Method of combined vector normalization of 3-D objects

Oleksandr Romanyuk*[a], Yevhen Zavalniuk[a], Oleksii Bobko[a], Ihor Arsenyuk[a], Oksana Romanyuk[a] Natalia Sachaniuk-Kavets'ka[a], Larysa Nykyforova[b], Andrzej Kotyra[c], Ainur Kozbakova[d]; Aliya Kalizhanova[d,e]

[a]Vinnytsia National Technical University, 95 Khmelnytske shose, Vinnytsia, Ukraine, 21021; [b]National University of Life and Environmental Sciences of Ukraine, 15 Heroiv Oborony St., Kyiv, Ukraine, 03041; [c]Lublin University of Technology, 38a Nadbystrzycka St., Lublin, Poland 20-618; [d]Institute Information and Computational Technologies CS MES RK, 29 Kurmangazy St., Almaty, Kazakhstan 050000; [e]Almaty University of Power Engineering and Telecommunications, 126/1 Baitursynov St., Almaty, Kazakhstan, Almaty 050013

ABSTRACT

The paper reveals a method of highly productive determination of normalized vectors for the surfaces of three-dimensional objects. The method is based on the approximated calculation of even vectors of the rasterization line by adding the odd neighboring unit vectors. For the determination of further need in the normalization of obtained vectors, the computation of special threshold metrics is proposed. For the accelerated calculation of the threshold metrics, the developed expressions are given. In case of normalization of even vectors, it is recommended to use the developed polynomial approximate expressions. The plots of relative errors between obtained simplified and reference expressions are given. The possibility of increasing the productivity of the method by calculating the shared vector coordinate increments for each rasterization line is analyzed. The experimental research of productivity gain from the new method usage is carried out. During the study, the six variants of possible method usage are considered. The results of the research analysis are given in the table. The new method is designed for usage in highly effective visualization systems.

Keywords: vector normalization, surface shading, linear interpolation, inverse square root, midpoint vector

1. INTRODUCTION

Current stage of computer graphics is characterized with paying a special attention to the creation of highly realistic graphic scenes. At the same time, the most computationally expensive is the rendering stage, where pixel coordinates and color intensity are determined for each point. Normal vector $\vec{N}$ [14], incident light vector $\vec{L}$ [14], vector to the viewer $\vec{V}$ [14] are used to calculate the color intensity. These vectors should have a unit length, because they are used for the dot product calculation. As the normalization of vectors is a computationally expensive procedure that requires the execution of division and square root value calculation, the task of decreasing the computational complexity of vector normalization is actual.

2. THE METHOD

The purpose of the study is to increase the productivity of the three-dimensional figure's surface shading through the development of the simplified vector normalization method. Approximate expressions for the threshold criteria calculation, as well as the approximate expressions for a midpoint vector normalization, were obtained, using the polynomial regression expressions. Data processing for polynomial regression was implemented in C#. The average triangle that contains 100 points was considered for performance gain from the proposed method applying calculation.

Linear interpolation (LERP)[6,7,10] is used for the interpolation of non-unit vectors through the rasterization line between normalized vectors $\overrightarrow{N_A}$ and $\overrightarrow{N_B}$;

*email: rom8591@gmail.com

Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments 2024, edited by Ryszard S. Romaniuk, Andrzej Smolarz, Waldemar Wójcik, Proc. of SPIE Vol. 13400, 134000O · © 2024 SPIE · 0277-786X · doi: 10.1117/12.3054900

$$\overrightarrow{N}=(1-t)\cdot\overrightarrow{N_A}+t\cdot\overrightarrow{N_B},$$

where $t\in[0,1]$.

