Algorithmic support for graphic images rotation in avionics

E V Kniga¹, A V Gurjanov¹, A V Shukalov², I O Zharinov²

¹ Learning-Science Centre, Stock Company «Experimental Design Bureau «Electroavtomatika» named after P A Yefimov, 40, Marshala Govorova St., Saint Petersburg, 198095, Russia
² Department of Computer Design of On-Board Computer Equipment, Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, 49, Kronverksky Av., Saint Petersburg, 197101, Russia

E-mail: mpbva@mail.ru

Abstract. The avionics device designing has an actual problem of development and research algorithms to rotate the images which are being shown in the on-board display. The image rotation algorithms are a part of program software of avionics devices, which are parts of the on-board computers of the airplanes and helicopters. Images to be rotated have the flight location map fragments. The image rotation in the display system can be done as a part of software or mechanically. The program option is worse than the mechanic one in its rotation speed. The comparison of some test images of rotation several algorithms is shown which are being realized mechanically with the program environment Altera QuartusII.

1. Introduction

Aircraft control and navigation equipment defines the interaction interface between the crew and the computer system, which is part of a flight control and navigation system. The most informative method for the crew is the indicator frame image, displayed on the LCD (liquid crystal display) or LED (light emitting diode) screen of the multi-function digital indicator [1-3]. Flight parameters values, general equipment conditions etc. are displayed on the indicator screen.

One of the indicator modes is the navigation conditions indication, which contains the indicator frame image of the map in the flight zone. During aircraft flight, the map image shifts on the indicator screen according to the chosen scale and aircraft velocity. The map rotation function [4, 5] on the indicator screen should be provided during the aircraft rotation. The function is realized by the hardware and software of graphic modules. Graphic modules can be part of the multi-function digital indicator or specialized onboard digital systems, which are designed according to integrated modular avionics principles. Processors and components like Field-Programmable Gate Array (FPGA) are the part of the graphic modules. It is shown that it is necessary to use the image construction method, based on the implemented functions of the redistribution program of the computer core and hardware functions implemented by FPGA components, which solve the graphic accelerator task for design of the mapping systems more efficiently by performance criteria.

The image rotation methods, which have the most practical use, are based on: using the direct or inverse sine-cosine rotation matrix; using the Owen-Makedon algorithm [6, 7]. The purpose of the paper is the practical results presentation of different ways of rotation of graphic images with studying the FPGA realization opportunity for a wide range of readers.
2. Image rotation by using direct sine-cosine rotation matrix

Figure 1 presents the image (test example), which is appropriate to use for analysis of the practical realization method of the rotation image function and its modeling in hardware-oriented software packages.

![Figure 1. Test example image.](image)

The test example image consists of three vertical strips of different colors (shades of color) and separate vertical, horizontal and diagonal lines with predetermined thickness. A sine-cosine rotation matrix at any angle $\alpha$ is:

$$
R(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}.
$$

(1)

Coordinate equations of the sine-cosine rotation matrix for rotation relatively any point are:

$$
\begin{align*}
x' &= x_0 + (x - x_0) \cos \alpha + (y_0 - y) \sin \alpha \\
y' &= y_0 + (x - x_0) \sin \alpha + (y - y_0) \cos \alpha,
\end{align*}
$$

(2)

where $(x_0, y_0)$ - point coordinates, relatively which the rotation takes place. That point can be the center of the indicated image, or, for example, the aircraft location projection on the digital map in the defined coordinate system, or the point of the top of the indicated image.

In general, it is appropriate to define the point relatively which the rotation takes place like the indicator screen center, which is combined with the aircraft silhouette position point on the digital map. Then, the (2) for the graphic image of 800 on 800 pixels with a center in the point with coordinate $x_0=400$, $y_0=400$ takes the form as:

$$
\begin{align*}
x' &= 400 + (x - 400) \cos \alpha + (y_0 - y) \sin \alpha \\
y' &= 400 + (x - 400) \sin \alpha + (y - 400) \cos \alpha,
\end{align*}
$$

(3)

where $x$ – graphic image coordinates on the horizontal axis $x \in \{0,1,\ldots,799\}$, $y$ – graphic image coordinates on the vertical axis, $y \in \{0,1,\ldots,799\}$.

Figure 2 presents the results of graphic image rotation, which is realized by software MATLAB on the computer tool, with different rotation angles.

![Figure 2. Testing image on different rotation angles: a) $\pi/6$, b) $\pi/4$, c) $\pi/3$, d) $\pi/2$.](image)
By using the FPGA from the Stratix II family, such realization of hardware schemes of the rotation function takes 8% from all topological elements: 3519 ALUTs (Adapted View of the Table) and 1980 dedicated logical registers.

Figure 3 presents the time diagram of the image rotation algorithm. The following symbols are used: \( \text{clk} \) – clock (100 MHz), \( x \) – coordinate \( x \), which discretely changes from 0 to 5; \( y \) - coordinate \( y \), which is equal to 0; \( \cos \alpha \) – the cosine of the rotation angle, which is equal to 0x40000000; \( \sin \alpha \) – the sine of the rotation angle, which is equal to 0x0. It is shown that coordinates of the rotated image are valid on the data bus in 42.6 ns of modeling time after all calculations.

The image, which is rotated at angles other than 0, \( \pi/2 \), \( \pi \) or \( 3\pi/2 \) by using the direct rotation matrix, consists of image defects — image fragments, which are away from the original image before its rotation. Appearance of those defects in the rotated image can be avoided by using mathematical Owen-Makedon algorithm.

