Global Optimization for Image Registration

James M. Calvin[1,b)], Craig J. Gotsman[1,c)] and Cuicui Zheng[1,a)]

[1] *Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, 07102, USA*

[a)] Corresponding author: cz296@njit.edu
[b)] calvin@njit.edu
[c)] gotsman@njit.edu

Abstract. Image processing tasks often require optimizing over some set of parameters. In the image registration problem, one attempts to determine the best transformation for aligning similar images. Such problems typically require minimizing a dissimilarity measure with multiple local minima. We describe a global optimization algorithm and apply it to the problem of identifying the best transformation for aligning two images.

INTRODUCTION

Image registration is the process of identifying a transformation that best aligns a *moving image* with a *fixed image*. We take as input two matrices of pixel intensities, possibly of different sizes. After interpolating the pixel values, we can view the fixed and moving images as mappings $I_f : \Omega_f \to \mathbb{R}$ and $I_m : \Omega_m \to \mathbb{R}$, respectively, where Ω_f and Ω_m are suitable rectangles in $\mathbb{R}^2$. (We assume grey-scale images.) We seek a transformation $T : \Omega_f \to \Omega_m$ such that $I_m \circ T$ is "close to" I_f. To quantify the notion of closeness we adopt a dissimilarity measure C and choose T to minimize $C(T; I_f, I_m)$. To obtain a tractable optimization problem, we restrict the class of transformations. In this paper we consider two classes of transformation: two-dimensional translation, and translation combined with rotation.

By suitable scaling, we consider the registration problem to be a global optimization problem over a parameter space $[0, 1]^d$, where in this paper we take $d = 2$ (two translation parameters) or $d = 3$ (two translation parameters and one rotation parameter). Many registration methods require the specification of a "starting point" in parameter space, from which an iterative local optimization method proceeds. Most of these methods are sensitive to this starting point, and if this point is not close enough to the global minimum, the method can get "stuck" in a local minimum. We are interested in methods that are automatic in the sense that no "starting point" or other parameters are required from the user.

Local optimization algorithms are often used in image registration. In [1] the authors examined the application of global optimization approaches, showing that local optimization schemes often did not find the optimum in their test medical imagery.

Other approaches to image registration include correlation-based methods, mutual information-based methods, and soft computing-based methods which include artificial neural networks, fuzzy sets and some optimization heuristics [2]. The main dissimilarity measures used are mutual information and sum of squared pixel differences. The use of mutual information as a dissimilarity measure is surveyed in [3]. Researchers have attempted to improve mutual information with other information-theoretic measures in [4].

We do not address the problems of the choice of dissimilarity measure or permissible transformations. Rather, we focus on the problem of optimizing the dissimilarity measure over the parameter space representing the set of possible transformations. We specifically address the issue of *automatic* image registration, where no special domain knowledge is required (for example to identify landmarks), and no tuning of input parameters is required.

The Algorithm

The algorithm subdivides the rectangular parameter space into subrectangles and sequentially chooses the next subrectangle to subdivide according to a certain numerical value. We use the rectangular subdivision introduced in [5]. The criterion for selecting the next subrectangle to subdivide is similar to the criterion used in [6]. This is based solely on evaluations of the cost function at selected points in the parameter space.

The algorithm operates by decomposing $[0, 1]^d$ into hyper-rectangles as follows. Denote by f the cost function to be minimized. The first operation is to evaluate f at the center of the unit hyper-cube $(1/2, 1/2, \ldots, 1/2)$. Given a current decomposition, choose one of the hyper-rectangles (according to the maximal value of a criterion to be defined below) and trisect it along the longest axis. The central sub-hyperrectangle retains its central function value, while the function is evaluated at the centers of the two other hyperrectangles.

After n iterations of the algorithm, f will have been evaluated $2n + 1$ times and the unit hyperrectangle will have been decomposed into $2n + 1$ sub-hyperrectangles.

Define

$$g_n(x) \equiv d\,(x \log(n))^{2/d}\,,$$

for $0 < x \leq 1$. Note that g_n is increasing and $g_n(x) \downarrow 0$ as $x \downarrow 0$.

It will be convenient to index quantities to be defined by the iteration number of the algorithm, which we denote by n, instead of the number of function evaluations denoted by $N = 2n + 1$.

After n iterations of the algorithm, there are $2n + 1$ rectangles. Let c_i be the center of rectangle R_i. Let $M_n = \min_{1\leq i\leq N} f(c_i)$ and denote the error by $\Delta_n = M_n - f(x^*)$ where x^* is a global optimizer of f. Let $v_i = |R_i|$ be the area (volume) of the ith rectangle, and set $v_n = \min_{1\leq i\leq N} v_i$.

