This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Globally Optimal Rigid Intensity Based Registration: A Fast Fourier Domain Approach

Behrooz Nasihatkon[1,3], Frida Fejne[1,3], and Fredrik Kahl[1,2,3]

[1]Department of Signals and Systems, Chalmers University of Technology, Sweden
[2]Centre for Mathematical Sciences, Lund University, Sweden
[3]MedTech West, Sweden
{behrooz.nasihatkon,frida.fejne,fredrik.kahl}@chalmers.se

Abstract

High computational cost is the main obstacle to adapting globally optimal branch-and-bound algorithms to intensity-based registration. Existing techniques to speed up such algorithms use a multiresolution pyramid of images and bounds on the target function among different resolutions for rigidly aligning two images. In this paper, we propose a dual *algorithm in which the optimization is done in the Fourier domain, and multiple resolution levels are replaced by multiple frequency bands. The algorithm starts by computing the target function in lower frequency bands and keeps adding higher frequency bands until the current subregion is either rejected or divided into smaller areas in a branch and bound manner. Unlike spatial multiresolution approaches, to compute the target function for a wider frequency area, one just needs to compute the target in the residual bands. Therefore, if an area is to be discarded, it performs just enough computations required for the rejection. This property also enables us to use a rather large number of frequency bands compared to the limited number of resolution levels used in the space domain algorithm. Experimental results on real images demonstrate considerable speed gains over the space domain method in most cases.*

1. Introduction

This paper proposes a new method of exhaustively searching the rigid motion space for intensity-based image registration. Here, the registration target function is computed in the frequency domain using the DFT coefficients of the images. The core idea is that many of the non-optimal areas of the parameter space can be rejected by just computing the target function in low-frequency bands. The method is inspired by the multiresolution approach of Nasihatkon and Kahl [11] in the spatial domain, but offers significant advantages by performing the registration in the frequency domain.

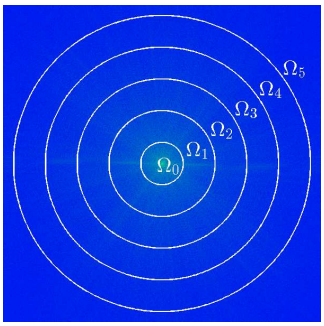


Figure 1. Partitioning the frequency domain into radial bands $\Omega_0, \Omega_1, \ldots, \Omega_5$. This forms nested frequency balls $\mathcal{B}_i = \cup_{j=0}^{i} \Omega_j$. The target function can be decomposed as the sum of target functions over each frequency band.

The main obstacle in applying global optimization techniques to intensity-based registration is the computational cost of calculating a similarity or dissimilarity measure between two images. That is why most of the globally optimal registration techniques are about the alignment of point sets [4, 14, 15, 8, 19], making them only applicable to feature-based registration. Reducing the computational cost of the exhaustive search for intensity-based alignment has been well studied in the context of pattern matching, where a small patch is searched against a large target image. Such approaches are quite diverse, but all are based on the same idea of *early rejection*: non-optimal solutions might be rejected without computing the target func-

tion by first examining a low-cost bound. One of the earliest methods of this type is the *successive elimination* approach of Li and Salari [9]. The succeeding approaches are mostly based on computing a sequence of bounds, where each bound is tighter compared to the previous one, but requires more computations [7, 3, 10, 17, 12, 13].

All the above approaches are restricted to searching over a grid of translations. For more general transformations, one can mention the work of Korman et al. [5] about affine pattern matching. The method is asserted to find the optimal solution with *high probability*. One of the disadvantages of this approach is that it only provides asymptotic forms for the bounds. The bounds used in practice for the branch and bound algorithm are *learned* experimentally from a database of images. Aside from pattern matching, little work has been done on global optimization for intensity-based image alignment. Cremers et al. [1] use Lipschitz optimization for the alignment of shape models. The optimization also involves estimating the rigid pose of the shapes. However, no effort has been made to reduce the computational cost of the algorithm.

Recently, Nasihatkon and Kahl proposed a multiresolution approach for reducing the cost of the exhaustive search in rigid intensity-based registration [11]. They use a multiresolution pyramid of images and show that many areas of the search space can be discarded by computing the target function in lower resolutions. The method is incorporated into a Lipschitz optimization framework and demonstrates a significant speed-up over the corresponding single resolution algorithm.

Here, we propose a similar approach in which the registration is performed in the frequency domain. Many of the existing frequency domain approaches are based on the Fourier shift property, i.e., the fact that a spatial translation of two images corresponds to a phase shift difference in the Fourier domain. A lot of them are extensions of the well-known phase correlation technique [6] which computes the cross-power spectrum and then finds the integer translation by locating the peak in the inverse spectrum. Furthermore, there exists a multitude of other approaches that extends [6] and includes additional transformations such as rotation and scaling. For example, Reddy et al. in [16] exploit the Fourier scaling and rotational properties in order to find the parameters of the transformation at a low computational cost. By applying a log-polar transformation of the Fourier magnitude spectra of the two images, the scale and rotation can be expressed as a phase shift, thus enabling the use of the phase correlation method. To decouple rotation and scale from translation they assume that the magnitude of the Fourier transforms is not a function of the translation parameter. This trick, however, is only valid when the images are exact transformed replica of each other. Therefore, the approach is not globally optimal if the images are different.

In our algorithm the frequency domain is divided into *radial bands* $\Omega_0, \Omega_1, \ldots, \Omega_P$ as shown in Fig. 1. The bands form nested balls $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_P$ where $\mathcal{B}_i = \cup_{j=0}^{i} \Omega_j$. Our approach can be seen as a dual to spatial multiresolution techniques in which the resolution levels are replaced by the frequency balls $\mathcal{B}_i$. For each candidate solution we keep moving from lower to higher frequency balls. For each ball we compute the target function and examine a rejection criterion. If the rejection criterion is met the candidate solution is discarded, otherwise, we move to the next frequency ball. Therefore, if a solution is to be rejected, we perform just enough computations to find it out. To sum up, the advantages of our approach are as follows

- For example, in [11], if the rejection test fails at a certain resolution, the target function is computed *from scratch* for the next resolution. However, here, having the target function for a ball $\mathcal{B}_i$, to get the target for $\mathcal{B}_{i+1}$ one just needs to compute the target within the residual band Ω_{i+1}.

- Also, due to the above, dividing the frequency domain into more frequency bands does not introduce considerable extra computations. Hence, the number of frequency bands can be much higher than the number of resolution levels.

Also, here, we do not need to deal with the errors induced by downsampling to create the multiresolution pyramid, such as the errors of approximating the ideal radial low-pass filter, or not sampling the filtered image outside the image boundaries [11]. However, using the DFT coefficients to approximate the image spectrum leads to space-domain aliasing. Therefore, measures like zero-padding or inverting the intensity values of the images might be necessary in both space domain and frequency domain algorithms.

