Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Optimal similarity registration of volumetric images

E. Kokiopoulou, D. Kressner, M. Zervos[*] and N. Paragios[†]

Research Report No. 2011-18
March 2011

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

[*]Dept. of Informatics and Telecommunications, University of Athens, Greece
[†]Laboratoire MAS, Ecole Centrale Paris, France

Optimal Similarity Registration of Volumetric Images

Effrosyni Kokiopoulou
Seminar for Applied Mathematics
ETH Zurich
effrosyni.kokiopoulou@sam.math.ethz.ch

Michail Zervos
Dept. of Informatics and Telecommunications
University of Athens
std04079@di.uoa.gr

Daniel Kressner
Seminar for Applied Mathematics
ETH Zurich
daniel.kressner@sam.math.ethz.ch

Nikos Paragios
Laboratoire MAS, Ecole Centrale Paris, France
Equipe GALEN, INRIA Saclay - Ile-de-France
nikos.paragios@ecp.fr

Abstract

This paper proposes a novel approach to optimally solve volumetric registration problems. The proposed framework exploits parametric dictionaries for sparse volumetric representations, ℓ^1 dissimilarities and DC (Difference of Convex functions) decomposition. The SAD (sum of absolute differences) criterion is applied to the sparse representation of the reference volume and a DC decomposition of this criterion with respect to the transformation parameters is derived. This permits to employ a cutting plane algorithm for determining the optimal relative transformation parameters of the query volume. It further provides a guarantee for the global optimality of the obtained solution, which – to the best of our knowledge – is not offered by any other existing approach. A numerical validation demonstrates the effectiveness and the large potential of the proposed method.

1. Introduction

Registration [1] is a fundamental problem in computer vision and in particular in medical image analysis. It is an elementary step towards bringing various volumetric data into the same reference space, which in turn permits to gather statistics and exploit similarities across subjects.

Geometric and iconic methods are often used to address this problem. Geometric methods [2] extract characteristic landmarks between two images, and then seek the optimal transformation that establishes geometric correspondences between the images. Unfortunately, such an approach may be very sensitive to the landmark extraction process. Furthermore, solving the correspondence problem between landmarks, which is a pre-step of the registration, is highly nontrivial. Often, robust EM-like methods are used for this purpose. These methods iteratively determine the optimal transformation for a set of correspondences and then improve the correspondences based on this transformation. Naturally such a method may converge to a local minimum, mostly due to erroneous correspondences.

Iconic methods [1] employ a (dis)similarity criterion on the observation space that is a function of rigid transformation parameters, which are optimized to minimize / maximize this criterion. The selection of the criterion and the optimization method are the two critical components of iconic registration. SAD, SSD, NCC, CR [3], as well as complex statistical metrics [4] in the case of multi-modal data have been considered. The optimization of the criterion is often performed using descent-like methods that are sensitive to initial conditions and do not provide guarantees on the optimality of the obtained solution. Recently the use of global optimization frameworks such as discrete MRFs was suggested [5]. However, the dimensionality of the resulting continuous search space makes its quantization quite problematic and even inefficient and therefore the results are far from being optimal.

Despite an enormous effort in the field [6], none of the existing methods can guarantee optimality of the obtained solution even in the case of volumes coming from the same modality. In this paper we propose a novel approach that estimates optimal transformation parameters. Global optimality is achieved through the expression of the objective function as a DC (difference of convex functions) decomposition and with the use of a cutting plane algorithm to estimate the optimal registration parameters.

Input volumes are sparsely represented over a redundant dictionary of geometric atoms. Using such a representation, the set of all transformations of a certain volume (which constitutes the so-called transformation manifold) admits a closed form expression with respect to the transformation parameters. This relation is used to derive a ℓ^1 criterion between the two volumes in terms of the registration parameters. Using basic theorems on DC functions [7, 8, 9], we prove that the resulting objective function admits a DC decomposition with respect to the rigid transformation parameters.

Once a DC decomposition is established, a number of algorithms are available to solve the optimization problem in an efficient and robust manner [7]. In this paper, we propose a modified version of the cutting plane algorithm [7, Thm 5.3] and use it to recover the optimal registration parameters. The modifications are introduced in order to accelerate the convergence of the original cutting plane algorithm. In addition, we implement the DC function evaluation on a graphics processing unit (GPU), in order to further speed up our method. We provide numerical experiments demonstrating the effectiveness and the global optimality property of our method.

The rest of this paper is organized as follows. In Section 2, we briefly present the sparse geometric representations of volumes as well as the corresponding transformation manifolds. Section 3 is devoted to the definition of the registration problem. Next, Section 4 provides some background material on DC functions and Section 5 introduces our modified cutting plane method. We discuss our GPU implementation of the DC function evaluation in Section 6. In Section 7, we present numerical experiments of our approach, followed by some conclusions in Section 8.

2. Volume transformation manifolds

In the following, we define and characterize the transformation manifold of a certain volume. For this purpose, we represent the volume by a parametric sparse model extracted from a dictionary of geometric functions. Such a geometric representation leads to a closed form expression for the transformation manifold, which is used in the computation of ℓ^1 dissimilarity measures.

2.1. Sparse atomic volumetric representations

We represent the volume of interest as a linear combination of geometric functions (usually called *atoms*), taken from a parametric and (typically overcomplete) dictionary $\mathcal{D} = \{\phi_\gamma,\ \gamma \in \Gamma\}$ spanning the input volume space. This representation generally captures the most prominent geometric features of the volume. The atoms in $\mathcal{D}$ are constructed by applying geometric transformations to a generating function denoted by ϕ. Representing the geometric transformation $\gamma \in \Gamma$ by an operator $U(\gamma)$, the parametric dictionary takes the form

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

In this work, a transformation $\gamma = (a, R, b) \in \Gamma$, will denote a synthesis of translations $\vec{b} \in \mathbb{R}^{3\times 1}$, anisotropic scalings $\vec{a} \in \mathbb{R}_+^{3\times 1}$ and rotations $R \in SO(3)$. The dictionary is built from three-dimensional atoms that can efficiently capture the salient geometrical features in volumetric images.

A sparse approximation of a given volume $v \in \mathbb{R}^{n_1\times n_2\times n_3}$ with atoms from the dictionary $\mathcal{D}$ can be obtained in various ways. Even if finding the sparsest approximation of v is generally a hard problem, effective suboptimal solutions are usually sufficient to capture the salient geometric structures of a signal with only a few atoms. Such solutions are obtained, for example, by Orthogonal Matching Pursuit (OMP) [10, Sec. 9.5.3] and Tree-based Pursuit [11], to name just a few. In this work we use Tree-based Pursuit, which organizes the dictionary in a tree structure and admits significantly faster searches over the dictionary compared to OMP. Hence, this provides an effective algorithm for computing sparse volume approximations in practice. After K steps of the algorithm, the volume v is approximated by a sparse linear combination of a few atoms i.e.,

$$v = \sum_{k=1}^{K} \xi_k \phi_{\gamma_k} + r_K, \tag{2}$$

where r_K is the residual of the approximation. In what follows we will assume that r_K is negligible and can be dropped.

2.2. Characterization of transformation manifolds

The set of all geometric transformations γ applied to a certain volume v generates a manifold $\mathcal{M}$ in the high-dimensional ambient observation volume space. Each point on this manifold corresponds to a transformed version of v. In the following, we only consider transformations $\eta = (s, G, t)$ consisting of a synthesis of translations $t = [t_x, t_y, t_z]$, *isotropic* scaling $s \in \mathbb{R}_+$ and rotations $G \in SO(3)$. Then the transformation manifold $\mathcal{M}$ can be expressed as follows:

$$\mathcal{M} = \{v(\eta) \equiv U(\eta)v,\ \text{where}\ \eta = (s, G, t)\}. \tag{3}$$

Note that although the manifold is embedded in a high-dimensional space, its intrinsic dimension is rather small and equals the number of transformation parameters.

