# Spherical-Homoscedastic Shapes

Onur C. Hamsici and Aleix M. Martinez
Department of Electrical and Computer Engineering
Ohio State University, Columbus, OH 43210
{hamsicio,aleix}@ece.osu.edu

## 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, which is shown to be very costly or impossible in most cases. The purpose of this paper is twofold. First, we show under which conditions the classification results obtained with complex Binghams are identical to those obtained with the easy-to-estimate complex Normal distribution. Second, we derive a kernel function which (intrinsically) maps the data into a space where the above conditions are satisfied and, hence, where the Normal model can be successfully used. This results in a simple, low-cost algorithm for representing and classifying shapes. We demonstrate the use of this technique in several experimental results for object and face recognition. Comparisons to other statistical shape representation/classification approaches demonstrate the superiority of the proposed algorithms in classification accuracy and computational time.*

## 1. Introduction

In 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 that best approximates one of the sample shapes; e.g., Procrustes analysis [2]. A much sought after alternative is given by the properties inherent to the complex domain, $\mathbb{C}^p$. Here, translation, scale and in-plane rotation invariance can be easily achieved by means of a simple mean-norm-normalization step followed by the modelling of the data using a complex probability distribution function (pdf) with the symmetric property,[1] such as the complex Bingham $\mathbb{C}B(\mathbf{A})$, with $\mathbf{A}$ the parameter matrix. The advantage of this approach (as opposed to its real domain counter part), is that shapes do not need to be aligned with respect to their rotation parameter, freeing ourselves from intensive computations. Unfortunately, the lack of exact solutions for the estimation of the parameters of complex pdfs make these algorithms impractical. Moreover, the high cost associated to the optimization alternatives for the estimation of the parameter matrix [6], make the approach unattractive.

In this paper, we show that, in general, one can use the zero-mean complex Normal distribution $\mathbb{C}N(\Sigma)$ in lieu of the complex Bingham $\mathbb{C}B(\mathbf{A})$. We will show that these properties correspond to a set of distributions (herein) termed spherical-homoscedastic [3]. By defining kernel functions which map the original data distributions to spherical-homoscedastic ones, we can derive simple, low computational cost algorithms for shape analysis and classification that are invariant to translation, scale and rotation, and are modelled using an easy-to-estimate pdf. We will further demonstrate that by using a scalar rotation invariant kernel, we can actually work with even simpler *real* distributions in the kernel-space. This final, key results will facilitate the design of a simple algorithm that can outperform all the others tested.

The rest of this paper is organized as follows. In section 2, 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 3, we introduce the concept of spherical-homoscedastic shapes. Section 4 presents the kernel approach. Examples and experimental results are in Section 5.

[1] A complex pdf $f$ has the symmetric property if $f(\mathbf{z}) = f(\mathbf{z}e^{i\theta})$ for all $\theta$, where $\mathbf{z} \in \mathbb{C}^p$.

## 2. Shape Analysis in the Complex Domain

Kendall's shape representation has been used by several authors to make shape descriptions invariant to translations and scales [2, 4, 9]. In this representation, the features of the vector $\mathbf{u}$ represent the complex coordinates of a set of points sampled from the shape contour.

### 2.1. Kendall's representation

To make shapes invariant to translation, Kendall's shape representation [4] uses a classical mean-normalization step (i.e., where all feature vectors have zero mean). This is achieved with a simple multiplication of each feature vector $\mathbf{u} \in \mathbb{C}^p$ with the $(p-1) \times p$ Helmert sub-matrix $\mathbf{H}$,

$$\begin{pmatrix} h_1 & -h_1 & 0 & \dots & \dots & 0 & 0 \\ h_2 & h_2 & -2h_2 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{p-1} & \dots & \dots & \dots & \dots & h_{p-1} & -(p-1)h_{p-1} \end{pmatrix},$$

where $h_j = -(j^2+j)^{-1/2}$. Note that this process will project the original feature vectors $\mathbf{u} \in \mathbb{C}^p$ to $\mathbb{C}^{p-1}$.

Scale changes are eliminated by normalizing the norm of each resulting shape feature vector $\mathbf{Hu}$ to have unit length. This maps the data onto the surface of a (p-2)-dimensional complex hypersphere $\mathbb{C}S^{p-2}$. More formally, $\mathbf{z} = \mathbf{Hu}/\|\mathbf{Hu}\| \in \mathbb{C}S^{p-2}$, with the resulting vector $\mathbf{z} \in \mathbb{C}S^{p-2}$ referred to as the *preshape*.

