# Minimum Distance between Pattern Transformation Manifolds: Algorithm and Applications

Effrosyni Kokiopoulou, *Student Member*, *IEEE*, and Pascal Frossard, *Senior Member*, *IEEE*

**Abstract**—Transformation invariance is an important property in pattern recognition, where different observations of the same object typically receive the same label. This paper focuses on a transformation-invariant distance measure that represents the minimum distance between the transformation manifolds spanned by patterns of interest. Since these manifolds are typically nonlinear, the computation of the manifold distance (MD) becomes a nonconvex optimization problem. We propose representing a pattern of interest as a linear combination of a few geometric functions extracted from a structured and redundant basis. Transforming the pattern results in the transformation of its constituent parts. We show that, when the transformation is restricted to a synthesis of translations, rotations, and isotropic scalings, such a pattern representation results in a closed-form expression of the manifold equation with respect to the transformation parameters. The MD computation can then be formulated as a minimization problem whose objective function is expressed as the difference of convex functions (DC). This interesting property permits optimally solving the optimization problem with DC programming solvers that are globally convergent. We present experimental evidence which shows that our method is able to find the globally optimal solution, outperforming existing methods that yield suboptimal solutions.

**Index Terms**—Transformation invariance, pattern manifolds, sparse approximations.

## 1 Introduction

With the ever-increasing amounts of digital information content created by different sources, efficient analysis and processing algorithms are becoming crucial to identifying meaningful information in high-dimensional and heterogenous data sets. In particular, pattern recognition and data mining algorithms have to be able to identify relevant objects in different representations or descriptions and transformation invariance certainly becomes key in the design of efficient data analysis methods. For example, in applications involving classification of visual objects, the classification function should be invariant to the geometric transformations of the objects, such as scalings, rotations, and translations. At the same time, the efforts toward defining features that support transformation invariance are aligned with bridging the so-called *semantic gap*, which refers to the shortcomings of low-level features to address high-level content analysis tasks. Transformation invariance is therefore a very important property of any learning algorithm and can have a big impact on its success in the context of various mining tasks.

The comparison of two patterns is generally meaningful if they are aligned first so that their distance reflects their structural and geometric differences. Alignment consists of estimating the relative transformation between patterns. The transformed version of a pattern can be described as a point of a (possibly nonlinear) manifold in a high-dimensional space, which is usually called the *transformation manifold*. If the distance between two patterns is truly invariant to transformations, it becomes equivalent to the distance between their corresponding transformation manifolds. This is called the two-sided manifold distance (MD) or the one-sided MD when one of the patterns is fixed. Observe that the MD between a pattern and its transformed version is zero; hence, MD is close to the semantic distance between any two patterns. The minimum manifold distance thus corresponds to the distance between patterns after proper alignment.

In this paper, we introduce a method for pattern alignment and MD computation which is able to find the globally optimal solution and provide global invariance to transformations. We represent the pattern of interest as a linear combination of geometric primitives, called atoms, which are chosen from a structured and possibly redundant basis or dictionary. A dictionary is called *structured* when it is built by atoms that are constructed by applying geometric transformations on a mother function (e.g., Gaussian functions [1]). When the pattern is transformed, the transformation is essentially applied on each constituent atom individually, resulting in a synthesis of transformations. Using the group theory of transformations, we build on [2] and show that the proposed representation allows for a closed-form expression of the manifold equation with respect to the transformation parameters. Next, we formulate the pattern alignment problem as a DC program by showing that the objective function is DC, i.e., that it can be written as a *difference of convex functions*. DC programs are nonconvex problems that can be globally solved by

• *The authors are with the Ecole Polytechnique Fédérale de Lausane (EPFL), Signal Processing Laboratory (LTS4), Lausanne-1015, Switzerland. E-mail: {effrosyni.kokiopoulou, pascal.frossard}@epfl.ch.*

*Manuscript received 24 Nov. 2007; revised 12 Apr. 2008; accepted 2 June 2008; published online 5 June 2008.*
*Recommended for acceptance by L. Bottou.*
*For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-11-0791.*
*Digital Object Identifier no. 10.1109/TPAMI.2008.156.*

exploiting their special structure. In order to solve the DC program, we employ an outer approximation-cutting plane method which converges to the globally optimal solution. The algorithm allows to estimate jointly the translation, rotation, and scaling parameters, and ensure convergence to the global minimum manifold distance. We finally provide experimental results which show the effectiveness of the proposed algorithm in robust pattern recognition and image alignment applications, where it outperforms existing solutions typically based on suboptimal manifold, tangent, or euclidean distance computation.

The rest of this paper is organized as follows: In Section 2, we review the related work about introducing transformation invariance in pattern recognition. Next, in Section 3, we discuss the representation of transformation manifolds using sparse geometric expansions and, in Section 4, we formulate the optimization problem. We show in Section 5 that the objective function is DC. We present its solution with an outer approximation-cutting plane algorithm in Section 6. Finally, the experimental results are presented in Section 7.

## 2 Related Work

In this section, we review the main research efforts on introducing transformation invariance in pattern recognition. One possible approach to introduce invariance into pattern recognition algorithms is to use transformation-invariant features. However, there are two main disadvantages with this approach: 1) Crucial information may be discarded and 2) it is hard to evaluate the impact of feature extraction on the classification error [3].

For this reason, a lot of research efforts have concentrated on seeking for invariance by the computation (or the approximation) of appropriate distance measures in the pattern space. The computation of the MD requires the alignment of the involved patterns. However, alignment is a hard optimization problem, mostly due to the nonconvexity of its objective function. One solution consists of the local linearization of the manifold, in order to compute the distance between tangent spaces instead of the true manifold distance. Simard et al. in [4] introduced the notion of the tangent distance (TD) of a reference pattern $p$ from the manifold $\mathcal{T}_s$ spanned by the transformed versions of the pattern $s$. The TD has been successfully applied in digit image recognition [5]. The main idea is to use a locally linear approximation of $\mathcal{T}_s$ around $s$. The linear approximation is constructed using the tangent vectors $\frac{\partial \mathcal{T}}{\partial \eta}$, where $\eta$ denotes the transformation parameters. The tangest distance is defined as the distance of $p$ from the tangent space (see Fig. 1 for a schematic illustration). The effectiveness of this approach is highly sensitive to the nonlinearity of the manifold. Since the obtained linear approximation using the tangent space is accurate only locally, this method provides local invariance to transformations.

The smallest distance of $p$ from the manifold $\mathcal{T}_s$ is called the MD (shown in Fig. 1) and is truly transformation invariant. Unfortunately, these manifolds have no analytic expression in general and they are typically nonlinear. Hence, the computation of MD involves the solution of a hard (typically nonconvex) optimization problem with an unknown number of local minima. One approach in computing the MD is to use a traditional method such as Newton's method or steepest descent. However, this will yield a suboptimal solution due to the presence of local minima. Along the same lines, in [6], the authors use probabilistic pattern models and they introduce priors both on the transformation parameters as well as the pattern that generates the manifold. They introduce the so-called joint manifold distance, defined as the distance between the two pattern manifolds that is optimized over the transformation parameters and the pattern themselves. The resulting optimization problem is solved using Levenberg-Marquardt and they apply their method to cluster faces in video.

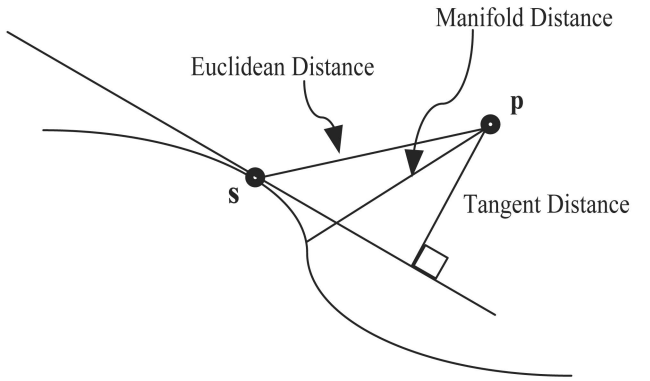



Fig. 1. Manifold distance is the distance from a reference point $p$ to the transformation manifold of $s$. The tangent distance is the distance from $p$ to the tangent space of the manifold with respect to $s$.

To alleviate the problem of local mimima, Vasconcelos and Lippman [3] proposed the multiresolution manifold distance, which attempts to compute the MD by using a multiresolution decomposition of the involved images and then employ Newton's method in each resolution level. Starting from the coarsest level, Newton's method is used in order to converge to a local minimum solution, which is used as the initial guess to Newton in the next level. This is iterated until the finest image resolution is reached. The intuition is that, at the coarsest level, the MD objective function will be less "bumpy" with less local minima. Thus, the hope is that Newton's method will be less susceptible to them. Although this methodology provides robustness to a wider range of transformations relative to TD and MD, there is no guarantee that it will converge to the global minimizer.

Another approach is to introduce invariance by learning from transformed examples, the so-called *virtual* examples. Typically, one applies various transformations on the training examples resulting in an expanded training set. This has the advantage that can be readily combined with any classification algorithm. However, the training set usually becomes huge and this approach is very costly in terms of memory requirements. Motivated by this shortcoming, DeCoste and Burl [7] introduced the concept of jittered queries. The main idea is to apply various transformations on the test samples resulting in an expanded set of transformed test samples. Shifting the responsibility of the invariance from the training to the

testing phase has certain advantages in terms of computational cost and memory usage.

A lot of research efforts have focused on introducing transformation invariance into Support Vector Machines (SVMs). This can be attempted by the following approaches (see [8] for more details and references): 1) design of invariant kernels [9] and 2) introduction of virtual support vectors [10], [11]. In particular, Schölkopf et al. introduced invariance into SVMs by generating virtual examples from the support vectors, rather than the whole training set. This yielded a training phase that is more computationally efficient since the training set size is controlled now. 3) Kernel jittering [7], where the main idea is the following: Instead of performing the transformations on the training examples before training, they are performed on-demand during kernel evaluation.

In the context of maximum margin classifiers, Graepel and Herbrich [12] proposed the semidefinite programming machines (SDPM). They represent the transformation manifold of each training sample by a polynomial trajectory (using Taylor expansion) and then formulate the margin maximization as a semidefinite program. Experimentally, SDPM performed slightly better than the virtual SVM. However, extending their methodology to more generic transformations consisting of more than one parameter is complex.

Another approach toward invariance in pattern recognition is to encode invariance into the classification algorithm itself. For example, Simard et al. in [4] modified the error function of neural networks in order to make it invariant to transformations. In particular, they included an additional term in the error function, which captures the magnitude of the gradient of the classifying function with respect to the transformation parameters. Thus, minimizing the total error function results in a classification function that is smooth with respect to the desired transformations.

