

Research Article

A Theoretical Scientific Programming Framework for Application of Linear Matrix Transformation in Plane Computer Graphics

Mingcan Chen 

College of Management, Henan Institute of Economics and Trade, Zhengzhou 450008, Henan, China

Correspondence should be addressed to Mingcan Chen; chenmingcan@henetc.edu.cn

Received 10 April 2022; Revised 16 May 2022; Accepted 8 June 2022; Published 21 July 2022

Academic Editor: Ahmed Farouk

Copyright © 2022 Mingcan Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Plane computer graphics are basic information carriers in many industrial scenarios, such as engineering simulation, automatic control, and software design. Plane computer graphics are generally a kind of digital signals guided by mathematical symbols, and each vertex of a plane computer graph forms a graph matrix. Therefore, linear matrix transformation serves as the most common algorithmic unit to realize various information processing operations. To improve ease of graph matrix computing in practical engineering scenarios, this paper proposes a theoretical scientific programming framework for application of linear matrix transformation in plane computer graphics. Firstly, theoretical basis of linear matrix transformation in homogeneous plane coordinates is displayed and analyzed. Then, the universal theorem about linear transformation of graph matrices is deduced, and corresponding proofs are also given. Finally, a case study is set up to demonstrate the main workflow of the proposed theoretical scientific programming framework. The simulative results reveal feasibility of the proposal.

1. Introduction

Geometric graphics transformation refers to some specific graphics process operations [1, 2], including movement of the defined graphics from one position to another [3], change of size and shape for graphics, or generation of complex graphics from existing graphics [4, 5]. And it is also named as graphics transformation for short [6, 7]. Graphics transformation can help deduce required graphics with the aid of two-dimensional or three-dimensional transformation operations [8, 9]. Figure 1 gives a typical example of 2D linear matrix transformation in plane space [10, 11]. Graphic transformation is usually realized via matrix transformation, and matrices used for transformation varies with different scenes [12, 13]. With the development of computer science and simulation modeling, graphics transformation is usually implemented by computer algorithms [14], forming the discipline of computer graphics [15]. In order to enhance engineering ease of digital signal processing in plane graphics [16], this work concentrates the theoretical scientific programming

framework for application of linear matrix transformation in plane computer graphics [17, 18].

Points are the basic elements of a graph [19]. In analytic geometry, points are represented by vectors [20, 21]. For example, points in two-dimensional space (plane) and in three-dimensional space are separately represented by

$$\begin{aligned} P_{2d} &\in (x, y)', \\ P_{3d} &\in (x, y, z)'. \end{aligned} \quad (1)$$

A planar (two-dimensional) figure or a spatial (three-dimensional) shape can be represented by a set of points (referred to as a set of points) [22, 23]. A planar graph can be represented by a matrix as follows:

$$\text{Mat}_{2d} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix}, \quad (2)$$

where the set of vertices $(x_1, y_1)', (x_2, y_2)', \dots, (x_n, y_n)'$ are included and a three-dimensional graph can be represented by a matrix as follows:

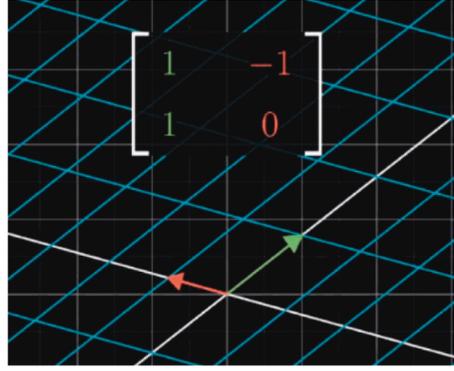


FIGURE 1: A typical example of linear matrix transformation in plane space.

$$\text{Mat}_{3,d} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & \cdots & z_n \end{bmatrix}. \quad (3)$$

When the point set is represented by a matrix, the transformation of the point can be realized by the following linear transformation:

$$\mathbf{Ax} = \mathbf{y}, \quad (4)$$

where \mathbf{x} denotes the old point coordinates, \mathbf{A} denotes the transformation matrix, and \mathbf{y} denotes the new point coordinates. There is a one-to-one correspondence between linear transformation and matrix.