We describe the basics of multiband method in Sect. 3 after providing a background in Sect. 2. In Sect. 4 we present a technique of integrating the multiband method into the Lipschitz global optimization framework. Experiments are conducted in Sect. 5.

2. Background

In this section, we describe our analysis framework. The continuous domain image pairs are represented as functions $f, g\colon \mathbb{R}^d \to \mathbb{R}$, where $d = 2$ or $d = 3$. The

target function is chosen to be the correlation between f and the transformed g:

$$Q(\mathtt{R},\mathbf{t}) = \int_{\mathbb{R}^d} f(\mathbf{x})\, g(\mathtt{R}(\mathbf{x}+\mathbf{t}))\, d\mathbf{x}, \tag{1}$$

where $\mathtt{R}$ and $\mathbf{t}$ denote the rotation matrix and the translation vector respectively. To perform registration we need to maximize (1) with respect to $\mathtt{R}$ and $\mathbf{t}$. Let $F, G\colon \mathbb{R}^d \to \mathbb{C}$ be the Fourier transforms of f and g respectively. Using the Parseval's theorem, (1) can be reformulated in the frequency domain as

$$Q(\mathtt{R},\mathbf{t}) = \int_{\mathbb{R}^d} \overline{F(\mathbf{z})}\, G(\mathtt{R}\mathbf{z})\, e^{2\pi j \mathbf{t}^T \mathbf{z}}\, d\mathbf{z}, \tag{2}$$

where $\overline{F(\mathbf{z})}$ represents the complex conjugate of $F(\mathbf{z})$ and $j = \sqrt{-1}$.

Discrete images are represented by a set of samples $\{f_{\mathbf{i}}\}$ at discrete locations $\mathbf{i} \in \mathbb{Z}^d$. For real images, the samples are usually zero outside a rectangular or cubical grid $\mathcal{P}$. In this paper for a two-dimensional $m{\times}n$ image, the grid is $\mathcal{P} = \{0,1,\ldots,m-1\} \times \{0,1,\ldots,n-1\}$ where $\times$ represents the Cartesian product. The grid is defined similarly for 3D images. Any discrete image corresponds to a *canonical* continuous image using *sinc* interpolation:

$$f(\mathbf{x}) = \sum_{\mathbf{i}\in\mathbb{Z}^d} f_{\mathbf{i}}\, \mathrm{sinc}(\mathbf{x}-\mathbf{i}), \tag{3}$$

where $\mathrm{sinc}(\mathbf{x}) = \prod_{i=1}^d \frac{\sin(\pi x_i)}{\pi x_i}$. Notice that $F(\mathbf{z})$, the Fourier transform of $f(\mathbf{x})$, is bandlimited to $(-\frac{1}{2},\frac{1}{2})$ along each dimension, that is, $F(\mathbf{z}) = 0$ for $\|\mathbf{z}\|_\infty > \frac{1}{2}$.

For each $m{\times}n$ 2D discrete image $\{f_{\mathbf{i}}\}$ we can define an $M{\times}N$ Discrete Fourier Transform (DFT) $\{F_{\mathbf{k}}\}$ on a grid $\mathcal{Q} = \{-\lfloor M/2\rfloor{+}1,\ldots,\lfloor M/2\rfloor\} \times \{-\lfloor N/2\rfloor{+}1,\ldots,\lfloor N/2\rfloor\}$ as

$$F_{\mathbf{k}} = \sum_{\mathbf{i}\in\mathcal{P}} f_{\mathbf{i}}\, e^{2\pi j\,(i_1 k_1/M + i_2 k_2/N)}, \tag{4}$$

for all $\mathbf{k} = (k_1,k_2) \in \mathcal{Q}$. Notice that $\mathbf{0} = (0,0) \in \mathcal{Q}$. It is easy to verify that for a discrete image $\{f_{\mathbf{i}}\}_{\mathbf{i}\in\mathcal{P}}$, with $\{F_{\mathbf{k}}\}$ defined on an $M{\times}N$ grid $\mathcal{Q}$ we have

$$F_{\mathbf{k}} = F(k_1/M, k_2/N), \tag{5}$$

where $F(\mathbf{z})$ is the Fourier transform of $f(\mathbf{x})$ defined in (3). By choosing larger M and N, and thus increasing the DFT resolution, one can have the values of $F(\mathbf{z})$ on a finer grid. Similar formulas hold for 3D images.

3. The multiband method

The frequency domain is partitioned into radial frequency bands $\Omega_0, \Omega_1, \ldots, \Omega_P$, where $\Omega_i = \{\mathbf{z} \,|\, r_i \le \|\mathbf{z}\| < r_{i+1}\}$ and $0 = r_0 < r_1 < \cdots < r_{P+1}$. This is illustrated in Fig. 1. Now, the target function (2) can be decomposed as a sum of separate target functions over the different frequency bands:

$$Q(\mathtt{R},\mathbf{t}) = \sum_{i=1}^{P} Q^{\Omega_i}(\mathtt{R},\mathbf{t}), \tag{6}$$

where

$$Q^{\Omega_i}(\mathtt{R},\mathbf{t}) = \int_{\Omega_i} \overline{F(\mathbf{z})}\, G(\mathtt{R}\mathbf{z})\, e^{2\pi j \mathbf{t}^T \mathbf{z}}\, d\mathbf{z}. \tag{7}$$

Using the Cauchy-Schwarz inequality we get

$$\begin{aligned} Q^{\Omega_i}(\mathtt{R},\mathbf{t}) &\le \sqrt{\textstyle\int_{\Omega_i} |F(\mathbf{z})|^2 d\mathbf{z}}\sqrt{\textstyle\int_{\Omega_i} |G(\mathtt{R}\mathbf{z})|^2 d\mathbf{z}} \\ &= \sqrt{\textstyle\int_{\Omega_i} |F(\mathbf{z})|^2 d\mathbf{z}}\sqrt{\textstyle\int_{\Omega_i} |G(\mathbf{z})|^2 d\mathbf{z}} \\ &\stackrel{\text{def}}{=} B_{\text{res}}^{\Omega_i}. \end{aligned} \tag{8}$$

Here, we call $B_{\text{res}}^{\Omega_i}$ the *residual bound*, which means the residual error if the target function is not computed at the frequency band Ω_i. Furthermore, we define frequency balls $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_P$ as $\mathcal{B}_p{=}\cup_{i=0}^p \Omega_i = \{\mathbf{z} \,|\, \|\mathbf{z}\| < r_{p+1}\}$. We define a target function for each ball $\mathcal{B}_p$ as $Q^{\mathcal{B}_p}(\mathtt{R},\mathbf{t}) = \sum_{i=0}^p Q^{\Omega_i}(\mathtt{R},\mathbf{t})$. The residual bound for $\mathcal{B}_p$ is defined as the sum of the residual bounds for the frequency bands *outside* $\mathcal{B}_p$, that is,

$$B_{\text{res}}^{\mathcal{B}_p} = \sum_{i=p+1}^{P} B_{\text{res}}^{\Omega_i}. \tag{9}$$

