Multiresolution Search of the Rigid Motion Space for Intensity-Based Registration

Behrooz Nasihatkon, Fredrik Kahl

***Abstract*—We study the relation between the correlation-based target functions of low-resolution and high-resolution intensity-based registration for the class of rigid transformations. Our results show that low-resolution target values can tightly bound the high-resolution target function in natural images. This can help with analyzing and better understanding the process of multiresolution image registration. It also gives a guideline for designing multiresolution algorithms in which the search space in higher resolution registration is restricted given the fitness values for lower resolution image pairs. To demonstrate this, we incorporate our multiresolution technique into a Lipschitz global optimization framework. We show that using the multiresolution scheme can result in large gains in the efficiency of such algorithms. The method is evaluated by applying to the problems of 2D registration, 3D rotation search, and the detection of reflective symmetry in 2D and 3D images.**

***Index Terms*—Intensity-based Image Registration, Multiresolution Registration, Lipschitz Optimization, Early Elimination**

I. Introduction

This paper studies the implications of low-resolution image registration on high-resolution registration. We focus on rigid intensity-based registration with correlation fitness measure, and demonstrate how the fitness computed for low-resolution image pairs can bound the high-resolution fitness. Our results can be used for the design and analysis of multiresolution registration techniques, especially globally optimal approaches.

Most of the approaches to globally optimally solving the registration problem deal with the alignment of point sets [1], [2], [3], [4], [5] or other geometric properties [6]. For example, Li and Hartley in [2] propose an algorithm to register two sets of points in a globally optimal way based on Lipschitz optimization [7]. Yang et al. [3] obtain a globally optimal solution by combing a branch and bound scheme with the Iterative Closest Point algorithm.

Such approaches have not been applied, as much, to intensity-based registration, due to the high computational cost. Nonetheless, in some applications, it is worth to sacrifice speed for accuracy. A notable example is the construction of Shape Models [8] where the models are built once and for all.

There are not many references on globally optimal intensity-based registration. Cremers et al. [9] use Lipschitz optimization to search for optimal rotation and translation parameters for alignment of 2D shape priors. The computational burden is still the major limiting factor of such algorithms. It is well known that the multiresolution techniques can speed up the registration algorithms [10], [11], [12]. However, to use a multiresolution scheme in a global optimization framework we need a theory which describes the relation among the target values of different resolutions. We provide such a theory in this paper, and based on that, propose an algorithm in which many candidate areas of the search space get rejected by examining the target value in lower resolutions.

This idea of *early elimination* has been well explored in the area of template matching, where a small template patch is searched in a target image. The successive elimination approach [13] rules out many candidate patch locations by examining a low-cost lower-bound on the cost function. The approach is extended in [14] to a multilevel algorithm, each level providing a tighter bound with higher computational cost. A multitude of approaches has been proposed based on the same idea of a multilevel succession of bounds [15], [16], [17], [18], [19], [20], [21]. In [22] some of the most successful algorithms of this class have been compared. Among them, the work of Gharavi-Alkhansari [23] is very related to ours. In this approach, each level corresponds to a different image resolution. By moving from low to high resolution we get tighter bounds requiring more computations. Typically, many of the candidate solutions are eliminated by computing the match measure in lower resolutions. However, in this method, like all the approaches above, the search is only performed on a grid of translation vectors with 1-pixel intervals. Most of the work to extend this to more general transformations do not fully search the parameter space. We should mention, however, the work of Korman et al. [24] for affine pattern matching. It considers a grid of affine transformations and makes use of *sublinear approximations* to reduce the cost of examining each transformation. It runs a branch and bound algorithm in which a finer grid is used at each stage. Obtaining a solution close enough to the optimal target is guaranteed with *high probability*. The bounds, however, are in the form of asymptotic growth rates and are *learned* empirically from a large dataset. The algorithm, therefore, is not provably globally optimal in the exact mathematical sense.

Our work mainly deals with the alignment of two images rather than matching a small patch. Similar to [23] we take a multiresolution approach by providing bounds between different resolutions. However, since rotation is involved and also sub-pixel accuracy is required, the registration problem is analyzed in a continuous domain, by interpolating the discrete images. We incorporate the multiresolution paradigm into a Lipschitz optimization framework which searches the entire rigid motion space for a globally optimal solution. We show

Behrooz Nasihatkon is with the Department of Signals and Systems at Chalmers University of Technology, Sweden, MedTech West, Sweden, and ICT innovation center, Sharif University of Technology, Iran. Email: nasihatkon@kntu.ac.ir

Fredrik Kahl is with the Department of Signals and Systems, Chalmers University of Technology, Centre for Mathematical Sciences, Lund University and MedTech West, Sweden. Email: fredrik.kahl@chalmers.se

that the multiresolution approach can lead to huge efficiency gains. Another closely related problem is estimating reflective symmetry in 2D and 3D images [25]. We demonstrate that our algorithm can readily be applied to this problem as well. In short, the major contributions of this paper are

- Providing a framework to analyze the relation among registration targets at different resolutions,
- presenting inter-resolution bounds for several scenarios such as the use of different interpolation techniques or when only one image is decimated, and giving insights about how tighter bounds could be found,
- A novel algorithm to integrate the multiresolution paradigm into a Lipschitz global optimization framework resulting in considerable savings in computations.

We also introduce new effective bounds for the Lipschitz constants which are much smaller than what proposed in the literature. We should mention, however, that our main concern here is to demonstrate the efficiency gains of a multiresolution approach, and not optimizing the single-resolution algorithm itself. After providing a background in Sect. II, we introduce the inter-resolution bounds in Sect. III. A basic grid search algorithm is presented in Sect. IV, before presenting the multiresolution Lipschitz algorithm in Sect. V. We conduct experiments in Sect. VI to evaluate our algorithm on 2D registration and reflective symmetry estimation in 2D and 3D images. Our current implementation is based on breadth-first search, which is very memory demanding for full 3D registration with 6 degrees of freedom. Therefore, we present an example of searching for the optimal rotation after the centre of masses of two 3D shapes are aligned.

II. Background

Consider two images f and g represented as continuous functions $\mathbb{R}^d \to \mathbb{R}$, where $d = 2$ or $d = 3$. For intensity-based rigid registration we seek to maximize the correlation function

$$Q(\mathtt{R}, \mathbf{t}) = \int f(\mathbf{x})\, g\left(\mathtt{R}(\mathbf{x}+\mathbf{t})\right)\, d\mathbf{x}, \tag{1}$$

over $\mathtt{R}$ and $\mathbf{t}$, the rotation matrix and the translation vector[1]. This is equivalent to minimizing the integrated squared error $\int \left(f(\mathbf{x}) - g(\mathtt{R}(\mathbf{x}+\mathbf{t}))\right)^2 d\mathbf{x}$, as the transformation is rigid.

The fitness function (1) can be reformulated in the frequency domain. Let $F(\mathbf{z})$ and $G(\mathbf{z})$ respectively represent the Fourier transforms of $f(\mathbf{x})$ and $g(\mathbf{x})$, and notice that the Fourier transform of $g(\mathtt{R}(\mathbf{x}+\mathbf{t}))$ is equal to $G(\mathtt{R}\mathbf{z})\, e^{2\pi i \mathbf{t}^T \mathbf{z}}$. Applying the Parseval's Theorem to (1) gives

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

where $\overline{F(\mathbf{z})}$ represents the complex conjugate of $F(\mathbf{z})$.

1) Discrete images: For discrete images we have samples $\{f_\mathbf{i}\}$ at discrete locations $\mathbf{i} \in \mathbb{Z}^d$ ($d = 2$ or 3). One could think of $\{f_\mathbf{i}\}$ as samples taken from a continuous image f according to $f_\mathbf{i} = f(T\,\mathbf{i})$, where the scalar T is the sampling period[2]. To get the continuous image from the discrete image we suppose that the continuous image is bandlimited to the corresponding Nyquist frequency $\frac{1}{2T}$. This means that the continuous image can be obtained by *sinc* interpolation:

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

where $\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ for scalars, and

$$\mathrm{sinc}(\mathbf{x}) = \prod_{i=1}^{d} \mathrm{sinc}(x_i), \tag{4}$$

for $\mathbf{x} \in \mathbb{R}^d$. Digital images are usually defined at a rectangular grid of pixels $P \subseteq \mathbb{Z}^d$. For consistency with (3), we extend them by setting $f_\mathbf{i}$=0 outside the image boundaries where $\mathbf{i} \notin P$. Thus, the summation in (3) can be over P rather than $\mathbb{Z}^d$. Notice that the continuous image created by (3) can be nonzero outside the image boundaries at non-integer coordinates.

The following proposition is useful for numerical integration in the frequency domain and can be directly verified from the definition of the Discrete Fourier Transform (DFT).

Proposition 1. *Consider a discrete image $\{f_\mathbf{i}\}_{\mathbf{i}\in P}$ defined over a grid P with n_i pixels along i-th dimension. Define f as in* (3)*. Then the corresponding DFT values of $\{f_\mathbf{i}\}_{\mathbf{i}\in P}$ are equal*[3] *to the samples of $F(\mathbf{z})$, the Fourier transform of $f(\mathbf{x})$, taken at $\frac{1}{n_i T}$ intervals along each dimension i.*

Other interpolation techniques: To accurately perform the sinc interpolation one needs to consider a large neighbourhood of pixels at any certain point. Therefore, in practice, kernels with a bounded support are used instead of the sinc function. A large class of interpolation techniques can be formulated as:

$$\hat{f}(\mathbf{x}) = \sum_{\mathbf{i}\in\mathbb{Z}^d} f_\mathbf{i}\ s\left(\frac{\mathbf{x}}{T} - \mathbf{i}\right), \tag{5}$$

where $s\colon \mathbb{R}^d \to \mathbb{R}$ is the interpolation kernel, which is typically a low-pass filter. To have a consistent interpolation, we must have $s(\mathbf{0}) = 1$, and $s(\mathbf{x}) = 0$ for any $\mathbf{x}$ with integer coordinates. Two examples of a bounded support kernel are:

$$s_\mathrm{n}(\mathbf{x}) = \begin{cases} 1 & \|\mathbf{x}\|_\infty \le \frac{1}{2}, \\ 0 & \text{elsewhere,} \end{cases} \tag{6}$$

which performs *nearest neighbour interpolation*, and

$$s_\mathrm{t}(\mathbf{x}) = s_\mathrm{n}(\mathbf{x}) * s_\mathrm{n}(\mathbf{x}), \tag{7}$$

performing *bilinear (2D) or trilinear (3D) interpolation.*

2) Computation of the target function: Assume that $\{f_\mathbf{i}\}$ and $\{g_\mathbf{i}\}$ are defined on a grid of pixels P. By substituting $\hat{f}$ (and $\hat{g}$) in (5) for f and g in (1), we can write the target function for the discrete image pair $\{f_\mathbf{i}\}$ and $\{g_\mathbf{i}\}$ as follows

$$Q(\mathtt{R}, \mathbf{t}) = \sum_{\mathbf{i}\in P}\sum_{\mathbf{j}\in P} f_\mathbf{i}\, g_\mathbf{j} \int s\left(\frac{\mathbf{x}}{T} - \mathbf{i}\right) s\left(\frac{\mathtt{R}(\mathbf{x}+\mathbf{t})}{T} - \mathbf{j}\right) d\mathbf{x}, \tag{8}$$

$$= T^d \sum_{\mathbf{i}\in P}\sum_{\mathbf{j}\in P} f_\mathbf{i}\, g_\mathbf{j}\ W_\mathtt{R}\left(\mathtt{R}\left(\mathbf{i} + \frac{\mathbf{t}}{T}\right) - \mathbf{j}\right) \tag{9}$$

[1] Here, all integrals are over $\mathbb{R}^d$ unless otherwise specified.

[2] We have assumed that the discrete and continuous coordinates share the same origin. More generally, we can write $f_\mathbf{i} = f(T\,\mathbf{i} - \mathbf{x}_0)$, where $\mathbf{x}_0 \in \mathbb{R}^d$ is the displacement of the origin.

[3] The equality might be up to a known global scaling factor depending on the convention used for defining DFT.

where

$$W_{\mathtt{R}}(\mathbf{d}) = \int s(\mathtt{R}\mathbf{x}+\mathbf{d})\, s(\mathbf{x})\, d\mathbf{x}. \tag{10}$$

Notice that $\frac{\mathbf{t}}{T}$ in (9) is the translation in the pixel coordinates. Now look at $\mathtt{R}(\mathbf{i}+\frac{\mathbf{t}}{T})-\mathbf{j}$. This is equivalent to rigidly transforming $\mathbf{i}$ according to $\mathtt{R}$ and $\frac{\mathbf{t}}{T}$, and then taking the difference with the pixel position $\mathbf{j}$. The weight given to each pair of pixel values $f_{\mathbf{i}}$ and $g_{\mathbf{j}}$ is equal to $W_{\mathtt{R}}(\mathtt{R}(\mathbf{i}+\frac{\mathbf{t}}{T})-\mathbf{j})$. The value of $W_{\mathtt{R}}(\mathbf{d})$ is expected to decay as the vector $\mathbf{d}$ grows in magnitude. If s has bounded support like (6) or (7), then $W_{\mathtt{R}}(\mathbf{d})$ will have a bounded support too. In other cases $W_{\mathtt{R}}(\mathbf{d})$ is negligible for large enough vectors $\mathbf{d}$. Therefore, for each $\mathbf{i}$, we can sum over pixels $\mathbf{j}$ within a certain neighbourhood of $\mathbf{i}$. This means that computing the target (9) needs $O(|P|)$ rather than $O(|P|^2)$ operations, where $|P|$ is the number of pixels[4].

