

Accelerated Least Squares Contour Alignment with a Filtering Process

Jie Yan and Wu-Sheng Lu

Department of Electrical and Computer Engineering

University of Victoria

Victoria, BC, Canada V8W 3P6

jyan@ece.uvic.ca, wslu@ece.uvic.ca

Abstract—A new algorithm for fast contour alignment of digital images is presented. The algorithm is based on a filter-bank approach in conjunction with a closed-form solution procedure for least-squares subproblems involved. Simulation results are presented, which demonstrate that the proposed algorithm achieves globally optimal solution and requires a computational complexity of $O(N \log N)$ compared to the complexity of $O(N^2)$ required by the existing algorithm.

I. INTRODUCTION

Shape analysis plays a key role in many computer vision systems. To represent the boundary of an object, several well-known models are proposed in the literature, including parametric models, active contours, differential equations and Fourier descriptors [1], [2]. One simple way to model an object contour consists of forming a sequence of N points. Let $\mathcal{P}$ denote the sequence points of the model contour $\mathcal{C}_1$, and $\mathcal{Q}$ the sequence of detected points of object contour $\mathcal{C}_2$, the problem arises when one tries to match $\mathcal{C}_2$ to $\mathcal{C}_1$ under affine transformation of detected contour points in $\mathcal{Q}$, i.e., scaling, rotation and translation of $\mathcal{C}_2$. The estimation of affine transformation parameters to match the object contour to the model contour has been extensively studied. One widely employed scheme is to formulate the problem as a nonlinear minimization problem in an explicit (see, e.g. [3]–[7]) or an implicit (see, e.g., [8]–[10]) framework, and solve the problem for local optimum under additional initial approximation. In [11] and [12], a nonlinear change of variables that transforms the original nonlinear optimization into a least-squares problem is proposed. The global minimum that corresponds to the best transformation can therefore be readily obtained. Note that this global solution is obtained under the assumption that the two sequences of points are with valid correspondence. Unfortunately, this assumption does not hold in many applications. In [12], the point correspondence problem was addressed by solving the least-squares problems N times which becomes extremely time consuming when the number of points in $\mathcal{P}$ and $\mathcal{Q}$ is large.

In this paper, we present an accelerated algorithm to solve the contour alignment problem in least-squares sense. The major contribution of the paper is the development of a filter-bank based approach to the point correspondence problem. Instead of solving N least-squares problems, in our approach two length- $2N$ sequences are filtered by a two-channel FIR filter bank whose output leads to the solution of the correspondence problem at hand. The filtering process considerably accelerates the solution procedure as it can be carried out by fast

convolutions leading to a reduced complexity of $O(N \log N)$ compared to $O(N^2)$ required by N least squares. In addition, closed-form formulas for inverting the Hessian matrix involved in the least-squares solution are derived to facilitate a fast solution procedure for the least-squares subproblems. Experimental results are presented to demonstrate the performance of the proposed method in comparison with the existing contour alignment algorithm.

II. PROBLEM STATEMENT

The contour $\mathcal{C}$ of an image in $\mathbb{R}^2$ can be represented by a set of N points denoted by

$$\mathbf{p}_i \in \mathcal{C} \subset \mathbb{R}^2 \quad \text{for } i = 1, \dots, N.$$

An efficient data structure to specify contour $\mathcal{C}$ is to align these N points next to each other so as to form a matrix $\mathbf{P} \in \mathbb{R}^{2 \times N}$ as

$$\mathbf{P} = [\mathbf{p}_1 \ \mathbf{p}_2 \ \dots \ \mathbf{p}_N] = \begin{bmatrix} p_{1x} & p_{2x} & \dots & p_{Nx} \\ p_{1y} & p_{2y} & \dots & p_{Ny} \end{bmatrix}$$