### 2.2. Complex Bingham

To be rotation invariant, we need to describe our sample preshapes $\mathbf{z}$ using a pdf that carries the following property $f(\mathbf{z}) = f(\mathbf{z}e^{i\theta})$, $\forall\theta \in [0, 2\pi]$. This is so, because multiplying with $e^{i\theta}$ describes all possible planar rotations of our preshapes defined in $\mathbb{C}S^{p-2}$. This property is the bases of the complex Bingham distribution [5] given by

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

where $C_{\mathbb{C}B}$ is a normalizing constant which guarantees that $\int_{\mathbb{C}S^{p-2}} f(\mathbf{z})\, d\mathbf{z} = 1$, and $\mathbf{z}^*$ is the complex conjugate of the transpose of $\mathbf{z}$ (i.e., $\mathbf{z}^* = \bar{\mathbf{z}}^T$).[2]

To estimate the parameters, we proceed as follows. Since $\mathbf{A}$ is a $(p-1) \times (p-1)$ Hermitian parameter matrix, its spectral decomposition can be written as $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^*$, where $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{p-1})$ is a matrix whose columns $\mathbf{q}_i$ correspond to the eigenvectors of $\mathbf{A}$ and $\mathbf{\Lambda} = diag\,(\lambda_1, \dots, \lambda_{p-1})$ is the diagonal matrix of corresponding eigenvalues. For a random set of unit vectors $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)$ 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 and $\mathbf{Z}$ is a $(p-1) \times n$ complex matrix with each of the $n$ columns corresponding to a sample shape feature vector on $\mathbb{C}S^{p-2}$. Since the $tr(\mathbf{SA})$ 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 $\Lambda$. This is because to compute $\Lambda$, one requires to first estimate the normalizing constant, which includes the generally impossible task of integrating the pdf over the nonlinear complex hypersphere $\mathbb{C}S^{p-2}$. Although Kent [5] expressed the normalizing constant as a function of the eigenvalues, $c_{\mathcal{C}B}(\Lambda) = 2\pi^p \sum_{j=1}^{p} a_j \exp(-\lambda_j)$, where $a_j^{-1} = \prod_{i \neq j}(\lambda_i - \lambda_j)$, this formulation cannot be used in practice because of the numerical instabilities given when some eigenvalues are (almost) equal [6].

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 [6]. This result allows us to define the following optimization problem to estimate $\Lambda$,

$$\hat{\Lambda} = \arg\max_{\Lambda}\; n\, tr(\hat{\Lambda}_S\Lambda) - n \log(\hat{c}_{\mathbb{C}B}(\Lambda)), \qquad (2)$$

where $\hat{\Lambda}$ is the estimated eigenvalue matrix, and $\hat{c}_{\mathbb{C}B}(\Lambda)$ is the saddlepoint approximation defined in [6].

### 2.3. Complex Normal

In what follows, we will show that generally one can substitute the complex Bingham pdf $\mathbb{C}B(\mathbf{A})$ by a complex zero-mean Normal distribution $\mathbb{C}N(\Sigma)$ and obtain the same classification results to those that would have been generated with the use of a complex Bingham. The pdf of a zero-mean multivariate complex Normal distribution (i.e., $\mathbf{z} \sim \mathbb{C}N(\Sigma)$) is given by

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

where $\Sigma$ is a $(p-1) \times (p-1)$ positive-definite complex Hermitian matrix and $C_{\mathbb{C}N}(\Sigma) = \pi^{p-1} \det(\Sigma)$ is the normalizing constant. Note that the eigenvectors of the sample covariance matrix $\widehat{\Sigma}$ are the same as those of $\mathbf{A}$. This is because the maximum likelihood of $\Sigma = \mathbf{S}$, since the distribution is zero mean.

The main advantage of the complex zero-mean Gaussian over the complex Bingham is the easiness associated to parameter estimation. 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

[2] Note that $f(\mathbf{z}e^{i\theta}) = C^{-1}_{\mathbb{C}B}(\mathbf{A}) \exp(e^{-i\theta}\mathbf{z}^*\mathbf{A}\,\mathbf{z}e^{i\theta}) = C^{-1}_{\mathbb{C}B}(\mathbf{A}) \exp(\mathbf{z}^*\mathbf{A}\mathbf{z}) = f(\mathbf{z})$.

obtained by calculation of the determinant of the covariance. The most efficient algorithm developed to date, has an upper-bound complexity of $p^{2.376}$ [1]. This means that our algorithm has a polynomial time complexity of degree $\leq 2.376$.

On the other hand, to compute an *approximate* result for the parameters of complex Binghams, we need to calculate the eigenvectors and eigenvalues of $\mathbf{S}$ and the optimization procedure defined in (2). An eigenvalue decomposition is already of $O(p^3)$. Adding, an optimization routine, such as the sequential quadratic programming (which requires solving a quadratic programming subproblem at each iteration), leads to a complexity of order 4.

It is clear that the use of complex Gaussians facilitates the computational burden. Furthermore, as we will prove next, the use of Gaussians can guarantee an optimal estimate, which is rarely the case with approximations.

## 3. Spherical-Homoscedastic Shapes

In the rest of this paper, we will make the assumption of equal priors, simplifying the Bayes decision rule to the comparison of the likelihoods.