Our algorithm will assign a numerical value to each hyper-rectangle in the subdivision. After n iterations of the algorithm, for each hyper-rectangle R_i, $1 \leq i \leq 2n + 1$, set

$$\rho_i^n \equiv \frac{|R_i|^{2/d}}{f(c_i) - M_n + g_n(v_n)}. \tag{1}$$

The choice of g_n and ρ_i^n is based on efficiency considerations; g_n must be large enough to ensure convergence yet small enough to lead to an efficient search.

A more formal description of the algorithm follows. In the description, N is the total number of function evaluations to make and j is the number of function evaluations that have been made. Let v, the volume of the smallest hyper-rectangle, take initial value 1 and let M, the minimum of the observed function values, take initial value $f(1/2, \ldots, 1/2)$.

The algorithm comprises the following steps.

1. We start with the hyper-rectangle $[0, 1]^d$ and with the function value at the center. Set j, the number of function evaluations (=number of hyper-rectangles), to 1.
2. For each hyper-rectangle R_i in the current collection $\{R_1, R_2, \ldots, R_j\}$, compute ρ_i^j, keeping track of the rectangle with the lowest index γ that has the maximal value of ρ_i^j.
3. From the hyper-rectangle with maximal value ρ_γ^j form three new hyper-rectangles as follows. Suppose that

   $$R_\gamma = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_d, b_d],$$

   and that k is the smallest index with $b_k - a_k \geq b_i - a_i$ for $1 \leq i \leq d$. The three new hyper-rectangles are
   (a) $R'_\gamma \equiv [a_1, b_1] \times \cdots \times [a_k, a_k + (a_k + b_k)/3] \times \cdots \times [a_d, b_d]$
   (b) $R''_\gamma \equiv [a_1, b_1] \times \cdots \times [a_k + (a_k + b_k)/3, a_k + 2(a_k + b_k)/3] \times \cdots \times [a_d, b_d]$
   (c) $R'''_\gamma \equiv [a_1, b_1] \times \cdots \times [a_k + 2(a_k + b_k)/3, b_k] \times \cdots \times [a_d, b_d]$.
   Evaluate f at the centers of R' and R''.
   If $\left|R_\gamma\right| = v$, then set $v \leftarrow v/3$, and if the smallest of the new function values is less than the previously observed minimum M, then set M to that new smallest value. Increment j by the number of new function evaluations $j \leftarrow j + 2$.
4. If $j < N$, return to step 2.

The only stopping rule that we consider is to initially specify a fixed number of function evaluations after which the algorithm terminates. The input to our problem is discrete, but by interpolating the pixel intensities we obtain a continuous optimization problem. Our main result follows.

Theorem 1: If f is continuous, then the error $\Delta_n = M_n - f(x^*)$ converges to 0 when $n \to \infty$, where $M_n = \min_{1 \leq i \leq n} f(c_i)$.

The algorithm described above uses only function values and no derivatives. After the algorithm terminates, it is natural to use a local optimization method, such as gradient descent, to improve on the terminal value. We continue to assume that only function values are available, and so finite-difference gradient approximations are used in the local search methods.

Numerical Experiments

We implemented the optimization algorithm for the 2-dimensional and 3-dimensional image registration problems in Matlab. When testing our algorithm, we found the performance improved by using the following modified definition of the selection criterion:

$$\rho_i^n \equiv \frac{|R_i|^{2/d} \log\left(1 + \frac{f(c_i) - M_n}{g_n(v_n)}\right)^{2/d}}{(f(c_i) - M_n + g_n(v_n))^{4/d}}, \qquad (2)$$

where we define $g_n(x)$ as

$$g_n(x) = 4x^{\frac{1}{2}} \left(d \log\left(\frac{3}{x^{2/d}}\right)\right)^{d/4}.$$

The two-dimensional image registration problem is to find the best translation (left to right and up-down), while for the three-dimensional registration problem we consider rotation in addition to translation of the image. The cost function f we used is the sum of squared image pixel intensity differences, which for three-dimensional image registration is:

$$f(x, y, r) = \sum_{i=1}^{i=t_m} \sum_{j=1}^{j=t_n} [p_1(i, j) - T(p_2(i + (x - 0.5)t_m, j + (y - 0.5)t_n), (r - 0.5) \cdot 2\pi)]^2 \qquad (3)$$

where t_m is the minimum width of the input images, t_n is the minimum height of the input images, and $T(\cdot)$ represents the transformation (translation and rotation). Figure 1 shows the cost surface for a two-dimensional section of the three-dimensional cost function (varying x and y translations with rotation fixed) for the example image used in our experiment. The many local minima are quite evident.