Finally, it is important to stress [3] that the problem of computing the manifold distance is closely related to the problem of image alignment or registration. For instance, the article in [13] proposes the use of multiresolution decomposition combined with Levenberg-Marquardt for image registration, which is close to the idea proposed in [3]. The authors in [14], [15] propose a maximum likelihood (ML) approach for translation, rotation, and scale estimation. The idea is to loop over the pixels and for each pixel to estimate scale and rotation by maximizing an approximation of the log-likelihood using quasi-Newton. The latter approximation is obtained by expanding the images in the Laguerre-Gauss transform domain. This approach does an exhaustive search over the possible translation parameters and solves the scale-rotation problem for each pixel. On the contrary, our methodology solves the full problem all at once. Due to lack of space, we will not elaborate more on the related work in image registration. The reader is referred to [16] for a survey on this topic.

# 3 Transformation Manifolds Built on Sparse Geometric Expansions

## 3.1 Visual Pattern Representation

We introduce here a pattern representation based on sparse geometric expansions, which yields a parametric model of the pattern. Such a geometric pattern representation leads to a closed-form expression of the transformation manifold equation with respect to the transformation parameters, which is later used for efficient computation of the manifold distance.

We propose representing the pattern of interest as a linear combination of geometric functions (usually called *atoms*), taken from a structured and possibly redundant dictionary $\mathcal{D} = \{\phi_\gamma, \gamma \in \Gamma\}$ spanning the input space. This representation generally allows us to capture the most prominent features of the pattern of interest that are generally also meaningful components. The atoms in a *structured* dictionary are constructed by applying geometric transformations to a generating mother function denoted by $\phi$. A geometric transformation $\gamma \in \Gamma$ can be represented by a unitary operator $U(\gamma)$ so that a structured dictionary takes the following form:

$$\mathcal{D} = \{\phi_\gamma = U(\gamma)\phi, \gamma \in \Gamma\}. \tag{1}$$

With digital images, the simplest transformation $\gamma_i$, applied to the $i$th atom, may be one of the following three types:

- *Translation* by $b_i = [b_{ix}\ b_{iy}]^\top$. $U(b_i)$ moves the generating function across the image, i.e., $U(b_i)\phi(x,y) = \phi(x - b_{ix}, y - b_{iy})$.
- *Rotation* by $\omega_i$. $U(\omega_i)$ rotates the generating function by angle $\omega_i$, i.e., $U(\omega_i)\phi(x,y) = \phi(\cos(\omega_i)x + \sin(\omega_i)y, \cos(\omega_i)y - \sin(\omega_i)x)$.
- *Anisotropic scaling* by $a_i = [a_{ix}\ a_{iy}]^\top$. $U(a_i)$ scales the generating function anisotropically in the two directions, i.e., $U(a_i)\phi(x,y) = \phi(\frac{x}{a_{ix}}, \frac{y}{a_{iy}})$.

Composing all of the above transformations yields a transformation with parameters $\gamma_i = \{b_i, a_i, \omega_i\} \in \Gamma$, which denotes a synthesis of translations, anisotropic scalings, and rotations. It can be observed that applying a transformation on the mother function is equivalent to transforming the coordinate system from $\{x, y\}$ to $\{\tilde{x}, \tilde{y}\}$ before applying $\phi(\cdot)$. When the $i$th atom in the structured dictionary (1) is built as $\phi_{\gamma_i} = U(\gamma_i)\phi(x,y)$, where $\gamma_i = \{b_i, a_i, \omega_i\} \in \Gamma$, it forms the same 2D function as $\phi(\tilde{x}, \tilde{y})$, where

$$\begin{aligned}\begin{bmatrix}\tilde{x}\\ \tilde{y}\end{bmatrix} &= \underbrace{\begin{bmatrix}\frac{1}{a_{ix}} & 0\\ 0 & \frac{1}{a_{iy}}\end{bmatrix}}_{A} \underbrace{\begin{bmatrix}\cos\omega_i & \sin\omega_i\\ -\sin\omega_i & \cos\omega_i\end{bmatrix}}_{R(\omega_i)} \underbrace{\begin{bmatrix}x - b_{ix}\\ y - b_{iy}\end{bmatrix}}_{t}\\ &= AR(\omega_i)t.\end{aligned} \tag{2}$$

The approximation of the pattern of interest $s$ with atoms from the dictionary $\mathcal{D}$ can be obtained in different ways. Even if finding the sparsest approximation of $s$ is generally a hard problem, effective suboptimal solutions are usually sufficient to capture the salient semantic and geometric structure of a pattern with only a few atoms. In this work, we have chosen to use Orthogonal Matching Pursuit (OMP)

Fig. 2. Progressive OMP approximation of the digit "5" (leftmost) with 10, 20, 30, 40, and 50 Gaussian atoms (from left to right).

[17, Section 9.5.3], which is a simple and effective algorithm for computing sparse approximations in practice.

Initially, OMP sets the residual $r_0 = s$ and then proceeds iteratively by selecting in the $j$th step the atom $\phi_{\gamma_j}$ that best matches the residual $r_{j-1}$, i.e., $\gamma_j = \arg_{\gamma\in\Gamma} \max |\langle r_{j-1}, \phi_\gamma\rangle|$. At each step, it updates the residual by orthogonal projection on the span of the selected atoms (i.e., $r_j = (I - P)r_{j-1}$, where $P$ is the orthogonal projector on the $\text{span}\{\phi_{\gamma_1}, \ldots, \phi_{\gamma_j}\}$). After $K$ steps of OMP, the pattern $s$ is approximated by a sparse linear combination of a few atoms, i.e.,

$$s = \sum_{k=1}^{K} \xi_k \phi_{\gamma_k} + r_K, \tag{3}$$

where $r_K$ is the residual of the approximation. We should mention that, besides OMP, there are several approaches for computing sparse approximations, such as the Sparse Greedy Matrix Approximation [18], [19] and Tree-based Pursuit [20], to name just a few. The two latter methods represent fast alternatives to the traditional Matching Pursuit algorithm by expediting the search in the dictionary.

We then propose building the dictionary using 2D atoms that can efficiently capture the geometrical information in natural images. The generating function $\phi$ of $\mathcal{D}$ could be, for instance, a Gaussian function, i.e.,

$$\phi(x, y) = \frac{1}{\rho} \exp\big(-(x^2 + y^2)\big), \tag{4}$$

or an Anisotropic Refinement (AR) function, i.e.,

$$\phi(x, y) = \frac{1}{\rho} \exp\big(-(x^2 + y^2)\big)(4x^2 - 2), \tag{5}$$

where $\rho$ is a normalization factor which ensures that the atom is of unity norm. The latter is an edge-like function that is Gaussian in one direction and second derivative of the Gaussian in the orthogonal direction. Fig. 2 shows the progressive approximation of the digit "5" from a Gaussian dictionary using OMP. Observe that only a few atoms are sufficient to capture the main geometric characteristics of the pattern and the representation (3) does not need to be very accurate before it is useful for alignment purposes.

### 3.2 Transformation Manifolds

When a pattern $s$ undergoes a geometric transformation $\eta$, it spans a high-dimensional manifold $\mathcal{T}$ in the ambient space. Assume that the transformation is $\eta = (\beta, \alpha, \theta)$; that is, it consists of a synthesis of translations $\beta = [\beta_x, \beta_y]$, isotropic scaling $\alpha$, and rotation $\theta$. The manifold $\mathcal{T}$ can be expressed mathematically as follows:

$$\mathcal{T} = \{s(\eta) = U(\eta)s, \quad \eta = (\beta, \alpha, \theta)\}. \tag{6}$$

Although the manifold is high dimensional, its intrinsic dimension is rather small and equal to the number of transformation parameters, which is 4. Fig. 3 shows a few samples from the transformation manifold of the digit "5," when the transformation is a simple rotation.

The transformations $\eta$ form a group, namely, the similitude group SIM(2) on the 2D plane. Let us denote by

$$R(\omega) = \begin{bmatrix} \cos\omega & \sin\omega \\ -\sin\omega & \cos\omega \end{bmatrix}, \quad 0 \le \omega < 2\pi,$$

the rotation matrix by angle $\omega$ in the 2D plane. If $(b, a, \omega)$ and $(b', a', \omega')$ are two elements of the SIM(2) group, then the group law is

$$(b, a, \omega) \circ (b', a', \omega') = (b + aR(\omega)b', aa', \omega' - \omega). \tag{7}$$

Using (3) and dropping the residual term $r_K$, it turns out that applying the transformation $\eta$ on the pattern $s$, results in

$$s(\eta) = U(\eta)s = \sum_{k=1}^{K} \xi_k U(\eta)\phi_{\gamma_k} = \sum_{k=1}^{K} \xi_k \phi_{\eta\circ\gamma_k}, \tag{8}$$

where $\eta \circ \gamma_k$ is a product of transformations. In words, the transformation is applied on each constituent atom individually. The group law (7) indeed applies [1] and can be further employed to determine the updated parameters of the transformed atoms. Equation (8) is of great importance since it expresses the manifold equation (6) in closed form with respect to the transformation parameters $\eta$. This is very valuable for the computation of the MD and, in particular, for the applicability of the DC programming methodology that is proposed in the next section.

## 4 Problem Formulation

In this paper, we are interested in computing the distance between visual patterns under transformation invariance. Suppose that we want to compute the MD from a reference pattern $p$ to the transformation manifold $\mathcal{T}$ described by (6). This is equivalent to estimating the optimal transformation parameters $\eta^*$. We formulate the parameter estimation problem as follows:

$$\eta^* = \arg \max_{\eta=(\beta,\alpha,\theta)} f(\eta), \quad \text{where} \quad f(\eta) = |\langle s(\eta), p\rangle|. \tag{9}$$

Recall that $s(\eta) \in \mathcal{T}$ denotes the transformed pattern $s$ when it is subject to transformation $\eta = (\beta, \alpha, \theta)$. We assume that the pattern of interest $s$ has been well approximated with a sparse expansion over $\mathcal{D}$, i.e.,

Fig. 3. Samples from the transformation manifold of the digit "5." From left to right, the samples correspond to rotation angles from 0 to $2\pi$ with step $\pi/4$.

$$s = \sum_{k=1}^{K} \xi_k \phi_{\gamma_k}, \tag{10}$$

which results from (3) by dropping the residual term. When the atom parameters $\gamma_k$ are fixed, the transformed pattern $s(\eta)$ in (8) provides a parametric pattern model with respect to $\eta$. Note that, in the above optimization problem, only the pattern $s$ is expanded in the redundant basis, i.e., the reference pattern $p$ is used as is.

The purpose of the optimization problem is to compute the exact transformation parameters $\eta^*$. This is typically a nonconvex nonlinear optimization problem [21] which is hard to solve using the traditional methods, such as steepest descent or Newton-type methods due to their local convergence property and the presence of an unknown number of local minima. However, it can be shown that the above objective function is a DC function or, equivalently, that it can be expressed as the difference of two convex functions.

