

УДК 004.94: 519.813.7

DOI: 10.18372/2310-5461.35.11837

*К. О. Моргун*

Національний авіаційний університет

orcid.org/0000-0002-0955-4750

e-mail: mrsammamaj@gmail.com

## ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РЕНДЕРІНГУ 3D-ОБ'ЄКТІВ

### Постановка проблеми

Технології створення синтезованих моделей засобами комп'ютерної графіки широко використовуються в мультимедіа, цифровій фотографії та сучасному кінематографі, де майже кожен кадр фільму являє собою високоякісне синтезоване об'ємне зображення, яке виходить з тривимірної моделі шляхом прорахунку освітленості. При цьому виконується величезна кількість обчислень, пов'язаних з розрахунками перевідбитків променів від поверхонь об'єктів сцени, що випускаються одним або декількома джерелами світла та потрапляють в об'єктив віртуальної камери, яка, в свою чергу, теж задається низкою параметрів. Така операція комп'ютерної графіки відома як рендерінг.

Метою рендерінгу є отримання фотореалістичних зображень, що вимагає розрахунку віддаленості елементарної точки зображення, визначення її освітленості та текстури. При по піксельному обчисленні всього зображення експерименти, що проводяться, тривають декілька годин на потужних сучасних комп'ютерах. Враховуючи, що для здійснення проміжних результатів виконується приблизно 100–1000 арифметичних результатів, що означає виконання приблизно  $10^{17}$ – $10^{20}$  операцій з плаваючою точкою, потребує також значного обсягу пам'яті для збереження проміжних результатів. Обчислювальна система із продуктивністю  $10^{14}$  операцій з плаваючою точкою при повній завантаженості та неефективному програмуванні буде здійснювати цей експеримент протягом декількох годин. Таким чином, економія обсягу пам'яті та скорочення обчислень потребує розроблення нових підходів для здійснення подібних операцій.

Так, прорахунок одного кадра зображення роздільною здатністю  $1920 \times 1080$  пікселів (FullHD), сцена якого складається з 230 656

полігонів, 4-х джерел світла і однієї камери) комп'ютером AMD Athlon (TM) X4 860K с процесором Quad 3,70 ГГц в тандемі з 16 Гб ОЗП займає близько півгодини.

Але 200 000 кадрів тієї ж роздільної здатності рендер-ферма REBUSfarm може виконати за 28,80 год, здійснивши для цього 500 обчислювальних вузлів.

Обчислення повинні застосувати такі програмні та апаратні засоби, які здатні їх проводити за обмежений час. Можливості підвищення показників апаратних засобів за рахунок підвищення частоти та щільності упакування логічних елементів на кристалі, поступово вичерпуються. Останнім часом розвиваються інші підходи, які основані на багато процесорній побудові комп'ютерних систем, набувають розвитку паралельні обчислення програмними засобами [1; 2].

Проблему збільшення швидкості візуалізації умовно можна вирішити двома способами: перший полягає в нарощуванні потужностей апаратної частини, а другий в оптимізації алгоритмів прорахунку зображень. Недоліками першого підходу є: збільшення площі, що займають обчислювальні вузли, підвищене енергоспоживання і тепловиділення; до недоліків другого — фізично не коректна освітленість. Оптимізація, як правило, полягає в зменшенні кількості полігонів і зниженні якості текстур об'єктів, що знаходяться в зоні слабкої видимості, що призводить до зниження якості зображень, що є не припустимим.

### Аналіз останніх публікацій

У попередніх працях [3–5] виконаний аналіз обчислювальних витрат існуючими програмними засобами та встановлені ефективні за часом засоби для побудови синтезованих об'єктів. Так, у працях [3; 4] проведена оцінка прорахунку

променя в барицентричних координатах, яка є кращою порівняно з методами розрахунку в декартових координатах; у праці [4] показано результати візуалізації різних тривимірних об'єктів, представлені результати оцінки освітленості тривимірного об'єкта методом Монте-Карло наведено у праці [6; 7]. У представлених роботах показано, що це завдання обчислювальноємне, що потребує проведення додаткової оптимізації. Крім того, існує третій шлях, заснований на розпаралелюванні існуючих алгоритмів на основі проведення звичайної і конвексної векторизації [2; 8].

**Мета** статті — опис розробленої інформаційної технології отримання реалістичних зображень засобами комп'ютерної графіки із визначенням основних напрямків оптимізації обчислень.