The transformations η form a group, namely the similitude group SIM(3) in $\mathbb{R}^3$. If (a, R, b) and (a', R', b') are two elements from SIM(3) then the group law is

$$(a, R, b) \circ (a', R', b') = (aa', RR', b + aRb'). \tag{4}$$

Using (2) and dropping the residual term r_K, it turns out that applying the transformation η to the volume v results in

$$v(\eta) = U(\eta)v = \sum_{k=1}^{K} \xi_k U(\eta)\phi_{\gamma_k} = \sum_{k=1}^{K} \xi_k \phi_{\eta\circ\gamma_k}, \tag{5}$$

where $\eta \circ \gamma_k$ is a product of transformations. In other words, the transformation is applied to each constituent atom individually, resulting in a sparse representation of the transformed volume over atoms with updated parameters. The group law (4) indeed applies [12] and can be further employed to work out the updated parameters of the transformed atoms. Equation (5) is of great importance in the proposed approach, since it expresses the manifold (3) in closed form with respect to the transformation parameters η. This is a key observation for the applicability of the DC programming methodology that is proposed in this work.

3. Rigid registration

After having introduced sparse geometric representations and transformation manifolds, we are now ready to provide the problem formulation. We are interested in estimating the transformation between two volumes. Suppose that we are given a query volume p, and we aim to estimate the optimal transformation parameters η^* that best align v with p. We formulate the transformation estimation problem as follows

$$\eta^* = \arg \min_{\eta=(s,G,t)} f(\eta), \quad \text{where} \quad f(\eta) = \|v(\eta) - p\|_1. \quad (6)$$

Here, $\|p\|_1 = \sum_{ijk} |p_{ijk}|$ denotes the ℓ^1 norm of a volume $p \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. The criterion (6) is also known as the sum of absolute differences (SAD) criterion.

Recall that $v(\eta) \in \mathcal{M}$ denotes the transformed volume v subject to a transformation $\eta = (s, G, t)$. We assume that the reference volume v has been well approximated by a sparse expansion over $\mathcal{D}$ according to (2) where r_K is negligible. Note that in the above optimization problem, only the reference volume v is expanded in the redundant basis and the query volume p is treated as is.

The optimization problem (6) is generally a non-convex nonlinear optimization problem [13] and hard to solve using traditional methods. For example, steepest descent or Newton-type methods converge only locally and may get trapped in local minima. To avoid these issues, we will exploit that the above objective function is a DC function with respect to the transformation parameters, i.e., it can be expressed as the difference of two convex functions.

Theorem 1 *The objective function*

$$f(\eta) = \|v(\eta) - p\|_1 = \left\| \sum_{k=1}^{K} \xi_k \phi_{\eta_k} - p \right\|_1, \quad (7)$$

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

The proof of this theorem is given in the appendix. The proof is constructive and provides a procedure for evaluating the two convex parts of f. Using Theorem 1, the optimization problem (6) can be formulated as a DC program [7, 8, 9], which can be optimally solved by exploiting the special structure of the objective function. In this paper, we employ a cutting plane method to solve the DC formulation of (6). The proposed method is guaranteed to converge to the global minimizer. To the best of our knowledge, this is the first globally optimal algorithm that is proposed for the problem of rigid volume registration.

4. DC functions

Before we introduce the proposed algorithm, we start with some background material about DC functions [7, 8, 9]. Let X be a convex subset of $\mathbb{R}^n$. A function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is called DC [7, 8, 9] on X, if there exist two convex functions $g, h : X \to \mathbb{R}$ such that f is expressed as

$$f(x) = g(x) - h(x). \quad (8)$$

A representation of the above form is called a DC decomposition of f. The DC decomposition of f is not unique, since one can obtain a different decomposition by adding the same convex function $c(x)$ in both convex parts of it. We provide below an example of a DC function.

Example 1 *Consider the function $f(x) = cos(x)$, $x \in [0, 2\pi)$ and suppose that we want to determine a DC decomposition of it. In other words, we seek two convex functions $g(x)$ and $h(x)$ such that $f(x) = g(x) - h(x)$. We know from Taylor's theorem that*

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

Grouping together the terms of the same sign, one obtains

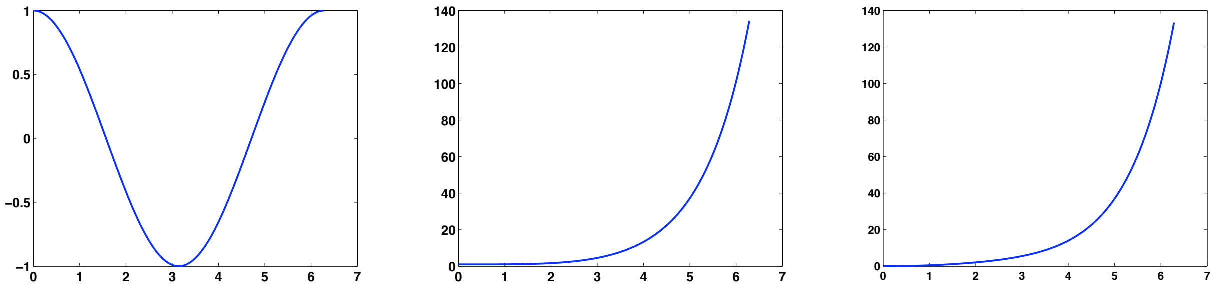
$$\cos x = \underbrace{\left[1 + \frac{x^4}{4!} + \ldots\right]}_{g(x)} - \underbrace{\left[\frac{x^2}{2!} + \frac{x^6}{6!} + \ldots\right]}_{h(x)}, \quad (9)$$

which readily provides a DC decomposition of $f(x)$, since $g(x)$ and $h(x)$ are convex functions. The obtained decomposition is illustrated in Fig. 1.

The example above shows that even a highly non-linear and non-convex function such as the cosine, can be decomposed into two convex parts giving rise to a special structure that can be further exploited for global optimization. We present now a few properties of DC functions.

Proposition 1 ([8, Sec 4.2]) *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:*

(a) $$\sum_{i=1}^{m} \lambda_i f_i = \Big[\sum_{\{i:\lambda_i \geq 0\}} \lambda_i g_i - \sum_{\{i:\lambda_i < 0\}} \lambda_i h_i \Big] - \Big[\sum_{\{i:\lambda_i \geq 0\}} \lambda_i h_i - \sum_{\{i:\lambda_i < 0\}} \lambda_i g_i \Big].$$



(a) $cos(x)$

(b) $g(x)$

(c) $h(x)$

Figure 1. The function $cos(x)$ and its convex parts $g(x)$ and $h(x)$ in (9).

(b) $|f| = 2\max\{g, h\} - (g + h)$.

(c) *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:*

$$f_1 \cdot f_2 = \frac{1}{2}[(g_1 + g_2)^2 + (h_1 + h_2)^2] - \frac{1}{2}[(g_1 + h_2)^2 + (g_2 + h_1)^2]. \quad (10)$$

In addition, it can be shown that the synthesis of a convex function and a DC function is again DC, which is particularly important for our further developments.

Proposition 2 *Let $f(x) : \mathbb{R}^n \to \mathbb{R}$ be DC and $q : \mathbb{R} \to \mathbb{R}$ be convex. Then,*

(a) *the composition $q(f(x))$ is DC [8, Sec 4.2].*

(b) *$q(f(x))$ has the following DC decomposition:*

$$q(f(x)) = p(x) - K[g(x) + h(x)], \quad (11)$$

where $p(x) = q(f(x)) + K[g(x) + h(x)]$ is a convex function and K is a constant satisfying $K \geq |q'(f(x))|$ [14, 15].

In the next section we discuss the proposed algorithm in more details.

5. Proposed cutting plane method

An optimization problem is called a DC program if it takes the form

$$\begin{aligned} \min_x \quad & f(x) = g(x) - h(x), \quad (12) \\ \text{s.t.} \quad & x \in X = \{x \in \mathbb{R}^n : \delta(x) \leq 0\}, \end{aligned}$$

where $g, h : X \to \mathbb{R}$ are convex functions and $\delta : \mathbb{R}^n \to \mathbb{R}$ is a convex function. Denote by ω^* the global minimum of (12). The next proposition provides an optimality condition for (12).