Suppose we are given two contours $\mathcal{C}_1$ and $\mathcal{C}_2$ which are represented by matrices $\mathbf{P} \in \mathbb{R}^{2 \times N}$ and $\mathbf{Q} \in \mathbb{R}^{2 \times N}$, respectively, and we want to examine the degree of similarity between the two contours. Obviously, this is a problem of comparing two point patterns, which involves two issues: (1) the point sequences given in $\mathbf{P}$ and $\mathbf{Q}$ may not be arranged correspondingly, in such a case it has little meaning to compare the i th point $\mathbf{p}_i$ in $\mathbf{P}$ with the i th point $\mathbf{q}_i$ in $\mathbf{Q}$. In other words, one needs to develop a computationally tractable method to generate a circularly shifted version of $\mathcal{C}_1$ for point correspondence with respect to contour $\mathcal{C}_2$. (2) With the point correspondence in place, one needs to define a quantitative measure for the assessment of similarity between the two contours, and develop a method for efficient numerical computation of this measure. In the rest of the paper, we shall refer the solution procedure that addresses the above issues to as *contour alignment*. We shall address the second issue in the rest of this section and Sec. III. The problem of finding point correspondence between two given contours turns out to be more challenging, and we shall address it in Sec. IV.

Assume we are given two contours $\mathcal{C}_1$ and $\mathcal{C}_2$ such that the point pattern $\mathbf{P}$ for $\mathcal{C}_1$ corresponds to the point pattern $\mathbf{Q}$ for $\mathcal{C}_2$. Typically, a common-sense similarity between two point patterns should be independent of their sizes, orientations and

locations. For this reason, it is only reasonable to quantitatively compare the two point patterns after one of the point patterns (say, $\mathbf{Q}$) is appropriately scaled, rotated and translated so as to produce a transformed version of the point pattern that is as close as possible to the other point pattern. A similarity measure can then be readily defined as a matrix norm of the difference between the transformed matrix $\mathbf{Q}$ and matrix $\mathbf{P}$ (see (1) below).

Consider a point $\mathbf{q} = [q_x \ q_y]^T$. A transformation that performs a scaling $s > 0$, a counterclockwise rotation by $\theta \in [-\pi, \pi)$ and a translation by a vector $\mathbf{a}$ of point $\mathbf{q}$ can be described by operator $\tau_{s,\theta,\mathbf{a}}$ as

$$\tau_{s,\theta,\mathbf{a}}(\mathbf{q}) = s\mathbf{R}(\theta)\mathbf{q} + \mathbf{a}$$

where the rotation matrix

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

is orthogonal. Applying the transformation to a point pattern $\mathbf{Q}$ leads to a point pattern $\tau_{s,\theta,\mathbf{a}}(\mathbf{Q})$ that is “equivalent” to point pattern $\mathbf{Q}$ in the sense that these two patterns (i.e., image contours) are similar to each other. A reasonable measure of similarity between two point patterns (contours) $\mathbf{P}$ and $\mathbf{Q}$ that are assumed to have point correspondence is given by

$$\mathcal{S}(\mathbf{P}, \mathbf{Q}) = \min_{s,\theta,\mathbf{a}} \|\mathbf{P} - \tau_{s,\theta,\mathbf{a}}(\mathbf{Q})\|_F \quad (1)$$

where $s > 0$, $\theta \in [-\pi, \pi)$, and $\|\cdot\|_F$ denotes the Frobenius norm. The problem in (1) in terms of variables s , θ and $\mathbf{a} = [a_x \ a_y]^T$ is highly nonlinear because of the presence of sinusoidal functions in the rotation matrix. In [11], [12], this problem is addressed by variable changes from (s, θ) to (b_x, b_y) where $b_x = s \cos \theta$ and $b_y = s \sin \theta$ so that problem (1) becomes

$$\mathcal{S}(\mathbf{P}, \mathbf{Q}) = \min_{\mathbf{x}} \sum_{i=1}^N \|\begin{bmatrix} \mathbf{R}_i & \mathbf{I} \end{bmatrix} \mathbf{x} - \mathbf{p}_i\|^2 \quad (2)$$

where

$$\mathbf{R}_i = \begin{bmatrix} q_{ix} & -q_{iy} \\ q_{iy} & q_{ix} \end{bmatrix}, \quad \mathbf{p}_i = \begin{bmatrix} p_{ix} \\ p_{iy} \end{bmatrix}, \quad \mathbf{x} = [b_x \ b_y \ a_x \ a_y]^T$$

