# Hierarchical Similarity Transformations Between Gaussian Mixtures

George Rigas, Christophoros Nikou, *Senior Member, IEEE*, Yorgos Goletsis, *Member, IEEE*, and Dimitrios I. Fotiadis, *Senior Member, IEEE*

***Abstract*— In this paper, we propose a method to estimate the density of a data space represented by a geometric transformation of an initial Gaussian mixture model. The geometric transformation is hierarchical, and it is decomposed into two steps. At first, the initial model is assumed to undergo a global similarity transformation modeled by translation, rotation, and scaling of the model components. Then, to increase the degrees of freedom of the model and allow it to capture fine data structures, each individual mixture component may be transformed by another, local similarity transformation, whose parameters are distinct for each component of the mixture. In addition, to constrain the order of magnitude of the local transformation (LT) with respect to the global transformation (GT), zero-mean Gaussian priors are imposed onto the local parameters. The estimation of both GT and LT parameters is obtained through the expectation maximization framework. Experiments on artificial data are conducted to evaluate the proposed model, with varying data dimensionality, number of model components, and transformation parameters. In addition, the method is evaluated using real data from a speech recognition task. The obtained results show a high model accuracy and demonstrate the potential application of the proposed method to similar classification problems.**

***Index Terms*— Expectation maximization (EM) algorithm, Gaussian mixture model, registration of point sets, similarity transformation.**

## I. Introduction

GAUSSIAN mixture models (GMMs) were extensively studied with applications in many domains, such as density estimation [1], clustering [2], [3], classification [4], image registration [5], [6], and regression [7], [8]. There are two main issues in the application of mixture models. The first is the estimation of model parameters. Parameter estimation is generally based on the maximum likelihood (ML) or maximum *a posteriori* (MAP) expectation maximization (EM) algorithm [9]–[11] or its variational extensions [12], [13]. The second issue is the choice of the number of mixture components. There are cases where the number of components is known *a priori*, (e.g., some classification problems). In the majority of applications, this number is, however, unknown [14]–[16], [18].

In many domains, such as speaker adaptation [19], image registration [5], and tracking [20], the following case of parameter estimation is often encountered: an initial GMM is considered, where the number of components and the model parameters are already estimated from a training data set and are considered known. The initial model is then geometrically transformed and a new data set is generated. To estimate the unknown transformation parameters, we may consider to simply retrain the model using the new data set as input. Without any restrictions or the imposition of constraints, this approach could lead to a violation of the one-to-one mapping between the components of the estimated and the initial GMM [6]. A common approach is the imposition application of constraints on the geometrically transformed model parameters (Fig. 1) where the geometric transformation is usually a similarity transformation consisting of rotation, scaling, and translation.

In this paper, we propose a method to estimate the transformation parameters between Gaussian mixtures, which is based on the EM algorithm. At first, we consider the case where a unique similarity transformation is applied to GMM components. We call this type of transformation as global transformation (GT). This is the case, for example, of the motion of a moving camera capturing a still scene. In [19], [21], this problem was discussed but the authors focused on the special case where the transformation consisted only of a scaling matrix and the covariance matrices of the Gaussian components were diagonal. Moss and Hancock [22] also addressed this constrained transformation problem but they were limited to the image registration problem, thus they used the 2-D mixture models. In this paper, we treat the genera $D$-dimensional problem including (apart from scaling) rotation and full covariance matrices.

The assumption made in [19], [21], and [22] of a unique similarity transformation applied to all mixture components, may hold for a number of problems. It could also be of great interest to allow each component to have an individual transformation and thus, to increase the degrees of freedom of the model. A second contribution of this paper is to consider another layer of transformations (apart from the GT) applied to each individual component with distinct parameters for each mixture component. We call them as local transformations

Manuscript received July 30, 2012; revised January 30, 2013 and May 28, 2013; accepted May 29, 2013. Date of publication June 28, 2013; date of current version October 15, 2013.

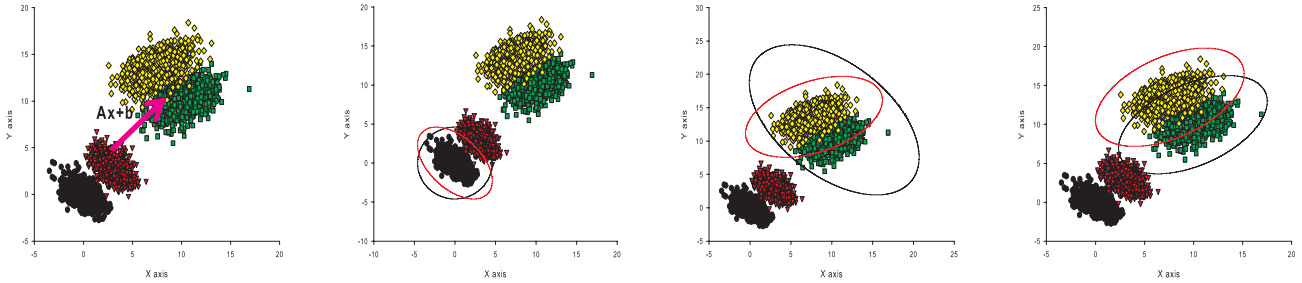
G. Rigas and C. Nikou are with the Department of Computer Science, University of Ioannina, Ioannina GR 451 10, Greece (e-mail: rigas@cs.uoi.gr; cnikou@cs.uoi.gr).

Y. Goletsis is with the Department of Economics, University of Ioannina, Ioannina GR 451 10, Greece (e-mail: goletsis@cc.uoi.gr).

D. I. Fotiadis is with Unit of Medical Technology and Intelligent Information Systems, Department of Materials Science and Engineering, University of Ioannina, Ioannina GR 451 10, Greece (e-mail: fotiadis@cs.uoi.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2267803




(a) (b) (c) (d)

Fig. 1. (a) Transformation of a 2-D mixture model with two components. (b) Fourth iteration. (c) 35th iteration. (d) 44th iteration (convergence of the method). Black circles: the first component of the initial population. Red triangles: the second one. Green squares: the first component of the transformed population. Yellow rhombs: the second one. Black and red ellipses: the estimations of the first and second Gaussian components, respectively.

(LTs). In the moving camera example, if some objects are also moving under different motion models, then to estimate both motions we need to increase the flexibility of the whole modeling. This modeling of LT could be considered in analogy with probabilistic principal component analysis (PCA) or factor analyzers where the data are described by a linear subspace and the remaining variation is captured by a spherical, diagonal, or full covariance [1]. Thus, we propose a MAP-EM approach to estimate both global and local similarity transformations.

The remainder of this paper is organized as follows. A general description of the similarity transformations between Gaussian mixtures is given in Section II. The presentation of the GT and LT models along with the estimation of parameters in a MAP-EM framework is accomplished in Section III. More specifically, the GT model is presented in Section III-A and the LT model is developed in Section III-B. The proposed mixture registration method is evaluated in Section IV, using both artificial and real data. The artificial data are used to examine the convergence of the method under different scenarios, including varying data dimensionality, number of components, and type of applied transformation. A real speech recognition data set is also used to illustrate a machine learning application of the proposed method. The obtained recognition accuracy using the proposed method is compared with a supervised classification (K-NN) and other possible GMM learning methods, such as the standard ML model inference of the GMM parameters. These results are presented and discussed in Section IV-B. Finally, in Section V, we present our conclusions and possible extensions of this paper.

## II. Similarity Transformations of GMMs

Assume an initial population $\mathbf{X}^0$ in a $D$-dimensional space $\mathbb{R}^D$, whose distribution is approximated by a GMM with $L$ components. If the elements of $\mathbf{X}^0$ are independent and identically distributed, which is a common assumption in many applications, such as image segmentation [23]–[25], then

$$P(\mathbf{X}^0) = \prod_{\mathbf{x}\in\mathbf{X}^0} \sum_{i=1}^{L} \pi_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \Sigma_i) \tag{1}$$

where $\pi_i$, $\boldsymbol{\mu}_i$, and $\Sigma_i$ are the mixing proportion, the mean vector, and covariance matrix of the $i$th mixture component, respectively. Furthermore, it holds that $\sum_{i=1}^{L} \pi_i = 1$. Let $\mathbf{X}$ be a new sample, which is generated according to the following:

$$\mathbf{X} = A\mathbf{X}^0 + \mathbf{b} \tag{2}$$

where matrix $A$ is the product of transformation matrices

$$A = S \prod_{i=1}^{P} R_i(\phi_i). \tag{3}$$

In (3), $S$ is a diagonal scaling matrix and $R_i$ is a rotation matrix representing a rotation by angle $\phi_i$ with respect to the $i$th dimension. Both scaling and rotation matrices are $n$-dimensional square matrices. The distribution of the new sample $\mathbf{X}$, under this transformation is as follows:

$$P(\mathbf{X}) = \prod_{\mathbf{x}\in\mathbf{X}} \sum_{i} \pi_i \mathcal{N}(\mathbf{x}|A\boldsymbol{\mu}_i + \mathbf{b}, A^T \Sigma_i A). \tag{4}$$

A more generic case of the above transformations, which is frequently seen in practice, is when scaling is applied only to the covariance matrices of the Gaussian distributions (the mean vector remains unchanged). Thus, the scaling transformation does not shrink or enlarge the whole distribution space but rather the range of each component, and may be written as follows:

$$P(\mathbf{X}) = \prod_{\mathbf{x}\in\mathbf{X}} \sum_{i} \pi_i \mathcal{N}(\mathbf{x}|A\boldsymbol{\mu}_i + \mathbf{b}, A^T S \Sigma_i S A) \tag{5}$$

and matrix $A$ does not depend on scaling

$$A = \prod_{i=1}^{P} R_i. \tag{6}$$

We should notice that the extension from the case in (6) to the case described by (3) is trivial. Therefore, in what follows, we focus on the transformation model described by (5). We also denote the parameters describing the original GMM model by $\mathbf{M} = \{\pi_i, \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^{L}$ and the transformation parameters by $\Theta = \{\{\phi_1, \ldots, \phi_D\}, \{s_1, \ldots, s_D\}, \{b_1, \ldots, b_D\}\}$. In the following section, we describe the estimation of the transformation parameters described in (5).

## III. Parameter Estimation

The transformation in (5) is applied to all mixture components and it may be considered as a GT. We first examine the parameter estimation problem for this case and then we extend our results for the more generic case, where apart from the GT, a separate, LT is applied to each component of the GMM.