### Точка перетину в декартових координатах

На початку роботи з об'єктами комп'ютерної графіки обирається система координат, напрямком обходу координат, визначається тип освітлення об'єкта та відбувається його синтез за методикою [5]. Важливим етапом моделювання є освітлення об'єкта, де встановлюється віддалення його від камери чи глядача. Оскільки об'єкт є сукупністю примітивів, то камера збирає промені від примітивів об'єкта, які є відбитками джерела освітлення, при цьому встановлюється віддаленість до кожного. Останню операцію зручно робити шляхом збирання пучка променів відбитих від примітивів об'єкта в параметричній формі [3], тоді точка відбиття є точкою перетину  $P$  променя та примітиву об'єкта

$$P = O + tR, \quad (1)$$

де  $O$  — точка розташування камери, можна вважати її як початок координат,  $t$  — параметр,  $t \in [0, T]$ ;  $T$  — найбільша відстань у межах сцени,  $R$  — напрямний вектор променя в тривимірному декартовому просторі.

Таким чином, щоб встановити точку перетину променю та примітиву треба знайти  $t$  та підставити його у рівняння (1). Відповідно до [3] параметр  $t$  визначається за формулою

$$t = \frac{n \cdot O + D}{n \cdot R}, \quad (2)$$

де  $n$  — нормаль до площини, де розташований примітив,  $D$  — віддаленість площини від камери. Враховуючи, що  $n$ ,  $O$ ,  $R$  — тривимірні вектори, то в чисельнику та знаменнику (2) знаходяться скалярні добутки цих векторів, при цьому повинно виключити випадок  $(n, R) \neq 0$ , коли промінь та площина паралельні. Нормаль  $n$  розраховується за координатами будь-якого кута примітиву як векторний добуток векторів, що його утворюють.

Розрахунок (2) без урахування розрахунку нормалі вимагає шість множень, п'ять додавань та одне ділення з дійсними числами. Враховуючи кількість примітивів в об'єкті та необхідність визначення у подальшому текстури об'єкта, стає зрозумілим необхідність в оптимізації проведених обчислень.

Найпростішою оптимізацією розрахунків є аналіз скалярного добутку знаменнику в рівнянні (2) та виключення неможливих та зайвих прорахунків. До цього відноситься виключення з розрахунків скалярних добутків  $(n, R) \leq 0$ , що відповідає випадкам паралельності площини та променю (рівність) та променів, що знаходяться поза камерою (негативність). Ще одну можливість в скороченні розрахунків дає врахування непрозорості об'єкту, тобто відбиття променів тільки від лицьової поверхні об'єкта, тобто виключення негативних скалярних добутків  $(n, O) < 0$  з чисельника (2).

### Точка перетину в барицентричних координатах

Внутрішня точка опуклої фігури, що знаходиться на площині, зручно представити залежністю в системі ненегативних дійсних чисел, для яких виконується тотожність [9]

$$\lambda_1 + \lambda_2 + \dots + \lambda_n = 1. \quad (3)$$

Числа  $\lambda_i \geq 0$  називають *барицентричними координатами*;  $n$  — кількість координат, що дорівнює кількості вершин фігури, а точка  $P$ , що знаходиться в середині фігури, визначається рівністю

$$P = V_0\lambda_1 + V_1\lambda_2 + \dots + V_n\lambda_n,$$

де  $V_i$  — вершини трикутника.

Якщо фігура трикутник, то  $n = 3$  і точка  $P$  в барицентричних координатах може бути записана як

$$P = A + \lambda_2(B - A) + \lambda_3(C - A).$$

У рівнянні (3) враховано, що  $\lambda_1 = 1 - \lambda_2 - \lambda_3$ . Це дозволяє скоротити кількість складних обчислень.

Відповідно до алгоритму Möller-Trumbore [3; 4] координати точки перетину трикутника можна знайти з рівняння

$$(t, \lambda_2, \lambda_3)^T = \det^{-1}(M) (\det(M_1), \det(M_2), \det(M_3))^T,$$

де  $M$ ,  $M_1$ ,  $M_2$ ,  $M_3$  — матриці виду

$$M = (-D, (B - A), (C - A));$$

$$M_1 = ((O - A), (B - A), (C - A));$$

$$M_2 = (-D, (O - A), (C - A));$$

$$M_3 = (-D, (B - A), (O - A)).$$

Перевагою цього алгоритму є незалежність барицентричних координат від перетворень, виконаних з об'єктом, оскільки їх відносять до координат площинного типу. Ця властивість корисна при збереженні кольорів окремих областей трикутника. Незважаючи на елегантність алгебричного рішення цієї проблеми, в обчислювальному сенсі воно досить громіздке. Кожен детермінант вимагає 9 множень, 3 віднімання та 2 додавання, взагалом це завдання для 1 трикутника потребує 66 множень, 12 віднімань, 8 додавань та 1 ділення з дійсними числами.