The likelihood of an observation $\mathbf{z}$ to belong to a class can be computed as

$$d^2_{\mathbb{C}N}(\mathbf{z}) = -\log f(\mathbf{z}) = \mathbf{z}^*\Sigma^{-1}\mathbf{z} + \log(C_{\mathbb{C}N}(\Sigma)) \quad (3)$$

for complex Normals, and

$$d^2_{\mathbb{C}B}(\mathbf{z}) \quad = \quad -\mathbf{z}^*\mathbf{A}\mathbf{z} + \log(C_{\mathbb{C}B}(\mathbf{A})) \quad (4)$$

for complex Bingham distributions.

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 all the same (i.e., $\Sigma_1 = \cdots = \Sigma_r$), where $\mu_1, \ldots, \mu_r$ are the means of the distributions. Homoscedastic Gaussian pdfs are relevant, because their Bayes decision boundaries are given by hyperplanes.

However, when all feature vectors are restricted to lay on the surfaces of a hypersphere, the definition of homoscedasticity given above becomes too restrictive. For example, if we use zero-mean Gaussian pdfs to model some spherical data that is complex symmetric about its mean, then only those distributions that are exactly identical, can be considered homoscedastic. This is illustrated in Fig. 1. Although the three classes shown in this figure have the same covariance matrix up to a rotation, only those that are identical (i.e., Class 1 and 3) are said to be homoscedastic. Nonetheless, the decision boundaries for each pair of classes in Fig. 1 are all hyperplanes [3]. Furthermore, we will show that these hyperplanes (given by approximating the original pdfs with Gaussians) are generally the same as those obtained using Bayes on the true underlying distributions. Therefore, we define a more general type of rotational-invariant homoscedasticity.

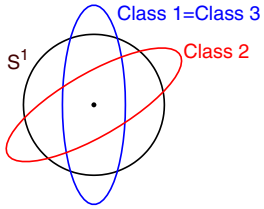


Figure 1. Assume we model the data of three classes laying on $S^1$ using three zero-mean Gaussian distributions. In this case, each set of Gaussian distributions can only be homoscedastic when they are the exact same distribution. In this figure, Class 1 and 3 are homoscedastic, but classes 1,2 and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

**Definition 1.** *[3] Two pdfs ($f_1$ and $f_2$) of the same form are said to be spherical-homoscedastic if the Bayes decision boundary between $f_1$ and $f_2$ is given by one or more hyperplanes and the variances defining the two distributions are the same.*

Our main goal in the rest of this section, is to demonstrate that the linear decision boundaries (given by Bayes) of a pair of spherical-homoscedastic complex Bingham distributions are the same as those obtained when these are assumed to be complex Gaussians. We start with the study of the zero-mean complex Gaussian distribution.

**Theorem 2.** *Two zero-mean complex Gaussian distributions, $\mathbb{C}N_1(\Sigma)$ and $\mathbb{C}N_2(\mathbf{R}^*\Sigma\mathbf{R})$, are spherical-homoscedastic if $\mathbf{R}$ is a complex Hermitian rotation matrix defining a planar rotation in the subspace spanned by any two of the eigenvectors of $\Sigma$, say $\mathbf{v}_1$ and $\mathbf{v}_2$.*

*Proof.* The Bayes classification boundary between these pdf can be obtained by making the ratio of the log-likelihood equations equal to one. Formally,

$$\begin{aligned} & -\log C_{\mathbb{C}N}(\Sigma) - \mathbf{z}^*\Sigma^{-1}\mathbf{z} = \\ & -\log C_{\mathbb{C}N}\left(\mathbf{R}^*\Sigma\mathbf{R}\right) - \mathbf{z}^*\left(\mathbf{R}^*\Sigma\mathbf{R}\right)^{-1}\mathbf{z}. \end{aligned}$$

Since $C_{\mathbb{C}N}(\Sigma) = C_{\mathbb{C}N}\left(\mathbf{R}^*\Sigma\mathbf{R}\right)$ we can simplify the above equation to yield $\mathbf{z}^*\Sigma^{-1}\mathbf{z} = \mathbf{z}^*\mathbf{R}^*\Sigma^{-1}\mathbf{R}\mathbf{z}$. Now, let the spectral decomposition of $\Sigma$ be $\mathbf{V}\Lambda\mathbf{V}^* = (\mathbf{v}_1, \cdots, \mathbf{v}_{p-1})\, diag(\lambda_1, \cdots, \lambda_{p-1})\, (\mathbf{v}_1, \cdots, \mathbf{v}_{p-1})^*$. This allows us to write $\Sigma^{-1}$ in an open form as

$$\sum_{i=1}^{p-1} \lambda_i^{-1}(\mathbf{z}^*\mathbf{v}_i)^2 = \sum_{i=1}^{p-1} \lambda_i^{-1}(\mathbf{z}^*\mathbf{R}^*\mathbf{v}_i)^2$$