### A. Global Transformations

As there is no closed form solution to the ML estimation of the transformation parameters $\Theta$, we recur to the EM algorithm [9]. The goal of the EM algorithm is to maximize at each step, the expected log likelihood of the complete data with respect to model's parameters

$$\Theta^k = \arg\max_{\Theta} \mathbb{E}\left[\log p(\mathbf{X}, \Omega|\Theta, \mathbf{M})|\mathbf{X}, \Theta^{\mathbf{k-1}}\right] \tag{7}$$

where $\Theta^{k-1}$ is the previous estimation of parameters, $\mathbf{M}$ is the initial model, and $\Omega$ is the collection of the corresponding unobserved information, $\omega_i$, which states that sample $x$ is generated by the $i$th component.

The expected log likelihood is as follows:

$$\begin{aligned}\mathcal{L}(\Theta^k|\Theta^{k-1}) &= \mathbb{E}[\log p(\mathbf{X}, \Omega|\Theta, \mathbf{M})|\mathbf{X}, \Theta^{k-1}] \\ &= \sum_{x\in\mathbf{X}}\sum_i p(\omega_i|x, \Theta^{k-1})[\log p(x|\omega_i, \Theta) \\ &\qquad + \log p(\omega_i|\Theta)] \\ &= \sum_{x\in\mathbf{X}}\sum_i p(\omega_i|x, \Theta^{k-1}) \qquad (8) \\ &\quad \cdot\left[-\frac{1}{2}\log|\Sigma_i'| - \frac{1}{2}(x-\mu_i')^T(\Sigma_i')^{-1}(x-\mu_i')\right] \\ &\quad + \sum_{x\in\mathbf{X}}\sum_i p(\omega_i|x, \Theta^{k-1})\log p(\omega_i|\Theta) \qquad (9)\end{aligned}$$

where $\mu_i' = A\mu_i + b$ and $\Sigma_i' = A^T S\Sigma_i SA$. The last term in (9) does not depend on the transformation parameters $\Theta$ and may be omitted. The first term in (9), denoted as $\mathfrak{J}$, following [19], may be written as follows:

$$\begin{aligned}\mathfrak{J} = \sum_i \boldsymbol{\pi}_i(\mathbf{X})\Big[&2\log|A| + \boldsymbol{y}^T S^{-1}\Sigma_i^{-1}S^{-1}\boldsymbol{y} \\ &+\mathrm{tr}\left[AS^{-1}\Sigma_i^{-1}S^{-1}A^T\hat{\Sigma}_i[\mathbf{X}]\right]\Big]\end{aligned} \tag{10}$$

where $\boldsymbol{y} = (A^T\mathbb{E}_i[\mathbf{X}] - \boldsymbol{\mu}_i - A^T\mathbf{b})$ and $\boldsymbol{\pi}_i(\mathbf{X})$ is

$$\boldsymbol{\pi}_i(\mathbf{X}) = \sum_{\mathbf{x}\in\mathbf{X}} h_i(\mathbf{x}) \tag{11}$$

where

$$h_i(\mathbf{x}) \equiv p(\omega_i|x, \Theta^{k-1}) = \frac{\pi_i P(\mathbf{x}|\boldsymbol{\mu}_i', \Sigma_i')}{\sum_{j=1}^L \pi_j P(\mathbf{x}|\boldsymbol{\mu}_j', \Sigma_j')}. \tag{12}$$

$\mathbb{E}_i[\mathbf{X}]$ and $\hat{\Sigma}_i[\mathbf{X}]$ are the sufficient statistics calculated in the **E**-step of the EM algorithm. The derivation of E-step and M-step, for the estimation of the parameter vector $\Theta$, is described in more detail in the following.

*E-Step:* In the E-step, we estimate the expected sufficient statistics of the data, given the current estimation of parameter vector $\Theta^{k-1}$ as follows:

$$\mathbb{E}_j[\mathbf{X}] = \frac{\sum_{i=1}^N \mathbf{x}_i\, p(\mathbf{x})_{ij}}{\sum_{i=1}^N\sum_{k=1}^L p(\mathbf{x})_{ik}} \tag{13}$$

$$\mathbb{E}_j[\mathbf{X}\mathbf{X}^T] = \frac{\sum_{i=1}^N \mathbf{x}_i\mathbf{x}_i^T\, p(\mathbf{x})_{ij}}{\sum_{i=1}^N\sum_k p(\mathbf{x})_{ik}} \tag{14}$$

$$\hat{\Sigma}_i[\mathbf{X}] = \mathbb{E}_j[\mathbf{X}\mathbf{X}^T] - \mathbb{E}_j[\mathbf{X}]\mathbb{E}_j[\mathbf{X}]^T \tag{15}$$

$$p(\mathbf{x})_{ij} = P(\mathbf{x}_i|A\boldsymbol{\mu}_j + \mathbf{b}, (SA)^T\Sigma_j SA).$$

*M-Step:* The update equations for all the transformation parameters (translation vector, scale factors, and rotation angles) are employed in the M-step.

*Mixing Coefficients*: The ML estimation of the mixture proportions is given as follows:

$$\pi_i^k = \frac{\boldsymbol{\pi}_i(\mathbf{X})}{N}. \tag{16}$$

*Translation Vector*: Taking the derivative of (10) with respect to the translation vector $\mathbf{b}$ and setting the derivative to zero, we obtain the following update equation:

$$\mathbf{b} = C^{-1}D \tag{17}$$

where $C$ and $D$ are defined as follows:

$$C = \left[\sum_i \boldsymbol{\pi}_i(\mathbf{X})\Sigma_i^{-1}S^{-1}A^T\right] \tag{18}$$

$$D = \left[\sum_i \boldsymbol{\pi}_i(\mathbf{X})\Sigma_i^{-1}S^{-1}A^T(\mathbb{E}_i[\mathbf{X}] - A\boldsymbol{\mu}_i)\right]. \tag{19}$$

*Scale Factors:* Taking the derivative of (10) with respect to $\mathbf{s}_j$ and setting the derivative to zero, we obtain the following:

$$a\mathbf{s}_j^2 - b\mathbf{s}_j - c = 0 \tag{20}$$

where $a$, $b$, and $c$ are defined as follows:

$$\begin{aligned}a &= \sum_i \frac{\boldsymbol{\pi}_i(\mathbf{X})}{N} = 1 \\ b &= \sum_i \frac{\boldsymbol{\pi}_i(\mathbf{X})}{N}\left\{\boldsymbol{x}_{ij}^T\boldsymbol{y}_{ij} + \mathrm{tr}(AS^{1j}\Sigma_i S^{2j}A^T\hat{\Sigma}_i[\mathbf{X}])\right\} \\ c &= \sum_i \frac{\boldsymbol{\pi}_i(\mathbf{X})}{N}\left\{\boldsymbol{x}_{ij}^T\boldsymbol{x}_{ij} + \mathrm{tr}(AS^{2j}\Sigma_i S^{2j}A^T\hat{\Sigma}_i[\mathbf{X}])\right\}.\end{aligned}$$

In the above equations, $\boldsymbol{x}_{ij}$ and $\boldsymbol{y}_{ij}$ are defined as follows:

$$\boldsymbol{x}_{ij} = L_i S^{1j}\left[A^T(\mathbb{E}_i[\mathbf{X}] - \mathbf{b}) - \boldsymbol{\mu}_i\right] \tag{21}$$

$$\boldsymbol{y}_{ij} = L_i S^{2j}\left[A^T(\mathbb{E}_i[\mathbf{X}] - \mathbf{b}) - \boldsymbol{\mu}_i\right] \tag{22}$$

where $L_i$ is derived from the Cholesky decomposition of the covariance matrix $\Sigma_i^{-1}$ as follows:

$$\Sigma_i^{-1} = L_i^T L_i \tag{23}$$

and

$$S_{ij}^{1k} = \begin{cases} 0, & \text{if } i = j = k \\ s_i, & \text{if } i = j \neq k \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

$$S_{ij}^{2k} = \begin{cases} 1, & \text{if } i = j = k \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

As $c$ is always positive, (20) has one positive and one negative solution and we select the positive one.

*Rotation Angles*: The derivation of the angle update formula is more involved. For simplicity, (10) may be expressed by the sum of two terms as follows:

$$\mathfrak{J} = \sum_i \boldsymbol{\pi}_i(\mathbf{X})[\mathfrak{J}_i^a + \mathfrak{J}_i^b] \tag{26}$$

where

$$\mathfrak{J}_i^a = \mathbf{F}^T S^{-1} S_i^{-1} S^{-1} \mathbf{F}, \tag{27}$$

$$\mathfrak{J}_i^b = \operatorname{tr}(A S^{-1} \Sigma_i^{-1} S^{-1} A^T \hat{\Sigma}_i[\mathbf{X}]) \tag{28}$$

and

$$\mathbf{F} = (A^{-1}\mathbb{E}_i[\mathbf{X}] - \boldsymbol{\mu}_i - A^{-1}\mathbf{b})^{\mathbf{T}}. \tag{29}$$

Now consider matrix $A$ to be a product of elementary transformations [26]

$$A = (R_{12}R_{13}\cdots R_{1n})(R_{23}\cdots R_{2n})\cdots(R_{D-1,D}) \tag{30}$$

where $R_{ij}$ is a $(n \times n)$ matrix representing the rotation across the plane produced by dimensions $i$ and $j$. Then, $R_{ij}$ has the following form:

$$\begin{array}{c} \quad i\text{th col.} \quad j\text{th col.} \\ \quad\quad\uparrow\quad\uparrow \\ \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & \cos(\phi_{ij}) & 0 & -\sin(\phi_{ij}) & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & \sin(\phi_{ij}) & 0 & \cos(\phi_{ij}) & 0 \\ 0 & 0 & 0 & 0 & I \end{pmatrix} \begin{array}{l} \\ \rightarrow i\text{th row} \\ \\ \rightarrow j\text{th row} \\ \\ \end{array} \end{array} \tag{31}$$

where $\phi_{ij}$ is the rotation angle. $R_{ij}$ may be further decomposed as follows:

$$R_{ij} = (I_{ij}\cos(\phi_{ij}) + J_{ij}\sin(\phi_{ij}) + K_{ij}) \tag{32}$$

where $I_{ij}$, $J_{ij}$, and $K_{ij}$ are defined in Appendix A. The partial derivative of $\mathfrak{J}$ in (26) with respect to $\phi_{ij}$ is given by (see Appendix A for a more detailed derivation)