The method of accumulative addition of increments[13] between $\overrightarrow{N_A}$ and $\overrightarrow{N_B}$ analogously provides the interpolation of non-unit vectors:

$$\overrightarrow{N}=\overrightarrow{N_A}+I\cdot\Delta\overrightarrow{N_{AB}},$$

where I - number of point in the rasterization line, $\Delta\overrightarrow{N_{AB}}=(\overrightarrow{N_B}-\overrightarrow{N_A})/m$, m - number of sectors between $\overrightarrow{N_A}$ and $\overrightarrow{N_B}$. Next, the normalization of the interpolated vector $\overrightarrow{N}$ at the point of rasterization line is done according to the following formula[3,5,11]

$$\frac{\overrightarrow{N}}{\left|\overrightarrow{N}\right|}=\frac{1}{\sqrt{N_x^{\,2}+N_y^{\,2}+N_z^{\,2}}}\cdot(N_x;\,N_y;\,N_z),$$

where $\left|\overrightarrow{N}\right|$ - the length of vector $\overrightarrow{N}$, N_x, N_y, N_z - normalized vector coordinates.

The disadvantage is the need for inverse square root calculation for vector normalization at every surface point. It essentially affects the general productivity of three-dimensional scene visualization. It was proposed by Analog Devices to use the fifth-degree polynomial for square root approximate calculation[9]

$$0.1121216z^5-0.536499z^4+1.106812z^3-1.34491z^2+1.454895z+0.2075806,$$

where $z=N_x^{\,2}+N_y^{\,2}+N_z^{\,2}$.

The formula allows to simplify the process of vector normalization with a high accuracy, but the necessity of the inversed expression determination is still present. Additionally, the lower-degree polynomials provide a sufficiently accurate approximation of square root and are more appropriate.

R. Lyon from Apple has developed the quadratic expression[8] for the inverse square root $1/\sqrt{z}$ approximation

$$1+\frac{(1-z)+(1-z)^2}{2}.$$

However, the accuracy level of normalized vector calculation is not sufficiently high. Spherical linear interpolation (SLERP)[17,18,19] provides the determination of unit vectors without the necessity of their normalization. The formula is used

$$\frac{\overrightarrow{N_A}\cdot\sin((1-t)x)}{\sin(x)}+\frac{\overrightarrow{N_B}\cdot\sin(tx)}{\sin(x)},$$

where x - angle between $\overrightarrow{N_A}$ and $\overrightarrow{N_B}$.

The disadvantage of SLERP is a high computational complexity of trigonometric functions calculation. T. Barrera et al. have proposed the method of an approximate normalization of the triangle's mid-edge vector, which lies in applying the formula[2]

$$(\overrightarrow{N_A}+\overrightarrow{N_B})\cdot(\frac{5-\cos(x)}{8}).$$

It is advisable to use this method only for small angles between normalized vectors because the maximum relative error of mid-edge vector calculation is 12%.

In addition to this, the normalized midpoint vector can be determined, using the dichotomy method[18]. The formula is used[18]

$$\frac{\overrightarrow{N_A}+\overrightarrow{N_{1/2^n}}}{\sqrt{2+z_{2^n}}},$$

where $\overrightarrow{N_{1/2^n}}$ - normalized vector, obtained at the n-th iteration of division, $z_{2^n}=\sqrt{2(1+\cos(x/2^{n-1}))}$, x - the angle between $\overrightarrow{N_A}$ and $\overrightarrow{N_B}$.

The disadvantage of the approach is the presence of two square roots in the expression. Hence, the development of methods that will provide the accelerated and accurate enough vector normalization is necessary.

RESULTS

Let the coordinates of normalized vectors (X_{NA}, Y_{NA}, Z_{NA}), (X_{NB}, Y_{NB}, Z_{NB}), (X_{NC}, Y_{NC}, Z_{NC}) at respective ABC triangle vertices (Fig. 1) be given. The triangle's vertex B is connected with the point K of the side AC in parallel.