In addition, we know that the complex rotation matrix $\mathbf{R}$ defines a rotation in the $(\mathbf{v}_1, \mathbf{v}_2)$-plane. This means that $\mathbf{R}^*\mathbf{v}_i = \mathbf{v}_i$ for $i \neq \{1, 2\}$ (i.e., the eigenvectors orthogonal to those defined by a planar rotation will not vary), which allows us to simplify our equation to

$$\begin{aligned} & \lambda_1^{-1}\left[(\mathbf{z}^*\mathbf{v}_1)^2 - (\mathbf{z}^*\mathbf{R}^*\mathbf{v}_1)^2\right] \\ = \quad & \lambda_2^{-1}\left[(\mathbf{z}^*\mathbf{R}^*\mathbf{v}_2)^2 - (\mathbf{z}^*\mathbf{v}_2)^2\right]. \end{aligned} \quad (5)$$

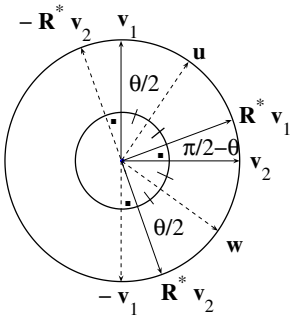


Figure 2. Shown here are two orthonormal vectors, $\mathbf{v}_1$ and $\mathbf{v}_2$, and their rotated versions, $\mathbf{R}^*\mathbf{v}_1$ and $\mathbf{R}^*\mathbf{v}_2$. We see that $\mathbf{R}^*\mathbf{v}_1 + \mathbf{v}_1 = 2\mathbf{u}\cos\left(\frac{\theta}{2}\right)$, $\mathbf{R}^*\mathbf{v}_2 +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

Assume that $\mathbf{R}$ rotates the vector $\mathbf{v}_1$ $\theta$ degrees in the clockwise direction yielding $\mathbf{R}^*\mathbf{v}_1$. Similarly, $\mathbf{v}_2$ becomes $\mathbf{R}^*\mathbf{v}_2$. From Fig. 2 we see that

$$\mathbf{v}_2 - \mathbf{R}^*\mathbf{v}_2 = 2\mathbf{u}\cos\left(\frac{\pi}{2} - \frac{\theta}{2}\right), \; \mathbf{v}_2 + \mathbf{R}^*\mathbf{v}_2 = 2\mathbf{w}\cos\left(\frac{\theta}{2}\right),$$
$$\mathbf{R}^*\mathbf{v}_1 - \mathbf{v}_1 = 2\mathbf{w}\cos\left(\frac{\pi}{2} - \frac{\theta}{2}\right), \; \mathbf{R}^*\mathbf{v}_1 + \mathbf{v}_1 = 2\mathbf{u}\cos\left(\frac{\theta}{2}\right),$$

where $\mathbf{u}$ and $\mathbf{w}$ are the unit vectors as shown in Fig. 2. Therefore,

$$\mathbf{v}_2 + \mathbf{R}^*\mathbf{v}_2 = (\mathbf{R}^*\mathbf{v}_1 - \mathbf{v}_1)\cot\left(\frac{\theta}{2}\right),$$
$$\mathbf{v}_2 - \mathbf{R}^*\mathbf{v}_2 = (\mathbf{R}^*\mathbf{v}_1 + \mathbf{v}_1)\tan\left(\frac{\theta}{2}\right).$$

If we use these results in (5), we find that

$$\lambda_1^{-1}\left[(\mathbf{z}^*\mathbf{v}_1)^2 - (\mathbf{z}^*\mathbf{R}^*\mathbf{v}_1)^2\right] = \lambda_2^{-1}\left[(\mathbf{z}^*\mathbf{v}_1)^2 - (\mathbf{z}^*\mathbf{R}^*\mathbf{v}_1)^2\right].$$

The two possible solutions to this equation provide the two hyperplanes for the Bayes classifier,

$$\mathbf{z}^*(\mathbf{R}^*\mathbf{v}_1 + \mathbf{v}_1) = 0, \tag{6}$$
$$\mathbf{z}^*(\mathbf{R}^*\mathbf{v}_1 - \mathbf{v}_1) = 0. \tag{7}$$

Therefore, $\mathbb{C}N_1$ and $\mathbb{C}N_2$ are spherical-homoscedastic. □

**Theorem 3.** *Two complex Bingham distributions, $\mathbb{C}B_1(\mathbf{A})$ and $\mathbb{C}B_2(\mathbf{R}^*\mathbf{A}\mathbf{R})$, are spherical-homoscedastic if $\mathbf{R}$ is a complex Hermitian rotation matrix defining a planar rotation in the subspace spanned by any two of the eigenvectors of $\mathbf{A}$, say $\mathbf{q}_1$ and $\mathbf{q}_2$.*

