

MATRICES IN COMPUTER GRAPHICS

Nataša LONČARIĆ, Damira KEČEK, Marko KRALJIĆ

Abstract: In this article, the usage of matrices and matrix operations in computer graphics is shown. A brief overview of geometric transformations in computer graphics is given. The "Matrix - Computer Graphics" application software is created for the representation and easier understanding of relations between geometric transformations and matrix calculus.

Keywords: computer graphics; geometric transformations; matrices; "Matrix - Computer Graphics" application software

1 INTRODUCTION

Computer graphics is a computing field that involves the creating, storing and processing of image content via computer. It has a wide application in science, engineering, art and especially in the field of entertainment, movies and video games. [3, 2] Computer graphics began to progress in the 1960s when the first interactive creation of graphic content was enabled, and since then it has been developing incredibly quickly [6]. It is important to note that computer graphics applications are used for various educational purposes. For example, computer graphics applications can, in the education process, help people with learning difficulties [10], assist students in understanding computer science algorithms [4], as well as increasing students' interest in programming and computer graphics [1].

2 2D TRANSFORMATIONS

The fundamental part of computer graphics are geometric transformations that can change the position, size, orientation and shape of an object. Basic geometric transformations in a 2D plane are called 2D transformations and those are translation, rotation, scaling, reflection and shear. Furthermore, each of the well-known transformations is briefly explained in [11], [9].

Before considering the abovementioned transformations, it should be noted that for any point $T(x,y)$ in a 2D plane, a column vector or row vector representation can be used. In this article, column vector representation is used, i.e. point T is represented as $T = \begin{bmatrix} x \\ y \end{bmatrix}$.

2.1 Translation

Translation is a geometric transformation that moves the current point $T = \begin{bmatrix} x \\ y \end{bmatrix}$ to the point $T' = \begin{bmatrix} x' \\ y' \end{bmatrix}$ for a given vector $\vec{t} = t_x\vec{i} + t_y\vec{j} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$. Translation is shown in Fig. 1.

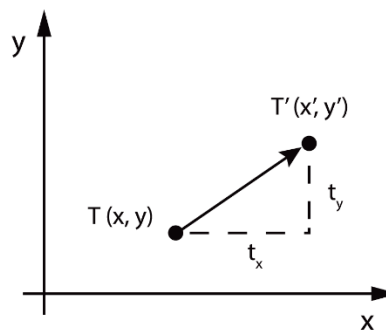


Figure 1 Translation

Therefore, point T' can be written as:

$$T' = T + \vec{t} \quad (1)$$

and in matrix notation as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (2)$$

2.2 Rotation

Let $O(0,0)$ and θ be a point and an angle given respectively. Rotation around the point $O(0,0)$ by the angle θ is a transformation that maps every point T in the plane to the point T' in such a manner that the following properties are true:

$$|OT| = |OT'| \text{ and } \angle TOT' = \theta$$

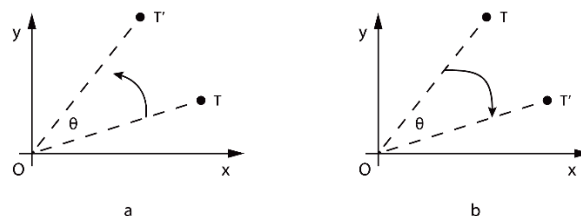


Figure 2 Rotation

By convention, rotation by the positive angle θ goes counter-clockwise (Fig. 2a), while rotation by the negative angle θ goes clockwise (Fig. 2b).

