Image Rotation Using an Augmented 2-Step Shear Transform

Hee-Choul Kwon, Heeyong Kwon

Abstract**—**Image rotation is one of main pre-processing steps for image processing or image pattern recognition. It is implemented with a rotation matrix multiplication. It requires a lot of floating point arithmetic operations and trigonometric calculations, so it takes a long time to execute. Therefore, there has been a need for a high speed image rotation algorithm without two major time-consuming operations. However, the rotated image has a drawback, i.e. distortions. We solved the problem using an augmented two-step shear transform. We compare the presented algorithm with the conventional rotation with images of various sizes. Experimental results show that the presented algorithm is superior to the conventional rotation one.

Keywords**—**High speed rotation operation, image rotation, transform matrix, image processing, pattern recognition.

I. INTRODUCTION

RECENTLY, image pattern recognition technology has been applied in a number of ways in various areas. The pre-processing step is essential in the technology [1]. The rotation transform is one of the major pre-processing methods. It is implemented with a rotation matrix multiplication, easily. However, it requires lots of floating point arithmetic operations and trigonometric function calculations, so it takes long execution time. Thus, a new high speed image rotation algorithm is required. In this paper, we present a competitive rotation algorithm which has much less operations and no floating point operations and trigonometric function calculations.

The rest of the paper consists of four parts: In Section II, we give a mathematical description of the problem. Then we present a detailed algorithm for the presented method. Finally, experimental results are shown and concluded.

II. ROTATION TRANSFORM

Conventional rotation transform could be described with a matrix multiplication form which has two sine and two cosine functions in (1) [2]. Let a pixel P in a given image location (x, y). New rotated pixel position of P, P', could be calculated as (1). Then the pixel P' is obtained by rotating the pixel P in counter-clockwise angle θ (Fig. 1).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

H. C. Kwon is with the Computer Science and Engineering Department, Anyang University, Anyang, Korea. He is also with NEXTSOFT, Seoul, Korea (e-mail: heeckwon@empal.com).

H. Kwon is with the Computer Science and Engineering Department, Anyang University, Anyang, Korea (e-mail: hykwon@anyang.ac.kr).

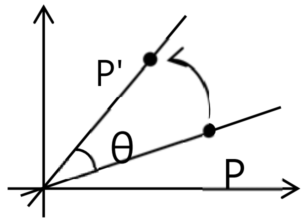


Fig. 1 Rotation of a pixel

A rotation transform is easily implemented with a matrix multiplication. However, it requires lots of floating point arithmetic operations and trigonometric function calculations. After the operations, interpolation process should be applied as well. It takes too much execution time. So, a new high speed image rotation algorithm is required. In real application environment, it could be implemented with pre-calculated sine and cosine values using look-up table. However, it requires too many operations as well.

The rotation transform could be replaced with 3-step shear transform by matrix decomposition as (2) [3]:

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma \\ 0 & 1 \end{bmatrix} \quad (2)$$

Here, $\alpha = \gamma = -\tan\theta/2$, $\beta = \sin\theta$. It means that the rotation transform could be accomplished with just 3 translation (shear) transforms without float multiplications. It is very efficient and fast.

In this paper, we will show that the 3-step shear operations can be reduced to 2-step with a little additional processing.

