African Journal of Physics Vol. 2 (5), pp. 010-017, May, 2014. Available online at www.internationalscholarsjournals.org © International Scholars Journals

Full Length Research Paper

Rotation matrix factorization Impact on the quality of images rotation

Kingsley Nathaniel* and Innocent Saffin

Department of Electrical Engineering, Energetic and Automatics, ENSAI, The University of Ngaoundéré, PO Box 455 Ngaoundéré – Cameroon.

Accepted 5 October, 2013

Most efficient algorithms for rotating 2D images are based on the succession of three translations, following the lines, then the columns and the lines of the image again (LCL). These translations result from the decomposition of the rotation matrix into a product of an upper triangular matrix, a lower triangular matrix and an upper triangular matrix (ULU). We have shown in this paper that the decomposition of the rotation matrix is not unique and have separated it into a product of a lower triangular matrix, an upper triangular matrix and a lower triangular matrix (LUL). This new decomposition led to a new algorithm based on a succession of three translations following the columns, then the lines and the columns of the images again (CLC). Statistical analysis of experimental results showed that the computational complexity of images rotation does not depend significantly (p>0.05) on the factorization ULU or LUL, whereas the precision depends on it significantly (p<0.05). Each of the two algorithms of rotation is more precise for certain images and for certain angles. The method can be generalized in a case of 3D image rotation using the Euler angles.

Key words: Images rotation, rotation matrix, matrix factorization, LCL algorithm, CLC algorithm.

INTRODUCTION

Rotation of a 2D image about its Cartesian origin can be accomplished by translating coordinates (x, y) of this image into coordinates ($x_r = x \cos - y \sin$, $y_r = x \sin + y \cos$), where is the counter-clockwise angle of rotation with respect to the horizontal axis of the input image (Pratt 1997). The vector-space representation of rotation is given by:

$$\begin{pmatrix} x_r \\ y_r \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\ ) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \theta$$

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$ being the rotation matrix.

The new coordinates x_r and y_r correspond to new pixels, generally obtained by interpolation. The basic principle of all methods of interpolation of images consists to determine the parameter of a representation of an analogical image obtained from the digital image. Different interpolation functions were developed, nearest neighbor, linear, cubic and quadratic B-Spline, etc. (Thevenaz et al., 2000). The quality of an algorithm for rotating images, in terms of precision, is directly related to the order of accuracy of the underlying interpolation mo-del (Schoenberg, 1946). Nearest neighbor functions are simple to implement but are likely to produce images with noticeable artifacts (Pratt, 1978). More satisfactory re-sults can be obtained using B-spline functions (Hou and Andrews, 1978; Keys, 1981; Unser et al., 1991). In terms of computational efficiency, the quality of images rotation depends on hardware architectures as well as algorithms (Suchitra et al., 2006).

The work presented in this paper is set in the framework of algorithms for rotating 2D images developed by Unser et al. (1995), Owen and Makedon (1997), algorithms based on a succession of translations of the lines and the columns of the image implemented by linear 1-D filtering. These translations result from the decomposi-

*Corresponding author. E-mail: kingsleynath@yahoo.com.Tel: +237 99 59 96 82. Fax: +237 22 25 27 51.

$R(\theta)$ into a product of triangular matrices. The first decomposition of $R(\theta)$ into a product of two triangular matrices were implemented in an algorithm for rotating images by Catmul and Smith (1980), Friedman (1981) and Fraser (1989):

$$R(\theta) = \begin{pmatrix} 1 & 0 \\ \tan(\theta) & \frac{1}{\cos(\theta)} \end{pmatrix} \times \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ 0 & 1 \end{pmatrix} \qquad (1)$$

Paeth (1986), Danielson and Hammerin (1992) factorized $R(\theta)$ into a product of three triangular matrices, an upper triangular matrix, a lower triangular matrix and an upper triangular matrix:

$$R(\theta) = \begin{pmatrix} 1 & -\tan(\theta/2) \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 \\ \sin(\theta) & 1 \end{pmatrix} \times \begin{pmatrix} 1 & -\tan(\theta/2) \\ 0 & 1 \end{pmatrix} \qquad (2)$$

The Upper-Lower -Upper (ULU) factorization gives in relation (2) leads to algorithms based on the succession of three translations, following the lines, then the columns and the lines of the image again called three-pass rotation algorithms. We demonstrate in section 2 that this factorization is not unique and give an alternative decomposition which leads in section 3 to a new algorithm. In section 4 we carried out rotation experiments with various algorithms based on the one hand on the ULU factorization and on the other hand on our factorization, and compare the quality of images obtained.