Rotation of the point $T(x, y)$ located at the angle ϕ from the horizontal x coordinate with a distance r from the origin of the coordinate system leads to a new point $T'(x', y')$. By using standard trigonometry, coordinates of the point $T(x, y)$ can be expressed as:

$$x = r \cos \phi \tag{3}$$

$$y = r \sin \phi \tag{4}$$

Similarly, coordinates of the point $T'(x', y')$ can be expressed as:

$$x' = r \cos(\phi + \theta), \tag{5}$$

$$y' = r \sin(\phi + \theta). \tag{6}$$

By using the equations (3) and (4), as well as addition formulas, the following formulas can be obtained:

$$x' = r \cos \phi \cos \theta - r \sin \phi \sin \theta = x \cos \theta - y \sin \theta, \tag{7}$$

$$y' = r \cos \phi \sin \theta + r \sin \phi \cos \theta = x \sin \theta + y \cos \theta, \tag{8}$$

which can be, in a matrix form, written as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}. \tag{9}$$

2.3 Scaling

Scaling is a mapping that expands or compresses each point coordinate. The scaling of each coordinate is determined by the scaling factor S . If the scaling factors for each coordinate are different, scaling is disproportionate. If the scaling factors are the same for all coordinates, scaling is proportional. For the point $T(x, y)$ and the scaling factor S_x and S_y that scales the point $T(x, y)$ in the x and y direction respectively, the following coordinates of the point T' are obtained:

$$x' = S_x \cdot x, \tag{10}$$

$$y' = S_y \cdot y. \tag{11}$$

In the matrix form, equations (10) and (11) can be written as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}. \tag{12}$$

For example, for the scaling factors $S_x = S_y = 2$, the scaling process of the square is shown in Fig. 3.

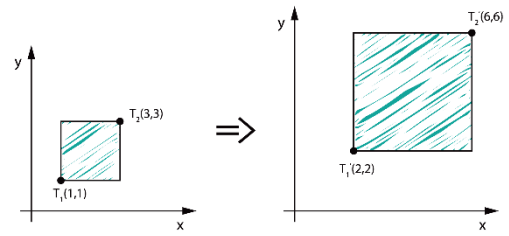


Figure 3 Proportional scaling of the square

2.4 Reflection

Reflection is a special case of a scaling where the size of the object does not change. For the scaling factors $S_x = -1$ and $S_y = 1$, reflecting is about the y axis. If the scaling factors are $S_x = 1$ and $S_y = -1$, there is a reflection about the x axis. For the scaling factors $S_x = S_y = -1$, reflection is about the x and y axes. In Fig. 4, reflections about the x and y axes of Super Mario, the main character in the Super Mario video games as an original object, are shown.

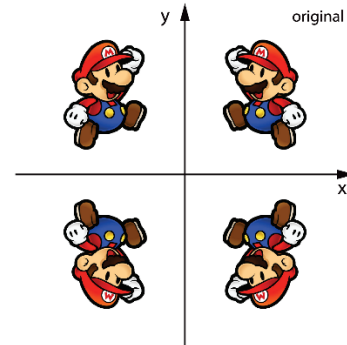


Figure 4 Reflection

2.5 Shear

Skewing the object with respect to the x or y axis is called shear transformation. Skewing by the angle α with respect to the x axis and by the angle β with respect to the y axis is illustrated in Fig. 5.

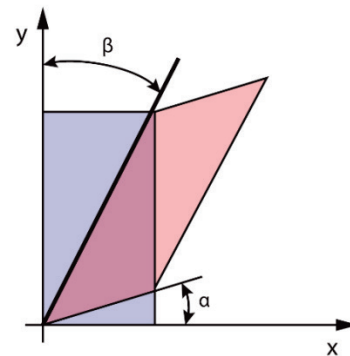


Figure 5 Shear

Coordinates of the point $T'(x', y')$ can be defined as:

$$x' = x + y \cdot \tan \beta, \tag{13}$$

$$y' = y + x \cdot \tan \alpha. \tag{14}$$

In the matrix form, point T' can be represented as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & \tan \beta \\ \tan \alpha & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{15}$$