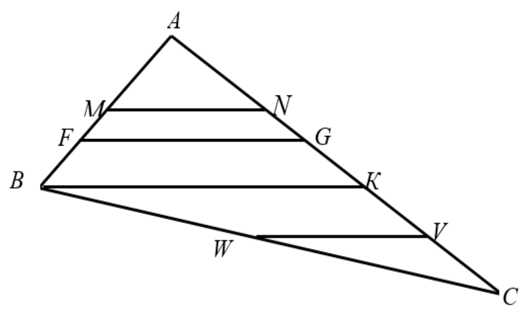

Figure 1. Triangle ABC with the rasterization lines MN, FG, BK, WV

The vector coordinate increments $\Delta X_{NFG}, \Delta Y_{NFG}, \Delta Z_{NFG}$ and $\Delta X_{NMN}, \Delta Y_{NMN}, \Delta Z_{NMN}$ are calculated for the rasterization lines between F, G and M, N respectively

$$\Delta X_{NFG}=\frac{X_{NG}\text{-}X_{NF}}{FG};\ \Delta Y_{NFG}=\frac{Y_{NG}\text{-}Y_{NF}}{FG};\ \Delta Z_{NFG}=\frac{Z_{NG}\text{-}Z_{NF}}{FG}. \tag{1}$$

$$\Delta X_{NMN}=\frac{X_{NN}\text{-}X_{NM}}{MN};\ \Delta Y_{NMN}=\frac{Y_{NN}\text{-}Y_{NM}}{MN};\ \Delta Z_{NMN}=\frac{Z_{NN}\text{-}Z_{NM}}{MN}. \tag{2}$$

The coordinates of vectors at the points F, G, M, N are calculated respectively to the coordinates of vectors at the triangle vertices A, B, C

$$X_{NF}=\frac{X_{NB}-X_{NA}}{AB}\cdot AF+X_{NA};\ Y_{NF}=\frac{Y_{NB}-Y_{NA}}{AB}\cdot AF+Y_{NA};\ Z_{NF}=\frac{Z_{NB}-Z_{NA}}{AB}\cdot AF+Z_{NA}.$$

$$X_{NG}=\frac{X_{NC}-X_{NA}}{AC}\cdot AG+X_{NA};\ Y_{NG}=\frac{Y_{NC}-Y_{NA}}{AC}\cdot AG+Y_{NA};\ Z_{NG}=\frac{Z_{NC}-Z_{NA}}{AC}\cdot AG+Z_{NA}.$$

$$X_{NM}=\frac{X_{NB}-X_{NA}}{AB}\cdot AM+X_{NA};Y_{NM}=\frac{Y_{NB}-Y_{NA}}{AB}\cdot AM+Y_{NA};Z_{NM}=\frac{Z_{NB}-Z_{NA}}{AB}\cdot AM+Z_{NA}.$$

$$X_{NN}=\frac{X_{NC}-X_{NA}}{AC}\cdot AN+X_{NA};\ Y_{NN}=\frac{Y_{NC}-Y_{NA}}{AC}\cdot AN+Y_{NA};\ Z_{NN}=\frac{Z_{NC}-Z_{NA}}{AC}\cdot AN+Z_{NA}.$$

The obtained formulas for vector coordinates calculation at the points F, G, M, N are substituted into (1)-(2). We get the expressions

$$\Delta X_{NFG}=\frac{X_{NC}-X_{NA}}{AC}\cdot\frac{AG}{FG}-\frac{X_{NB}-X_{NA}}{AB}\cdot\frac{AF}{FG};\quad \Delta X_{NMN}=\frac{X_{NC}-X_{NA}}{AC}\cdot\frac{AN}{MN}-\frac{X_{NB}-X_{NA}}{AB}\cdot\frac{AM}{MN};$$

$$\Delta Y_{NFG}=\frac{Y_{NC}-Y_{NA}}{AC}\cdot\frac{AG}{FG}-\frac{Y_{NB}-Y_{NA}}{AB}\cdot\frac{AF}{FG};\quad \Delta Y_{NMN}=\frac{Y_{NC}-Y_{NA}}{AC}\cdot\frac{AN}{MN}-\frac{Y_{NB}-Y_{NA}}{AB}\cdot\frac{AM}{MN};$$

$$\Delta Z_{NFG}=\frac{Z_{NC}-Z_{NA}}{AC}\cdot\frac{AG}{FG}-\frac{Z_{NB}-Z_{NA}}{AB}\cdot\frac{AF}{FG};\quad \Delta Z_{NMN}=\frac{Z_{NC}-Z_{NA}}{AC}\cdot\frac{AN}{MN}-\frac{Z_{NB}-Z_{NA}}{AB}\cdot\frac{AM}{MN}.$$