Generalised decomposition of the rotation matrix

The rotation matrix $R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$ can be decomposed into a product of three triangular matrices according to four transformations resulting from the expansion of $\cos(\theta)$ and $\sin(\ )$ given by relations (7) and (8) respectively. These transformations derive from three basic trigonometric relations (3), (4) and (5).

$$1 - \cos(\theta) = 2\sin^2(\theta/2) \qquad (3)$$

$$1 + \cos(\theta) = 2\cos^2(\theta/2) \qquad (4)$$

$$\sin(\theta) = 2\sin(\theta/2)\cos(\theta/2) \qquad (5)$$

From relation (5) we deduce

$$\sin(\theta/2) = \frac{1}{2}\frac{\sin(\theta)}{\cos(\theta/2)}, \text{ i.e.} \qquad (6)$$

$$\sin^2(\theta/2) = \frac{1}{2}\sin(\theta)\tan(\theta/2) \qquad (7)$$

Relations (3) and (6) give $\cos(\theta) = 1 - \sin(\theta)\tan(\theta/2)$

By dividing relation (5) by relation (4) we obtain relation (8)

$$\sin(\theta) = (1 + \cos(\theta))\tan(\theta/2) \qquad (8)$$

First transformation: transformation of the elements of the first line of $R(\theta)$

By replacing the elements of the first line of the rotation matrix $R(\theta)$ by the expressions of the relation (7) and (8), we obtain

$$R(\theta) = \begin{pmatrix} 1 - \sin(\theta)\tan(\theta/2) & -(1+\cos(\theta))\tan\frac{\theta}{2} \\ \sin(\theta) & \cos(\theta) \end{pmatrix},$$

which can be written as a product of the following three triangular matrices.

$$R(\theta) = \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ \sin(\theta) & 1-\sin(\theta)\tan\frac{\theta}{2} \end{pmatrix} \qquad (9)$$

$$= \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 \\ \sin(\theta) & 1 \end{pmatrix} \times \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix}$$

This decomposition is a product of three matrices, an upper triangular matrix, a lower triangular matrix and an upper triangular matrix (ULU factorization).

Second transformation: transformation of the elements of the second column of $R(\theta)$

By replacing the elements of the second column of the rotation matrix $R(\theta)$ by the expressions of the relation (7) and (8), we obtain

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -(1+\cos(\theta))\tan\frac{\theta}{2} \\ \sin(\theta) & 1-\sin(\theta)\tan(\theta/2) \end{pmatrix}$$

which can be written as a product of the following three triangular matrices.

$$R(\theta)=\begin{pmatrix}\cos(\theta) & -\tan\frac{\theta}{2}\\ \sin(\theta) & 1\end{pmatrix}\times\begin{pmatrix}1 & -\tan\frac{\theta}{2}\\ 0 & 1\end{pmatrix} \qquad (10)$$

$$=\begin{pmatrix}1 & -\tan\frac{\theta}{2}\\ 0 & 1\end{pmatrix}\times\begin{pmatrix}1 & 0\\ \sin(\theta) & 1\end{pmatrix}\times\begin{pmatrix}1 & -\tan\frac{\theta}{2}\\ 0 & 1\end{pmatrix}$$

This decomposition is a product of three matrices, an upper triangular matrix, a lower triangular matrix and an upper triangular matrix (ULU factorization).

Third transformation: transformation of the elements of the second line of $R(\theta)$

By replacing the elements of the second line of the rotation matrix $R(\theta)$ by the expressions of the relation (7) and (8), we obtain

$$R(\theta)=\begin{pmatrix}\cos(\theta) & -\sin(\theta)\\ (1+\cos(\theta))\tan\frac{\theta}{2} & 1-\sin(\theta)\tan\left(\frac{\theta}{2}\right)\end{pmatrix}$$

which can be written as a product of the following three triangular matrices.

$$R(\theta)=\begin{pmatrix}1 & 0\\ \tan\frac{\theta}{2} & 1\end{pmatrix}\times\begin{pmatrix}\cos(\theta) & -\sin(\theta)\\ \tan\frac{\theta}{2} & 1\end{pmatrix} \qquad (11)$$

$$=\begin{pmatrix}1 & 0\\ \tan\frac{\theta}{2} & 1\end{pmatrix}\times\begin{pmatrix}1 & -\sin(\theta)\\ 0 & 1\end{pmatrix}\times\begin{pmatrix}1 & 0\\ \tan\frac{\theta}{2} & 1\end{pmatrix}$$

This decomposition is a product of three matrices, a lower triangular matrix, an upper triangular matrix and a lower triangular matrix (LUL factorization).