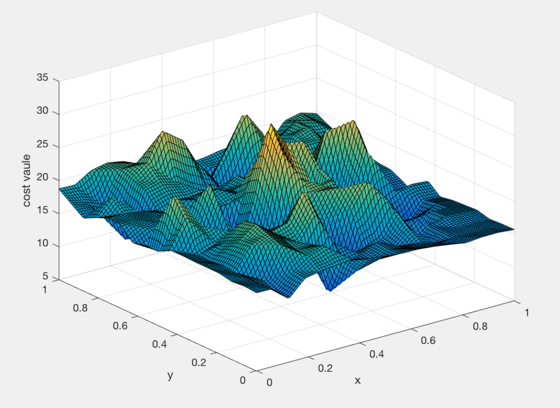

FIGURE 1: Cost function with fixed rotation.

We compared our global optimization algorithm with the following alternatives: 1) An exhaustive search algorithm (searching over a fixed equi-spaced grid of points), followed by gradient descent from the best point; 2) The Matlab built-in simulated annealing algorithm [7], followed by gradient descent from the best point.

We performed 4,000 iterations, followed by gradient descent, for all algorithms because that resulted in a reasonable visual match of the registered image. In Figure 2, the x-axis indicates the smallest cost value obtained by the algorithm and the y-axis indicates the proportion of runs for which the algorithm achieved the cost value.

Simulated annealing is a randomized algorithm with considerable variability in the minimal cost obtained. To get a sense of the range of solutions we ran the algorithm 100 times independently and plotted the empirical cumulative distribution function of the results as the blue line in Figure 2. The other two algorithms are deterministic. For the algorithm we proposed, we randomized it by applying a random perturbation to the starting point. For each of 100 independent replications we started with the center translated by a uniformly distributed offset in $[-0.05, 0.05]^3$. For exhaustive search,we used a three-dimensional grid of $16^3 = 4096$ points, and we randomly perturbed the grid by a uniformly distributed offset in $[-0.03125, 0.03125]^3$ for each of 100 independent replications.

Our algorithm, named "rectangle+grad" in Figure 2, obtained the smallest average cost value of 1.04; the range is shown by the red star line in Figure 2. The exhaustive search algorithm, which obtained a smallest cost value around 2.84, performed as indicated by the green star line. Simulated annealing, with an average cost over 3, performed as shown by the blue star line in Figure 2. Because we repeated simulated annealing 100 times independently (with 4,000 iterations on each replication), there were several runs on which simulated annealing worked really well but on average the simulated annealing algorithm obtained worse results than the proposed algorithm.

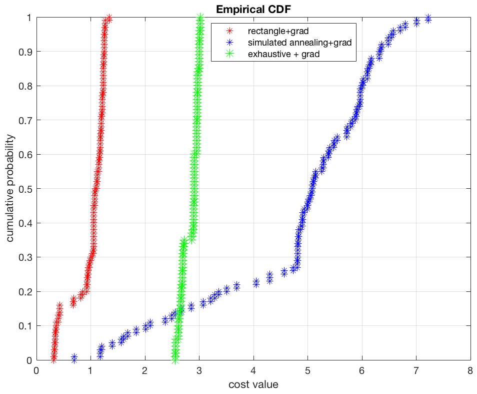

FIGURE 2: Distribution of cost values obtained by three algorithms after 4,000 iterations.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. CMMI-1562466.

REFERENCES

[1] M. Jenkinson and S. Smith, Medical Image Analysis **5**, 142–156 (2001).
[2] S. Nag, arXiv:1712.07540 (2017).
[3] J. P. Pluim, J. B. Maintz, and M. A. Viergever, IEEE Trans Medical Imaging **22**, 986–1004 (2003).
[4] R. D. Eastman, N. S. Netanyahu, and J. le Moigne, Image Registration for Remote Sensing 35–78 (2011).
[5] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, Journal of Optimization Theory and Application **79**, 157–181 (1993).
[6] J. M. Calvin, G. Gimbutienė, W. O. Phillips, and A. Žilinskas, Journal of Global Optimization **71**, 165–191 (2018).
[7] C. Zhang and H. P. Wang, Engineering Optimization **21**, 277–291 (1993).