$$\begin{aligned} \frac{\partial \mathfrak{J}}{\partial \phi_{ij}} = \sum_i \pi_i(\boldsymbol{x}) \Big[ \Big[ & (\boldsymbol{x}_{ij}^T \boldsymbol{x}_{ij} - \boldsymbol{y}_{ij}^T \boldsymbol{y}_{ij}) \\ & + \operatorname{tr}(A_j^c \Sigma_i^{-1} (A_j^c)^T \hat{\Sigma}_i[\mathbf{X}]) \\ & - \operatorname{tr}(A_j^s \Sigma_i^{-1} (A_j^s)^T \hat{\Sigma}_i[\mathbf{X}]) \Big] \sin(2\phi_{ij}) \\ & + 2[\boldsymbol{x}_{ij}^T \boldsymbol{y}_{ij} + \operatorname{tr}(A_j^s S_i^{-1} (A_j^c)^T \hat{\Sigma}_i[\mathbf{X}])] \cos(2\phi_{ij}) \\ & + 2[\boldsymbol{x}_{ij}^T (\boldsymbol{z}_{ij} + \boldsymbol{w}_i)] \sin(\phi_{ij}) - 2[\boldsymbol{y}_{ij}^T (\boldsymbol{z}_{ij} + \boldsymbol{w}_i)] \cos(\phi_{ij}) \Big]. \end{aligned} \tag{33}$$

where $\boldsymbol{x}_{ij} = L_i^T A_j^c(\mathbb{E}_i[\mathbf{X}] - \mathbf{b})$, $\boldsymbol{y}_{ij} = L_i^T A_j^s(\mathbb{E}_i[\mathbf{X}] - \mathbf{b})$, $\boldsymbol{z}_{ij} = L_i^T A_j^k(\mathbb{E}_i[\mathbf{X}] - \mathbf{b})$ and $\boldsymbol{w}_i = L_i^T \boldsymbol{\mu}_i$. $L_i$ is the lower triangular matrix of the Cholesky decomposition of the covariance matrix $\Sigma_i^{-1}$ and

$$A_j^c = R_{D-1}^T \cdots I_j^T \cdots R_2^T R_1^T S^{-1} \tag{34}$$

$$A_j^s = R_{D-1}^T \cdots J_j^T \cdots R_2^T R_1^T S^{-1} \tag{35}$$

$$A_j^k = R_{D-1}^T \cdots K_j^T \cdots R_2^T R_1^T S^{-1}. \tag{36}$$

Setting (33) to zero, we obtain (see Appendix A)

$$a\cos(2\phi) + b\sin(2\phi) + c\cos(\phi) + d\sin(\phi) = 0 \tag{37}$$

where

$$\begin{aligned} a &= \sum_i \pi_i(\boldsymbol{X}) \Big[ (\boldsymbol{x}_{ij}^T \boldsymbol{x}_{ij} - \boldsymbol{y}_{ij}^T \boldsymbol{y}_{ij}) + \operatorname{tr}(A_j^c \Sigma_i^{-1} (A_j^c)^T \hat{\Sigma}_i[\mathbf{X}]) \\ &\qquad - \operatorname{tr}(A_j^s \Sigma_i^{-1} (A_j^s)^T \hat{\Sigma}_i[\mathbf{X}]) \Big] \\ b &= 2\sum_i \boldsymbol{\pi}_i(\mathbf{X}) [\boldsymbol{x}_{ij}^T \boldsymbol{y}_{ij} + \operatorname{tr}(A_j^s \Sigma_i^{-1} (A_j^c)^T \hat{\Sigma}_i[\mathbf{X}])] \\ c &= 2\sum_i \pi_i(\mathbf{X}) [\boldsymbol{x}_{ij}^T (\boldsymbol{z}_{ij} + \boldsymbol{w}_i)] \sin(\phi_j) \\ d &= -2\sum_i \boldsymbol{\pi}_i(\mathbf{X}) [\boldsymbol{y}_{ij}^T (\boldsymbol{z}_{ij} + \boldsymbol{w}_i)] \cos(\phi_j). \end{aligned}$$

To solve (37) for $\phi$, we use a nonlinear optimization method (Levenberg–Marquardt). It is, however, important to notice, that in cases where angles are expected to be small, small angle approximations lead to closed form solutions. For angles up to 10° approximately, we could use the approximations $\sin(x) \approx x$ and $\cos(x) \approx 1$, resulting in the following:

$$\phi = -\frac{c + a}{2b + d}. \tag{38}$$

### *B. Local Transformations*

The assumption of a unique transformation applied to all mixture model components, may hold for some problems, but in many cases it could be a very strict constraint. To add more flexibility to the model, we could allow each component to have an individual LT as well. For clarity of presentation, we will consider the 2-D case, but the following results may be extended to larger dimensions. We start from the definition of both GT and LT in rotation, scaling, and translation. The rotation matrix $R_j^{(GL)}$ for a 2-D case can be written as follows:

$$R_j^{(GL)} = \begin{pmatrix} \cos(\phi + \phi_j) & -\sin(\phi + \phi_j) \\ \sin(\phi + \phi_j) & \cos(\phi + \phi_j) \end{pmatrix} \tag{39}$$

where $\phi$ is the rotation applied to all components (global) and $\phi_j$ the rotation applied to the specific $j$th component (local). Matrix $R_j$ can also be expressed as the product of a local and a global rotation

$$\begin{aligned} R_j^{(GL)} &= R^{(G)} \cdot R_j^{(L)} \\ R^{(G)} &= \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \\ R_j^{(L)} &= \begin{pmatrix} \cos(\phi_j) & -\sin(\phi_j) \\ \sin(\phi_j) & \cos(\phi_j) \end{pmatrix}. \end{aligned} \tag{40}$$

For the translation vector, we can write the following:

$$b_j^{(GL)} = b^{(G)} + b_j^{(L)} \tag{41}$$

and for the scaling factors

$$S_j^{(GL)} = \begin{pmatrix} s_1^{(G)} \cdot s_{1j}^{(L)} & 0 \\ 0 & s_2^{(G)} \cdot s_{2j}^{(L)} \end{pmatrix}. \quad (42)$$

The basic assumption for the LT parameters is that they are in magnitude smaller than the global ones. Thus, a prior distribution is imposed on each one of these parameters as follows:

$$\phi_j^{(L)} \propto \mathcal{N}(0, \sigma_\phi^2), \quad s_{kj}^{(L)} \propto \mathcal{N}(1, \sigma_s^2), \quad \mathbf{b} \propto \mathcal{N}(\mathbf{0}, \sigma_{\mathbf{b}}^2 \mathbf{I}) \quad (43)$$

where $\sigma_\phi^2, \sigma_s^2$, and $\sigma_{\mathbf{b}}^2$ are the variances of the respective distributions. The new set of transformation parameters and the incorporation of the prior on the distribution of LTs lead to a MAP-EM estimation for the optimization of the new complete data log likelihood

$$\begin{aligned} \mathfrak{J} = \sum_i \boldsymbol{\pi}_i(\mathbf{X}) \Big[ & 2\log|A_i^{(GL)}| + \boldsymbol{y}^T (S_i^{(GL)})^{-1} \Sigma_i^{-1} (S_i^{(GL)})^{-1} \boldsymbol{y} \\ & + \operatorname{tr}\big[A_i^{(GL)} (S_i^{(GL)})^{-1} \Sigma_i^{-1} (S_i^{(GL)})^{-1} (A_i^{(GL)})^T \hat{\Sigma}_i[\mathbf{X}]\big]\Big] \\ & - \log c_1 + \sum_{j=1}^{P} \lambda_\phi (\phi_{ij}^{(L)})^2 - \log c_2 + \sum_{j=1}^{D} \lambda_s (1 - s_{ij}^{(L)})^2 \\ & - \log c_3 + \lambda_{\mathbf{b}} \mathbf{b}^T \mathbf{b} \end{aligned} \quad (44)$$

with respect to both local and GT parameters. The constants $c_1$, $c_2$, and $c_3$ do not depend on the transformation parameters, $P$ is the number of angles in the transformation, $D$ is the dimensionality, $\lambda_\phi = 1/(2\sigma_\phi^2)$, $\lambda_s = 1/(2\sigma_s^2)$, and $\lambda_{\mathbf{b}} = 1/(2\sigma_{\mathbf{b}}^2)$. Furthermore, $\boldsymbol{y} = [(A_i^{(GL)})^T \mathbb{E}_i[\mathbf{X}] - \boldsymbol{\mu}_i - (A_i^{(GL)})^T \mathbf{b}]$.

*E-Step:* The expectations $\mathbb{E}_i[\mathbf{X}]$ and $\mathbb{E}_i[\mathbf{X}\mathbf{X}^T]$ calculated in the E-step, are now given by

$$\mathbb{E}_j[\mathbf{X}] = \frac{\sum_{i=1}^n \mathbf{x}_i \hat{p}(\mathbf{x})_{ij}}{\sum_{i=1}^n \sum_{k=1}^L \hat{p}(\mathbf{x})_{ik}} \quad (45)$$

$$\mathbb{E}_j[\mathbf{X}\mathbf{X}^T] = \frac{\sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T \hat{p}(\mathbf{x})_{ij}}{\sum_{i=1}^n \sum_{k=1}^L \hat{p}(\mathbf{x})_{ik}} \quad (46)$$

$$\Gamma_i \equiv S_i^{(GL)} A_i^{(GL)} \quad (47)$$

$$\hat{p}(\mathbf{x})_{ij} = P(\mathbf{x}_i | A_i^{(GL)} \boldsymbol{\mu}_j + \mathbf{b}, \Gamma_i^T \Sigma_j \Gamma_i).$$

*M-Step:* The update equations for the GT parameters, are similar to refUpdateMixing), (17), (20), and (37), presented in Section III-A. The only difference lies on the definition of matrices $A_j^c$, $A_j^s$, and $A_j^k$

$$\begin{aligned} A_{ji}^c &= (R_{P,i}^{(GL)})^T \cdots I_j^T (R_{j,i}^{(L)})^T \cdots (R_{1,i}^{(GL)})^T (S_i^{(GL)})^{-1} \\ A_{ji}^s &= (R_{P,i}^{(GL)})^T \cdots J_j^T (R_{j,i}^{(L)})^T \cdots (R_{1,i}^{(GL)})^T (S_i^{(GL)})^{-1} \\ A_{ji}^k &= (R_{P,i}^{(GL)})^T \cdots K_j^T (R_{j,i}^{(L)})^T \cdots (R_{1,i}^{(GL)})^T (S_i^{(GL)})^{-1} \end{aligned}$$

where $j$ is the $j$th rotation angle, whereas $i$ is the $i$th component.

Furthermore, in all the equations presented for the GT, we should interchange the transformation parameters with the new transformation containing both local and GT. We proceed with the update equations of the LTs.