Fourth transformation: transformation of the elements of the first column of $R(\theta)$

By replacing the elements of the first column of the rotation matrix $R(\theta)$ by the expressions of the relation (7) and (8), we obtain

$\begin{pmatrix}1-\sin(\theta)\tan\left(\frac{\theta}{2}\right) & -\sin(\theta)\\ (1+\cos(\theta))\tan\frac{\theta}{2} & \cos(\theta)\end{pmatrix}$ which can be written as a product of the following three triangular matrices.

$$R(\theta)=\begin{pmatrix}1 & -\sin(\theta)\\ \tan\frac{\theta}{2} & \cos(\theta)\end{pmatrix}\times\begin{pmatrix}1 & 0\\ \tan\frac{\theta}{2} & 1\end{pmatrix}$$

$$=\begin{pmatrix}1 & -\sin(\theta)\\ \tan\frac{\theta}{2} & 1-\sin(\theta)\tan\frac{\theta}{2}\end{pmatrix}\times\begin{pmatrix}1 & 0\\ \tan\frac{\theta}{2} & 1\end{pmatrix}$$

$$=\begin{pmatrix}1 & 0\\ \tan\frac{\theta}{2} & 1\end{pmatrix}\times\begin{pmatrix}1 & -\sin(\theta)\\ 0 & 1\end{pmatrix}\times\begin{pmatrix}1 & 0\\ \tan\frac{\theta}{2} & 1\end{pmatrix}$$

This decomposition is a product of three matrices, a lower triangular matrix, an upper triangular matrix and a lower triangular matrix (LUL factorization).

Application to the rotation of images

2D image rotation

The first two decompositions of the rotation matrix are the ULU factorizations, which generate classical algorithms of rotating images based on a succession of three translations:

Translation of the image following the lines by applying an upper triangular matrix,

Translation of the image following the columns by applying a lower triangular matrix,

Translation of the image following the lines by applying an upper triangular matrix.

The last two decompositions of the rotation matrix are the LUL factorizations, where the lower triangular matrix is $\begin{pmatrix}1 & 0\\ \tan\frac{\theta}{2} & 1\end{pmatrix}$ and the upper matrix $\begin{pmatrix}1 & -\sin(\theta)\\ 0 & 1\end{pmatrix}$. A pixel with coordinates (x,y) to which the lower triangular matrix $\begin{pmatrix}1 & 0\\ \tan\frac{\theta}{2} & 1\end{pmatrix}$ is applied is translated at the coordinates $\left(x,\ y+x\tan\left(\frac{\theta}{2}\right)\right)$. Its coordinates therefore undergoes a translation of $+x\tan\left(\frac{\theta}{2}\right)$ in the y direction. As a consequence, the columns of an image on which this matrix is applied are translated to the left at each point (x,y) of $x\tan\left(\frac{\theta}{2}\right)$ and the image is sheared in the y direction. When the upper triangular matrix $\begin{pmatrix}1 & -\sin(\theta)\\ 0 & 1\end{pmatrix}$ is applied to coordinates, it is translated at the coordinates $(x-y\sin(\theta),y)$. The coordinates (x,y) therefore undergoes a translation of $-y\sin(\theta)$ in the x direction. As a consequence, the lines of an image on which this matrix is applied are translated to the right at each point (x,y) of $y\sin(\theta)$ and the image is sheared in the x direction. We conclude that the last two decompositions of the rotation matrix yield algorithms of rotating images based on the succession of the following three translations:

Translation of the image following the columns by applying the:

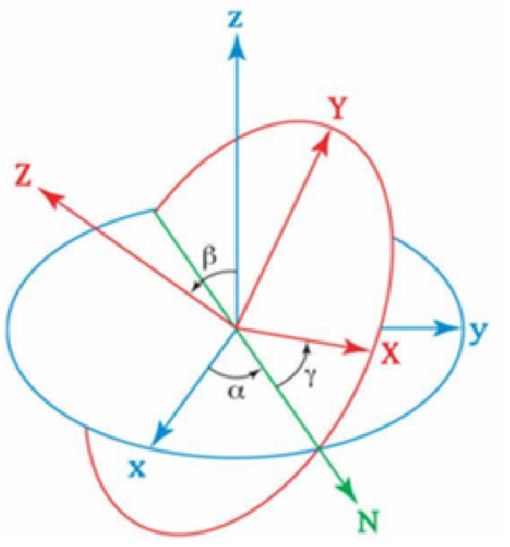


Figure 1. Euler angles - The xyz (fixed) system is shown in blue, the XYZ (rotated) system is Shown in red. The line of nodes, labelled N, is shown in green.