Proposition 3 ([7]) *The point $x^* \in X$ is an optimal solution to the DC problem (12) if and only if there exists $t^* \in \mathbb{R}$ such that*

$$\begin{aligned} 0 = \; & \inf\{-h(x) + t : \; x \in X, \; t \in \mathbb{R}, \\ & g(x) - t \leq g(x^*) - t^*\}. \quad (13) \end{aligned}$$

One may solve optimally the DC program (12) using the cutting plane method of [7, Thm 5.3], which is briefly discussed in the sequel. The cutting plane algorithm seeks a point x^* that satisfies the global optimality condition (13). Each iteration involves the minimization of the concave function $-h(x) + t$ over a convex constraint set of the form $\mathcal{C}^k := \{g(x) - t \leq \omega^k\}$, where ω^k is the best upper bound for ω^* as of iteration k. The cutting plane method uses a polytope that provides an outer approximation of the set $\mathcal{C}^k$; hence, the minimizer x^k of the concave function $-h(x) + t$ can be readily found at one of the extreme vertices of the polytope. A cutting plane is defined based on x^k and further used to refine the polytope by excluding points that are guaranteed to violate the constraints $\mathcal{C}^k$. The same process is repeated in the next iteration. Once a better bound ω^k of ω^* has been found in the course of the algorithm, the constraint set $\mathcal{C}^k$ is updated. More details about the cutting plane method can be found in [7, Thm 5.3].

We introduce two modifications to the standard cutting plane method described above, in order to accelerate its convergence. The main steps of the modified cutting plane method are summarized in Algorithm 1.

- First, we use *two* cutting planes in each iteration in order to make the cut more effective. To define the two cutting planes, we use the minimizer and the second best minimizer[1] of $-h(x) + t$ over the polytope (see also Lines 7 and 19). Empirically we found no further improvement by using more than two cutting planes; thus, we employ two cutting planes in our experiments.

[1] Actually, we mean the next possible minimizer which is not removed by the cutting plane of the best minimizer.

Algorithm 1 Modified Cutting Plane Algorithm

```
1: Initialization: Set $\omega^0 = g(y^0) - h(y^0)$, the first upper bound of the optimal value $\omega^*$ of the Problem (12).
2: Construct a polytope $P^0$ that contains $\{(x,t) : x \in X,\ t \in \mathbb{R},\ g(x) - t - \omega^* = 0\}$.
3: Compute the vertex set $V(P^0)$ of the polytope $P^0$.
4: $f^0_{\min} = \infty$.
5: Set $k = 0$.
6: Iteration:
7: Compute the minimizer $(x^k_1, t^k_1)$ and second best minimizer $(x^k_2, t^k_2)$ of the problem:
       $\min\{-h(x) + t : (x,t) \in V(P^k)\}$.
8: if $-h(x^k_1) + t^k_1 = 0$ then
9:     $y^k$ is the optimal solution with optimal value $\omega^k$.
10: else
11:     Compute $s^k_i \in \partial g(x^k_i),\ i = 1, 2$.
12:     $f^{k+1}_{\min} = \min\{g(x) - h(x) : (x,t) \in V(P^k)\}$.
13:     if $f^{k+1}_{\min} < f^k_{\min}$ then
14:         Identify $\bar{x}^k$ such that $f^{k+1}_{\min} = f(\bar{x}^k)$.
15:         $f^{k+1}_{\min}$ = subgradientDescent($\bar{x}^k$).
16:     end if
17:     Compute the improved upper bound $\omega^{k+1} = \min\{\omega^k, f^{k+1}_{\min}\}$.
18:     Update $y^{k+1}$ such that $g(y^{k+1}) - h(y^{k+1}) = \omega^{k+1}$.
19:     Construct the cutting planes:
            $l^k_i(x,t) = (x - x^k_i)^\top s^k_i + g(x^k_i) - \omega^{k+1} - t,\ i = 1, 2$.
20:     Set $P^{k+1} = P^k \cap \{(x,t) : l^k_1(x,t) \le 0, l^k_2(x,t) \le 0\}$ and compute $V(P^{k+1})$.
21: end if
22: Set $k = k + 1$ and go to step 6.
```

- Second, we employ subgradient descent as a means to further improve the bound for ω^* (see also Lines 12-17). Experimental evidence suggests that this modification often greatly accelerates the convergence of the method. It is important to emphasize that subgradient descent has only an auxiliary role: it is only used as a means to provide improved bounds ω^{k+1} to the global minimizer ω^* in the early iterations of the algorithm. Of course, the global optimality property of the proposed modified cutting plane method still holds, and this is verified in practice as will be shown below in the numerical experiments.

6. Computational aspects

Evaluating the DC decomposition of the objective function f scales proportionally with $K \cdot n_1 \cdot n_2 \cdot n_3$, due to the fact that the DC decomposition needs to be evaluated for each voxel of the K atoms of size $n_1 \times n_2 \times n_3$. In order to accelerate the computational speed, we perform the function evaluation on the graphics processing unit (GPU). In what follows, we provide an overview of the GPU implementation.

volume size	CPU	GPU
128x128x10	12.1 sec	390ms
64x64x20	5.2 sec	190ms
32x32x20	2.5 sec	60ms

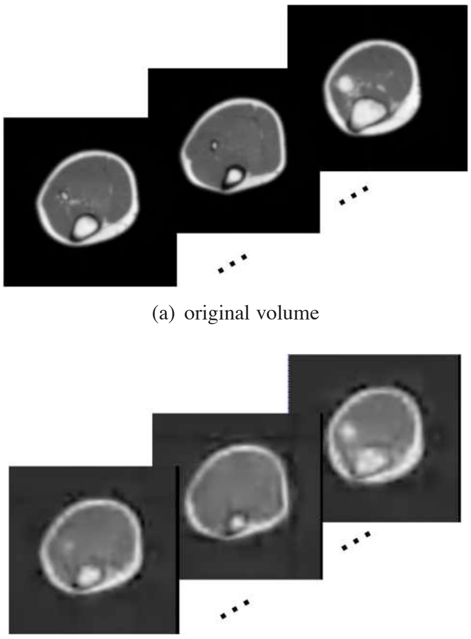
Table 1. Average timings of CPU and GPU implementations.

GPU architecture and CUDA. We make use of a CUDA enabled GPU; more specifically, we used the NVIDIA Tesla C1060. This processor consists of 30 multiprocessors, with each of them containing a collection of 8 CUDA cores. Following the SIMD model, a CUDA function, called kernel, is executed simultaneously by each thread on every CUDA core, i.e., there are 240 threads for the Tesla C1060. Threads are organized in blocks and the blocks in turn are organized in a grid.

The memory of a CUDA device is organized in a hierarchical way. All threads have access to the global memory, threads within a block share an on-chip, fast memory space called shared memory and each thread has its own private memory space and registers. We refer to the CUDA Programming Guide for further details.

GPU implementation. Since the evaluation of the DC decomposition is not only the computationally most expensive task but also well structured and highly parallelizable, it is well suited for GPU implementation. To compute this decomposition for a vertex in the polytope generated by the cutting plane method, the decomposition is evaluated at every voxel and summed up. In our GPU implementation, the loops over the vertices and the voxels are parallelized, leaving only one loop over the atoms in CPU code. A 2-D grid of blocks is created, with the blocks in the ith grid row evaluating the ith vertex. Each thread evaluates the decomposition of one voxel. The final summation for each vertex is performed using a parallel reduction algorithm as described in [16]. Vectors like μ, ν and ζ, see Lemma 1, that are required for the decomposition of every vertex are computed once and stored in the global memory, along with the sparse representation of the volume. Since not all vertices may fit into global memory, they are split into several batches. The fast, on-chip, shared memory is used as cache, accelerating memory accesses and enabling higher bandwidth.

Benchmarks. We compare the execution of the CPU and GPU implementations using an Intel Xeon X5570 (2.93



(a) original volume

(b) sparse volume representation (1000 atoms)

Figure 2. A sample MRI calf volume and its sparse representation.

GHz) and the NVIDIA Tesla C1060, respectively. For the benchmarks we used volumes of sizes $32 \times 32 \times 20$, $64 \times 64 \times 20$ and $128 \times 128 \times 10$, decomposed into $K = 50$ atoms. The comparison of the average time needed for evaluating the DC decomposition is shown in Table 1, demonstrating a remarkable speedup.