Notice that, as the quantities $B_{\text{res}}^{\Omega_i}$ are nonnegative, B_{res}^{p} is descending with respect to p:

$$B_{\text{res}}^{\mathcal{B}_0} \ge B_{\text{res}}^{\mathcal{B}_1} \ge \ldots \ge B_{\text{res}}^{\mathcal{B}_P} = 0. \tag{10}$$

Now, assume that we have computed the target function for frequency bands $\Omega_0, \Omega_1, \ldots, \Omega_p$, for any $0 \le p \le P$. Then (6) and (8) give

$$\begin{aligned} Q(\mathtt{R},\mathbf{t}) &\le \sum_{i=0}^{p} Q^{\Omega_i}(\mathtt{R},\mathbf{t}) + \sum_{i=p+1}^{P} B_{\text{res}}^{\Omega_i} \\ &= Q^{\mathcal{B}_p}(\mathtt{R},\mathbf{t}) + B_{\text{res}}^{\mathcal{B}_p}. \end{aligned} \tag{11}$$

Thus, as p increases we have tighter and tighter bounds $B_{\text{res}}^{\mathcal{B}_p}$, and, $Q^{\mathcal{B}_p}(\mathtt{R},\mathbf{t})$ becomes a better approximation of $Q(\mathtt{R},\mathbf{t})$. Finally at $p = P$ we have $B_{\text{res}}^{\mathcal{B}_P} = 0$ and $Q^{\mathcal{B}_P}(\mathtt{R},\mathbf{t}) = Q(\mathtt{R},\mathbf{t})$.

Now, consider a search scheme where $Q(\mathtt{R},\mathbf{t})$ is to be maximized over a discrete grid of parameters. Consider a candidate solution $(\mathtt{R},\mathbf{t})$ in the grid and the best solution Q^* found so far. We keep computing $Q^{\mathcal{B}_p}(\mathtt{R},\mathbf{t})$

for $p = 0, 1, \ldots, P$ in order. If $Q^{\mathcal{B}_p}(\mathtt{R}, \mathbf{t}) + B^{\mathcal{B}_p}_{\text{res}} < Q^*$ for some p, then $(\mathtt{R}, \mathbf{t})$ cannot be an optimal solution and is discarded. If the solution does not get rejected at any $p = 0, 1, \ldots, P$, then $Q^{\mathcal{B}_P}(\mathtt{R}, \mathbf{t}) = Q(\mathtt{R}, \mathbf{t}) \geq Q^*$. In this case the best solution Q^* gets updated. Notice that if the rejection criterion fails at p, for computing the target at $p{+}1$ we just need to compute the target inside the frequency band Ω_{p+1}, as $Q^{\mathcal{B}_{p+1}}(\mathtt{R}, \mathbf{t}) = Q^{\mathcal{B}_p}(\mathtt{R}, \mathbf{t}) + Q^{\Omega_{p+1}}(\mathtt{R}, \mathbf{t})$. This is in contrast with the multiresolution approach of [11] where the target has to be computed from scratch for each resolution level. This advantage also allows us to use a large number of frequency bands without considerable extra cost, whereas [11] has to use a fairly few number of resolution levels. Our method, therefore, for each candidate solution performs just enough computations required to make the reject/update decision.

4. Integrating into a global optimization framework

4.1. Lipschitz optimization

Lipschitz optimization is based on the idea of Lipschitz continuity. A function $Q\colon \mathcal{C} \subseteq \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous if there exists a constant L such that

$$|Q(\mathbf{u}) - Q(\mathbf{u}')| \leq L \, \|\mathbf{u} - \mathbf{u}'\| \,, \tag{12}$$

for all $\mathbf{u}, \mathbf{u}' \in \mathcal{C}$. The smallest such L is called the *Lipschitz constant*. For differentiable functions the Lipschitz constant is equal to the supremum of the function derivative inside the domain $\mathcal{C}$. Usually, the target function has different sets of parameters, $Q(\mathbf{u}) = Q(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q)$. In this case one can define a constant L_i for each subset of variables $\mathbf{u}_i$. Then we have

$$\big|Q(\mathbf{u}_1, .., \mathbf{u}_q) - Q(\mathbf{u}'_1, .., \mathbf{u}'_q)\big| \leq \sum_{i=1}^{q} L_i \, \|\mathbf{u}_i - \mathbf{u}'_i\| \,. \tag{13}$$

In Lipschitz optimization the parameter space is divided into a number of subregions, usually hypercubes. For a hypercube $\mathcal{C}$ the target function is computed at its centre $\mathbf{u}_{\mathcal{C}}$ to get $Q(\mathbf{u}_{\mathcal{C}})$. Using (12) or (13) we have

$$Q(\mathbf{u}) \leq Q(\mathbf{u}_{\mathcal{C}}) + B_{\text{Lip}}(\mathcal{C}), \tag{14}$$

where $B_{\text{Lip}}(\mathcal{C})$ is the Lipschitz bound corresponding to the cube $\mathcal{C}$. For a single set of parameters it can be obtained using (12) as

$$B_{\text{Lip}}(\mathcal{C}) = L \cdot r(\mathcal{C}), \tag{15}$$

where $r(\mathcal{C}) = \sup_{\mathbf{u} \in \mathcal{C}} \|\mathbf{u} - \mathbf{u}_{\mathcal{C}}\|$. Likewise, it can be obtained from (13) for different sets of parameters. Assume $Q(\mathbf{u})$ is to be maximized and let Q^* be the so far best target value. For each cube $\mathcal{C}$, if $Q(\mathbf{u}_{\mathcal{C}}) + B_{\text{Lip}}(\mathcal{C}) < Q^*$ then the optimal solution cannot be inside the cube $\mathcal{C}$, and the cube can be safely discarded. If $Q(\mathbf{u}_{\mathcal{C}}) + B_{\text{Lip}}(\mathcal{C}) \geq Q^*$ then $\mathcal{C}$ is split into smaller cubes, and each of the sub-cubes is scheduled to be examined.

4.2. The Lipschitz bounds

Lipschitz bounds have previously been used in rigid registration, see, for example, [1]. We adapt the bounds from [11] to our frequency domain framework. These bounds are shown to be tighter than those of [1].