III. A LEAST-SQUARE SOLUTION OF PROBLEM (2)

The objective function in (2) can be further expressed as

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{H} \mathbf{x} - 2\mathbf{x}^T \mathbf{r} + \kappa_1 \quad (3)$$

where

$$\mathbf{H} = \begin{bmatrix} \sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i & \sum_{i=1}^N \mathbf{R}_i^T \\ \sum_{i=1}^N \mathbf{R}_i & N\mathbf{I} \end{bmatrix} \quad (4)$$

is positive definite, and

$$\mathbf{r} = \sum_{i=1}^N \begin{bmatrix} \mathbf{R}_i^T \mathbf{p}_i \\ \mathbf{p}_i \end{bmatrix}, \quad \kappa_1 = \sum_{i=1}^N \mathbf{p}_i^T \mathbf{p}_i$$

As a result, the unique global minimizer of $f(\mathbf{x})$ is given by

$$\mathbf{x}^* = \mathbf{H}^{-1} \mathbf{r} \quad (5)$$

and the minimum of $f(\mathbf{x})$ at $\mathbf{x}^*$ is found to be

$$\begin{aligned} f(\mathbf{x}^*) &= -\mathbf{r}^T \mathbf{H}^{-1} \mathbf{r} + \kappa_1 \\ &= \kappa_1 - \begin{bmatrix} \sum_{i=1}^N \mathbf{p}_i^T \mathbf{R}_i & \mathbf{p}_a^T \end{bmatrix} \mathbf{H}^{-1} \begin{bmatrix} \sum_{i=1}^N \mathbf{R}_i^T \mathbf{p}_i \\ \mathbf{p}_a \end{bmatrix} \end{aligned} \quad (6)$$

where $\mathbf{p}_a = \sum_{i=1}^N \mathbf{p}_i$. It turns out that efficient evaluation of $f(\mathbf{x}^*)$ is of critical importance in solving the point correspondence problem which will be addressed in Sec. IV. The difficulty to evaluate $f(\mathbf{x}^*)$ in (6) is the computation of $\mathbf{H}^{-1}$. Since $\mathbf{H}^{-1}$ is symmetric, we can write

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \quad (7)$$

Explicit expressions for matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are provided in the Appendix.

IV. A FILTERING APPROACH TO FINDING POINT CORRESPONDENCE

A. The approach

As argued earlier, in many applications the point patterns $\mathbf{P}$ and $\mathbf{Q}$, that represent contours $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, do not hold point correspondence. In [12], the point correspondence was tackled by solving N least-squares problems where each one minimizes the similarity measure between a circularly shifted version of point pattern $\mathbf{P}$ and point pattern $\mathbf{Q}$. The procedure requires a total of N minimizations, hence a complexity of $O(N^2)$. Below we present a different approach to the problem based on digital filtering.

Let $\{\mathbf{p}_{i(k)}\}$ denote the set of sequential points generated from the point sequence $\mathbf{P}$ for contour $\mathcal{C}_1$ by a circular shift by k so that the i th point $\mathbf{p}_{i(k)}$ in the shifted point sequence comes from the $i(k)$ th point in $\mathbf{P}$ where $i(k) = N - (N - i + k) \bmod N$. From Secs. II and III, it follows that the similarity between a circularly shifted point pattern $\{\mathbf{p}_{i(k)}\}$ and point pattern $\{\mathbf{q}_i\}$ is given by

$$f(\mathbf{x}_k^*) = \kappa_1 - \begin{bmatrix} \sum_{i=1}^N \mathbf{p}_{i(k)}^T \mathbf{R}_i & \mathbf{p}_a^T \end{bmatrix} \mathbf{H}^{-1} \begin{bmatrix} \sum_{i=1}^N \mathbf{R}_i^T \mathbf{p}_{i(k)} \\ \mathbf{p}_a \end{bmatrix} \quad (8)$$

with $k = 0, 1, \dots, N-1$. Note that formula (6) is a special case of (8) with $k = 0$. The filtering approach described below avoids computing $f(\mathbf{x}_k^*)$ in (8) N times.