*Translation Vector*: Taking the derivative of the log likelihood (44) with respect to the translation vector $\mathbf{b}^L$ and setting it to zero, we obtain the following update equation:

$$\mathbf{b}_i^L = (C_i + \lambda_{\mathbf{b}} I)^{-1} D_i \quad (48)$$

where $C_i$ and $D_i$ are defined as follows:

$$C_i = \left[\Sigma_i^{-1} \Gamma_i^{-1}\right] \quad (49)$$

$$D_i = \left[\Sigma_i^{-1} \Gamma_i^{-1} (\mathbb{E}_i[\mathbf{X}] - A_i^{(GL)} \boldsymbol{\mu}_i - \mathbf{b}^{(\mathbf{G})})\right] \quad (50)$$

where $\Gamma_i$ defined in (47). The difference between (48) and (17), is the introduced term $\lambda_{\mathbf{b}} I$. The larger the $\lambda_{\mathbf{b}}$, the smaller the norm of $\mathbf{b}^L$.

*Scale Factors:* Taking the derivative of the log likelihood with respect to the scale $\mathbf{s}_{ij}^{(L)}$ of the $i$th component at $j$th dimension and setting the derivative to zero, we obtain the following:

$$(\mathbf{s}_{ij}^{(L)})^2 - (b - \lambda_s)\mathbf{s}_{ij}^{(L)} - (c + \lambda_s) = 0 \quad (51)$$

where $b$ and $c$ are defined as follows:

$$b = \boldsymbol{y}_{ij}^T \boldsymbol{y}_{ij} + \operatorname{tr}[A_i^{(GL)} S_i^{1j} \Sigma_j S_i^{2j} (A_i^{(GL)})^T \hat{\Sigma}_i[\mathbf{X}]] \quad (52)$$

$$c = \boldsymbol{x}_{ij}^T \boldsymbol{x}_{ij} + \operatorname{tr}[A_i^{(GL)} S_i^{2j} \Sigma_j S_i^{2j} (A_i^{(GL)})^T \hat{\Sigma}_i[\mathbf{X}]]. \quad (53)$$

In the above equations, $\boldsymbol{x}_{ij}$ and $\boldsymbol{y}_{ij}$ are defined as follows:

$$\boldsymbol{x}_{ij} = L_i S_i^{1j} \left[(A_i^{(GL)})^T \Delta_i - \boldsymbol{\mu}_i\right] \quad (54)$$

$$\boldsymbol{y}_{ij} = L_i S_i^{2j} \left[(A_i^{(GL)})^T \Delta_i - \boldsymbol{\mu}_i\right] \quad (55)$$

$$\Delta_i = \mathbb{E}_i[\mathbf{X}] - \mathbf{b}_i^{GL} \quad (56)$$

$$(S_p^{1k})_{ij} = \begin{cases} s_{kp}^{GL}, & \text{if } i = j \neq k \\ 0, & \text{otherwise} \end{cases} \quad (57)$$

$$(S_p^{2k})_{ij} = \begin{cases} s_{kp}^{G}, & \text{if } i = j = k \\ 0, & \text{otherwise.} \end{cases} \quad (58)$$

Similar to (20) and (51) has two possible solutions for $\mathbf{s}_{ij}^{(L)}$, one negative and one positive and we again select the positive one.

To clarify the impact of the term $\lambda_s$ on the values obtained for $\mathbf{s}_{ij}^{(L)}$, solving (51), consider $\lambda_s >> 1, \lambda_s >> b$, and $\lambda_s >> c$. Then, we can neglect the quadratic term in (51) and $\mathbf{s}_{ij}^{(L)} \approx 1$.

*Rotation Angles:* In a derivation similar with the one for the GT, we obtain an equation of the following form:

$$\begin{aligned} a\cos(2\phi_{ij}^{(L)}) + b\sin(2\phi_{ij}^{(L)}) + c\cos(\phi_{ij}^{(L)}) \\ + d\sin(\phi_{ij}^{(L)}) + \lambda_\phi \phi_{ij}^{(L)} = 0 \end{aligned} \quad (59)$$

where

$$\begin{aligned} a = \big[ & (\boldsymbol{x}_{ij}^T \boldsymbol{x}_{ij} - \boldsymbol{y}_{ij}^T \boldsymbol{y}_{ij}) + \operatorname{tr}(A_j^c \Sigma_i^{-1} (A_j^c)^T \hat{\Sigma}_i[\mathbf{X}]) \\ & + \operatorname{tr}(A_j^s \Sigma_i^{-1} (A_j^s)^T \hat{\Sigma}_i[\mathbf{X}])\big] \end{aligned} \quad (60)$$

$$b = 2[\boldsymbol{x}_{ij}^T \boldsymbol{y}_{ij} + \operatorname{tr}(A_j^s S_i^{-1} (A_j^c)^T \hat{\Sigma}_i[\mathbf{X}])] \quad (61)$$

$$c = 2[\boldsymbol{x}_{ij}^T (\boldsymbol{z}_{ij} + \boldsymbol{w}_i)] \sin(\phi_j) \quad (62)$$

$$d = -2[\boldsymbol{y}_{ij}^T (\boldsymbol{z}_{ij} + \boldsymbol{w}_i)] \cos(\phi_j) \quad (63)$$

**Algorithm 1** Global/Local MAP-EM

**while** $\|\Theta^k - \Theta^{k-1}\| < \epsilon$ **do**
**E**-step: Calculate expected statistics from (45) and (46).
**M**-step:
Update all global rotations $\phi_i^{(G)}$ using the solution of (37).
Update all global scale coefficients $s_i^{(G)}$ using (20).
Update all local rotations $\phi_{ij}^{(L)}$ using the solution of (59).
Update all local scale coefficients $s_{ij}^{(L)}$ using the positive solution of (51).
Update global translation $\mathbf{b}^{(\mathbf{G})}$ using (17).
Update local translations $\mathbf{b}_\mathbf{i}^{(\mathbf{L})}$ using (48).
$k = k + 1$
**end while**

and

$$\boldsymbol{x}_{ij} = L_i^T A_j^c(\mathbb{E}_i[\mathbf{X}] - \mathbf{b}_i^{GL}) \tag{64}$$

$$\boldsymbol{y}_{ij} = L_i^T A_j^s(\mathbb{E}_i[\mathbf{X}] - \mathbf{b}_i^{GL}) \tag{65}$$

$$\boldsymbol{z}_{ij} = L_i^T A_j^k(\mathbb{E}_i[\mathbf{X}] - \mathbf{b}_i^{GL}) \tag{66}$$

$$\boldsymbol{w}_i = L_i^T \boldsymbol{\mu}_i. \tag{67}$$

Comparing (59) with (37), the additional term $\lambda_\phi \phi_{ij}^{(L)}$, forces $\phi_{ij}^{(L)}$ to zero, for large values of $\lambda_\phi$. Furthermore, we should also notice that setting $\lambda_b$, $\lambda_s$, and $\lambda_\phi$ coefficients to zero in (48), (51), and (59), we obtain equations of the same form as (17), (20), and (37), respectively. The statistics used in the former equations are, however, calculated regarding only the specific component. The MAP-EM algorithm estimating both GT and LT is given in Algorithm 1.

The variance in the prior distribution (43) of the transformation parameters is the parameter governing the freedom given to the model. If the variance is large then the model diverges from the initial assumption of a geometric transformation of an initial population. Each component will probably converges to the closest component of the new population. On the other hand, using a very small initial variance, we restrict the model and we are neglecting the LT parameters. A desired behavior would be, if initially we restrict the LT parameters and progressively, while the GT converges to the actual solution, loose the constraints and allow the learning of the local parameters as well. This behavior can be achieved using a parameter $\lambda$ varying across iterations

$$\lambda_k = \lambda_0 \exp(-\gamma k) + \lambda_0'. \tag{68}$$

Parameter $\gamma$ controls the behavior of the algorithm, outlined in Algorithm 1. If $\gamma$ is large then the constraints are quickly dropped and this algorithm is equivalent to an algorithm where at each iteration we seek for both LT and GT parameters without constraints. On the other hand, if $\gamma$ is very small and we also imply large constraints $\lambda_0$ the algorithm initially fits the global transformation, and after a number of iterations fits the local parameters.

## IV. Results and Discussion

### A. Experiments Using Artificial Data

A set of experiments is performed using artificial data to investigate whether the proposed method is able to detect the correct solution for different scenarios. To better understand the method's behavior, we present the estimation of the new mixture, for a 2-D problem (for visualization purposes) having two components (Fig. 1). Three different iteration steps of the EM algorithm are presented: 1) the 4th step in Fig. 1; 2) the 35th step in Fig. 1; and 3) the 44th step in Fig. 1. From an initially distant starting point, the algorithm correctly identifies the correct solution.

At first, we examine the estimation of GT parameters under: 1) different overlapping conditions and 2) different rotation angles and number of components of the GMM. The results are compared with those obtained using the standard EM approach. Next, we compare the performance of the simultaneous GT and LT approaches with the performance of the GT and EM, in the presence of LTs.

*1) Overlapping Components:* As EM is a local optimization method, the existence of many local maxima could have a serious effect on the method performance. In the case of a unique maximum, the algorithm is guaranteed to convergence to this maximum. Thus, it is of great interest to study the shape of the likelihood function, given the parameters of the problem.

