AN EFFICIENT IMPLEMENTATION OF AFFINE TRANSFORMATION USING ONE-DIMENSIONAL FFT'S

Erwin Pang and Dimitrios Hatzinakos

University of Toronto, Department of Electrical and Computer Engineering
10 King's College Road, Toronto, Ontario, M5S-3G4, Canada
Tel: (416) 978-1613, Fax: (416) 978-4425, email: dimitris@comm.toronto.edu

ABSTRACT

In this paper, we propose a new decomposition scheme and an efficient interpolation algorithm for affine transformation of a digital image. We try to reconstruct the affine-transformed image by resampling it with the highest possible quality, lowest complexity and throughput rate. Based on the proposed decomposition, the transform is completed by a sequence of 3-pass translations and a scaling operation where each of them is one-dimensional in nature. This method preserves quality and guarantees simplicity. We place the emphasis on the feasibility of a parallel implementation that can benefit from pipeline technologies. Further, an efficient FFT-based implementation of this new algorithm is suggested. Experimental evidence of the effectiveness and robustness of the proposed method is reported. The problem is relevant to video transmission, image registration, and computer graphics manipulation.

1. INTRODUCTION

Geometrical transformation of digital image is a common application in image processing [1] [2]. The uniformly distributed samples on a two-dimensional plane, after being transformed, will be misaligned with the reference grid pattern which are only defined for discrete locations. An interpolation of these transformed pixels is thus needed to recover those on the grid points. This process can be treated as resampling of the transformed image. High quality resampling algorithm can contribute to image compression, registration of multispectral satellite information, and improve the fidelity of video transmission. An efficient algorithm can also give benefits to medical imagery [3], radar-directed navigation, and many other real-time applications [4]. It becomes necessary to transform digital image in a fast manner and preserve high quality simultaneously.

Generally, two issues should be considered, namely the transformation scheme and interpolation algorithm. The transformation scheme deals with strategies than can transform an image efficiently. Performance parameters like throughput rate, complexity, amount of distortion, and requirement of memory space are often sensitive to the particular transformation scheme used. Recently, with the evolution of parallel processors, the compatibility of the scheme with a pipeline structure has become a valuable asset. The interpolation algorithm involves recovering pixel values at inter-pixel positions. This algorithm affects image quality, transform rate, and complexity of the entire process.

This work was supported in part by Natural Sciences and Engineering Research Council of Canada and in part by Information Technology Research Center.

In this paper, we consider a general affine transformation of an image. Unlike the rigid-body transform which constrains image dynamics to translation, rotation, and resizing, an affine transform defines projection of complex 3-D motions on a 2-D plane. For example, images taken by satellite sensors can be modeled by affine transform since these sensors often incline with the earth surface during image acquisition. We propose a new approach to decompose the intrinsically 2-D operation into three passes of 1-D shifting plus one scaling operation, and we concentrate on a FFT-based method to resample the translated signal. While conventional two-pass methods cause severe distortion, the proposed three-pass one reduces the errors significantly. Moreover, the complementary FFT-based interpolator, which corresponds to an ideal low-pass filter, has been shown to be more accurate and simpler than other spatial interpolation methods [5]. The emergence of the parallel computation techniques enables the use of pipeline-FFT microcircuits to perform real-time resampling. We show that the quality and efficiency of the three-pass FFT-based method is superior than other two-pass spatial algorithms.

2. AFFINE MODEL FOR THREE-DIMENSIONAL MOTIONS

A general affine transformation relates two image frames $f_1(x_1, y_1)$ and $f_2(x_2, y_2)$ with the following mapping equations [6]

$$x_2 = Ax_1 + By_1 + E \quad (1)$$

$$y_2 = Cx_1 + Dy_1 + F \quad (2)$$

which can be compactly expressed in matrix form as

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} A & B & E \\ C & D & F \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \quad (3)$$

Affine transform preserves parallel lines and equispaced points. Due to its linearity and simplicity, affine model is widely used in block matching and motion estimation problems for compression purposes [6]. Another credit of affine transform is its generality to represent 2-D projections of complex 3-D motions on an image plane. This implies that the affine transform, as uniquely defined by the six parameters in (3), is adequate to characterize projections of six degrees of freedom in a three-dimensional space. This includes three-dimensional translations, rotations, and scaling as special cases.