An essential part in computing (9) is to find the weights $W_{\mathtt{R}}(\mathtt{R}(\mathbf{i}+\frac{\mathbf{t}}{T})-\mathbf{j})$. For the nearest neighbour kernel (6), the integral (10) is simply the intersection area of two squares. As for the sinc kernel, a formula for (10) can be calculated in the frequency domain using the Parseval's theorem. Nevertheless, even if large neighbourhoods are avoided by using bounded support kernels, the computation of $W_{\mathtt{R}}(\mathbf{d})$ itself is still costly. One solution is to precompute $W_{\mathtt{R}}(\mathbf{d})$ on a look-up grid of $\mathtt{R}$ and $\mathbf{d}$ values. Most registration algorithms, however, consider a simple form for $W_{\mathtt{R}}(\mathbf{d})$ which does not depend on $\mathtt{R}$ once $\mathbf{d}$ is known. Basically, what they do is discretizing the correlation integral. Let us rewrite (8) as

$$Q(\mathtt{R},\mathbf{t}) = \int \textstyle\sum_{\mathbf{k}\in P} f_{\mathbf{k}}\, s(\frac{\mathbf{x}}{T}-\mathbf{k}) \sum_{\mathbf{j}\in P} g_{\mathbf{j}}\, s(\frac{\mathtt{R}(\mathbf{x}+\mathbf{t})}{T}-\mathbf{j})\, d\mathbf{x}. \tag{11}$$

Now, we discretize the above integral at $\mathbf{x} = T\mathbf{k}$ for all $\mathbf{k} \in P$:

$$\begin{aligned} Q(\mathtt{R},\mathbf{t}) &\approx T^d \sum_{\mathbf{i}\in P} \sum_{\mathbf{k}\in P} f_{\mathbf{k}}\, s(\mathbf{i}-\mathbf{k}) \sum_{\mathbf{j}\in P} g_{\mathbf{j}}\, s\big(\mathtt{R}(\mathbf{k}+\frac{\mathbf{t}}{T})-\mathbf{j}\big)\, d\mathbf{x}, \\ &= T^d \sum_{\mathbf{i}\in P} f_{\mathbf{i}} \sum_{\mathbf{j}\in P} g_{\mathbf{j}}\, s\big(\mathtt{R}(\mathbf{i}+\frac{\mathbf{t}}{T})-\mathbf{j}\big)\, d\mathbf{x}. \end{aligned} \tag{12}$$

Here, the weights are simply the kernel values $s(\mathtt{R}(\mathbf{i}+\frac{\mathbf{t}}{T})-\mathbf{j})$ which do not depend on $\mathtt{R}$ if $\mathtt{R}(\mathbf{i}+\frac{\mathbf{t}}{T})-\mathbf{j}$ is known. Another observation is that (12) is independent of what kernel is used for interpolating $\{f_{\mathbf{i}}\}$ in the case where $\{f_{\mathbf{i}}\}$ and $\{g_{\mathbf{i}}\}$ are interpolated using two different kernels. For the nearest neighbour interpolation, $\sum_{\mathbf{j}\in P} g_{\mathbf{j}}\, s(\mathtt{R}(\mathbf{i}+\frac{\mathbf{t}}{T})-\mathbf{j})$ is simply the intensity value of the closest pixel $\mathbf{j}$ to $\mathtt{R}(\mathbf{i}+\frac{\mathbf{t}}{T})$. For the sinc interpolation, this value may be approximated efficiently by looking up in an upsampled version of $\{g_{\mathbf{j}}\}$. In our experiments we observed that discretized computation of the correlation integral does not cause major problems, unless for extremely decimated images or when a very high accuracy is expected (e.g. less than .2 pixels for translation). However, one ideally needs to further consider the discretization error.

III. Implications of low-resolution registration

Assume that the discrete images $\{f_{\mathbf{i}}\}$ and $\{g_{\mathbf{i}}\}$ are decimated to low-resolution images $\{f^l_{\mathbf{i}}\}$ and $\{g^l_{\mathbf{i}}\}$. One may ask the question "What can the fitness (9) computed for lower resolution images say about the fitness for higher resolution images?". This is an important question since the target function can be computed much faster in low resolution. If the images are decimated by a factor of m, then computing the target requires m^d times fewer operations, where $d = 2, 3$ for 2D and 3D images respectively. This fact is clear from (9), where it is shown that the number of computations is proportional to the number of pixels[5].

Decimation of a discrete image $\{f_{\mathbf{i}}\}$ may be carried out by

1) low-pass filtering the corresponding continuous image f obtained from (3), and
2) sampling the filtered images at a period of mT, where T is the sampling period of $\{f_{\mathbf{i}}\}$.

The low-pass filter handles the aliasing distortion caused by downsampling, and thus, must have a cutoff frequency of $\frac{1}{2mT}$ or less along every direction. To simplify the calculations, we consider a radial filter with the ideal frequency response

$$L(\mathbf{z}) = \begin{cases} 1 & \|\mathbf{z}\| \le \frac{1}{2mT}, \\ 0 & \text{otherwise}, \end{cases} \tag{13}$$

where $\|\mathbf{z}\|$ is the l^2-norm (length) of $\mathbf{z}$. The filtered image f^l can be obtained as $f^l = l * f$ where l is the impulse response of L and "$*$" is the convolution operator. This filter eliminates all the frequency components outside a ball of radius $\frac{1}{2mT}$ around the origin. Therefore, nothing is lost by sampling the filtered image f^l at intervals mT to obtain $\{f^l_{\mathbf{i}}\}$, the low resolution image. We can also define a complementary high-pass filter

$$H(\mathbf{z}) = \begin{cases} 0 & \|\mathbf{z}\| \le \frac{1}{2mT}, \\ 1 & \|\mathbf{z}\| \ge \frac{1}{2mT},\ \ \|\mathbf{z}\|_\infty \le \frac{1}{2T}. \end{cases} \tag{14}$$

The value of $H(\mathbf{z})$ for $\|\mathbf{z}\|_\infty > \frac{1}{2T}$ do not matter, as H is applied to f, g which are bandlimited to $\frac{1}{2T}$ due to (3). One could filter f with H to obtain f^h. Notice that $f(\mathbf{x}) = f^l(\mathbf{x}) + f^h(\mathbf{x})$. To get a discrete image $\{f^h_{\mathbf{i}}\}$ we sample f^h at intervals T. Therefore, $\{f^h_{\mathbf{i}}\}$ has the same size as $\{f_{\mathbf{i}}\}$ while $\{f^l_{\mathbf{i}}\}$ has roughly $1/m$ less samples in every direction[6]. The corresponding continuous images of $\{f^l_{\mathbf{i}}\}$ and $\{f^h_{\mathbf{i}}\}$ computed from (3) are exactly equal to f^l and f^h respectively. Note that to use (3) on $\{f^l_{\mathbf{i}}\}$ one must use mT instead of T.

Now, let us have a closer look at the continuous images f^l, f^h, g^l and g^h. An important observation is that f^l and $g^h_{\mathtt{R},\mathbf{t}} : \mathbf{x} \mapsto g^h(\mathtt{R}(\mathbf{x}+\mathbf{t}))$ are orthogonal for any choice of $\mathtt{R}$ and $\mathbf{t}$. This is because f^l has no frequency component outside the ball of radius $\frac{1}{2mT}$, while $g^h_{\mathtt{R},\mathbf{t}}$ has no frequency components inside this ball. One way to see this is to write

[4]We use worst case complexity because in practice only the pixels corresponding to the intersecting area between the two images is computed.

[5]In fact, the double summation in (9) implies that the number of computations is proportional to the square of the number of pixels. But, in practice, we only consider $\mathbf{j}$-s within a fixed neighbourhood of each $\mathbf{i}$.

[6]We do not sample outside the boundaries of $f_{\mathbf{i}}$ for creating $f^l_{\mathbf{i}}$ and $f^h_{\mathbf{i}}$, even though the samples might not be exactly zero after filtering. The loss is supposedly negligible given that the images have a dark (near zero) margin.

the inner product (correlation) between these two functions in the frequency domain using the Parseval's theorem:

$$Q(\mathtt{R},\mathbf{t}) = \int f^l(\mathbf{x})\, g^h\big(\mathtt{R}(\mathbf{x}{+}\mathbf{t})\big)\, d\mathbf{x} \tag{15}$$

$$= \int \overline{F^l(\mathbf{z})}\, G^h(\mathtt{R}\mathbf{z})\, e^{2\pi i \mathbf{t}^T\mathbf{z}}\, d\mathbf{z} = 0. \tag{16}$$

Similarly, f^h and $g^l_{\mathtt{R},\mathbf{t}}\colon \mathbf{x} \mapsto g^l\big(\mathtt{R}(\mathbf{x}{+}\mathbf{t})\big)$ are orthogonal. Thus

$$\begin{aligned} Q(\mathtt{R},\mathbf{t}) &= \int f(\mathbf{x})\, g\big(\mathtt{R}(\mathbf{x}{+}\mathbf{t})\big)\, d\mathbf{x} \\ &= \int f^l(\mathbf{x})\, g^l\big(\mathtt{R}(\mathbf{x}{+}\mathbf{t})\big)\, d\mathbf{x} + \int f^h(\mathbf{x})\, g^h\big(\mathtt{R}(\mathbf{x}{+}\mathbf{t})\big)\, d\mathbf{x} \\ &= Q^l(\mathtt{R},\mathbf{t}) + \int f^h(\mathbf{x})\, g^h\big(\mathtt{R}(\mathbf{x}{+}\mathbf{t})\big)\, d\mathbf{x}, \end{aligned} \tag{17}$$

where $Q^l(\mathtt{R},\mathbf{t})$ is the target value for f^l and g^l. Therefore

$$\big|Q(\mathtt{R},\mathbf{t}) - Q^l(\mathtt{R},\mathbf{t})\big| = \left|\int f^h(\mathbf{x})\, g^h\big(\mathtt{R}(\mathbf{x}{+}\mathbf{t})\big)\, d\mathbf{x}\right| \tag{18}$$

$$\leq \sqrt{\int \big|f^h(\mathbf{x})\big|^2 d\mathbf{x}}\, \sqrt{\int \big|g^h(\mathbf{x})\big|^2 d\mathbf{x}} = \big\|f^h\big\|\, \big\|g^h\big\|, \tag{19}$$

where $\|f\|$ represents the L^2-norm of f. We obtain (19) using the Cauchy-Schwarz inequality followed by a change of variables $\mathbf{x} \leftarrow \mathtt{R}(\mathbf{x}{+}\mathbf{t})$ for the right square root. It follows

$$Q^l(\mathtt{R},\mathbf{t}) - E^h_{fg} \leq Q(\mathtt{R},\mathbf{t}) \leq Q^l(\mathtt{R},\mathbf{t}) + E^h_{fg} \tag{20}$$

where $E^h_{fg} = \big\|f^h\big\|\, \big\|g^h\big\|$. The idea here is that E^h_{fg} tends to be small as for natural images the energy is mostly concentrated in the lower frequency bands.

Now, suppose that we want to find the maximum of $Q(\mathtt{R},\mathbf{t})$ over a grid $\{\mathtt{R}_k,\mathbf{t}_k\}$ of registration parameters. Assume that we have computed $Q^l(\mathtt{R},\mathbf{t})$ for the lower resolution images over the whole grid. Represent respectively by $\mathtt{R}^{l*},\mathbf{t}^{l*}$ and $\mathtt{R}^*,\mathbf{t}^*$ the maximizers of $Q^l(\mathtt{R},\mathbf{t})$ and $Q(\mathtt{R},\mathbf{t})$ over the grid, where $\mathtt{R}^*,\mathbf{t}^*$ are yet unknown. Using (20) we get

$$Q(\mathtt{R}^*,\mathbf{t}^*) \geq Q(\mathtt{R}^{l*},\mathbf{t}^{l*}) \geq Q^l(\mathtt{R}^{l*},\mathbf{t}^{l*}) - E^h_{fg}. \tag{21}$$

It means that we could rule out any $\mathtt{R},\mathbf{t}$ for which $Q^l(\mathtt{R},\mathbf{t}) < Q^l(\mathtt{R}^{l*},\mathbf{t}^{l*}) - 2E^h_{fg}$, since in that case

$$\begin{aligned} Q(\mathtt{R},\mathbf{t}) &\leq Q^l(\mathtt{R},\mathbf{t}) + E^h_{fg} \\ &< Q^l(\mathtt{R}^{l*},\mathbf{t}^{l*}) - E^h_{fg} \leq Q(\mathtt{R}^*,\mathbf{t}^*) \end{aligned} \tag{22}$$

Fig. 1(c) illustrates this idea. To further limit the search space one can compute the high resolution target function Q at $\mathtt{R}^{l*},\mathbf{t}^{l*}$ and discard all $\mathtt{R},\mathbf{t}$ with $Q^l(\mathtt{R},\mathbf{t}) < Q(\mathtt{R}^{l*},\mathbf{t}^{l*}) - E^h_{fg}$, as shown in Fig. 1(d). The example provided in Fig. 1 shows the effectiveness of the proposed bound even when the images are decimated by a factor of 16 in each dimension.

A. Bounded support interpolation

The results in the previous section hold when the sinc kernel (4) is used to compute the target functions (9) or (12). In this case, $s(\mathbf{d})$ and $W_{\mathtt{R}}(\mathbf{d})$ do not have a compact support, and for (20) to hold we need a large neighbourhood of pixels $\mathbf{j}$ around the location of each transformed pixel $\mathtt{R}(\mathbf{i}{+}\frac{\mathbf{t}}{T})$, which requires a lot of computations. We now consider computing the target function (9) with two kernels of bounded support, namely the box kernel (6) and the triangular kernel (7) which respectively correspond to nearest neighbour and bilinear (or trilinear) interpolation methods. Many other interpolation techniques can be treated in a similar manner.

Here, we assume that the low-resolution images are obtained, as before, by performing the ideal filter (13) on the continuous images f and g obtained by the *sinc* kernel using (3), and then resampling. However, for computing the target function using (9), we use a box or triangular kernel for both low-resolution and high-resolution images. Now, let us see what happens in the frequency domain. The interpolation formula (5) can be written as the following convolution

$$\hat{f}(\mathbf{x}) = s\big(\frac{\mathbf{x}}{T}\big) * \Big(f(\mathbf{x}) \cdot \sum_{\mathbf{i}\in\mathbb{Z}^d} \delta(\mathbf{x} - T\mathbf{i})\Big), \tag{23}$$

where δ is the Dirac delta distribution. The Fourier transform of $s(\mathbf{x})$ is $\mathrm{sinc}^\alpha(\mathbf{z})$ with $\alpha = 1, 2$ for the box kernel (6) and the triangular kernel (7) respectively. The Fourier transform of (23) is then

$$\hat{F}(\mathbf{z}) = \mathrm{sinc}^\alpha(T\mathbf{z}) \cdot \sum_{\mathbf{i}\in\mathbb{Z}^d} F\big(\mathbf{z} - \frac{\mathbf{i}}{T}\big). \tag{24}$$

Notice that $F(\mathbf{z})$ is bandlimited to $\frac{1}{2T}$ in every dimension. Therefore, the term $\sum_{\mathbf{i}\in\mathbb{Z}^d} F(\mathbf{z} - \frac{\mathbf{i}}{T})$ is just the periodic repetition of $F(\mathbf{z})$ with period $\frac{1}{T}$ along every dimension. In the same way, for the low-resolution image $\{f^l_{\mathbf{i}}\}$ we have

$$\hat{F}^l(\mathbf{z}) = \mathrm{sinc}^\alpha(mT\mathbf{z}) \cdot \sum_{\mathbf{i}\in\mathbb{Z}^d} F^l\big(\mathbf{z} - \frac{\mathbf{i}}{mT}\big). \tag{25}$$

Notice that $F^l(\mathbf{z}) = F(\mathbf{z}) \cdot L(\mathbf{z})$ is bandlimited to $\frac{1}{2mT}$ along every dimension. Similarly, $\hat{G}(\mathbf{z})$ and $\hat{G}^l(\mathbf{z})$ can be obtained.