The formed formulas for calculating the increments of vector coordinates between F, G and M, N differ only by the second multipliers of terms ($\frac{AG}{FG}$ and $\frac{AN}{MN}$, $\frac{AF}{FG}$ and $\frac{AM}{MN}$). Since the triangles AFG and AMN are characterized by the equality of respective angles and are similar, the specified multipliers are equal

$$\frac{AG}{FG}=\frac{AN}{MN},\ \frac{AF}{FG}=\frac{AM}{MN}.$$

Then, $\Delta X_{NFG}=\Delta X_{NMN}$, $\Delta Y_{NFG}=\Delta Y_{NMN}$, $\Delta Z_{NFG}=\Delta Z_{NMN}$.

The triangles ABK and AFG are analogously characterized by the equality of similar angles and are similar. Therefore, $\Delta X_{NFG}=\Delta X_{NBK}$, $\Delta Y_{NFG}=\Delta Y_{NBK}$, $\Delta Z_{NFG}=\Delta Z_{NBK}$.

It similarly can be proven that $\Delta X_{NWV}=\Delta X_{NBK}$, $\Delta Y_{NWV}=\Delta Y_{NBK}$, $\Delta Z_{NWV}=\Delta Z_{NBK}$. The considered triangles ABK and BKC have the shared line BK. Hence, the increment of vector coordinates is constant for every triangle's rasterization line. The proven property provides the significant productivity gain for normal vectors determination, because the increments are determined for each rasterization line of triangle at once. During the shading of each triangle's rasterization line, the increments of vector components are calculated. Then, the midpoint vectors are determined by means of linear interpolation. Later, they are normalized.

It is proposed to use the method of vector calculation (Fig. 2), which lies in that the normalized vectors are determined only at odd points of rasterization line. Vectors at even points of rasterization line are found by adding the normalized vectors using the formula

$$\frac{\overrightarrow{N_{I-1}}+\overrightarrow{N_{I+1}}}{2},$$

where $\overrightarrow{N_{I-1}},\overrightarrow{N_{I+1}}$ - neighboring normalized vectors at odd points of the rasterization line.

During such approach, the error is not accumulating. It is clear that the vectors at even points are non-unit. If the error is small, the normalization of vectors at even points of the rasterization line can be avoided. Let us consider the situations, when the normalization of midpoint vectors is optional. It is proposed to use two different approaches.

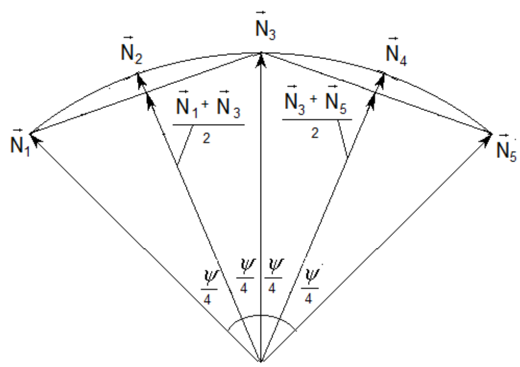

Figure 2. Method of simplified vector calculation at even points of the rasterization line

The first of them lies in the calculation of metric $d1$, which determines the length of non-normalized vector [15]

$$d1 = \cos(\frac{x}{m}),$$

where x - the angle between end vectors $\overrightarrow{N_A}$ and $\overrightarrow{N_B}$ of the rasterization line.

This expression is characterized with the computationally expensive calculation of arccosine for determining the angle x. We find the approximate expression $d1_a$ for $d1$ calculation ($m \in [3,12]$), using the linear regression equation [4]

$$\beta = (X^T X)^{-1} X^T y, \tag{3}$$

where β - vector of $d1_a$ linear coefficients, y - vector of $d1$ values, X includes ones vector, vectors of $\cos(x)$ and m values.