3. Image rotation using mathematical Owen-Makedon algorithm

Owen-Makedon algorithm [6, 7] is based on using the sine-cosine rotation matrix decomposition into following constituents:

\[
R(\alpha) = \begin{bmatrix}
\cos \alpha & -\sin \alpha \\
\sin \alpha & \cos \alpha
\end{bmatrix} = \begin{bmatrix}
1 & -\frac{\text{tg} \alpha}{2} \\
0 & 1
\end{bmatrix} \begin{bmatrix}
1 & 0 \\
\frac{\text{sin} \alpha}{1} & 1
\end{bmatrix} \begin{bmatrix}
1 & -\frac{\text{tg} \alpha}{2} \\
0 & 1
\end{bmatrix}.
\] (4)

Thus, graphic image rotation in Owen-Makedon algorithm is made during three consistent stages. During the first stage, there is lines shift, during the second stage - columns shift, during the third stage - lines shift. Algorithm coordinate equations for each stage of rotation relatively any point are:

\[
x' = x - \frac{\text{tg} \alpha}{2} (y - y_0),
\]

\[
y' = y
\]

\[
x'' = x
\]

\[
y'' = (x' - x_0) \sin \alpha + y''
\]

\[
x''' = x' - \frac{\text{tg} \alpha}{2} (y'' - y_0)
\]

\[
y''' = y''
\] (7)

Figure 4 presents a graphic image after using Owen-Makedon algorithm for rotation, received in software MATLab.
Figure 4. A graphic image after rotation using Owen-Makedon algorithm: a) $\pi/6$, b) $\pi/4$.

Figure 5 presents time diagram of schematics, which realizes Owen-Makedon algorithm for image rotation in software Altera Quartus II. The following symbols are used: $clk$ – clock (100 MHz), $x$ – coordinate $x$, which discretely changes from 2 to 7; $y$ - coordinate $y$, which changes from 0 to 1; $tg\frac{\alpha}{2}$ – the tangent of half the angle, which is equal to 0x0; $\sin \alpha$ – the sine of the rotation angle, which is equal to 0x0. It is shown that coordinates of the rotated image are valid on the data bus in 25.2 ns of modeling time after all calculations.

Using the FPGA from Stratix II family, such scheme of the rotation function of hardware realization takes 10% from all topological elements: 4264 ALUTs (Adapted View of the Table) and 2228 dedicated logical registers.

4. Image rotation using reflecting sine-cosine rotation matrix

For image rotation, an inverse sine-cosine rotation matrix can be used:

$$R^{-1}(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}. \quad (8)$$

In that case, coordinate equations for graphic image rotation relatively any point, using the inverse sine-cosine matrix, are:

$$\begin{align*}
x &= x_0 + (x'-x_0)\cos \alpha - (y_0 - y')\sin \alpha \\
y &= y_0 - (x'-x_0)\sin \alpha + (y'-y_0)\cos \alpha,
\end{align*} \quad (9)$$

where $x$ – graphic image coordinates on horizontal axis $x \in \{0,1,\ldots,799\}$; $y$ – graphic image coordinates on the vertical axis, $y \in \{0,1,\ldots,799\}$.

Practical realization of the rotation method based on using the inverse sine-cosine rotation matrix...
shows that hardware realization of the calculation method of the inverse matrix is the same as realization of the direct rotation matrix. Time characteristics are also equal to characteristics of calculation of the direct rotation matrix, which are represented in figure 3.

5. Conclusion
In the end of the study, three different ways of graphic image rotation are received in software MATLAB, in hardware Altera Quartus II and on the onboard aviation hardware prototype: using the direct rotation matrix (1), Owen-Makedon algorithm (4) and the inverse rotation matrix (8).

Main results of the study are:
- graphic image rotation by using the direct rotation matrix (1) is implemented with additional image distortions, which are shown as «lost» pixels, which is unacceptable in the display digital map in avionics;
- graphic image rotation using Owen-Makedon algorithm (4) needs more computing resources for its practical realization on FPGA implementation.
- graphic image rotation using the inverse rotation matrix (8) combines advantages of both previous ways.

If there is an image displacement, when the initial version never had it, the visual perception of such image has worsened significantly because there are some graphical primitives steps (line, sign and etc.). The most sensitive images are line rotations, which are programmed facially and with vectors at the small angles of about $\pi/12-\pi/10$. In avionics, there are some special smoothing algorithms to fix the images displacements, which are described in detail in [8-10].

Received results can be used for design graphic modules software and hardware, which are the part of the onboard graphic stations and digital indicators, which realize processing video, in particular, the graphic image rotation.

Modeling results are received on the computer tool with the following characteristics: ASUS K56CB-X0391H, Intel(R) Core(TM) i5-3337U, 4 cores, 1.8 GHz, RAM 6 Gb with Windows 8.1.

References
[1] Livada B 2012 Scientific Technical Review 62 70-79
[2] Verma P, Karar V, Niranjan V and Saini S S 2015 International Journal of Computer Applications 122 13-17
[3] Ananda C M, Venkatanarayana K G, Preme M and Raghu M 2011 Defence Science Journal 61 289-298
[4] Kihara K, Kawahara J, Takeda Y 2010 Behavior Research Methods 42 1105-13
[5] Nathaniel K, Saffin I 2014 2014 African Journal of Physics 27 1205-12
[6] Bordoloi H, Sarm K K 2009 World Academy of Science, Engineering and Technology 27 1205-12
[7] Laurent B, Ousman B 2007 International Journal of Physical Sciences 2 281-289
[8] Sutopo H 2011 International Journal of Computer Theory and Engineering 3 422-426
[9] Gaol F L 2013 Journal of Computers 8 69-78
[10] Dan S, Jie B, Zhibo Sh 2014 Journal of Multimedia 9 765-773