**Визначення текстури**

Текстура відповідно до визначення з [10] є зображення, що відтворює візуальні властивості поверхні чи об'єкта. В найпростішому випадку вона характеризується яскравістю та кольором матеріалу, що покриває об'єкт. Ці параметри визначаються на кінцевому етапі вирішення задачі визначення освітлення об'єкта шляхом вирішення інтегрального рівняння Фредгольма 2-го роду [6; 7; 11; 12].

$$L(x, \omega, \lambda, t) = \int_{\Omega} L_i(x, \omega_i, \lambda, t) d\omega_i, \quad (4)$$

де  $L(\cdot)$  — кількість світла, що діє на спостерігача з точки  $x$ , яку він бачить під напрямком  $\omega$  в діапазоні довжин хвиль видимого світла  $\lambda$  в момент часу  $t$ ;  $\Omega$  — освітлювана область;  $i$  — її окрема ділянка.

Для вирішення рівняння (4) застосовують метод Монте-Карло, який дозволяє інтеграл правої частини цього рівняння замінити виразом [6; 7]

$$I_x = \frac{\lambda_{2r} - \lambda_{1r}}{N} \sum_{i=0}^{N-1} S(\lambda) x(\lambda). \quad (5)$$

У виразі (5)  $N$  — обсяг вибірки, яка утворюється кількістю променів, що приймаються обраною довжиною хвилі, що знаходиться в діапазоні  $\lambda \in [\lambda_1, \lambda_2]$ ;  $S(\lambda)$  — спектральна щільність сили світла;  $x(\lambda)$  — функція спостереження кольорових координат, значення якої дорівнюють силі відбитого світла. Величина  $S(\lambda)$  обирається зі спеціальних таблиць для визначеної довжини хвилі світла. Якщо обрано кольорове подання об'єкту, тоді слід визначати (5) для кожної кольорової координати:

$$I_x = \frac{\lambda_{2r} - \lambda_{1r}}{N} \sum_{i=0}^{N-1} S(\lambda) x_r(\lambda);$$

$$I_y = \frac{\lambda_{2g} - \lambda_{1g}}{N} \sum_{i=0}^{N-1} S(\lambda) x_g(\lambda);$$

$$I_z = \frac{\lambda_{2b} - \lambda_{1b}}{N} \sum_{i=0}^{N-1} S(\lambda) x_b(\lambda),$$

де  $\lambda_{1r}, \lambda_{2r}, \lambda_{1g}, \lambda_{2g}, \lambda_{1b}, \lambda_{2b}$  — границі довжини хвиль червоного ( $r$ ), зеленого ( $g$ ) та синього кольорів ( $b$ ).

Перехід від координатної системи  $X, Y, Z$  до простору RGB здійснюється звичайним лінійним перетворенням [6, 7]

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 0,41847 & -0,15865 & -0,08284 \\ -0,09117 & 0,25243 & 0,01571 \\ 0,00092 & -0,00255 & 0,17860 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}.$$

Оскільки виведення зображення відбувається на екран монітора, що є нелінійним пристроєм, то виникає необхідність додаткової гамма-корекції над кожним значенням моделі RGB у формі [6, 13]

$$C_{RGB} = \begin{cases} 12,92 C_{lim}, & \text{якщо } C_{lim} \leq 0,0031308 \\ (1+a) C_{lim}^{1/2,4} - a, & \text{якщо } C_{lim} > 0,0031308 \end{cases} \quad (6)$$

де  $a = 0,55$ . При цьому отримується модель sRGB.

Отримані з рівняння (6) значення приводяться до діапазону  $[0, 1]$  та далі перетворюються до діапазону  $[0, 255]$  шляхом множення на коефіцієнт 255, результат множення округлюється до цілих чисел.

**Збереження інформації**

Інформація щодо сцени зберігається у файлах, серед яких найбільш поширеними вважаються файли форматів OBJ та FBX. Ці формати дозволяють зберігати дані сцени, які складаються з геометричних даних (координат вершин, текстурних координат, нормалей), дані камери та розмір зображення. Приклад опису куба в форматі OBJ подано на рис. 1.

```
PointsPolygons [4 4 4 4 4] [0 1 3 2 2 3 5 4 4 5 7 6 6 7 1
0 1 7 5 3 6 0 2 4] "P" [-0.5 -0.5 0.5 0.5 -0.5 0.5 -0.5 0.5
0.5 0.5 0.5 -0.5 0.5 -0.5 0.5 0.5 -0.5 -0.5 -0.5 0.5 -
0.5 -0.5] "facevarying normal N" [0 0 1 0 0 1 0 0 1 0 0 1 0
1 0 0 1 0 0 1 0 0 0 -1 0 0 -1 0 0 -1 0 0 -1 0 -1 0 0 -1
0 0 -1 0 0 -1 0 1 0 0 1 0 0 1 0 0 1 0 0 -1 0 0 -1 0 0 -1 0 0 -
1 0 0] "facevarying float s" [0.375 0.625 0.625 0.375
0.375 0.625 0.625 0.375 0.375 0.625 0.625 0.375 0.375
0.625 0.625 0.375 0.625 0.875 0.875 0.625 0.125 0.375
0.375 0.125] "facevarying float t" [0 0 0.25 0.25 0.25 0.25
0.5 0.5 0.5 0.5 0.75 0.75 0.75 0.75 1 1 0 0 0.25 0.25 0 0
0.25 0.25] "constant string primtype" ["mesh"]
```