2. Plane Graph Transformation in Homogeneous Coordinates

First of all, it is supposed to illustrate four basic transformation types which are shown in Figure 2: translation, scale, rotation, and skew [24, 25]. The commonly used transformation matrix of plane graphics contains four types: proportional transformation, symmetrical transformation, rotation transformation, and the translation transformation [26, 27].

Definition 1. The proportional transformation is represented as the following format:

$$T = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}, \quad (5)$$

where s_x and s_y are the proportional coefficient along the x -axis and y -axis, respectively, and they satisfy the condition: $s_x > 0, s_y > 0$. The different value setting of s_x and s_y leads to different scenes.

- (i) When $s_x = s_y = 1$, it is a constant proportional transformation, that is, the figure is unchanged.
- (ii) When $s_x = s_y > 1$, the figure magnifies proportionally along the two axes.
- (iii) When $s_x = s_y < 1$, the figure shrinks proportionally along the two axes.
- (iv) When $s_x \neq s_y$, the figure changes nonuniformly along the two axes, and the result is graphic distortion.

Definition 2. The symmetrical transformation is represented as the following format:

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad (6)$$

where $a, b, c,$ and d are the symmetrical coefficients. The different value setting of $a, b, c,$ and d leads to different scenes.

- (i) When $a = 1, d = -1, b = c = 0$, it is a symmetrical transformation about the X -axis.
- (ii) When $a = -1, d = 1, b = c = 0$, it is the symmetric transformation about the Y -axis.
- (iii) When $a = -1, d = -1, b = c = 0$, it is the symmetric transformation about the origin.
- (iv) When $a = d = 0, b = c = 1$, it is the symmetric transformation about the straight line $x - y = 0$.
- (v) When $a = d = 0, b = c = -1$, it is the symmetric transformation about the straight line $x + y = 0$.

Definition 3. The rotation transformation is represented as the following format:

$$T = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (7)$$

where θ is the angle rotated around the origin of the coordinate. The different value setting of θ leads to different scenes.

- (i) The value of this angle is positive ($\theta > 0$) when it rotates counterclockwise.
- (ii) The value of this angle is negative ($\theta < 0$) when it rotates clockwise.

Definition 4. The translation transformation is represented as the following format:

$$T = \begin{cases} x' = x + m, \\ y' = y + n, \end{cases} \quad (8)$$

or the following format:



FIGURE 2: Four basic transformation types: translation, scale, rotation, and skew.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m \\ n \end{bmatrix}. \quad (9)$$

As the translation transformation cannot be written in form of $\mathbf{Ax} = \mathbf{y}$, it is not with the linear type. In order to change translation transformation into linear transformation, we can uniformly use linear matrix operations to study plane graphic transformation [28]. The homogeneous coordinates are generated, in which n -dimensional vectors are represented by $(n + 1)$ -dimensional ones. When the homogeneous coordinate of the vector $(x_1, x_2, \dots, x_n)'$ equals to $(hx_1, hx_2, \dots, hx_n, h)'$, where h is a real number, the homogeneous coordinate representation of a vector is not unique, and different values of the homogeneous coordinate h represents the same point. A homogeneous coordinate $(8, 4, 2)'$, $(4, 2, 1)'$ represents the point $(4, 2)'$ on the plane. Hence, the following definition can be deduced.

Definition 5. When $h = 1$, the homogeneous coordinate $(x_1, x_2, \dots, x_n, h)'$ is called as the normalized homogeneous coordinate.

After the introduction of homogeneous coordinates, translation transformation defined in Definition 4 can be written as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & m \\ 0 & 1 & n \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + m \\ y + n \\ 1 \end{bmatrix}. \quad (10)$$

It is a linear transformation, and the different value setting of a, b, c , and d leads to different scenes.

