2D and 3D shape datasets. In our comparative studies, we use the two approaches presented in this article. The first approach uses the complex Normal approximation defined in Section III. This approach is very fast but assumes the data is SH and, hence, will only work in a number of datasets. The second approach is the kernel SH algorithm defined in Section IV. This alternative requires that we estimate the kernel parameter σ. We provide comparative results between cross-validation (CV) and the SH criterion defined in Section V. We demonstrate that the two optimization methods lead to similar results, but that the CV method requires of at least ten times the computational time used by the SH criterion.

A. Algorithm Overview

Before we provide the comparison results described above, we will introduce a variety of alternative algorithms that have been presented in the literature or can be easily derived from the theory presented in this paper. This is in addition to the two methods we have already derived in the preceding section, which were given in Algorithms 1 and 2.

The first algorithm we will employ to provide comparative results is that given by simply using Procrustes alignment procedures. Procrustes alignment of two shapes determines the translation, scaling and rotation that minimize the LS error between two shapes [5]. After the Procrustes alignment, we can either use the nearest mean (NM) or the nearest neighbor (NN) classifier. In our results, we will refer to these two methods as *Proc. NM* and *Proc. NN*, respectively. Another option in Procrustes analysis is to first compute the mean feature vector of each class and then project the aligned data onto its tangent space. In this tangent space, the projected data can be modeled using a Gaussian distribution and, hence, classification is given by maximum likelihood. We call this method *Proc. TS*, for Procrustes Tangent Space. This algorithm is very similar to Kent's hybrid model [14], which uses the same procedure but first scales the data to have unit norm. Therefore, we also provide comparison results with Kent's hybrid model.

(a)

(b)

Fig. 8. (a) Shown here is an example image for each of the eight categories in the ETH-80 database. (b) The ten objects in the category cars.

TABLE I

THE AVERAGE RECOGNITION RATES OBTAINED ON THE ETH-80 DATASET USING THE LEAVE-ONE-OBJECT-OUT.

	Proc. NM	Proc. NN	Proc. TS	Kernel Proc.	*Kents' Hybrid*	*complex Bingham*	*complex Normal*	RIK_{CV}	*RIK*
Recognition Rate	79.02	82.10	79.12	86.34	79.66	86.95	87.5	91.22	**92.29**
Training Time (in seconds)	0.21	N/A	1.25	1680.4	1.06	18.34	**0.95**	3049.8	89
Testing Time (in seconds)	**0.01**	7.05	0.16	2.16	0.05	**0.02**	**0.02**	3.07	3.50

The results in the *complex Normal* column correspond to the solution given by the approach presented in Section III, RIK_{CV} are the results obtained with cross-validation and the kernel SH method presented in Section IV, and *RIK* are the results obtained with the proposed kernel method and the SH criterion introduced in Section V.

Finally, we realize that we could also define a transformation invariant kernel in the original space, similar to what was done in [32], [9]. To achieve this, we can use the same kernels we have derived in this paper, but apply them to the original data. The disadvantage is that we do not have a criterion for selecting the kernel parameter σ. This forces us to employ the classical CV approach, which is computationally expensive. We will call this last method *Kernel Proc.*, because it corresponds to Procrustes analysis in the kernel space. In this case, we also employ the NM classifier.

B. Object categorization

The first problem we study is that of recognizing the category of an object by means of its 2D shape. For this purpose, we use the shape vectors of the eight categories in the ETH-80 dataset [19]. These object classes (categories) are: apples, pears, tomatoes, cows, dogs, cars, cups and pears. Each of these categories contains 10 different objects (e.g., 10 cars) photographs at 41 orientations, Fig. 8. The contour shape of each object silhouette is sampled with 100 equidistant points, providing the shape vectors $\mathbf{z}_j$, $j = \{1, \ldots, 3280\}$. Recall that each feature vectors $\mathbf{z}_j$ has been mean-norm-normalized.