Now, we bound $\left|\hat{Q}(\mathtt{R},\mathbf{t}) - \hat{Q}^l(\mathtt{R},\mathbf{t})\right|$, where $\hat{Q}(\mathtt{R},\mathbf{t})$ is the target value (1) calculated for $\hat{f}$ and $\hat{g}$, and $\hat{Q}^l(\mathtt{R},\mathbf{t})$ is the target for $\hat{f}^l$ and $\hat{g}^l$. Here, we present a simple bound, and leave more elaborate bounds for future research. Define the energy of a function F within the frequency region S by

$$E_S(F) = \int_S |F(\mathbf{z})|^2 d\mathbf{z}, \tag{26}$$

denote by Ω the ball of radius $\frac{1}{2mT}$ around the origin, and let $\bar{\Omega}$ be its complement. Then we have the following proposition:

Proposition 2. *The absolute difference on the target values* $\left|\hat{Q}(\mathtt{R},\mathbf{t}) - \hat{Q}^l(\mathtt{R},\mathbf{t})\right|$ *is bounded from above by*

$$\sqrt{E_\Omega(\hat{F})\, E_\Omega(\hat{G}{-}\hat{G}^l)} + \sqrt{E_\Omega(\hat{G})\, E_\Omega(\hat{F}{-}\hat{F}^l)} \tag{27}$$

$$+\sqrt{E_\Omega(\hat{F}{-}\hat{F}^l)\, E_\Omega(\hat{G}{-}\hat{G}^l)} \tag{28}$$

$$+\sqrt{E_{\bar{\Omega}}(\hat{F})\, E_{\bar{\Omega}}(\hat{G})} + \sqrt{E_{\bar{\Omega}}(\hat{F}^l)\, E_{\bar{\Omega}}(\hat{G}^l)} \tag{29}$$

Notice that outside the frequency ball Ω one expects $\hat{F},\hat{G},\hat{F}^l$ and $\hat{G}^l$ to have low energy and hence (29) is supposedly small. As for (27) and (28), it is expected that within Ω, the energies $E_\Omega(\hat{F}{-}\hat{F}^l)$ and $E_\Omega(\hat{G}{-}\hat{G}^l)$ are small,

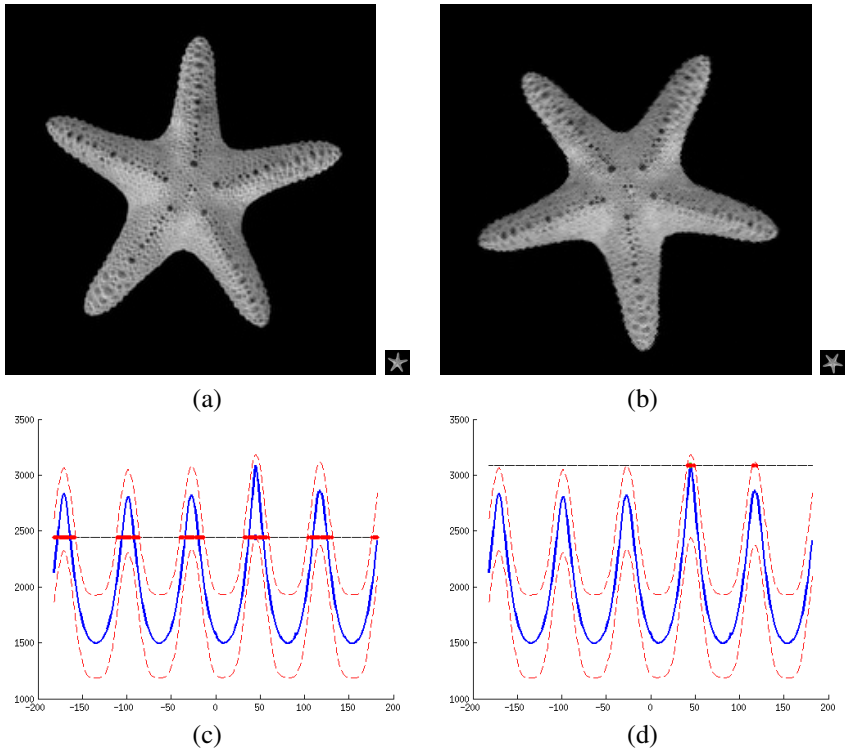


Fig. 1. Bounds on high resolution registration fitness based on low resolution registration. (a) A starfish image (obtained from http://www.ck12.org) is chosen as the reference image $\{f_{\mathbf{i}}\}$, as it exhibits multiple local maxima, (b) $\{f_{\mathbf{i}}\}$ is rotated by 45 degrees around its centre and then undergoes a slight affine warp to produce $\{g_{\mathbf{i}}\}$. The only registration parameter is the rotation angle θ. Both images are decimated by a factor of $m = 16$ to create $\{f^l_{\mathbf{i}}\}$ and $\{g^l_{\mathbf{i}}\}$ (tiny images on the bottom-right of each image). (c) the lower and upper bounds $Q^l(\theta) \pm E^h_{fg}$ (red dashed graphs) envelop the high resolution fitness $Q(\theta) = Q(\mathtt{R}, \mathbf{t})$ (blue solid graph). The horizontal line shows the value $Q^l(\mathtt{R}^{l*}, \mathbf{t}^{l*}) - E^h_{fg}$. The red thick parts of this line show the reduced search space, where we have $Q^l(\mathtt{R}, \mathbf{t}) + E^h_{fg} \geq Q^l(\mathtt{R}^{l*}, \mathbf{t}^{l*}) - E^h_{fg}$ or $Q^l(\mathtt{R}, \mathbf{t}) \geq Q^l(\mathtt{R}^{l*}, \mathbf{t}^{l*}) - 2E^h_{fg}$. (d) to further restrict the search space the high resolution fitness Q is computed at $(\mathtt{R}^{l*}, \mathbf{t}^{l*}) \equiv \theta^{l*}$. The horizontal line shows the value $Q(\mathtt{R}^{l*}, \mathbf{t}^{l*})$. The red thick parts of this line shows the reduced search space, namely where $Q^l(\mathtt{R}, \mathbf{t}) + E^h_{fg} \geq Q(\mathtt{R}^{l*}, \mathbf{t}^{l*})$. Notice that decimating by a factor of 16 reduces the computations by a factor of 256 for 2D images. The amount of reduction in the search space provided by such a low resolution registration is notable.

as we will soon see. The proof is is given in Appendix A. Here are the value of the energy terms E_Ω calculated for F.

$$E_\Omega(\hat{F}) = \int_\Omega |\mathrm{sinc}^\alpha(T\mathbf{z})|^2 \, |F(\mathbf{z})|^2 d\mathbf{z}, \tag{30}$$

$$E_\Omega(\hat{F}-\hat{F}^l) = \int_\Omega |\mathrm{sinc}^\alpha(T\mathbf{z}) - \mathrm{sinc}^\alpha(mT\,\mathbf{z})|^2 |F(\mathbf{z})|^2 d\mathbf{z} \tag{31}$$

$$E_{\bar{\Omega}}(\hat{F}) = \int_{C\setminus\Omega} \mathrm{sinc}^{2\alpha}(T\mathbf{z}) \, |F(\mathbf{z})|^2 d\mathbf{z} \tag{32}$$

$$+ \int_C \big(\Phi_\alpha(T\mathbf{z}) - \mathrm{sinc}^{2\alpha}(T\mathbf{z})\big) \, |F(\mathbf{z})|^2 d\mathbf{z}, \tag{33}$$

$$E_{\bar{\Omega}}(\hat{F}^l) = \int_\Omega \big(\Phi_\alpha(mT\mathbf{z}) - \mathrm{sinc}^{2\alpha}(mT\mathbf{z})\big) \, |F(\mathbf{z})|^2 d\mathbf{z}, \tag{34}$$

where C is the l^∞-ball of radius $\frac{1}{2T}$, that is $\{\mathbf{z} \,|\, |z_i| \leq \frac{1}{2T}\}$, and the function Φ_α is defined on the l^∞-ball of radius $\frac{1}{2}$ as

$$\Phi_\alpha(\mathbf{z}) = \sum_{\mathbf{i}\in\mathbb{Z}^d} \mathrm{sinc}^{2\alpha}(\mathbf{z}+\mathbf{i}). \tag{35}$$

For the 1D case ($d = 1$) with $-\frac{1}{2} \leq z \leq \frac{1}{2}$ we have

$$\Phi_\alpha(z) = \mathrm{sinc}^{2\alpha}(z)\big(1 + \frac{\psi_{2\alpha-1}(1+z) + \psi_{2\alpha-1}(1-z)}{z^{-2\alpha}(2\alpha-1)!}\big), \tag{36}$$

where ψ_k is the polygamma function of order k. For $d > 1$ we have $\Phi_\alpha(\mathbf{z}) = \prod_{i=1}^d \Phi_\alpha(z_i)$. Considering Proposition 1 the integrals (30-34) can be computed numerically using DFT.

It is simple to derive (30) and (31) from (24) and (25) considering the fact that $\sum_{\mathbf{i}\in\mathbb{Z}^d} F(\mathbf{z}-\frac{\mathbf{i}}{T})$ and $\sum_{\mathbf{i}\in\mathbb{Z}^d} F^l(\mathbf{z}-\frac{\mathbf{i}}{mT})$ are both equal to $F(\mathbf{z})$ inside Ω. Notice that (31) tends to be relatively small for natural images, since for $\mathbf{z}$ with a small magnitude the difference $|\mathrm{sinc}^\alpha(T\mathbf{z}) - \mathrm{sinc}^\alpha(mT\,\mathbf{z})|$ is small and for larger $\mathbf{z}$ the spectrum $F(\mathbf{z})$ tends to be small. With a similar argument one can assert that the integrals (33) and (34) are small. The integral (32) is small as it is over $C \setminus \Omega$. To obtain (32) and (33) observe that $E_{\bar{\Omega}}(\hat{F})$ is equal to

$$\int_{C\setminus\Omega} |\hat{F}(\mathbf{z})|^2 d\mathbf{z} + \int_{\bar{C}} |\hat{F}(\mathbf{z})|^2 d\mathbf{z} \tag{37}$$

As $\hat{F}(\mathbf{z}) = \mathrm{sinc}^\alpha(T\,\mathbf{z})F(\mathbf{z})$ for all $\mathbf{z} \in C$, the integral over $C\setminus\Omega$ is equal to (32). For the integral over $\bar{C}$, first for a vector $\mathbf{v}$ and a set S define $\mathbf{v}+S \stackrel{\text{def}}{=} \{\mathbf{v}+\mathbf{z} \,|\, \mathbf{z} \in S\}$. Then, we have

$$\begin{aligned}\int_{\bar{C}} |\hat{F}(\mathbf{z})|^2 d\mathbf{z} &= \sum_{\mathbf{i}\in\mathbb{Z}^d\setminus\mathbf{0}} \int_{\mathbf{i}/T+C} |\hat{F}(\mathbf{z})|^2 d\mathbf{z} \\ &= \sum_{\mathbf{i}\in\mathbb{Z}^d\setminus\mathbf{0}} \int_C |\hat{F}(\mathbf{z}+\mathbf{i}/T)|^2 d\mathbf{z} \\ &= \sum_{\mathbf{i}\in\mathbb{Z}^d\setminus\mathbf{0}} \int_C |F(\mathbf{z})|^2 \mathrm{sinc}^2(T\mathbf{z}+\mathbf{i}) d\mathbf{z}, \end{aligned} \tag{38}$$

which is equal to (33) as $\sum_{\mathbf{i}\in\mathbb{Z}^d\setminus\mathbf{0}}\operatorname{sinc}^2(T\mathbf{z}+\mathbf{i})=\Phi_\alpha(T\mathbf{z})-\operatorname{sinc}^{2\alpha}(T\mathbf{z})$. In a similar way (34) can be obtained; only the period is $\frac{1}{mT}$ instead of $\frac{1}{T}$. This means that, instead of C, we must consider the l^∞ ball C' of radius $\frac{1}{mT}$. Notice that in this case $\int_{C'\setminus\Omega}|\hat{F}^l(\mathbf{z})|^2d\mathbf{z}$ is zero, as $F^l(\mathbf{z})=F(\mathbf{z})\cdot L(\mathbf{z})$ is zero outside Ω. All the integrals can be numerically computed using DFT in the light of Proposition 1. Fig. 2(a-c) illustrates the effect of the bound (27-29) for the example of Fig. 1.

Finally, we consider the case where the discretized integration (12) is used for computing the target function. As discussed earlier, this approximation does not depend on how $\{f_\mathbf{i}\}$ is interpolated. Thus, we assume that $\{f_\mathbf{i}\}$ is interpolated with a sinc kernel, giving $\hat{F}{=}F$ and $\hat{F}^l{=}F^l$. Now, one can show that $\left|\hat{Q}(\mathtt{R},\mathbf{t})-\hat{Q}^l(\mathtt{R},\mathbf{t})\right|$ is bounded from above by

$$\sqrt{E_\Omega(F)\,E_\Omega(\hat{G}{-}\hat{G}^l)}+\sqrt{E_{\bar{\Omega}}(F)\,E_{\Omega'\setminus\Omega}(\hat{G})} \qquad (39)$$

where Ω' is the ball of radius $\frac{\sqrt{d}}{2T}$ (see Appendix B). The effect of using this bound is depicted in Fig. (2)(d,e,f). The bound is generally smaller than (27-29). However, here the discretization error for computing the target integral has been neglected. This is evident from the slight fluctuations in the bounds in Fig. (2)(d,e) for very low resolutions.