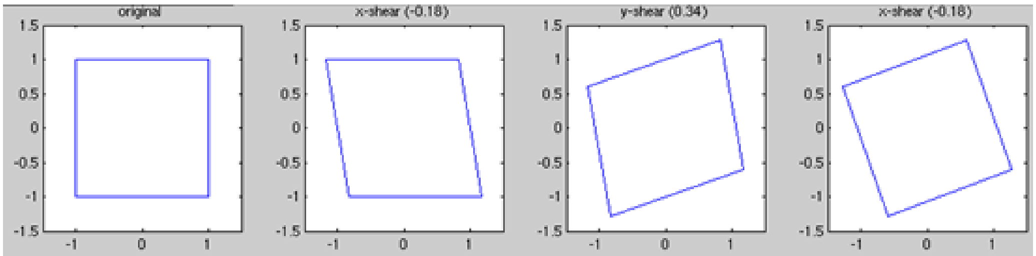


Fig. 2 Rotation by 3-step shear transforms

III. High Speed Image Rotation

A. Digital Differential Analyzer (DDA)

We introduce DDA idea to our rotation transform algorithm. In computer graphics, DDA is used for linear interpolation of variables over an interval between start and end point. DDAs are used for rasterization of lines, triangles and polygons. The DDA chooses the smaller of dy or dx, the unit increment in the y or x axis direction, to start the algorithm. Then, the point on the given segment is sampled at a position shifted by a unit increment from the previous point, and an integer value closest to the point is assigned to the new coordinate. If the slope is positive and less than or equal to 1, the sample is performed at a unit increment x (dx = 1), and the corresponding successive y value is calculated as

$$y += m$$

It is repeated from start to end point of x by unit increment. y value is rounded off to nearest integer. [2].

DDA algorithm has a drawback, which is a round-off error while converting a floating point variable to an integer one. However, it is fast and has small number of operations. So, it is very efficient and used in many applications [4]-[6]. We apply the idea of DDA's unit increment to our rotation algorithm.

Fig. 3 Incremental line drawing by DDA

B. High Speed Image Rotation by 2-Step Shear Transforms

Generally, in image pattern recognition system, input images are slanted. It should be oriented correctly. A slant angle is needed for the correction. We find it by probing 3 positions on the horizontal upper line and 3 other positions from the left vertical line. Then the angle could be represented with two unit increments, WX and HY.

$$\mathrm{WX} = \left\lfloor \frac{A.x - B.x}{A.y - B.y} \right\rfloor \qquad \mathrm{HY} = \left\lfloor \frac{A'.y - B'.y}{A'.x - B'.x} \right\rfloor \tag{3}$$

Equation (3) determines a horizontal increment and a vertical one. Let us define the major slope (horizontal increment) as WX, and the minor slope (vertical increment) as HY. The image rotation process using WX and HY consists of two steps. WX denotes the number of X pixels per a Y pixel and HY refers to the number of Y pixels per an X pixel. To handle one horizontal line of pixels from the input image, the horizontal increment (or decrement) of the line should be calculated, firstly. Next, the vertical increment of each pixel is calculated. Finally, a rotated image is made from the translation transform.

(a) (b)

Fig. 4 Input image rotation (a) before (b) after

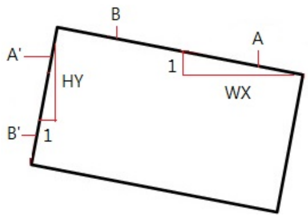


Fig. 5 Horizontal and vertical unit increments

There are two cases in slanted images (Fig. 6). The orientation could be determined by a comparison WX and HY. Case (a) in the figure has a positive WX and a negative HY. Case (b) has a reverse value. Then, the image could be rotated correctly.

(a) (b)

Fig. 6 Two cases of a slanted image

When rotating pixels in a given image using WX, in the case of (a), the pixels are moved upward while proceeding from left to right as shown in Fig. 7 (top alignment). In the case of (b), the pixels are moved downward while proceeding from left to right. When rotating pixels in the image using HY, in the case of (a), the pixels are moved to the left while proceeding from bottom to top as shown in Fig. 8 (left alignment). In the case of (b), the pixels are moved to the right while proceeding from top to bottom. For example, If WX = 2, the movement in the X-axis by 2 results in the pixels lower by 1 in the Y-axis. It should be noted that there is a possibility of bringing some pixels out of range. In that case, the pixels are filled with 0 (where 0 means the background). If HY = 2, the movement in the Y-axis by 2 results in the pixels right by 1 in the X-axis. This also has the same possibility of bringing some pixels out of range. In that case, the same post-processing should be applied.