Remember that in (7) the target function was written as the sum of target functions over different frequency bands $Q^{\Omega_i}(\mathtt{R}, \mathbf{t})$. Here, we find a Lipschitz constant for each Q^{Ω_i}. First, the target function (7) is reformulated as

$$Q^{\Omega_i}(\mathtt{R}, \mathbf{t}) = \int_{\Omega_i} \mathbf{F}(\mathbf{z})^T \, \Gamma(2\pi \mathbf{t}^T \mathbf{z}) \, \mathbf{G}(\mathtt{R}\mathbf{z}) \, d\mathbf{z}, \tag{16}$$

where $\mathbf{F}(\mathbf{z}), \mathbf{G}(\mathbf{z}) \in \mathbb{R}^2$ are two-vectors containing the real and imaginary parts of $F(\mathbf{z})$ and $G(\mathbf{z})$ respectively, and $\Gamma(\beta)$ is the $2{\times}2$ rotation matrix of angle β. Notice that (16) is valid only when F and G are Fourier transforms of *real* signals. Using this representation, the magnitude of the derivatives of the target function is bounded as follows:

$$\begin{aligned}
\left\| \frac{d}{d\mathbf{t}} Q^{\Omega_i}(\mathtt{R}, \mathbf{t}) \right\| &\leq 2\pi \int_{\Omega_i} \left\| \mathbf{z} \, \mathbf{F}(\mathbf{z})^T \, \Gamma'(2\pi \mathbf{t}^T \mathbf{z}) \, \mathbf{G}(\mathtt{R}\mathbf{z}) \right\| d\mathbf{z} \\
&\leq 2\pi \, r_{i+1} \int_{\Omega_i} |\mathbf{F}(\mathbf{z})^T \, \Gamma'(2\pi \mathbf{t}^T \mathbf{z}) \, \mathbf{G}(\mathtt{R}\mathbf{z})| \, d\mathbf{z} \\
&\leq 2\pi \, r_{i+1} \int_{\Omega_i} \|\mathbf{F}(\mathbf{z})\| \, \|\mathbf{G}(\mathtt{R}\mathbf{z})\| \, d\mathbf{z} \\
&\leq 2\pi \, r_{i+1} \sqrt{\textstyle\int_{\Omega_i} \|\mathbf{F}(\mathbf{z})\|^2 \, d\mathbf{z}} \sqrt{\textstyle\int_{\Omega_i} \|\mathbf{G}(\mathbf{z})\|^2 \, d\mathbf{z}} \\
&\stackrel{\text{def}}{=} L^{\Omega_i}_{\mathbf{t}}.
\end{aligned} \tag{17}$$

Notice that $\Gamma'(\beta) = \frac{d}{d\beta}\Gamma(\beta)$ is again a rotation matrix and $r_{i+1} = \sup_{\mathbf{z} \in \Omega_i} \|\mathbf{z}\|$. For the 2D case, the only rotation parameter is an angle of rotation θ. Then

$$\begin{aligned}
|\frac{d}{d\theta} Q^{\Omega_i}(\mathtt{R}, \mathbf{t})| &\leq \int_{\Omega_i} \left| \mathbf{F}(\mathbf{z})^T \, \Gamma(2\pi \mathbf{t}^T \mathbf{z}) \, J_{\mathbf{G}}(\mathtt{R}\mathbf{z}) \frac{d\mathtt{R}}{d\theta} \mathbf{z} \right| d\mathbf{z} \\
&\leq \sqrt{\int_{\Omega_i} \|\mathbf{F}(\mathbf{z})\|^2 \, d\mathbf{z}} \sqrt{\int_{\Omega_i} \left\| J_{\mathbf{G}}(\mathbf{z}) \frac{d\mathtt{R}}{d\theta} \mathtt{R}^{-1} \mathbf{z} \right\|^2 d\mathbf{z}}, \\
&= \sqrt{\textstyle\int_{\Omega_i} \|\mathbf{F}(\mathbf{z})\|^2 \, d\mathbf{z}} \sqrt{\textstyle\int_{\Omega_i} \|J_{\mathbf{G}}(\mathbf{z}) \mathbf{z}^{\perp}\|^2 \, d\mathbf{z}} \stackrel{\text{def}}{=} L^{\Omega_i}_{\theta},
\end{aligned} \tag{18}$$

where $J_{\mathbf{G}}$ is the Jacobian matrix of $\mathbf{G}$ and and $\mathbf{z}^{\perp}$ is the vector $\mathbf{z}$ rotated by 90 degrees clockwise. For the 3D case (as well as the cases of reflective symmetry), the

Lipschitz constant can be adapted from [11] in a similar way. Here, to obtain even tighter estimates of the Lipschitz constant, we further divide each frequency band into smaller sub-bands, and get the Lipschitz constant as the sum of the constants over each sub-band.

For 2D registration the cubes have the form $\mathcal{C}{=}\mathcal{C}_{\mathbf{t}}{\times}\mathcal{C}_{\theta}$ where $\mathcal{C}_{\mathbf{t}}$ and $\mathcal{C}_{\theta}$ are the cubes for the translation vector and the rotation angle respectively. From (13) the Lipschitz bound can be found as

$$B_{\text{Lip}}^{\Omega_i}(\mathcal{C}) = L_{\mathbf{t}}^{\Omega_i}\, r(\mathcal{C}_{\mathbf{t}}) + L_{\theta}^{\Omega_i}\, r(\mathcal{C}_{\theta}), \tag{19}$$

where $r(\mathcal{C}_{\mathbf{t}})$ and $r(\mathcal{C}_{\theta})$ can be calculated similarly to $r(\mathcal{C})$ in (15). For 3D registration (and symmetry detection) $B_{\text{Lip}}^{\Omega_i}(\mathcal{C})$ can be found in a similar way. Similarly, we can define Lipschitz bounds for each frequency ball:

$$B_{\text{Lip}}^{\mathcal{B}_p}(\mathcal{C}) = \sum_{i=0}^{p} B_{\text{Lip}}^{\Omega_i}(\mathcal{C}). \tag{20}$$

Notice that, as $B_{\text{Lip}}^{\Omega_i}(\mathcal{C}) \geq 0$, we have

$$B_{\text{Lip}}^{\mathcal{B}_0}(\mathcal{C}) \leq B_{\text{Lip}}^{\mathcal{B}_1}(\mathcal{C}) \leq \ldots \leq B_{\text{Lip}}^{\mathcal{B}_P}(\mathcal{C}). \tag{21}$$

One can easily check that

$$Q^{\mathcal{B}_p}(\mathtt{R}, \mathbf{t}) \leq Q^{\mathcal{B}_p}(\mathtt{R}_{\mathcal{C}}, \mathbf{t}_{\mathcal{C}}) + B_{\text{Lip}}^{\mathcal{B}_p}(\mathcal{C}), \tag{22}$$

where $\mathtt{R}_{\mathcal{C}}$ and $\mathbf{t}_{\mathcal{C}}$ correspond to the central points of $\mathcal{C}_{\theta}$ and $\mathcal{C}_{\mathbf{t}}$ respectively. Now, for each cube $\mathcal{C}$ we define a *total bound* for all p,

$$B_{\text{total}}^{\mathcal{B}_p}(\mathcal{C}) = B_{\text{res}}^{\mathcal{B}_p} + B_{\text{Lip}}^{\mathcal{B}_p}(\mathcal{C}). \tag{23}$$