Let

$$\mathbf{u}(k) = \sum_{i=1}^N \mathbf{R}_i^T \mathbf{p}_{i(k)} \quad (9)$$

By (7), we can write

$$f(\mathbf{x}_k^*) = -[\mathbf{u}^T(k) \mathbf{A} \mathbf{u}(k) + 2\mathbf{u}^T(k) \mathbf{B} \mathbf{p}_a] + \kappa_2$$

where $\kappa_2 = \kappa_1 - \mathbf{p}_a^T \mathbf{C} \mathbf{p}_a$. Since $\mathbf{A}$ is a scaled identity matrix, we have

$$f(\mathbf{x}_k^*) = -[\gamma_4 \mathbf{u}^T(k) \mathbf{u}(k) + 2\mathbf{u}^T(k) \mathbf{B} \mathbf{p}_a] + \kappa_2 \quad (10)$$

It follows that if $\{\mathbf{u}(k), k = 0, 1, \dots, N\}$ is evaluated, then the minimum of $f(\mathbf{x}_k^*)$ with respect to k can readily be identified. Hence the problem that remains to be resolved is how to

evaluate $\{\mathbf{u}(k)\}$ quickly. Let $\mathbf{u}(k) = [u_{kx} \ u_{ky}]^T$, from (9) we write

$$\mathbf{u}(k) = \sum_{i=1}^N \mathbf{R}_i^T \mathbf{p}_{i(k)} = \sum_{i=1}^N \begin{bmatrix} q_{ix} & q_{iy} \\ -q_{iy} & q_{ix} \end{bmatrix} \begin{bmatrix} p_{i(k)x} \\ p_{i(k)y} \end{bmatrix} \quad (11)$$

which leads to

$$u_{kx} = \sum_{i=1}^N (q_{ix} \cdot p_{i(k)x} + q_{iy} \cdot p_{i(k)y}) \quad (12a)$$

$$u_{ky} = \sum_{i=1}^N (-q_{iy} \cdot p_{i(k)x} + q_{ix} \cdot p_{i(k)y}) \quad (12b)$$

A two-channel filter bank can now be built based on the formulas in (12). Referring to Fig. 1 where $H_1(z)$ and $H_2(z)$ are the transfer functions of two length- N FIR digital filters given by

$$\begin{aligned} H_1(z) &= q_{1x} + q_{2x}z^{-1} + \dots + q_{Nx}z^{-(N-1)} \\ H_2(z) &= q_{1y} + q_{2y}z^{-1} + \dots + q_{Ny}z^{-(N-1)} \end{aligned}$$

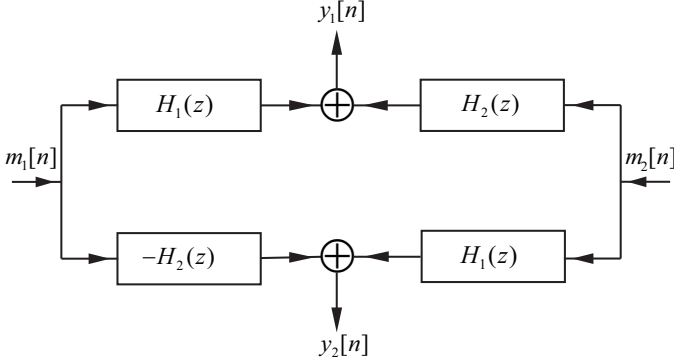


Fig. 1: A two-channel filter bank that generates $\mathbf{u}(k)$

The input sequences to the filter bank are given by

$$\begin{aligned} \{m_1[n]\} &= \{p_{Nx}, \dots, p_{2x}, p_{1x}, p_{Nx}, \dots, p_{2x}, p_{1x}\} \\ \{m_2[n]\} &= \{p_{Ny}, \dots, p_{2y}, p_{1y}, p_{Ny}, \dots, p_{2y}, p_{1y}\} \end{aligned}$$