1 0
θ lower triangular matrix,
tan 1 2

Translation of the image following the lines by applying

the $1 \quad -\sin(\theta)$ upper triangular matrix,

0 1

Translation of the image following the columns by

applying the $\begin{matrix} 1 & 0 \\ \tan \frac{6}{2} & 1 \end{matrix}$ lower triangular matrix.

At the end of these three translations, following the columns, then the lines and finally the columns (CLC) a pixel of coordinates (x, y) is displaced at the coordinates $(x - (y + x\tan(\ 2))\sin\ , y + x\tan(\ 2) + [x - (y + x\tan(\ 2))\sin\ ]\tan(\ 2))$. The determinant of each of the elementary matrices in (9), (10), (11) and (12) is one, indicating that all areas in the image are preserved throughout the translation process (Unser et al., 1995).

3D image rotation

In a case of 3D image rotation using the Euler angles (Figure 1), the rotation matrixes R is defined as follow.

$$R = \begin{matrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{matrix} \begin{matrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{matrix} \begin{matrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{matrix}$$

R is a product of three square (3x3) matrices, R_γ, R_β and R_α.

$$R_\gamma = \begin{matrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{matrix},$$

$$R_\beta = \begin{matrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{matrix},$$

$$R_\alpha = \begin{matrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{matrix}$$

Each of these matrixes can be decomposed into a product of a lower triangular matrix, an upper triangular matrix and a lower triangular matrix (LUL). Then, R becomes $R_{\gamma L} R_{\gamma U} R_{\gamma L}\ R_{\beta L} R_{\beta U} R_{\beta L}\ R_{\alpha L} R_{\alpha U} R_{\alpha L}$, which is a LULULUL matrix that leads to an algorithm based on a succession of 7 translations following the x, y

RESULTS AND DISCUSSION

Experimental setting

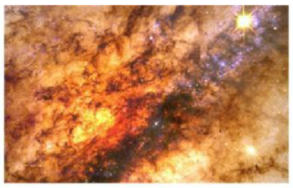
The two types of factorization of the rotation matrix, the LUL and the ULU factorizations were compared in their efficiency for implementing the rotation of 2D images. The efficiency of the algorithms for rotating images derived from the two types of factorization was evaluated on two criteria of performance: the residual root mean square (RMS) error and the CPU time in Intel Pentium III 670 MHz PC with 256 MB RAM. The Rotation was implemented in a separable fashion using Matlab routines (Gonzalez et al. 2004). Three interpolation functions were considered: nearest-neighbor, linear B-spline, cubic B-spline. In our experiments, three square (256 x 256) ima-(Figure 2) and three rectangular images (Figure 3) were used. The residual RMS error was computed for a series of back and forth rotations at various angle ranging from 5° to 45°, on the 128 x128 central portion of the image to factor out boundary effects. The CPU time was measured after a complete or z direction or z direction of four suc-

a)

b)

c)

Figure 2. Square (256 x 256) images involved in our experiments, a) Fra, b) Lena, c) Bongou.

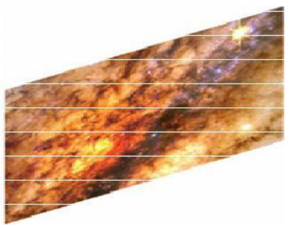


a)

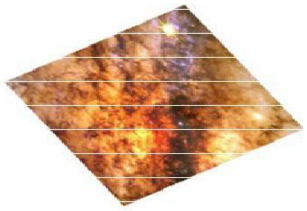
b)

c)

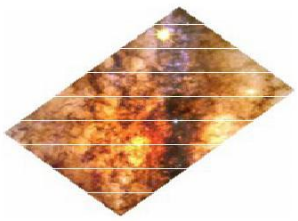
Figure 3. Rectangular images (Photo © NASA / Hubble heritage team, STSci / Aura) involved in our experiments, a) Centaurus (358x540), b) Sombrero (302 x 540), c) Whirlpool (358 x 540).



a)



b)



c)

Figure 4. Illustration of the three-pass rotation algorithm derived from the ULU factorization of the rotation matrix (Original image: Centaurus), a) Shearing along the y-axis b) Shearing along the x-axis c) Shearing along the y-axis which yields the rotated image.