One of the factors affecting the number of local maxima of the log-likelihood function is the sparsity of the mixture components. This was verified theoretically in [27]–[29]. The more dense the distribution, the larger the number of local maxima in the search of the optimal transformation parameters that fit the data. This factor is also investigated in our experiments. A measure of overlapping is, however, necessary to quantitatively relate component overlap and registration accuracy. Therefore, we use the measure of the Gaussian overlap introduced in [28]. Initially, $\gamma_{ij}(\mathbf{x})$ is defined as $\gamma_{ij}(\mathbf{x}) = (\delta_{ij} - h_i(\mathbf{x}))h_j(\mathbf{x})$ for $i, j = \{1, ..., L\}$, where $\delta_{ij}$ is the Kronecker function and $h_i(\mathbf{x})$ is defined in (12). The overlap measure of two mixture components is defined as follows:

$$e_{ij}(\mathbf{M}) = \int_{\mathbf{R}^d} |\gamma_{ij}(x)| P(\mathbf{x}|\mathbf{M}) dx \quad \text{for} \quad i, j = \{1, ..., L\} \tag{69}$$

where $\mathbf{M}$ is the GMM model considered and $e_{ij}(\mathbf{M}) \leq 1$ as $|\gamma_{ij}(x)| \leq 1$. The maximum overlapping $e(\mathbf{M})$, is defined as follows:

$$e(\mathbf{M}) = \max_{ij}\{e_{ij}(\mathbf{M})\}. \tag{70}$$

More details can be found in [28]. A number of experiments are conducted to examine the impact of overlapping mixture components on the estimation of the transformation parameters. We examine 2-D, 3-D, 4-D, and 5-D problems. The number of components is set to $2^{D-2} + 1$, where $D$ is the dimensionality of the problem. To control the overlapping between components, we produce a grid on the $\mathbb{R}^D$ dimensional space, where each cell of the grid is located at distance $\alpha$ from the nearest adjacent cell. Then, the mean of each

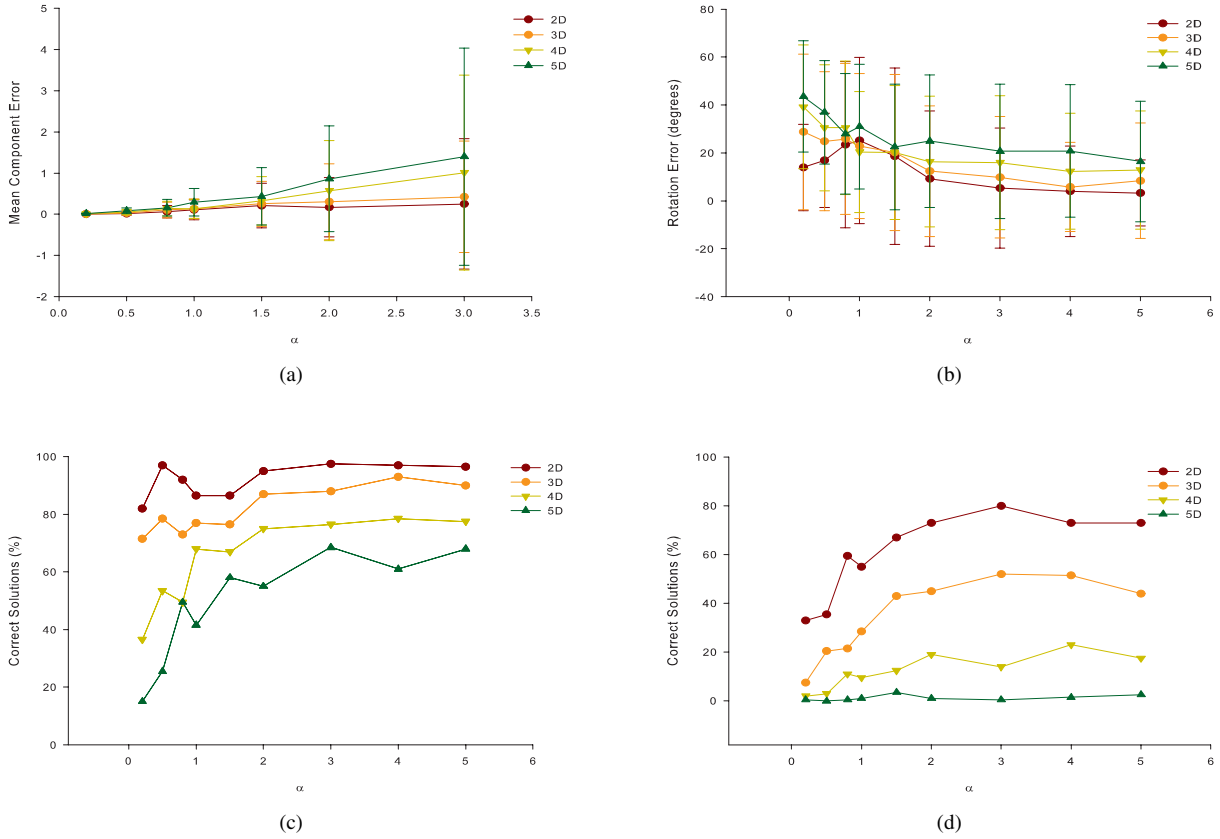



Fig. 2. (a) Mean component error, (b) rotation error, and (c) percentage of correct solution identification as a function of overlapping parameter $a$ for the GT approach. (d) Percentage of correct solution identification using the standard EM algorithm as a function of overlapping parameter $a$.

component is assigned randomly to a specific cell's centroid. The covariance for the $D$-dimensional problem is a Toeplitz matrix produced by considering the first $D$ elements from vector $v = [1, -0.2, -0.1, 0, 0]$.

The initial rotation angles are set randomly in the interval $[-\pi/4, \pi/4]$ and each dimension of the translation vector is also randomly chosen in $[-5, 5]$. The scale matrix is always the identity matrix, to maintain the same overlapping in the transformed space. Furthermore, to have similar component overlap $e(\mathbf{M})$ for all the examined dimensions (2-D, 3-D, 4-D, and 5-D), each component has $100 \cdot \pi^{(D-2)/(2)} \cdot 2^{D-2}$ samples, where $D$ is the dimensionality. This number of samples gives a constant number of samples per volume for any $D$-dimensional Gaussian distribution. A total of 200 realizations of the experiment for each dimensionality and for each value of $\alpha$ are performed. The examined values of parameter $\alpha$ are $0.5, 1, 1.5, 2, 3, 4$, and 5. For $\alpha = 0.5$, there is high overlapping between components, whereas for $\alpha = 5$, the overlapping tends to zero. The measure used for evaluation is the mean absolute error in the estimated rotation angles, scale, and translation as well as the mean squared difference between the estimated means of the components and the ground truth.

In Fig. 2, the mean and standard deviation for the error in the component's mean (Fig. 2) and rotation angle (Fig. 2), are shown. The average and standard deviation are calculated in the cases, where the solution is correctly identified. A solution is correct if each component of the estimated GMM is closest to the corresponding component of the true GMM. If this is the case, we claim that a solution is correctly identified. The percentage of the times where the corrected solution are identified, is also shown in Fig. 2. For comparison purposes, the correct solution identification, using the standard EM algorithm, is shown in Fig. 2. For $a$ larger than 0.5, when the components are well separated (given the specific covariance matrix), the percentage of correct solution identification increases drastically for the GT approach, especially for higher order dimensions. The increase is significantly smaller for the standard EM algorithm.

*2) Rotation and Number of Components:* For the cases of 2-D, 3-D, 4-D, and 5-D dimensional problems, we investigate the impact of the rotation angle and the number of components on the estimation of the correct solution. We choose only to examine the rotation angle as its estimation is the hardest one and seems to govern the correct solution identification, compared with the rest of the transformation parameters. The number of components is also a crucial parameter with a twofold contradictory role. On the one hand, the larger the number of components, the larger the number of points available to estimate the transformation's parameters are. On the other hand, the larger the number of components, the larger the degree of overlap is, considering that all components are spread at a restricted space, making parameter estimation more difficult. The initialization of GMM's components is similar

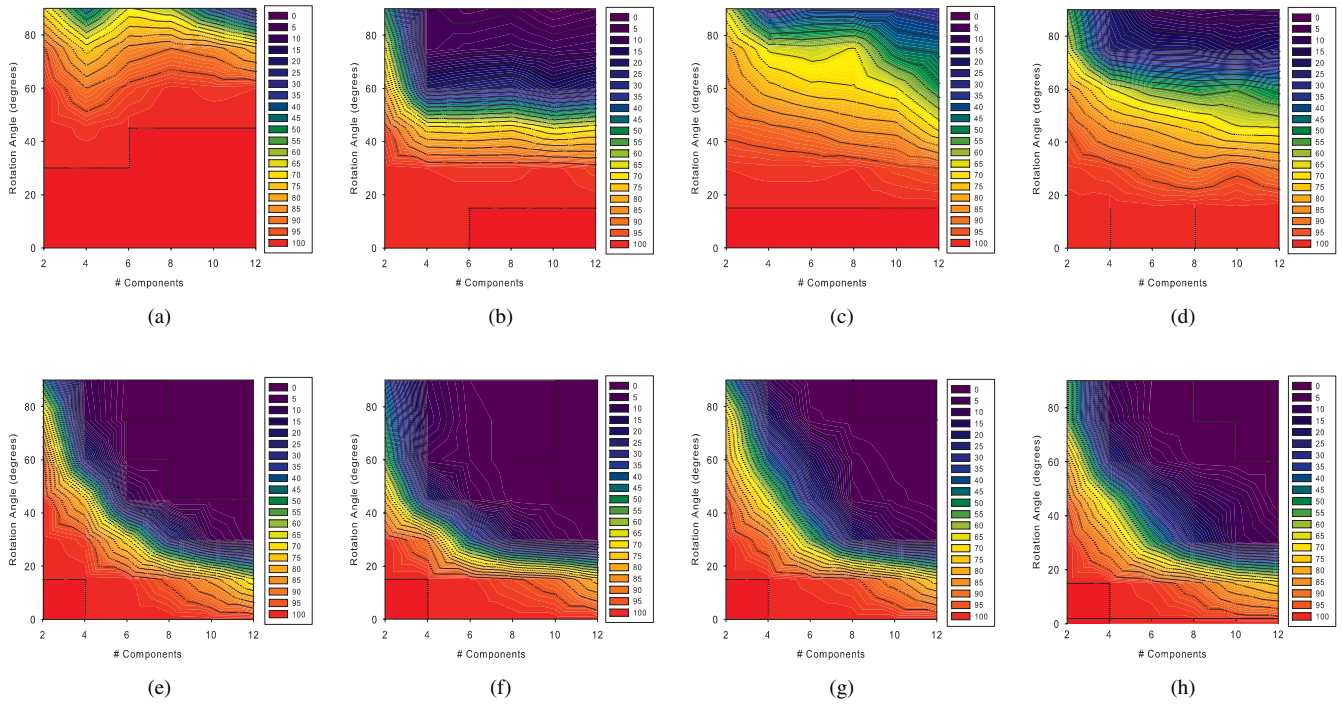



Fig. 3. Percentage of correct identification and mean component error for 2-D, 3-D, 4-D, and 5-D problems for (a)–(d) GT/LT and (e)–(h) standard EM algorithm.

to that described in the previous set of experiments. In all cases, the parameter $\alpha$ is set to 3. The number of components examined varies from 2 to 12. The rotation angles vary from 0 to $\pi/2$ with a step of $\pi/12$. In Fig. 3, the percentage of correct identification in 200 realizations in all dimensions for both GT and classical EM approaches, are presented. In all problems, we observe that with an increased number of components, resulting in a higher component overlapping, the probability of correct solution identification is reduced. For the cases where the solution was, however, correctly identified the error in the parameters estimation is reduced. This is also shown in Fig. 4 where the rotation error is averaged only on the cases where the solution is correctly identified. The results from the experiments with rotation angles set to $\pi/4$, however similar are the results for any other angles, are shown in Fig. 4.