It is rather straightforward to verify that sequence $\{\mathbf{u}(k), k = 0, 1, \dots, N-1\}$ as the output sequences $y_1[n]$ and $y_2[n]$ at time instances from $n = N-1$ to $n = 2N-2$, namely,

$$\mathbf{u}(k) = \begin{bmatrix} u_{kx} \\ u_{ky} \end{bmatrix} = \begin{bmatrix} y_1[k+N-1] \\ y_2[k+N-1] \end{bmatrix}, \text{ for } k = 0, 1, \dots, N-1$$

In words, sequence $\{\mathbf{u}(k)\}$ can be regarded as an excerpt of the output from the two-channel filter bank in Fig. 1. With $\{\mathbf{u}(k)\}$ obtained, (10) can be used to identify an index k^* at which $f(\mathbf{x}_{k^*}^*)$ achieves its minimum. The identified index k^* corresponds to an optimal circular shift for contour $\mathcal{C}_1$ in order to make the two point patterns correspondent with each other.

We remark that, for efficient implementation, $\{m_1[n]\}$ and $\{m_2[n]\}$ can be concatenated as one single sequence $\{m[n]\} = \{m_1[0], \dots, m_1[2N-1], m_2[0], \dots, m_2[2N-1]\}$. In this way, $\{m[n]\}$ needs to be filtered by $H_1(z)$ and $H_2(z)$ only once.

B. Computational complexity

The major computational cost in solving the contour alignment problem is spent on computing $\mathbf{u}(k)$ for $k = 0, 1, \dots, N$. The method in [12] basically obtains $\mathbf{u}(k)$ by computing Eq. (8) N times. Assume that matrices $\mathbf{R}_i$ in (2), $\mathbf{H}^{-1}$ in (7) and vector $\mathbf{p}_a$ hence vector $\mathbf{B}\mathbf{p}_a$ are pre-calculated, it follows from (9) and (10) that for each shift k , computing the value of optimized objective function $f(\mathbf{x}_{k^*}^*)$ requires $O(N)$ multiplications. Consequently, the complexity of the method in [12] is $O(N^2)$.

On the other hand, the computational cost of the proposed algorithm is mainly due to the convolutions of a length- $4N$ sequence with two FIR filters of length N , which require $O(N \log N)$ multiplications when the convolutions are performed using FFT.

V. EXPERIMENTAL RESULTS AND COMPARISON

A. Performance evaluation

The proposed algorithm was applied to two images taken from [8] and [12], see the first row in Fig. 2. Canny's edge detector was utilized to detect the images' edges, see the second row in Fig. 2. For each image, a sequence of $N = 500$ points was produced starting from an arbitrary point on the edge and collecting the subsequent points by sampling the entire image edge.

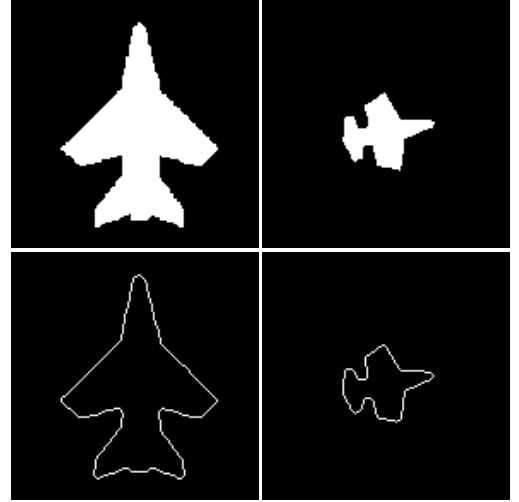


Fig. 2: Images in the 1st row: two jets, images in the 2nd row: edges of the above jets.