- (i) When $m > 0$, the graph translates positively along the X -axis.
- (ii) When $m < 0$, the graph translates negatively along the X -axis.
- (iii) When $n > 0$, the graph translates positively along the Y -axis.
- (iv) When $n < 0$, the graph translates negatively along the Y -axis.

3. Theorem and Proof for Plane Graphics Transformation

Theorem 1. When the transformation matrix of the plane graph is denoted as follows:

$$T = \begin{bmatrix} a & b & m \\ c & d & n \\ 0 & 0 & s \end{bmatrix}, \quad (11)$$

the $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ causes proportional, symmetric, and rotational transformation of the planar graph, and $\begin{bmatrix} m \\ n \end{bmatrix}$ causes the plane graph to produce translation transformation.

Proof. Supposing that $y = Tx$, the following formula can be deduced:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & m \\ c & d & n \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (12)$$

where y, x , and T are separately represented as follows:

$$\begin{aligned} y &= \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} Y \\ 1 \end{bmatrix}, \\ x &= \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ 1 \end{bmatrix}, \\ T &= \begin{bmatrix} a & b & m \\ c & d & n \\ 0 & 0 & s \end{bmatrix} = \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix}. \end{aligned} \quad (13)$$

According to the matrix block method, the following six formulas can be further deduced as follows:

$$\begin{aligned} Y &= \begin{bmatrix} x' \\ y' \end{bmatrix}, \\ X &= \begin{bmatrix} x \\ y \end{bmatrix}, \\ T_1 &= \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \\ T_2 &= \begin{bmatrix} m \\ n \end{bmatrix}, \\ T_3 &= (0, 0), \\ T_4 &= s. \end{aligned} \quad (14)$$

Then, the $y = Tx$ can be rewritten as follows:

$$\begin{aligned} \begin{bmatrix} Y \\ 1 \end{bmatrix} &= T \begin{bmatrix} X \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} T_1X + T_2 \\ T_3X + T_4 \end{bmatrix} \\ &= \begin{bmatrix} T_1X + T_2 \\ T_4 \end{bmatrix}. \end{aligned} \quad (15)$$

Obviously, T_1X is the basic transformation of two-dimensional graphics, thus translation transformation $T_1X + T_2$ can be deduced as follows:

$$T_1X + T_2 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m \\ n \end{bmatrix}. \quad (16)$$

To sum up, the transformation of plane graphics can be represented by a unified matrix.

$$T = \begin{bmatrix} a & b & m \\ c & d & n \\ 0 & 0 & s \end{bmatrix}. \quad (17)$$

The proof for Theorem 1 is to be finished so far.

Points are often used in computer graphics, and if the position of point changes, so does the figure [29]. Therefore, if transform the graph, it is needed to change the point [30]. The effect of the product of a matrix and a vector in linear transformation provides a method of graphic transformation, which can be used to produce the effect of the graphic change in the computer [31, 32].

For a closed figure in a plane coordinate system, it is represented by a matrix x , and each column of x represents the coordinates of a vertex of the graph. To close the graph, the last column of x is the same as the first column [33]. In order to realize the translation of graphics by using linear transformation, a row with elements of 1 is added to matrix x so that the shape of matrix $[3 \times n]$.

The following two transformation matrices are given as

$$\begin{aligned} M &= \begin{bmatrix} 1 & 0 & m \\ 0 & 1 & n \\ 0 & 0 & 1 \end{bmatrix}, \\ R &= \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned} \quad (18)$$

for two transformation processes $Y_1 = MX$ and $Y_2 = RX$, they can be separately explained as follows:

- (i) Y_1 is the transformation result of X , in which m is translated along the positive direction of the X -axis,

n is translated along the positive direction of the Y -axis.

- (ii) Y_2 is the result of the inverse (clockwise) rotation of the θ angle of the graph X with the coordinate origin as the center.