Рис. 1. Приклад опису шестигранника

Опис починається ключовим словом «PointsPolygons», яке повідомляє програмі відтворення, що подальша супроводжувальна інформація відноситься до полігональної мережі. Перший масив індексів багатокутників, який складається з набору цілих чисел, говорить про тип використаних багатокутників в мережі. В прикладі розмір цього масиву дорівнює 6, що

відповідає кількості граней (багатокутників) у об'єкта, кожна площина має 4 вершини. Наступний масив — це індекси вершин. Перші чотири числа — це індекси вершин першої грані об'єкта і т. д. Координати вершин задаються масивом, який слідує за змінною "P". Ключовим словом «facevarying» об'являються простіші змінні вершин і кожної грані, до яких відносяться нормалі  $N$  та координати текстури  $s$ ,  $t$ . Опис нормалей слідує за змінною  $N$ , а координати  $s$  та  $t$  подані змінними  $s$  плаваючою точкою.

### Технологія обчислень

Найбільш вдалим підходом щодо розбиття повної задачі на простіші задачі є рівномірне поділення часу розрахунку між окремими процесорами. Крім того, повинно бути введено специфічні властивості, які забезпечують виконання таких операцій. До них відносяться зернистість, за допомогою якої зображення поділяється на елементарні блоки. Розрізняють крупнозернисті та дрібнозернисті зображення, які відрізняються обсягом обчислень. При цьому зрозуміло, чим крупніше елементарні блоки, тим швидше відбувається оброблення. Явне чи неявне паралельне обчислення передбачає розроблення шляхів паралельного обчислення або проведення розрахунків на багатопроекторному комп'ютері. При цьому повинні бути враховані масштабованість, синхронізація та затримання процесів. Масштабованість надає можливість збільшення потужності (підключення додаткових процесорів) для вирішення початкового завдання. Синхронізація не припускає накладання обчислювальних процесів. Затримка процесів повинна забезпечити передавання даних за запитами [8].

Оскільки кількість процесорів для виконання поставленого завдання є обмеженим числом, то відповідно й кількість обчислювальних потоків повинна бути обмеженою.

Якщо обчислювальний потік обслуговує один процесор, то кількість обчислювальних потоків задається кількістю процесорів в комп'ютерній системі.

Таким чином й зображення повинне бути поділений на таку ж кількість частин.

Різні підходи до ділення зображення на чотири частини показані на рис. 2.

Обчислювальна система будується за принципом клієнт-серверної системи.

Вихідне зображення знаходиться на клієнтському процесорі, поділяється на частини, кількість яких відповідає кількості процесорів в системі. Кожна частина зображення створює обчислювальний потік, який направляється для проведення розрахунків відповідним процесорам, які попередньо налагоджуються для проведення розрахунків.

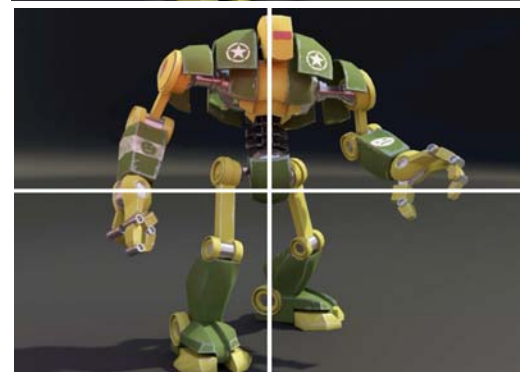
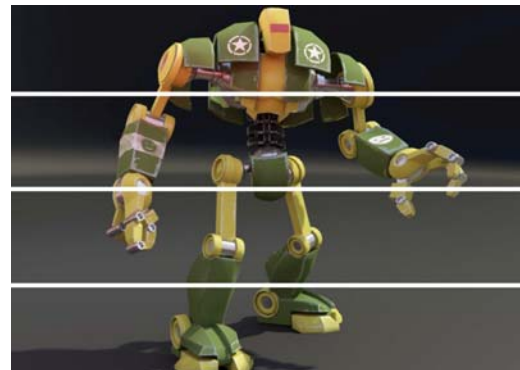
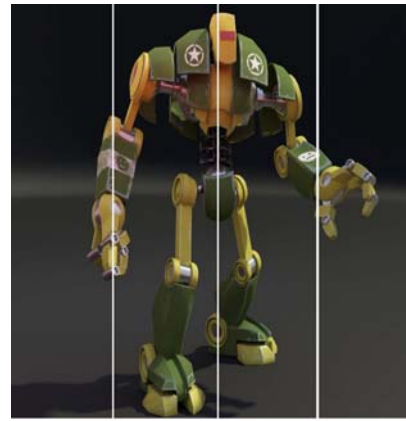