The obtained formula is

$$d1_a = 0.06 \cdot \cos(x) + 0.007 \cdot m + 0.894 .$$

Fig. 3 shows the plot of maximum relative error δ of $d1_a$ from $d1$ for $m \in [3,12]$, $x \in [0, \pi/2]$.

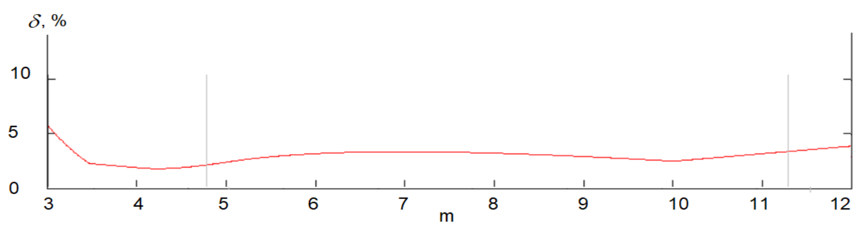

Figure 3. Plot of maximum relative errors of $d1_a$ from $d1$

The maximum relative error between $d1_a$ and $d1$ is 5%. The second approach is used when we know the cosine of angle between neighboring vectors at odd rasterization line points. It lies in calculating the metric $d2$, the formula of which was obtained according to (3)

$$d2 = \begin{cases} 0.25\cos(x) + 0.75, & x < \pi/4 \\ 0.305\cos(x) + 0.712, & x >= \pi/4 \end{cases} .$$

This formula is also applied when only one midpoint vector is calculated for the whole rasterization line. The maximum relative error from the expression $d1 = \cos(\frac{x}{2})$ ($m=2$) doesn't exceed 0.7% (Fig. 4).

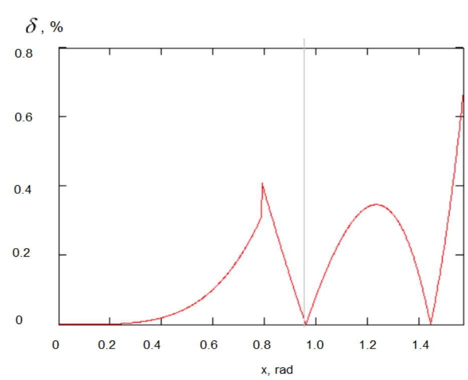

Figure 4. Plot of relative errors between d2 and d1 (m=2)

Squared Error [16, 20]) applying was conducted. If the value of $d1_a >= 0.9$ or $d2 >= 0.9$, NMSE doesn't exceed 0.0001, which means the absence of visual differences between images. If the value of $d1_a$ or d2 is smaller than 0.9, it is necessary to normalize the midpoint vector. Let us consider the question of accelerated normalized vectors calculation at the midpoint points. The formula of the midpoint unit vector is formulated according to the expression [18]

In order to compare the formed and reference images, the experimental research with the NMSE (Normalized Mean Squared Error [16, 20]) applying was conducted. If the value of $d1_a >= 0.9$ or $d2 >= 0.9$, NMSE doesn't exceed 0.0001, which means the absence of visual differences between images. If the value of $d1_a$ or d2 is smaller than 0.9, it is necessary to normalize the midpoint vector. Let us consider the question of accelerated normalized vectors calculation at the midpoint points. The formula of the midpoint unit vector is formulated according to the expression [18]

$$\frac{\overrightarrow{N_{I-1}} + \overrightarrow{N_{I+1}}}{\sqrt{2(1+\cos(\psi))}}$$

where ψ is an angle between $\overrightarrow{N_{I-1}}, \overrightarrow{N_{I+1}}$.