We used the leave-one-object-out procedure for testing. This means that, at each iteration, we select the 41 images of one of the objects as members of the test set and use the rest for training. This process is repeated 80 times, one for each of the objects that can be left out.

The results of the proposed approach are in Table I. In this table, we provide the results obtained with the SH methods presented in Sections III-V. We also include the results obtained with the algorithms defined in Section VI-A and with the use of the Bingham distribution. To estimate the parameters of the Bingham, we employed the saddle point approximation algorithm of Kume and Wood [17]. In this experiment, we further aided Kume and Wood's optimization algorithm by removing the noisy bases of the data by keeping 99% of the data variance as provided by a principal components analysis (otherwise the algorithm did not converge).

As expected the proposed approach, which uses a Rotation Invariant Kernel (RIK), outperforms the rest. The complex Normal approximation is comparable to the use of Binghams and is slightly better than Kent's hybrid model. The main advantage of the Normal approximation is its computational time, which results (on average)

TABLE II
COMPARATIVE RESULTS USING A SINGLE CUE ON THE ETH-80 DATASET

	RIK	Color	D_xD_y	Mag-Lap	PCA Masks	PCA Gray
Recognition Rate	**92.29**	64.85	79.79	82.23	83.41	82.99
	SC Greedy	SC+DP	MDS+SC+DP	IDSC+DP	LDA_{shape}	SDA_{shape}
Recognition Rate	86.40	86.40	86.80	88.11	59	75

The comparative results are from [19], [20], [35]. *Color* corresponds to the classification obtained from a color histogram of the data, D_xD_y uses the histogram of the Gaussian filtered images, *Mag-Lap* uses the magnitude of the image responses of the Gaussian filtered and the Laplacian images, *PCA-Masks* and *PCA Gray* are given by the PCA algorithm, and SC Greedy and DP correspond to the Shape Context with greedy and Dynamic Programming approaches, respectively. The rest of the algorithms all use the 2D shape of the objects. *MDS+SC+DP* combines the inner distance and multidimensional scaling, building the shape context on the signatures and using dynamic programming to match shapes [20]. *IDSC+DP* uses the inner distance with shape context and dynamic programming for matching [20]. And LDA and SDA are the classification results of obtained with Linear Discriminant Analysis (LDA) and Subclass Discriminant Analysis (SDA) [35]. All the algorithms use the nearest neighbor classifier except for the RIK which uses the nearest mean direction vector.

TABLE III
AVERAGE RECOGNITION RATES FOR THE COIL-100 DATASET

	Proc. NM	Proc. NN	Proc. TS	Kernel Proc.	*Kents' Hybrid*	*complex Bingham*	*complex Normal*	RIK_{CV}	*RIK*
Recognition Rate	72.19	94.76	88.12	94.34	90.47	91.75	**95.47**	**95.64**	**95.82**
Training Time (in seconds)	5.96	N/A	7.27	1099	6.12	109.12	**4.85**	10800	1298
Testing Time (in seconds)	**2.97**	57.67	28.74	547.37	25.01	**2.99**	**2.98**	792	780

in the processing of 50 frames per second.[5] The Procrustes analysis algorithms provide a comparable or inferior performance to that of the Normal approximation.

The two RIK algorithms tested use the cross-validation (CV) test or the SH criterion to optimize the kernel parameter. We see that, indeed, the proposed SH criterion slightly outperforms the CV solution, while reducing the training time by a factor of 34. The two implementations will generally result in different values for σ. In this particular case, the RIK_{CV} implementation resulted in an average $\sigma = 0.083$ and the SH criterion in RIK in an average $\sigma = 0.048$. In this case, the kernel Procrustes algorithm results in a much lower classification than those obtained with the proposed methods.

In Table II we provide comparative results between the proposed RIK algorithm and those obtained using other shape descriptors and a variety of image cues. We see that the proposed shape representation outperforms the other shape algorithms as well as the other feature representations.