Modelling of affine transform to project 3-D rotations of planar objects on an image plane is illustrated as an example. Orthographic projection is assumed which gives

Figure 1. Modelling of 3-D rotations by affine transform. (Left) Original image plane lying on xy-plane. (Right) Projection of the planar object on xy-plane after rotating by $(\theta,\ \alpha,\ \beta) = (30^o,\ 30^o,\ 30^o)$ using proposed method.

parallel projection and avoids perspective distortion. We define R_β, R_α, and R_θ as the three anticlockwise rotations about the Cartesian coordinate axes in the order of z-axis, x-axis, and then y-axis, i.e.

$$I_2 = (R_\beta\, R_\alpha\, R_\theta)\ I_1 = R\, I_1 \tag{4}$$

where θ, α, and β are rotated angles about z, x, and y-axis respectively, and I_1, I_2 are the original $([x_1\, y_1\, z_1]^T)$ and transformed coordinates $([x_2\, y_2\, z_2]^T)$, respectively. R is a 3x3 matrix representing the combined rotations, and can be shown to be

$$R = \begin{bmatrix} c_\beta c_\theta + s_\beta s_\alpha s_\theta & s_\beta s_\alpha c_\theta - c_\beta s_\theta & s_\beta c_\alpha \\ c_\alpha s_\theta & c_\alpha c_\theta & -s_\alpha \\ c_\beta s_\alpha s_\theta - s_\beta c_\theta & s_\beta s_\theta + c_\beta s_\alpha c_\theta & c_\beta c_\alpha \end{bmatrix} \tag{5}$$

where $\{c_\angle \equiv \cos\angle,\ s_\angle \equiv \sin\angle \mid \angle : \theta, \alpha, \beta\}$. Since the transformed planar object is projected on an image plane, only the x and y coordinates are visualized, giving

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} c_\beta c_\theta + s_\beta s_\alpha s_\theta & s_\beta s_\alpha c_\theta - c_\beta s_\theta \\ c_\alpha s_\theta & c_\alpha c_\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + z_1 \begin{bmatrix} s_\beta c_\alpha \\ -s_\alpha \end{bmatrix} \tag{6}$$

Provided z_1 is known, (6) is an affine transformation. If the planar object is initially lying parallel to the xy-plane, the translation terms become constants. Figure 1 shows a 2-D projection of a planar image which undergoes rotations with $(\theta,\ \alpha,\ \beta) = (30^o,\ 30^o,\ 30^o)$ and the proposed methods described in the following sections are used. The original image is assumed to be lying on the xy-plane such that $z_1 = 0$. Generally, more complex motion and image distortion can be correctly modeled by an affine transformation [5].

3. TRANSFORMATION SCHEME

3.1. One-Pass and Two-Pass Method

The one-pass method transforms an image directly using two-dimensional interpolator like bilinear, bicubic, and biharmonic method. It requires two separate buffer frames to support the operations. This may not be favourable for transforming large images. However, its distortion is found to be less than that of a two-pass method [2] [7]. This concludes that memory usage and complexity is a limiting factor for a one-pass process.

The two-pass method breaks the 2-D process into two consecutive 1-D processes. The decomposition takes the form of

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ r & l \end{bmatrix} \begin{bmatrix} p & q \\ 0 & 1 \end{bmatrix} \tag{7}$$

where $p = A$, $q = B$, $r = \frac{C}{A}$, $l = D - \frac{BC}{A}$. For the special case of rotation, the decomposition becomes

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \tan\theta & \sec\theta \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ 0 & 1 \end{bmatrix} \tag{8}$$

which is referred to twice skew transformation in [1]. The first operation is purely horizontal along the x-axis while the subsequent operation is purely vertical along the y-axis. Each of the operation involves a translation and a scaling along one axis so that only one line buffer is required to complete the transformation. Moreover, interpolation is simpler since it can be carried out line-by-line, and parallel implementation is possible. However, due to the intermediate scaling involved, severe distortion will result [8].

3.2. Proposed Three-pass Decomposition

A three-pass method has been proposed in [9] to perform rotation only, where the decomposition is

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \equiv \begin{bmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin\theta & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{bmatrix} \tag{9}$$

It can be seen that all three steps are one-dimensional and they involve no scaling at all. This method reduces distortion significantly [8]. Moreover, the distortion introduced is "added" or non-destructive which means complete elimination of it is possible by some post-processing.

