"SUPER-RESOLUTION CURVE" AND IMAGE REGISTRATION

Minghui Xia and Bede Liu

Department of Electrical Engineering, Princeton University
Princeton, NJ 08544 (Contact: {mxia,liu}@ee.princeton.edu)

ABSTRACT

An affine invariant 2-D curve matching and alignment approach is proposed to solve the image registration problem. Superior accuracy and efficiency is achieved by a concept called *super-resolution curve* and a technique called *B-spline fusion*. A *super-resolution curve* is formed by superimposing two affine related and registered curves (in discrete form) upon each other. The *B-spline fusion* technique is designed to obtain a single B-spline approximation of the *super-resolution curve* and registration estimate simultaneously. Occlusion is handled by curve segmentation using inflections and cusps, which are affine invariant. Partial match then becomes possible by matching segments of curves. A complete image registration framework based on edge detection and curve alignment is then proposed. Accurate registration results have been achieved.

1. INTRODUCTION

Image registration has found numerous applications in medical operations [1], digital video processing([2, 3]), and synthetic aperture radar (SAR) image analysis [4]. The research areas related to this problem include image normalization and invariant pattern recognition. In this paper, we propose an efficient curve matching and alignment method to solve affine image registration and invariant shape description problems. (Note that weak perspective projections can also be approximated by affine transforms.) We introduce a *B-spline fusion* technique to accurately recover the transform parameters between two affine related curves. The accuracy and efficiency is achieved by the concept of *super-resolution curve*. A *super-resolution curve* is a curve formed by two affine related curves (in discrete form) registered and superimposed upon each other. Curve matching and registration can then be accomplished in one step. We address the occlusion problem by curve segmentation using inflections and cusps, which are affine invariant. A complete image registration framework is then formed based on the proposed techniques.

This research is supported by an NJ State R&D excellence grant.

2. 2-D CURVE REPRESENTATION BY B-SPLINES

In this paper, 2-D points are represented in vector form, i.e., point $\mathbf{p}_i = [x_i \, y_i]^T$. A curve in discrete form is a point list, $\mathbf{p} = [\mathbf{p}_0 \, \mathbf{p}_1 \, \ldots \, \mathbf{p}_{M-1}]$. A B-Spline representation of curve $\mathbf{p}$ divides the curve into L segments and approximates each segment by a linear combination of parametric polynomial base functions. Specifically, we

$$minimize \sum_{i=0}^{M-1} \|\mathbf{p}_i - \sum_{j=0}^{L+N-1} \mathbf{d}_j \mathbf{B}_j^N(\frac{t_i - u_j}{u_{j+1} - u_j})\|^2 \quad (1)$$

where $\|\cdot\|$ denotes L_2 norm, and

$$B_j^N(\xi) = \sum_{k=0}^{N} a_{jk}\xi^{N-k}, \; j \in [0, L+N-1], \; \xi \in [0,1] \quad (2)$$

The minimization can be achieved by finding appropriate $\{\mathbf{d}_j\}$ and $\{t_i\}$. N is the degree of the B-spline base function $\{B_j^N(\xi)\}$. M is the total number of discrete points. $\{t_i\}$ is the parameter value for each point. $\{\mathbf{d}_j\}$ are called *control points*. The coefficients $\{a_{jk}\}$ of the B-spline base function can be determined by enforcing continuity of the $N-1$ derivatives at the end points of each segment. The points connecting neighboring segments are called *knots*, the parameters of which are denoted by $\{u_j\}$. In addition to being an efficient representation, B-splines have the advantages of invariance to affine transformation, smoothing noise, and natural decoupling of *x* and *y* coordinates. Solving Eq.(1) is a non-linear problem. However, $\{t_i\}$ can be initialized and iteratively updated by heuristic methods [5]. Linear least squares (LS) solution for $\mathbf{d}$ can then be obtained by:

$$\mathbf{d} = \mathbf{H}^\dagger \mathbf{p} \quad (3)$$

where "†" denotes pseudo inverse matrix computation through singular value decomposition (SVD), and

$$\mathbf{H}_{j,k} = \sum_{i=0}^{M-1} B_j^N(\frac{t_i - u_j}{u_{j+1} - u_j}) B_k^N(\frac{t_i - u_k}{u_{k+1} - u_k}) \quad (4)$$
$$0 \le j,k \le L+N-1$$

3. USING B-SPLINES IN AFFINE CURVE MATCHING AND ALIGNMENT

Given two curves $\mathbf{p}$ and $\tilde{\mathbf{p}}$, we want to decide if they are affine related, and if so, to determine the specific transform. We first consider the case where all curves are open curves and ignore the occlusion problem. Closed curve matching and occlusion problem will be addressed later in this section. It is tempting to match two affine related curves by their control points from their B-spline representations, using the *affine invariance* property. However, B-spline representation of a curve is not unique, as different parameter initializations may lead to quite different control point sets. Therefore, there is no guarantee that the two control point sets will relate to each other by the same affine transformation. Attempts to circumvent this difficulty have not been very successful.