**Proposition 1.** *The objective function*

$$f(\eta) = |\langle s(\eta), p\rangle| = \left| \sum_{k=1}^{K} \xi_k \langle \phi_{\eta_k}, p\rangle \right|, \tag{11}$$

*where $\eta_k = \eta \circ \gamma_k$ is DC.*

We show in the next section that this proposition is true. We first recall basic properties of DC functions. Then, we show that the geometric transformation of an atom by translation, scaling, and rotation is equivalent to a change in the coordinate system. The transformed coordinate system $(\tilde{x}, \tilde{y})$ of the $k$th atom $\phi_{\eta_k}$ depends on the transformation parameters $\eta$. With this transformation in the coordinate system, we can demonstrate that $\tilde{x}(\eta_k)^2 + \tilde{y}(\eta_k)^2$ is a DC function of $\eta$, which in turn will allow us to express the pixels of each atom $\phi_{\eta_k}$ in a DC form. Then, we prove that the inner product $\langle \phi_{\eta_k}, p\rangle$ between the pattern $p$ and each of the atoms is also DC. Finally, we obtain the DC formulation of the objective function $|\langle s(\eta), p\rangle|$.

Due to Proposition 1, the optimization problem can therefore be formulated as a DC program and globally solved by a fast cutting plane method [22, Theorem 5.3]. To the best of our knowledge, this is the first globally optimal algorithm that is proposed in the context of transformation-invariant distance computation.

## 5 DC Decomposition

### 5.1 Properties of DC Functions

We show in this section that the objective function $f$ in (9) is DC, in several steps. We start with some definitions and background material about DC functions [22], [23], [24] and their properties.

First, let $X$ be a convex subset of $\mathbb{R}^n$. A function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is called DC on $X$, if there exist two convex functions $g, h : X \to \mathbb{R}$ such that $f$ is expressed as

$$f(x) = g(x) - h(x). \tag{12}$$

A representation of the above form is called the DC decomposition of $f$. The DC decomposition is clearly not unique, since if $c(x)$ is a convex function, then $f(x) = (g(x) + c(x)) - (h(x) + c(x))$ is another DC decomposition of $f$. In what follows, the notations $\succ 0$ and $\succeq 0$ denote positive definiteness and positive semidefiniteness accordingly. We present now a few properties of DC functions.

**Proposition 2 [23, Section 4.2].** *Properties of DC functions. Let $f = g - h$ and $f_i = g_i - h_i$, $i = 1, \ldots, m$ be DC functions. Then, the following functions are also DC:*

1. $$\sum_{i=1}^{m} \lambda_i f_i = \left[ \sum_{\{i:\lambda_i \geq 0\}} \lambda_i g_i - \sum_{\{i:\lambda_i < 0\}} \lambda_i h_i \right] - \left[ \sum_{\{i:\lambda_i \geq 0\}} \lambda_i h_i - \sum_{\{i:\lambda_i < 0\}} \lambda_i g_i \right].$$
2. $|f| = 2\max\{g, h\} - (g + h)$.
3. *If $f_1$ and $f_2$ are DC functions, then the product $f_1 \cdot f_2$ is DC. Moreover, if $f_1$ and $f_2$ have nonnegative convex parts, the following DC decomposition holds:*

$$f_1 \cdot f_2 = \frac{1}{2}\left[(g_1 + g_2)^2 + (h_1 + h_2)^2\right] - \frac{1}{2}\left[(g_1 + h_2)^2 + (g_2 + h_1)^2\right]. \tag{13}$$

In addition, it can be shown that the synthesis of a convex function and a DC function is also DC, which is particularly relevant for our particular objective function, as shown below.

**Proposition 3.** *Let $f(x) : \mathbb{R}^n \to \mathbb{R}$ be DC and $q : \mathbb{R} \to \mathbb{R}$ be convex. Then,*

1. *the composition $q(f(x))$ is DC [23, Section 4.2] and*
2. *$q(f(x))$ has the following DC decomposition:*

$$q(f(x)) = p(x) - K[g(x) + h(x)], \tag{14}$$

*where $p(x) = q(f(x)) + K[g(x) + h(x)]$ is a convex function and $K$ is a constant satisfying $K \geq |q'(f(x))|$ [25].*

Note that, in part 2 of the above proposition, we use slightly different conditions than those in [25]. For this reason, we provide a proof of part 2 in Appendix A.

### 5.2 DC Decomposition of Transformed Atoms

We show here that the transformed atom $\phi_{\eta_k}$ can be expressed in a DC form. For notational ease, we drop the subscript $k$ since it is clear from the context that we refer to the $k$th atom. Note first that the transformation of an atom by scaling, rotation, and translation is equivalent to a change in the coordinate system.

**Lemma 1.** *The transformed coordinates of an atom in (11) have the following form:*

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} \mu_1 \frac{\cos\theta}{\alpha} + \mu_2 \frac{\sin\theta}{\alpha} + \mu_3 \frac{\tau_x}{\alpha} + \mu_4 \frac{\tau_y}{\alpha} + \mu_5 \\ \nu_1 \frac{\cos\theta}{\alpha} + \nu_2 \frac{\sin\theta}{\alpha} + \nu_3 \frac{\tau_x}{\alpha} + \nu_4 \frac{\tau_y}{\alpha} + \nu_5 \end{pmatrix}, \tag{15}$$

*where the $\mu_i$s and $\nu_i$s are constants depending on the atom parameters which are fixed and $\tau_x$ and $\tau_y$ are related to $\beta_x$ and $\beta_y$ by the following relation:*

$$\begin{pmatrix}\tau_x\\ \tau_y\end{pmatrix} = -R(-\theta)\begin{pmatrix}\beta_x\\ \beta_y\end{pmatrix}. \tag{16}$$

**Proof.** The proof is given in Appendix A. □

With the change of variables in (16), the optimization variables become $\tau_x$, $\tau_y$, $\alpha$, and $\theta$. This representation of the transformation parameters is equivalent to the previous one since, from the $\tau$s, we can recover the $\beta$s and vice versa (using (16)). Hence, for notational ease, we will continue using $\eta$ for denoting the transformation parameters with respect to $\tau_x$, $\tau_y$, $\alpha$, and $\theta$.

The above lemma implies that

$$\begin{aligned}\tilde{x}(\eta)^2 = {} & \mu_1^2\frac{\cos^2\theta}{\alpha^2} + \mu_2^2\frac{\sin^2\theta}{\alpha^2} + \mu_3^2\frac{\tau_x^2}{\alpha^2} + \mu_4^2\frac{\tau_y^2}{\alpha^2} + \mu_5^2 \\ & + 2\mu_1\mu_2\frac{\cos\theta\sin\theta}{\alpha^2} + 2\mu_1\mu_3\frac{\cos\theta\tau_x}{\alpha^2} + 2\mu_1\mu_4\frac{\cos\theta\tau_y}{\alpha^2} \\ & + 2\mu_1\mu_5\frac{\cos\theta}{\alpha} + 2\mu_2\mu_3\frac{\sin\theta t_x}{\alpha^2} + 2\mu_2\mu_4\frac{\sin\theta\tau_y}{\alpha^2} \\ & + 2\mu_2\mu_5\frac{\sin\theta}{\alpha} + 2\mu_3\mu_4\frac{\tau_x\tau_y}{\alpha^2} + 2\mu_3\mu_5\frac{\tau_x}{\alpha} \\ & + 2\mu_4\mu_5\frac{\tau_y}{\alpha}\end{aligned} \tag{17}$$

and similarly for $\tilde{y}(\eta)$ by replacing the $\mu$s by $\nu$s. In order to show that $\phi_\eta$ is DC, we need to show that every single term in the above equation is DC as well. In other words, we have the following proposition:

**Proposition 4.** *The expression corresponding to the transformed coordinate system $\tilde{x}(\eta)^2$ (and, similarly, $\tilde{y}(\eta)^2$) in (17) is DC.*

In what follows, we provide a few lemmas that will help us show that this is true. In particular, we show that several of the constituent functions are DC: $f(\theta) = \cos\theta$, $f(\theta) = \sin\theta$, $f(\theta,\alpha) = \frac{\theta}{\alpha}$, $f(\theta,\alpha) = \frac{\cos\theta}{\alpha}$, and $f(\theta,\alpha) = \frac{\sin\theta}{\alpha}$.

**Lemma 2.** *The following functions are DC:*

1. $f_1(\theta) = \cos\theta,\ \theta \in [0, 2\pi)$.
2. $f_2(\theta) = \sin\theta,\ \theta \in [0, 2\pi)$.

**Proof.**

1. From Taylor's theorem, it is known that

$$\cos\theta = \sum_{n=0}^{\infty}\frac{(-1)^n\theta^{2n}}{(2n)!} = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \frac{\theta^8}{8!} - \ldots.$$

By grouping the terms of the same sign, we have that

$$\begin{aligned}\cos\theta = {} & \left[1 + \frac{\theta^4}{4!} + \frac{\theta^8}{8!} + \frac{\theta^{12}}{12!} + \ldots\right] \\ & - \left[\frac{\theta^2}{2!} + \frac{\theta^6}{6!} + \frac{\theta^{10}}{10!} + \ldots\right].\end{aligned} \tag{18}$$

Since $\theta \in [0, 2\pi)$, all of the powers of the form of $\theta^n$, $n \geq 0$, are convex [26, Chapter 3]. Thus, the above equation provides a DC decomposition of the cosine function when $\theta \in [0, 2\pi)$.

2. Similarly, the sine function is written in the following form:

$$\begin{aligned}\sin\theta = {} & \sum_{n=0}^{\infty}\frac{(-1)^n\theta^{2n+1}}{(2n+1)!} \\ = {} & \left[\theta + \frac{\theta^5}{5!} + \frac{\theta^9}{9!} + \frac{\theta^{13}}{13!} + \ldots\right] \\ & - \left[\frac{\theta^3}{3!} + \frac{\theta^7}{7!} + \frac{\theta^{11}}{11!} + \ldots\right],\end{aligned} \tag{19}$$

which is DC using similar arguments as above. □

**Lemma 3.** *The function $f(\theta,\alpha) = \frac{\theta}{\alpha} : \mathbb{R}\times\mathbb{R}^+ \to \mathbb{R}$ is DC with the following DC decomposition:*

$$f(\theta,\alpha) = \frac{\theta}{\alpha} = g(\theta,\alpha) - h(\theta,\alpha) = \frac{1}{2}\frac{(\theta+1)^2}{\alpha} - \frac{1}{2}\frac{(\theta^2+1)}{\alpha}. \tag{20}$$