C. Object recognition

Our second experiment utilizes the COIL-100 object dataset [26]. This database consists of 100 object classes, each containing 72 images. The 72 images are obtained by photographing each object at intervals of 5 degrees apart. Our test uses a 9-fold cross-validation approach. This means that the set of samples in each object class is randomly divided into 9 groups. At each iteration, one of these groups is used for testing, while the rest are combined to create the training set. This is repeated for each of the 9 possible groups that can be used for testing.

The average recognition rates for each of the nine algorithms described above are in Table III. As in our first experiment, *Proc. NM*, *Proc. TS* and Kent's hybrid model provide the lowest classification accuracy, followed by the approximation to the complex Bingham. In this application, the results obtained with the complex Normal algorithm are comparable to those of the kernel extension (RIK). This is because, in this particular application, the assumption of spherical-homoscedasticity provides an appropriate assumption for the class distributions in COIL. These results are slightly superior to those given by *Proc. NN* and (the nonlinear) *Kernel Proc*. The main advantage

[5]The computational times we provide in our tables are in seconds and were obtained with a Matlab® implementation of the algorithms running in a Pentium 4 at 3GHz.

TABLE IV
COMPARATIVE RESULTS ON THE COIL-100 DATASET

Algorithm	Recognition Rate
RIK_{shape}	85.87
$RIK_{shape\&edge}$	**92.87**
SNoW$_{pixel}$	81.46
Linear SVM$_{pixel}$	78.50
NN$_{pixel}$	74.63
SNoW$_{edge}$	88.28
MSER+LAF+tree	**98.2**

NN refers to the nearest neighbor algorithm. SNoW stands for Sparse Network of Winnows algorithm. SVM is a linear Support Vector Machine. MSER+LAF+tree refers to maximally stable extremal regions and local affine frames learned using a tree structure [27]. SNoW$_{pixel}$, Linear SVM$_{pixel}$, Nearest Neighbor$_{pixel}$, SNoW$_{edge}$ are from [34], with Algorithm$_{pixel}$, Algorithm$_{edge}$ and Algorithm$_{shape}$ referring to the use of appearance (i.e., pixel information only), edges and shape features. $RIK_{shape\&edge}$ combines the use of shape as described in this article and that given by the edges of the image. Here, SNoW uses the network of linear functions, SVM uses the support vector classifier, MSER+LAF+tree uses a tree, NN uses nearest neighbor and RIK uses the nearest mean direction vector for classification.

of the complex Normal approach is its computational simplicity, which provides the lowest training and testing times while preserving the highest classification accuracy.

As above, we see that the time required to estimate the kernel parameter with our SH criterion is about 20 times faster than using CV. Here too, the kernel parameters obtained with these two techniques are different. When using CV, we have an average $\sigma = 0.0152$. With the SH criterion the average $\sigma = 0.0407$.

In some instances, we would like to train our system using a smaller number of training samples. In such a case, shape descriptions will not suffice, because there will not be enough information to fully estimate the underlying distribution of each class. Yang et al. [34] and Obdrzálek and Matas [27] provide state of the art results on the COIL-100 database using a variety of algorithms trained with only 4 object views and tested on the remaining 68 views. Table IV summarizes the results of [34], [27] and those obtained using the RIK algorithm introduced in this paper. In this case, we provide classification accuracies given by RIK with shape descriptions and shape + edge information. Edges are given by the Laplacian operator. As we can see, object edges provide the necessary information to make them comparable to those of the state of the art.