B. Low-resolution to high-resolution registration

In this section we examine the situation where only one of the images $\{f_\mathbf{i}\}$ is decimated. We study how this affects the amount of computations and the quality of the bounds. We also consider the case where the other image $\{g_\mathbf{i}\}$ is upsampled. Let $\{f^l_\mathbf{i}\}$ be the discrete image $\{f_\mathbf{i}\}$ decimated by a factor of m. Using (5) a corresponding continuous image is

$$\hat{f}^l(\mathbf{x})=\sum_{\mathbf{i}\in P^l}f^l_\mathbf{i}\;s\big(\frac{\mathbf{x}}{mT}-\mathbf{i}\big), \qquad (40)$$

where P^l is the grid of low-resolution pixels. For the image $\{g_\mathbf{i}\}$ we do not change the resolution, and thus, we have $\hat{g}(\mathbf{x})=\sum_{\mathbf{i}\in P}g_\mathbf{j}\;s(\frac{\mathbf{x}}{T}-\mathbf{j})$. Now, there are two ways to compute the correlation target between $\hat{f}^l(\mathbf{x})$ and $\hat{g}(\mathtt{R}(\mathbf{x}{+}\mathbf{t}))$: the exact way like (9) and the approximate way by discretizing the integral like (12). We leave it to the reader to check that for computing the exact integral similar to (9), lowering the resolution of just one image does not significantly reduce the computations. This is because while the number of pixels in the first image is reduced by a factor of m^d, the size of the neighbourhood around each transformed pixel required for the computation of $W_\mathtt{R}(\mathbf{d})$ is increased by about the same factor. However, if we discretize the integral, like in (12) we get

$$\hat{Q}^{hl}(\mathtt{R},\mathbf{t})=\int\hat{f}^l(\mathbf{x})\,\hat{g}\big(\mathtt{R}(\mathbf{x}+\mathbf{t})\big)\,d\mathbf{x} \qquad (41)$$

$$\approx(mT)^d\sum_{\mathbf{i}\in P^l}f_\mathbf{i}\sum_{\mathbf{j}\in P}g_\mathbf{j}\,s\big(\mathtt{R}(m\mathbf{i}+\frac{\mathbf{t}}{T})-\mathbf{j}\big)\,d\mathbf{x}. \qquad (42)$$

The above shows that if s has a compact support then for every $\mathbf{i}$ we only need to consider those pixels $\mathbf{j}$ which are in a neighbourhood of $\mathtt{R}(m\mathbf{i}{+}\frac{\mathbf{t}}{T})$ whose size is the same as that of (12). Since the first sum is over the low-resolution pixel grid P^l which has about m^d times fewer pixels than P, the number of computations is about the same as when both images are decimated. This, of course, comes at the cost of having an approximate integral by discretization.

Now, let us see what happens to the bounds. If the interpolation is done using the sinc kernel ($s=\operatorname{sinc}$, and thus $\hat{f}^l{=}f^l$ and $\hat{g}{=}g$), then the target function (41) is equal to $Q^l(\mathtt{R},\mathbf{t}){=}\int f^l(\mathbf{x})\,g^l(\mathtt{R}(\mathbf{x}+\mathbf{t}))\,d\mathbf{x}$, and hence, the bound (20) does not change. This is due to the fact that f^l and $g^h_{\mathtt{R},\mathbf{t}}:\mathbf{x}\mapsto g^h(\mathtt{R}(\mathbf{x}{+}\mathbf{t}))$ are orthogonal, and thus, the correlation between $f^l(\mathbf{x})$ and $g(\mathtt{R}(\mathbf{x}{+}\mathbf{t}))=g^l(\mathtt{R}(\mathbf{x}{+}\mathbf{t}))+g^h(\mathtt{R}(\mathbf{x}{+}\mathbf{t}))$ is the same as the correlation between $f^l(\mathbf{x})$ and $g^l(\mathtt{R}(\mathbf{x}+\mathbf{t}))$. However, for other interpolation techniques the bounds can be further tightened. In a similar way to Proposition 2 we can obtain

$$\left|\hat{Q}(\mathtt{R},\mathbf{t})-\hat{Q}^{hl}(\mathtt{R},\mathbf{t})\right|\le\sqrt{E_\Omega(\hat{G})\,E_\Omega(\hat{F}{-}\hat{F}^l)}+\sqrt{E_{\bar{\Omega}}(\hat{F})\,E_{\bar{\Omega}}(\hat{G})}+\sqrt{E_{\bar{\Omega}}(\hat{F}^l)\,E_{\bar{\Omega}}(\hat{G})}, \qquad (43)$$

which is generally smaller than the bound in Proposition 2. If the discretized integral is used, similar to the way (39) is obtained, one can obtain the following bound

$$\sqrt{E_{\bar{\Omega}}(F)\,E_{\Omega'\setminus\bar{\Omega}}(\hat{G})}, \qquad (44)$$

which is obtained by replacing $\hat{G}^l$ with $\hat{G}$ in (39). Fig. 3(a) shows an example of applying this bound.

Now, let us see what happens if the second image $\{g_\mathbf{i}\}$ is upsampled by a factor of $p\ge2$ to obtain $\{g^u_\mathbf{i}\}$. We assume that the new samples are obtained by the natural sinc interpolation. Therefore, the corresponding continuous function remains the same, which means $g^u(\mathbf{x})=g(\mathbf{x})$. Similarly to the way we obtained (24), if $\{g^u_\mathbf{i}\}$ is interpolated by the nearest neighbour ($\alpha{=}1$) or the bilinear ($\alpha{=}2$) method to obtain the interpolated image $\hat{g}^u$, the corresponding Fourier transform will be

$$\hat{G}^u(\mathbf{z})=\operatorname{sinc}^\alpha\Big(\frac{T\mathbf{z}}{p}\Big)\cdot\sum_{\mathbf{i}\in\mathbb{Z}^d}G\Big(\mathbf{z}-\frac{p\mathbf{i}}{T}\Big). \qquad (45)$$

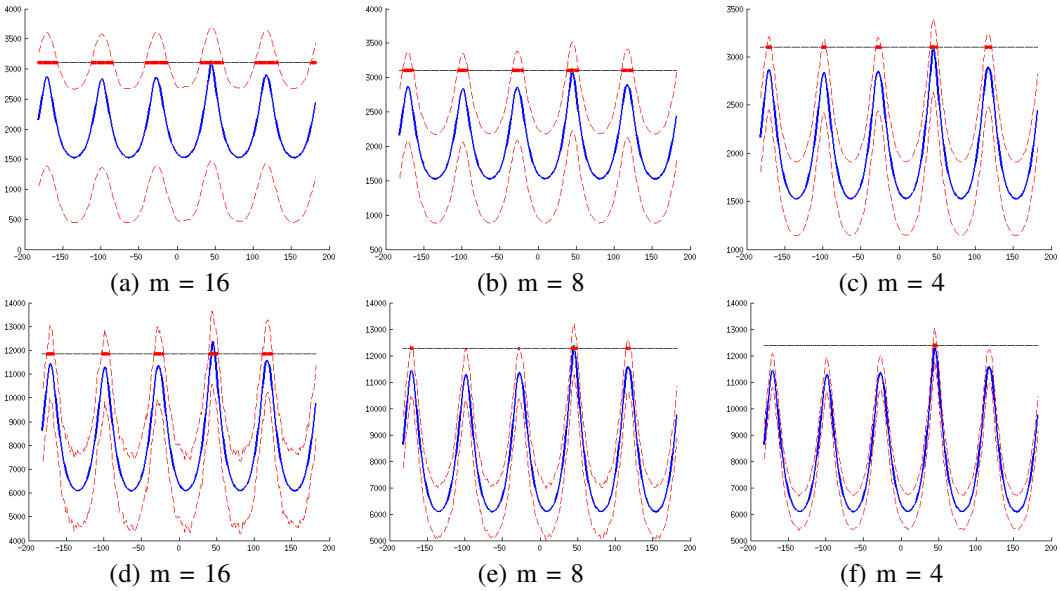
Notice that $\sum_{\mathbf{i}\in\mathbb{Z}^d}G(\mathbf{z}-\frac{p\mathbf{i}}{T})$ is the periodic repetition of $G(\mathbf{z})$ with period p/T. Since $G(\mathbf{z})$ is bandlimited to $\frac{1}{2T}$ in every dimension, this means that for $p\ge2$ there exist a lot of *empty spaces* in the spectrum (45) in which $\hat{G}^u(\mathbf{z})=0$. Specially, $\hat{G}^u(\mathbf{z})$ is zero when $\frac{1}{2T}<\|\mathbf{z}\|_\infty<\frac{p}{2T}$. Thus, $\hat{G}^u(\mathbf{z}){=}0$ inside $\Omega'\setminus C$, where C is the l^∞-ball of radius $\frac{1}{2T}$, and $E_{\Omega'\setminus\Omega}(\hat{G}^u)=E_{C\setminus\Omega}(\hat{G}^u)$. Now, if we derive (44) for $\hat{G}^u(\mathbf{z})$ instead of $\hat{G}(\mathbf{z})$ we will get the following bound

$$\sqrt{E_{\bar{\Omega}}(F)\,E_{C\setminus\Omega}(\hat{G}^u)}. \qquad (46)$$

Notice that for all $\mathbf{z}\in C$ we have

$$\hat{G}^u(\mathbf{z})=\operatorname{sinc}^\alpha\big(T\mathbf{z}/p\big)\,G(\mathbf{z}). \qquad (47)$$

Thus, inside C, $\hat{G}^u(\mathbf{z})$ approaches to $G(\mathbf{z})$ as p gets larger. This means that as p increases, (46) gets closer to the original bound E^h_{fg} in (19), since $E_{\bar{\Omega}}(F)=\int_{\bar{\Omega}}|F(\mathbf{z})|^2d\mathbf{z}=\int\big(f^h(\mathbf{x})\big)^2\,d\mathbf{x}$. Nonetheless, one should bear in mind that here we have neglected the discretization error for computing the



(a) m = 16 (b) m = 8 (c) m = 4

(d) m = 16 (e) m = 8 (f) m = 4

Fig. 2. The bounds obtained for the nearest neighbour interpolation for the registration problem given in Fig. 1. Images are decimated by a factor of (a) 16, (b) 8 and (c) 4. The search space reduction approach is the same as that of Fig. 1(d). The bounds are not as effective as that of the sinc interpolation, but still are useful. (d,e,f) the same figures, but this time discretized integration is used instead of exact integration to compute the target function, and (39) is used for the bounds. Notice that the discretized integration has resulted in fluctuations in the bounds which increase as the decimation rate m gets larger.

target integral[7]. Fig. 3(b,c) shows examples of using (46) for $p = 2, 4$. We have observed experimentally that upsampling by a factor larger than 2 does not help much with reducing the search space. We, therefore, use $p = 2$ in all our experiments.

C. Obtaining Tighter bounds

We can obtain slightly tighter bounds by dividing the high-frequency area into radial bands. By applying the Parseval's theorem to (18), and noticing that $F^h(\mathbf{z}){=}H(\mathbf{z})\,F(\mathbf{z})$ and $G^h(\mathbf{z}){=}H(\mathbf{z})\,G(\mathbf{z})$ where H was defined in (14), we get

$$|Q(\mathtt{R},\mathbf{t}) - Q^l(\mathtt{R},\mathbf{t})| = \left| \int_{\bar{\Omega}} F(\mathbf{z})\, G\big(\mathtt{R}\mathbf{z}\big)\, e^{2\pi i \mathbf{t}^T \mathbf{x}}\, d\mathbf{z} \right| \qquad (48)$$

where $\bar{\Omega}$ is the area outside the ball of radius $\frac{1}{2mT}$ around the origin. Now, we partition $\bar{\Omega}$ to radial bands $\Omega_1, \Omega_2, \ldots, \Omega_P$ as illustrated in Fig. 4. More precisely, $\Omega_i = \{\mathbf{z} \,|\, r_i \le \|\mathbf{z}\| < r_{i+1}\}$ for radii $r_1 < r_2 < \cdots < r_{P+1}$. Then we have

$$\begin{aligned} |Q(\mathtt{R},\mathbf{t}) - Q^l(\mathtt{R},\mathbf{t})| &= \left| \sum_{k=1}^{P} \int_{\Omega_k} F(\mathbf{z})\, G\big(\mathtt{R}\mathbf{z}\big)\, e^{2\pi i \mathbf{t}^T \mathbf{x}}\, d\mathbf{z} \right| \\ &\le \sum_{k=1}^{P} \sqrt{\textstyle\int_{\Omega_k} |F(\mathbf{z})|^2 d\mathbf{z}} \sqrt{\textstyle\int_{\Omega_k} |G(\mathbf{z})|^2 d\mathbf{z}} \qquad (49) \end{aligned}$$

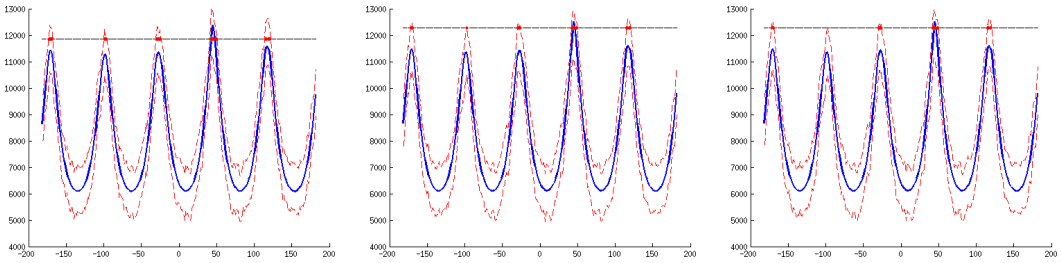
where (49) is obtained by using the triangle inequality followed by the Cauchy-Schwarz inequality, and a change of variables $\mathbf{z} \leftarrow \mathtt{R}\mathbf{z}$ in the second integral. Notice that $\mathtt{R}\mathbf{z} \in \Omega_k$ if and only if $\mathbf{z} \in \Omega_k$. The bound (49) can be computed numerically using DFT (see Proposition 1). We have tighter bounds if more frequency bands are used. The extreme case is when each band only contains DFT samples with the same distance to the origin. On the other extreme, when $P = 1$, then (49) is reduced to (19). Our experiments show that for natural image pairs (49) is only slightly better than (19). For instance, for Fig. 1, the bound (19) is equal to 372.6 while (49) is 356.4 in the best case (maximum P). Nevertheless, dividing the frequency domain into radial bands can be a useful trick in similar problems, like obtaining Lipschitz bounds (Sect. V-A).

[7]The reader might have noticed that the bound (46) is actually smaller than (19). This, however, this does not make it a better bound, since the corresponding target functions are different.