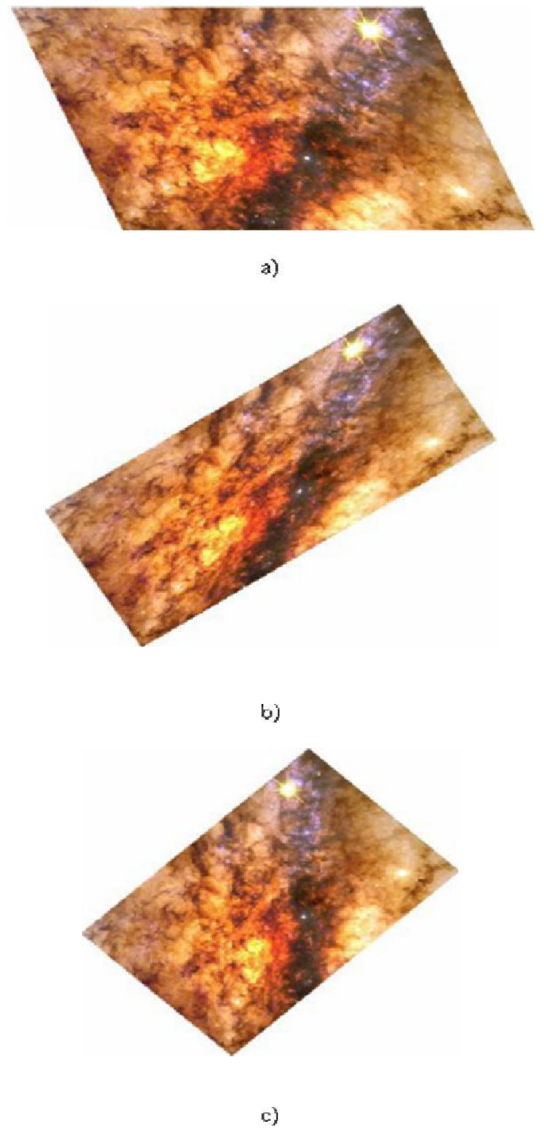


Figure 5. Illustration of the three-pass rotation algorithm derived from the LUL factorization of the rotation matrix (Original image: Centaurus), a) Shearing along the x-axis b) Shearing along the y-axis c) Shearing along the x-axis which yields the rotated image.

cessive rotations of 90°, at the end of which the image is back in its initial position. The residual RMS error measure was repeated twice, and the CPU time measured was repeated five times. Results were statistically analyzed using Statgraphics and Statistica to know if the LUL and the ULU factorizations are significantly different in terms of efficiency of the algorithms for rotating ima-ges, and estimate the amount of variability contributed by each of the factors influencing this result.

Figures 4 and 5 illustrate the images rotation with each algorithm. The rotated images obtained from an original image do not exhibit the same characteristics.

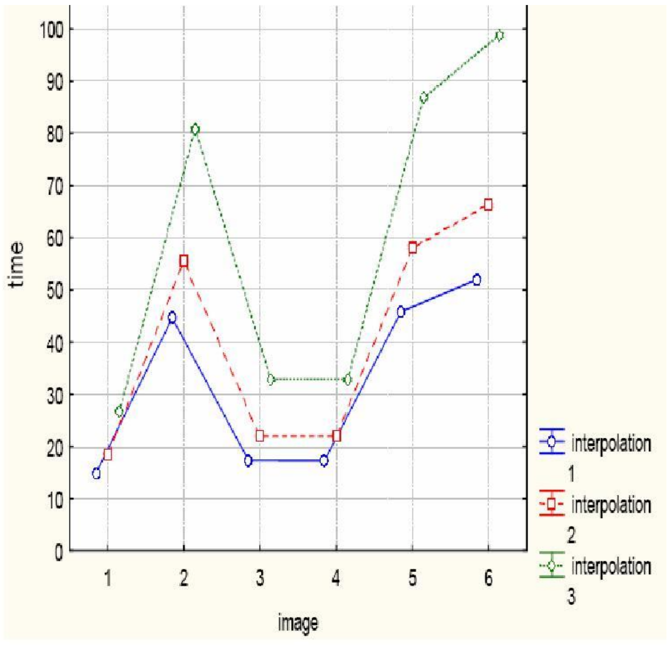
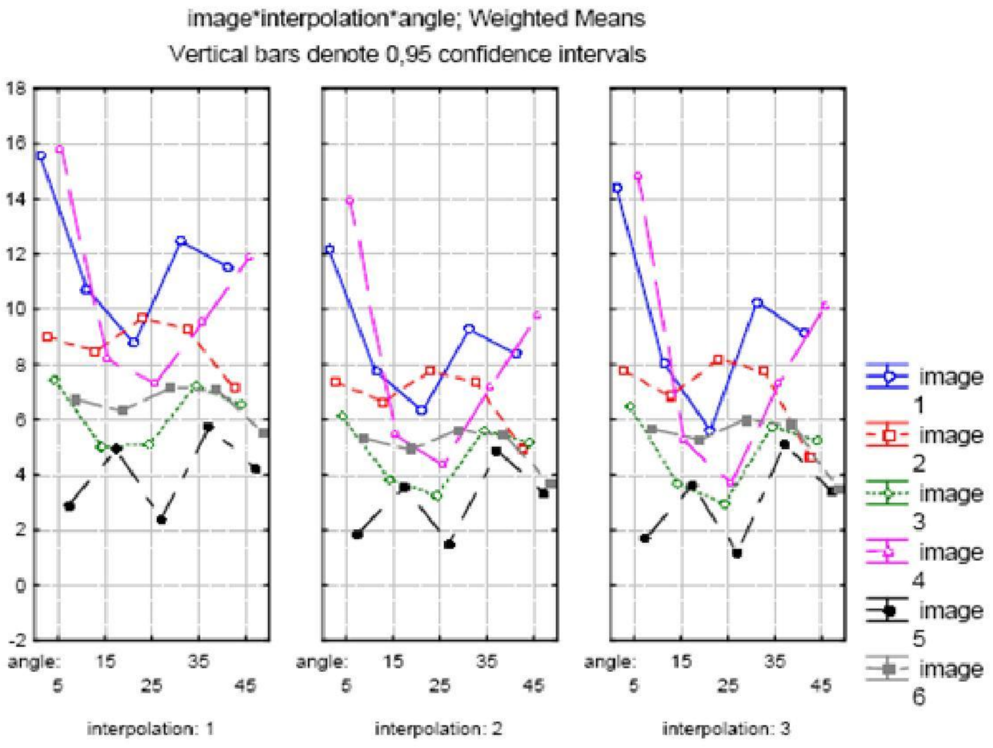