D. Three-dimensional Face Recognition

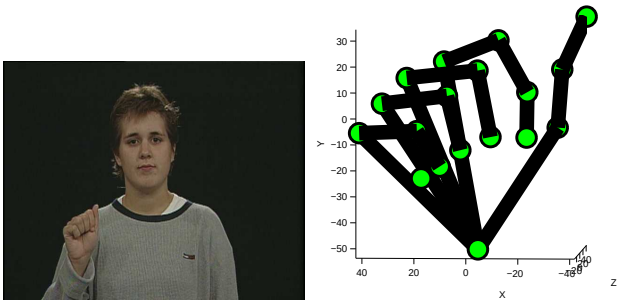
We further tested our algorithms using the target set range images in the FRGC version 2 dataset [28]. This set includes 3D range scans of $4,007$ faces. From this set we used all of the classes that have at least eighteen images, since the proposed algorithms require of a sufficiently large number of samples per class to estimate the data distribution or kernel representations. This corresponds to a total of 869 samples associated to 44 distinct classes. As in the previous 2D shape analysis examples, these 3D range scans require of a normalization with regard to their mean and variance. Such a normalization is used to eliminate the translation and scale variations associated to each image.

In this test, we run a leave-10-out test. The average recognition rates obtained from 80 runs of each algorithm are shown in Table V. The major problem with this dataset is the large dimensionality of the data. This makes it difficult for the distribution-based algorithms, which would generally require of a larger number of training samples per class. Nevertheless, the proposed RIK only requires of an estimate of the kernel space, which (as shown by many other results in machine learning and pattern recognition) can be achieved from a smaller set. This fact makes the RIK approach the preferred solution in this experiment. In fact, we note that these results are comparable to those obtained with alternative algorithms specifically designed for face recognition in FRGC [12], [6]. A direct comparison with such alternate algorithms is however not possible, because our algorithm still requires of a higher than available sample-to-dimensionality ratio.

TABLE V

THE AVERAGE RECOGNITION RATES OBTAINED ON THE FRGC DATASET USING 10-FOLD CROSS-VALIDATION

	Proc. NM	Proc. NN	Proc. TS	Kernel Proc.	*Kents' Hybrid*	*complex Bingham*	*complex Normal*	RIK_{CV}	*RIK*
Recognition Rate	47.50	68.90	46.75	93.25	42.28	9.33	41.73	**94.78**	**94.13**
Training Time (in seconds)	36.97	N/A	1.34	40.38	5.60	70.33	5.60	2594.4	263.4
Testing Time (in seconds)	2.97	35.53	**0.02**	0.09	1.86	0.11	0.46	**.01**	**.01**



(a) (b)

Fig. 9. (a) Shown here is one of the images of a video sequence of an ASL sign. (b) The 20 landmark points representing the reconstructed 3D handshape.

E. Handshape Recognition

Handshape recognition is a growing area of interest in computer vision with potential applications in a variety of problems, e.g., in human-computer interaction systems and for the recognition of sign languages. In our next experiment, we will concentrate with the latter application.

Sign languages are the primary means of communication of the deaf around the world. The phonology, morphology, syntax, semantics and pragmatics components are encoded in a range of facial, body and hand configurations. To design systems that can interpret or generate signs, we must be able to model and recognize the handshapes.

Here, we test the classification performance of our algorithm on 19 different 3D handshapes. Each of these handshapes is signed by 10 subjects, resulting in a total 190 shapes. The 3D handshapes were obtained from the Purdue ASL database [22] using the method described in [4]. In this approach, each handshape is represented by 20 landmark points, as shown in Fig. 9.

To gain invariance to 3D translation and scale, we perform a mean and variance normalization. Then, we used the methods presented in this paper to represent and classify shapes. For testing, we used a leave 3-subject out test, where the handshapes of 7 subjects are used for training and those of the remaining 3 subjects (i.e., 57 instances) are employed for testing. The average recognition rates are shown in Table VI. The kernel parameters in RIK_{CV} is also optimized using a leave 3-subject out strategy on the training set. In this case, the algorithm used to optimize the parameters of the complex Bingham did not converge to a good local minimum, resulting in poor classification results. Also consistent with our previous results and our theory, the RIK approach achieves comparable (or better) classification results than RIK_{CV} but with a fraction of the time (~ 0.045 of the time). In this case, we see that *Proc. NM* provides comparable results to RIK. This demonstrates that (LS) Procrustes alignment results in good estimates under some conditions, but not others (since our previous results with this method were among the lowest).