IV. A BASIC ALGORITHM

We showed that the search space for high-resolution images can be limited given the low-resolution target values. This inspires a multiresolution search strategy whereby the target values at each resolution level further restrict the search space for the next (higher) level of resolution, as formalized in Algorithm 1. This is a very basic algorithm searching a grid $\mathcal{G}$ of registration parameters. The resolution levels are $\ell = 0, 1, \ldots, \ell_{\max}$, where $\ell{=}0$ and $\ell{=}\ell_{\max}$ respectively correspond to highest and lowest resolutions. Here, we replace the superscript l by the cursive letter ℓ to denote "resolution level" rather than "low resolution". Thus, Q^ℓ represents the target function for resolution level ℓ, and $Q^0(\mathtt{R},\mathbf{t}) = Q(\mathtt{R},\mathbf{t})$. For each ℓ the algorithm finds the maximizer $(\mathtt{R}^{\ell*}, \mathbf{t}^{\ell*})$ of Q^ℓ over $\mathcal{G}$ (line 3), updates Q^*, the so-far best target value (line 4), and discards some elements of $\mathcal{G}$ based on Q^* and the inter-resolution bounds B^ℓ_{res} (line 5), where B^ℓ_{res} can be any of the bounds introduced in the previous section. Such a *grid search* algorithm might not seem interesting in a continuous parameter domain. Nonetheless, it demonstrates a strategy purely based on our inter-resolution bounds and gives insights about how to reduce computations in globally optimal registration algorithms where the whole parameter space is searched, which is the subject of the next section.

We test Algorithm 1 for the registration of the image pair in Fig. 1. First, we consider only the rotation parameter



(a) m = 16, p = 1 (b) m = 16, p = 2 (c) m = 16, p = 4

Fig. 3. Bounds for low-resolution to high-resolution registration with nearest neighbour interpolation and discretized computation of the target integral for the registration problem of Fig. 1. In all cases, the image f is decimated by a factor of $m = 16$, and g has been upsampled by a factor of (a) $p = 1$, (b) $p = 2$ and (c) $p = 4$. Changing p from 2 to 4 has not made a noticeable difference.

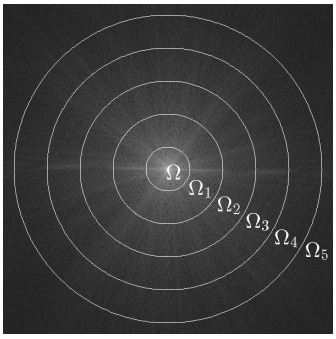


Fig. 4. The frequency domain outside Ω has been partitioned into nested radial frequency bands $\Omega_1, \Omega_2, \ldots, \Omega_P$ $(P = 5)$.

Algorithm 1 A basic multiresolution rigid motion space search algorithm for intensity-based registration.

Input:

$\{f_{\mathbf{i}}^{\ell}\}, \{g_{\mathbf{i}}^{\ell}\}$: registration image pairs for each resolution ℓ,
B_{res}^{ℓ}: inter-resolution bounds for each ℓ,
$\mathcal{G}$: a search grid of parameters $(\mathtt{R}, \mathbf{t})$,
$\mathtt{R}_0, \mathbf{t}_0$: an initial solution (optional).

1: $Q^* \leftarrow Q(\mathtt{R}_0, \mathbf{t}_0)$ or $-\infty$ ▷ so-far best target value
2: **for** $\ell \leftarrow \ell_{\max}$ **downto** 0 **do** ▷ low to high resolution
3: $\mathtt{R}^{\ell *}, \mathbf{t}^{\ell *} \leftarrow \arg\max_{\mathtt{R},\mathbf{t} \in \mathcal{G}} Q^{\ell}(\mathtt{R}, \mathbf{t})$
4: $Q^* \leftarrow \max(Q^*, Q(\mathtt{R}^{\ell *}, \mathbf{t}^{\ell *}))$
5: $\mathcal{G} \leftarrow \mathcal{G} \setminus \{\mathtt{R}, \mathbf{t} \in \mathcal{G} \,|\, Q^{\ell}(\mathtt{R}, \mathbf{t}) < Q^* - B_{\text{res}}^{\ell}\}$
6: **end for**
7: **return** $\mathtt{R}^{0*}, \mathbf{t}^{0*}$

decimation	ruled out (rot.)	ruled out (rot. + trans.)
$m = 32$	99.25 %	99.97 %
$m = 16$	0.500 %	0.0220 %
$m = 8$	0.139 %	0.0055 %
$m = 4$	0.028 %	7.0e-6 %
$m = 2$	0.056 %	1.8e-6 %
$m = 1$	0 %	0 %

TABLE I
THE PERCENTAGE OF GRID ELEMENTS RULED OUT AT EACH LEVEL OF RESOLUTION FOR A SEARCH GRID OF ROTATION (COLUMN 2) AND ROTATION PLUS TRANSLATION (COLUMN 3) PARAMETERS.

using a grid of rotation angles between -180 to 180 degrees with 0.1 degree intervals. Six levels of resolution are used with decimation factors 32, 16, 8, 4, 2 and 1. For further implementation details, we refer the reader to Sect. VI-A. It takes about 0.2 seconds for the algorithm to find the best grid element. The second column of Table I shows the percentage of grid points ruled out at each resolution level by the bounds (in line 5 of Algorithm 1). It is interesting that more than 99 percent of the grid elements are thrown away at the lowest resolution where the images are decimated by a factor of 32, and thus, have about 1000 times fewer pixels. It is also evident from the last row of the table that the best grid point has been already found at the second-highest level of resolution ($m = 2$). The third column of Table I shows the same figures for when translation is also included. The image g undergoes a translation of $(10, 40)$ pixels on top of the current rotation. Here, the rotation grid has 1-degree intervals and the translation grid is in polar coordinates where the angle of translation has 1-degree intervals and the length of translation is half of the image size in each direction with intervals of 1 pixel. It took 31 minutes to search the whole grid. Here, 99.9 percent of the grid values were ruled out at the lowest resolution level, which means a huge decrease in computations. We should mention, however, that the image of Fig. 1 has a rather compact spectrum, and also, the local maxima exhibit sharp peaks in the target function. Depending on these factors, the multiresolution scheme might lead to different efficiency gains in different images.

V. EMBEDDING INTO THE LIPSCHITZ GLOBAL OPTIMIZATION FRAMEWORK

In this section, we show how the multiresolution technique can be integrated into the a globally optimal search algorithm. We consider the Lipschitz optimization framework [7], a branch and bound approach which exploits the concept of Lipschitz continuity to obtain upper bounds within each subregion. A function $h\colon D \subseteq \mathbb{R}^n \to \mathbb{R}$ is called Lipschitz continuous if there exists a constant L such that for any $\mathbf{x}_1, \mathbf{x}_2 \in D$

$$|h(\mathbf{x}_2) - h(\mathbf{x}_1)| \leq L \, \|\mathbf{x}_2 - \mathbf{x}_1\| \,. \tag{50}$$

The smallest such L is called the Lipschitz constant. For differentiable h the Lipschitz constant is equal to the supremum of gradient magnitude within D. If $h(\mathbf{x}, \mathbf{y})$ has Lipschitz constants $L_{\mathbf{x}}$ and $L_{\mathbf{y}}$ with respect to $\mathbf{x}$ and $\mathbf{y}$ respectively, then

$$|h(\mathbf{x}_2, \mathbf{y}_2) - h(\mathbf{x}_1, \mathbf{y}_1)| \leq L_{\mathbf{x}} \, \|\mathbf{x}_2 - \mathbf{x}_1\| + L_{\mathbf{y}} \, \|\mathbf{y}_2 - \mathbf{y}_1\|. \tag{51}$$

In Lipschitz optimization, the target function is computed at the centre of each subregion, and then, using (50) or (51) an

upper-bound is found on the target values within the subregion. The subregion is either rejected or split based on whether or not the upper bound is smaller than the so far best target value.

A. Lipschitz bounds

The efficiency of the algorithm highly depends on the quality of the Lipschitz bounds. An example of Lipschitz bounds for intensity-based registration is given in [9] in the context of shape models. Here, we present better bounds by looking at the target function in the frequency domain. For the images we have experimented on, our bounds are smaller by at least one order of magnitude (see Sect. VI). Let $\mathbf{F}(\mathbf{z}) = [\mathrm{real}(F(\mathbf{z})), \mathrm{imag}(F(\mathbf{z}))]^T \in \mathbb{R}^2$, and similarly define $\mathbf{G}(\mathbf{z})$. Then using the fact that $F(-\mathbf{z}) = \overline{F(\mathbf{z})}$ and $G(-\mathbf{z}) = \overline{G(\mathbf{z})}$ for real-valued f and g, we can rewrite (2) as

$$Q(\mathtt{R}, \mathbf{t}) = \int \mathbf{F}(\mathbf{z})^T \, \Gamma(2\pi \mathbf{t}^T \mathbf{z}) \, \mathbf{G}(\mathtt{R}\mathbf{z}) \, d\mathbf{z}. \tag{52}$$

where $\Gamma(\phi)$ is the 2D rotation matrix of angle ϕ, that is

$$\Gamma(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{53}$$

Now, the gradient of the target with respect to $\mathbf{t}$ is

$$\frac{d}{d\mathbf{t}} Q(\mathtt{R}, \mathbf{t}) = 2\pi \int \mathbf{z} \, \mathbf{F}(\mathbf{z})^T \, \Gamma'(2\pi \mathbf{t}^T \mathbf{z}) \, \mathbf{G}(\mathtt{R}\mathbf{z}) \, d\mathbf{z}. \tag{54}$$

where $\Gamma'(\phi) = \frac{d}{d\phi}\Gamma(\phi)$, which is also a rotation matrix. To obtain a fairly good upper bound on the magnitude of (54) we first divide the frequency domain into radial bands $\Omega_0, \Omega_1, \ldots, \Omega_P$, where $\Omega_i = \{\mathbf{z} \,|\, r_i \leq \|\mathbf{z}\| < r_{i+1}\}$ for radii $0 = r_0 < r_1 < \cdots < r_{P+1}$. This is similar to the approach taken in Sect. III-C, only instead of partitioning the area outside Ω, we partition the entire frequency domain (look at Fig. (4) and replace Ω by Ω_0). Now, from (54) we can write

$$\begin{aligned}
\left\| \frac{d}{d\mathbf{t}} Q(\mathtt{R}, \mathbf{t}) \right\| &\leq 2\pi \sum_{i=0}^{P} \int_{\Omega_i} \|\mathbf{z}\| \left| \mathbf{F}(\mathbf{z})^T \, \Gamma'(2\pi \mathbf{t}^T \mathbf{z}) \, \mathbf{G}(\mathtt{R}\mathbf{z}) \right| d\mathbf{z} \\
&\leq 2\pi \sum_{i=0}^{P} (\sup_{\mathbf{z} \in \Omega_i} \|\mathbf{z}\|) \int_{\Omega_i} \left| \mathbf{F}(\mathbf{z})^T \, \Gamma'(2\pi \mathbf{t}^T \mathbf{z}) \, \mathbf{G}(\mathtt{R}\mathbf{z}) \right| d\mathbf{z} \\
&\leq 2\pi \sum_{i=0}^{P} r_{i+1} \int_{\Omega_i} \|\mathbf{F}(\mathbf{z})\| \, \|\mathbf{G}(\mathtt{R}\mathbf{z})\| \, d\mathbf{z} \\
&\leq 2\pi \sum_{i=0}^{P} r_{i+1} \sqrt{\int_{\Omega_i} \|\mathbf{F}(\mathbf{z})\|^2 \, d\mathbf{z}} \sqrt{\int_{\Omega_i} \|\mathbf{G}(\mathbf{z})\|^2 \, d\mathbf{z}}.
\end{aligned} \tag{55}$$

In the third line of the above we used the fact that $\Gamma'(2\pi \mathbf{t}^T \mathbf{z})$ is a rotation matrix. Also, notice that $\|\mathbf{F}(\mathbf{z})\|$ is simply equal to the magnitude of the complex quantity $F(\mathbf{z})$. The integrals can be computed similarly to Sect. III-C. What makes the above a good bound is that large values of r_{i+1} are multiplied by high-frequency components of F and G, which are typically small in natural images. This has been possible due to the division of the frequency domain into radial bands Ω_i.

Assume that the rotation matrix $\mathtt{R}$ is parameterized as a function of a set of parameters, and let $\rho \in \mathbb{R}$ be one of the parameters. Then

$$\frac{d}{d\rho} Q(\mathtt{R}, \mathbf{t}) = \int \mathbf{F}(\mathbf{z})^T \, \Gamma(2\pi \mathbf{t}^T \mathbf{z}) \, J_{\mathbf{G}}(\mathtt{R}\mathbf{z}) \frac{d\mathtt{R}}{d\rho} \mathbf{z} \, d\mathbf{z}, \tag{56}$$

where $J_{\mathbf{G}}$ is the Jacobian matrix of $\mathbf{G}$. It follows

$$\begin{aligned}
\left| \frac{d}{d\rho} Q(\mathtt{R}, \mathbf{t}) \right| &\leq \sum_{i=0}^{P} \int_{\Omega_i} \left| \mathbf{F}(\mathbf{z})^T \, \Gamma(2\pi \mathbf{t}^T \mathbf{z}) \, J_{\mathbf{G}}(\mathtt{R}\mathbf{z}) \frac{d\mathtt{R}}{d\rho} \mathbf{z} \right| d\mathbf{z} \\
&\leq \sum_{i=0}^{P} \sqrt{\int_{\Omega_i} \|\mathbf{F}(\mathbf{z})\|^2 \, d\mathbf{z}} \sqrt{\int_{\Omega_i} \left\| J_{\mathbf{G}}(\mathbf{z}) \frac{d\mathtt{R}}{d\rho} \mathtt{R}^{-1} \mathbf{z} \right\|^2 d\mathbf{z}}.
\end{aligned} \tag{57}$$

For 2D the only parameter is the angle of rotation $\rho = \theta$. Notice that $\frac{d\mathtt{R}}{d\theta}\mathtt{R}^{-1}$ is equivalent to a 90 degrees rotation counterclockwise. Thus, we have

$$J_{\mathbf{G}}(\mathbf{z}) \frac{d\mathtt{R}}{d\theta} \mathtt{R}^{-1} \mathbf{z} = J_{\mathbf{G}}(\mathbf{z}) \mathbf{z}^{\perp}, \tag{58}$$

where $\mathbf{z}^{\perp} = [-z_2, z_1]^T$ is the vector $\mathbf{z}$ rotated by 90 degrees.