Then using (11) and (22) we have

$$Q(\mathtt{R}, \mathbf{t}) \leq Q^{\mathcal{B}_p}(\mathtt{R}_{\mathcal{C}}, \mathbf{t}_{\mathcal{C}}) + B_{\text{total}}^{\mathcal{B}_p}(\mathcal{C}). \tag{24}$$

4.3. A multiband Lipschitz optimization algorithm

We are now ready to describe the algorithm in depth. For each cube $\mathcal{C}$ we compute the total bound $B_{\text{total}}^{\mathcal{B}_p}(\mathcal{C})$ for all p, and determine the minimizer p^* as

$$p^* = \operatorname*{argmin}_{p}\, B_{\text{total}}^{\mathcal{B}_p}(\mathcal{C}). \tag{25}$$

Now, we keep computing $Q^{\mathcal{B}_p}(\mathtt{R}_{\mathcal{C}}, \mathbf{t}_{\mathcal{C}})$ for $p = 0, 1, 2, \ldots$ up to p^*. For each p we examine the upper bound $Q^{\mathcal{B}_p}(\mathtt{R}_{\mathcal{C}}, \mathbf{t}_{\mathcal{C}}) + B_{\text{total}}^{\mathcal{B}_p}(\mathcal{C})$ in (24). If this upper bound is smaller than Q^*, the best target value found so far, then the current cube is rejected. If the cube is not rejected for $p = 0, 1, \ldots, p^*$, we split the cube $\mathcal{C}$. We do not examine values of p above p^*, as then the total bound would become larger, but one cannot say that the target function becomes smaller. In most cases, if $Q^{\mathcal{B}_p}(\mathtt{R}_{\mathcal{C}}, \mathbf{t}_{\mathcal{C}}) + B_{\text{total}}^{\mathcal{B}_p}(\mathcal{C})$ is larger than Q^* for $p = 0, \ldots, p^*$, it remains larger than Q^* for $p > p^*$.

From (10) and (21) we know that as p increases the value of $B_{\text{res}}^{\mathcal{B}_p}$ decreases, while $B_{\text{Lip}}^{\mathcal{B}_p}(\mathcal{C})$ increases. At the early iterations of the algorithm, the cubes $\mathcal{C}$ are large and $B_{\text{Lip}}^{\mathcal{B}_p}(\mathcal{C})$ is considerably larger compared to $B_{\text{res}}^{\mathcal{B}_p}$. Therefore, the bound $B_{\text{total}}^{\mathcal{B}_p}(\mathcal{C}) = B_{\text{res}}^{\mathcal{B}_p} + B_{\text{Lip}}^{\mathcal{B}_p}(\mathcal{C})$ has its minimum at a small p, and p^* is small. Therefore, only a few number of bands needs to be examined before making the split/reject decision. As the algorithm continues and the cubes are split to smaller cubes, the bounds $B_{\text{Lip}}^{\mathcal{B}_p}(\mathcal{C})$ become smaller compared to $B_{\text{res}}^{\mathcal{B}_p}$, and p^* grows larger. Therefore, the target function is computed up to higher frequency bands, which results in a more accurate approximation of the actual target function. In this way, many *split vs. reject* decisions are made at the initial stages of the algorithm by just computing the target value in lower frequency bands, and fine tuning is performed in the final stages by examining higher frequency bands. Fig. 2 plots $B_{\text{total}}^{\mathcal{B}_p}(\mathcal{C})$ for different cube sizes.

The procedure is summarized in Algorithm 1. In order to make a fair experimental comparison with [11], we use the same tree search scheme, namely the *breadth-first* search. A delimiter called **new-level** is used to indicate that an entire level of the search tree has been visited. When this happens, we first compute the target function Q at $\mathbf{u}_{\text{lb}}$, the parameter giving the best (largest) lower bound Q_{lb}^* on the optimal target value, and update the best target value if necessary (line 9). Then we check the finishing condition $Q_{\text{up}}^* - Q^* < \epsilon$ (line 10). It basically checks if the currently best target value Q^* is within a margin ϵ of the optimal target value. This is done using an upper bound Q_{up}^* on the entire target function, which is computed in the course of searching an entire level of the search tree.

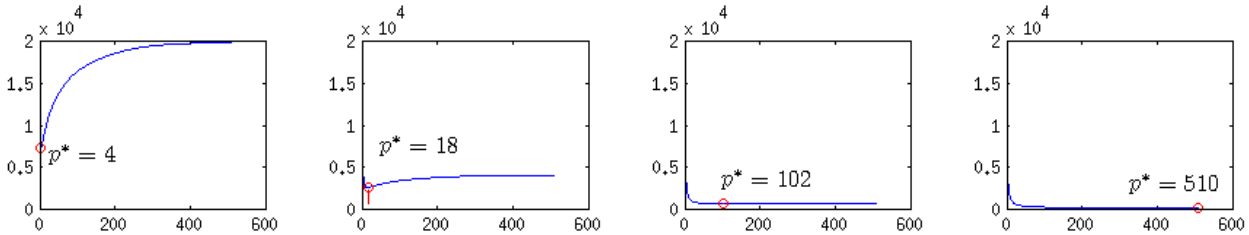
In line 30 of the algorithm the cube is split. It happens when $Q^{\mathcal{B}_{p^*}}(\mathtt{R}_{\mathcal{C}}, \mathbf{t}_{\mathcal{C}}) + B_{\text{total}}^{\mathcal{B}_{p^*}}(\mathcal{C}) > Q^*$, which means

$$B_{\text{Lip}}^{\mathcal{B}_{p^*}}(\mathcal{C}) > Q^* - Q^{\mathcal{B}_{p^*}}(\mathbf{u}_{\mathcal{C}}) - B_{\text{res}}^{\mathcal{B}_{p^*}}(\mathcal{C}) \stackrel{\text{def}}{=} B_{\text{margin}}. \tag{26}$$

Therefore, B_{margin} is the maximum permitted value of $B_{\text{Lip}}^{\mathcal{B}_{p^*}}(\mathcal{C})$ for not splitting. The splitting procedure works as follows. Consider q sets of registration parameters $\mathbf{u}_1, \ldots, \mathbf{u}_q$, with $\mathcal{C}{=}\mathcal{C}_{\mathbf{u}_1}{\times}\mathcal{C}_{\mathbf{u}_2}{\times}\cdots{\times}\mathcal{C}_{\mathbf{u}_q}$. In 2D registration $\mathbf{u}_1 = \mathbf{t}$ and $\mathbf{u}_2 = \theta$. The Lipschitz bound is the sum of Lipschitz bounds of different parameters:

$$B_{\text{Lip}}^{\mathcal{B}_{p^*}}(\mathcal{C}) = B_{\text{Lip}}^{\mathcal{B}_{p^*}}(\mathcal{C}_{\mathbf{u}_1}) + \ldots + B_{\text{Lip}}^{\mathcal{B}_{p^*}}(\mathcal{C}_{\mathbf{u}_q}) \tag{27}$$

where $B_{\text{Lip}}^{\mathcal{B}_{p^*}}(\mathcal{C}_{\mathbf{u}_i}) = L_{\mathbf{u}_i}^{\mathcal{B}_{p^*}}\, r(\mathcal{C}_{\mathbf{u}_i})$ as in (19). By possibly relabeling the parameters, we assume that



