https://www.sworldjournal.com/index.php/swj/article/view/swj21-01-049
DOI: 10.30888/2663-5712.2023-21-01-049

УДК 004.92

THE ANALYSIS OF POLYGONAL MODELS DENSITY IN THREE-DIMENSIONAL COMPUTER GAMES

АНАЛІЗ ЩІЛЬНОСТІ ПОЛІГОНАЛЬНИХ МОДЕЛЕЙ У ТРИВИМІРНИХ КОМП'ЮТЕРНИХ ІГРАХ

Zavalniuk Ye.K. / Завальнюк Є.К.
PhD student / аспірант
ORCID: 0009-0005-1202-4653
Vinnytsia National Technical University, Vinnytsia, Khmelnytsky highway 95, 21021
Вінницький національний технічний університет, Вінниця, Хмельницьке шосе 95, 21021
Romanyuk O.N. / Романюк О.Н.
d.t.s., professor / д.т.н., проф.
ORCID: 0000-0002-2245-3364
Vinnytsia National Technical University, Vinnytsia, Khmelnytsky highway 95, 21021
Вінницький національний технічний університет, Вінниця, Хмельницьке шосе 95, 21021
Kotlyk S.V. / Котлик С.В.
c.t.s., as.prof. / к.т.н., доц.
ORCID: 0000-0001-5365-1200
Odesa National University of Technologies, Odesa, Kanatna St 112, 65039
Одеський національний технологічний університет, м. Одеса, Канатна 112, 65039
Stakhov O.Ya. / Стахов О.Я.
c.t.s., sen. lect. / к.т.н., ст. викл.
ORCID: 0009-0003-8986-185
Vinnytsia National Technical University, Vinnytsia, Khmelnytsky highway 95, 21021
Вінницький національний технічний університет, Вінниця, Хмельницьке шосе 95, 21021

Abstract. *The article analyzes the features of choosing the polygonal model density level for a three-dimensional computer game. The concepts of triangulation, polygonal mesh density, and level of detail are described. The relevance of adjusting the density of the object's polygonal mesh is justified. The main types of game objects are defined. The dependence of the object surface's number of triangles on the position, type and curvature of the figure, target style, platform and mode of the game, the total number of characters, the presence of animated transformations is examined. The features of calculating the curvature of a three-dimensional surface point, a polygonal mesh triangle's vertex, and a whole triangle are analyzed. The method of determining the need for additional triangulation of a triangle based on the assessment of its overall curvature is described. A table that shows the typical sizes of polygonal models in popular three-dimensional computer games over the past decade has been constructed. The features of software tools for automatization of the surface's level of detail selection are described.*

Key words: *triangulation, curvature of figure, level of detail, polygonal model density.*

Introduction.

The first major subsystem of the graphics pipeline for computer games scenes creating is the geometrical transformations subsystem [1]. One of the main steps of the stage is the formation of polygonal models [2] of scene's objects. A common approach to creating polygonal meshes is triangulation, which lies in dividing the surface of a figure into triangles. One of the characteristics of the triangulation process, which significantly affects the productivity and realism of the game scene formation, is the density [3] of the final polygonal mesh. This characteristic is defined as the number of surface polygons relative to the size of the figure. The high density of the polygonal model provides a highly realistic reproduction of the characteristics of the figure, but it does not always provide the possibility of dynamic graphics scene creation. Therefore, it is important to choose the optimal number of surface triangles for each frame’s object.

Purpose of the work.

The purpose of the article is to analyze the features of using different polygonal models density levels in three-dimensional computer games.

Game objects conventionally can be divided into characters and props (weapons, cars). Game characters are the central objects of the game, so they are visualized in more details. Usually, characters are visualized in computer games using 10 - 100 thousand triangles [4] (1 - 10 thousand for mobile games). 1-20 thousand triangles [4] are usually used to render game props (100-2 thousand for mobile games).

Mobile game models are rendered with a smaller number of triangles due to the weaker computing capabilities of phones.

The density of objects’ polygonal models depends on the target style [5] of the scene. If the goal is a photorealistic reproduction of the real world (for example, in Microsoft Flight Simulator), the surfaces of the figures are divided into the maximum reasonable number of triangles. If the goal is to form a stylized scene (for example, in Fortnite), the use of a relatively small number of triangles is necessary.

The density of the polygonal model also depends on the location and number of objects [4] in the frame. In real time, it is possible to shade only a finite number of frame figures (and, accordingly, polygons). Therefore, the more objects are placed in the frame, the fewer triangles on average can fit a single object.