For the more complex 3D case we use the upper bound:

$$\left\| J_{\mathbf{G}}(\mathbf{z}) \frac{d\mathtt{R}}{d\rho} \mathtt{R}^{-1} \mathbf{z} \right\| \leq \|J_{\mathbf{G}}(\mathbf{z})\| \left\| \frac{d\mathtt{R}}{d\rho} \mathtt{R}^{-1} \mathbf{z} \right\|, \tag{59}$$

where $\|J_{\mathbf{G}}(\mathbf{z})\|$ is the spectral norm of the matrix $J_{\mathbf{G}}(\mathbf{z})$. We use the axis-angle representation of the rotation, with the unit vector $\boldsymbol{\omega}$ representing the axis of rotation and θ representing the rotation angle. The rotation can be formulated as

$$\mathtt{R}\mathbf{z} = \cos\theta \, (\mathtt{I} - \boldsymbol{\omega}\boldsymbol{\omega}^T) \, \mathbf{z} + \sin\theta \, (\boldsymbol{\omega} \times \mathbf{z}) + \boldsymbol{\omega}\boldsymbol{\omega}^T \, \mathbf{z} \tag{60}$$

where $\times$ denotes the cross product. Similarly, $\mathtt{R}^{-1}\mathbf{z}$ is obtained by replacing θ in the above with $-\theta$. Simple calculations give

$$\frac{d\mathtt{R}}{d\theta} \mathtt{R}^{-1} \mathbf{z} = \boldsymbol{\omega} \times \mathbf{z} \leq \|\mathbf{z}\|. \tag{61}$$

Now, we parameterize $\boldsymbol{\omega}$ in the spherical coordinates using two angles ϕ and ψ, as $\boldsymbol{\omega} = [\cos\phi\cos\psi, \ \sin\phi\cos\psi, \ \sin\psi]^T$. Let $\boldsymbol{\tau} = [-\sin\phi, \ \cos\phi, \ 0]^T$ and $\boldsymbol{\nu} = [-\cos\phi\sin\psi, \ -\sin\phi\sin\psi, \ \cos\psi]^T$, and notice that $\boldsymbol{\omega}$, $\boldsymbol{\tau}$ and $\boldsymbol{\nu}$ form an orthonormal basis with $\boldsymbol{\omega} \times \boldsymbol{\tau} = \boldsymbol{\nu}$. Further, we have

$$\frac{d\boldsymbol{\omega}}{d\phi} = \cos\psi \ \boldsymbol{\tau}, \quad \frac{d\boldsymbol{\omega}}{d\psi} = \boldsymbol{\nu}. \tag{62}$$

By taking $s = \sin(\frac{\theta}{2})$ and $c = \cos(\frac{\theta}{2})$, and using $\boldsymbol{\nu} \times \mathbf{z} = (\boldsymbol{\tau}\boldsymbol{\omega}^T - \boldsymbol{\omega}\boldsymbol{\tau}^T) \, \mathbf{z}$ we get

$$\frac{d\mathtt{R}}{d\psi} \mathbf{z} = 2s^2 \, (\boldsymbol{\omega}\boldsymbol{\nu}^T + \boldsymbol{\nu}\boldsymbol{\omega}^T) \, \mathbf{z} + 2cs \, (\boldsymbol{\nu} \times \mathbf{z}) = 2s \, \mathtt{U}\mathtt{S}\mathtt{U}^T \mathbf{z}, \tag{63}$$

where $\mathtt{U} = [\boldsymbol{\omega}, \boldsymbol{\tau}, \boldsymbol{\nu}]$ and

$$\mathtt{S} = \begin{bmatrix} 0 & -c & s \\ c & 0 & 0 \\ s & 0 & 0 \end{bmatrix}. \tag{64}$$

As $\mathtt{U}$ is orthogonal and $\mathtt{S}$ has spectral norm 1, we have

$$\left\| \frac{d\mathtt{R}}{d\psi} \mathtt{R}^{-1} \mathbf{z} \right\| = \left\| 2s \, \mathtt{U}\mathtt{S}\mathtt{U}^T \mathtt{R}^{-1} \mathbf{z} \right\| \leq 2 \, |\sin(\theta/2)| \, \|\mathbf{z}\|. \tag{65}$$

In a similar way, for ϕ one can obtain

$$\left| \frac{d\mathtt{R}}{d\phi} \mathtt{R}^{-1} \mathbf{z} \right| \leq 2 \, |\sin(\theta/2)| \, |\cos(\psi)| \, \|\mathbf{z}\|. \tag{66}$$

Lipschitz bounds is obtained from (65) and (66) by maximizing $|\sin(\theta/2)|$ and $|\cos(\psi)|$ within each subregion. By substituting into (57), one realizes that the integrand in the second square root linearly grows with the magnitude of $\mathbf{z}$.

However, this is compensated by the fact that larger vectors $\mathbf{z}$ lie inside higher frequency bands Ω_i, and thus, are multiplied by high-frequency components of F (and $J_{\mathbf{G}}$) which are typically small. Another important observation is that the Lipschitz bounds get smaller as images are more decimated. This can be easily seen by looking at (55) and (57), and noticing that the spectrum of a low-resolution image is obtained by zeroing out the high-resolution spectrum outside a certain ball.

B. A multiresolution branch and bound algorithm

We divide the registration parameter space into a number of hypercubes. In Lipschitz optimization, each hypercube either gets rejected or gets split depending on whether or not the corresponding upper bound on the target value is smaller than the largest target value found so far. The idea behind our multiresolution algorithm is to reduce the computations by using lower resolution images, as much as possible, to make the reject/split decision. To see how this is done, consider a cube $\mathcal{C} \subseteq SE(d)$ whose centre corresponds to parameters $\mathtt{R}_c, \mathbf{t}_c$. Like Sect. IV, we have a set of resolution levels $\ell = 0, 1, \ldots, \ell_{\max}$. For each ℓ there are two different bounds: the inter-resolution bound B^{ℓ}_{res} and the Lipschitz bound B^{ℓ}_{Lip}. The Lipschitz bound can be obtained from (50) or (51), and bounds the deviation of $Q^{\ell}(\mathtt{R}, \mathbf{t})$ from $Q^{\ell}(\mathtt{R}_c, \mathbf{t}_c)$ within the cube $\mathcal{C}$, that is $|Q^{\ell}(\mathtt{R}, \mathbf{t}) - Q^{\ell}(\mathtt{R}_c, \mathbf{t}_c)| \leq B^{\ell}_{\text{Lip}}$ for all $(\mathtt{R}, \mathbf{t}) \in \mathcal{C}$. The inter-resolution bound gives $|Q^{\ell}(\mathtt{R}_c, \mathbf{t}_c) - Q(\mathtt{R}_c, \mathbf{t}_c)| \leq B^{\ell}_{\text{res}}$. Thus, we can define a total bound $B^{\ell}_{\text{total}}{=}B^{\ell}_{\text{res}} + B^{\ell}_{\text{Lip}}$ giving

$$\left|Q(\mathtt{R}, \mathbf{t}) - Q^{\ell}(\mathtt{R}_c, \mathbf{t}_c)\right| \leq B^{\ell}_{\text{total}} \quad \text{for all } (\mathtt{R}, \mathbf{t}) \in \mathcal{C}. \tag{67}$$

Therefore, if $Q^{\ell}(\mathtt{R}_c, \mathbf{t}_c){+}B^{\ell}_{\text{total}}$ is smaller than Q^*, the currently best target value, we can safely discard $\mathcal{C}$. Otherwise, we test the next (higher) resolution level $\ell{-}1$. More precisely, our strategy is as follows. First, for any cube $\mathcal{C}$ we define $\ell_{\min} = \operatorname{argmin}_{\ell} B^{\ell}_{\text{total}}$. For each cube, we examine the upper bounds for $\ell = \ell_{\max}, \ldots, \ell_{\min}$ in order. If $Q^{\ell}(\mathtt{R}_c, \mathbf{t}_c) + B^{\ell}_{\text{total}} \leq Q^*$ for any ℓ then we discard $\mathcal{C}$, otherwise, we split it. We do not go further below $\ell_{\min}$ as otherwise B^{ℓ}_{total} would become larger and there would be a high change of $Q^{\ell}(\mathtt{R}_c, \mathbf{t}_c) + B^{\ell}_{\text{total}}$ falling above Q^* again. Notice that B^{ℓ}_{res} is equal or proportional to the residual of the energy after low-pass filtering. Thus, $0{=}B^0_{\text{res}} \leq B^1_{\text{res}} \leq \ldots \leq B^{\ell_{\max}}_{\text{res}}$. On the contrary, we discussed in the previous section that $B^0_{\text{Lip}} \geq B^1_{\text{Lip}} \geq \ldots \geq B^{\ell_{\max}}_{\text{Lip}}$. When the algorithm starts the cubes are large, and therefore, B^{ℓ}_{Lip}-s dominate B^{ℓ}_{res}-s. Consequently, $\ell_{\min}$ becomes close to $\ell_{\max}$, and only lower resolution images are used to make the reject/split decision. As the algorithm goes on, the cubes get divided into smaller cubes. Thus, B^{ℓ}_{Lip} becomes smaller compared to B^{ℓ}_{res}, and $\ell_{\min}$ moves towards zero. Therefore, higher resolution images are examined. Due to this strategy, a large portion of the search space is explored in the lower-resolution domains.

Algorithm 2 describes our approach. It uses the *breadth-first search* strategy, using a *queue* data structure. We use a special element called **level-delimiter** to separate different levels of the search tree in the queue. After searching all cubes of each level, we have a candidate optimal parameter $(\mathtt{R}^{\ell *}, \mathbf{t}^{\ell *})$, which is the maximizer of $Q^{\ell_{\min}}(\mathtt{R}_c,\mathbf{t}_c){-}B^{\ell_{\min}}$ among all cubes in this level (see line 21 of the algorithm). We examine the original cost function $Q{=}Q^0$ on this candidate parameter set. If $Q(\mathtt{R}^{\ell *}, \mathbf{t}^{\ell *}) > Q^*$, then Q^*, $\mathtt{R}^*$ and $\mathbf{t}^*$ are updated (line 8).

It is not always necessary to split across every dimension of the cube. Notice that B^{ℓ}_{Lip} is the sum of the bounds for each parameter, as shown in (51) for two parameters. We sort the bounds descendingly, start from the parameter with the biggest bound, and keep splitting across the parameters, until the sum of the rest of the bounds are smaller than $Q^* - B^{\ell_{\min}}_{\text{res}}$, which is the margin required for rejecting the cube at resolution $\ell_{\min}$.

The algorithm continues until FINISHCONDITION( ) is satisfied. This happens when Q^* is guaranteed to be within a certain distance ε of the optimal target. For this, first, for each certain cube, we compute an upper bound $Q^{\text{up}}_{\mathcal{C}} = \min_{\ell=\ell_{\min},\ldots,\ell_{\max}} Q^{\ell}(\mathtt{R}_c, \mathbf{t}_c) + B^{\ell}_{\text{total}}$. The upper-bound Q^{up} is the maximum of $Q^{\text{up}}_{\mathcal{C}}$ among all cubes $\mathcal{C}$ in the queue. The algorithm finishes when $|Q^{\text{up}} - Q^*| < \varepsilon$.

Algorithm 2 A multiresolution Lipschitz optimization algorithm for rigid intensity-based registration.

Input:
$\{f^{\ell}_{\mathbf{i}}\}, \{g^{\ell}_{\mathbf{i}}\}$: registration image pairs for each resolution ℓ,
B^{ℓ}_{res}: inter-resolution bounds for each ℓ,
L^{ℓ}: Lipschitz constant estimates for all parameters, for each ℓ,
$\mathcal{C}_0$: the initial hypercube containing the parameter space,
$\mathtt{R}_0, \mathbf{t}_0$: an initial solution (optional).

1: $Q^* \leftarrow Q(\mathtt{R}_0, \mathbf{t}_0)$ or $-\infty$ ▷ so-far best target value
2: $Q^{\ell *} \leftarrow -\infty$ ▷ best low-resolution lower-bound on target
3: QUEUE.ENQUEUE($\mathcal{C}_0$)
4: QUEUE.ENQUEUE(**level-delimiter**)
5: **repeat**
6: $\quad\mathcal{C} \leftarrow$ QUEUE.DEQUEUE( )
7: $\quad$**if** $\mathcal{C}$ = **level-delimiter then** ▷ level completed
8: $\quad\quad Q^*, \mathtt{R}^*, \mathbf{t}^* \leftarrow$ AMAX$(Q^*, \mathtt{R}^*, \mathbf{t}^*, Q(\mathtt{R}^{\ell *}, \mathbf{t}^{\ell *}), \mathtt{R}^{\ell *}, \mathbf{t}^{\ell *})$
9: $\quad\quad$QUEUE.ENQUEUE(**level-delimiter**)
10: $\quad\quad$**go to line** 26 ▷ next iteration
11: $\quad$**end if**
12: $\quad\mathtt{R}_c, \mathbf{t}_c \leftarrow$ GETCENTRE($\mathcal{C}$)
13: $\quad B^{\ell}_{\text{Lip}} \leftarrow$ LIPSCHITZBOUND$(L^{\ell}, \mathcal{C})$ **for all** ℓ
14: $\quad B^{\ell}_{\text{total}} \leftarrow B^{\ell}_{\text{res}} + B^{\ell}_{\text{Lip}}$ **for all** ℓ
15: $\quad\ell_{\min} \leftarrow \operatorname{argmin}_{\ell} B^{\ell}_{\text{total}}$
16: $\quad$**for** $\ell \leftarrow \ell_{\max}$ **downto** $\ell_{\min}$ **do**
17: $\quad\quad$**if** $Q^{\ell}(\mathtt{R}_c, \mathbf{t}_c) + B^{\ell}_{\text{total}} \leq Q^*$ **then**
18: $\quad\quad\quad$**go to line** 26 ▷ reject $\mathcal{C}$, next iteration
19: $\quad\quad$**end if**
20: $\quad$**end for**
21: $\quad Q^{\ell *}, \mathtt{R}^{\ell *}, \mathbf{t}^{\ell *} \leftarrow$ AMAX$(Q^{\ell *}, \mathtt{R}^{\ell *}, \mathbf{t}^{\ell *}, Q^{\ell_{\min}}(\mathtt{R}_c, \mathbf{t}_c) - B^{\ell_{\min}}_{\text{res}}, \mathtt{R}_c, \mathbf{t}_c)$
22: $\quad Q^*, \mathtt{R}^*, \mathbf{t}^* \leftarrow$ AMAX$(Q^*, \mathtt{R}^*, \mathbf{t}^*, Q^{\ell *}, \mathtt{R}^{\ell *}, \mathbf{t}^{\ell *})$
23: $\quad$**for** $\mathcal{C}'$ **in** SPLIT($\mathcal{C}$) **do**
24: $\quad\quad$QUEUE.ENQUEUE($\mathcal{C}'$)
25: $\quad$**end for**
26: **until** FINISHCONDITION( )
27: **return** $\mathtt{R}^*, \mathbf{t}^*$

28: **procedure** AMAX$(Q_1, \mathtt{R}_1, \mathbf{t}_1, Q_2, \mathtt{R}_2, \mathbf{t}_2)$
29: $\quad$**if** $Q_1 \geq Q_2$ **then return** $Q_1, \mathtt{R}_1, \mathbf{t}_1$
30: $\quad\quad\quad$**else return** $Q_2, \mathtt{R}_2, \mathbf{t}_2$
31: **end procedure**

VI. Experiments

The only globally optimal approach we could find on rigid intensity-based registration is [9], a single-resolution Lipschitz optimization for 2D images with larger Lipschitz bounds than ours. For the starfish example in Fig. 1, their proposed bound is 15.1 and 11.8 times larger than ours for the rotation and translation parameters, respectively[8]. Similar results are observed for other images. Therefore, here we compare the single-resolution and multiresolution versions of our own algorithm and report the multiresolution speed gain.