7. Numerical experiments

We apply the method proposed in this work to medical imaging. For this purpose, we use MRI calf muscle volumes, which have been obtained with 0.7812x0.7812x7 mm voxel spacing using a 1.5T Siemens scanner. Each volume slice is 7mm thick. Figure 2 shows a sample volume used in our experiments and its sparse representation with 1000 atoms. It is important to emphasize that for the purposes of rigid alignment, one does not actually need an accurate sparse approximation of the reference shape [15]. In practice, relatively few atoms providing a crude shape approximation are typically sufficient for successful alignment.

7.1. 5D parameter space

In the first experiment, we consider transformations $\eta = (q_0, q_1, q_2, q_3, s)$ consisting of five parameters, where q_0, q_1, q_2, q_3 are the quaternion parameters of the 3D rotation and s is the (isotropic) scaling (see the Appendix A for more details on the transformation parametrization). The reference volume consists of a $128 \times 128 \times 10$ sparse representation of the calf volume with $K = 50$ atoms, and it is to be aligned with a query volume that is a rotated and scaled version of it. The ground-truth transformation parameters are $\eta^* = (q_0, q_1, q_2, q_3, s) = (-0.73, -0.36, 0.54, 0.18, 0.8)$. We give the query volume as input to the proposed cutting plane method which converged to the exact transformation parameters η^* in 25 iterations. We did the same alignment experiment using subgradient descent that converged to the estimate $(\hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{s}) = (0.11, 0.55, -0.66, 0.5, 0.8)$, which is clearly suboptimal. Hence, we verify in practice the global optimality properties of our algorithm and its superiority over subgradient descent that is typically trapped in a local minimum.

Next, for the sake of completeness, we do the same experiment with $64 \times 64 \times 20$ volume sizes and $K = 100$ atoms in the sparse representation of the reference volume. We used the same ground-truth transformation parameters η^* as above. Our cutting plane method converged to η^* in 15 iterations, whereas subgradient descent converged to $(\hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3, \hat{s}) = (0.67, -0.23, 0.64, -0.3, 0.79)$, which is suboptimal as well.

7.2. 8D parameter space

In the sequel, we consider a full transformation $\eta = (q_0, q_1, q_2, q_3, t_1, t_2, t_3, s)$ consisting of eight parameters: q_0, q_1, q_2, q_3 are the quaternion parameters of the 3D rotation, t_1, t_2, t_3 are the translation parameters and s is the (isotropic) scaling. In our experiment, we used $\eta^* = (-0.963, 0.053, -0.03, 0.26, 3, 2, 1, 0.8)$ as a ground-truth transformation. The sparse representation of the reference volume consists of $K = 20$ atoms of size $32 \times 32 \times 20$. After 1003 iterations, the cutting plane provided the following transformation estimate: $\hat{\eta} = (-0.969, 0.051, -0.033, 0.24, 2.91, 2.26, 0.25, 0.798)$.
Observe that the scaling and rotation parameters are almost exact and the translation parameters have been estimated with sub-pixel accuracy. Hence, the estimate from our method is indeed very close to the exact transformation η^*. On the contrary, the estimate obtained from subgradient descent was found to be $(-0.028, -0.027, 0.102, -0.994, -3.35, -3.00, 3.18, 0.8)$, which is again far from the optimal solution.

7.3. Warped query volumes

So far, we have only considered query volumes that have been constructed by applying a transformation on the sparse representation of the reference volume. In this experiment, we consider warped query volumes, where the intensity value at each transformed voxel position is obtained with nearest neighbor interpolation. This setup makes the alignment experiment even more challenging due to the involved approximation error, since the query volume does not exactly represent a transformed version of the sparse approximation of the reference volume.

We try to align a $32 \times 32 \times 20$ sparse representation of the reference volume with $K = 100$ atoms with a warped query volume corresponding to transformation $\eta^* = (q_0, q_1, q_2, q_3, s) = (0.924, 0.125, -0.327, 0.151, 1.2)$. After 107 iterations, our cutting plane method provided the estimate $\hat{\eta} = (0.927, 0.127, -0.321, 0.147, 1.15)$ that is very close to the optimal transformation η^*. Hence, our method was still able to compute the optimal solution, despite the fact that the query volume does not exactly lie on the transformation manifold of the sparse reference volume.

Next, we explore the robustness of the method against different sparse representations of the reference volume. This also helps studying the impact of the information loss due to the sparse representation. To this end, we repeated the above experiment with different sparse representations obtained with increasing number of atoms. The obtained results are shown in Table 2, where the *End Point Difference* (EPD) corresponds to the distance between the optimally transformed voxel position and the one obtained with the estimated transformation. The *Angular Error* (AE) corresponds to the angle between them when treated as three-dimensional vectors. Note that the results for $K = 100$ atoms have been already reported above.

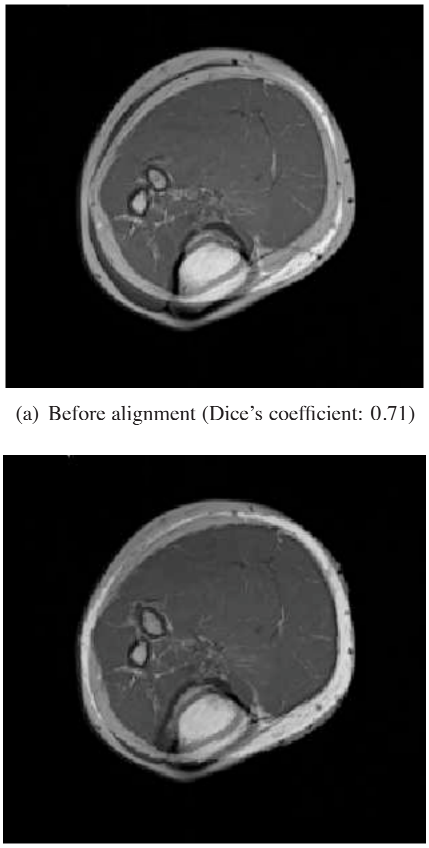
Atoms	Estimated transformation
10	(0.934, 0.131, -0.317, 0.092, 1.142)
30	(0.920, 0.126, -0.339, 0.143, 1.157)
50	(0.922, 0.118, -0.332, 0.156, 1.154)
100*	(0.927, 0.127, -0.321, 0.147, 1.152)
200	(0.927, 0.125, -0.321, 0.146, 1.156)

Atoms	Mean EPD (std)	Mean AE (std)
10	1.3508 (0.4953)	0.1032 (0.0222)
30	0.4917 (0.1565)	0.0233 (0.0070)
50	0.4822 (0.1485)	0.0150 (0.0053)
100*	0.4846 (0.1491)	0.0117 (0.0035)
200	0.4624 (0.1429)	0.0130 (0.0037)

Table 2. Performance of the method with different sparse representations.

Observe that once the sparse representation of the reference volume becomes reasonable (e.g., more than 30 atoms), the solution is very close to the optimal one. Although increasing the number of atoms reduces the approximation error, it does not really influence the estimated transformation which remains very close to the optimal transformation (notice that in all cases the error is below voxel accuracy).

A few remarks are in order. First, we confirm that for the purpose of linear registration where one seeks an optimal global estimation of the parameters, a very accurate approximation of the reference volume is not necessary. Second, although the sparse representation of the reference volume induces an information loss, it does not harm the optimality of the method, provided that the representation sufficiently captures the pattern shape. Moreover, there is no issue with the multiple approximations due to the redundancy of the dictionary. As a last note we emphasize that the approximation doesn't even have to be sparse. The sparsity of the representation can only impact the computational cost of the function evaluation, and it cannot compromise the global optimality of the obtained solution.



(a) Before alignment (Dice's coefficient: 0.71)

(b) After alignment (Dice's coefficient: 0.77)

Figure 3. Atlas obtained from the two volumes: before and after the alignment.

7.4. Alignment between two different volumes

In this last experiment, we consider the even more challenging case where another volume (i.e., from a different person) is used as a query volume, and we try to align it with the reference volume. In this case, the optimal transformation is unknown. The sparse representation of the reference volume consists of $K = 50$ atoms of size $128 \times 128 \times 20$. After 1416 iterations of the cutting plane method, the obtained transformation estimate is $\hat{\eta} = (-0.998, 0.01, 0.007, 0.051, 1.087)$, implying that the respective rotation is very small and that the query volume is slightly larger than the reference volume (see also Fig. 3). The Dice's coefficient has increased from 0.71 (before alignment) to 0.77 (after alignment), which is very satisfactory for a rigid alignment method.