In this paper, we introduce a concept called *super-resolution curve* to solve this problem. Assume we know the transform between the original and the transformed curves, they can then be registered and superimposed upon each other in one frame. This "combined" curve is called a *super-resolution curve*. We can then use a single B-spline to approximate this *super-resolution curve*. The desired B-spline should match the *super-resolution curve* in the original frame, while its affine transform should match the transformed *super-resolution curve* in the transformed frame. The advantage of using a single B-spline to represent two curves is accuracy and efficiency. In comparison to the separate B-spline representation of two curves, the *super-resolution curve* provides twice the amount of data for improved accuracy, while a single B-spline representation provides efficiency.

Since the transform is not known in advance, we can estimate the B-spline and the transform parameters simultaneously using the concept of *super-resolution*. We call this approach *B-spline fusion*. It can be formulated as follows. Using homogeneous coordinates, an affine transform can be represented as:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_x^T \\ \mathbf{a}_y^T \\ 0 \;\; 0 \;\; 1 \end{bmatrix} \qquad (5)$$

where $\mathbf{a}_x = [a_{11} \;\; a_{12} \;\; a_{13}]^T$ and $\mathbf{a}_y = [a_{21} \;\; a_{21} \;\; a_{23}]^T$. Its inverse is

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{a}_x^{\star T} \\ \mathbf{a}_y^{\star T} \\ 0 \;\; 0 \;\; 1 \end{bmatrix} \qquad (6)$$

We seek the transform $\mathbf{A}$ and the control points $\{\mathbf{d}\}$ to

$$minimize \;\; \|\mathbf{H}\mathbf{d}_x - \begin{bmatrix} \mathbf{p}_x \\ \tilde{\mathbf{p}}_x^\star \end{bmatrix}\| + \|\mathbf{H}\mathbf{d}_y - \begin{bmatrix} \mathbf{p}_y \\ \tilde{\mathbf{p}}_y^\star \end{bmatrix}\|$$
$$+ \|\tilde{\mathbf{H}}\tilde{\mathbf{d}}_x - \begin{bmatrix} \mathbf{p}_x^\star \\ \tilde{\mathbf{p}}_x \end{bmatrix}\| + \|\tilde{\mathbf{H}}\tilde{\mathbf{d}}_y - \begin{bmatrix} \mathbf{p}_y^\star \\ \tilde{\mathbf{p}}_y \end{bmatrix}\| \qquad (7)$$

In this equation, $\mathbf{H}$ is given by Eq.(4), $\mathbf{d}_x$ and $\mathbf{d}_y$ are the x and y coordinates of spline control vertices for the original curve in column vector form, $\mathbf{p}_x$ and $\mathbf{p}_y$ denote the coordinates of the original point list in column vector form, tilde, $\tilde{\cdot}$, denotes its affine transformed counterpart. $\mathbf{p}_x^\star$ and $\mathbf{p}_y^\star$ denote the original point set mapped to the transformed frame by $\mathbf{A}$. $\tilde{\mathbf{p}}_x^\star$ and $\tilde{\mathbf{p}}_y^\star$ denote the transformed point set projected back to the original frame by $\mathbf{A}^{-1}$. Since we use the same B-spline in both the original frame and the transformed frame, the control points $\tilde{\mathbf{d}}_x$ and $\tilde{\mathbf{d}}_y$ are related to $\mathbf{d}_x$ and $\mathbf{d}_y$ by the same affine transformation $\mathbf{A}$. Specifically,

$$\begin{aligned} \mathbf{p}_x^\star &= [\mathbf{p}_x \; \mathbf{p}_y \; \mathbf{1}] \; \mathbf{a}_x \\ \mathbf{p}_y^\star &= [\mathbf{p}_x \; \mathbf{p}_y \; \mathbf{1}] \; \mathbf{a}_y \\ \tilde{\mathbf{p}}_x^\star &= [\tilde{\mathbf{p}}_x \; \tilde{\mathbf{p}}_y \; \mathbf{1}] \; \mathbf{a}_x^\star \\ \tilde{\mathbf{p}}_y^\star &= [\tilde{\mathbf{p}}_x \; \tilde{\mathbf{p}}_y \; \mathbf{1}] \; \mathbf{a}_y^\star \\ \tilde{\mathbf{d}}_x &= [\mathbf{d}_x \; \mathbf{d}_y \; \mathbf{1}] \; \mathbf{a}_x \\ \tilde{\mathbf{d}}_y &= [\mathbf{d}_x \; \mathbf{d}_y \; \mathbf{1}] \; \mathbf{a}_y \end{aligned} \qquad (8)$$

