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.

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 re-sizing, 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]

$$x_2 = Ax_1 + By_1 + E$$
$$y_2 = Cx_1 + Dy_1 + F$$

which can be compactly expressed in matrix form as

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

(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
The two-pass method breaks the 2-D process into two consecutive 1-D processes. The decomposition takes the form
\[
\begin{bmatrix}
A & B \\
C & D
\end{bmatrix} = \begin{bmatrix}
p & q \\
r & l
\end{bmatrix} \begin{bmatrix}
p & q \\
r & l
\end{bmatrix}
\]
(7)
where \(p = A, q = B, r = C, l = D - \frac{BC}{A}\). For the special case of rotation, the decomposition becomes
\[
\begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix} = \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix} \begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix}
\]
(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
\[
\begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix} \equiv \begin{bmatrix}
1 & -\tan \frac{\theta}{2} \\
0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\]
(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
\[
\begin{bmatrix}
A & B & E \\
C & D & F \\
0 & 0 & 1
\end{bmatrix} \equiv \begin{bmatrix}
a & b & 0 \\
0 & 1 & c \\
0 & 0 & 1
\end{bmatrix}
\]
(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, \([AB; CD]\) 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...
where $f_{\text{inc}}(n)$ is the resampled version of $f(n)$ after translating to 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

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

(15)

$$F(M - k) = F^*(k) \quad \text{for} \quad 1 \leq k \leq M \frac{N}{2} - 1$$

(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 $\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$. According to Table 2, the three-pass parameters in (10) are found to be $a = \frac{A}{2}$, $b = \frac{B}{2}$, $c = \frac{C}{2}$, $d = 0$, $e = \frac{F}{2}$, $f = 0$. 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 $p = \frac{r}{2}$, $q = \frac{\sqrt{3}}{2}$, $r = 2\sqrt{3}$, $l = \frac{3}{2}$. 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

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

(17)

where $0 \leq T(x, y) \leq 1$ and $\lambda$ 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 $\lambda = 2$ and $\lambda = 3$. This is because the three-pass transformation involves a scaling down of $a = \frac{3}{2}$, so that the image is expected to give good results only if it is bandlimited and have $\lambda > \frac{3}{2} \lambda_n \approx 2.687$. 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 blurring 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Synthetic pattern ($\lambda = 4$)</th>
<th>Lena</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubic 2-Pass</td>
<td>-7.65 dB</td>
<td>-18.27 dB</td>
</tr>
<tr>
<td>FFT 3-Pass</td>
<td>-24.92 dB</td>
<td>-20.18 dB</td>
</tr>
</tbody>
</table>

Table 3. Tabulated results of the simulations

REFERENCES


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.