8. Conclusions

We have proposed a globally optimal method for rigid registration between volumetric images by transformation parameter estimation. The proposed methodology is based on sparse volumetric representations. We have shown that under such a representation, the ℓ^1 similarity is a DC function of the transformation. The rigid registration problem becomes then equivalent to a DC optimization problem that can be optimally solved. We have proposed a modified cutting plane method for computing the globally optimal solution to the above problem. Finally, we have presented experimental results that (i) verify in practice the global optimality property of the method (ii) demonstrate its superiority over other descent-type of methods and finally show its large potential.

9. Acknowledgements

The first author would like to thank Prof. Pascal Frossard for his valuable feedback regarding the paper.

References

[1] J. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, pages 1–16, 1998.

[2] A. Rangarajan, H. Chui, and J. Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, pages 425–440, 1999.

[3] A. Roche, X. Pennec et al. Generalized correlation ratio for rigid registration of 3d ultrasound with mr images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 567 – 577, 2000.

[4] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. In *IEEE Transactions on Medical Imaging*, pages 187–19, 1997.

[5] B. Glocker, D. Zikic, N. Komodakis, N. Paragios, and N. Navab. Linear image registration through MRF optimization. In *IEEE International Symposium on Biomedical Imaging*, 2009.

[6] J. West, M. Fitzpatrick et al. Comparison and evaluation of retrospective intermodality brain image registration techniques. *Journal of Computer Assisted Tomography*, pages 554–568, 1997.

[7] R. Horst and N. V. Thoai. DC Programming: Overview. *Journal of Optimization Theory and Applications*, 102(1):1–43, October 1999.

[8] R. Horst, P. M. Pardalos, and N. V. Thoai. *Introduction to Global Optimization*, volume 48 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, 2nd edition, 2000.

[9] R. Horst and P. M. Pardalos. *Handbook of Global Optimization*, volume 2 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, 1995.

[10] S. Mallat. *A Wavelet Tour of Signal Processing, 2nd edn*. Academic Press, 1998.

[11] P. Jost, P. Vandergheynst, and P. Frossard. Tree-based pursuit: Algorithm and properties. *IEEE Transactions on Signal Processing*, 54(12):4685–4697, 2006.

[12] R. Figueras i Ventura, P. Vandergheynst, and P. Frossard. Low rate and flexible image coding with redundant representations. *IEEE Transactions on Image Processing*, 15(3):726–739, March 2006.

[13] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[14] S. Ellis and M. Nayakkankuppam. Phylogenetic analysis via DC programming (preprint). *Department of Mathematics and Statistics, UMBC, Baltimore, MD*, 2003.

[15] E. Kokiopoulou and P. Frossard. Minimum distance between pattern transformation manifolds: Algorithm and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1225–1238, July 2009.

[16] S. Sengupta, M. Harris, and M. Garland. Efficient parallel scan algorithms for GPUs. NVIDIA Technical Report NVR-2008-003, December 2008.

A. Proof of Theorem 1

We show in several steps that Theorem 1 is true. Using the fact that the geometric transformation of an atom ϕ_{η_k} is equivalent to a change in the coordinate system before applying $\phi(\cdot)$, we show that the transformed coordinate system $(\tilde{x}, \tilde{y}, \tilde{z})$ explicitly depends on the transformation parameters η. This is then used to show that $\tilde{x}(\eta_k)^2 + \tilde{y}(\eta_k)^2 + \tilde{z}(\eta_k)^2$ is a DC function of η, which in turn allows us to express the voxels of each atom ϕ_{η_k} in DC form. Based on the above developments, we finally obtain the DC decomposition of the objective function $\|v(\eta)-p\|_1$.

A.1. DC decomposition of transformed atoms

In what follows, we show that the transformed atom ϕ_{η_k} can be expressed in DC form. For notational convenience, we will drop the subscript k.

We first note that an atom in a parametric dictionary (1) is constructed by applying geometric transformations on the generating function ϕ. Applying a transformation $\gamma = (a, R, b)$ to the generating function is equivalent to transforming the coordinate system from $\{x, y, z\}$ to $\{\tilde{x}, \tilde{y}, \tilde{z}\}$ before applying $\phi(\cdot)$. More specifically, this means that an atom $\phi_\gamma = U(\gamma)\phi(x, y, z)$ coincides with $\phi(\tilde{x}, \tilde{y}, \tilde{z})$, where

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = AR^\top \begin{bmatrix} x - b_x \\ y - b_y \\ z - b_z \end{bmatrix}, \qquad (14)$$

and $A = \mathrm{diag}(1/a_x, 1/a_y, 1/a_z)$.

As we have already mentioned in Sec. 2.2, a transformation η applied to ϕ_γ results in a synthesis of the two transformations η and γ. Therefore, the transformed atom $\phi_{\eta\circ\gamma}$ can be readily constructed by applying the resulting transformation $\eta \circ \gamma$ directly to the mother function as shown in the paragraph above. One should make a clear distinction between γ, which denotes the (fixed) individual transformation for each atom and η, which is the global transformation applied to the entire volume (and hence to all atoms according to (5)). The transformed coordinate system of $\phi_{\eta\circ\gamma}$ therefore depends only on η (as γ is considered fixed).

In what follows, we derive the explicit dependence between the transformed coordinate system and the transformation parameters. For this purpose, we parametrize $\eta = (s, G, t)$ using quaternions for the rotation matrix G and let (q_0, q_1, q_2, q_3) denote the quaternion parameters. This results in eight optimization variables $(q_0, q_1, q_2, q_3, t_x, t_y, t_z, s)$ for representing the transformation η.

Lemma 1 *The transformed coordinates of an atom in (7) have the form*

$$\begin{aligned} \tilde{x}(\eta) \;=\; & \mu_0 \frac{q_0^2}{\sigma} + \mu_1 \frac{q_1^2}{\sigma} + \mu_2 \frac{q_2^2}{\sigma} + \mu_3 \frac{q_3^2}{\sigma} + \mu_4 \frac{q_1 q_2}{\sigma} \\ & + \mu_5 \frac{q_0 q_3}{\sigma} + \mu_6 \frac{q_0 q_2}{\sigma} + \mu_7 \frac{q_1 q_3}{\sigma} + \mu_8 \frac{q_2 q_3}{\sigma} \\ & + \mu_9 \frac{q_0 q_1}{\sigma} + \mu_{10} \frac{\tau_x}{\sigma} + \mu_{11} \frac{\tau_y}{\sigma} + \mu_{12} \frac{\tau_z}{\sigma} \\ & + \mu_{13}, \end{aligned} \qquad (15)$$

and similarly for $\tilde{y}$ and $\tilde{z}$ by replacing μ_i by ν_i and ζ_i, respectively. All μ_i, ν_i and ζ_i are constants depending only on the fixed atom parameters. In addition, σ as well as τ are related to s and t, respectively, by the following relations

$$\begin{aligned} \sigma &= N(q)s, \\ \tau &= \tilde{G}^\top t. \end{aligned}$$

Here, $N(q) = q_0^2 + q_1^2 + q_2^2 + q_3^2$ denotes the quaternion norm and $\tilde{G}$ denotes the (unnormalized) rotation matrix

$$\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_0q_1 + q_2q_3) \\ 2(q_0q_2 + q_1q_3) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}.$$

Proof. The proof is given in the Appendix B. □

With the change of variables suggested by the lemma above, the new optimization variables become (q_0, q_1, q_2, q_3, τ_x, τ_y, τ_z, σ). Note that we can always recover the original parameters t, s from τ, σ, and vice versa, using Lemma 1 (since the quaternion parameters are known). For notational convenience, we will assume in the following that this change of variables has been performed and continue to use η for denoting the (new) transformation parameters.