The letter A is moved 15 up, 30 to the left, the $\pi/3$ counterclockwise, and then 30 up, 20 to the right, then turn counterclockwise by $3/4\pi$, and the MATLAB software is used to draw transformed graphics [34, 35]. The graphic matrix of the letter A can be constructed from Table 1. For this example, X is deduced as follows:

$$X = \begin{bmatrix} 0 & 4 & 6 & 10 & 8 & 5 & 3.5 & 6.1 & 6.5 & 3.2 & 2 & 0 \\ 0 & 14 & 14 & 0 & 0 & 11 & 6 & 6 & 4.5 & 4.5 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (19)$$

From the meaning of the question, it can be seen that the transformation matrix of figure a moving up 15 and moving left 30, which can be represented as the following matrix:

$$M_1 = \begin{bmatrix} 1 & 0 & -30 \\ 0 & 1 & 15 \\ 0 & 0 & 1 \end{bmatrix}. \quad (20)$$

And the transformation matrix can be represented as

$$R_1 = \begin{bmatrix} \cos \frac{\pi}{3} & -\sin \frac{\pi}{3} & 0 \\ \sin \frac{\pi}{3} & \cos \frac{\pi}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (21)$$

where rotating counterclockwise of graph A is $\pi/3$. The transformation matrices of moving figure A up 30, 20 to the right, and $3/4\pi$ counterclockwise, respectively, are as follows:

$$\begin{aligned} M_2 &= \begin{bmatrix} 1 & 0 & 20 \\ 0 & 1 & 30 \\ 0 & 0 & 1 \end{bmatrix}, \\ R_2 &= \begin{bmatrix} \cos \frac{3\pi}{4} & -\sin \frac{3\pi}{4} & 0 \\ \sin \frac{3\pi}{4} & \cos \frac{3\pi}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (22)$$

So, these three transformation matrices are respectively R_1 , M_1 , and R_2M_2 , and one of them R_2M_2 is a combinatorial transformation. \square

TABLE 1: Data sheet with letter A.

x	0	4	6	10	8	5	3.5	6.1	6.5	3.2	2	0
y	0	14	14	0	0	11	6	6	4.5	4.5	0	0

```

INPUT:  $X, M_1, M_2, R_1, R_2$ 
OUTPUT:  $Y_1, Y_2$ 
(1) while Each graph do
(2)   for Dimension: 1  $\rightarrow$  3 do
(3)     Plot plane graph of letter A
(4)     Define transformation matrix  $M_1$  for letter A
(5)     Calculate transformation result  $Y_1$  for letter A
(6)     Define transformation matrix  $R_1$  for letter A
(7)     Calculate transformation result  $Y_2$  for letter A
(8)     Define transformation matrices  $R_2$  and  $M_2$  for letter A
(9)     Calculate transformation result  $Y_3$  for letter A
(10)    Output graphics for letter A
(11)  end for
(12) end while

```

ALGORITHM 1: Workflow of the proposed theoretical programming framework.

4. Scientific Programming Demo for the Case Study

Having analyzed basic linear matrix transformation theory and proved its efficiency, this section manages to give a practical scientific programming demo for the above case study. Its pseudo codes reveal major workflow of the proposed theoretical scientific programming framework, which is illustrated in Algorithm 1. Through such suggested theoretical scientific programming framework, it is expected to facilitate engineering practice in relevant areas. The proposed scientific programming demo for the above case study in MATLAB is given as follows:

```

>>close all;
>>X = [0,4,6,10,8,5,3.5,6.1,6.5,3.2,2,0;
0,14,14,0,0,11,6,6,4.5,4.5,0,0;
ones(1,12)];
>>plot(X(1,:),X(2,:));
% Draw the letter A figure hold on;
>>M1 = [1,0,-30; 0,1,15; 0,0,1];
>>Y1 = M1 * X;
hold on;
>>fill(Y1(1,:),Y1(2,:), "red");
%Draw the figure after the translation of the letter A.
>>R1 = [cos(pi/3), -sin(pi/3),0;
sin(pi/3),cos(pi/3),0;
0,0,1];
>>Y2 = R1 * X;
hold on;
>>fill(Y2(1,:),Y2(2,:), "blue");