TABLE VI

AVERAGE RECOGNITION RATES OBTAINED ON THE 3D HANDSHAPE DATASET

	Proc. NM	Proc. NN	Proc. TS	Kernel Proc.	*Kents' Hybrid*	*complex Bingham*	*complex Normal*	RIK_{CV}	*RIK*
Recognition Rate	**93.45**	5.36	63.71	49.12	54.24	26.68	69.48	90.22	**92.03**
Training Time (in seconds)	2.41	N/A	0.15	15.54	0.11	22.12	0.08	1426	64.93
Testing Time (in seconds)	1.48	10.37	**0.01**	0.09	**0.01**	**0.01**	**0.01**	2.09	2.09

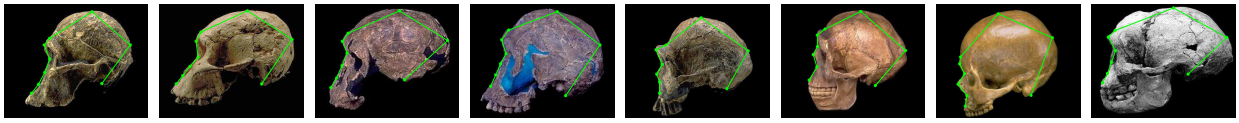


Fig. 10. Nine landmark points shown on the skulls of STS 5, STS 71, KNM ER 1470, 1813, KNM ER 3733, Peking man, cro-magnon and LB1 respectively.

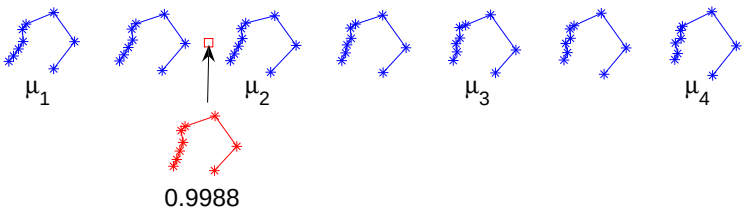


Fig. 11. Shape model describing the morphological transitions between species of *Australopiths* and *Homo*. The mean shapes, $\mu_1, \ldots, \mu_4$ correspond to the following species: *A. africanus*, *early Homo*, *H. erectus* and *H. sapiens*. The shape of LB1 is most similar to the position marked by the square in the figure (with a correlation on .9988). This corresponds to a specimen in transition between genera, with highest resemblance to *early Homo*.

F. Shape models

As mentioned earlier in this paper, in many applications we are not only interested in classifying shapes, but also in creating models that describe the shape variations of a given set of objects. One recent application is the classification of specimens in anthropology and paleontology [23], [33]. In this application, we are interested in analyzing and modeling and classifying newly discovered fossils into a continuum. Statistical shape models can aid in this classification by providing the best fit of the new fossil within a shape model describing within and between class variations. As an example, we apply the shape modeling described in this paper to find the best match classification for LB1, a specimen discovered in 2003 in the island of Flores, Indonesia [1]. While the original team classified this specimen as the member of a previously unknown species of *Homo*, *H. floresiensis*, retractors argue that it is a deformed *H. sapiens*. To resolve this problem, we can construct a shape model defining the morphological variations undergone by a large variety of species along the evolutionary tree [23]. Then, the classification task reduces to finding the best fit for LB1 within this shape model.

To construct such a model we collected the images of a profile view for two specimens of *A. africanus* (STS 5 and STS 71), *early Homo* (KNM ER 1470 and 1813), *H. erectus* (KNM ER 3733 and Pekin man), and two *H. sapiens* (cro-magnon 1). Fig. 10 shows the 9 landmarks we have used to shape of the skull.