The next step in order to show that ϕ_η is DC, is to show that every constituent function in (15) is DC as well. In what follows, we provide a few lemmas towards this direction. In particular, we show that the following functions are DC: $f(q_i, \sigma) = \frac{q_i^2}{\sigma}$, $i = 0, 1, 2, 3$, $f(q_i, q_j, \sigma) = \frac{q_i q_j}{\sigma}$, $i, j = 0, 1, 2, 3$ and $i \neq j$, $f(\tau_x, \sigma) = \frac{\tau_x}{\sigma}$, $f(\tau_y, \sigma) = \frac{\tau_y}{\sigma}$ and $f(\tau_z, \sigma) = \frac{\tau_z}{\sigma}$.

Lemma 2 *The function $f(x, \alpha) = \frac{x}{\alpha} : \mathbb{R} \times \mathbb{R}_+^* \to \mathbb{R}$ is DC with the following DC decomposition*

$$f(x, \alpha) = \frac{x}{\alpha} = \frac{1}{2}\frac{(x+1)^2}{\alpha} - \frac{1}{2}\frac{(x^2+1)}{\alpha}. \qquad (16)$$

Proof. The proof can be found in [15, Lemma 3]. □

The above lemma implies that the constituent functions $\frac{\tau_x}{\sigma}$, $\frac{\tau_y}{\sigma}$ and $\frac{\tau_z}{\sigma}$ in (15) are DC.

Lemma 3 *The function $f(x, \alpha) = \frac{x^2}{\alpha} : \mathbb{R} \times \mathbb{R}_+^* \to \mathbb{R}$ is convex.*

Proof. The Hessian matrix of f is

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

Observe that the term $\frac{1}{\alpha^3}$ is positive, so we only need to prove that the remaining matrix is positive semi-definite. Call λ_1 and λ_2 its eigenvalues. Then observe that its determinant is $\lambda_1\lambda_2 = 4\alpha^2x^2 - 4x^2\alpha^2 = 0$. Thus either λ_1 or λ_2 are zero. Now, observe that the trace is $\lambda_1 + \lambda_2 = 2\alpha^2 + 2x^2 > 0$. Therefore, the Hessian matrix is positive semi-definite and f is convex.□

According to the above lemma, the constituent functions $\frac{q_i^2}{\sigma}, i = 0,1,2,3$ in (15) are DC.

Lemma 4 *The function* $f(x,y,\alpha) = \frac{xy}{\alpha} : \mathbb{R}\times\mathbb{R}\times\mathbb{R}_+^* \to \mathbb{R}$ *is DC, with the following decomposition*

$$\frac{xy}{\alpha} = \frac{1}{2}\frac{(x+y)^2}{\alpha} - \frac{1}{2}\frac{x^2+y^2}{\alpha}. \qquad (17)$$

Proof. The proof is given in the Appendix C. □

The above lemma implies that the functions $\frac{q_iq_j}{\sigma}$, where $i,j = 0,1,2,3$ and $i \neq j$ in (15) are DC.

To summarize, we have shown the DC property of *all* constituent functions in (15). We can therefore write (15) in the more abstract form

$$\tilde{x}(\eta) = \sum_{i=0}^{13} \mu_i f_i = \sum_{i=0}^{13} \mu_i (g_i - h_i), \qquad (18)$$

where $g_i - h_i$ is the DC decomposition of each function f_i. Moreover, note that each convex part g_i, h_i is nonnegative. This allows us to conclude that the transformed coordinates are DC.

Lemma 5 *The functions $\tilde{x}^2(\eta)$ (and similarly $\tilde{y}^2(\eta)$ and $\tilde{z}^2(\eta)$) introduced in Lemma 1 are DC functions of η.*

Proof. From (18) we have that $\tilde{x}^2(\eta) = \sum_{\substack{i,j=0\\ i\neq j}}^{13} 2\mu_i\mu_j f_i f_j + \sum_{i=0}^{13} \mu_i^2 f_i^2$. Proposition 1 (c) states that the product of two DC functions (with nonnegative convex parts) is also DC. Using the results developed above, this implies that all summands in $\tilde{x}^2(\eta)$ are DC. Since the linear combination of DC functions is again DC by Proposition 1 (a) we have finally obtained that $\tilde{x}^2(\eta)$ is DC. □

Lemma 5 implies that $w(\eta) = \tilde{x}(\eta)^2 + \tilde{y}(\eta)^2 + \tilde{z}(\eta)^2$ is DC and we denote its DC decomposition by $w(\eta) = g_w(\eta) - h_w(\eta)$.

A.2. DC form of the objective function

Finally we are ready to prove the main result of our paper, namely Theorem 1, which states that the objective function of the optimization problem (6) is DC. Recall that the construction of geometric atoms by transforming the generating function is equivalent to considering the generating function on the transformed coordinates $\tilde{x}, \tilde{y}$ and $\tilde{z}$ computed above. Given these developments, it remains to show that the transformed generating functions are DC, and that the ℓ^1 distance between the transformed volume $v(\eta)$ and the query volume p is DC. We prove this for the Gaussian generating function i.e., $\phi(x,y,z) = \exp(-(x^2+y^2+z^2))$. Note that the atoms ϕ_γ are not normalized; the L^2 norm of ϕ_γ will be denoted by $\|\phi_\gamma\|$.

Proof of Theorem 1.

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

where we have introduced the function

$$\begin{aligned}\delta(\eta) &= w(\eta) + \ln s + \ln\|\phi_\gamma\| \\ &= g_w(\eta) - h_w(\eta) + \ln s + \ln\|\phi_\gamma\|. \qquad (19)\end{aligned}$$

Recall from Lemma 1 that $s = \frac{\sigma}{N(q)}$, where $N(q) = q_0^2 + q_1^2 + q_2^2 + q_3^2$, which is rewritten as

$$\ln s = \ln\sigma - \ln N(q). \qquad (20)$$

Note that $\ln\sigma$ is concave. Unfortunately, $\ln N(q)$ is *not* a convex function in the quaternion parameters, and we therefore need a DC decomposition for $\ln N(q)$. We show in the Appendix D that

$$\begin{aligned}\ln N(q) &= \left[\ln(N(q)) + \sum_{i=0}^{3}\ln(q_i^2)\right] - \left[\sum_{i=0}^{3}\ln(q_i^2)\right] \\ &= g_{Nq}(\eta) - h_{Nq}(\eta)\end{aligned}$$

is a decomposition of $\ln N(q)$, where both components are concave. Inserting this decomposition into (20) yields $\ln s = \ln\sigma - g_{Nq}(\eta) + h_{Nq}(\eta)$. Putting all facts together, we can rewrite (19) as

$$\begin{aligned}\delta(\eta) &= [g_w(\eta) - g_{Nq}(\eta) + \ln\|\phi_\gamma\|] \\ &\quad - [h_w(\eta) - \ln\sigma - h_{Nq}(\eta)], \qquad (21)\end{aligned}$$

which readily provides a DC decomposition for $\delta(\eta)$.

Next, we make use of Proposition 2 (b), which states that the synthesis of a convex function with a DC function is again DC. This shows that every voxel of ϕ_η is DC with the following decomposition: $e^{-\delta(\eta)} = [e^{-\delta(\eta)} + K(g_\delta(\eta) + h_\delta(\eta))] - [K(g_\delta(\eta) + h_\delta(\eta))]$. This holds for each atom in the sparse approximation of the volume v.

Consider now the kth atom and let $\phi_{\eta_k} = g_k(\eta) - h_k(\eta)$ denote the DC decomposition of its voxels. Next, we use once again Proposition 1 (a) to come up with the DC decomposition of $v(\eta) = \sum_{k=1}^{K} \xi_k \phi_{\eta_k}$, which reads $v(\eta) = \left[\sum_{\{k:\xi_k \geq 0\}} \xi_k g_k - \sum_{\{k:\xi_k<0\}} \xi_k h_k\right] - \left[\sum_{\{k:\xi_k \geq 0\}} \xi_k h_k - \sum_{\{k:\xi_k<0\}} \xi_k g_k\right] \equiv g_v(\eta) - h_v(\eta)$.