2.6 Homogeneous Coordinate System

The process of implementing several successive transformations is, by using Cartesian coordinates, computationally more extensive. Namely, it can be noticed that to perform translation, matrix addition is carried out, while to perform other transformations, only matrix multiplication is carried out. In order to treat all transformations in the same manner, i.e. to reduce them to matrix multiplication, homogeneous coordinates are introduced. Thus, each Cartesian point $T(x,y)$ can be converted into homogeneous coordinates $T(x,y,w)$, and a coordinate system with this point representation is called a homogeneous coordinate system. Although triples of coordinates commonly display a point in space, here they represent a point in the plane or 2D point. Two sets of homogeneous coordinates, (x,y,w) and (x',y',w') , represent the same point if one is a multiple of the other. For example, coordinates $(1,2,3)$ and $(2,4,6)$ represent the same point. It can be concluded that one point in homogeneous coordinates can be displayed in infinitely many ways, that is, one whole class of triples of coordinates represents the same point in the plane. This class of triples of coordinates which represent one point in the plane (t_x, t_y, t_w) , where $t \neq 0$, actually represents a line in the 3D space. It should be noted that at least one homogeneous coordinate must be different from zero, meaning that the point $(0,0,0)$ is not allowed. If the coordinate $w \neq 0$, each of them can be divided by w , which is why the point T can be written as:

$$T(x, y, w) = T\left(\frac{x}{w}, \frac{y}{w}, 1\right) = (x_w, y_w, 1) \tag{16}$$

All such points form a plane with the equation $w = 1$ in the 3D space. In the case of $w = 0$, it is a point in infinity, which will not be considered in more detail.

The usage of homogeneous coordinates is of great importance in computer graphics. Homogeneous coordinates provide a uniform mathematical tool for all transformations, where matrix transformation through homogeneous coordinates must be in the 3×3 form [8].

In that sense, homogeneous translation matrix becomes:

$$M_{tr} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{17}$$

where translation is reduced to a matrix multiplication:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{18}$$

Furthermore, the homogeneous translation matrix is:

$$M_{tr} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{19}$$

homogeneous rotation matrix:

$$M_{ro} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{20}$$

homogeneous scaling matrix:

$$M_{sc} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$

and homogeneous shear matrix:

$$M_{sh} = \begin{bmatrix} 1 & \tan \beta & 0 \\ \tan \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{22}$$

In order to conduct several sequences of the transforms, all these transformations must be compound. After the implementation of all transformations, for example scaling, rotation and translation over the point T in this original order, and the coordinates of the point T' could be calculated in the following way:

$$T' = M_{tr} \cdot M_{ro} \cdot M_{sc} \cdot T, \tag{23}$$

i.e. they could be calculated by multiplying the homogeneous scaling matrix, homogeneous rotation matrix and homogeneous translation matrix, respectively, into a single matrix.

3 "MATRIX – COMPUTER GRAPHICS"

In high school or at the faculty level education, matrix calculus is an integral and significant part of the mathematics curriculum for students. One of the possible ways of matrix operation implementation is computer graphics.

In order to better illustrate the relations between geometric transformations and the matrix calculus described in the previous section, the "Matrix - Computer Graphics" application software that provides a simple interface for studying 2D graphics is created. This application software is written in C#, an object-oriented programming language. One can find more about C#, for example, in [5,7]. The "Matrix – Computer Graphics" application is available at: http://www.mediafire.com/file/pgv2jkrpr0ggq47/Matrix_-_Computer_Graphics.exe or

http://www.mediafire.com/file/84b4dhwonqcb78c/Matrix_-_Computer_Graphics.zip

Using the "Matrix - Computer Graphics" application you can easily apply one of the abovementioned transformations on the desired object. Students can easily draw the desired object in the coordinate system and optionally, by using the determined matrices, apply the corresponding geometric transformations. One of the possibilities is to optionally enter data in the upper matrices for the corresponding transformation (translation, rotation, scaling or shear) of the drawn object, or to directly insert only the rotation angle or shear angles of the drawn object with respect to the x or y axis in the two lower matrices as shown in Fig. 6.