We model each class (*A. africanus*, *early Homo*, *H. erectus* and *H. sapiens*) with the complex Normal model defined in this paper. The Procrustes mean shape is given by their first eigenvectors, while the morphological path between class means estimated as the shortest path on the hypersphere. Fig. 11 shows the class mean direction vectors, $\mu_1, \ldots, \mu_4$, corresponding to each of the four classes. The shapes between classes are the mid points when moving from one class to the other in our shape model. We now searched for the highest correlation point between the shape model and the shape of LB1. The highest correlation point (i.e., magnitude of the cosine distance) is .9988, which is shown in Fig. 11 as a square. Therefore, our shape model identifies LB1 as an early member of *early Homo*. These results are consistent with the ones reported in [23], which were independently obtained using a different approach.

VII. Conclusion

Defining algorithms that provide invariance to translation, scale and rotation is essential for the efficient analysis and classification of shapes. While Procrustes-based methods have been attractive, those based on the properties provided by complex scalar-rotation-invariant distributions defined on the complex hypersphere (e.g., the complex Bingham distribution) are generally preferred. Unfortunately, the non-linearity associated with the parameter estimation of these distributions has made this approach unattractive, because their parameters either cannot be calculated

or the optimization procedures defined to estimate them are time consuming and only converge to local minima that are not guarantee to provide a good solution.

In the present paper, we have first formally shown that one can substitute the estimate of the complex Bingham distribution with that of the complex Normal whenever the distributions defining the classes are spherical-homoscedastic (SH), i.e., rotated versions of one another. This substitution guarantees a Bayes optimal classification of the shape, guaranteing no loss in classification accuracy. Moreover, SH shapes transform an inherently non-linear problem into a simple to tackle linear one.

However, the interchange of complex Binghams for complex Normals is not guaranteed to yield accurate results when the distributions are not SH. When this is not the case, we need to first map our original shape distributions to a space where they become SH. In order to define this space without the need to increase the dimensionality of the original representation, we have employed the well-known idea of the kernel-trick. Therefore, in our approach, one first needs to find that kernel that transforms the shape representation into a SH one and then employ complex Normal distribution to optimally (wrt Bayes) represent and classify shapes.

In the above defined approach, the *key* concept is to define a kernel map that provides the much needed rotation-invariance for representing shapes. This can be achieved by defining a kernel map which provide the same projection for a feature vector $\mathbf{z}$ and its rotated version $\mathbf{z}e^{i\theta}$, $\forall\theta \in [0, 2\pi]$. We referred to this type of kernel as *Rotation Invariant Kernels* (RIK).

A main advantage of the RIK approach is that it can be used for 2D and 3D shapes and, theoretically, to any other dimensionality. Therefore, the approach introduced in this article solves two of the major problems associated with the statistical representation of shapes – the estimation of complex spherical distributions and their limitedness to work with 2D shapes.

We have provided several experimental results for the representation and classification of 2D and 3D shapes. Our experimental results demonstrated that the proposed approach provides superior classification results to those defined in the literature. We have also illustrated how the proposed approach can be employed to build models of shape. One requirement for our approach is that the number of samples per class has to be sufficient to provide a reasonable estimate of the parameters of the underlying pdf.

VIII. Acknowledgments

This research was partially supported by the National Institutes of Health under grant R01 DC 005241 and by the National Science Foundation under grant IIS 0713055.

References

[1] P. Brown, T. Sutikna, M. J. Morwood, R. P. Soejono, Jatmiko, E. W. Saptomo, and R. A. Due. A new small-bodied hominin from the late pleistocene of flores, indonesia. *Nature*, 431:1055–1061, 2004.

[2] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[3] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251–280, 1990.

[4] L. Ding and A. M. Martinez. Recovering the linguistic components of the manual signs in american sign language. In *Proceedinggs of IEEE Conference on Advanced Video and Signal-based Surveillance (AVSS), London (UK)*, 2007.