```

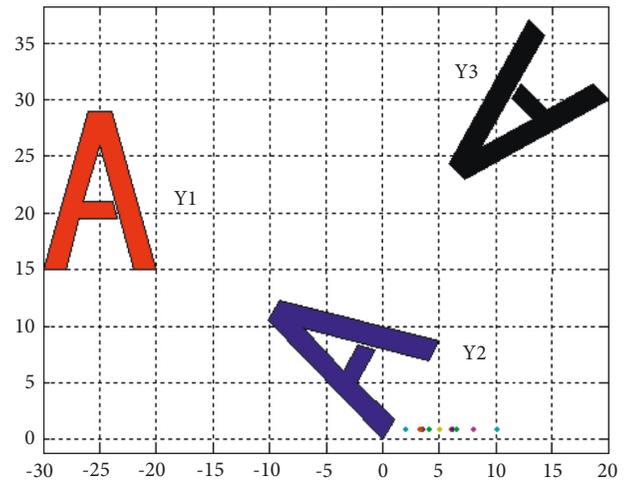


FIGURE 3: Transformation diagram of letter A.

```

%Draw the figure after the letter A is rotated.

```

```

>>M2 = [1,0,20; 0,1,30; 0,0,1];
>>R2 = [cos(3 * pi/4), -sin(3 * pi/4),0;
sin(3 * pi/4),cos(3 * pi/4),0;
0,0,1];
>>Y3 = M2 * R2 * X;
hold on;
>>fill(Y3(1,:),Y3(2,:), "black");
%Draw the figure after the letter A is translated and rotated.

```

Running result of the above programming code leads to the graph in Figure 3.

5. Conclusion

Plane graph transformation mainly includes proportion, symmetry, rotation, translation, and so on [36]. These transformations are unified into linear transformations under homogeneous coordinates and studied by the matrix [37]. This provides a unified transformation matrix method for coordinate transformation of two-dimensional, three-dimensional, and even high-dimensional space of geometric graphics, and these transformations can be easily combined [38]. The complexity of the transformation matrix is related to transformation steps, that is, the more steps, the more transformation matrix, which can produce the transformation of the complex graph [39, 40].