The contour matching results for the two jet images are illustrated in Fig. 3. In the figure, two original contours were marked in red and black, respectively. Applying the proposed algorithm, we were able to compute the optimal scaling factor s^* , rotation angle θ^* , and translation vector $\mathbf{a}^*$ that produce the minimum objective function of problem (1) for all point-shifts. The blue contour was obtained by applying the transformation operator $\tau_{s^*, \theta^*, \mathbf{a}^*}$ to the black contour. It can be observed that the blue contour well matches the red one. All three contours were superimposed in the same plot for visual inspection where the first matching points were marked in green circles. Fig. 4 displays the minimum objective function values achieved with

different circularly shifted point sequences. It was found that the global minimum was achieved at $k^* = 397$, as marked with a green circle.

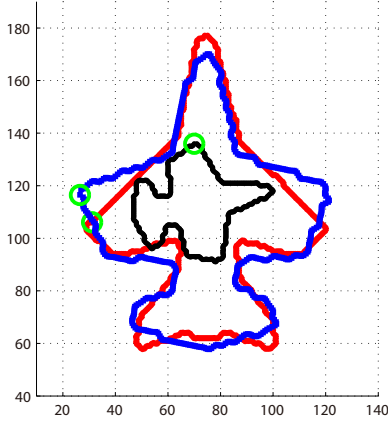


Fig. 3: Contour alignment for two jets in Fig. 2. Red: model contour C_1 ; Black: object contour C_2 ; Blue: matched contour with C_1 by transforming C_2 .

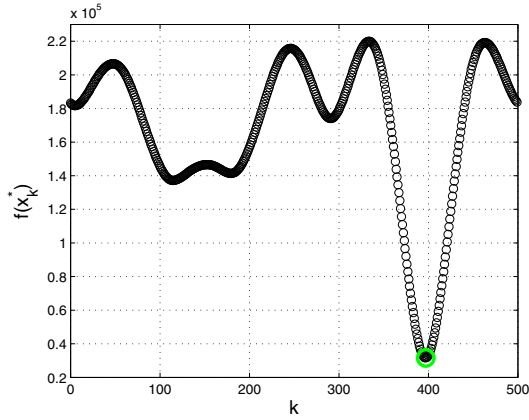


Fig. 4: Minimum objective function $f(\mathbf{x}_k^*)$ evaluated at different shifts. Global minimum achieved at the optimal shift index $k^* = 397$ is marked in green

The proposed algorithm was also applied to four images depicted in Fig. 5 in order to align the contour of each of these images with that of the upper left image in Fig. 2. The contour alignment results are illustrated in Fig. 6.

B. Comparison of computational complexity

In this section, we compare the computational time required by the proposed filter-bank based algorithm with the method in [12] where problem (1) was minimized a total of N times for N shifts. Fig. 7 displays the computational time required by the two methods versus contour length N , where the two methods were implemented in MATLAB run on a PC laptop with dual 2.67GHz CPU. For the proposed algorithm, the filtering process was carried out by MATLAB function `filter` which uses FFT for fast convolution.

The simulations show that for an N as large as 10^5 , the method in [12] requires 29 minutes to complete, whereas the