**Proof.** We need to show that both parts are convex. The first part $g(\theta,\alpha) = \frac{(\theta+1)^2}{\alpha}$ is convex since

$$\begin{aligned}\nabla^2 g(\theta,\alpha) &= \frac{2}{\alpha^3}\begin{bmatrix}\alpha^2 & -(\theta+1)\alpha\\ -(\theta+1)\alpha & (\theta+1)^2\end{bmatrix} \\ &= \frac{2}{\alpha^3}\begin{bmatrix}\alpha\\ -(\theta+1)\end{bmatrix}\begin{bmatrix}\alpha\\ -(\theta+1)\end{bmatrix}^\top \succeq 0,\end{aligned}$$

when $\alpha > 0$. For the second part $h(\theta,\alpha) = \frac{(\theta^2+1)}{\alpha}$, we have that

$$\nabla^2 h(\theta,\alpha) = \frac{2}{\alpha^3}\begin{bmatrix}\alpha^2 & -\theta\alpha\\ -\theta\alpha & \theta^2+1\end{bmatrix} \succeq 0,$$

since $\alpha > 0$ and $v^\top\nabla^2 h(\theta,\alpha)v \geq 0$, for every $v = [v_1\ v_2]^\top \neq 0$. To see why this is true, observe that

$$\begin{aligned}&[v_1\ v_2]\begin{bmatrix}\alpha^2 & -\theta\alpha\\ -\theta\alpha & \theta^2+1\end{bmatrix}\begin{bmatrix}v_1\\ v_2\end{bmatrix} \\ &\quad = \alpha^2 v_1^2 - \theta\alpha v_1 v_2 - \theta\alpha v_1 v_2 + \theta^2 v_2^2 + v_2^2 \\ &\quad = (\theta v_2 - \alpha v_1)^2 + v_2^2 \geq 0.\end{aligned}$$

□

**Lemma 4.** *The function $f(\theta,\alpha) = \frac{\theta^k}{\alpha} : [0, 2\pi)\times\mathbb{R}^+ \to \mathbb{R}$, $k \geq 2$, is convex.*

**Proof.** The Hessian matrix of $f$ is

$$\nabla^2 f(\theta,\alpha) = \frac{\theta^{k-2}}{\alpha^3}\begin{bmatrix}k(k-1)\alpha^2 & -k\theta\alpha\\ -k\theta\alpha & 2\theta^2\end{bmatrix}.$$

Observe that the term $\frac{\theta^{k-2}}{\alpha^3}$ is positive, so we only need to prove that the remaining matrix is positive semidefinite. Call $\lambda_1$ and $\lambda_2$ its eigenvalues. Then, observe that its determinant is

$$\lambda_1\lambda_2 = \theta^2\alpha^2[2k(k-1) - k^2] = \theta^2\alpha^2 k(k-2) \geq 0$$

since $k \geq 2$. Thus, $\lambda_1$ and $\lambda_2$ have always the same sign. Now, observe also that the trace of this matrix is

$$\lambda_1 + \lambda_2 = k(k-1)\alpha^2 + 2\theta^2 > 0.$$

Thus, we conclude that both eigenvalues are nonnegative. Therefore, the Hessian matrix is positive semidefinite and $f$ is convex. □

**Lemma 5.** *The following functions are DC:*

1. $f_1(\theta, \alpha) = \frac{\cos\theta}{\alpha} : [0, 2\pi) \times \mathbb{R}^+ \to \mathbb{R}$.
2. $f_2(\theta, \alpha) = \frac{\sin\theta}{\alpha} : [0, 2\pi) \times \mathbb{R}^+ \to \mathbb{R}$.

**Proof.**

1. Using (18), we have that

$$\frac{\cos\theta}{\alpha} = \left[\frac{1}{\alpha} + \frac{\theta^4}{\alpha 4!} + \frac{\theta^8}{\alpha 8!} + \frac{\theta^{12}}{\alpha 12!} + \dots\right] - \left[\frac{\theta^2}{\alpha 2!} + \frac{\theta^6}{\alpha 6!} + \frac{\theta^{10}}{\alpha 10!} + \dots\right].$$

   Each of the individual parts is convex because it consists of a sum of convex functions according to Lemma 4.
2. Similarly, using (19), we have that

$$\frac{\sin\theta}{\alpha} = \left[\frac{\theta}{\alpha} + \frac{\theta^5}{\alpha 5!} + \frac{\theta^9}{\alpha 9!} + \frac{\theta^{13}}{\alpha 13!} + \dots\right] - \left[\frac{\theta^3}{\alpha 3!} + \frac{\theta^7}{\alpha 7!} + \frac{\theta^{11}}{\alpha 11!} + \dots\right].$$

   The second part of the sine is convex, again due to Lemma 4. Unfortunately, this is not the case for the first part since the function $\frac{\theta}{\alpha}$ is not convex. However, we know from Lemma 3 that it is DC. If we substitute the DC decomposition of $\frac{\theta}{\alpha}$ into the above formula, we get

$$\frac{\sin\theta}{\alpha} = \frac{1}{\alpha}\left[\frac{(\theta+1)^2}{2} + \frac{\theta^5}{5!} + \frac{\theta^9}{9!} + \frac{\theta^{13}}{13!} + \dots\right] - \frac{1}{\alpha}\left[\frac{(\theta^2+1)}{2} + \frac{\theta^3}{3!} + \frac{\theta^7}{7!} + \frac{\theta^{11}}{11!} + \dots\right]. \quad (21)$$

   The above formula now yields a DC decomposition of $\frac{\sin\theta}{a}$ since both parts are convex.

Note in passing that, in the DC decompositions of both $\frac{\cos\theta}{a}$ and $\frac{\sin\theta}{a}$, the convex parts are nonnegative. This property will be used later on. □

**Lemma 6.** *The following functions:*

$$\left\{\frac{\cos\theta\sin\theta}{\alpha^2}, \frac{\cos\theta\tau_x}{\alpha^2}, \frac{\sin\theta\tau_x}{\alpha^2}, \frac{\cos\theta\tau_y}{\alpha^2}, \frac{\sin\theta\tau_y}{\alpha^2}, \frac{\cos^2\theta}{\alpha^2}, \frac{\sin^2\theta}{\alpha^2}, \frac{\tau_x^2}{\alpha^2}, \frac{\tau_y^2}{\alpha^2}, \frac{\tau_x\tau_y}{\alpha^2}\right\},$$

*defined on $\theta \in [0, 2\pi)$, $\alpha \in \mathbb{R}^+$, $\tau_x \in \mathbb{R}$, $\tau_x \in \mathbb{R}$, are DC.*

**Proof.** According to Lemma 5, the functions $\frac{\cos\theta}{\alpha}$ and $\frac{\sin\theta}{\alpha}$ are DC with nonnegative convex parts. Also, the functions $\frac{\tau_x}{\alpha}$ and $\frac{\tau_y}{\alpha}$ are DC with nonnegative convex parts, due to Lemma 3. The fact that all of the above functions are DC results from property 3 of Proposition 2, which states that the product of two DC functions (with nonnegative convex parts) is also DC. □

Coming back to the transformed coordinates axes of the $k$th atom $\phi_{\eta_k}$ in (11), we can now show that the transformed coordinates system is represented by DC functions.

**Proof of Proposition 4.** According to Lemma 6, $\tilde{x}(\eta)^2$ (and similarly $\tilde{y}(\eta)^2$) in (17) is DC since it is a linear combination of DC functions (property 1 of Proposition 2). Thus, $z(\eta) = \tilde{x}(\eta)^2 + \tilde{y}(\eta)^2$ is also DC and suppose that $z(\eta) = g_z(\eta) - h_z(\eta)$ is its DC decomposition. □

### 5.3 DC Form of the Objective Function

Now, we are ready to prove the main result of our paper, namely Proposition 1, which claims that the objective function of the optimization problem (9) is DC. The construction of a geometric atom by transformation of the generating function is equivalent to apply the generating functions on the transformed coordinates computed above. Given the above developments, it finally remains to show that the transformed generating functions are DC and that the inner products between the atoms and the pattern of interest $p$ are also DC functions. We prove these properties for the two different generating functions $\phi$ used in this paper.

**Proof of Proposition 1.**

- If $\phi$ is Gaussian, then

$$\begin{aligned}\phi_\eta &\triangleq \phi(\tilde{x}(\eta), \tilde{y}(\eta)) = \frac{e^{-\left(\tilde{x}(\eta)^2 + \tilde{y}(\eta)^2\right)}}{\alpha\|\phi\|} = \frac{e^{-z(\eta)}}{\alpha\|\phi\|} \\ &= e^{-z(\eta) - \ln\alpha - \ln\|\phi\|} = e^{-[z(\eta) + \ln\alpha + \ln\|\phi\|]} \\ &= e^{-w(\eta)},\end{aligned}$$

  where we have introduced the function

$$\begin{aligned}w(\eta) &= z(\eta) + \ln\alpha + \ln\|\phi\| \\ &= [g_z(\eta) + \ln\|\phi\|] - [h_z(\eta) - \ln\alpha] \\ &= g_w(\eta) - h_w(\eta),\end{aligned} \quad (22)$$

  which is also DC since $\ln\alpha$ is a concave function of $\alpha$ and, therefore, $-\ln\alpha$ is convex. In the above formulas, $\|\phi\|$ denotes the norm of the nontransformed atom and $\alpha\|\phi\|$ is the norm of the transformed atom. Putting these pieces together, we conclude that $\phi_\eta$ is DC with the following decomposition $e^{-w(\eta)} = [e^{-w(\eta)} + K(g_w(\eta) + h_w(\eta))] - [K(g_w(\eta) + h_w(\eta))]$, according to Proposition 3, part 2.
- If $\phi$ is AR, then

$$\begin{aligned}\phi_\eta &\triangleq \phi(\tilde{x}(\eta), \tilde{y}(\eta)) = \frac{e^{-\left(\tilde{x}(\eta)^2 + \tilde{y}(\eta)^2\right)}}{\alpha\|\phi\|} \cdot \left[4\tilde{x}(\eta)^2 - 2\right] \\ &= 4\tilde{x}(\eta)^2 \frac{e^{-z(\eta)}}{\alpha\|\phi\|} - 2\frac{e^{-z(\eta)}}{\alpha\|\phi\|},\end{aligned}$$

which is also DC since it is a linear combination of two DC terms (the first term is DC since it is the product of two DC functions (property 3 of Proposition 2)).

Next, we need to show that the inner product $\zeta(\eta) \triangleq \langle \phi_\eta, p \rangle$ is also DC. Assume that $M$ is the number of pixels of the images and $\phi_m = g_m - h_m$ is the DC decomposition of the $m$th pixel of $\phi_\eta$. Then, $\zeta(\eta)$ is DC with the following decomposition:

$$\begin{aligned}\zeta(\eta) = \langle \phi_\eta, p \rangle &= \sum_{m=1}^{M} \phi_m p_m \\ &= \left[ \sum_{\{m:p_m \geq 0\}} p_m g_m - \sum_{\{m:p_m<0\}} p_m h_m \right] \\ &\quad - \left[ \sum_{\{m:p_m \geq 0\}} p_m h_m - \sum_{\{m:p_m<0\}} p_m g_m \right],\end{aligned}$$

based on property 1 of Proposition 2.