$\mathcal{C}_\theta$: 20 degrees, $\mathcal{C}_\mathbf{t}$: 25×25 pixels | $\mathcal{C}_\theta$: 4 degrees, $\mathcal{C}_\mathbf{t}$: 5×5 pixels | $\mathcal{C}_\theta$: 0.5 degrees, $\mathcal{C}_\mathbf{t}$: 1×1 pixels | $\mathcal{C}_\theta$: 0.1 degrees, $\mathcal{C}_\mathbf{t}$: 0.2×0.2 pixels

Figure 2. Plotting $B^{\mathcal{B}_p}_{\text{total}}(\mathcal{C})$ as a function of p for different sizes for the cube $\mathcal{C} = \mathcal{C}_\theta \times \mathcal{C}_\mathbf{t}$. The frequency domain is divided into 511 radial bands $p = 0, 1, \ldots, 510$. The minimizer p^* increases as the cube gets smaller.

$B_{\text{Lip}}(\mathcal{C}_{\mathbf{u}_1}) \leq B_{\text{Lip}}(\mathcal{C}_{\mathbf{u}_2}) \leq \ldots \leq B_{\text{Lip}}(\mathcal{C}_{\mathbf{u}_q})$. Now, for each i we split the cube $\mathcal{C}$ across the parameter $\mathbf{u}_i$ if $\sum_{j=i}^{q} B^{\mathcal{B}_{p^*}}_{\text{Lip}}(\mathcal{C}_{\mathbf{u}_j}) \geq B_{\text{margin}}$.

4.4. Implementation

To implement the algorithm we need to compute integrals over each band Ω_i in three occasions, namely for computing the target function (7), the residual bounds (8), and the Lipschitz constants (17) and (18). Here, the integrals are computed numerically, using the DFT coefficients in the light of equation (5). Since the bands are circular, it is convenient to use a square $M \times M$ grid for the DFT. Nonetheless, non-square DFTs can also be used, in which the circular bands will turn to elliptic bands in the DFT pixel domain. In this paper we use $M = N = \max(m_f, n_f, m_g, n_g)$ for an $m_f \times n_f$ image $\{f_i\}$ and an $m_g \times n_g$ image $\{g_i\}$, to compute the bounds. However, since these bounds are computed just once at the beginning of the algorithm, one can use a higher resolution DFT as well.

As for computing the target function, first for $\{f_\mathbf{i}\}$ we take an $M \times M$ DFT, where M is chosen as above. Then we can approximate the target integral as

$$Q^{\Omega_i}(\mathtt{R}, \mathbf{t}) = \int_{\Omega_i} \overline{F(\mathbf{z})}\, G(\mathtt{R}\mathbf{z})\, e^{2\pi j \mathbf{t}^T \mathbf{z}}\, d\mathbf{z}.$$
$$\approx \frac{1}{M^d} \sum_{\mathbf{k} \in M \cdot \Omega_i} \overline{F_\mathbf{k}}\, \tilde{G}(\mathtt{R}\mathbf{k}/M)\, e^{2\pi j \mathbf{t}^T \mathbf{k}/M}, \quad (28)$$

where $M \cdot \Omega_i = \{M\mathbf{z} \,|\, \mathbf{z} \in \Omega_i\}$ and $\tilde{G}(\mathbf{z})$ approximates the value of $G(\mathbf{z})$ by interpolating the DFT coefficient of $\{g_\mathbf{i}\}$. Notice that $F_\mathbf{k} = F(\mathbf{k}/M)$ according to (5). For $\{g_\mathbf{i}\}$ we perform an $M_g \times M_g$ DFT where M_g is usually larger than M. Now, we can find $\tilde{G}(\mathbf{z})$ from the DFT coefficients $G_\mathbf{l}$ as follows

$$\tilde{G}(\mathbf{z}) = \sum_{\mathbf{l}} G_\mathbf{l}\, h(\mathbf{l} - M_g \mathbf{z}) \quad (29)$$

where h is the interpolation kernel. In this paper we use a bilinear interpolation for which h is the multiplication of *triangular kernels* along each dimension. Therefore, to compute (29) we just need to consider four neighbouring DFT coefficients of $M_g\mathbf{z}$, denoted here by $\mathcal{N}_4(M_g\mathbf{z})$. By replacing into (28) we have

$$Q^{\Omega_i}(\mathtt{R}, \mathbf{t}) \approx M^{-d} \sum_{\mathbf{k} \in M \cdot \Omega_i} \overline{F_\mathbf{k}} \sum_{\mathbf{l} \in \mathcal{N}_4(r(\mathbf{k}))} w_{\mathbf{k},\mathbf{l}}\, G_\mathbf{l}\, e^{\frac{2\pi j \mathbf{t}^T \mathbf{k}}{M}}, \quad (30)$$

where $w_{\mathbf{k},\mathbf{l}} = h(\mathbf{l} - r(\mathbf{k}))$, and $r(\mathbf{k}) = \frac{M_g \mathtt{R}\mathbf{k}}{M}$. Notice that, the number of computations required for computing the above for all Ω_i is proportional to the number of the DFT pixels $F_\mathbf{k}$ which is M^2 for the 2D case. Therefore, taking a higher resolution DFT of $\{f_\mathbf{i}\}$ (i.e. increasing M) will result in more computations. However, we can still get better approximations by taking a higher resolution DFT for $\{g_\mathbf{i}\}$. We have found experimentally that using a bilinear interpolation and choosing $M_g = 6M$ gives sufficiently accurate results in most cases. Therefore, we use this setup in our experiments. Since, $M = \max(m_f, n_f, m_g, n_g)$ is about the size of the image dimensions, we can encounter the problem of *aliasing* in the space domain. The problem is not serious for images with a fairly large dark margin. For other images one can use tricks such as inverting the image intensity in case of a light background, or zero padding the original image. The latter, though, is equivalent to increasing M, and thus the computation time of the target function. One also needs to limit the amount of translation in each dimension. Notice that (30) is periodic in $\mathbf{t}$ with a period of M. Therefore, one hard constraint on the amount of translation is $M/2$ along each dimension. However, this can be smaller depending on the size of dark margins of the images and the value taken for M. Therefore, using a larger M might be necessary in some cases.