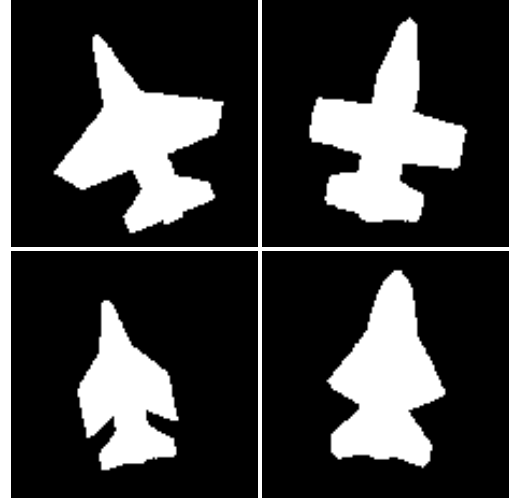


Fig. 5: Four jet images

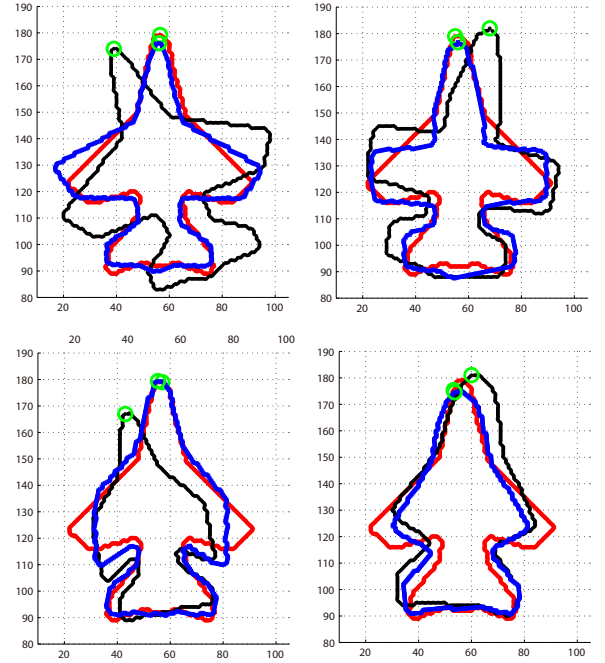


Fig. 6: Contour alignment for the four images in Fig. 5

proposed method requires approximately 47 seconds, indicating a significant improvement in favor of the proposed method.

VI. CONCLUSIONS

An acceleration scheme to the least-squares contour alignment problem has been proposed where a filter-bank structure is incorporated for fast computation of global minimum of the similarity measure between object contour and N possible shifts of the model contour. A closed-form solution for least-squares subproblems is developed for efficient computation. The proposed algorithm has a computational complexity $O(N \log N)$ and is practical for contour alignment problems of large scale.

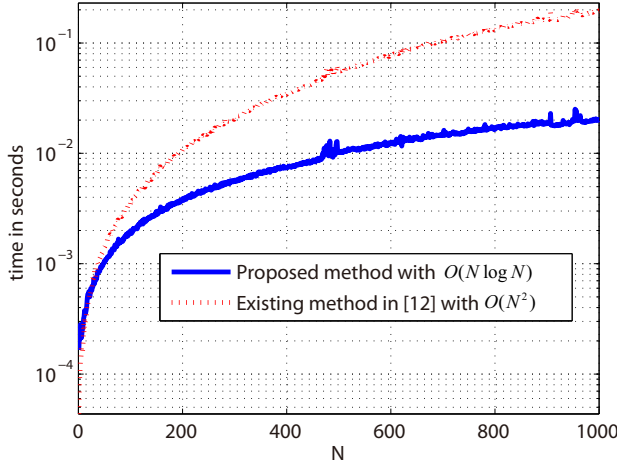


Fig. 7: Comparison of computational complexity

ACKNOWLEDGMENT

The authors are grateful to the Natural Sciences and Engineering Research Council of Canada for supporting this research.

VII. APPENDIX: COMPUTATION OF $\mathbf{H}^{-1}$

The expression of $\mathbf{H} \in \mathbb{R}^{4 \times 4}$ is given in (4). Based on the equality condition $\mathbf{H}\mathbf{H}^{-1} = \mathbf{I}$ and Eq. (7), a set of four linear equations involving matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are produced as

$$\sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i \cdot \mathbf{A} + \sum_{i=1}^N \mathbf{R}_i^T \cdot \mathbf{B}^T = \mathbf{I} \quad (15a)$$

$$\sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i \cdot \mathbf{B} + \sum_{i=1}^N \mathbf{R}_i^T \cdot \mathbf{C} = \mathbf{0} \quad (15b)$$

$$\sum_{i=1}^N \mathbf{R}_i \cdot \mathbf{A} + N \cdot \mathbf{B}^T = \mathbf{0} \quad (15c)$$

$$\sum_{i=1}^N \mathbf{R}_i \cdot \mathbf{B} + N \cdot \mathbf{C} = \mathbf{I} \quad (15d)$$