*Proof.* Making the ratio of the log-likelihood equations equal to one yields

$$\begin{aligned} &- \log(C_{\mathbb{C}B}(\mathbf{A})) + \mathbf{z}^*\mathbf{A}\mathbf{z} = \\ &- \log(C_{\mathbb{C}B}(\mathbf{R}^*\mathbf{A}\mathbf{R})) + \mathbf{z}^*\mathbf{R}^*\mathbf{A}\mathbf{R}\mathbf{z}. \end{aligned} \tag{8}$$

Since the normalizing constant of a complex Bingham distribution depends only on the eigenvalues, $C_{\mathbb{C}B}(\mathbf{A}) = C_{\mathbb{C}B}(\mathbf{R}^*\mathbf{A}\mathbf{R})$. Furthermore, since the rotation is defined in the subspace spanned by $\mathbf{q}_1$ and $\mathbf{q}_2$, and $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^*$, we can follow the proof of Theorem 2 to show that the two hyperplanes defining the Bayes decision boundary are

$$\mathbf{z}^*(\mathbf{R}^*\mathbf{q}_1 + \mathbf{q}_1) = 0, \tag{9}$$
$$\mathbf{z}^*(\mathbf{R}^*\mathbf{q}_1 - \mathbf{q}_1) = 0. \tag{10}$$

This means that $\mathbb{C}B_1$ and $\mathbb{C}B_2$ are spherical-homoscedastic. □

We are now in a position to prove that the decision boundaries obtained using two zero-mean complex Gaussian distributions are the same as those defined by two spherical-homoscedastic complex Bingham pdfs.

**Theorem 4.** *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 $\mathbb{C}N_2(\mathbf{R}^*\Sigma\mathbf{R})$, with $\Sigma = \mathbf{S}$.*

*Proof.* Since the data sampled from a complex Bingham distribution is symmetric with respect to the center of coordinates, its mean will be an all zeros vector, $(0, \ldots, 0)^T$. Hence, the sample covariance matrix will be equal to the sample autocorrelation matrix $\mathbf{S} = n^{-1}\mathbf{X}\mathbf{X}^*$. In short, the estimated Gaussian pdf of $\mathbb{C}B_1(\mathbf{A})$ will be $\mathbb{C}N_1(\mathbf{S})$. This means that the m.l.e. of the orthonormal matrix $\mathbf{Q}$ (where $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^*$) is given by the eigenvectors of $\mathbf{S}$. As a consequence, the two (zero-mean) complex Gaussian pdfs representing the data sampled from two spherical-homoscedastic complex Bingham distributions, $\mathbb{C}B_1(\mathbf{A})$ and $\mathbb{C}B_2(\mathbf{R}^*\mathbf{A}\mathbf{R})$, are $\mathbb{C}N_1(\mathbf{S})$ and $\mathbb{C}N_2(\mathbf{R}^*\mathbf{S}\mathbf{R})$.

Following Theorem 2, $\mathbb{C}N_1(\mathbf{S})$ and $\mathbb{C}N_2(\mathbf{R}^*\mathbf{S}\mathbf{R})$ are spherical-homoscedastic if $\mathbf{R}$ is spanned by any two eigenvectors of $\mathbf{S}$. Since the eigenvectors of $\mathbf{A}$ and $\mathbf{S}$ are the same ($\mathbf{v}_i = \mathbf{q}_i$ for all $i$), $\mathbb{C}N_1(\mathbf{S})$ and $\mathbb{C}N_2(\mathbf{R}^*\mathbf{S}\mathbf{R})$ are spherical-homoscedastic. Moreover, the hyperplanes of the spherical-homoscedastic complex Bingham pdfs $\mathbb{C}B_1(\mathbf{A})$ and $\mathbb{C}B_2(\mathbf{R}^*\mathbf{A}\mathbf{R})$, Eqs. (9 - 10), and the hyperplanes of the spherical-homoscedastic complex Gaussian pdfs $\mathbb{C}N_1(\mathbf{S})$ and $\mathbb{C}N_2(\mathbf{R}^*\mathbf{S}\mathbf{R})$, Eqs. (6 - 7), are also the same. □

This result allows us to introduce a very important concept: that of spherical-homoscedastic shapes.

**Definition 5.** *Two shape feature vectors $\mathbf{z}_i \in \mathbb{C}S^{p-2}$ and $\hat{\mathbf{z}}_i \in \mathbb{C}S^{p-2}$ are called* spherical-homoscedastic shapes *if $\mathbf{z}_i \sim \mathbb{C}B(\mathbf{A})$ and $\hat{\mathbf{z}}_i \sim \mathbb{C}B(\mathbf{R}^*\mathbf{A}\mathbf{R})$, where $\mathbf{R}$ is defined by two eigenvectors of $\mathbf{A}$.*

Next, note that the more two shape feature vectors deviate from this spherical-homoscedastic definition, the more the decision boundary between them will deviate from the classification hyperplanes derived above. This means, that (in general) as the distributions (describing the shape data) deviate from spherical-homoscedastic, the classification error will increase. Therefore, our final goal is to make our data as close as possible to the spherical-homoscedastic definition introduced in this section. This is to be addressed next.