5. Experiments

All experiments are conducted on an Intel Xeon 3.40GHz machine with 24GB of RAM. The algorithm is written in C++. In all the experiments the images are rotated by random angles that are drawn uniformly

Algorithm 1 A Lipschitz optimization algorithm for intensity-based rigid registration in the frequency domain.

1: $Q^* \leftarrow -\infty$ ▷ so-far best target value
2: $Q^*_{\text{lb}} \leftarrow -\infty$ ▷ best lower-bound on optimal target
3: $Q_{\text{up}} \leftarrow -\infty$ ▷ an upper-bound on target function
4: QUEUE.PUSH($\mathcal{C}_0$)
5: QUEUE.PUSH(**new-level**)
6: **loop**
7: $\mathcal{C} \leftarrow$ QUEUE.POP()
8: **if** $\mathcal{C} =$ **new-level then** ▷ new level
9: $Q^*, \mathbf{u}^* \leftarrow$UPDATEMAX$(Q^*, Q(\mathbf{u}^*_{\text{lb}}), \mathbf{u}^*, \mathbf{u}^*_{\text{lb}})$
10: **if** $Q^*_{\text{up}} - Q^* < \epsilon$ **then**
11: **return** $\mathbf{u}^*$
12: **end if**
13: QUEUE.PUSH(**new-level**)
14: $Q_{\text{up}} \leftarrow -\infty$
15: **continue loop** ▷ go to line 6
16: **end if**
17: $\mathbf{u}_{\mathcal{C}} \leftarrow \mathcal{C}$.CENTRE()
18: $p^* \leftarrow \text{argmin}_p\, B^{\mathcal{B}_p}_{\text{total}}(\mathcal{C})$
19: $Q \leftarrow 0$ ▷ $Q^{\mathcal{B}_p}(\mathbf{u}_{\mathcal{C}})$
20: **for** $p \leftarrow 0$ **to** p^* **do**
21: $Q \leftarrow Q + Q^{\Omega_p}(\mathbf{u}_{\mathcal{C}})$ ▷ update $Q^{\mathcal{B}_p}(\mathbf{u}_{\mathcal{C}})$
22: **if** $Q + B^{\mathcal{B}_p}_{\text{total}}(\mathcal{C}) < Q^*$ **then**
23: **continue loop** ▷ reject $\mathcal{C}$, go to line 6
24: **end if**
25: **end for**
26: $Q_{\text{lb}} \leftarrow Q - B^{\mathcal{B}_{p^*}}_{\text{res}}$
27: $Q_{\text{up}} \leftarrow \max(Q_{\text{up}}, Q + B^{\mathcal{B}_{p^*}}_{\text{total}}(\mathcal{C}))$
28: $Q^*_{\text{lb}}, \mathbf{u}^*_{\text{lb}} \leftarrow$UPDATEMAX$(Q^*_{\text{lb}}, Q_{\text{lb}}, \mathbf{u}^*_{\text{lb}}, \mathbf{u}_{\mathcal{C}})$
29: $Q^*,\ \mathbf{u}^* \leftarrow$UPDATEMAX$(Q^*, Q^*_{\text{lb}}, \mathbf{u}^*, \mathbf{u}^*_{\text{lb}})$
30: **for** $\mathcal{C}'$ **in** $\mathcal{C}$.SPLIT() **do**
31: QUEUE.PUSH($\mathcal{C}'$)
32: **end for**
33: **end loop**

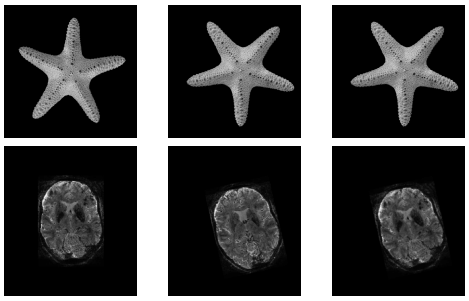
34: **procedure** UPDATEMAX$(Q_1, Q_2, \mathbf{u}_1, \mathbf{u}_2)$
35: **if** $Q_1 \geq Q_2$ **then**
36: **return** $Q_1, \mathbf{u}_1$
37: **else**
38: **return** $Q_2, \mathbf{u}_2$
39: **end if**
40: **end procedure**

in the interval $(-\pi, \pi)$. Next the images are translated according to two random parameters that are drawn uniformly in the intervals $(-m, m)/8$ and $(-n, n)/8$, where m is the number of rows and n is the number of columns of the image, respectively.

The only work we found on speeding up globally optimal rigid intensity-based registration is [11] and therefore the performance of the algorithm will be compared to this work. In the code available for that method, the preprocessing stages such as obtaining the bounds and building the multiresolution pyramid is done using Matlab, while the main branch-and-bound loop is written in C++. Therefore, in order to make a fair comparison we just compare the running time of the main loop for both the algorithms. The termination threshold ϵ is 0.01 of the geometric average of the energies of the two images, as in [11]. This choice is crucial for a fair comparison.

In our current implementation, the computation of the target function with a single choice of rotation and translation parameters is considerably slower than the computation in the space domain as implemented in [11]. This can sometimes be up to a factor of 8. There are several reasons behind this, e.g. that in the space domain only the intersecting parts of the two images are computed. However, the most important reason is that in the implementation of the space domain algorithm, the pixels are visited in order: after (i_1, i_2) it visits (i_1, i_2+1). Thus, for rotating the pixel (i_1, i_2+1) we just need to add a constant vector to the rotated version of (i_1, i_2). However, in the frequency domain approach the DFT pixels are visited in the order of their corresponding frequency bands. Furthermore, we use a large array for computing the DFT of G as the reference image, as $M_g = 6M$, then the DFT pixels of $\{G_{\mathbf{k}}\}$ are 36 times that of $\{F_{\mathbf{k}}\}$, and more than 72 times that of $\{f_{\mathbf{i}}\}$ since it contains complex numbers. Since the elements of this large array are not nearly accessed in the order they stored in the memory and the pixels are visited in the order of the frequency bands, it cannot fully take advantage of the memory caching of the CPU. For each experiment we report the speed-up factor when using the frequency based approach relative to the space domain counterpart. We also report the speed-up factor after normalizing with a correlation factor, i.e., the speed-up factor is multiplied by the average time it takes to compute a single correlation for the multiband approach divided by the corresponding number for the multiresolution approach.

First, we experiment on two examples from [11], the starfish and brain MRI images, see Fig. 3. The starfish image is of size 870×870 pixels and is registered to a rigidly transformed copy of itself. The brain images are two different slices of a brain MRI volume with sizes 312×276 pixels, where the second image is randomly transformed. These two cases are of interest since, according to [11], a multiresolution scheme is about 260 times faster than the single-resolution approach for the starfish image. For the brain images, however, the speed gain is about 4.5 which is relatively



(a) (b) (c)

Figure 3. Registering two images of a starfish and two different slices of a single brain MRI image for random values of the rotation and translation. (a) the first image, (b) the second image rotated and translated, and (c) the first image registered to the second image. The starfish image is downloaded from http://www.ck12.org and the MRI data is obtained from [2].

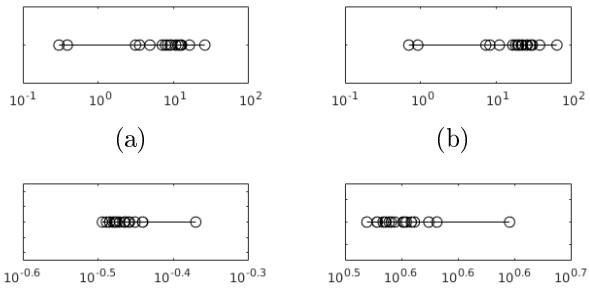
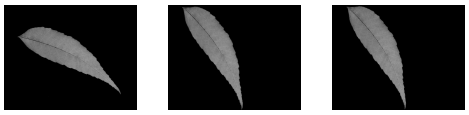


Figure 4. The speed-up factors and normalized speed-up factors for (a-b) the starfish and (c-d) the brain MRI image. Notice that the results are reported in logarithmic scale.

small. The speed-up factors of using the multiband approach instead of the multiresolution approach are reported in Fig. 4 for 20 runs of the algorithms. In the starfish example, even the unnormalized speed gain is considerably large in most cases, where for the brain image the multiband approach is always slower if not normalized for the computation time of a single correlation. One explanation could be as follows. The brain MRI image does not fully take advantage from a multi-level approach, as reported in [11]. Thus, it is expected that in many iterations most of the bands or resolution levels of the two images are examined to make the reject vs. split decision, and the algorithm does not take full advantage of the multi-level paradigm. Therefore, the higher computation time of a single correlation in the multiband approach shows its effect.

We also make a comparison of the algorithms for a data set consisting of 32 different leaves with sizes 760×960 pixels [18], see Fig. 5. We ran the algorithm 10 times for each leaf, where in each case the second image obtained by randomly rotating and translating the first image, plus applying a slight affine warp. The histogram of the speed-up factors of the individual experiments for all the leaves when using the multiband approach is shown in Fig. 6. Clearly the multiband algorithm outperforms the multiresolution algorithm for most of the cases. Note that the histograms are computed on the 10-logarithm of the speed-up factors.



(a) (b) (c)

Figure 5. Registering two images of leaf number 17 from the leaf data set [18], (a) the first image, (b) the second image, and (c) the first image registered to the second image.

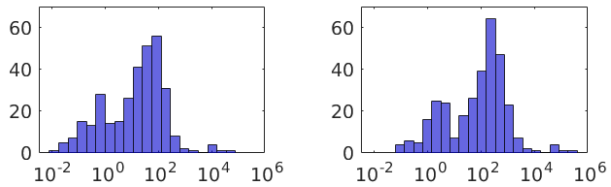


Figure 6. The histogram of the (left) unnormalized, and (right) normalized speed-up factors of the multiband over the multiresolution approach for the leaf data set.

6. Conclusion and future work

In this paper, we have proposed a novel algorithm that performs an exhaustive search for rigid intensity-based registration in the frequency domain. We have shown that by letting the algorithm operate from lower to higher frequencies, rather than from lower to higher resolutions, it is possible to reduce the number of extra computations in a globally optimal approach. We demonstrated considerable speed gains over the multiresolution approach in cases where the images had a potential to take advantage of a multi-level approach. For the cases where a multi-level approach did not offer a huge speed-up, our implementation of the multiband approach can perform worse. This is because in our current implementation a single evaluation of the target function is much slower than the multiresolution approach in the space domain. This problem might be resolved by resampling the DFTs in the polar coordinates. One possible extension of our work would be to use another basis function with better energy compaction properties than the Fourier basis. Furthermore, extending the algorithm to cover more general transformations such as similarity, affine, and nonlinear mappings would be another interesting line of research.

References

[1] D. Cremers, F. R. Schmidt, and F. Barthel. Shape priors in variational image segmentation: Convexity, lipschitz continuity and globally optimal solutions. Anchorage, Alaska, June 2008.

[2] B. U. Forstmann, M. C. Keuken, A. Schafer, P.-L. Bazin, A. Alkemade, and R. Turner. Multi-modal ultra-high resolution structural 7-tesla mri data repository. *Scientific data*, 1, 2014.

[3] M. Gharavi-Alkhansari. A fast globally optimal algorithm for template matching using low-resolution pruning. *Image Processing, IEEE Transactions on*, 10(4):526–533, 2001.

[4] D. P. Huttenlocher and W. J. Rucklidge. A multi-resolution technique for comparing images using the hausdorff distance. Technical report, Cornell University, 1992.

[5] S. Korman, D. Reichman, G. Tsur, and S. Avidan. Fast-match: Fast affine template matching. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2331–2338. IEEE, 2013.

[6] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. In *IEEE Conference on Cybernetics and Society*, pages 163–165, 1975.

[7] C.-H. Lee and L.-H. Chen. A fast motion estimation algorithm based on the block sum pyramid. *Image Processing, IEEE Transactions on*, 6(11):1587–1591, 1997.

[8] H. Li and R. Hartley. The 3d-3d registration problem revisited. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[9] W. Li and E. Salari. Successive elimination algorithm for motion estimation. *Image Processing, IEEE Transactions on*, 4(1):105–107, 1995.

[10] S. Mattoccia, F. Tombari, and L. D. Stefano. Fast full-search equivalent template matching by enhanced bounded correlation. *Image Processing, IEEE Transactions on*, 17(4):528–538, 2008.

[11] B. Nasihatkon and F. Kahl. Multiresolution search of the rigid motion space for intensity based registration. *preprint*, 2015.

[12] W. Ouyang and W.-K. Cham. Fast algorithm for walsh hadamard transform on sliding windows. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (1):165–171, 2009.

[13] W. Ouyang, F. Tombari, S. Mattoccia, L. D. Stefano, and W.-K. Cham. Performance evaluation of full search equivalent pattern matching algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(1):127–143, 2012.

[14] A. J. Parra Bustos, T.-J. Chin, and D. Suter. Fast rotation search with stereographic projections for 3d registration. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3930–3937. IEEE, 2014.

[15] F. Pfeuffer, M. Stiglmayr, and K. Klamroth. Discrete and geometric branch and bound algorithms for medical image registration. *Annals of Operations Research*, 196(1):737–765, 2012.

[16] B. S. Reddy and B. N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *Trans. Img. Proc.*, 5(8):1266–1271, Aug. 1996.

[17] F. Tombari, S. Mattoccia, and L. Di Stefano. Full-search-equivalent pattern matching with incremental dissimilarity approximations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):129–141, 2009.

[18] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pages 11–16. IEEE, 2007.

[19] J. Yang, H. Li, and Y. Jia. Go-icp: Solving 3d registration efficiently and globally optimally. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.