So far, we have shown that the transformed reference volume $v(\eta)$ is a DC decomposition of η. Since, the query volume p is fixed, the same holds for the difference volume $v(\eta) - p$. Proposition 1 (b) permits to compute the DC decomposition of the ith voxel of $|v(\eta)-p|$, which is given by $|v(\eta)-p|_i = 2\max\{g_i, h_i\} - (g_i+h_i)$, where $g_i - h_i$ is the DC decomposition of the ith voxel of $v(\eta) - p$. Finally, the objective function in (7) is DC, as it is a sum over the voxels of $|v(\eta) - p|$, which have been shown to be DC functions. □

Finally, we note that the proof above can be extended from the Gaussian generating function to other generating functions (similarly to the proof in [15]). We conclude that the objective function is a DC function, which permits the application of DC programming methods for computing the global minimizer of the optimization problem (6).

B. Proof of Lemma 1

Suppose that the atom under consideration has parameters $\gamma = (a, R, b)$, where $a = [a_x, a_y, a_z]$ and $b = [b_x, b_y, b_z]$. If we denote by $\eta = (s, G, t)$ the transformation, then according to the SIM(3) group law (4), the transformed parameters of the atom will be

$$\eta \circ \gamma = (sa, GR, t + sGb).$$

If we denote $A = \mathrm{diag}(1/a_x, 1/a_y, 1/a_z)$, then the transformed axes $[\tilde{x}, \tilde{y}, \tilde{z}]^\top$ according to (14) will be

$$\begin{aligned} \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} &= \tfrac{1}{s} A R^\top G^\top \left[\begin{pmatrix} x \\ y \\ z \end{pmatrix} - t - sGb \right] \\ &= AR^\top \left[\tfrac{G^\top}{s} \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \tfrac{G^\top}{s} t - b \right]. \quad (22) \end{aligned}$$

In the above equation we have used the fact that G is a rotation matrix i.e., $G^\top G = I$.

We use quaternions to parametrize the unknown rotation matrix G. Consider a quaternion $q = q_0 + iq_1 + jq_2 + kq_3$ with norm $N(q) = q_0^2 + q_1^2 + q_2^2 + q_3^2$. Then the corresponding rotation matrix takes the form

$$G = \frac{1}{N(q)} \tilde{G},$$

where

$$\tilde{G} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_0q_1 + q_2q_3) \\ 2(q_0q_2 + q_1q_3) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (23)$$

denotes the unnormalized rotation matrix. Inserting this representation into (22) gives

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = AR^\top \left[\tfrac{\tilde{G}^\top}{N(q)s} \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \tfrac{\tilde{G}^\top}{N(q)s} t - b \right]. \quad (24)$$

Defining

$$\begin{aligned} \sigma &= N(q)s, & (25) \\ \tau &= \tilde{G}^\top t, & (26) \end{aligned}$$

the original set of optimization variables $(q_0, q_1, q_2, q_3, t_x, t_y, t_z, s)$ becomes $(q_0, q_1, q_2, q_3, \tau_x, \tau_y, \tau_z, \sigma)$. Note that these two variable representations are equivalent and one may switch from the first one to the second and vice versa via the use of equations (25) and (26). In what follows we use the second representation and rewrite (24) as

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{pmatrix} = AR^\top \left[\tfrac{1}{\sigma} \tilde{G}^\top \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \tfrac{1}{\sigma} \tau - b \right]. \quad (27)$$

Recall that in the above, A, R, b, x, y and z are constant. In particular, the matrix $AR^\top$ is constant and its entries are denoted as follows:

$$AR^\top = \begin{bmatrix} \rho_1 & \rho_2 & \rho_3 \\ \rho_4 & \rho_5 & \rho_6 \\ \rho_7 & \rho_8 & \rho_9 \end{bmatrix}.$$

The right hand side of (27) thus takes the form

$$\begin{bmatrix} \rho_1 & \rho_2 & \rho_3 \\ \rho_4 & \rho_5 & \rho_6 \\ \rho_7 & \rho_8 & \rho_9 \end{bmatrix} \frac{1}{\sigma} \left[\tilde{G}^\top \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \tau \right] - \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix}, \quad (28)$$

where $[c_x, c_y, c_z]^\top = AR^\top b$.

Next, we will use (23) to compute the explicit dependence of $\tilde{x}$ on the optimization variables. Note that

$$\tilde{G}^\top \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} x(q_0^2 + q_1^2 - q_2^2 - q_3^2) + 2y(q_1q_2 - q_0q_3) + 2z(q_0q_2 + q_1q_3) \\ 2x(q_1q_2 + q_0q_3) + y(q_0^2 - q_1^2 + q_2^2 - q_3^2) + 2z(q_2q_3 - q_0q_1) \\ 2x(q_1q_3 - q_0q_2) + 2y(q_0q_1 + q_2q_3) + z(q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix}.$$

Putting all the above facts together one finally obtains, after some straightforward algebraic manipulation,

$$\begin{aligned} \tilde{x} = \; & \mu_0 \frac{q_0^2}{\sigma} + \mu_1 \frac{q_1^2}{\sigma} + \mu_2 \frac{q_2^2}{\sigma} + \mu_3 \frac{q_3^2}{\sigma} + \mu_4 \frac{q_1q_2}{\sigma} + \mu_5 \frac{q_0q_3}{\sigma} \\ & + \mu_6 \frac{q_0q_2}{\sigma} + \mu_7 \frac{q_1q_3}{\sigma} + \mu_8 \frac{q_2q_3}{\sigma} + \mu_9 \frac{q_0q_1}{\sigma} + \mu_{10} \frac{\tau_x}{\sigma} \\ & + \mu_{11} \frac{\tau_y}{\sigma} + \mu_{12} \frac{\tau_z}{\sigma} + \mu_{13}, \quad (29) \end{aligned}$$

where

$$
\begin{aligned}
\mu_0 &= \rho_1 x + \rho_2 y + \rho_3 z \\
\mu_1 &= \rho_1 x - \rho_2 y - \rho_3 z \\
\mu_2 &= -\rho_1 x + \rho_2 y - \rho_3 z \\
\mu_3 &= -\rho_1 x - \rho_2 y + \rho_3 z \\
\mu_4 &= 2\rho_1 y + 2\rho_2 x \\
\mu_5 &= -2\rho_1 y + 2\rho_2 x \\
\mu_6 &= 2\rho_1 z - 2\rho_3 x \\
\mu_7 &= 2\rho_1 z + 2\rho_3 x \\
\mu_8 &= 2\rho_2 z + 2\rho_3 y \\
\mu_9 &= -2\rho_2 z + 2\rho_3 y \\
\mu_{10} &= -\rho_1 \\
\mu_{11} &= -\rho_2 \\
\mu_{12} &= -\rho_3 \\
\mu_{13} &= -c_x.
\end{aligned}
$$

Observe that all μ_i are constant. This concludes the proof for $\tilde{x}$. The derivation for $\tilde{y}$ and $\tilde{z}$ is similar and therefore omitted. □

C. Proof of Lemma 4

We need to show that the two components in (17) are convex. We start with the function $f(x, y, a) = \frac{(x+y)^2}{a}$, $a > 0$. The Hessian matrix $\nabla^2 f(x, y, a)$ is

$$
a^3 \begin{bmatrix} 2a^2 & 2a^2 & -2a(x+y) \\ 2a^2 & 2a^2 & -2a(x+y) \\ -2a(x+y) & -2a(x+y) & (x+y)^2 \end{bmatrix}.
$$

Now consider a vector $v = [v_1, v_2, v_3]^\top$ and observe that

$$
\begin{aligned}
v^\top \nabla^2 f(x, y, a) v &= 2a^2 v_1^2 + 2a^2 v_2^2 + 2(x+y)^2 v_3^2 \\
&\quad + 4a^2 v_1 v_2 - 4a(x+y) v_1 v_3 \\
&\quad - 4a(x+y) v_2 v_3 \\
&= 2(a v_1 + a v_2 - (x+y) v_3)^2 \geq 0.
\end{aligned}
$$

Hence the first component is convex.