Рис. 2. Підходи до ділення зображень

Після проведення розрахунків кожна частина повертається клієнтському процесору, де зображення збирається у повне.

Оцінка якості зображення проводиться за критеріями відношення сигнал/шум (PSNR) та середньоквадратичної похибки (MSE). Встановлюються граничні показники цих критеріїв  $PSNR > 20$  дБ та  $MSE < 0.001$ , які приймаються за показники якісного зображення. Якщо якість зображення не відповідає цим показникам, зображення повертається для оброблення. Зображення, що пройшло оброблення та задовольняє встановленим показникам якості повертається на клієнтський процесор для відтворення засобами відображення, дії серверних процесорів деактивуються.

З метою підвищення швидкості розрахунків кожна частина поділяється на менші за розміром та здійснюється розрахунок з більш дрібними частинами.

**Алгоритм**

Блок схема алгоритму трасування променя з використанням глобальної моделі освітлення показана на рис. 3. В алгоритмі використовується стек променів магазинного типу, який призначений для передачі інформації про відбите і пропущене світло між елементами дерева променя. В кожен момент часу стек містить в собі тільки частину дерева, тому його об'єм можна обмежити найдовшою ймовірною гілкою.

Гілка дерева закінчується, коли відбитий і заломлений промені йдуть за межі сцени або у випадку переповнення стеку. Якщо обидва промені залишають сцену, то їх внесок в інтенсивність початкового променя рівна нулю. Якщо стек переповнюється, то алгоритм розраховує інтенсивність первісного променя, використовуючи лише компоненти розсіяного, дифузного і дзеркального відбиття в точці падіння променя.