A. Implementation

Algorithm setup: We use the *low-resolution to high-resolution* registration of Sect. III-B with discretized integration and the bound (46), where g is upsampled by a factor of 2, and bilinear (or trilinear) interpolation is used. For all 2D images, the highest decimation rate is set to the largest power of 2 where $E^h_{fg} = \left\|f^h\right\| \left\|g^h\right\|$ in (20) is less than 0.05 of $\|f\| \, \|g\|$. For the example of Fig. 1, this gives a maximum decimation rate of 32. However, for images with a less compact spectrum, this can be smaller. For 3D volumes (all binary shapes of vertebrae) we use a maximum decimation rate of 16. We set the convergence threshold ε to $0.01 \, \|f\| \, \|g\|$ (see Sect. V-B). For the starfish example of Fig. 1, this gives an accuracy of about 0.2 pixels for translation and 0.04 degrees for rotation.

Platform: The main algorithm is implemented in C++ while constructing the multiresolution pyramid and the bounds are done under Matlab. The machine has Intel Xeon 3.40GHz CPU and 24GB of RAM.

Low-pass filtering and downsampling: To implement the ideal filter (13) we zero out the DFT coefficients of the discrete image outside the corresponding ball around the origin, and then take inverse DFT. This is not exactly equivalent to applying the ideal filter (13) to the corresponding continuous image. The error, however, is negligible for images with a dark margin, which can be obtained by zero padding, or reversing the intensity values for images with a bright background.

Upsampling: To upsample, zero-valued higher-frequency components are added to the DFT of the discrete image, and inverse DFT is taken. Again, this is not equal to upsampling with sinc interpolation, but is accurate enough for images with a dark margin, or when higher-resolution DFT is used.

Image intensity: All images are converted to grayscale prior to registration. For images with a white or light background, the intensity values are inverted.

B. 2D registration

First, we consider the starfish image of Fig. 1(a). The original 782×782 image undergoes a random rotation between 0 to 360 degrees, a random translation between 0 and one-sixth of the image size in each coordinate, and a slight affine warp so that the images are not completely identical. The top row of Fig. 5 shows the results for 20 different random trials. The average speed gain is around 100 which is quite considerable.

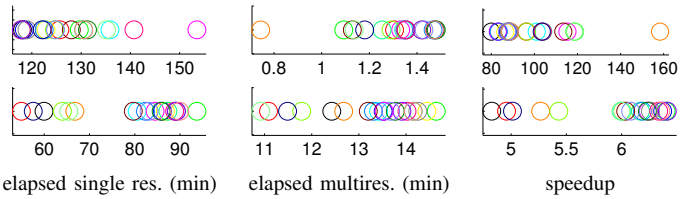


Fig. 5. Applying the Lipschitz optimization algorithm on the starfish image of Fig. 1 (first row) and the brain MRI image of Fig. 6 (second row). In each case, we depict the elapsed time in *minutes* for the single resolution and multiresolution approaches, plus the speed gain given by the multiresolution approach. Different colours represent different random experiments.

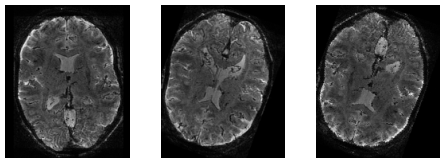


Fig. 6. Registering two different slices of a brain MRI image. (a) the first slice, (b) the second slice rotated and translated, and (c) the first slice registered to the second slice. The data is obtained from [26].

Next, we apply our algorithm to register two different slices of the same brain MRI volume being 10 slices away from each other (Fig. 6). We randomly rotate and translate the second slice as in the previous case. The images are 192×156. The second row of Fig. 5 shows the elapsed times and speedup factors in this case. Here, the speedup is around 5.5 times. This shows that the efficiency gain can vary based on how compact the image energy is in the frequency domain and how the target value of the true solution stands out in the registration parameter space.

Now, we apply our algorithm on a dataset of 32 different leaves [27] (see Fig. 7). The images are 570 by 720. The random trials are conducted as in the above. We run 10 random trials for each leaf, resulting in a total of 320 trials. The histograms of the elapsed times and the speedup factor have been depicted in Fig. 7. Notice that the x-axes have logarithmic scales. The speed gain is significant in most cases.

Fig. 7. Samples of the leaf dataset [27].

C. 3D rotation search

Our breadth-first search strategy is too memory-demanding for a full 3D registration with 6 degrees of freedom. What is done here is to align the centres of masses of the two images and then perform a rotation-only search (3 degrees of freedom) around the centre of mass. We register binary vertebra volumes

[8]Estimating the Lipschitz constant for the *rotation angle* in [9] involves computing the maximum of the magnitude of gradient in the second image. To get a stable estimation, we first applied a Gaussian smoothing with $\sigma = 3$ pixels to the second image and then used the difference of the neighbouring pixels to compute the gradient. Without the smoothing the Lipschitz constant would be 70 times larger than ours.

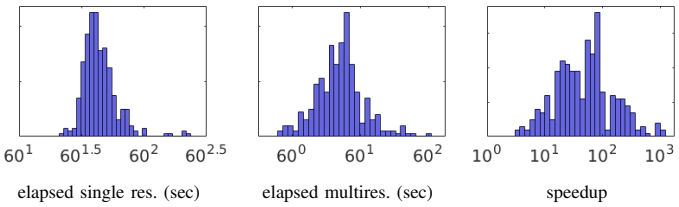


Fig. 8. Histograms of single resolution and multiresolution elapsed time in seconds, and the speed gain achieved by the multiresolution approach. Note that the histograms are computed in logarithmic scale.

of two different subjects (Fig. 9) for 17 different vertebrae. The volumes are obtained by manually labelling CT images[9]. The second volume is rotated to make the registration more challenging. The accurate alignment of such images is usually required as the first step of constructing a shape model. An example of the registration is illustrated in Fig. 9. The elapsed times and speed gains are reported in Fig. 10. The speed gain is remarkable in most cases.

Fig. 9. 3D registration of binary vertebra shapes, (left, middle) the registration image pair, (right) the first image registered to the second image.

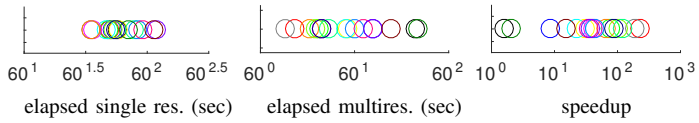


Fig. 10. Elapsed times for the single resolution and multiresolution approaches, plus the multiresolution speed gain for 3D rotation search. Note that the results are plotted in logarithmic scale.

D. Reflective symmetry detection

A closely related problem to rigid registration is detecting the axis or plane of symmetry. Most natural images are not perfectly symmetric, yet, one can estimate symmetry by maximizing the correlation of an image with its reflection across a line (2D) or plane (3D). The problem formulation and other details are given in Appendix C. Here, we demonstrate the application of the multiresolution Lipschitz algorithm.

We experiment on images of a butterfly, a face and a brain MRI slice. For each image, 20 trials are run where images are randomly rotated and translated similar to Sect. VI-B. Samples of the estimated axis of symmetry are shown in Fig. 11. The elapsed times for the single resolution and multiresolution approaches as well and the speed gains are reported in Fig. 12.

For the 3D case, we apply our algorithm on the binary volumes of 18 different vertebrae of a single patient from the dataset used in Sect. VI-B. The first and second dimensions of the binary volumes range from about 100 to 150 voxels and the third dimension ranges from 24 to 34 voxels. Fig. 13 shows

[9]The data is obtained from CSI 2014 Workshop http://csi-workshop.weebly.com/challenges.html, also see [28]

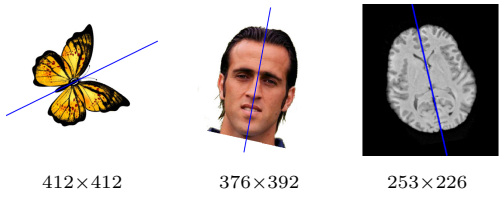


Fig. 11. Reflective symmetry detection for images of the butterfly, face and brain MRI slice. The butterfly image is from http://hdwallpapersrocks.com. The brain MRI image was obtained from http://www.brain-map.org.

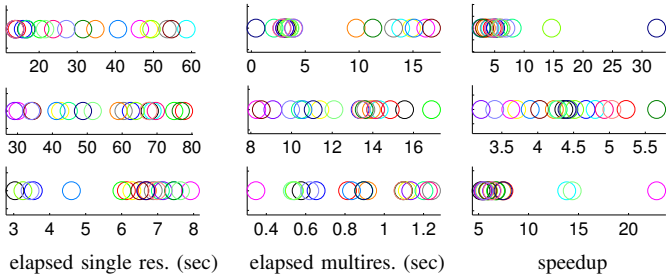


Fig. 12. elapsed times of single resolution and multiresolution approaches, and the multiresolution speed gain for the butterfly image (top row), the face (middle row) and the brain MRI slice (bottom row)

the output of our algorithm applied on 3 different vertebrae. The elapsed times and speed gain for each of the 18 cases are illustrated in Fig. 14 in logarithmic coordinates. The speed gain varies from 3 to 380 but is notable in many cases.

VII. Conclusion and Future Work

We showed that low-resolution target values can tightly bound the high-resolution target function for rigid intensity-based image registration. This led to a multiresolution search scheme in which the search at each resolution limits the search space for the next higher resolution level. By embedding into the Lipschitz optimization framework, we showed that this strategy can significantly speed up the globally optimal registration algorithms. An extension to this work could be to find bounds for more general transformations (similarity, affine, nonlinear, etc.) and other types of target functions (robust, information theoretic, etc.). The effectiveness of our

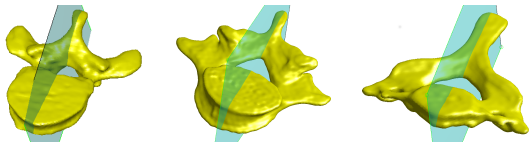


Fig. 13. Estimating the plane of symmetry for the binary volume of three different vertebrae. Notice that the right most vertebra is not symmetric, so the globally optimal solution compromises the tail to capture the symmetry in rest of the body of the vertebra with more voxels.

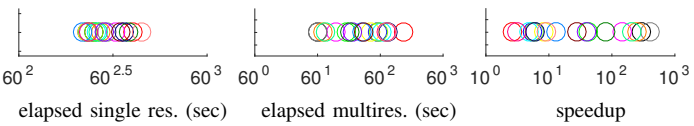


Fig. 14. Elapsed times for single resolution and multiresolution approaches, plus the multiresolution speed gain (all in logarithmic scale), for 3D reflective symmetry detection.

multiresolution approach comes from the energy compaction in natural images in the Fourier basis. An important extension is to explore similar ideas for other bases, and obtaining optimized bases tailored for each image pair.

APPENDIX A
BOUNDED SUPPORT INTERPOLATION

Proof of Proposition 2. The difference $\left|\hat{Q}(\mathtt{R},\mathbf{t}) - \hat{Q}^l(\mathtt{R},\mathbf{t})\right|$ can be written in the frequency domain using (2):

$$\left|\int \overline{\hat{F}(\mathbf{z})}\,\hat{G}(\mathtt{R}\mathbf{z})\,e^{2\pi i\mathbf{t}^T\mathbf{z}}\,d\mathbf{x} - \int \overline{\hat{F}^l(\mathbf{z})}\,\hat{G}^l(\mathtt{R}\mathbf{z})\,e^{2\pi i\mathbf{t}^T\mathbf{z}}d\mathbf{x}\right|. \quad (68)$$

Using the triangle inequality, we break the above into integrations over Ω and it complement $\bar{\Omega}$:

$$\left|\int D(\mathbf{z})d\mathbf{x}\right| \le \left|\int_{\Omega} D(\mathbf{z})d\mathbf{x}\right| + \left|\int_{\bar{\Omega}} D(\mathbf{z})d\mathbf{x}\right|, \quad (69)$$

where $D(\mathbf{z}) = e^{2\pi i\mathbf{t}^T\mathbf{z}}\left(\overline{\hat{F}(\mathbf{z})}\,\hat{G}(\mathtt{R}\mathbf{z}) - \overline{\hat{F}^l(\mathbf{z})}\,\hat{G}^l(\mathtt{R}\mathbf{z})\right)$. Let us first deal with the integral inside Ω. Since $F(\mathbf{z})$=$F^l(\mathbf{z})$ for all $\mathbf{z}\in\Omega$, we expect $\hat{F}(\mathbf{x})$ and $\hat{F}^l(\mathbf{x})$ to be close. A similar argument goes for $\hat{G}(\mathbf{x})$ and $\hat{G}^l(\mathbf{x})$. Now using the equality