It is proposed to determine the unit vector according to the simplified formula, that doesn't include computationally complex operations of dividing and calculating the square root. We approximate the inverse square root term (InvC) through the polynomial regression usage. For determining the first-degree approximate expression ($\mathrm{InvC1}$), the equation (3) was used. For determining the second-degree approximate expression ($\mathrm{InvC2}$), the system of equations was used

$$\begin{bmatrix} n & \sum_{i=0}^{n} X_i & \sum_{i=0}^{n} X_i^2 \\ \sum_{i=0}^{n} X_i & \sum_{i=0}^{n} X_i^2 & \sum_{i=0}^{n} X_i^3 \\ \sum_{i=0}^{n} X_i^2 & \sum_{i=0}^{n} X_i^3 & \sum_{i=0}^{n} X_i^4 \end{bmatrix} \cdot \beta = \begin{bmatrix} \sum_{i=0}^{n} y_i \\ \sum_{i=0}^{n} X_i y_i \\ \sum_{i=0}^{n} X_i^2 y_i \end{bmatrix},$$

where X – vector of $\cos(\psi)$ ($\psi \in [0, \pi/2]$) values, n – number of vector X rows, y – vector of InvC values, β - vector of the polynomial expression coefficients. The obtained approximate formulas of the first degree and the second degree are $\mathrm{InvC1}(x) = -0.191 \cdot \cos(\psi) + 0.683$, $\mathrm{InvC2}(x) = 0.098 \cdot \cos(\psi)^2 - 0.3 \cdot \cos(\psi) + 0.703$.

Fig. 5 shows the plots of relative errors of InvC1 and InvC2 from InvC for $\psi \in [0, \pi/2]$. Maximum relative error between InvC1 and InvC is 3%, between InvC2 and InvC – 0.58%, and on the bigger part of the interval it doesn't exceed 0.27%.

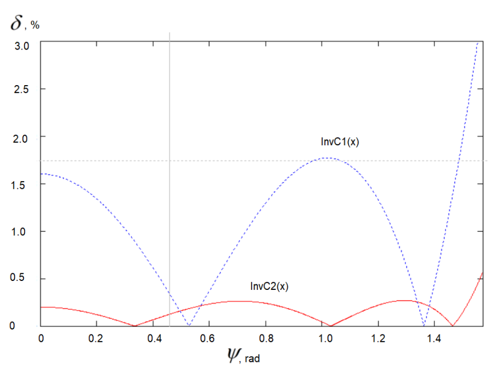

Figure 5 – Plots of relative errors of $\mathrm{InvC1}$ and $\mathrm{InvC2}$ from InvC respectively

The first variant (V1) of the method usage includes the steps: $d1_a$ criterion calculation, $d1_a < 0.9$, calculation of even midpoint vector, approximated normalization of vector using InvC1. The second variant (V2) includes the steps: $d1_a$

criterion calculation, $d1_a < 0.9$, calculation of even midpoint vector, approximated normalization of vector using InvC2 InvC2. The third variant (V3) includes the steps: $d1_a$ criterion calculation, $d1_a >= 0.9$, calculation of even midpoint vector. The fourth variant (V4) includes: d2 criterion calculation, $d2 < 0.9$, calculation of even midpoint vector, approximated normalization of vector using InvC1. The fifth variant (V5) includes: d2 criterion calculation, $d2 < 0.9$, calculation of even midpoint vector, approximated normalization of vector using InvC2. The sixth variant (V6) includes: d2 criterion calculation, $d2 >= 0.9$, calculation of even midpoint vector.

Table 1. Results of analysis of the productivity grain from the new proposed method usage

Variant of method usage	Productivity gain	Vector normalization error, %
V1	1.42	3
V2	1.37	0.58
V3	1.74	up to 10
V4	1.44	3
V5	1.39	0.58
V6	1.67	up to 10

Therefore, the highest productivity gain (1.74 times) during the average triangle shading is obtained when the deviation of $d1_a$ from unit length doesn't exceed 10% and, as result, even vectors of the rasterization line are found by adding the neighboring odd vectors.

CONCLUSIONS

In the work, it was proposed the new highly effective approach to vector normalization for the surfaces of three-dimensional objects. The approach provides the productivity increase during the shading of average triangle by 1.37-1.74 times through one-time calculation of vector coordinate increments for each rasterization line and approximate calculation of vectors at even points of the rasterization line. The proposed method can be used in highly productive systems of three-dimensional computer graphics.