## 4. Kernel Spherical-Homoscedastic Shapes

We will now employ the idea of the kernel trick to define algorithms that are nonlinear in the original space, but linear in the kernel one. This will be used to tackle the general spherical-heteroscedastic case as if the shape distributions were linearly separable spherical-homoscedastic.

*Key* to this process is to realize that by using rotation invariant kernels, we can drop the requirement of working in the complex domain, since this was precisely done to gain rotation invariance.

One such kernel can be derived as follow,

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\exp(-i\theta_{\mathbf{xy}})\|^2}{2\sigma^2}\right),$$

where $\mathbf{x},\mathbf{y} \in \mathbb{C}S^{p-2}$, $\sigma$ is the kernel parameter to be optimized, and $\theta_{\mathbf{xy}} = \angle(\mathbf{x}^*\mathbf{y})$ defines the angle between $\mathbf{x}$ and $\mathbf{y}$. We see that the kernel defined above is invariant to any rotation of $\mathbf{y}$, since $\|\mathbf{x}-\mathbf{y}\exp(-i\theta_{\mathbf{xy}})\|^2 = \mathbf{x}^*\mathbf{x}+\mathbf{y}^*\mathbf{y} - \mathbf{x}^*\mathbf{y}\exp(-i\theta_{\mathbf{xy}}) - \exp(i\theta_{\mathbf{xy}})\mathbf{y}^*\mathbf{x} = 2-2\|\mathbf{x}^*\mathbf{y}\|$. We also note that our kernel carries an inherent mapping resulting in a kernel space that is also spherical, i.e. $k(\mathbf{x},\mathbf{x}) = 1$.

Since the data is already rotation invariant in the kernel space, we can now model the data with real Bingham distributions, $\mathbf{B}_i(\mathbf{A})$, instead.

Let two spherical-homoscedastic Bingham distributions be $B_1(\mathbf{A})$ and $B_2(\mathbf{R}^T\mathbf{A}\mathbf{R})$, $\mathbf{R}$ a planar rotation. Since two spherical-homoscedastic Binghams can be represented with Normals, we can derive our classifier, following the real counterpart of (5). That is, $\mathbf{x}$ will be classified in the first distribution, $B_1$, if the following holds

$$\begin{aligned} & \lambda_1\left((\mathbf{x}^T\mathbf{q}_1)^2 - (\mathbf{x}^T\mathbf{R}^T\mathbf{q}_1)^2\right) \\ + \quad & \lambda_2\left((\mathbf{x}^T\mathbf{q}_2)^2 - (\mathbf{x}^T\mathbf{R}^T\mathbf{q}_2)^2\right) > 0. \end{aligned}$$

Using the result shown in Theorem 2, where we expressed $\mathbf{q}_2$ as a function of $\mathbf{q}_1$, we can simplify the above equation to

$$(\lambda_1-\lambda_2)\left((\mathbf{x}^T\mathbf{q}_1)^2 - (\mathbf{x}^T\mathbf{R}^T\mathbf{q}_1)^2\right) > 0.$$

Therefore, if $\lambda_1 > \lambda_2$, $\mathbf{x}$ will be in $B_1$ when

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

This result can be stated in compact form as

$$|\mathbf{x}^T\mathbf{q}_1| > |\mathbf{x}^T\mathbf{R}^T\mathbf{q}_1|. \tag{11}$$

The relevance of (11) is that, a test feature vector $\mathbf{x}$ is classified to that class providing the largest inner product value. We can now readily extend this result to the multi-class problem. For this, let $B_1(\mathbf{A})$ be the distribution of the first class and $B_a(\mathbf{R}_a^T\mathbf{A}\mathbf{R}_a)$ that of the $a^{th}$ class, where now $\mathbf{R}_a$ is defined by two eigenvectors of $\mathbf{A}$, $\mathbf{q}_{a_1}$ and $\mathbf{q}_{a_2}$, with corresponding eigenvalues $\lambda_{a_1}$ and $\lambda_{a_2}$ and we have assumed $\lambda_{a_1} > \lambda_{a_2}$. Then, the class of a new test feature vector $\mathbf{x}$ is given by

$$\arg\max_a |\mathbf{x}^T\mathbf{q}_{a_1}|. \tag{12}$$

Our next step is to derive the same classifier in the kernel space. From our discussion, we require to find the eigenvectors of the covariance matrix. The covariance matrix in the kernel space is $\Sigma_a^\Phi = \Phi(\mathbf{X}_a)\Phi(\mathbf{X}_a)^T$, where $\mathbf{X}_a$ is a matrix whose columns are the sample feature vectors of class $a$, $\mathbf{X}_a = \left(\mathbf{x}_{a_1},\mathbf{x}_{a_2},\ldots,\mathbf{x}_{a_{n_a}}\right)$, and $n_a$ is the number of samples in class $a$.