We note that the sum of the first two terms in Eq.(7) is the B-spline fitting error in the original frame, and the sum of the last two terms is the B-spline fitting error in the transformed frame. According to the *affine invariance* property of B-spline fitting, a B-spline's LS fitting to the *super-resolution curve* in the original frame will guarantee its LS fitting to the transformed *super-resolution curve*. In other words, minimization of the first two terms in Eq.(7) is equivalent to minimization of the last two terms. Therefore, we only keep the last two terms for minimization.

At this stage, we need to estimate $\tilde{\mathbf{t}}$ contained in $\tilde{\mathbf{H}}$, and $\{\tilde{\mathbf{d}}_x, \tilde{\mathbf{d}}_y, \mathbf{a}_x, \mathbf{a}_y\}$. The non-linear parameters $\tilde{\mathbf{t}}$ can be initialized and updated similar to the approach in Section 2. The linear solutions for $\{\tilde{\mathbf{d}}_x, \tilde{\mathbf{d}}_y, \mathbf{a}_x, \mathbf{a}_y\}$ can then be obtained by:

$$minimize \;\; f_x(\tilde{\mathbf{d}}_x, \mathbf{a}_x) = \|\tilde{\mathbf{H}}\tilde{\mathbf{d}}_x - \begin{bmatrix} \mathbf{P}\,\mathbf{a}_x \\ \tilde{\mathbf{p}}_x \end{bmatrix}\|$$
$$and$$
$$minimize \;\; f_y(\tilde{\mathbf{d}}_y, \mathbf{a}_y) = \|\tilde{\mathbf{H}}\tilde{\mathbf{d}}_y - \begin{bmatrix} \mathbf{P}\,\mathbf{a}_y \\ \tilde{\mathbf{p}}_y \end{bmatrix}\|$$

where $\mathbf{P} = [\mathbf{p}_x \;\; \mathbf{p}_y \;\; \mathbf{1}]$, and separation of the two terms is due to decoupling of *x* and *y* coordinates. This leads to:

$$\left\{ \begin{array}{l} \mathbf{a}_x = \mathbf{CD}\tilde{\mathbf{p}}_x \\ \tilde{\mathbf{d}}_x = \mathbf{D}\tilde{\mathbf{p}}_x \end{array} \right. \quad \left\{ \begin{array}{l} \mathbf{a}_y = \mathbf{CD}\tilde{\mathbf{p}}_y \\ \tilde{\mathbf{d}}_y = \mathbf{D}\tilde{\mathbf{p}}_y \end{array} \right. \qquad (9)$$

$$where \quad \begin{cases} \mathbf{C} = (\mathbf{P}^T\mathbf{P})^{\dagger}\mathbf{P}^T\tilde{\mathbf{H}}_1 \\ \mathbf{D} = (\tilde{\mathbf{H}}^T\tilde{\mathbf{H}} - \tilde{\mathbf{H}}_1^T\mathbf{P}\mathbf{C})^{\dagger}\tilde{\mathbf{H}}_2^T \\ \tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{H}}_1 \\ \tilde{\mathbf{H}}_2 \end{bmatrix} \end{cases}$$

$\tilde{\mathbf{H}}_1$ is formed by the parameter values of $(\mathbf{p}_x^\star, \mathbf{p}_y^\star)$, and $\tilde{\mathbf{H}}_2$ is formed by the parameter values of $(\tilde{\mathbf{p}}_x, \tilde{\mathbf{p}}_y)$. It should be pointed out that the linear LS solution for $\{\tilde{\mathbf{d}}_x, \tilde{\mathbf{d}}_y, \mathbf{a}_x, \mathbf{a}_y\}$ does not involve iteration. Iteration is involved only when we need to update parameters $\tilde{\mathbf{t}}$. No iteration is necessary if the initial parameter values are accurate enough. The solution usually converges to a good estimate within 5-10 iterations and to an accurate one within 20 iterations. Each iteration takes about 1 second.