Taking the advantages of the characteristics of such three-pass algorithm, we generalize the decomposition scheme and apply it on affine transform. We propose to break the 3x3 transform matrix in (3) into separate processes as

$$\begin{bmatrix} A & B & E \\ C & D & F \\ 0 & 0 & 1 \end{bmatrix} \equiv \begin{bmatrix} a & b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ c & 1 & d \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & e & f \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

which corresponds to a sequence of three-pass, one dimensional operations along either axis in each pass. These operations are translations (t_0) as shown in Table 1, and the parameters on the right side of (10) are defined in Table 2. Since scaling is performed at the final stage, the loss in the intermediate image is minimized, in contrast to the two-pass method which requires two translations and two scaling operations. In other words, we intend to trade a scaling for a translation in an effort to reduce distortion and improve the resampling quality. For the case of xy-plane rotation about the origin, $[A\ B; C\ D]$ in (10) (displacement $E = F = 0$) reduces to (9).

4. INTERPOLATION AND RESAMPLING ALGORITHM

4.1. Spatial Domain Interpolator

There are several interpolating functions for image reconstruction [10]. The simplest one is the nearest neighbour

First Pass	$x_2'' = x_1 + \overbrace{ey_1 + f}^{t_0}$	$y_2'' = y_1$
Second Pass	$x_2' = x_2''$	$y_2' = y_2'' + \overbrace{cx_2'' + d}^{t_0}$
Third Pass	$x_2 = \overbrace{ax_2'}^{scaling} + \overbrace{by_2'}^{t_0}$	$y_2 = y_2'$

Table 1. Three-pass sequence for affine transform

a	b	c
$AD - BC$	$B - \frac{A}{C}(D-1)$	C
d	e	f
$F - C$f	$\frac{D-1}{C}$	$\frac{1}{a}(E - F$b$)$

Table 2. Parameters formulating the decomposition

function where the value at any resampled point is taken as the nearest pixel value. Another algorithm is linear interpolation where the interpolating function is simply constructed by joining samples with straight lines. Cubic interpolation [11] is more accurate since it uses more neighbouring points for resampling. Four points are used in a one-dimensional case. Spline functions [12] are more complex. They are positive everywhere and tend to smooth the resampled image. Despite the high accuracy of spline function, the complexity is too large to be used in real-time applications. A frequency reshaping method for signal interpolation is introduced in the next section.

4.2. FFT-Based Interpolation

In correspondence with the approach of the proposed transformation scheme, we should put the emphasis on resampling a translated signal. Starting with the shifting property of Fourier transform for signal translations, we have

$$f(t - t_0) \Longleftrightarrow e^{-j\omega t_0} F(w) \tag{11}$$

where $F(\omega)$ is the Fourier transform of $f(t)$ and t_0 is the translation parameter which can take any real value. To interpret (11) in an alternative way, we write

$$f(t - t_0) \Longleftrightarrow e^{-j\omega t_0}|F(\omega)|e^{j\theta(\omega)} = |F(\omega)|e^{j\left(\theta(\omega) - \theta'(\omega)\right)} \tag{12}$$

where $\theta(\omega)$ is the phase function of $F(\omega)$ and $\theta'(\omega) = -\omega t_0$ is a phase reshaping factor. In the discrete case, the DFT of the shifted function can be obtained by modifying the phase of the DFT coefficients of the unshifted function. For a band-limited signal which is sampled above the Nyquist rate, the following relationship holds

$$\theta'(k\frac{2\pi}{N}) = \begin{cases} -k\frac{2\pi}{N}t_0 & 0 \le k < \frac{N}{2} \\ 0 & k = \frac{N}{2} \\ (N-k)\frac{2\pi}{N}t_0 & \frac{N}{2} < k \le N-1 \end{cases} \tag{13}$$

where we assume N, the signal length, is even. It is shown in [9] that the above method gives the best quality to image rotation. As a result, the entire shifting can be achieved by using FFT as follows

$$f_{t_0}(n) = \mathrm{IFFT}\Big\{\mathrm{FFT}\big[\mathrm{f(n)}\big] \cdot \mathrm{e}^{\mathrm{j}\theta'(\mathrm{k}\frac{2\pi}{\mathrm{N}})}\Big\} \tag{14}$$

where $f_{t_0}(n)$ is the resampled version of $f(n)$ after translating t_0 units. Spatial scaling is done by a subset or padding operation of the FFT coefficients. Assuming we have to resample M samples from N samples. For signal expansion, $M > N$ and the equivalent operations are

$$F(k) = \begin{cases} F(k) & 0 \le k \le \frac{N}{2} - 1 \\ \frac{1}{2}F(k) & k = \frac{N}{2} \\ 0 & \frac{N}{2} + 1 \le k \le \frac{M}{2} \end{cases} \tag{15}$$

$$F(M-k) = F^*(k) \qquad \text{for} \quad 1 \le \mathrm{k} \le \frac{\mathrm{M}}{2} - 1 \tag{16}$$