$$ab - a^lb^l = a(b-b^l) + b(a-a^l) - (a-a^l)(b-b^l) \quad (70)$$

for any four complex numbers a, b, a^l and b^l we can write

$$\begin{aligned}
&\left|\int_{\Omega}\left(\overline{\hat{F}(\mathbf{z})}\,\hat{G}(\mathtt{R}\mathbf{z}) - \hat{F}^l(\mathbf{z})\,\hat{G}^l(\mathtt{R}\mathbf{z})\right)e^{2\pi i\mathbf{t}^T\mathbf{z}}\right| \\
\le\ &\left|\int_{\Omega}\overline{\hat{F}(\mathbf{z})}\,\left(\hat{G}(\mathtt{R}\mathbf{z}) - \hat{G}^l(\mathtt{R}\mathbf{z})\right)e^{2\pi i\mathbf{t}^T\mathbf{z}}\right| \\
&+\left|\int_{\Omega}\hat{G}(\mathtt{R}\mathbf{z})\left(\overline{\hat{F}(\mathbf{z})} - \overline{\hat{F}^l(\mathbf{z})}\right)e^{2\pi i\mathbf{t}^T\mathbf{z}}\right| \\
&+\left|\int_{\Omega}\left(\overline{\hat{F}(\mathbf{z})} - \overline{\hat{F}^l(\mathbf{z})}\right)\left(\hat{G}(\mathtt{R}\mathbf{z}) - \hat{G}^l(\mathtt{R}\mathbf{z})\right)e^{2\pi i\mathbf{t}^T\mathbf{z}}\right| \\
\le&\sqrt{\textstyle\int_{\Omega}|\hat{F}(\mathbf{z})|^2d\mathbf{z}}\sqrt{\textstyle\int_{\Omega}\left|\hat{G}(\mathbf{z}) - \hat{G}^l(\mathbf{z})\right|^2 d\mathbf{z}} \\
&+\sqrt{\textstyle\int_{\Omega}|\hat{G}(\mathbf{z})|^2d\mathbf{z}}\sqrt{\textstyle\int_{\Omega}\left|\hat{F}(\mathbf{z}) - \hat{F}^l(\mathbf{z})\right|^2 d\mathbf{z}} \\
&+\sqrt{\textstyle\int_{\Omega}\left|\hat{F}(\mathbf{z})-\hat{F}^l(\mathbf{z})\right|^2 d\mathbf{z}}\sqrt{\textstyle\int_{\Omega}\left|\hat{G}(\mathbf{z})-\hat{G}^l(\mathbf{z})\right|^2 d\mathbf{z}} \quad (71)
\end{aligned}$$

which is exactly equal to lines (27) and (28) of the bound. The first inequality above is due to the triangle inequality. The second inequality is obtained by Cauchy-Schwarz followed by a change of variables $\mathbf{z}\leftarrow\mathtt{R}\mathbf{z}$. Now, we consider the integral outside Ω. Since, outside Ω the signals $\hat{F}$ and $\hat{F}^l$ (also $\hat{G}$ and $\hat{G}^l$) are not similar we consider them separately:

$$\begin{aligned}
&\left|\int_{\bar{\Omega}}\left(\overline{\hat{F}(\mathbf{z})}\,\hat{G}(\mathtt{R}\mathbf{z}) - \hat{F}^l(\mathbf{z})\,\hat{G}^l(\mathtt{R}\mathbf{z})\right)e^{2\pi i\mathbf{t}^T\mathbf{z}}d\mathbf{z}\right| \\
&\le \left|\textstyle\int_{\bar{\Omega}}\overline{\hat{F}(\mathbf{z})}\,\hat{G}(\mathtt{R}\mathbf{z})e^{2\pi i\mathbf{t}^T\mathbf{z}}d\mathbf{z}\right| + \left|\textstyle\int_{\bar{\Omega}}\hat{F}^l(\mathbf{z})\,\hat{G}^l(\mathtt{R}\mathbf{z})e^{2\pi i\mathbf{t}^T\mathbf{z}}d\mathbf{z}\right| \\
&\le \sqrt{\textstyle\int_{\bar{\Omega}}|\hat{F}(\mathbf{z})|^2d\mathbf{z}}\sqrt{\textstyle\int_{\bar{\Omega}}|\hat{G}(\mathbf{z})|^2d\mathbf{z}} \\
&\quad+\sqrt{\textstyle\int_{\bar{\Omega}}|\hat{F}^l(\mathbf{z})|^2d\mathbf{z}}\sqrt{\textstyle\int_{\bar{\Omega}}|\hat{G}^l(\mathbf{z})|^2d\mathbf{z}}. \quad (72)
\end{aligned}$$

This is equal to (29). □

APPENDIX B

To obtain (39), we treat the frequency areas Ω and $\bar{\Omega}$ separately, like in (69). We use the same approach from which (27) and (28) are obtained. However, since in this case $\hat{F}=F$ and $\hat{F}^l = F^l$, and also $F(\mathbf{z}) = F^l(\mathbf{z})$ inside Ω, then we have $E_{\Omega}(\hat{F}-\hat{F}^l) = E_{\Omega}(F-F^l) = 0$. Therefore, (27) and (28) reduce to $\sqrt{E_{\Omega}(F)\,E_{\Omega}(\hat{G}-\hat{G}^l)}$ which is the first term in (39). Now, outside Ω we have $F^l(\mathbf{z})=0$, and therefore

$$\begin{aligned}
\left|\int_{\bar{\Omega}} D(\mathbf{z})d\mathbf{x}\right| &= \left|\int_{\bar{\Omega}}\overline{F(\mathbf{z})}\hat{G}(\mathtt{R}\mathbf{z})e^{2\pi i\mathbf{t}^T\mathbf{z}}d\mathbf{z}\right| \\
&= \left|\int_{\Omega'\setminus\Omega}\overline{F(\mathbf{z})}\hat{G}(\mathtt{R}\mathbf{z})e^{2\pi i\mathbf{t}^T\mathbf{z}}d\mathbf{z}\right| \quad (73)\\
&\le \sqrt{\textstyle\int_{\Omega'\setminus\Omega}|F(\mathbf{z})|^2d\mathbf{z}}\sqrt{\textstyle\int_{\Omega'\setminus\Omega}|\hat{G}(\mathbf{z})|^2d\mathbf{z}} \quad (74)\\
&= \sqrt{E_{\bar{\Omega}}(F)}\,\sqrt{E_{\Omega'\setminus\Omega}(\hat{G})}, \quad (75)
\end{aligned}$$

where $D(\mathbf{z})$=$\left(\overline{F(\mathbf{z})}\,\hat{G}(\mathtt{R}\mathbf{z}) - F^l(\mathbf{z})\,\hat{G}^l(\mathtt{R}\mathbf{z})\right)e^{2\pi i\mathbf{t}^T\mathbf{z}}$ and Ω' is the ball of radius $\frac{\sqrt{d}}{2T}$. This is equal to the second term in (39). The equality (73) holds since $F(\mathbf{z})$ is bandlimited to $\frac{1}{2T}$ in every dimension, and hence the integrand is zero outside Ω'. Last line uses the fact that $E_{\Omega'\setminus\Omega}(F) = E_{\bar{\Omega}}(F)$.

APPENDIX C
AXIS AND PLANE OF SYMMETRY

We represent a plane by its normal $\mathbf{u}$ and its distance α from the origin. The reflection of a point $\mathbf{x}$ with respect to the plane can be formulated as $\mathbf{x}' = (\mathtt{I}-2\mathbf{u}\mathbf{u}^T)(\mathbf{x}-2\alpha\mathbf{u})$. The correlation-based target function then becomes

$$Q(\mathbf{u},\alpha) = \int f(\mathbf{x})\,f\left((\mathtt{I}-2\mathbf{u}\mathbf{u}^T)(\mathbf{x}-2\alpha\mathbf{u})\right)\,d\mathbf{x}. \quad (76)$$

All the inter-resolution bounds in the registration problem works here by simply replacing g with f. One can verify this by using f, $(\mathtt{I}-2\mathbf{u}\mathbf{u}^T)$ and $-2\alpha\mathbf{u}$, respectively, instead of g, $\mathtt{R}$ and $\mathbf{t}$ in the derivations. The Lipschitz bounds can be found similarly to Sect. V-A, by first writing $Q(\mathbf{u},\alpha)$ as

$$Q(\mathbf{u},\alpha) = \int \mathbf{F}(\mathbf{z})^T\,\Gamma(-4\pi\alpha\mathbf{u}^T\mathbf{z})\,\mathbf{F}((\mathtt{I}-2\mathbf{u}\mathbf{u}^T)\mathbf{z})\,d\mathbf{z}. \quad (77)$$

Here, we just report the bounds which are obtained in a similar way to Sect. V-A:

$$\left|\frac{d}{d\alpha}Q(\mathbf{u},\alpha)\right| \le 4\pi\sum_{i=0}^{P} r_{i+1}\int_{\Omega_i}\|\mathbf{F}(\mathbf{z})\|^2\,d\mathbf{z}. \quad (78)$$

To find bounds for $\mathbf{u}$ we represent it in polar/spherical coordinates. In 2D let $\mathbf{u} = [\cos(\phi),\sin(\phi)]^T$. Then

$$\begin{aligned}
\left|\frac{d}{d\phi}Q(\mathbf{u},\alpha)\right| \le& 4\pi\alpha\sum_{i=0}^{P} r_{i+1}\,E_{\Omega_i}(F) \\
&+2\sum_{i=0}^{P}E_{\Omega_i}(F)^{1/2}\sqrt{\int_{\Omega_i}\|J_{\mathbf{F}}(\mathbf{z})\,\mathbf{z}^{\perp}\|^2\,d\mathbf{z}}, \quad (79)
\end{aligned}$$

where $\mathbf{z}^{\perp}$ was defined in (58). For 3D, we use spherical coordinates: $\mathbf{u} = [\cos(\phi)\cos(\psi), \sin(\phi)\cos(\psi), \sin(\psi)]^T$. Thus,

$$\begin{aligned}|\frac{d}{d\phi}Q(\mathbf{u},\alpha)| \leq & 4\pi\alpha\,|\cos\psi| \sum_{i=0}^{P} r_{i+1}\, E_{\Omega_i}(F) \\ & + 2\,|\cos\psi| \sum_{i=0}^{P} E_{\Omega_i}(F)^{1/2} \sqrt{\int_{\Omega_i} \|J_{\mathbf{F}}(\mathbf{z})\|^2 \|\mathbf{z}\|^2 \, d\mathbf{z}}, \quad (80)\end{aligned}$$

where $\|J_{\mathbf{F}}(\mathbf{z})\|$ denotes the spectral norm of $J_{\mathbf{F}}(\mathbf{z})$, and

$$\begin{aligned}|\frac{d}{d\psi}Q(\mathbf{u},\alpha)| \leq & 4\pi\alpha \sum_{i=0}^{P} r_{i+1}\, E_{\Omega_i}(F) \\ & + 2 \sum_{i=0}^{P} E_{\Omega_i}(F)^{1/2} \sqrt{\int_{\Omega_i} \|J_{\mathbf{F}}(\mathbf{z})\|^2 \|\mathbf{z}\|^2 \, d\mathbf{z}}. \quad (81)\end{aligned}$$

REFERENCES

[1] D. P. Huttenlocher and W. J. Rucklidge, "A multi-resolution technique for comparing images using the hausdorff distance," Cornell University, Tech. Rep., 1992.

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

[3] 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.

[4] 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*. IEEE, 2014, pp. 3930–3937.

[5] F. Pfeuffer, M. Stiglmayr, and K. Klamroth, "Discrete and geometric branch and bound algorithms for medical image registration," *Annals of Operations Research*, vol. 196, no. 1, pp. 737–765, 2012.

[6] C. Olsson, F. Kahl, and M. Oskarsson, "Branch-and-bound methods for euclidean registration problems," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 783–794, 2009.

[7] P. Hansen and B. Jaumard, "Lipschitz optimization," in *Handbook of Global Optimization*, R. Horst and P. M. Pardalos, Eds. Springer, 1995.

[8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

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

[10] M. Corvi and G. Nicchiotti, "Multiresolution image registration," in *Image Processing, 1995. Proceedings., International Conference on*, vol. 3, Oct 1995, pp. 224–227 vol.3.

[11] P. Thevenaz and M. Unser, "Optimization of mutual information for multiresolution image registration," *Image Processing, IEEE Transactions on*, vol. 9, no. 12, pp. 2083–2099, Dec 2000.

[12] F. Maes, D. Vandermeulen, and P. Suetens, "Comparative evaluation of multiresolution optimization strategies for multimodality image registration by maximization of mutual information," *Medical Image Analysis*, vol. 3, no. 4, pp. 373 – 386, 1999.

[13] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *Image Processing, IEEE Transactions on*, vol. 4, no. 1, pp. 105–107, 1995.

[14] C.-H. Lee and L.-H. Chen, "A fast motion estimation algorithm based on the block sum pyramid," *Image Processing, IEEE Transactions on*, vol. 6, no. 11, pp. 1587–1591, 1997.

[15] X. Gao, C. Duanmu, and C. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," *Image Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 501–504, 2000.

[16] Y.-S. Chen, Y.-P. Hung, and C.-S. Fuh, "Fast block matching algorithm based on the winner-update strategy," *Image Processing, IEEE Transactions on*, vol. 10, no. 8, pp. 1212–1222, 2001.

[17] L. Di Stefano and S. Mattoccia, "Fast template matching using bounded partial correlation," *Machine Vision and Applications*, vol. 13, no. 4, pp. 213–221, 2003.

[18] Y. Hel-Or and H. Hel-Or, "Real-time pattern matching using projection kernels," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 9, pp. 1430–1445, 2005.

[19] S. Mattoccia, F. Tombari, and L. D. Stefano, "Fast full-search equivalent template matching by enhanced bounded correlation," *Image Processing, IEEE Transactions on*, vol. 17, no. 4, pp. 528–538, 2008.

[20] 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*, vol. 31, no. 1, pp. 129–141, 2009.

[21] W. Ouyang and W.-K. Cham, "Fast algorithm for walsh hadamard transform on sliding windows," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 1, pp. 165–171, 2009.

[22] 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*, vol. 34, no. 1, pp. 127–143, 2012.

[23] M. Gharavi-Alkhansari, "A fast globally optimal algorithm for template matching using low-resolution pruning," *Image Processing, IEEE Transactions on*, vol. 10, no. 4, pp. 526–533, 2001.

[24] 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*. IEEE, 2013, pp. 2331–2338.

[25] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A reflective symmetry descriptor for 3d models," *Algorithmica*, vol. 38, no. 1, pp. 201–225, 2004.

[26] 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*, vol. 1, 2014.

[27] 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*. IEEE, 2007, pp. 11–16.

[28] J. Yao, J. E. Burns, H. Munoz, and R. M. Summers, "Detection of vertebral body fractures based on cortical shell unwrapping," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*. Springer, 2012, pp. 509–516.

Behrooz Nasihatkon Behrooz Nasihatkon received PhD degree from the Australian National University in 2014. His thesis was about generalizing the theory of projective reconstruction in computer vision for the design and analysis of projective reconstruction algorithms. He then worked as a postdoctoral fellow at Chalmers University of Technology, Sweden, developing image registration techniques with applications in medical image analysis. He then did a short postdoc at the Sharif University of Technology in Iran, before starting to work as an assistance professor at the Khajeh Nasir University of Technology, Tehran, Iran.

Fredrik Kahl received his MSc degree in computer science and technology in 1995 and his PhD in mathematics in 2001. His thesis was awarded the Best Nordic Thesis Award in pattern recognition and image analysis 2001-2002 at the Scandinavian Conference on Image Analysis 2003. He is currently a Professor both at the Centre for Mathematical Sciences, Lund University and at the Image Analysis and Computer Vision Group at Chalmers University.