Data Availability

The data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. Lee and C. Tseng, "Design of graph filter based on transformation matrix and spectral response," in *Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 16–19, IEEE, Hualien City, Taiwan, November 2021.
- [2] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: a survey," *Computer Graphics Forum*, vol. 37, no. 6, pp. 35–58, 2018.
- [3] G. Teh, W. C. Teh, "Elementary matrix equivalence and core transformation graphs for parikh matrices," *Discrete Applied Mathematics*, vol. 251, pp. 276–289, 2018.
- [4] L. Zhen, Y. Zhang, K. Yu, N. Kumar, A. Barnawi, and Y. Xie, "Early collision detection for massive random access in satellite-based internet of things," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5184–5189, 2021.
- [5] Z. Guo, L. Tang, T. Guo, K. Yu, M. Alazab, and A. Shalaginov, "Deep graph neural network-based spammer detection under the perspective of heterogeneous cyberspace," *Future Generation Computer Systems*, vol. 117, pp. 205–218, 2021.
- [6] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5gb remote e-health," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 54–61, 2021.
- [7] H. Y. Tian, B. Tian, Y. Sun, and C. R. Zhang, "Three-component coupled nonlinear Schrödinger system in a multimode optical fiber: darbox transformation induced via a rank-two projection matrix," *Communications in Nonlinear Science and Numerical Simulation*, vol. 107, Article ID 106097, 2022.
- [8] Z. Guo, K. Yu, A. Jolfaei, A. K. Bashir, A. O. Almagrabi, and N. Kumar, "A Fuzzy Detection System for Rumors through Explainable Adaptive Learning," *IEEE Trans. Fuzzy Syst*, vol. 29, 2021.
- [9] P. Liang, H. T. Likassa, C. Zhang, and J. Guo, "New robust PCA for outliers and heavy sparse noises' detection via affine transformation, the $L \{*, w\}$ and $L 2, 1$ norms, and spatial weight matrix in high-dimensional images: from the perspective of signal processing," *International Journal of Mathematics and Mathematical Sciences*, vol. 2021, Article ID 3047712, 9 pages, 2021.
- [10] Z. Guo, K. Yu, Y. Li, G. Srivastava, and J. C. W. Lin, "Deep learning embedded social internet of things for ambiguity-aware social recommendations," *IEEE Trans. Netw. Sci. Eng.* vol. 9, no. 3, pp. 1067–1081, 2021.
- [11] S. Vologiannidis, E. N. Antoniou, N. P. Karampetakis, and A. I. G. Vardoulakis, "Polynomial matrix equivalences: system transformations and structural invariants," *IMA Journal of Mathematical Control and Information*, vol. 38, no. 1, pp. 54–73, 2021.
- [12] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C. wei Lin, and T. Sato, "Secure Artificial Intelligence of Things for Implicit Group Recommendations," *IEEE Internet Things J*, vol. 9, no. 4, 2021.
- [13] R. Geng and X. Geng, "A matrix yajima-oikawa long-wave-short-wave resonance equation, darbox transformations and rogue wave solutions," *Communications in Nonlinear Science and Numerical Simulation*, vol. 90, Article ID 105408, 2020.
- [14] S. C. Chu, Y. Chen, F. Meng, C. Yang, J. S. Pan, and Z. Meng, "Internal search of the evolution matrix in quasi-affine transformation evolution (QUATRE) algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 5, pp. 5673–5684, 2020.
- [15] L. M. Anguas, M. I. Bueno, and F. M. Dopico, "Conditioning and backward errors of eigenvalues of homogeneous matrix polynomials under Möbius transformations," *Mathematics of Computation*, vol. 89, no. 322, pp. 767–805, 2019.
- [16] G. Horváth, I. Horváth, S. A. D. Almousa, and M. Telek, "Numerical inverse laplace transformation using concentrated matrix exponential distributions," *Performance Evaluation*, vol. 137, Article ID 102067, 2020.
- [17] K. Usevich, J. Li, and P. Comon, "Approximate matrix and tensor diagonalization by unitary transformations: convergence of Jacobi-type algorithms," *SIAM Journal on Optimization*, vol. 30, no. 4, pp. 2998–3028, 2020.
- [18] B. Zhang, N. Uysal, D. Fan, and R. Ewetz, "Handling stuck-at-fault defects using matrix transformation for robust inference of dnns," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2448–2460, 2020.
- [19] R. Wang and Y. Wang, "Forward looking infrared target matching algorithm based on depth learning and matrix double transformation," *Cluster Computing*, vol. 22, no. S3, pp. 7055–7062, 2019.
- [20] A. Kahla and H. B. Kahla, "An upper J-Hessenberg reduction of a matrix through symplectic householder transformations," *Computers & Mathematics with Applications*, vol. 78, no. 1, pp. 178–190, 2019.
- [21] V. A. Sanzharov and V. V. Sanzharov, "Modern problems of software integration in computer graphics applications and ways to solve them," *Programming and Computer Software*, vol. 44, no. 4, pp. 233–239, 2018.
- [22] S. Le Borne, "Factorization, symmetrization, and truncated transformation of radial basis function-ga stabilized Gaussian radial basis functions," *SIAM Journal on Matrix Analysis and Applications*, vol. 40, no. 2, pp. 517–541, 2019.
- [23] P. C. Lopes, R. L. Rangel and L. F. Martha, "An interactive user interface for a structural analysis software using computer graphics techniques in MATLAB," *Computer Applications in Engineering Education*, vol. 29, no. 6, pp. 1505–1525, 2021.
- [24] W. E. Carlson, R. Hackathorn, and R. E. Parent, "Computer graphics and animation at the Ohio state university," *IEEE*