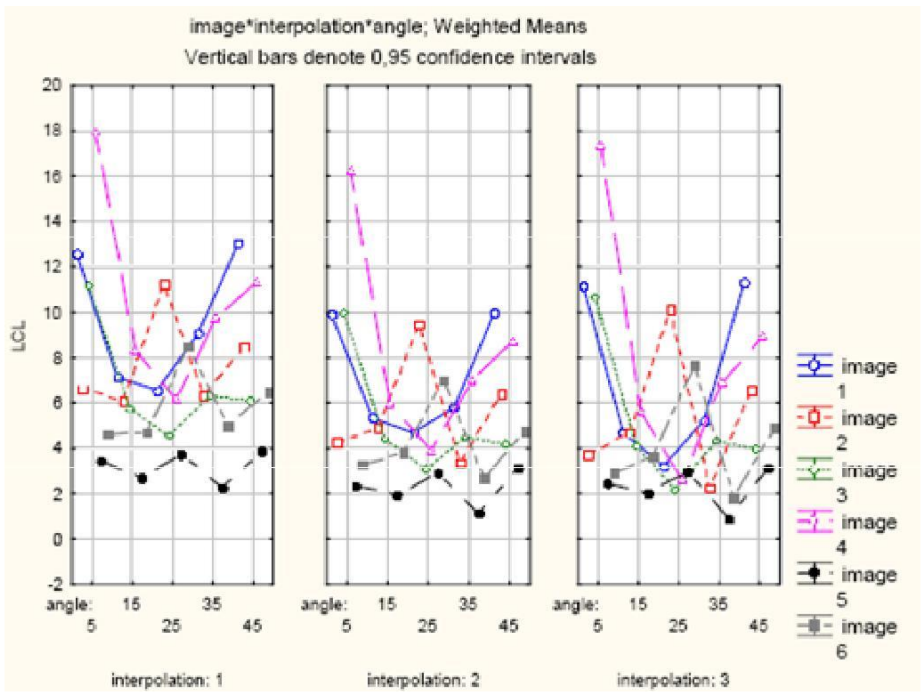
Figure 6. The CPU time (s) of rotation of images. Image 1: “Bongou”, Image 2: “Centaurus”, Image 3: “Fra”, Image 4: “Lena”, Image 5: “Sombrero”, Image 6: “Whirlpool”. Interpolation 1: nearest-neighbor, Interpolation 2: linear B-spline, Interpolation 3: cubic B-spline.