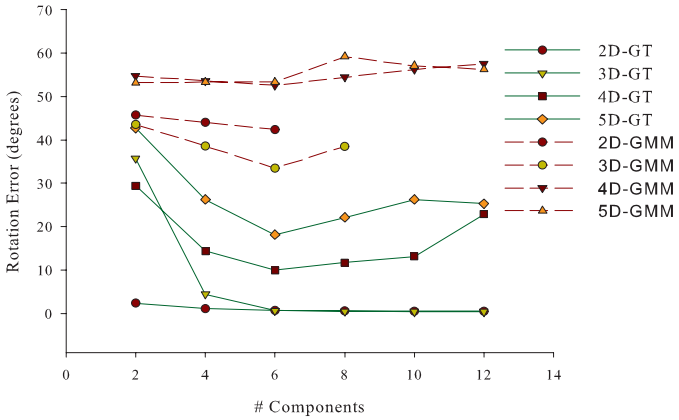



Fig. 4. Rotation error versus the number of components for 2-D, 3-D, 4-D, and 5-D problems for GT approach and standard EM algorithm, averaged on the cases where a correct solution is identified. The angle for all cases is $\pi/4$. For the standard EM algorithm, where no correct solution is identified, points are missing.

Finally, GT approach outperforms the standard EM algorithm for larger rotations, which is shown in Figs. 3 and 4. This finding leads to the conclusion that the proposed method has very good performance when many sparse (non overlapping) components are present. It should be noticed that a very sparse distribution of components leads to a performance degradation, as large gaps in the distribution space are translated into wide valleys in the likelihood function, which slows down the convergence or even trap the EM algorithm in local maxima.

*3) GT/LT Approaches:* The convergence of the proposed method for estimating both GT and LT is also examined. The main question posed here is if using GT/LT approach, we can correctly identify cases where simple GT fails. To examine this issue, we follow the experimental setting described in Section IV-A1, for 2-D, 3-D, 4-D, and 5-D problems. The parameter $\alpha$, which controls the degree of overlapping between components, is set to 3 for all cases. In these experiments, for each component mean $\mu_i$, we, however, add a random translation $\mathbf{b}_i$ where $b_{ij} \in [-\alpha/2, \alpha/2]$ and random local rotations where $\phi_{ij} \sim N(0, 5)$ in degrees. This additional translation and rotation are sufficient to drop the condition of a unique GT on all components. The combination of GT and LT can be considered as a GT with noise added to the parameters of the transformation of the global model. For each dimension, we repeat 200 realizations for different values of $\gamma$ (0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, and 10), which determines the degree of freedom for the LT transformation parameters, defined in (68). For each case, we test both GT

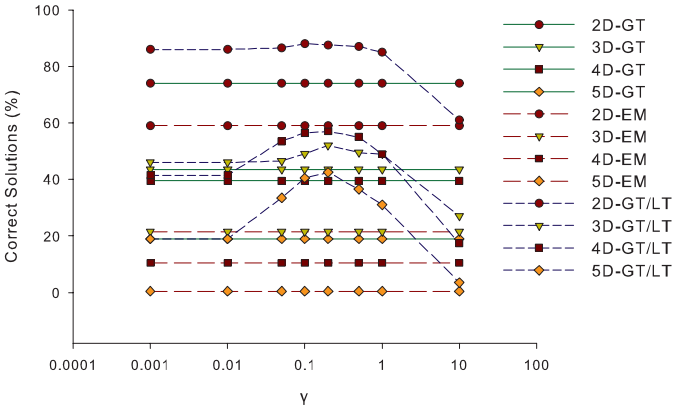



Fig. 5. Percentage of correct solution identification for 2-D, 3-D, 4-D, and 5-D problems with EM, GT, and GT/LT approaches.

and GT/LT approaches, as well as the standard EM algorithm for comparison purposes. In Fig. 5, the percentage of correct solution identification for all dimensions and values of $\gamma$ is presented. GT/LT approach outperforms GT for a large range of $\gamma$, whereas both methods provide clearly better results than the standard EM. Furthermore, the gain of using the GT/LT compared with GT and EM grows with the dimension and the complexity of the problem.

### *B. Experiments Using Real Data*

The performance of the proposed method is tested in two speech recognition data sets consisting of individual speakers. The features from one speaker are considered to be generated from a distribution, which is a transformation of a common phoneme generating distribution. The first data set is the deterding data set [31] consisting of the steady-state portion of 11 vowels in British English, spoken in the context of h*d. The recorded speech samples are low-pass filtered at 4.7 KHz before being digitized at 10 KHz with a 12-b resolution. With the linear predictive analysis, 10 features are calculated. The data set consists of 15 speakers, where eight speakers are used for training and seven for testing. There were several papers presented in the literature, which used the specific data set [33], [34]. Their approaches were, however, based on classifiers with no learning on each speaker. The second data set is the phoneme data set [32] with 256 features and five phonemes. For this data set, the first speaker is used for training, and the rest seven speakers for testing. For better understanding of the problem, both data sets are visualized using the first two PCA components (Fig. 6). It can be observed that the Phoneme data set seems well separated. To decrease the distance between classes for the phoneme problem, a random transformation is applied to each speaker individually. The random transformation includes only a rotation matrix with angles randomly drawn in $[0, \theta]$, where $\theta$ is set to 0°, 10°, or 20°.

For fitting the training model to a new speaker (test), we apply the following procedure.

1) For the deterding data set, a simple normalization is applied to the subjects of the training set, by translating the samples to register them with the first subject's data, which is considered as a reference subject for the training set. For the phoneme data set, dimensionality reduction with PCA is performed and then the random transformation is applied.
2) A GMM model on the training set is learnt with the number of components being equal to the number of classes (phonemes).
3) The initial model is transformed to match each subject of the test set. The compared methods are the GT, GT/LT, and ML-EM approaches for GMM learning.
4) Each datum is classified based on its posterior probability.

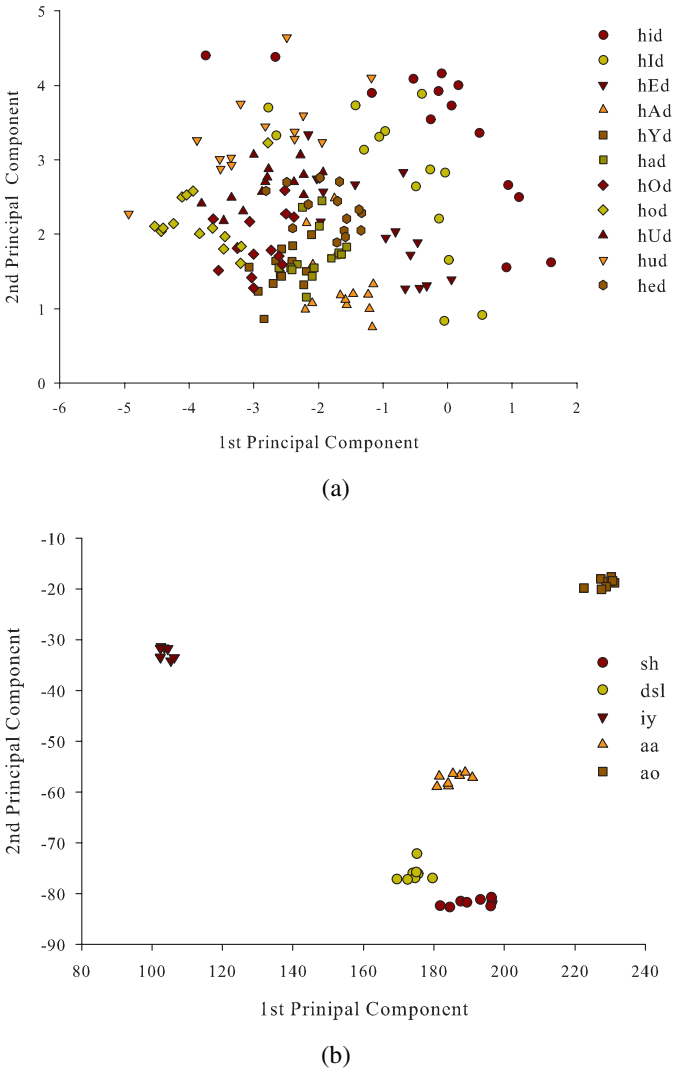



Fig. 6. Visualization of the mean feature vector for each phoneme class of the (a) deterding and (b) phoneme data sets. Each point corresponds to a different speaker after projecting the data onto a 2-D space using PCA.

In Table I, which concerns the deterding data set, we present the classification accuracy for each test subject and the overall accuracy for each of the following examined classification methods: 1) GMM without learning; 2) GMM with ML-EM learning; 3) GMM with GT learning; 4) GMM with GT/LT learning; and 5) $K$-NN ($K = \{3, 5, 7\}$). We observe that the GT/LT approach gives the best vowel recognition results.

TABLE I

CLASSIFICATION ACCURACY ON THE DETERDING DATA SET USING DIFFERENT METHODS WITH AND WITHOUT LEARNING ($\gamma = 0.001$ FOR GT/LT). FOR EACH SUBJECT, THE METHOD WITH THE HIGHEST ACCURACY IS HIGHLIGHTED

| Speaker | No-learning | GMM ML-EM | GMM GT | GMM GT/LT | $K$-NN K=7 |
|---|---|---|---|---|---|
| 8 | 0.52 | **0.82** | 0.76 | 0.68 | 0.45 |
| 9 | 0.64 | 0.80 | 0.71 | **0.94** | 0.59 |
| 10 | 0.67 | 0.65 | 0.85 | **0.92** | 0.74 |
| 11 | 0.61 | **0.94** | 0.92 | 0.91 | 0.68 |
| 12 | 0.56 | 0.67 | **0.86** | 0.71 | 0.56 |
| 13 | 0.21 | 0.50 | 0.52 | **0.68** | 0.36 |
| 14 | 0.56 | 0.88 | 0.71 | **0.98** | 0.50 |
| Overall | 0.54 | 0.75 | 0.76 | **0.83** | 0.56 |

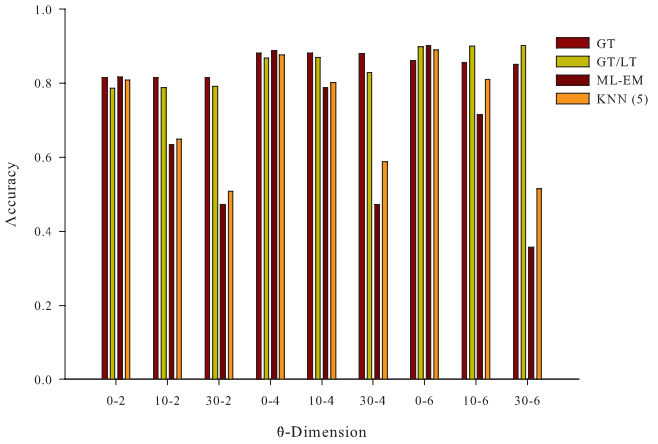



Fig. 7. Classification accuracy for the phoneme data set for different rotations ($\theta$ is the maximum angle) and data dimensions (after the application of PCA).