[5] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, West Sussex, England, 1998.

[6] T. Faltemier, K. W. Bowyer, and P. J. Flynn. A region ensemble for 3d face recognition. *IEEE Transactions on Information Forensics and Security*, 3:62–73, 2008.

[7] J. H. Friedman. Another approach to polychotomous classification. Technical report, Stanford Department of Statistics, 1996.

[8] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[9] B. Haasdonk and H. Burkhardt. Invariant kernel functions for pattern analysis and machine learning. *Machine Learning*, 68:35–61, 2007.

[10] O. C. Hamsici and A. M. Martinez. Spherical-homoscedastic distributions: The equivalency of spherical and Normal distributions in classification. *Journal of Machine Learning Research*, 8:1583–1623, 2007.

[11] O. C. Hamsici and A. M. Martinez. Spherical-homoscedastic shapes. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2007.

[12] I. Kakadiaris, G. Passalis, G. Toderici, N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:640–649, 2007.

[13] D. G. Kendall. Shape-manifolds, Procrustean metrics and complex projective spaces. *Bulletin of the London Mathematical Society*, 16:81–121, 1984.

[14] J. T. Kent. The complex Bingham distribution and shape analysis. *Journal of the Royal Statistical Society - Series B*, 56:285–299, 1994.

[15] J. Koenderink and A. V. Doorn. Affine structure from motion. *Journal of the Optical Society of America A*, 8:377–385, 1990.

[16] R. Kondor. A complete set of rotationally and translationally invariant features for images. `arXiv:cs/0701127v3`, 2007.

[17] A. Kume and A. T. A. Wood. Saddlepoint approximations for the Bingham and Fisher-Bingham normalising constants. *Biometrika*, 92:465–476, 2005.

[18] F. D. la Torre, A. Collet, J. F. Cohn, and T. Kanade. Filtered component analysis to increase robustness to local minima in appearance models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[19] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[20] H. Ling and D. Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):286–299, 2007.

[21] K. V. Mardia and P. E. Jupp. *Directional Statistics*. John Wiley & Sons, West Sussex, England, 1999.

[22] A. Martinez, R. Wilbur, R. Shay, and A. Kak. The purdue asl database for the recognition of american sign language. In *In Proc. IEEE Multimodal Interfaces, Pittsburgh (PA)*, November 2002.

[23] A. M. Martinez and O. C. Hamsici. Who is LB1? discriminant analysis for the classification of specimens. *Pattern Recognition*, In Press.

[24] A. M. Martinez and M. Zhu. Where are linear feature extraction methods applicable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1934–1944, 2005.

[25] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[26] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical report, Columbia University CUCS-006-96, 1996.

[27] S. Obdrzálek and J. Matas. Sub-linear indexing for large scale object recognition. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 1–10, 2005.

[28] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[29] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Compututation*, 10(5):1299–1319, 1998.

[30] C. M. Theobald. An inequality for the trace of the product of two symmetric matrices. *Proceedings of the Cambridge Philosophical Society*, 77:256–267, 1975.

[31] A. Veeraraghavan, R. K. Roy-Chowdhury, and R. Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909, 2005.

[32] C. Walder and O. Chapelle. Learning with transformation invariant kernels. In *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems (NIPS 2007)*, 2007.

[33] L. Wei, E. Keogh, X. Xi, and S.-H. Lee. Supporting anthropological research with efficient rotation invariant shape similarity measurement. *Journal of the Royal Society Interface*, 4:207–222, 2007.

[34] M.-H. Yang, D. Roth, and N. Ahuja. Learning to recognize 3D objects with SNoW. In *Proceedinggs of European Conference on Computer Vision*, pages 439–454, 2000.

[35] M. Zhu and A. M. Martinez. Subclass discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1274–1286, 2006.