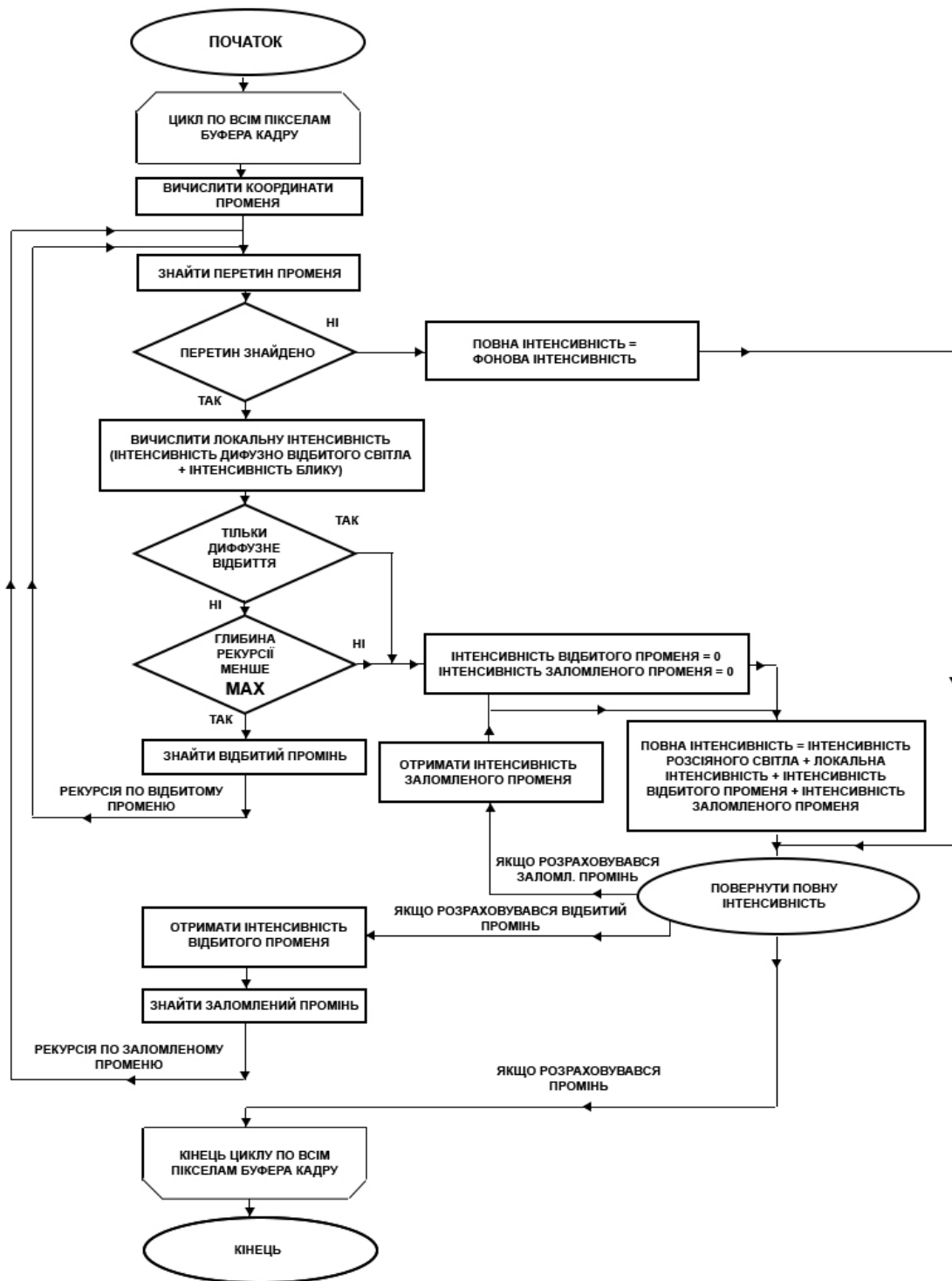


Рис. 3. Алгоритм розрахунку глобального освітлення



Стек променів містить дані номера променя, що однозначно визначає промінь, тип променя та номер первісного променя — номер, що породжує даний промінь.

### Моделювання

Створюється 3D-модель об'єкта відповідно до методики викладеної в праці [5]. Персонаж піддавався полігональному моделюванню та текстуруванню. В разі проведених операцій було отримано модель, яка складається із 29462 трикутників, 2-х кольорових текстур та 4 матеріалів (рис. 4). Після створення моделі відбувається налаштування матеріалів (шейдерів) та текстур зображення. На рис. 5. показаний інтерфейс панелі матеріалів середовища 3ds max.



Рис. 4. Форма моделі у середовищі 3ds Max

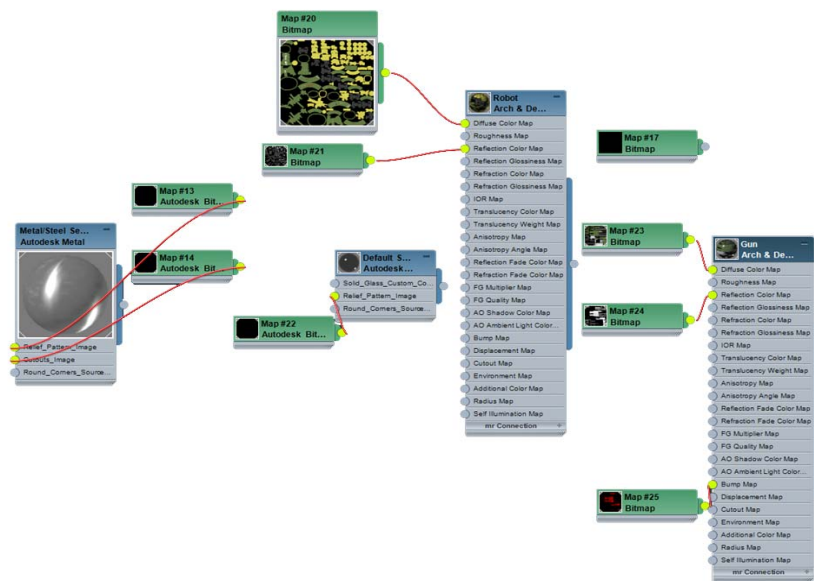


Рис. 5. Підключення текстур і матеріалів до моделі

На рис. 6. Зображено процес візуалізації зображення обчислювальними блоками ЦПП (процесор Intel Core i7 4770 - 4 ядра (8 потоків), 3.4GHz).

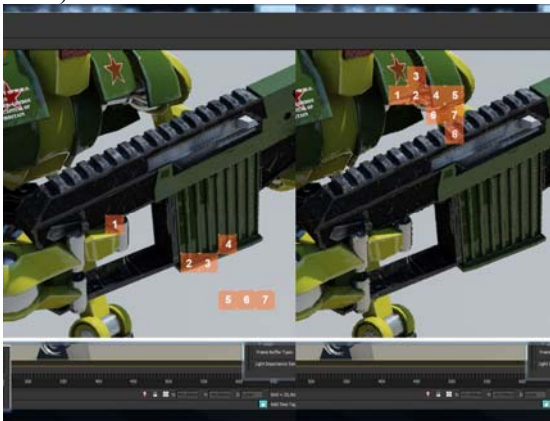


Рис. 6. Розрахунок зображення за допомогою восьми логічних блоків процесора

Кожен з логічних блоків відображається у вигляді квадрату і переміщується в рамках зображення під час обчислень, що дає змогу відслідковувати прораховані області зображення, та фіксувати проміжні стадії (рис. 7.) рендерингу.