From (15b) and (15d), we derive an linear equation for $\mathbf{B}$

$$\left[\sum_{i=1}^N \mathbf{R}_i^T \mathbf{R}_i - \frac{1}{N} \sum_{i=1}^N \mathbf{R}_i^T \sum_{i=1}^N \mathbf{R}_i \right] \mathbf{B} = -\frac{1}{N} \sum_{i=1}^N \mathbf{R}_i^T \quad (16)$$

Define

$$\gamma_1 = \sum_{i=1}^N q_{ix}, \quad \gamma_2 = \sum_{i=1}^N q_{iy} \quad (17)$$

such that

$$\sum_{i=1}^N \mathbf{R}_i = \begin{bmatrix} \gamma_1 & -\gamma_2 \\ \gamma_2 & \gamma_1 \end{bmatrix}$$

Eq. (19) can be further cast as

$$\left[\sum_{i=1}^N (q_{ix}^2 + q_{iy}^2) - \frac{\gamma_1^2 + \gamma_2^2}{N} \right] \mathbf{B} = -\frac{1}{N} \begin{bmatrix} \gamma_1 & \gamma_2 \\ -\gamma_2 & \gamma_1 \end{bmatrix}$$

Therefore with the denotations of

$$\gamma_3 = \gamma_1^2 + \gamma_2^2, \quad \gamma_4 = \frac{1}{\sum_{i=1}^N (q_{ix}^2 + q_{iy}^2) - \gamma_3/N} \quad (18)$$

we have the expression for $\mathbf{B}$ as

$$\mathbf{B} = -\frac{\gamma_4}{N} \begin{bmatrix} \gamma_1 & \gamma_2 \\ -\gamma_2 & \gamma_1 \end{bmatrix} \quad (19)$$

By substituting $\mathbf{B}$ into (15d), matrix $\mathbf{C}$ can be readily computed as

$$\mathbf{C} = \frac{1}{N} \left(1 + \frac{\gamma_3 \gamma_4}{N} \right) \mathbf{I} \quad (20)$$

Finally, we compute $\mathbf{A}$ by substituting $\mathbf{B}$ into (15a) and obtain

$$\mathbf{A} = \gamma_4 \mathbf{I} \quad (21)$$

REFERENCES

- [1] P. J. Van Otterloo, *A contour-oriented approach to shape analysis*. Prentice Hall International (UK) Ltd., 1991.
- [2] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [3] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr, "Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional," *International journal of computer vision*, vol. 50, no. 3, pp. 295–313, 2002.
- [4] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham *et al.*, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.
- [5] J. S. Marques and A. J. Abrantes, "Shape alignment: optimal initial point and pose estimation," *Pattern Recognition Letters*, vol. 18, no. 1, pp. 49–53, 1997.
- [6] J. S. Marques, "A fuzzy algorithm for curve and surface alignment," *Pattern recognition letters*, vol. 19, no. 9, pp. 797–803, 1998.
- [7] S. Mahmoodi, B. S. Sharif, E. G. Chester, J. P. Owen, and R. Lee, "Skeletal growth estimation using radiographic image processing and analysis," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 4, no. 4, pp. 292–297, 2000.
- [8] A. Tsai, A. Yezzi Jr, W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *Medical Imaging, IEEE Transactions on*, vol. 22, no. 2, pp. 137–154, 2003.
- [9] D. Cremers, S. J. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *International Journal of Computer Vision*, vol. 69, no. 3, pp. 335–351, 2006.
- [10] N. Paragios, M. Rousson, and V. Ramesh, "Non-rigid registration using distance functions," *Computer Vision and Image Understanding*, vol. 89, no. 2, pp. 142–165, 2003.
- [11] A. Antoniou and W.-S. Lu, *Practical optimization: algorithms and engineering applications*. Springer, 2007.
- [12] I. Markovsky and S. Mahmoodi, "Least-squares contour alignment," *Signal Processing Letters, IEEE*, vol. 16, no. 1, pp. 41–44, 2009.