The GT approach fails for the specific data set, as the GT assumption is not sufficient for the specific problem. K-NN provides the same results, as those reported in [31], which are similar to the simple GMM classification without learning. In [33], [34], more advanced classification methods were applied on the deterding data set, providing better results than KNN. More specifically, the error rates on the test sets were 38.8% and 38%, respectively, yielding classification accuracies of 61.2% and 62%, which are still inferior than those obtained using the proposed methodology.

In Fig. 7, we present the overall accuracy of each method for different dimensions obtained using PCA on the phoneme data set of the 256 features and the random transformation added to each speaker. From the results, it can be observed that whereas for ML-EM and KNN there is a significant reduction in the accuracy of the classification, both GT and GT/LT methods sustain their accuracy.

## V. CONCLUSION

In this paper, we addressed the problem of estimating the transformation parameters of a GMM, with respect to an original model. The method presented here was based on the EM framework. We considered both the cases where a GT was applied on all GMM's components and the case where a LT was also applied on each component. With a proper formulation of the problem as a constrained optimization problem, many global optimization techniques could be applied. In this paper, we, however, focused on the application of the MAP-EM as it was apparently very suitable for solving the specific problem.

Initially, we examined our method for the GT case. A set of experiments was performed to examine the impact of GMM overlapping in the correct solution identification. We verified that the larger the overlapping, the more difficult was to identify the correct solution, regardless of the problem's dimension.

Next, we examined the impact of the rotation applied on the GMM. Our experiments showed, as expected, that the larger the rotation, the more difficult was to identify the correct solution. We may, however, easily overcome this problem using an EM approach with multiple starts and different configurations for the initial rotation angles. The impact of the number of components was also examined. Larger number of components led to larger component overlapping and more local minima, where EM can be trapped.

In addition, a set of experiments was performed to demonstrate that the GT/LT approach was able to identify the correct solution when the simple GT approach failed. The failure of the GT approach was expected when the assumption of a unique transformation on all components did not hold. Our experiments, however, showed that the deviation of the applied transformation from a global one was also important. GT/LT was compared with GT and EM approaches and provided better results for a wide range of the $\gamma$ parameter, which controlled the models deviation from the GT assumption.

Finally, the proposed method was applied on two real-world problems of vowel recognition from different speakers, demonstrating also its practical value. The GT/LT approach was compared with the simple GT approach as well as with other GMM learning methods (ML-EM) and classification methods (K-NN). The GT/LT learning provided the best results with significant difference from the other approaches in the deterding data set where classes were not well separated between speakers. The inferior performance of the GT approach, compared with the GT/LT approach, may imply that the assumption of a GT was not accurate for the specific problem. Given a random transformation, the assumption of an affine transformation may lead the GT, GT/LT approaches to be trapped to a local minimum. This can also explain the few cases where ML-EM outperformed GT and GT/LT methods.

For the phoneme data set, it was demonstrated that the GT and GT/LT approaches clearly outperformed ML-EM and KNN methods when an additional noise transformation was added on each speaker, increasing the learning difficulty.

In the proposed methodology, a one-to-one mapping was considered for the components of the initial and the resulting model. For the identification of the initial model's components, there were methods that could provide an estimation of the optimal number of components [16]. Cases were adaptation was required of an initial model with $N$ components to a new distribution with $M$ components where $M < N$ or $M > N$ was not handled in this paper. In the case of $M < N$, a possible solution was to allow the $\pi$ coefficients tended to zero. Handling unequal component numbers was considered

as a very interesting extension of the proposed methodology and planned as future work.

The computational cost of the GT, GT/LT approaches was higher than the one of classic EM, as it involved an optimization step. Considering diagonal or spherical covariance matrix, the number of parameters was, however, significantly reduced and the estimation of parameters was simplified eliminating the optimization step and speeding up the procedure. We also described the small rotation angles, where using the small angle approximations of (37) can also speed up the method. Those simplifications were also necessary when applying the proposed methodology to very high-dimensional problems. Other possible applications of our method are image registration or objects tracking in video from a moving camera. A significant extension of our method, suitable for the latter application, is the online estimation of the transformation parameters. This extension seems feasible for small rotation angles, where small angle approximations of (37) were accurate, and it is planned as future work.

## APPENDIX
## MAXIMIZATION OF (10)

We introduce the following matrices:

$$I_{ij}^{(n\times n)}(k,l)=\begin{cases}1,\ k=i \text{ and } l=i\\ 1,\ k=j \text{ and } l=j\\ 0,\quad \text{otherwise}\end{cases} \tag{71}$$

$$J_{ij}^{(n\times n)}(k,l)=\begin{cases}-1,\ k=i \text{ and } l=j\\ 1,\ \ k=j \text{ and } l=i\\ 0,\quad \text{otherwise}\end{cases} \tag{72}$$

$$K_{ij}^{(n\times n)}(k,l)=\begin{cases}1,\ k=l \text{ and } k\neq i \text{ and } l\neq j\\ 0,\quad \text{otherwise.}\end{cases} \tag{73}$$

The log likelihood may be split as the sum of two terms as follows:

$$\Im=\sum_i \boldsymbol{\pi}_i(\mathbf{X})[\Im_i^a+\Im_i^b] \tag{74}$$

where $J_i^b$ and $J_i^a$ are given by (27) and (28), respectively. $J_i^a$ can be written in an expanded form as

$$\begin{aligned}\Im_i^a=&\boldsymbol{x}_{ij}^T\boldsymbol{x}_{ij}\cos(\phi_j)^2+\boldsymbol{y}_{ij}^T\boldsymbol{y}_{ij}\sin(\phi_j)^2\\ &+2\boldsymbol{x}_{ij}^T\boldsymbol{y}_{ij}\cos(\phi_j)\sin(\phi_j)\\ &+2\boldsymbol{x}_{ij}^T(\boldsymbol{z}_{ij}+\boldsymbol{w}_i)\cos(\phi_j)\\ &+2\boldsymbol{y}_{ij}^T(\boldsymbol{z}_{ij}+\boldsymbol{w}_i)\sin(\phi_j).\end{aligned} \tag{75}$$

Taking the derivative of $J_i^a$ with respect to $\phi_j$, we obtain the following:

$$\begin{aligned}\frac{\partial\Im_i^a}{\partial\phi_j}=&2[(\boldsymbol{x}_{ij}^T\boldsymbol{x}_{ij}-\boldsymbol{y}_{ij}^T\boldsymbol{y}_{ij})]\cos(\phi_j)\sin(\phi_j)\\ &+2\boldsymbol{x}_{ij}^T\boldsymbol{y}_{ij}\sin(\phi_j)^2-2\boldsymbol{x}_{ij}^T\boldsymbol{y}_{ij}\cos(\phi_j)^2\\ &+2\boldsymbol{x}_{ij}^T(\boldsymbol{z}_{ij}+\boldsymbol{w}_i)\sin(\phi_j)-2\boldsymbol{y}_{ij}^T(\boldsymbol{z}_{ij}+\boldsymbol{w}_i)\cos(\phi_j)\\ =&(\boldsymbol{x}_{ij}^T\boldsymbol{x}_{ij}-\boldsymbol{y}_{ij}^T\boldsymbol{y}_{ij})]\sin(2\phi_j)+2\boldsymbol{x}_{ij}^T\boldsymbol{y}_{ij}\cos(2\phi_j)\\ &+2\boldsymbol{x}_{ij}^T(\boldsymbol{z}_{ij}+\boldsymbol{w}_i)\sin(\phi_j)\\ &-2\boldsymbol{y}_{ij}^T(\boldsymbol{z}_{ij}+\boldsymbol{w}_i)\cos(\phi_j).\end{aligned} \tag{76}$$

The term $\Im_i^b$ can be written in expanded form as follows:

$$\begin{aligned}\Im_i^b=&tr(AS_i^{-1}(A)^T\hat{\Sigma}_{ij}[\mathbf{X}])\\ =&\operatorname{tr}(A_j^cS_i^{-1}(A_j^c)^T\hat{\Sigma}_{ij}[\mathbf{X}])\cos(\phi_j)^2\\ &+\operatorname{tr}(A_j^sS_i^{-1}(A_j^s)^T\hat{\Sigma}_{ij}[\mathbf{X}])\sin(\phi_j)^2\\ &+2\operatorname{tr}(A_j^sS_i^{-1}(A_j^c)^T\hat{\Sigma}_{ij}[\mathbf{X}])\cos(\phi_j)\sin(\phi_j)\\ &+2\operatorname{tr}(A_j^cS_i^{-1}(A_j^k)^T\hat{\Sigma}_{ij}[\mathbf{X}])\cos(\phi_j)\\ &+2\operatorname{tr}(A_j^sS_i^{-1}(A_j^k)^T\hat{\Sigma}_{ij}[\mathbf{X}])\sin(\phi_j).\end{aligned} \tag{77}$$

Then, we consider the derivative of $\Im_i^b$ with respect to $\phi_j$

$$\begin{aligned}\frac{\partial\Im_i^b}{\partial\phi_j}=&(\operatorname{tr}(A_j^cS_i^{-1}(A_j^c)^T\hat{\Sigma}_{ij}[\mathbf{X}])\sin(2\phi_j)\\ &-\operatorname{tr}(A_j^sS_i^{-1}(A_j^s)^T\hat{\Sigma}_{ij}[\mathbf{X}])\sin(2\phi_j)\\ &+2\operatorname{tr}(A_j^sS_i^{-1}(A_j^c)^T\hat{\Sigma}_{ij}[\mathbf{X}])\cos(2\phi_j)\\ &+2\operatorname{tr}(A_j^cS_i^{-1}(A_j^k)^T\hat{\Sigma}_{ij}[\mathbf{X}])\sin(\phi_j)\\ &-2\operatorname{tr}(A_j^sS_i^{-1}(A_j^k)^T\hat{\Sigma}_{ij}[\mathbf{X}])\cos(\phi_j).\end{aligned} \tag{78}$$