Objects in the background of the scene have a small size, the increase in the number of triangles on their surfaces is often imperceptible to the player. Therefore, there may be a need to extract a significant number of microtriangles that do not affect the perception of the scene image. Different levels of detail are applied to objects for the efficient usage of computing resources [6]. The objects’ surfaces that are farthest from the camera are divided into a relatively small number of large triangles. The density of the polygonal model of the closest to the camera objects is the highest. It is recommended to reduce the number of triangles by 50% at each level of detail [6].

Game objects operating in the multiplayer mode [4] are characterized by additional restrictions on the number of surface triangles. In addition to rendering a large number of characters, computing resources are spent on updating the state of the game server.

The number of polygonal model's triangles directly depends on the level of curvature of the object’s surface. To preserve the properties of complex curved surfaces, an increase in the number of surface triangles is necessary. The curvature of a three-dimensional surface at a point can be represented by Gaussian [7] or mean [7] curvature. The Gaussian curvature of the surface K is calculated [8] as the product of the principal curvatures k_1, k_2 (the maximum and minimum curvatures of the object’s surface intersections with the normal planes)

$$K = k_1 \cdot k_2 = \frac{LN - M^2}{EG - F^2},$$

where $E = 1 + f_x^{\,2}$, $F = f_x \cdot f_y$, $G = 1 + f_y^{\,2}$,

$$L = \frac{f_{xx}}{\sqrt{1 + f_x^{\,2} + f_y^{\,2}}},\ M = \frac{f_{xy}}{\sqrt{1 + f_x^{\,2} + f_y^{\,2}}},\ N = \frac{f_{yy}}{\sqrt{1 + f_x^{\,2} + f_y^{\,2}}}$$ (f_{coef} – the derivative of the surface formula $f(x,y)$ by $coef$ variable).

The mean curvature H is the mean value of principal curvatures. It is calculated using the formula [8]

$$H = \frac{k_1 + k_2}{2} = \frac{1}{2} \cdot \frac{LG - 2MF + NE}{EG - F^2}.$$

The calculation of the curvature at the triangle's vertex [9] is as follows. A vertex is selected around which a circle of triangles is positioned (Fig. 1). Then the Gaussian curvature is calculated according to the expression [9]

$$K = \frac{2\pi - \sum \theta_j}{A},$$

where θ_j - the angle of the adjacent to the vertex $j-th$ triangle, A - the sum of the thirds of areas of the adjacent to the vertex triangles.

The mean curvature at the triangle's vertix [9] is calculated using the formula

$$\frac{1}{2}\left\| \frac{1}{2A} \sum (\cot \alpha_{ij} + \cot \beta_{ij})(p_j - p_i) \right\|,$$

where p_j, p_i - the positions of the adjacent and central vertex respectively, $\cot \alpha_{ij} + \cot \beta_{ij}$ - cotangent weights.

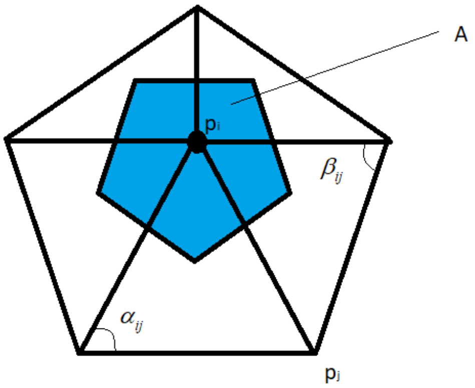

Figure 1 – The data for triangle curvature calculation

Source: [9]

If the triangle has a large curvature, it can be divided into flatter triangles to simplify the shading procedure. The total curvature can be estimated by many methods.

In the simplest case, the deviation of the sum of triangle's angles from π is used to estimate the total curvature of a triangle.

One of the approaches is based on the analysis of a spherical triangle [10] formed from the normals $\vec{N}_A, \vec{N}_B, \vec{N}_C$ of the polygon's vertices. It is assumed that the normals of the triangle's vertices come from a common point (Fig. 2a [10]). The curvature is large if the angles (Fig. 2 b [10]) between the edges of a spherical triangle are greater than the threshold value. To simplify the comparison of angles, it is possible to use modules of vectors $(\vec{N}_A - \vec{N}_B)$, $(\vec{N}_A - \vec{N}_C)$, $(\vec{N}_B - \vec{N}_C)$.