Signal contraction is simply done by preserving a subset of coefficients for reconstruction. This is lossless provided the signal is oversampled.

5. SIMULATION RESULTS

Simulations are performed to demonstrate the effectiveness of our proposed algorithms. We will use the affine transform described in Section 2. The corresponding transformation matrix is $[\frac{7}{8}\ \frac{-\sqrt{3}}{8}\ 0; \frac{\sqrt{3}}{4}\ \frac{3}{4}\ 0;\ 0\ 0\ 1]$. According to Table 2, the three-pass parameters in (10) are found to be $a = \frac{3}{4}$, $b = \frac{1}{2\sqrt{3}}$, $c = \frac{\sqrt{3}}{4}$, $d = 0$, $e = \frac{-1}{\sqrt{3}}$, $f = 0$. Cubic interpolation is used to compare with the FFT-based method. The two-pass decomposition is applied for the cubic method as given in (7) with $p = \frac{7}{8}$, $q = \frac{-\sqrt{3}}{8}$, $r = \frac{2\sqrt{3}}{7}$, $l = \frac{6}{7}$. First we transform a synthetic image with simple sinusoidal circular pattern of given frequency or wavelength. The test image with dimension 256x256 with origin at image centre is defined to have the distribution

$$T(x,y) = 0.5\Big[1 + \cos(\frac{2\pi}{\lambda}\sqrt{x^2 + y^2})\Big] \tag{17}$$

where $0 \le T(x,y) \le 1$ and λ is spatial wavelength measured in units of sample spacing. It can be seen that $\lambda_n = 2$ corresponds to Nyquist limit. To eliminate edge effects, only central regions are used in error analysis.

Figure 2 shows the root-mean-square (rms) error of the transform for cubic and FFT method at different wavelength. Note that 0 dB corresponds to rms error of 1, the case of complete failure. The three-pass and FFT-based method has significantly smaller error than the two-pass and cubic method. Note that the curve for the FFT method drops rapidly between $\lambda = 2$ to $\lambda = 3$. This is because the three-pass transformation involves a scaling down of $a = \frac{3}{4}$, so that the image is expected to give good results only if it is bandlimited and have $\lambda > \frac{4}{3}\lambda_n \approx 2.6667$. The curve for the cubic method, however, only decays slowly as the wavelength increases. To illustrate the difference between the two methods, we transform the synthetic image forward and backward twice, and then forward again to the transformed pattern for $\lambda = 4$, and show the results in Figure 3. The cubic method gives a blurred pattern while the FFT method gives pattern nearly indistinguishable from the expected results. Real images are also transformed for comparison. Since there is no transformed image to compare, we transform the image forward and backward five times and find the errors with respect to the original image. Figure 4 shows the resulting images for both methods. Note that the FFT method preserves the details better. Table 3 summaries the performance of the synthetic and real cases. Real images do not give very outstanding results as synthetic images do since they are not bandlimited.

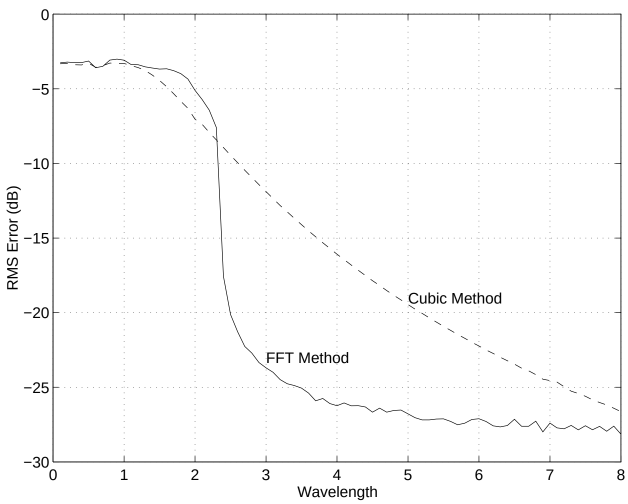

Figure 2. RMS error of affine transformation of synthetic image. (Dotted line): Cubic method using two-pass decomposition. (Solid line): Proposed FFT method using three-pass decomposition