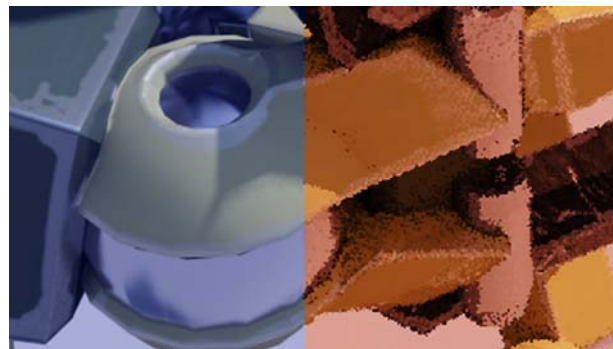


Рис. 7. Різниця фінального зображення (ліворуч) і проміжного етапу прорахунку (праворуч)

Обчислення відбувається в два етапи: прорахунок точок заключного збору зображення (Final gathering) та прорахунок фінального етапу. Заключний збір є методом оцінки глобальної освітленості для заданої точки зображення шляхом вибірки ряду напрямків над півкулею над цією точкою (такий набір семплів називається заключною точкою зібрання), або шляхом усереднення ряду сусідніх точок (з метою оптимізації обчислень).

У першому випадку орієнтація півкулі визначається нормаллю трикутника, на поверхні якого лежить точка.

У прорахунках були використані наступні налаштування фінальної збірки: початкова щільність точок — 0.8 на піксель; кількість променів на точку зображення — 250; інтерполяція — 30 вибірок.

Кожен з логічних блоків відображається у вигляді квадрату і змінюється в рамках зображення під час обчислень, що дає змогу відслідковувати прораховані області зображення, та фіксувати проміжні стадії.

Заключний етап виконується за методом глобальної освітленості для заданої точки зображення шляхом вибірки ряду напрямків над півкулею над цією точкою або шляхом усереднення ряду сусідніх точок.

У першому випадку орієнтація півкулі визначається нормаллю трикутника, на поверхні якого лежить точка.

У прорахунках були використані наступні налаштування фінальної збірки: початкова щільність точок — 0.8 на піксель; кількість променів на точку зображення — 250; інтерполяція — 30 вибірок.

### Висновки

У статті представлена інформаційна технологія, яка використовується при моделюванні фото-реалістичних зображень засобами комп'ютерної графіки. Технологія складається з трьох етапів трасування сцени, рішення завдання глобального освітлення та збереження цієї інформації.

Така технологія вимагає багато часу на виконання, що потребує оптимізації деяких процесів обчислення. Така оптимізація полягає у детальному аналізі зображення, виключенні зайвих проходів та обчислень при трасуванні та застосуванні розбивки освітлення між окремими процесами, що дозволяє отримати позитивний результат.

Подальші дослідження слід направити на детальне вивчення технології паралельного розподілу обчислювальних процесів при проведенні розрахунків.

### ЛІТЕРАТУРА

1. **Стиренко С. Г.** Форма представления функциональных программ для автоматического параллельного выполнения / С. Г. Стиренко, Д. В. Грибенко, А. И. Зиненко // Вісник НТУУ «КПІ», Інформатика, управління та обчислювальна техніка. — № 50. — 2009. — С. 190–194.

2. **Воеводин В. В.** Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. — СПб. : БХВ-Петербург, 2002. — 608 с.

3. **Каазик Ю. А.** Математический словарь / Ю. А. Каазик. — Таллин : Валгус, 1985. — 296 с.

4. **Моргун К. О.** Алгоритми трасіровки променів в завданнях комп'ютерної графіки / К. О. Моргун, Д. П. Кучеров, Р. Г. Кацалап // Вісник інженерної академії. — Вип. 4. — 2015. — С. 97–102.

5. **Möller T.** Fast, Minimum Storage Ray/Triangle Intersection / T. Möller, B. Trumbore // Journal of Graphics Tools. — Vol. 2. — №1. — 1997. — P. 21–28, doi: 10.1145/1198555.1198746 (eng)

6. **Моргун К. О.** 3D-моделювання анімованих персонажей / К. О. Моргун, Д. П. Кучеров // Наукоємні технології. — №2 (26). — 2015. — С. 133–140, doi:10.18372/2310-5461.26.8713 (eng)

7. **Моргун К. О.** Вычисление интегралов методом Монте-Карло при решении задач освещённости синтезированных объектов / К. О. Моргун, Д. П. Кучеров, Т. И. Голеньковская // Вісник інженерної академії. — Вип. 3. — 2016. — С. 82–87.