This allows us to obtain the eigenvectors of the covariance matrix from $\Sigma_a^\Phi\mathbf{V}_a^\Phi = \mathbf{V}_a^\Phi\Lambda_a^\Phi$. These $d$-dimensional eigenvectors $\mathbf{V}_a^\Phi = \{\mathbf{v}_{a_1}^\Phi,\ldots,\mathbf{v}_{a_{n_a}}^\Phi\}$ are not only the same as those of $\mathbf{A}_a^\Phi$, but are sorted in the same order too.

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

$$\Phi(\mathbf{X}_a)\Phi(\mathbf{X}_a)^T\Phi(\mathbf{X}_a)\Delta_a = \Phi(\mathbf{X}_a)\Delta_a\Lambda_a.$$

Multiplying both sides by $\Phi(\mathbf{X}_a)^T$ and cancelling terms, we can simplify this equation to

$$\mathbf{K}_a\Delta_a = \Delta_a\Lambda_a,$$

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

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

$$\mathbf{V}_a^\Phi = \Phi(\mathbf{X}_a)\Delta_a{\Lambda_a}^{-1/2},$$

where $\mathbf{V}_a^\Phi = \{\mathbf{v}_{a_1}^\phi,\ldots,\mathbf{v}_{a_{n_a}}^\phi\}$, and $\mathbf{v}_{a_i}^\phi \in S^d$, and $d$ is the dimensionality of the kernel space.

The classification scheme derived in (12) can now be extended to classify $\phi(\mathbf{x})$ as

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

| | | *Kents' Hybrid* | *complex Bingham* | *complex Normal* | *Kernel SH* |
|---|---|---|---|---|---|
| ETH | *Recognition Rate* | 79.66 | 86.95 | 87.5 | **91.59** |
| | *Training Time (in seconds)* | 1.06 | 18.34 | **0.95** | 55.59 |
| | *Testing Time (in seconds)* | 0.05 | **0.02** | **0.02** | 0.03 |
| COIL | *Recognition Rate* | 90.47 | 91.75 | 95.47 | **96.61** |
| | *Training Time (in seconds)* | 6.12 | 109.12 | **4.85** | 241.53 |
| | *Testing Time (in seconds)* | 25.01 | **2.99** | **2.98** | **0.36** |

Table 1. The average recognition rates obtained on the ETH-80 and COIL-100 datasets using the leave-one-object-out and 9-fold cross-validation, respectively.

where the index $i = \{1, \ldots, n_a - 1\}$ defining the eigenvector $\mathbf{v}^{\phi}_{a_i}$ must be kept constant for all $a$.

This final result, can be written using a kernel as

$$\arg\max_a \left| \sum_{l=1}^{n_a} \frac{k(\mathbf{x}, \mathbf{x}_l)\delta_{a_i}(l)}{\sqrt{\lambda^{\phi}_{a_i}}} \right|, \tag{13}$$

where $\Delta_a = \{\delta_{a_1}, \ldots, \delta_{a_{n_a}}\}$, and $\delta_{a_i}(l)$ is the $l^{th}$ coefficient of the vector $\delta_{a_i}$. The simplest algorithm that can be implemented is by assigning $i = 1$, i.e., classification based on the first basis vector of the class distributions in the kernel space. This is the implementation that we will test next.

## 5. Experiments

Our first experiment will test the classification capabilities of the proposed algorithms in an object categorization problem. We use the shape vectors of the eight categories in the ETH-80 dataset [7], i.e., 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. The contour shape of each object silhouette is sampled with 100 equidistant points, and mean and norm normalization are applied. We then 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 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).

We tested four different algorithms. The first is Kent's hybrid model [5], which is one of the most used approaches to represent and classify shapes. It first maps the spherical data to the tangent hyperplane that best describes the training data and, then, uses a Gaussian to estimate the projected data. Our second algorithm, uses the parameter approximation algorithm of Kume and Wood [6], given in Eq. (2), to fit the training data with a complex Bingham distribution. The third algorithm, uses the complex Normal fit presented in Section 3, which provides optimal results whenever the data is spherical-homoscedastic. Our final algorithm, uses the kernel approach defined in Section 4, which employs a rotation invariant RBF kernel and the (real) classifier derived in Eq. (13). This last algorithm is referred to as *kernel SH*, where *SH* stands for spherical-homoscedastic.

These results are summarized in Table 1. In these experiments, we further aided Kume and Wood's optimization 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). The resulting classification rate was superior to that of Kent's hybrid model, and comparable to our Normal fit (as stated by our theory). However, we see that our algorithm was much faster in both, training and testing (since the polynomial complexity in our algorithm is of a much lower order, as shown in Section 3).