Method	Synthetic pattern ($\lambda = 4$)	Lenna
Cubic 2-Pass	-7.65 dB	-18.27 dB
FFT 3-Pass	-24.92 dB	-20.18 dB

Table 3. Tabulated results of the simulations

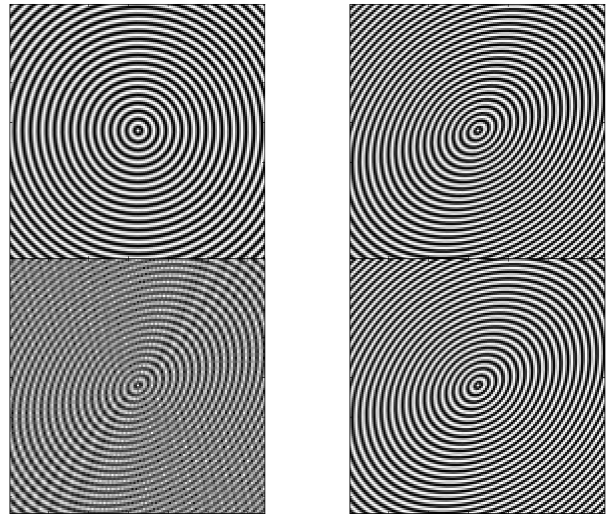

Figure 3. (top left): Original synthetic pattern. (top right): Expected transformed pattern. (bottom left): Transformed pattern using cubic 2-pass method. (bottom right): Transformed pattern using proposed FFT 3-pass method.

Figure 4. (top): Original image. (middle): Image transformed forward and backward using cubic 2-pass method. (bottom): Image transformed forward and backward using FFT 3-pass method.

REFERENCES

[1] N. T. *et al.*, "Hardware for Image Rotation by Twice Skew Tranformations," *IEEE Trans. on ASSP*, vol. 35, pp. 527–531, Apr 1987.

[2] D. Fraser, "Comparison at High Spatial Frequencies of Two-Pass and One-Pass Geometric Transformation Algorithms," *Computer Vision, Graphics, and Image Processing*, vol. 46, pp. 267–283, 1989.

[3] D. L. G. Hill and D. J. Hawkes, "Medical Image Registration using Knowledge of Adjacency of Anatomical Structures," *Image and Vision Computing*, vol. 12, pp. 173–178, Apr 1994.

[4] D. Dooling, "Navigating Close to Shore," *IEEE Spectrum*, pp. 24–31, Dec 1994.

[5] C. Bussiere and D. Hatzinakos, "Intensity Scale Invariant Estimation with Rotation and Spatial Scaling Information," *ICASSP*, vol. 5, pp. 233–236, 1994.

[6] V. Seferidis and M. Ghanbari, "General Approach to Block-Matching Motion Estimation," *Optical Engineering*, vol. 32, pp. 1464–1474, Jul 1993.

[7] T. Thong, "Frequency Domain Analysis of Two-Pass Rotation Algorithm," *ICASSP*, vol. 3, pp. 1333–1336, 1985.

[8] D. Fraser and R. A. Schowengerdt, "Avoidance of Additional Aliasing in Multipass Image Rotations," *IEEE Trans. on Imag. Proc.*, vol. 3, pp. 721–735, Nov 1994.

[9] M. U. *et al.*, "Convolution-Based Interpolation for Fast, High-Quality Rotation of Images," *IEEE Trans. on Imag. Proc.*, vol. 4, pp. 1371–1381, Oct 1995.

[10] E. Maeland, "On the Comparison of Interpolation Methods," *IEEE Trans. on Med. Imag.*, vol. 7, pp. 213–217, Sept 1988.

[11] R. G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Trans. on ASSP*, vol. 29, pp. 1153–1160, Dec 1981.

[12] H. S. Hou and H. C. Andrews, "Cubic Splines for Image Interpolation and Digital Filtering," *IEEE Trans. on ASSP*, vol. 26, pp. 508–517, Dec 1978.