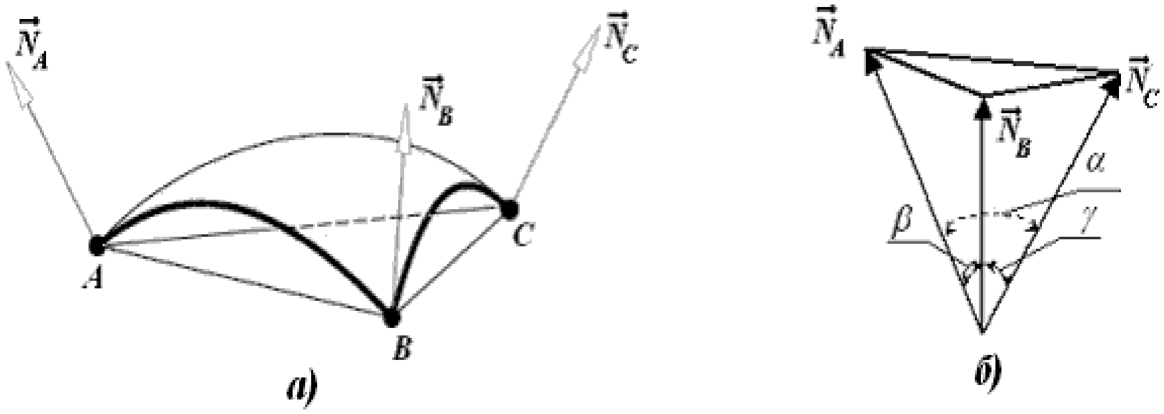

Figure 2 – The usage of triangle vertices' normals for the calculation of triangle's curvature

Source: [10]

If the polygonal model is deformed during scene animation, the number of surface triangles should not be too small. Otherwise, the loss of a smooth transition between different states of the object is possible.

Table 1 provides information about the number of used surface polygons [11, 12] in famous computer games.

Table 1 – The number of used surface polygons in famous computer games

Game	*Year*	*Number of used polygons*
Assassin's Creed 3	2012	Benjamin Franklin – 17 744 Charles Lee - 25 994
Mass Effect 3	2012	Shepard – 25 440 EDI – 28 831 Reaper – 121 021
Castlevania: Lords of Shadow	2013	Black knight – 21 718 Cornell – 19 031
Infamous Second Son	2014	Character – 120 000
Call of Duty: Black Ops 3	2015	Futuristic Car – 29 150 VTOL Aircraft – 67 154
Crash Bandicoot N. Sane Trilogy	2017	Spaceship – 80 750 Motorcycle – 59 483
Vampyr	2018	Bed – 11 218 Bookcase – 6 531 Piano – 23 449
Hell Let Loose	2018	Knife – 3 000 Howitzer – up to 37 400 Willis Jeep –up to 110 000 Land Mine – up to 5 486

Source: [11, 12]

As shown in the table, the number of surface polygons of characters and objects varies significantly depending on the characteristics and goals of the game.

Another category of objects are test figures that are used to test the methods of game scenes redering. Usually, testing is done on figures of several levels of complexity. The quality of the simplest objects shading is checked on low-polygonal figures. For example, Sphere and Suzanne in Blender are built from 80 [13] and 968 [14] triangles, respectively. For testing more complex models, the Utah teapot, types of which are built from 3488 – 145620 triangles [15], can be used. When testing the most complex shapes, the "Stanford Dragon" (1132830 triangles) or "Armadillo" (7500000 triangles) can be used [16].

Adjusting the densities of polygonal object models is a complex operation during game development. Therefore, technologies for automatic determination of the object's level of detail have been developed. For example, Nanite [17] in Unreal Engine 5 facilitates the process of shading polygons in real time by dynamically optimizing the polygonal model. The downside of Nanite is that the technology only applies to static objects.

Conclusions.

To ensure the formation of the game frame in real time, it is necessary to adjust the density values of the polygonal object models. If the object is the central character, is located in the foreground, is characterized by high curvature, the density of its polygonal model is high. If the object is a background object, has a flat shape, then a small number of triangles is used.

References:

1. Romanyuk, O. et al. (2023) ‘The Overview of Neural Rendering’, *Modern Engineering And Innovative Technologies*, (27), pp. 129–134. doi:https://doi.org/10.30890/2567-5273.2023-27-01-060.

2. Romaniuk, O., Romaniuk, O. and Chekhmestruk, R. (2023) *Kompiuterna hrafika [Computer Graphics]*. Vinnytsia, Ukraine: VNTU.

3. Obidnyk, M. and Romaniuk, O. (2013) ‘Pryskorena normalizatsiia vektoriv dlia formuvannia zobrazhen vysokopolihonalnykh stsen’ [‘Accelerated normalization of vectors for image formation of highly-polygonal scenes’], *Informatsiini tekhnolohii ta kompiuterna inzheneriia*, (1), pp. 66–73.