Considering now the second component $f(x, y, a) = \frac{x^2+y^2}{a}$, $a > 0$, the Hessian matrix is

$$
\nabla^2 f(x, y, a) = a^3 \begin{bmatrix} 2a^2 & 0 & -2xa \\ 0 & 2a^2 & -2ya \\ -2xa & -2ya & 2(x^2+y^2) \end{bmatrix}.
$$

Similarly as above, consider a vector $v = [v_1, v_2, v_3]^\top$ and observe that

$$
\begin{aligned}
v^\top \nabla^2 f(x, y, a) v &= 2[a^2 v_1^2 + a^2 v_2^2 \\
&\quad + (x^2 + y^2) v_3^2 - 2xa v_1 v_3 \\
&\quad - 2ya v_2 v_3] \\
&= 2(a v_1 - x v_3)^2 + 2(a v_2 - y v_3)^2 \geq 0,
\end{aligned}
$$

which shows that the second part is also convex. □

D. Decomposition of $\ln(x^2 + y^2 + z^2 + w^2)$

Let $N = x^2 + y^2 + z^2 + w^2$. We will show that the following decomposition

$$
\begin{aligned}
\ln N = &\left[\ln(N) + \ln(x^2) + \ln(y^2) + \ln(z^2) + \ln(w^2)\right] - \\
&\left[\ln(x^2) + \ln(y^2) + \ln(z^2) + \ln(w^2)\right],
\end{aligned} \tag{30}
$$

has concave components.

Proof. The second component is concave as it consists of a sum of concave scalar functions. We now focus on the first part, whose Hessian matrix is

$$
H = \frac{1}{N^2} \begin{bmatrix} \zeta(x) & -4xy & -4xz & -4xw \\ -4xy & \zeta(y) & -4yz & -4yw \\ -4xz & -4yz & \zeta(z) & -4zw \\ -4xw & -4yw & -4zw & \zeta(w) \end{bmatrix},
$$

where we have introduced the function $\zeta(x) = \frac{2x^2N - 2N^2 - 4x^4}{x^2}$ for notational convenience. Consider a vector $v = [v_1, v_2, v_3, v_4]^\top$. After factoring out the term $\frac{1}{x^2y^2z^2w^2}$ and some algebraic manipulation, it holds that

$$
\begin{aligned}
v^\top H v = &-2v_1^2 w^6 y^2 z^2 - 2v_1^2 w^4 x^2 y^2 z^2 - 4v_1^2 w^4 y^4 z^2 \\
&-4v_1^2 w^4 y^2 z^4 - 4v_1^2 w^2 x^4 y^2 z^2 - 2v_1^2 w^2 x^2 y^4 z^2 \\
&-2v_1^2 w^2 x^2 y^2 z^4 - 2v_1^2 w^2 y^6 z^2 - 4v_1^2 w^2 y^4 z^4 \\
&-2v_1^2 w^2 y^2 z^6 - 8v_1 v_2 w^2 x^3 y^3 z^2 - 8v_1 v_3 w^2 x^3 y^2 z^3 \\
&-8v_1 v_4 w^3 x^3 y^2 z^2 - 2v_2^2 w^6 x^2 z^2 - 4v_2^2 w^4 x^4 z^2 \\
&-2v_2^2 w^4 x^2 y^2 z^2 - 4v_2^2 w^4 x^2 z^4 - 2v_2^2 w^2 x^6 z^2 \\
&-2v_2^2 w^2 x^4 y^2 z^2 - 4v_2^2 w^2 x^4 z^4 - 4v_2^2 w^2 x^2 y^4 z^2 \\
&-2v_2^2 w^2 x^2 y^2 z^4 - 2v_2^2 w^2 x^2 z^6 - 8v_2 v_3 w^2 x^2 y^3 z^3 \\
&-8v_2 v_4 w^3 x^2 y^3 z^2 - 2v_3^2 w^6 x^2 y^2 - 4v_3^2 w^4 x^4 y^2 \\
&-4v_3^2 w^4 x^2 y^4 - 2v_3^2 w^4 x^2 y^2 z^2 - 2v_3^2 w^2 x^6 y^2 \\
&-4v_3^2 w^2 x^4 y^4 - 2v_3^2 w^2 x^4 y^2 z^2 - 2v_3^2 w^2 x^2 y^6 \\
&-2v_3^2 w^2 x^2 y^4 z^2 - 4v_3^2 w^2 x^2 y^2 z^4 - 8v_3 v_4 w^3 x^2 y^2 z^3 \\
&-4v_4^2 w^4 x^2 y^2 z^2 - 2v_4^2 w^2 x^4 y^2 z^2 - 2v_4^2 w^2 x^2 y^4 z^2 \\
&-2v_4^2 w^2 x^2 y^2 z^4 - 2v_4^2 x^6 y^2 z^2 - 4v_4^2 x^4 y^4 z^2 \\
&-4v_4^2 x^4 y^2 z^4 - 2v_4^2 x^2 y^6 z^2 - 4v_4^2 x^2 y^4 z^4 \\
&-2v_4^2 x^2 y^2 z^6,
\end{aligned} \tag{31}
$$

which can be re-written as

$$\begin{aligned} v^\top Hv \quad = \quad & -2w^2z^2(v_1y^3+v_2x^3)^2 - 2x^2y^2z^2w^2(v_1y+v_2x)^2 \\ & -2w^2y^2(v_1z^3+v_3x^3)^2 - 2x^2y^2z^2w^2(v_1z+v_3x)^2 \\ & -2y^2z^2(v_1w^3+v_4x^3)^2 - 2x^2y^2z^2w^2(v_1w+v_4x)^2 \\ & -2w^2x^2(v_2z^3+v_3y^3)^2 - 2x^2y^2z^2w^2(v_2z+v_3y)^2 \\ & -2x^2z^2(v_2w^3+v_4y^3)^2 - 2x^2y^2z^2w^2(v_2w+v_4y)^2 \\ & -2x^2y^2(v_3w^3+v_4z^3)^2 - 2x^2y^2z^2w^2(v_3w+v_4z)^2 \\ & -4v_1^2w^4y^4z^2 - 4v_1^2w^4y^2z^4 - 4v_1^2w^2x^4y^2z^2 \\ & -4v_1^2w^2y^4z^4 - 4v_2^2w^4x^4z^2 - 4v_2^2w^4x^2z^4 \\ & -4v_2^2w^2x^4z^4 - 4v_2^2w^2x^2y^4z^2 - 4v_3^2w^4x^4y^2 \\ & -4v_3^2w^4x^2y^4 - 4v_3^2w^2x^4y^4 - 4v_3^2w^2x^2y^2z^4 \\ & -4v_4^2w^4x^2y^2z^2 - 4v_4^2x^4y^4z^2 - 4v_4^2x^4y^2z^4 \\ & -4v_4^2x^2y^4z^4 \le 0. \qquad (32) \end{aligned}$$

Hence, the first part of the decomposition is concave. □

Research Reports

No.	Authors/Title
11-18	*E. Kokiopoulou, D. Kressner, M. Zervos and N. Paragios* Optimal similarity registration of volumetric images
11-17	*D. Marazzina, O. Reichmann and Ch. Schwab* *hp*-DGFEM for Kolmogorov-Fokker-Planck equations of multivariate Lévy processes
11-16	*Ch. Schwab and A.M. Stuart* Sparse deterministic approximation of Bayesian inverse problems
11-15	*A. Barth and A. Lang* Almost sure convergence of a Galerkin–Milstein approximation for stochastic partial differential equations
11-14	*X. Claeys* A single trace integral formulation of the second kind for acoustic scattering
11-13	*W.-J. Beyn, C. Effenberger and D. Kressner* Continuation of eigenvalues and invariant pairs for parameterized nonlinear eigenvalue problems
11-12	*C.J. Gittelson* Adaptive stochastic Galerkin methods: beyond the elliptic case
11-11	*C.J. Gittelson* An adaptive stochastic Galerkin method
11-10	*C.J. Gittelson* Stochastic Galerkin approximation of operator equations with infinite dimensional noise
11-09	*R. Hiptmair, A. Moiola and I. Perugia* Error analysis of Trefftz-discontinuous Galerkin methods for the time-harmonic Maxwell equations
11-08	*W. Dahmen, C. Huang, Ch. Schwab and G. Welper* Adaptive Petrov-Galerkin methods for first order transport equations
11-07	*V.H. Hoang and Ch. Schwab* Analytic regularity and polynomial approximation of stochastic, parametric elliptic multiscale PDEs
11-06	*G.M. Coclite, K.H. Karlsen. S. Mishra and N.H. Risebro* A hyperbolic-elliptic model of two-phase flow in porous media - Existence of entropy solutions