The last two terms containing matrix $A_j^k$, which is defined by (36), give zero in the diagonal, thus we obtain the following:

$$\begin{aligned}\frac{\partial\Im^b}{\partial\phi_j}=&\Big[\operatorname{tr}(A_j^cS_i^{-1}(A_j^c)^T\hat{\Sigma}_{ij}[\mathbf{X}])\\ &-\operatorname{tr}(A_j^sS_i^{-1}(A_j^s)^T\hat{\Sigma}_{ij}[\mathbf{X}])\Big]\sin(2\phi_j)\\ &+2\operatorname{tr}(A_j^sS_i^{-1}(A_j^c)^T\hat{\Sigma}_{ij}[\mathbf{X}])\cos(2\phi_j).\end{aligned} \tag{79}$$

Combining (76) and (79) with (74), we finally obtain the following:

$$\begin{aligned}\frac{\partial\Im}{\partial\phi_i}=&\sum_i n_i[[(\boldsymbol{x}_{ij}^T\boldsymbol{x}_{ij}-\boldsymbol{y}_{ij}^T\boldsymbol{y}_{ij})\\ &+\operatorname{tr}(A_j^cS_i^{-1}(A_j^c)^T\hat{\Sigma}_{ij}[\mathbf{X}])\\ &-\operatorname{tr}(A_j^sS_i^{-1}(A_j^s)^T\hat{\Sigma}_{ij}[\mathbf{X}])]\sin(2\phi_j)\\ &+2[\boldsymbol{x}_{ij}^T\boldsymbol{y}_{ij}+\operatorname{tr}(A_j^sS_i^{-1}(A_j^c)^T\hat{\Sigma}_{ij}[\mathbf{X}])]\cos(2\phi_j)\\ &+2[\boldsymbol{x}_{ij}^T(\boldsymbol{z}_{ij}+\boldsymbol{w}_i)]\sin(\phi_j)\\ &-2[\boldsymbol{y}_{ij}^T(\boldsymbol{z}_{ij}+\boldsymbol{w}_i)]\cos(\phi_j).\end{aligned} \tag{80}$$

Setting (80) equal to zero, an equation of the following form is obtained:

$$a\cos(2\phi)+b\sin(2\phi)+c\cos(\phi)+d\sin(\phi)=0 \tag{81}$$

where $a, b, c$, and $d$ are defined in (38), respectively.

## REFERENCES


[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[2] M. Aitkin, "Likelihood and Bayesian analysis of mixtures," *Stat. Model.*, vol. 1, no. 4, pp. 287–304, 2001.

[3] L. A. Goodman, "Exploratory latent structure analysis using both identifiable and unidentifiable models," *Biometrika*, vol. 61, no. 2, pp. 215–231, 1974.

[4] G. McLachlan and D. Peel, *Finite Mixture Models*. New York, NY, USA: Wiley, 2000.

[5] D. Gerogiannis, C. Nikou, and A. Likas, "The mixtures of Student's t-distributions as a robust framework for rigid registration," *Image Vis. Comput.*, vol. 27, no. 9, pp. 1285–1294, 2009.

[6] B. Jian and B. C. Vemuri, "A robust algorithm for point set registration using mixture of Gaussian," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, vol. 2. Oct. 2005, pp. 1246–1251.
[7] M. Aitkin, "A general maximum likelihood analysis of overdispersion in generalized linear models," *Stat. Comput.*, vol. 6, no. 3, pp. 251–262, 1996.
[8] S. S. Brandt, "Maximum likelihood robust regression by mixture models," *J. Math. Imag. Vis.*, vol. 25, no. 1, pp. 25–48, 2006.
[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.
[10] C. Fraley and A. Raftery, "Bayesian regularization for normal mixture estimation and model-based clustering," *J. Classification*, vol. 24, no. 2, pp. 155–181, 2007.
[11] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, nos. 1–3, pp. 19–41, 2000.
[12] Q. Huang, J. Yang, and Y. Zhou, "Variational Bayesian method for speech enhancement," *Neurocomputing*, vol. 70, nos. 16–18, pp. 3063–3067, 2007.
[13] W. Shinji, S. Atsushi, and N. Atsushi, "Automatic determination of acoustic model topology using variational Bayesian estimation and clustering for large vocabulary continuous speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 3, pp. 855–872, May 2006.
[14] M. Jordan and J. Liu, "The BYY annealing learning algorithm for Gaussian mixture with automated model selection," *Pattern Recognit.*, vol. 40, no. 7, pp. 2029–2037, 2007.
[15] N. Vlassis and A. Likas, "A greedy EM algorithm for Gaussian mixture learning," *Neural Process. Lett.*, vol. 15, no. 1, pp. 77–87, 2002.
[16] C. Constantinopoulos and A. Likas, "Unsupervised learning of Gaussian mixtures based on variational component splitting," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 745–755, May 2007.
[17] K. Blekas, A. Likas, N. P. Galatsanos, and I. E. Lagaris, "A spatially constrained mixture model for image segmentation," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 494–498, Mar. 2005.
[18] A. Penalver, F. Escolano, and J. Saez, "Color image segmentation through unsupervised Gaussian mixture models," *Adv. Artif. Intell. IBERAMIA-SBIA*, LNCS 4140, pp. 149–158, Jan. 2006.
[19] V. Digalakis, D. Rtischev, L. Neumeyer, and E. Sa, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 357–366, Sep. 1995.
[20] G. Xiong, C. Feng, and L. Ji, "Dynamical Gaussian mixture model for tracking elliptical living objects," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 838–842, 2006.
[21] E. Sa, V. Digalakis, and L. Neumeyer, "Speaker adaptation using combined transformation and Bayesian methods," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 4, pp. 294–300, May 1995.
[22] S. Moss and E. R. Hancock, "Cartographic matching with millimetre radar images," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Dec. 1996, pp. 70–76.
[23] C. Nikou, A. C. Likas, and N. P. Galatsanos, "A Bayesian framework for image segmentation with spatially varying mixtures," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2278–2289, Sep. 2010.
[24] G. Sfikas, C. Nikou, N. Galatsanos, and C. Heinrich, "Spatially varying mixtures incorporating line processes for image segmentation," *J. Math. Imag. Vis.*, vol. 36, no. 2, pp. 91–110, 2010.
[25] G. Sfikas, C. Nikou, N. Galatsanos, and C. Heinrich,"Majorization-minimization mixture model determination in image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 2169–2176.
[26] R. Yang and J. O. Bergers, "Estimation of a covariance matrix using the Reference Prior," *Annu. Stat.* vol. 22, no. 3, pp. 1195–1211, 1994.
[27] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Comput.*, vol. 8, pp. 129–151, Jan. 1995.
[28] M. Jordan and S. Fu, "On the correct convergence of the EM algorithm for Gaussian mixtures," *Pattern Recognit.*, vol. 38, no. 12, pp. 2602–2611, 2005.
[29] M. Jordan, L. Xu, and M. I. Jordan, "Asymptotic convergence rate of the EM algorithm for Gaussian mixtures," *Neural Comput.*, vol. 12, no. 12, pp. 2881–2907, 2000.
[30] J. L. Gauvain, C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov Chains," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.
[31] D. H. Deterding, "Speaker normalization for automatic speech recognition," Ph.D. dissertation, Dept. Int. Graduate School, Univ. Trento, Italy, 1990.
[32] T. Hastie, A. Buja, and R. Tibshirani, "Penalized discriminant analysis," *Anna. Stat.*, vol. 23, no. 1, pp. 73–102, 1995.
[33] M. I. Layton and M. J. F. Gales, "Maximum margin training of generative kernels," Tech. Rep., Dept. Eng., Cambridge Univ., Cambridge, U.K., 2004.
[34] D. J. Miller and H. S. Uyar, "Combined learning and use for a mixture model equivalent to the RBF classifier," *Neural Comput.*, vol. 10, no. 2, pp. 281–293, 1998.

**George Rigas** was born in Pella, Greece, in 1981. He received the bachelor's and M.Sc. degrees in computer science from the Department of Computer Science, University of Ioannina, Ioannina, Greece, in 2003 and 2005, respectively, and the Ph.D. degree in computer science from the same department in 2009.

He has been a Post-Doctoral Researcher with the Unit of Medical Technology and Intelligent Information Systems, University of Ioannina, since 2010. His current research interests include signal processing, machine learning, and pattern recognition.

**Christophoros Nikou** (S'97–M'05–SM'11) received the Diploma degree in electrical engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1994, and the D.E.A. and Ph.D. degrees in image processing and computer vision from Louis Pasteur University, Strasbourg, France, in 1995 and 1999, respectively.

He was a Senior Researcher with the Department of Informatics, Aristotle University of Thessaloniki, in 2001. From 2002 to 2004, he was a Research Engineer and Project Manager with Compucon S.A., Thessaloniki. He was a Lecturer with the Department of Computer Science, University of Ioannina, Ioannina, Greece, from 2004 to 2009, where he has been an Assistant Professor since 2009. His current research interests include image processing and computer vision and their application to medical imaging.

Dr. Nikou is an Associate Editor for the *EURASIP Journal on Advances in Signal Processing*. He is a member of EURASIP.

**Yorgos Goletsis** (M'03) received the Diploma degree in electrical engineering and the Ph.D. degree in operations research from the National Technical University of Athens, Athens, Greece.

He is a Lecturer with the Department of Economics, University of Ioannina, Ioannina, Greece. His current research interests include operations research, decision support systems, multicriteria analysis, quantitative analysis, data mining, artificial intelligence, and project evaluation.

**Dimitrios I. Fotiadis** (M'01–SM'07) was born in Ioannina, Greece, in 1961. He received the Diploma degree in chemical engineering from the National Technical University of Athens, Athens, Greece, in 1985, and the Ph.D. degree in chemical engineering from the University of Minnesota, Minneapolis, MN, USA, in 1990.

He was with the Department of Computer Science, University of Ioannina, Ioannina, Greece, from 1995 to 2008, where he is currently a Professor with the Department of Materials Science and Technology, the Director of the Unit of Medical Technology and Intelligent Information Systems, and the President of the Science and Technology Park of Epirus. He is with the Biomedical Research Institute, Foundation for Research and Technology-Hellas, Ioannina. His current research interests include biomedical technology, biomechanics, scientific computing, and intelligent information systems.