4. Maxwell, W. (2021) *How many polygons should a game model have, CG Obsession*. Available at: https://cgobsession.com/how-many-polygons-should-a-game-model-have/ (Accessed: 20 September 2023).

5. *How Many Polygons Can Unreal Engine Handle, OpenWorldLearning*. Available at: https://www.openworldlearning.org/how-many-polygons-can-unreal-engine-handle/#google_vignette (Accessed: 17 September 2023).

6. *Geometry, Android Developers*. Available at: https://developer.android.com/games/optimize/geometry (Accessed: 20 September 2023).

7. Gauthier, S. et al. (2017) ‘Digitized 3D mesh segmentation based on curvature analysis’, *Electronic Imaging*, 29(20), pp. 33–38. doi:10.2352/issn.2470-1173.2017.20.3dipm-005.

8. Jia, Y.-B. (2020) *Gaussian and Mean Curvatures, Faculty Website Directory*. Available at: https://faculty.sites.iastate.edu/jia/files/inline-files/gaussian-curvature.pdf (Accessed: 20 September 2023).

9. Vaillant, R. *Curvature of a triangle mesh, definition and computation, Rodolphe Vaillant's homepage*. Available at: http://rodolphe-vaillant.fr/entry/33/curvature-of-a-triangle-mesh-definition-and-computation (Accessed: 20 September 2023).

10. Romaniuk, O. (2008) 'Metod pryskorenoho zafarbovuvannia tryvymirnykh poverkhon z urakhuvanniam yikh lokalnoi kryvyzny' ['The method of accelerated coloring of three-dimensional surfaces taking into account their local curvature'], *Visnyk Skhidnoukrainskoho natsionalnoho universytetu*, (12), pp. 166–172.

11. *Polycounts in Next Gen Games Thread!, Polycount.* Available at: https://web.archive.org/web/20211207155252/http://polycount.com/discussion/141061/polycounts-in-next-gen-games-thread (Accessed: 20 September 2023).

12. '*Yes, but how many polygons?' an artist blog entry with interesting numbers, Beyond3D Forum*. Available at: https://forum.beyond3d.com/threads/yes-but-how-many-polygons-an-artist-blog-entry-with-interesting-numbers.39321/ (Accessed: 20 September 2023).

13. *Icosphere, 3D Warehouse*. Available at: https://3dwarehouse.sketchup.com/model/f9a3a35d8cbae5671d8475b23434c252/Icosphere?hl=en (Accessed: 20 September 2023).

14. *Sample 3D Model: Suzanne, Wolfram Data Repository*. Available at: https://datarepository.wolframcloud.com/resources/Sample-3D-Model-Suzanne/ (Accessed: 20 September 2023).

15. Johnson, D. *Utah Teapot, Utah Model Repository*. Available at: https://users.cs.utah.edu/~dejohnso/models/teapot.html#:~:text=Utah%20Teapot&amp;text=This%20format%20is%20readable%20by,and%20145%2C620%20triangles. (Accessed: 20 September 2023).

16. *The Stanford 3D Scanning Repository, Computer Graphics at Stanford University*. Available at: http://graphics.stanford.edu/data/3Dscanrep/ (Accessed: 20 September 2023).

17. Maxwell, W. (2021) *Does Unreal engines 5 nanite make poly count irrelevant, CG Obsession*. Available at: https://cgobsession.com/does-unreal-engines-5-nanite-make-poly-count-irrelevant/ (Accessed: 20 September 2023).

Анотація. *У статті проаналізовано особливості вибору рівня щільності полігональної моделі для тривимірної комп'ютерної гри. Описано поняття триангуляції, щільності полігональної сітки, рівня деталізації. Обґрунтовано актуальність налаштування щільності полігональної сітки об'єкта. Визначені основні типи ігрових об'єктів. Розглянуто залежність кількості трикутників поверхні об'єкта від положення, типу та кривизни фігури, цільового стилю, платформи та режиму гри, загального числа персонажів, наявності анімаційних трансформацій. Проаналізовано особливості обчислення кривизни точки тривимірної поверхні, вершини трикутника полігональної сітки, цілого трикутника. Описано метод визначення необхідності додаткової триангуляції трикутника на основі оцінки його загальної кривизни. Побудовано таблицю, що відображає типові розміри полігональних моделей у популярних тривимірних комп'ютерних іграх за останнє десятиріччя. Описано особливості програмних інструментів для автоматизації вибору рівня деталізації фігури.*

Ключові слова: *триангуляція, кривизна фігури, рівень деталізації, щільність полігональної моделі.*

Article sent:2023.