The CPU time of rotation

The analysis of variance (ANOVA) of the CPU time measured after a complete rotation, by a series of four successsive rotations of 90°, at the end of which the image is back in its initial position, shows that this CPU time depends significantly ($p<0.05$) on the image and the interpolation function but does not depend significantly ($p>0.05$) on the decomposition ULU or LUL of the rotation matrix. Unser et al. (1995) reported this dependence, but the amount of variability contributed by each of the factors was not evaluated. The variance components analysis of this time of rotation, which estimates the amount of variability contributed by each of the factors, shows that the factor contributing the most variance is the image whose contribution represents 65.4% of the total variation in time, followed by the interpolation function (34.6%). The analysis of variance (ANOVA) for the CPU times measured with each algorithm for rotating images (LCL, CLC) derived from the two types of factorization shows that they are not significantly different. This result indi-cates that the computational complexity of the two algorithms is identical. The nearest-neighbor interpolation func-tion gives the lowest time of rotation, followed by the linear B-spline function and the cubic B-spline function (Figure 6). This result is in agreement with other studies (Unser et al. 1995, Thevenaz et al., 2000). We observe in Figure 6



a)



b)

Figure 7. The RMS error measured for the six images of Figures 1 and 2. a) CLC algorithm b) LCL algorithm. Image 1: “Bongou”, Image 2: “Centaurus”, Image 3: “Fra”, Image 4: “Lena”, Image 5: “Sombrero”, Image 6: “Whirlpool”. Interpolation 1: nearest-neighbour, Interpolation 2 : linear B-spline, Interpolation 3 : cubic B-spline.