REFERENCES

[1] Akenine-Moller, T. et al. [Real-Time Rendering 4th ed.], A K Peters / CRC Press, Boca Raton 56-124 (2018).

[2] Barrera, T., Hast, A. and Bengtsson, E., "Fast Near Phong-Quality Software Shading". In : Jorge J, Skala V editors. WSCG2006 Full Papers proceedings, January 31-February 2 2006, Plzen. Plzen: University of West Bohemia; p. 109-116 (2006)

[3] Gambetta, G., [Computer Graphics from Scratch], No Starch Press, San Francisco, 96-130 (2021)

[4] Goodfellow, I., Bengio, Y. and Courville, A., [Deep Learning], MIT Press, Cambridge, 104-106 (2016)

[5] Gordon, V. S. and Clevenger, J., [Computer Graphics Programming in OpenGL with C++ 2nd ed.] Dulles: Mercury Learning and Information; 26-54 (2021)

[6] Kaleniuk, O., [Geometry for Programmers], Manning, Shelter Island, 123-130 (2023)

[7] Lin, D, Seiler, L. and Yuksel, C., "Hardware Adaptive High-Order Interpolation for Real-Time Graphics". Computer Graphics Forum 40(8), 1-16 (2021); https://doi.org/10.1111/cgf.14377

[8] Lyon, R. F., "Phong Shading Reformulation for Hardware Renderer Simplification". Cupertino (CA): Apple Computer, Inc., Report No. 43, 12-33 (1993).

[9] Mar, A, (ed.), [Digital Signal Processing Applications Using the ADSP-2100 Family Vol. 1], Prentice Hall, Englewood Cliffs, 59-65 (1992).

[10] Marschner, S. et al., [Fundamentals of Computer Graphics 5th ed.], A K Peters/CRC Press, Boca Raton, 126-138 (2021).

[11] Pharr, M., Jakob, W. and Humphreys, G., [Physically Based Rendering 4th ed.], MIT Press, Cambridge 78-91 (2023)
[12] Que, A., "Polynomial regression. A PHP regression class", Polynomial Regression: Elmwood Park; (2023) [Accessed 06 June 2024] http://polynomialregression.drque.net/math.html
[13] Romanyuk, O. N., "Combined Usage of Binary and Code Linear Interpolation for Vector Normals Normalization During Three-Dimensional Objects Shading", The Bulletin of Kherson National Technical University 25, 408-411, (2006).
[14] Romanyuk, O. N. et al. "New Surface Reflectance Model with the Combination of Two Cubic Functions Usage". Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska. 13(3), 101-106 (2023)
[15] Romanyuk, O. N., Melnykov, O. M., "Adaptive Normal Vector Normalization During the Calculation of Diffuse and Specular Color Components", Data Recording, Storage & Processing 8(3), 11-19 (2006).
[16] Romanyuk, O. N. and Zavalniuk, Y. K., "Deep learning-based determination of optimal triangles number of graphic object's polygonal model" in Hovorushchenko, T., Savenko, O., Popov, P. and Lysenko, S. (eds), IntelITSIS 2024, March 28 2024, Khmelnytskyi. Khmelnytskyi: CEUR-WS, 39-51 (2024)
[17] Szeliski, R., [Computer Vision: "Algorithms and Applications" 2nd ed.], Springer Nature Switzerland, Cham, 29-59 (2022).
[18] Voitko, V. V. and Romanyuk, O. V., "Analysis of methods for normalization of vectors of normals for the tasks of formation of three dimensional images", Scientific Works of VNTU 1, 1-7 (2009)
[19] Wang, G. et al. "S-Velocity Profile of Industrial Robot Based on NURBS Curve and Slerp Interpolation", Processes 11, 2195 (2022); https://doi.org/10.3390/pr10112195.
[20] Zhang, H. et al. "Iterative reconstruction for dual energy CT with an average image-induced nonlocal means regularization", Phys Med Biol. 62(13), 5556–5574 (2017); https://doi.org/10.1088/1361-6560/aa7122