Assume now that $\langle \phi_{\eta_k}, p \rangle = g_k(\eta) - h_k(\eta)$ is the DC decomposition of the inner product of the $k$th atom with $p$. In the sequel, we again use property 1 of Proposition 2 to come up with the DC decomposition of $\sum_{k=1}^{K} \xi_k \langle \phi_{\eta_k}, p \rangle$ which reads

$$\begin{aligned}\sum_{k=1}^{K} \xi_k \langle \phi_{\eta_k}, p \rangle &= \underbrace{\left[ \sum_{\{k:\xi_k \geq 0\}} \xi_k g_k - \sum_{\{k:\xi_k<0\}} \xi_k h_k \right]}_{g} \\ &\quad - \underbrace{\left[ \sum_{\{k:\xi_k \geq 0\}} \xi_k h_k - \sum_{\{k:\xi_k<0\}} \xi_k g_k \right]}_{h} \\ &= g(\eta) - h(\eta).\end{aligned}$$

Finally, the objective function in (11) is DC with the following decomposition, following from the property 2 of Proposition 2:

$$f(\eta) = \left| \sum_{k=1}^{K} \xi_k \langle \phi_{\eta_k}, p \rangle \right| = 2 \max\{g, h\} - (g + h).$$

□

**Algorithm 1**. Outer approximation cutting plane

1: **Initialization:** Set $\omega^0 = g(y^0) - h(y^0)$, the first upper bound of the optimal value $\omega^*$ of the Problem (23).
2: Compute a subgradient $s \in \partial g(y^0)$ and construct the affine function $l(x) = (x - y^0)^\top s + g(y^0)$.
3: Construct a simplex $S^0 \supseteq X$ with vertex set $V(S^0) = \{v^1, \ldots, v^{n+1}\}$. Choose $\bar{\omega}$ and $\bar{t}$ such that $\bar{\omega} = \min\{l(x) : x \in V(S^0)\} - \max\{h(x) : x \in V(S^0)\}$ $\bar{t} > \max\{g(x) : x \in V(S^0)\} - \bar{\omega}$.
4: Construct a polytope $P^0 = \{(x,t) : x \in S^0,\ t \leq \bar{t},\ l(x) - t - \omega^0 \leq 0\} \supseteq \{(x,t) : x \in X,\ t \in \mathbb{R},\ g(x) - t - \omega^* = 0\}$ and compute its vertex set $V(P^0) = \{(v^i, \bar{t}), i = 1, \ldots, n+1$ and $(v^i, l(v^i) - \omega^0), i = 1, \ldots, n+1\}$.
5: Set $k = 0$.
6: **Iteration:**
7: Compute an optimal solution $(x^k, t^k)$ of the problem $\min\{-h(x) + t : (x,t) \in V(P^k)\}$
8: **if** $-h(x^k) + t^k = 0$ **then**
9: $y^k$ is the optimal solution with optimal value $\omega^k$.
10: **else**
11: Compute $s^k \in \partial g(x^k)$
12: Compute the improved upper bound $\omega^{k+1} = \min\{\omega^k, g(x^k) - h(x^k)\}$.
13: Update $y^{k+1}$ such that $g(y^{k+1}) - h(y^{k+1}) = \omega^{k+1}$.
14: Construct the cutting plane $l^k(x,t) = (x - x^k)^\top s^k + g(x^k) - \omega^{k+1} - t$
15: Set $P^{k+1} = P^k \cap \{(x,t) : l^k(x,t) \leq 0\}$ and compute $V(P^{k+1})$.
16: **end if**
17: Set $k = k + 1$ and go to step 7.

The objective function is therefore a DC function, which permits the application of DC programming methods for finding the global minimum to the MD computation. Note that the DC decomposition of $f$ is both inefficient and complicated to be obtained in closed form with respect to $\eta$. On the contrary, the values of $g$ and $h$ can be obtained in practice by sequentially applying the above properties. Actually, this can be done at the cost of evaluating only one of them (that is, the evaluation of $g$ yields the corresponding value of $h$ for free and vice versa).

## 6 Solution of the DC Program

### 6.1 The Cutting Plane Algorithm

After showing that our objective function is DC, we discuss in this section how one can solve a DC program. A global optimization problem is called a DC program if it has the following form:

$$\begin{aligned}\min_x \quad & f(x) = g(x) - h(x), \\ \text{s.t.} \quad & x \in X = \{x \in \mathbb{R}^n : \delta(x) \leq 0\},\end{aligned} \tag{23}$$

where $g, h : X \to \mathbb{R}$ are convex functions and $\delta : \mathbb{R}^n \to \mathbb{R}$ is a convex function. Assume that the DC Problem (23) is solvable and denote by $\omega^*$ its global minimum. The next proposition gives an optimality condition for Problem (23).

**Proposition 5 [22].** *The point $x^* \in X$ is an optimal solution to Problem (23) if and only if there exists a $t^* \in \mathbb{R}$ such that*

$$0 = \inf\{-h(x) + t :\ x \in X,\ t \in \mathbb{R},\ g(x) - t \leq g(x^*) - t^*\}. \tag{24}$$

In this work, we have chosen to solve the DC Program (23) by the outer approximation cutting plane algorithm proposed in [22, Section 5.3], for its simplicity and also due to the fact that the parameter space in our problem is 4D. However, we should mention that our framework could also be combined with other DC solvers such as Branch-and-Bound schemes [22, Sections 5.1 and 5.2] and DCA [27].

The cutting plane algorithm, as well as any DC solver, needs an explicit DC decomposition of the objective function before it can be employed in practice. In the sequel, we briefly discuss the basics of the cutting plane algorithm, illustrated in Algorithm 1. The cutting plane

algorithm seeks a point $x^*$ that will satisfy the optimality condition (24). Each iteration involves the minimization of the concave function $-h(x)+t$ under the convex constraints $g(x)-t \leq \omega^k$, where $\omega^k$ is the best upper bound for $\omega^*$ as of iteration $k$. Thus, the right-hand side of the convex constraint changes over the iterations. The algorithm starts with an initial feasible point $y^0$ which yields an initial upper bound $\omega^0$. As soon as this bound improves, the convex constraint is refined. In the first iteration, the cutting plane method starts with a simple polyhedron $P^0$, which is required to contain the feasible set. The polyhedron $P^0$ is defined by the intersection of several halfspaces, i.e., linear inequalities of the form $a_i^\top x + b_i \leq 0$, $i = 1, \ldots, m$.

It is known that a concave function defined over a convex polyhedron is minimized at one of its vertices [23, Theorem 1.19]. In iteration $k$, the vertex $x^k$ which minimizes $-h(x)+t$ over $P^k$ (see Line 7) is used to define a cutting plane $l^k(x,t)$ which updates the polytope, by excluding points that do not satisfy the linear constraint $l^k(x,t) \leq 0$. The intersection of $P^k$ with the cutting plane defines a new polytope $P^{k+1}$ of smaller volume (see Line 15). The minimization of $-h(x)+t$ is repeated on the updated $P^{k+1}$ and the algorithm continues by repeating the same process. The objective function $f$ is evaluated on each vertex of $P^k$ and, whenever a better bound of $\omega^*$ is found, $\omega^k$ is updated (see Line 12) and the convex constraint is updated as well. This process is repeated until $\omega^k$ reaches $\omega^*$ and, in this case, the vertex which achieves $-h(x)+t=0$ will satisfy the optimality condition (24). The cutting plane algorithm always converges to the global minimizer of Problem (23) [22, Theorem 5.3].

The cutting plane algorithm is an efficient algorithm which typically reaches the vicinity of the global minimizer quickly. Each step involves only function and subgradient evaluations, as well as vertex updates of the current polytope $P^k$. The latter process is called *vertex enumeration* and the reader is referred to [23, Section 3.3.4] for further details. We have chosen to omit the details of vertex enumeration due to lack of space.

### 6.2 Computational Cost Analysis

Let us provide an analysis of the computational complexity of the proposed methodology. Notice that each step of a DC solver, such as the cutting plane, calls for function and subgradient evaluations. Since each subgradient evaluation calls for two function evaluations, it is sufficient to study the cost of function evaluations alone. Before delving into the cost analysis, it is important to differentiate between the complexity of evaluating the DC decomposition of $f$ and the complexity of the DC solver. The latter depends on the particular characteristics of the DC solver that is employed and it is beyond the scope of the present work.

We turn our attention now to the cost of evaluating the convex parts $g$ and $h$ of the objective function $f$ in (11). Recall that the evaluation of $g$ yields the corresponding value of $h$ for free and vice versa. This is due to the fact that the values of $g(\eta_0)$ and $h(\eta_0)$ at a certain point $\eta_0$ of the parameter space are obtained simultaneously by applying sequentially the properties described in Section 5. Overall, the computational complexity of each function evaluation is

$$C_{\text{comp}} = O(K \cdot n_1 \cdot n_2),$$

where $K$ is the number of atoms and $M = n_1 \cdot n_2$ is the number of pixels of the image. This arises from the fact that a pattern is represented by $K$ atoms and every pixel of each atom is expressed in DC form. Note that $K$ is typically small (e.g., less than 100) and it is independent of the image size. It solely depends on the geometric complexity (shape) of the pattern and the properties of the dictionary. Therefore, we observe that the sparsity of the pattern and the image resolution are the main factors affecting the cost of function evaluations.

## 7 Experimental Results

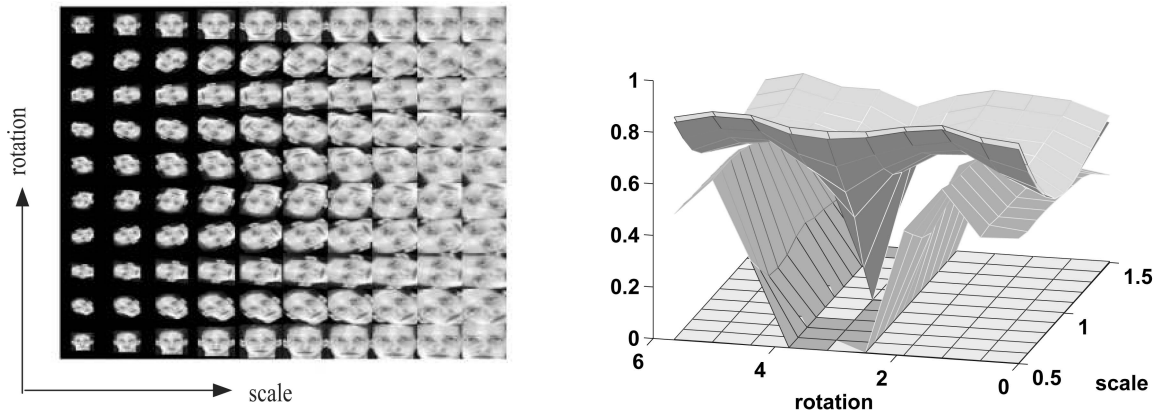
In this section, we compare the proposed method, called MDDC (i.e., manifold distance measure using DC programming) with related methods from the literature. We have observed in practice that our method quickly reaches the vicinity of the global minimizer, but then the convergence rate drops before it computes the global minimizer with high accuracy. This is a common characteristic of cutting plane methods in general. On the other hand, Newton is a very accurate and fast method, but it requires an initial guess that is close to the global one. For this reason, we combine both methods to get the best of both worlds. Thus, what we call MDDC is essentially a hybrid method which first employs the cutting plane method to reach the vicinity of the global minimizer and then switches to Newton with the initial parameters that are obtained from the cutting plane algorithm.