8. **Kucherov D. P.** Calculation of integrals by Monte Carlo in the illumination problem of synthesized objects / D. P. Kucherov, I. V. Ogirko, K. O. Morgun, // Science and education a new dimension. Natural and Technical Sciences. — Vol. 96. — № IV (11). — 2016. — P. 42–47.

9. **Saxena S.** Image processing tasks using parallel computing in Multi core architecture and its applications in medical imaging / S. Saxena, N. Sharma, S. Sharma // International Journal of Advanced Research in Computer and Communication Engineering. — Vol. 2. — Iss. 4. — 2013. — P. 1896–1900, doi: 10.1109/iccct.2014.7001475 (eng)

10. **Текстура (изображение).** [Електронний ресурс]. — Режим доступу: <https://ru.wikipedia.org/wiki>

11. **Kajiya J. T.** The rendering equation / J. T. Kajiya // Proceedings SIGGRAPH '86. Computer Graphics, 1986. — Vol. 20. — P. 143–150, doi: 10.1145/15886.15902 (eng)

12. **Васильева Ю. О.** Вычисление глобальной освещённости методом Монте-Карло / Ю. О. Васильева, Е. Н. Ляшенко // Світлотехніка і електроенергетика. — №3. — 2010. — С. 16–20.

13. **sRGB** [Електронний ресурс]. — Режим доступу: <https://ru.wikipedia.org/wiki/sRGB>.

14. **XYZ (цветовая модель)** — [Електронний ресурс]. — Режим доступу: [https://traditio.wiki/XYZ\\_](https://traditio.wiki/XYZ_).

**Моргун К. О.**

### **ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РЕНДЕРИНГУ 3D-ОБ'ЄКТІВ**

*У статті розглядається інформаційна технологія створення тривимірних геометричних фігур засобами комп'ютерної графіки. Інформаційна технологія розглядається як послідовний процес створення графічної тривимірної моделі, визначення геометрії цієї моделі, освітлення віртуальної сцени та збереження вихідного зображення у відповідних форматах, що містять інформацію про геометрію, а також про параметри тривимірної сцени (освітленість, текстурні координати, координати розташування камер та джерел світла). У статті наведено методи розрахунку точки перетину променя елемента поверхні об'єкта при обході променя, визначення освітлення точки, підходи до оптимізації обчислень основані на розбитті сцени під час візуалізації (рендерингу) на ділянки, що відповідають окремим потоком розрахунків. Пропонуються результати моделювання та рендерингу з використанням відомих програмних засобів.*

**Ключові слова:** трасування сцени, освітлення сцени, оптимізація розрахунку, збереження файлу, моделювання сцени.

**Моргун К. О.**

### **ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ РЕНДЕРИНГА 3D-ОБЪЕКТОВ**

*В статье рассматривается информационная технология создания трехмерных геометрических фигур средствами компьютерной графики. Информационная технология рассматривается как последовательный процесс создания графической трехмерной модели, определение геометрии этой модели, освещения виртуальной сцены и сохранения исходного изображения в соответствующих форматах, содержащих информацию о геометрии, а также о параметрах трехмерной сцены (освещенность, текстурные координаты, координаты расположения камер и источников света). В статье приводятся методы расчета точки пересечения луча элемента поверхности объекта при обходе луча, определение освещения точки, подходы к оптимизации вычислений основаны на разбиении сцены во время визуализации (рендеринга) на участки, соответствующие отдельным потокам расчетов. Предлагаются результаты моделирования и рендеринга с использованием известных программных средств.*

**Ключевые слова:** трассировка сцены, освещение сцены, оптимизация расчета, сохранение файла, моделирование сцены.

**Morhun K. O.**

### **INFORMATION TECHNOLOGY OF 3D OBJECT'S RENDERING**

*The paper considers the information technology of creating three-dimensional objects using computer graphics. Information technology is considered as a coherent process of creating a 3d model, determining the geometry of a 3d model, lighting and preserving an output image. In the paper methods of calculating the point of intersection of the camera ray of the element of the surface of the object when traversing the beam, determining the point of illumination, approaches to optimization of calculations based on the division of the scene, during visualisation, into sections that corresponds of independent flows of calculations. Modelling results presented using known software.*

**Keywords:** scene tracing, scene lighting, optimization of calculation, file save, scene simulation.

Стаття надійшла до редакції 29.08.2017 р.

Прийнято до друку 01.09.2017 р.

Рецензент – канд. техн. наук, доц. Артамонов Є. Б.