- Computer Graphics and Applications*, vol. 41, no. 3, pp. 8–17, 2021.
- [25] B. S. Santos, G. Domik, and E. F. Anderson, “Foreword to the special section on computer graphics education in the time of covid,” *Graphics and Visual Computing*, vol. 4, Article ID 200028, 2021.
- [26] E. Y. Denisov, A. G. Voloboy, E. D. Biryukov, M. S. Kopylov, and I. A. Kalugina, “Automated software testing technologies for realistic computer graphics,” *Programming and Computer Software*, vol. 47, no. 1, pp. 76–87, 2021.
- [27] T. Feng, F. Fan, and T. Bednarz, “A review of computer graphics approaches to urban modeling from a machine learning perspective,” *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 7, pp. 915–925, 2021.
- [28] C. Andujar, C. R. Vijulie, À. Vinacua, B. S. Santos, and G. Alford, “Syntactic and semantic analysis for extended feedback on computer graphics assignments,” *IEEE Computer Graphics and Applications*, vol. 40, no. 3, pp. 105–111, 2020.
- [29] M. Li and Y. Li, “The application of computer graphics processing in visual communication design,” *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 4, pp. 5183–5191, 2020.
- [30] N. Christoff, “A survey of human modelling in computer graphics,” *Journal Multim. Process. Technology*, vol. 11, no. 4, pp. 130–136, 2020.
- [31] N. Bonneel, D. Coeurjolly, J. Digne, and N. Mellado, “Code replicability in computer graphics,” *ACM Transactions on Graphics*, vol. 39, no. 4, p. 93, 2020.
- [32] M. Long, F. Peng, and Y. Zhu, “Identifying natural images and computer generated graphics based on binary similarity measures of PRNU,” *Multimedia Tools and Applications*, vol. 78, no. 1, pp. 489–506, 2019.
- [33] V. Cantoni, P. Dondi, L. Lombardi, and A. Setti, “Teaching computer graphics through a digital humanities project,” *IEEE Computer Graphics and Applications*, vol. 39, no. 2, pp. 89–94, 2019.
- [34] M. Hoppe and H. Hoppe, “An adaptive multi-grid solver for applications in computer graphics,” *Computer Graphics Forum*, vol. 38, no. 1, pp. 138–150, 2019.
- [35] L. L. Khoroshko, P. A. Ukhov, and P. P. Keyno, “Development of massive open online courses based on 3d computer graphics and multimedia,” *International Journal of Engineering Pedagogy (ijEP)*, vol. 9, no. 4, p. 4, 2019.
- [36] M. Huang, M. Xu, T. Qiao, T. Wu, and N. Zheng, “Designing statistical model-based discriminator for identifying computer-generated graphics from natural images,” *Journal of Universal Computer Science*, vol. 25, no. 9, pp. 1151–1160, 2019.
- [37] R. F. Sproull, “Publishing a computer graphics book with prototype desktop publishing tools,” *IEEE Annals of the History of Computing*, vol. 40, no. 4, pp. 69–76, 2018.
- [38] A. Agrawal, “Application of machine learning to computer graphics,” *IEEE Computer Graphics and Applications*, vol. 38, no. 4, pp. 93–96, 2018.
- [39] D. Meng, Y. Xiao, Z. Guo et al., “A data-driven intelligent planning model for uavs routing networks in mobile internet of things,” *Computer Communications*, vol. 179, pp. 231–241, 2021.
- [40] Q. Zhang, K. Yu, Z. Guo et al., “Graph neural networks-driven traffic forecasting for connected internet of vehicles,” *IEEE Transactions on Network Science and Engineering*, 2021.