### 7.1 Comparison of Different Methods

In the first experiment, we compare MDDC with ED, TD, and MD (using Newton's method). For our comparison, we use a facial image from the ORL database [28] (see Section 7.2 for more details on this database). Fig. 4a shows how this face transforms in the scale-rotation transformation space. Since the MD (semantic) between all these facial images is zero, we would like our methods to give a distance between them which is as small as possible. Hence, the smaller the obtained distance, the better the method.

We build a parametric model of the facial image with $K = 50$ Gaussian atoms (see (10)) using OMP. We sample the transformation parameter space by sampling the rotation angle $\theta$ uniformly in $[0, 2\pi)$ with step $\pi/5$ and the scale $\alpha$ in $[0.5, 1.5]$ with step 0.1. For each combination of scale and rotation values, we build a transformed image $s(\eta)$ using (8) and employ it as the reference image $p$. Then, we compute the distance between $s$ and $p$ using all methods and compare the results. In the TD method, the tangent vectors are computed by finite differences using (8) and the manifold is linearized around $\theta = 0$ and $\alpha = 1$ (which corresponds to the nontransformed pattern). In the case of MD and MDDC, we report the distance $\|s(\hat{\eta}) - s(\eta^*)\|_2$, where $\eta^*$ and $\hat{\eta}$ is the exact and estimated transformation, respectively. Note that, in MDDC, we use 100 iterations of the cutting plane method before switching to Newton.

Fig. 4b shows the obtained distance surfaces from all methods. First observe that the ED surface is nonconvex, which reveals the difficulty of the problem in practical scenarios. Notice that ED and TD have comparable




(a) (b)

Fig. 4. Invariance of various methods with respect to scale and rotation. (a) Transformed faces. (b) Distance surfaces (from top to bottom) ED, TD, MD, and MDDC.

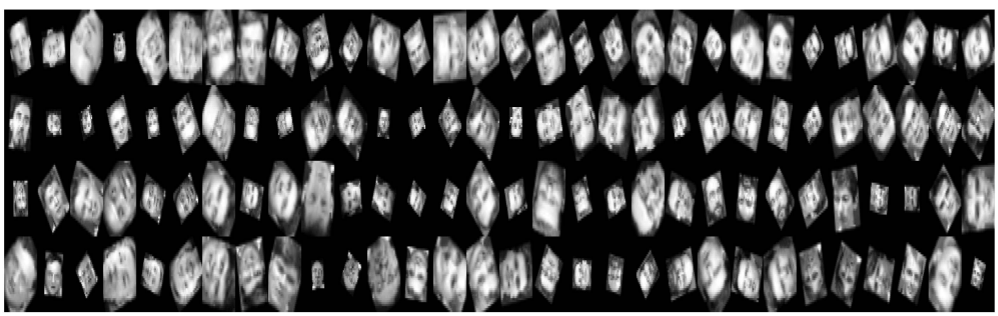


Fig. 5. The transformed test face images used for face recognition.

performance, with the latter bringing a minor improvement over the former. Next, observe that Newton's method provides a dramatic improvement over the previous methods. We see also that its local convergence property is also verified in practice, i.e., for small transformations it is able to converge to the global minimizer. However, for transformations of larger magnitude it gets trapped in local minima. Finally, MDDC seems to provide superior results compared to the other methods. It provides almost zero distance over all pairs of scale and rotation, verifying its ability to converge to the globally optimal solution. Thus, it corresponds indeed to a close approximation of the semantic distance between the transformed faces.

### 7.2 Transformation-Invariant Face Recognition

In the second experiment, we compare all methods in the context of transformation-invariant face recognition. We use the ORL (formerly Olivetti) database [28], which contains 40 individuals and 10 different images for each individual, including variation in facial expression (smiling/nonsmiling) and pose. We form the training set by picking the first seven facial expressions/poses per subject and use the remaining three as a test set. This results in 280 training samples and 120 test samples. Our evaluation metric is the classification error rate which is the percentage of test samples that are misclassified.

First, we compute the pattern model of all training images using $K = 50$ Gaussian atoms in OMP. This is an offline step and is performed once and for all. Then, each method is combined with the nearest-neighbor (NN) classifier yielding a transformation-invariant classifier. In particular, each test face image is first geometrically transformed and then compared to each training face image. The test sample is given the class label of the "closest" training image. The distance between the test and the training image is computed according to the method that is tested: 1) For ED, it is the euclidean distance between the two, 2) for TD, it is the tangent distance, and 3) for MD and MDDC, it is the euclidean distance obtained after alignment. In the latter case, alignment is accomplished by estimating the transformation parameters first and then by image warping using the estimated parameters. For computational ease, the test image is aligned with only one representative image per subject, namely, the first facial expression/pose. Then, the same transformation is used for aligning the test image with the remaining images of the same subject.

Note that this experiment is more challenging than the previous one because the training and the test images (which are to be aligned) correspond to different expressions/poses of the same person or to different persons. In MDDC, we use 30 iterations of the cutting plane method before switching to Newton. In the testing phase, each test facial image is scaled by a random $\alpha$, uniformly distributed in $[0.5, 1.5]$ and rotated by a random $\theta$, uniformly distributed in $[0, 2\pi)$. Each test image is transformed independently from its peers. Observe that the testing phase scales linearly with the training set size. Fig. 5 shows the transformed versions of the test facial images.

TABLE 1
Transformation-Invariant Face Recognition Results

| Classif. error rate(%) | ED | TD | MD | VS | MDDC |
|---|---|---|---|---|---|
| L2 | 95.8 | 98.3 | 80.8 | 37.5 | 26.6 |
| L1 | 94.2 | - | 81.6 | 29.2 | 22.5 |

For the sake of completeness, we also include in our comparisons the virtual samples (VS) method. The latter is employed as a baseline method in order to show that expanding the training set with transformed samples can be helpful in robust recognition against the presence of transformations in the test set. In particular, for each training image, we build a set of VS by applying geometric transformations corresponding to each possible pair $(\theta_i, \alpha_j)$, $i = 1, \ldots, n_\theta$, $j = 1, \ldots, n_s$. We discretize the rotation space $[0, 2\pi)$ with $n_\theta = 10$ points and the scale space [0.5, 1.5] with $n_s = 5$ points. In the VS method, the size of the training set is expanded by $n_\theta \cdot n_s$ times, which is 50 in our case. This makes it memory intensive, which is one of the main pitfalls of this method. Note that one may discretize even more finely the transformation space, but we should bear in mind that in practice this may not be always feasible. This method typically bridges the gap between local methods (i.e., that are built for small transformations) and global methods, like ours.

Table 1 shows the classification error rates of all methods, under both L1 and L2 metrics. Notice that ED, TD, and MD break down due to the presence of large transformations. However, MDDC yields a satisfactory performance, which shows its truly transformation-invariant properties. The VS method has comparable performance to MDDC, though at the cost of highly increased memory requirements.

We also discuss the alignment performance of Newton and MDDC. We compute the histogram of the absolute errors of both scale and rotation using 50 bins. In particular, we report the absolute errors $|\theta - \hat{\theta}|$ and $|\alpha - \hat{\alpha}|$, where $\hat{\theta}$ and $\hat{\alpha}$ denote the estimated transformation parameters of rotation and scale, respectively. We observed that, in many cases, the global minimizer in the alignment between *different* persons does not coincide with the exact transformation parameters. For this reason, in the histogram computation, we include only the alignment performance of the pairs corresponding to different expressions/poses of the *same* subject. Fig. 6 shows the results of both Newton and MDDC. Notice that, as expected, Newton is not able to converge to the true transformation parameters, due to its local convergence property. On the other hand, MDDC is successful in the vast majority of the cases. The very few outlier cases are due to the fact that we run only 30 iterations of the cutting plane algorithm (before switching to Newton).

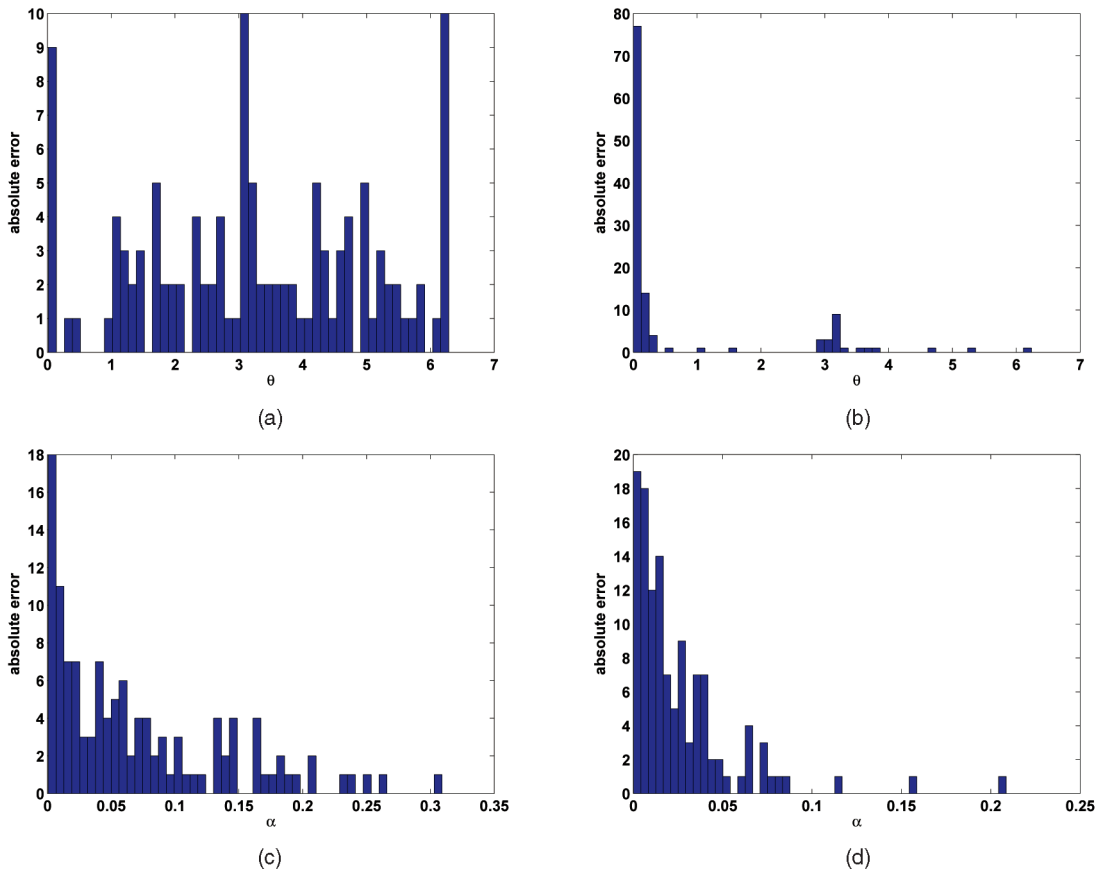
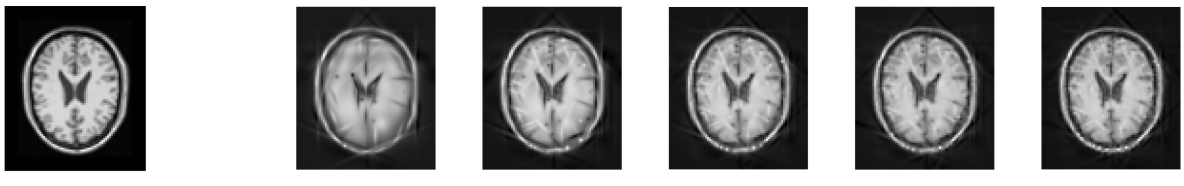
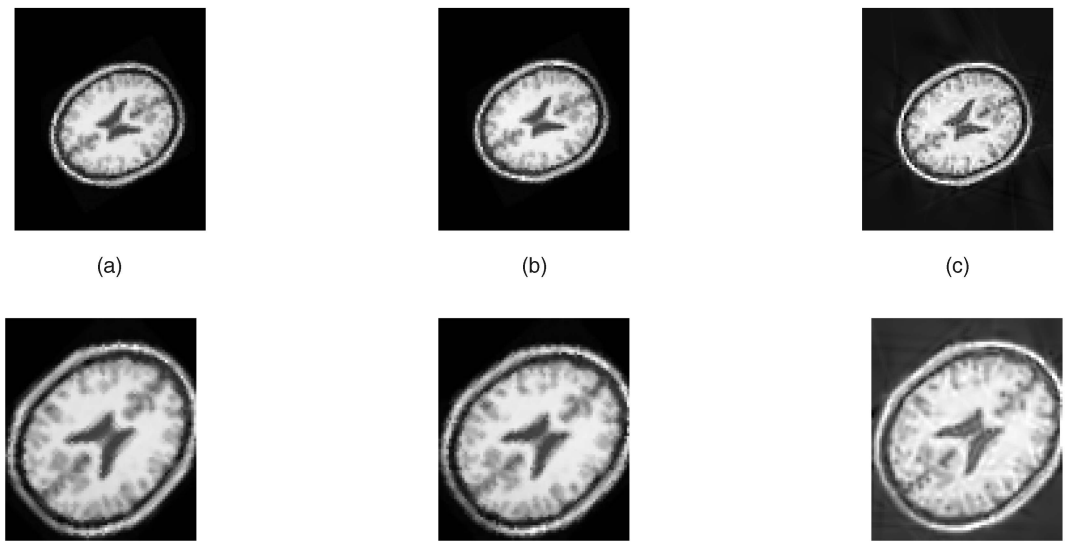



Fig. 6. Histogram of the absolute error of the rotation angle (first row) and of the scale (second row). (a) $|\theta - \hat{\theta}|$, Newton. (b) $|\theta - \hat{\theta}|$, MDDC. (c) $|\alpha - \hat{\alpha}|$, Newton. (d) $|\alpha - \hat{\alpha}|$, MDDC.



(a) (b)

Fig. 7. The MRI brain image and its successive approximations. (a) MRI image. (b) OMP approximations using 100, 200, 300, 400, and 500 atoms, respectively (left to right).



(a) (b) (c)

(d) (e) (f)

Fig. 8. The transformed MRI brain images and their estimated transformed images. The first row corresponds to $\eta_1$ and the second row to $\eta_2$. (a) Transformed image with $\eta_1$. (b) Estimated transformed image. (c) $s(\hat{\eta}_1)$. (d) Transformed image with $\eta_2$. (e) Estimated transformed image. (f) $s(\hat{\eta}_2)$.

### 7.3 Application to Image Registration

Finally, we provide an illustrative example of the application of our method to medical image registration. In this example, we consider $\eta$ to be a synthesis of translations, scalings, and rotations. We consider a typical brain MRI image $s$ as shown in Fig. 7a. We build a pattern model of $s$ using OMP with $K = 500$ Gaussian atoms. Fig. 7b shows the obtained successive approximations while increasing the number of atoms from 100 to 500 with step 100. Observe that a few atoms are sufficient to capture the main geometric structure of the pattern. Hence, in the context of alignment, one does not need to compute a very accurate and expensive approximation of $s$. In other words, it is possible to estimate the transformation parameters using only a crude pattern approximation, provided that it captures its salient geometric characteristics. For this reason, we use only $K = 100$ atoms in the following alignment experiments.

We perform two experiments with two different transformations. First, we apply a geometric transformation $\eta_1 = (\beta_1, \alpha_1, \theta_1)$ on $s$, where $\beta_1 = [3,\ 2]^\top$, $\alpha_1 = 0.8$, and $\theta_1 = \pi/3 \simeq 1.047$. The transformed image after image warping with $\eta_1$ is shown in Fig. 8a. We apply the cutting plane method and, after 500 iterations, the estimated transformation parameters are: $\hat{\beta}_1 = [3.15,\ 0.61]^\top$, $\hat{\alpha}_1 = 0.8$, and $\hat{\theta}_1 = 1.08$. Fig. 8b shows the warped image using the estimated transformation parameters $\hat{\eta}_1$. Also, in Fig. 8c, we show the transformed pattern $s(\hat{\eta}_1)$ (see (8)) using 500 atoms (for illustration purposes).

In the second experiment, we test with a new transformation $\eta_2 = (\beta_2, \alpha_2, \theta_2)$, with $\beta_2 = [1,\ 4]^\top$, $\alpha_2 = 1.2$, and $\theta_2 = 5\pi/4 \simeq 3.93$. The transformed image after image warping with $\eta_2$ is shown in Fig. 8d. After 500 iterations of the cutting plane algorithm, the estimated transformation parameters are $\hat{\beta}_2 = [1.54,\ 4.33]^\top$, $\hat{\alpha}_2 = 1.19$, and $\hat{\theta}_2 = 4$. In Figs. 8e and 8f, we show the warped image with $\hat{\eta}_2$ and $s(\hat{\eta}_2)$, respectively. Observe that, in both cases, the cutting plane algorithm is able to converge very close to the global minimizer. Therefore, if it is combined with more accurate local convergence methods, it can potentially yield a very accurate and robust method for image registration.

### 7.4 Discussion

The above experimental results indicate that our algorithm converges to the global minimizer in the presence of large transformations when they consist of synthesis of translations, rotations, and scalings. In this section, our goal is to show that our method is applicable to robust pattern classification and image registration problems, while at the same time enjoying theoretical merits of convergence to the globally optimal solution. These examples are, however, mostly illustrative and do not permit to extrapolate that the proposed framework alone will provide state-of-the-art

performances in more generic problems, like image registration in the presence of occlusion or background noise.

Also, we can note that the proposed framework has targeted transformations that can be represented as a composition of rotation, translation, and isotropic scaling. While such transformations represent a wide range of image processing applications, the proposed framework could also be applied to more generic transformations models, such as fully affine, nonrigid, etc., as long as the effect of the transformation on a set of atoms can be properly expressed in a DC form. In our future work, we plan to pursue the above problems and explore the potential of our method for specific image processing applications.

## 8 Conclusions

We have proposed a globally optimal method for computing a transformation-invariant distance measure. When a pattern is geometrically transformed, it spans a manifold in the high-dimensional space. We use sparse geometric expansions to represent the transformation manifold. A few atoms are sufficient to capture the main geometric structure of the pattern, which is further used for alignment. We formulate the minimization of the distance of the reference pattern from the manifold, as a DC program, by proving that the objective function is DC. We solve the DC program using an outer approximation-cutting plane method which converges to the globally optimal solution. The experimental results show that the proposed method is successful in finding the global minimizer in practice, with applications to robust face recognition and image alignment.

## Appendix A

### A.1 Proof of Proposition 3, Part 2

**Proof.** Let $f(x) = g(x) - h(x)$ be the DC decomposition of $f$. Then, the second part of (14) is convex since $g$ and $h$ are convex. Hence, we need to show that $p(x)$ is convex. Note that the $(i,j)$ entry of the Hessian of $p(x)$ is

$$\begin{aligned}
H_{ij} &= \frac{\partial^2 p(f(x))}{\partial x_i \partial x_j} = \frac{\partial^2 q(f(x))}{\partial f^2} \cdot \frac{\partial f(x)}{\partial x_i} \cdot \frac{\partial f(x)}{\partial x_j} \\
&\quad + \frac{\partial q(f(x))}{\partial f} \cdot \frac{\partial^2 f(x)}{\partial x_i \partial x_j} + K\left[\frac{\partial^2 g(x)}{\partial x_i \partial x_j} + \frac{\partial^2 h(x)}{\partial x_i \partial x_j}\right] \\
&= q''(f(x)) \cdot \frac{\partial f(x)}{\partial x_i} \cdot \frac{\partial f(x)}{\partial x_j} + q'(f(x)) \cdot \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \\
&\quad + K\left[\frac{\partial^2 g(x)}{\partial x_i \partial x_j} + \frac{\partial^2 h(x)}{\partial x_i \partial x_j}\right] \\
&= q''(f(x)) \cdot \frac{\partial f(x)}{\partial x_i} \cdot \frac{\partial f(x)}{\partial x_j} + [K + q'(f(x))] \frac{\partial^2 g(x)}{\partial x_i \partial x_j} \\
&\quad + [K - q'(f(x))] \frac{\partial^2 h(x)}{\partial x_i \partial x_j}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
H = \nabla_p^2 &= q''(f(x)) \nabla_f \nabla_f^\top \\
&\quad + [K + q'(f(x))] \nabla_g^2 + [K - q'(f(x))] \nabla_h^2.
\end{aligned} \tag{25}$$

The convexity of $q$ implies that $q''(f(x)) \geq 0$. Also, the convexity of $g$ and $h$ implies that $\nabla_g^2$ and $\nabla_h^2$ are positive semidefinite. Hence, when $K \geq |q'(f(x))|$, we conclude that $H$ is positive semidefinite and $p(x)$ is convex. □

### A.2 Proof of Lemma 1

**Proof.** Suppose that the atom has parameters $\gamma = (b, a, \omega)$, with $a = [a_x, a_y]$ and $b = [b_x, b_y]$. If we denote by $\eta = (\beta, \alpha, \theta)$ the transformation parameters, then, according to the group law, the transformed parameters of the atom will be

$$(\eta \circ \gamma) = (\beta + \alpha R(\theta) b,\ \alpha a,\ \omega - \theta).$$

If we denote

$$A = \begin{pmatrix} \frac{1}{\alpha a_x} & 0 \\ 0 & \frac{1}{\alpha a_y} \end{pmatrix}$$

and

$$R(\omega) = \begin{pmatrix} \gamma_1 & \gamma_2 \\ \gamma_3 & \gamma_4 \end{pmatrix},$$

then the transformed axes $\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}$ will be

$$\begin{aligned}
&AR(\omega)R(-\theta)\left[\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} \beta_x \\ \beta_y \end{pmatrix} - \alpha R(\theta)\begin{pmatrix} b_x \\ b_y \end{pmatrix}\right] \\
&= AR(\omega)\left[R(-\theta)\begin{pmatrix} x \\ y \end{pmatrix} \underbrace{-R(-\theta)\begin{pmatrix} \beta_x \\ \beta_y \end{pmatrix}}_{[\tau_x\ \tau_y]^\top} - \alpha\begin{pmatrix} b_x \\ b_y \end{pmatrix}\right] \\
&= AR(\omega)\left[R(-\theta)\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} -\alpha b_x + \tau_x \\ -\alpha b_y + \tau_y \end{pmatrix}\right] \\
&= AR(\omega)\begin{bmatrix} \cos\theta & -\sin\theta & -\alpha b_x + \tau_x \\ \sin\theta & \cos\theta & -\alpha b_y + \tau_y \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= A\begin{bmatrix} \gamma_1 & \gamma_2 \\ \gamma_3 & \gamma_4 \end{bmatrix}\begin{bmatrix} \cos\theta & -\sin\theta & -\alpha b_x + \tau_x \\ \sin\theta & \cos\theta & -\alpha b_y + \tau_y \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \epsilon_1 \frac{\cos\theta}{\alpha} + \epsilon_2 \frac{\sin\theta}{\alpha} & -\epsilon_1 \frac{\sin\theta}{\alpha} + \epsilon_2 \frac{\cos\theta}{\alpha} & \epsilon_3 + \epsilon_4 \frac{\tau_x}{\alpha} + \epsilon_5 \frac{\tau_y}{a} \\ \zeta_1 \frac{\cos\theta}{\alpha} + \zeta_2 \frac{\sin\theta}{\alpha} & -\zeta_1 \frac{\sin\theta}{\alpha} + \zeta_2 \frac{\cos\theta}{\alpha} & \zeta_3 + \zeta_4 \frac{\tau_x}{\alpha} + \zeta_5 \frac{\tau_y}{a} \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix},
\end{aligned}$$

where we have introduced $\epsilon_1 = \frac{\gamma_1}{a_x}$, $\epsilon_2 = \frac{\gamma_2}{a_x}$, $\epsilon_3 = \frac{-b_x\gamma_1 - b_y\gamma_2}{a_x}$, $\epsilon_4 = \epsilon_1$, $\epsilon_5 = \epsilon_2$, and, similarly, $\zeta_1 = \frac{\gamma_3}{a_y}$, $\zeta_2 = \frac{\gamma_4}{a_y}$, $\zeta_3 = \frac{-b_x\gamma_3 - b_y\gamma_4}{a_y}$, $\zeta_4 = \zeta_1$, and $\zeta_5 = \zeta_2$. Then, the above formula reads

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} (\epsilon_1 x + \epsilon_2 y)\frac{\cos\theta}{\alpha} + (\epsilon_2 x - \epsilon_1 y)\frac{\sin\theta}{\alpha} + \epsilon_4 \frac{\tau_x}{\alpha} + \epsilon_5 \frac{\tau_y}{\alpha} + \epsilon_3 \\ (\zeta_1 x + \zeta_2 y)\frac{\cos\theta}{\alpha} + (\zeta_2 x - \zeta_1 y)\frac{\sin\theta}{\alpha} + \zeta_4 \frac{\tau_x}{\alpha} + \zeta_5 \frac{\tau_y}{\alpha} + \zeta_3 \end{pmatrix},$$

which yields (15). □

## ACKNOWLEDGMENTS

The authors would like to thank Dr. Meritxell Bach Cuadra for providing the MRI brain image. This work has been partly supported by NCCR IM2.

## REFERENCES

[1] R.F.I Ventura, P. Vandergheynst, and P. Frossard, "Low Rate and Flexible Image Coding with Redundant Representations," *IEEE Trans. Image Processing,* vol. 15, no. 3, pp. 726-739, Mar. 2006.
[2] E. Kokiopoulou and P. Frossard, "Image Alignment with Rotation Manifolds Built on Sparse Geometric Expansions," *Proc. IEEE Int'l Workshop Multimedia Signal Processing,* Oct. 2007.
[3] N. Vasconcelos and A. Lippman, "A Multiresolution Manifold Distance for Invariant Image Similarity," *IEEE Trans. Multimedia,* vol. 7, no. 1, pp. 127-142, Feb. 2005.
[4] P. Simard, Y.L. Cun, J. Denker, and B. Victorri, "Transformation Invariance in Pattern Recognition—Tangent Distance and Tangent Propagation," *Neural Networks: Tricks of the Trade,* pp. 239-274, 1998.
[5] D. Keysers, J. Dahmen, T. Theiner, and H. Ney, "Experiments with an Extended Tangent Distance," *Proc. 15th IEEE Int'l Conf. Pattern Recognition,* vol. 2, pp. 38-42, 2000.
[6] A. Fitzgibbon and A. Zisserman, "Joint Manifold Distance: A New Approach to Appearance Based Clustering," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2003.
[7] D. DeCoste and M. Burl, "Distortion-Invariant Recognition via Jittered Queries," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2000.
[8] D. Decoste and B. Schölkopf, "Training Invariant Support Vector Machines," *Machine Learning,* vol. 46, no. 1, pp. 161-190, 2002.
[9] B. Haasdonk and D. Keysers, "Tangent Distance Kernels for Support Vector Machines," *Proc. 16th IEEE Int'l Conf. Pattern Recognition,* vol. 2, pp. 864-868, 2002.
[10] B. Schölkopf, C. Burges, and V. Vapnik, "Incorporating Invariances in Support Vector Learning Machines," *Artificial Neural Networks—ICANN,* vol. 96, pp. 47-52, 1996.
[11] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik, "Prior Knowledge in Support Vector Kernels," *Proc. Advances in Neural Information Processing Systems,* vol. 10, pp. 640-646, 1998.
[12] T. Graepel and R. Herbrich, "Invariant Pattern Recognition by Semidefinite Programming Machines," *Proc. Advances in Neural Information Processing Systems,* vol. 16, pp. 33-40, 2004.
[13] P. Thévenaz, U.E. Ruttimann, and M. Unser, "A Pyramidal Approach to Subpixel Registration Based on Intensity," *IEEE Trans. Image Processing,* vol. 7, no. 1, pp. 27-41, Jan. 1998.
[14] A. Neri and G. Jacovitti, "Maximum Likelihood Localization of 2-D Patterns in the Gauss-Laguerre Transform Domain: Theoretic Framework and Preliminary Results," *IEEE Trans. Image Processing,* vol. 13, no. 1, pp. 72-85, Jan. 2004.
[15] M. Carli, F. Coppola, G. Jacovitti, and A. Neri, "Translation, Orientation and Scale Estimation Based on Laguerre-Gauss Circular Harmonic Pyramids," *Proc. SPIE, Image Processing: Algorithms and Systems,* vol. 4667, pp. 55-65, 2002.
[16] B. Zitová and J. Flusser, "Image Registration Methods: A Survey," *Image and Vision Computing,* vol. 21, no. 11, pp. 977-1000, 2003.
[17] S. Mallat, *A Wavelet Tour of Signal Processing,* second ed. Academic Press, 1998.
[18] A.J. Smola and B. Schölkopf, "Sparse Greedy Matrix Approximation for Machine Learning," *Proc. Int'l Conf. Machine Learning,* pp. 911-918, 2000.
[19] B. Schölkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Dec. 2001.
[20] P. Jost, P. Vandergheynst, and P. Frossard, "Tree-Based Pursuit: Algorithm and Properties," *IEEE Trans. Signal Processing,* vol. 54, no. 12, pp. 4685-4697, 2006.
[21] J. Nocedal and S.J. Wright, *Numerical Optimization,* P. Glynn and S.M. Robinson, eds. Springer, 1999.
[22] R. Horst and N.V. Thoai, "DC Programming: Overview," *J. Optimization Theory and Applications,* vol. 102, no. 1, pp. 1-43, Oct. 1999.
[23] R. Horst, P.M. Pardalos, and N.V. Thoai, *Introduction to Global Optimization,* second ed. Kluwer Academic Publishers, 2000.
[24] R. Horst and P.M. Pardalos, *Handbook of Global Optimization,* Kluwer Academic Publishers, 1995.
[25] S. Ellis and M. Nayakkankuppam, "Phylogenetic Analysis via DC Programming (Preprint)," Dept. of Math. and Statistics, Univ. of Maryland, Baltimore County, 2003.
[26] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge Univ. Press, 2004.
[27] L.T.H. An and P.D. Tao, "The DC (Difference of Convex Functions) Programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems," *Annals of Operations Research,* vol. 133, pp. 23-46, 2005.
[28] F. Samaria and A. Harter, "Parameterisation of a Stochastic Model for Human Face Identification," *Proc. Second IEEE Workshop Applications of Computer Vision,* Dec. 1994.

**Effrosyni Kokiopoulou** received the diploma in engineering from the University of Patras, Greece, in 2002 and the MSc degree in computer science from the University of Minnesota in 2005 under the supervision of Professor Yousef Saad. She is currently working toward the PhD degree under the supervision of Professpr Pascal Frossard. In September 2005, she joined the LTS4 Lab of the Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. Her research interests include multimedia data mining, machine learning, computer vision, and numerical linear algebra. She is a student member of the IEEE.

**Pascal Frossard** received the MS and PhD degrees in electrical engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1997 and 2000, respectively. Between 2001 and 2003, he was a member of the research staff at the IBM T.J. Watson Research Center, Yorktown Heights, New York, where he worked on media coding and streaming technologies. Since 2003, he has been an assistant professor at EPFL, where he heads the Signal Processing Laboratory (LTS4). His research interests include image representation and coding, nonlinear representations, visual information analysis, joint source and channel coding, multimedia communications, and multimedia content distribution. He was the general chair of the IEEE ICME 2002 and Packet Video 2007 and a member of the organizing or technical program committees of numerous conferences. He is an associate editor of the *IEEE Transactions on Multimedia* (2004-) and of the *IEEE Transactions on Circuits and Systems for Video Technology* (2006-). He is an elected member of the IEEE Image and Multidimensional Signal Processing Technical Committee (2007-), the IEEE Visual Signal Processing and Communications Technical Committee (2006-), and the IEEE Multimedia Systems and Applications Technical Committee (2005-). He served as a vice-chair of the IEEE Multimedia Communications Technical Committee (2004-2006) and as a member of the IEEE Multimedia Signal Processing Technical Committee (2004-2007). He received the Swiss NSF Professorship Award in 2003 and the IBM Faculty Award in 2005. He is a senior member of the IEEE.