The approach as described above solves the curve matching and alignment problem in one step. To find one or more affine matches in a database, and then recover the affine transform, all we need to do is to solve Eq.(7) for each curve pair, and 5 to 10 iterations would reveal whether or not the two curves match by thresholding the B-spline fitting error. If the match is good, the iteration will continue until accurate alignment is obtained. Otherwise, we move on to the next pair.

Our approach also applies to partial curve matching. If there is occlusion, each curve is segmented by using inflections and cusps which are invariant to affine transform. Partial match then becomes possible by matching segments of curves. Rough registration can therefore be obtained for parameter initialization in the next B-spline fusion step. Closed curve matching will be handled similarly as the occlusion problem. We omit details here due to limit of space.

4. IMAGE REGISTRATION BY CURVE MATCHING AND ALIGNMENT

Our approach to find the registration of a digitized original image with a transformed image is summarized in the diagram of Fig. 1. Canny edge detector is used in our test. Of the multiple curves found at this stage, curves that are long enough (i.e., 50 pixels long) to be useful for matching and alignment are selected. We first search for complete curve matches with no occlusion. This can be achieved by mapping each curve to a canonical frame, using two end points and the geometric center as references (or correspondences). A modified Hausdorff distance is employed to measure similarity among curves. At this stage, subsampling is used to speed up computation. The curve pair with the smallest distance will be selected for the next stage. If a good match is found, B-spline fusion will be performed on the pair. Otherwise, we assume there is occlusion. Each curve will be approximated by a B-spline. It is then segmented by inflections and/or cusps. The segmented curves in the original image will be matched with those in the transformed image. The closest partial match will be found at this stage and rough registration will be available to initialize parameter values $\mathbf{t}$ for the next step. The final B-spline fusion step will then produce accurate alignment between the two curves and therefore the registration between the two images.

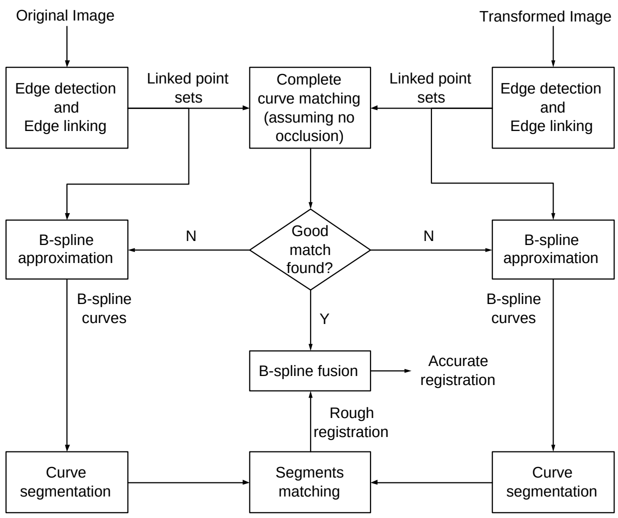

Fig. 1. Overall diagram for Image Registration

5. RESULTS

Our image registration technique has been tested on binary as well as gray level images, and under noisy conditions. Cases involving partial matches or closed curve matches are also tested. The affine transform under the test is general, which includes rotation, non-uniform scaling, translation and shearing. Due to limit of space, we only show results using gray level images, with and without noise.

Fig. 2 shows our registration technique applied to a 256x256 gray level image, with i.i.d. Gaussian noise $N(0, \sigma^2)$, $\sigma^2 = 256$ added to the transformed image. The noise is rather strong considering the image's gray level range [0,255]. The first row shows the original and the transformed images. The transform consists of shearing in X direction by a factor of $\beta = 1.0$, 5 degrees rotation, scaling of 0.9 and 1.2 in X and Y direction, 10 and 20 pixels translation in X and Y direction respectively. The resulting transform matrix is

$$\mathbf{A} = \begin{bmatrix} 0.8966 & 0.8181 & 10.0000 \\ 0.1046 & 1.3000 & 20.0000 \\ 0 & 0 & 1.0000 \end{bmatrix} \quad (10)$$

The second row in Fig. 2 shows the detected edges. The third row shows the best curve match found. In this case, it is a partial match between two curves. The fourth row shows rough registration by segment matching on the left, and accurate registration recovered by B-spline fusion (with two registered curves superimposed upon each other) on the right. Convergence is achieved within 19 iterations.

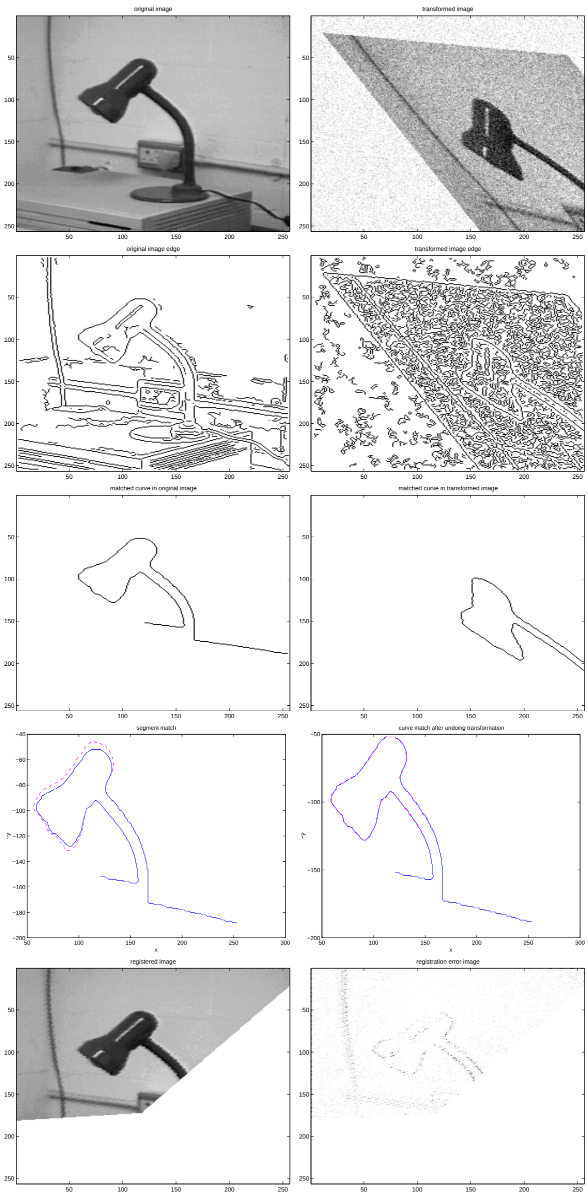

Fig. 2. Noisy gray level image registration results

We compare this result with that obtained without noise. The estimated transform without noise is:

$$\hat{\mathbf{A}} = \begin{bmatrix} 0.8959 & 0.8193 & 9.9471 \\ 0.1041 & 1.3002 & 20.0889 \\ 0 & 0 & 1.0000 \end{bmatrix} \tag{11}$$

The estimated transform under noisy condition is:

$$\hat{\mathbf{A}} = \begin{bmatrix} 0.8939 & 0.8172 & 10.3228 \\ 0.1028 & 1.3009 & 20.1979 \\ 0 & 0 & 1.0000 \end{bmatrix} \tag{12}$$

We note that most estimated coefficients has achieved 1% accuracy. $\hat{\mathbf{A}}^{-1}$ is then applied to the transformed image and the result is compared with the original image for registration error. The last row in Fig. 2 shows the back projected transformed image along with the difference between the original and the back projected transformed image (with the out-of-frame difference ignored). Darker color denotes larger difference. The Gaussian noise in these images has been removed for comparison purpose. The registration errors are compared in Table 1. Under noisy conditions, all the errors increased. However, the positional registration mean squared error (MSE) is still kept within half a pixel. This test demonstrates the robustness of our approach under noisy conditions.

	w/o noise	with noise
Positional Reg. MSE	0.1216 pixel	0.3044 pixel
Positional Reg. Error Max	0.2895 pixel	0.5997 pixel
B-spline Approx. MSE	0.1980 pixel	0.3221 pixel
Pixel Value Reg. MSE	2.9154	3.9913

Table 1. Registration error statistics

6. REFERENCES

[1] J. B. A. Maintz and M. A. Viergever, “A survey of medical image registration,” *Medical Image Analysis*, vol. 2, no. 1, pp. 1–37, 1998.

[2] R. J. Radke, V. Zagorodnov, S. R. Kulkarni, and P. J. Ramadge, “Estimating correspondence in digital video,” *IEEE International Conference on Information Technology, Coding, and Compression*, pp. 196–201, Apr. 2001.

[3] L. Van Gool V. Ferrari, T. Tuytelaars, “Real-time affine region tracking and coplanar grouping,” *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 226–233, Dec. 2001.

[4] H. Li, B. S. Manjunath, and S. K. Mitra, “A contour-based approach to multisensor image registration,” *IEEE Trans. on Image Processing*, vol. 4, no. 3, pp. 320–334, 1995.

[5] G. E. Farin, *Curves and surfaces for computer aided geometric design: a practical guide*, Academic Press, 4th edition, 1997.