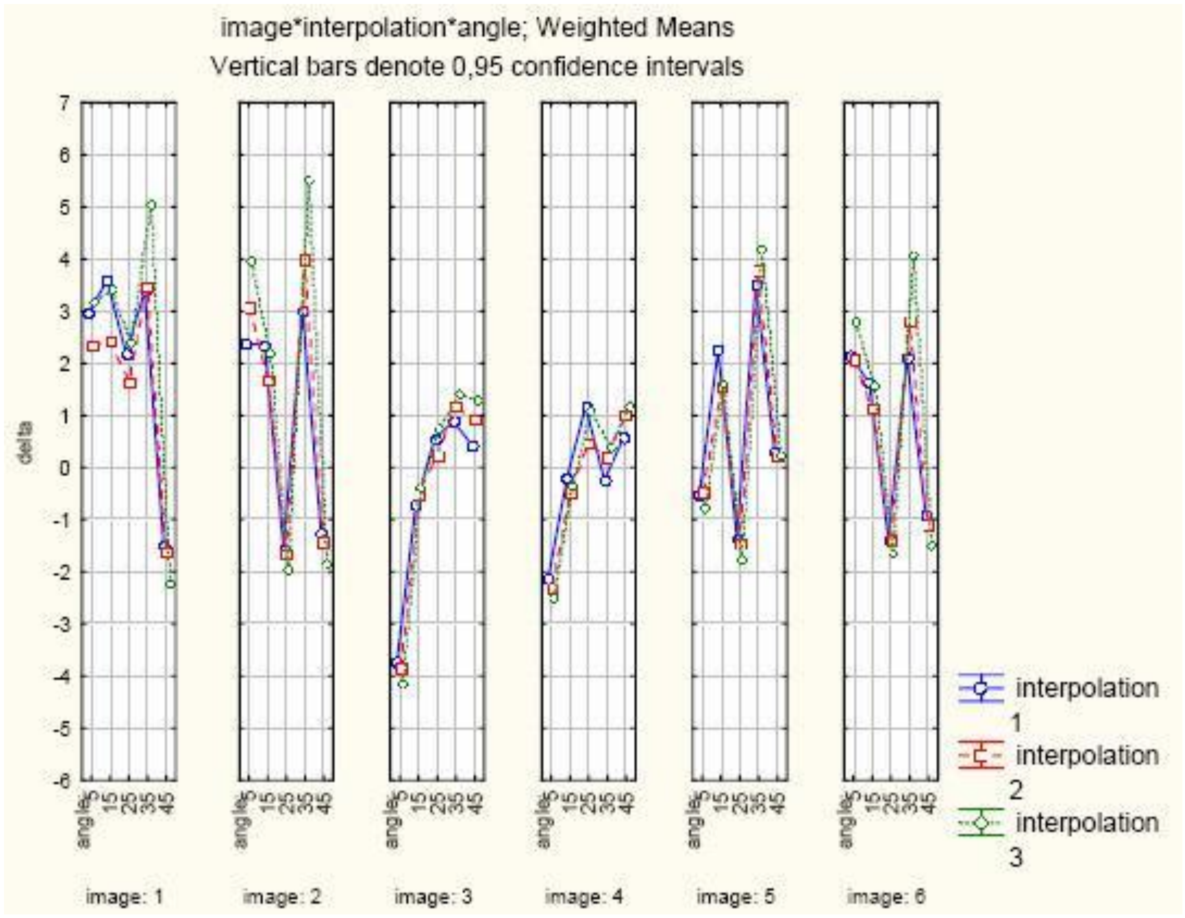


Figure 8. The difference of RMS error calculated between CLC algorithm and LCL algorithm. Image 1: “Bongou”, Image 2: “Centaurus”, Image 3: “Fra”, Image 4: “Lena”, Image 5: “Sombrero”, Image 6: “Whirlpool”. Interpolation 1: nearest-neighbour, Interpolation 2: linear B-spline, Interpolation 3: cubic B-spline.

that the time of rotation is proportional to the size of the image.

The residual root mean square (RMS) error

The analysis of variance (ANOVA) of the residual RMS error computed for a series of back and forth rotations at various angle ranging from 5° to 45°, on the 128 x 128 central portion of the image shows that this RMS error depends significantly ($p<0.05$) on the image, the interpolation function, the angle and the decomposition ULU or LUL of the rotation matrix. Each algorithm for rotating images (LCL, CLC) derived from the two types of factorization depends significantly ($p<0.05$) on the image (86.6% for CLC and 84% for LCL), the interpolation function (12.9% for CLC and 11.6% for LCL) and the angle (0.5% for CLC and 4.4% for LCL). Our results (Figure 7) showing the RMS error measured for six images are in agreement with earlier reports (Unser et al. 1995). The analysis of variance (ANOVA) of the difference computed between the RMS errors measured for each rotating images algorithm, called “delta”, shows that it depends significantly ($p<0.05$) on the image and the angle but does not depend significantly ($p>0.05$) on the interpolation function. The variance components analysis of “delta”, which estimates the amount of variability contributed by each of the factors, shows that the factor contributing the most variance is the image whose contribution represents 73.1% of the total variation in RMS errors, followed by the angle (22.2%). This difference “delta” plotted in Figure 8 indicates that each of the two algorithms of rotation is more precise for certain images and for certain angles. For example, algorithm CLC is more precise at 45° for images 1, 2 and 6, whereas algorithm LCL is more precise at 5° for images 1, 2 and 6.

Conclusion

We have shown in this paper that the decomposition of the rotation matrix is not unique and have separated it into a product of a lower triangular matrix , an upper triangular matrix and a lower triangular matrix. This new decomposition led to algorithm based on a succession of three translations following the columns, then the lines and the columns of the images again. Statistical analysis showed that this new algorithm compare very well to the classical algorithm based on the succession of three translations, following the lines, then the columns and the lines of the image again in terms of the time of rotation. Concerning the RMS error, each of the two algorithms of rotation is more precise for certain images and for certain angles.

REFERENCES

Danielson PF, Hammerin M (1992). High-accuracy rotation of images, CVGIP: Graphical models and image processing, 54: 4, 340-344.

Gonzalez RC, Woods RE, Steven LE (2004). Digital Image processing using MATLAB, Prentice Hall.

Hou HS, Andrews HC (1978). Cubic splines for image interpolation and digital filtering, IEEE trans. acoust., speech, signal processing, 26: 508-517.

Keys RG (1981). Cubic convolution interpolation for digital image processing, IEEE trans. acoust., speech, signal processing, 29, 1153-1160.

Owen C, Makedon F (1997). Bottleneck-Free Separable Affine Image Warping; International Conference on Image Processing (ICIP'97), 1: 683

Paeth A. (1986). A fast algorithm for general raster rotation, in proc. Graphics interface 86: 77-81.

Pratt WK (1978). Digital Images Processing; John Wiley & Sons, New York.

Pratt WK (1997). Digital Images Processing, 2nd edition, John Wiley & Sons, New York.

Suchitra S, Lam SK, Clarke CT, Srikanthan T (2006). Accelerating rotation of high-resolution images, IEEE proceedings. Vision, image and signal processing, 153 (6): 815-824.

Thevenaz P, Blu T, Unser M (2000). Image Interpolation and Resampling, in Handbook of Medical Imaging, Processing and Analysis, Bankman Ed., Academic Press, San Diego Ca, USA,pp 393-420.

Unser M, Aldroubi A, Eden M (1991). Fast B-spline transforms for continuous representation and interpolation, IEEE trans.pattern anal. machine intell.13: 277-285.

Unser M, Thevenaz P, Yaroslavsky L (1995). Convolution-based interpolation for fast, high-quality rotation images; IEEE transactions on image processing, 4(10):1371-1381.