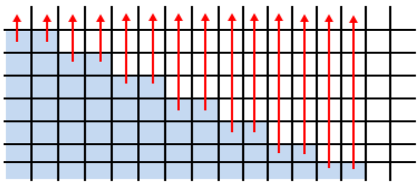


Fig. 7 Translation with WX=2

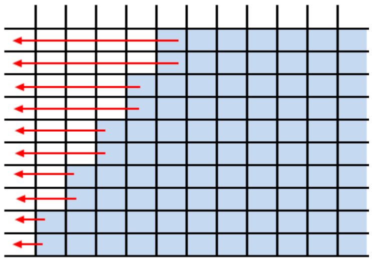


Fig. 8 Translation with HY=2

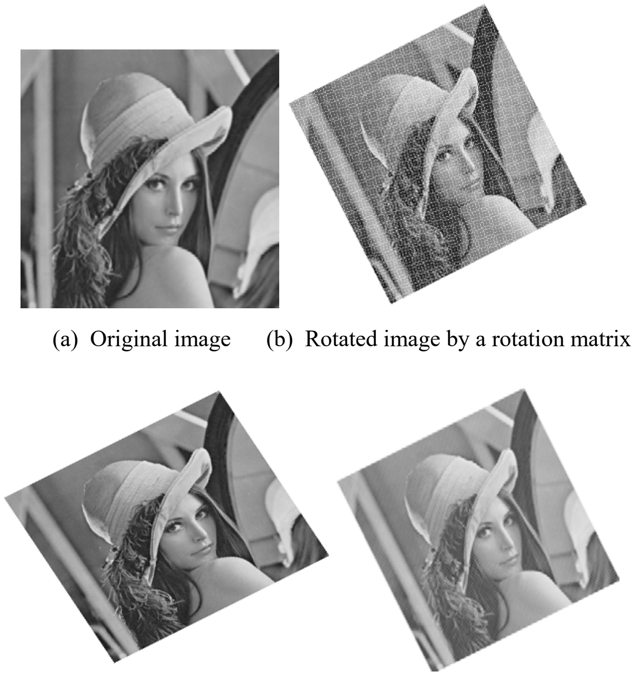
The resulting image after rotating a given image using WX and HY is shown in Fig. 4 (b). After the slant correction, the shape of a complete banknote could be obtained. [7]

C. Augmented 2-Step Shear Transforms

However, precisely speaking, a rotated image by the 2-step shear transform is different from the original image, like in Fig. 9 (c). There is a serious distortion in it. The larger the rotation angle, the larger the distortion. In Fig. 10, the rectangular 1 is an original, and the 2 is a result of the first step, and the 3 is a result of the second step. We can found a distortion in the 3. The pixels on the edges are not even. Its corners are not a right angle. This is because the shear transformation of the second step is performed based on the previous Y-axis even though the Y-axis is moved in the first step, too. So, we solved this problem by starting a second shear transform on the shifted Y-axis, i.e. a line segment AB in Fig. 10.

IV. Experiments and Analysis

In order to demonstrate the performance of the presented algorithm, time-complexity analysis and experiments are conducted with various size images for the three algorithms: Conventional rotation matrix algorithm, its table look-up version, and the presented one.



(a) Original image (b) Rotated image by a rotation matrix

(c) Rotated image by 2-step (d) Rotated image by proposed one

Fig. 9 Rotation quality comparison with a natural image

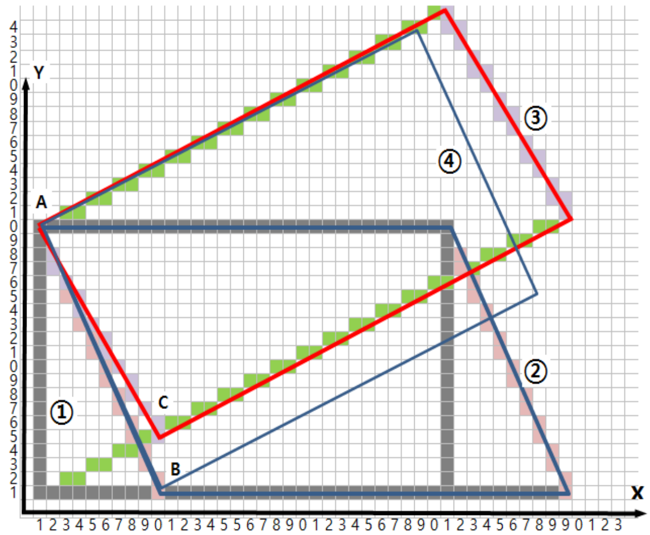


Fig. 10 2-Step shear transform and augmented one

In the case of the conventional rotation matrix algorithm, it needs two sine and cosine function evaluation, four floating point multiplications, two floating point additions and one comparison for a pixel. The trigonometric function evaluation is a time consuming process. We use a pre-evaluated values table look-up method and remove the trigonometric function evaluation step in the second algorithm. It takes a relatively long time for the floating point operations, too. Finally, we substitute the operations with integer ones using the idea of DDA's unit increment. It needs only two comparisons and two addition operations.

The experimental results using a banknote in Fig. 4 show that the proposed one is faster than the others with ten or two times as shown in Table I. Regardless of image size, the performance comparison is constant.

TABLE I
PERFORMANCE COMPARISON (MS)

Image Size	Rotation Matrix		Proposed Algorithm
	No Table	Table	
100 by 100	0.85	0.13	0.06
500 by 500	17.86	3.13	1.46
1000 by 1000	69.80	12.24	5.97
1500 by 1500	157.13	28.41	13.74
2000 by 2000	282.50	52.84	24.88
5000 by 5000	1748.60	349.83	162.29

V. CONCLUSION

In this paper, we present a high speed image rotation algorithm using augmented 2-step shear operations. The time complexity of the algorithm significantly lowered in terms of the number of operations. The floating point operations are substituted with integer ones. There is no trigonometric functions and interpolation. Eventually a given image could be rotated in real-time. The overall speed improvement was achieved.

Image quality after the rotation is very important. Fig. 9 shows four images: (a) Original image, (b) Rotated ones by rotation transform matrix multiplication, (c) By previous 2-step shear operations, and (d) By presented method. The conventional method usually rotates an image with rotation matrix multiplication and interpolates the result for the anti-aliasing effect. In this case, there is a big difference between the original image and the conventional result (Fig. 9 (b)). Therefore, it requires an additional interpolation process. It is not good for the pattern recognition process. But the proposed method generates a precise image (Fig. 9 (d)). There is no loss, even a pixel.

In order to compare with a natural image, we use Lena image. The conventional method rotates the image naturally (Fig. 9 (b)). There is a little distortion and artifacts in the image rotated by the previously proposed our method (Fig. 9 (c)). But the augmented one has no distortion (Fig. 9 (d)).

There is slight accuracy degradation due to the difference between city-block distance used by the presented method and the Euclidean distance used by a conventional one. But the loss of the rotated image data compared with the original one is not meaningful in a small rotation angle.

ACKNOWLEDGMENT

This work (Grants No. S2401435) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2016.

REFERENCES

[1] A. K. Jain, *Fundamentals of digital image processing*. Prentice-Hall International, 1989.
[2] E. Angel, *Interactive Computer Graphics.* Pearson International Edition, 2009, pp. 197–220.
[3] Alan W. Paeth, “A Fast Algorithm for General Raster Rotation”, Graphics Interface ‘86 (5).
[4] M. S. Park, J. Y. Choi, "Datawise Discriminant Analysis for Feature Extraction", Journal of Korean institute of intelligent systems, v.19 no.1, 2009, pp.90-95
[5] S. W. Shin, K. S. Kim, S. M. Lee, C. G. Song, "Image Enhancement with Rotating Kernel Transformation Filter Generated by Bresenham's Algorithm", The Korean Institute of Electrical Engineers, Volume 61, Issue, 6, 2012, pp.872-878
[6] L. Moroz, J. L. Cieslinski, M. Stakhiv, and V. Maksymovych, “A Comparison of Standard One-Step DDA Circular Interpolators with a New Cheap Two-Step Algorithm”, Hindawi Publishing Corporation, Modelling and Simulation in Engineering, Volume 2014.
[7] H. C. Kwon, H. J. Cho, and H. Y. Kwon, “High Speed Image Rotation Algorithm”, APRIL 08-09, 2015 International Science Index Marrakech Morocco, 17(4) Part XX, 2015, pp. 2006-2009.