We also note that the kernel approach provides even better classification results. This is because we could optimize the parameters of our rotation invariant RBF kernel to make the data more spherical-homoscedastic. The optimization of the kernel was done using the classical leave-one-sample-out strategy on the training set. The best fit was given at $\sigma = .1$. However, this result does come with an increase of the computational time required to optimize the kernel.

In our tables, we also provide the time required to train each algorithm using a leave-one-object set, and the test time required to test the 41 images left out.[3] Clearly, the algorithm described in Section 3, which uses complex Normal distributions, is the fastest in both, training and testing. Kents' hybrid model is the second fastest, but provides the lowest classification rate. As anticipated, the kernel approach carries the highest computational time in training, since we need to run a leave-one-sample-out procedure on the training set to optimize the parameter of the kernel.

Our second experimental results utilize the COIL-100 object dataset [8]. This database consists of 100 object classes, each containing 72 images. These images are photographed at intervals of 5 degrees apart, providing the 72 samples per class. We carry a 9-fold cross-validation test. Here, 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 four algorithms described above are in

[3] Computational times is provided in seconds, obtained with Matlab code in a Pentium 4 at 3GHz.

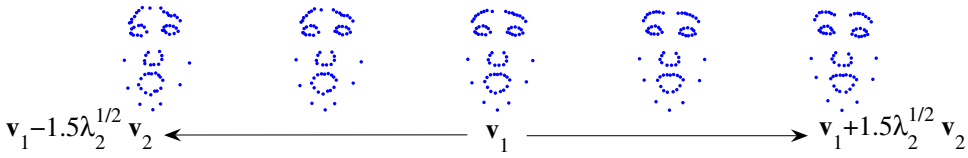


Figure 3. Here we show the dominant eigenvector of the complex sample autocorrelation matrix, $\mathbf{v}_1$, calculated from the mean-norm normalized face shapes with different expressions, fear and happy. We also show the $\pm 1.5\sqrt{\lambda_2}\mathbf{v}_2$ deviations from $\mathbf{v}_1$.

Table 1.

The pattern observed in the ETH-80 results repeats itself. Kent's hybrid model provides the lowest classification accuracy, followed by the approximation to the complex Bingham, the complex Normal algorithm presented in Section 3, and the kernel SH approach derived in Section 4 (where the kernel parameter was, once more, optimized using the leave-one-sample- out strategy on the training set, resulting in $\sigma = .01$). Although, the kernel method provides higher recognition results than those obtained with complex Normals, this comes with an extra cost in the training stage. When this is not an issue, the kernel method will be preferred. When training time is relevant, the complex Normal approach will be the most attractive alternative, since this provides the best complexity-to-accuracy ratio.

The experimental results reported thus far addressed the problem of classification. We now show an example for the representation of facial expressions. Here, we use the shape extracted from two distinct expressions – fear and happy. The representation is given by 70 landmarks, 7 to represent the chin line and 63 uniformly distributed over the contours of the eyes, brows, mouth and nose. The first eigenvector $\mathbf{v}_1$ of the complex Normal pdf that best describes this data correspond to the Procrustes mean shape. The second eigenvector $\mathbf{v}_2$ can then be used to deform this mean shape with regard to some of the shape parameters of the face. As shown in Fig. 3, $\mathbf{v}_2$ is mostly associated to changes of the eyebrows and slight changes of the mouth and chin.

## 6. Conclusions

Defining algorithms that provide invariance to translation, scale and in-plane rotation is essential for the efficient analysis and classification of shapes. While Procrustes-based methods have been attractive, theoretically, those based on the properties provided by the complex scalar-rotation-invariant distributions defined on the complex hypersphere (e.g., Bingham) would be preferred. Unfortunately, the difficulty associated with the parameter estimation of these distributions, has made this approach unattractive. In this paper, we have demonstrated that we can substitute the estimate of the complex Bingham distribution with that of the complex Normal, which can be estimated using low computational cost methods. Our second goal was to use the idea of the kernel trick to map the original data into a space where the equivalency between Bingham and Normals is optimized. This resulted in higher recognition rates.

## Acknowledgments

This research was partially supported by NIH under grant R01 DC 005241.

## References

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

[2] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, West Sussex, England, 1998. 1, 2

[3] 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. 1, 3

[4] D. G. Kendall. Shape-manifolds, Procrustean metrics and complex projective spaces. *Bulletin of the London Mathematical Society*, 16:81–121, 1984. 2

[5] J. T. Kent. The complex Bingham distribution and shape analysis. *Journal of the Royal Statistical Society - Series B*, 56:285–299, 1994. 2, 6

[6] A. Kume and A. T. A. Wood. Saddlepoint approximations for the Bingham and Fisher-Bingham normalising constants. *Biometrika*, 92:465–476, 2005. 1, 2, 6

[7] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2003. 6

[8] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical report, Columbia University CUCS-006-96, 1996. 6

[9] 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. 2