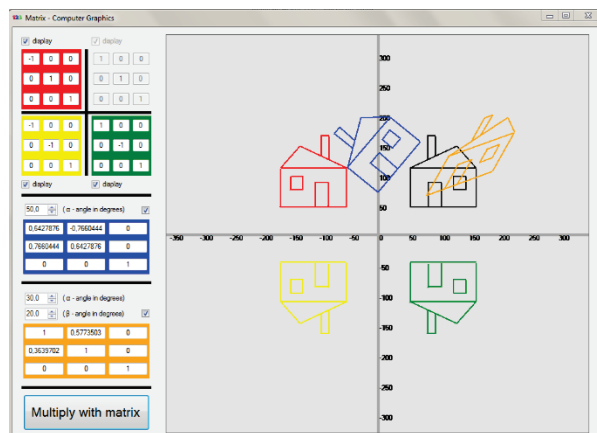


Figure 6 "Matrix - Computer Graphics" example

4 CONCLUSION

Computer graphics is an unavoidable part of everyday life. As it is widely known, computer graphics is present in computer games, films, television commercials, as well as in various branches of the industry. In this paper, authors tried to get closer to the students who are dealing with computer science, especially computer graphics, the mathematical background of computer graphics in which the term matrix is in the center. Although computer graphics is nowadays usually 3D, in this paper the authors looked only at 2D transformations in order to show, in a simpler way, the relationship between the matrix calculus and geometric transformations used in computer graphics. For this purpose, the "Matrix - Computer Graphics" application gives a contribution in the visualization and easier learning of relationships between the matrices and their applications on display and positioning, as well as on transforming different objects on a computer.

5 REFERENCES

- [1] Bruno, J., Jan, V., & Slaby, A. (2009). Computer Graphics in Computer Science Education. *Problems of Education in the 21st Century*, 11, 60-68.
- [2] Hearn, D. D., Baker, M. P., & Carithers, W. (2014). *Computer Graphics with Open GL*, 4th Edition. Pearson Education Limited.
- [3] Hughes, J. F., van Dam, A., McGuire, M., Sklar, D. F., & Foley, J. D. (2014). *Computer Graphics: Principles and Practice*, 3rd Edition. Addison-Wesley.
- [4] Naps, T. L. (2005). JHAVÉ: Supporting Algorithm Visualization. *IEEE Computer Graphics and Applications*, 25(5), 49-55. <https://doi.org/10.1109/MCG.2005.110>
- [5] Petricek, T. & Skeet, J. (2010). *Real-World Functional Programming: With Examples in F# and C#*, 1st Edition. Manning Publications.
- [6] Salomon, D. (2011). *The computer graphics manual*, Volume 2. Springer-Verlag London Limited. <https://doi.org/10.1007/978-0-85729-886-7>
- [7] Sharp, J. (2013). *Microsoft Visual C# 2013 Step by Step (Step by Step Developer)*, 1st Edition. Microsoft Press.
- [8] Skala, V. (2006). Length, Area and Volume Computation in Homogeneous Coordinates. *International Journal of Image and Graphics*, 06(04), 625-639. <https://doi.org/10.1142/S0219467806002422>
- [9] Srivastav, M. K. (2016). Transformation of an Object in Computer Graphics: A Case Study of Mathematical Matrix Theory. *Elixir Comp. Engg.* 100, 43396-43399.
- [10] Vera, L., Campos, R., Herrera, G., & Romero, C. (2007). Computer graphics applications in the education process of people with learning difficulties. *Computers & Graphics*, 31, 649-658. <https://doi.org/10.1016/j.cag.2007.03.003>
- [11] Vince, J. (2010). *Mathematics for Computer Graphics*, 3rd Edition. Springer-Verlag London Limited. <https://doi.org/10.1007/978-1-84996-023-6>

Authors' contacts:

Nataša LONČARIĆ, Lecturer
Tehnička škola Čakovec
Športska 5, 40000 Čakovec, Croatia
e-mail: naloncaric@unin.hr

Damira KEČEK, Senior Lecturer
University North, University Center Varaždin
104. brigade 3, 42000 Varaždin, Croatia
e-mail: dkecek@unin.hr

Marko KRALJIĆ, Student
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia