УДК 004.92

О. Різник, О. Лисак
Національний університет “Львівська політехніка”,
кафедра інформаційних технологій видавничої справи

МОДЕЛЮВАННЯ ВІДОБРАЖЕННЯ НАВКОЛИШНЬОГО СЕРЕДОВИЩА В 3D-ОБ'ЄКТАХ У РЕАЛЬНОМУ РЕЖИМІ ЧАСУ

Запропоновано вдосконалений метод візуалізації 3D-об’єктів, що мають властивість відображення на своїх поверхнях навколишнього середовища. Розглянуто процес візуалізації відбиття світла від поверхонь 3D-об’єктів за законом Ламберта та процес візуалізації відображення навколишнього середовища на поверхнях 3D-об’єктів із використанням кубічних карт. Відкоректовано вектор відбиття під час його обчислення в процесі візуалізації відбиття навколишнього середовища. Розроблено процес пост - обробки кубічних карт. Комплексне врахування цих процесів забезпечує покращення реалістичності зображень.

Ключові слова: візуалізація, моделі освітлення, методи відображення, кубічні карти.

An improved method for visualization of 3D-objects that have the ability to display on their surface environment. The process of rendering the reflection of light from the surfaces of 3D-objects using Lambert law and process visualization environment mapping on the surfaces of 3D-objects using cube maps technique. Reflection vector is corrects when it calculates in process of reflection of environment. The mechanism has been designed after processing cube map. This method guarantees the objects image quality because there is complicity usage of this processes.

Key words: visualization, lighting models, methods of display, cube maps.

Вступ

Ефектність і реалістичність зображень об'єктів залежить не тільки від того, наскільки професійно виконано моделювання, освітлення і текстурування сцен, а й від особливостей їх візуалізацій. Однією з найбільш використовуваних категорій ефектів навколишнього середовища є відображення. Існує кілька методів імітації відображень, серед яких великою популярністю користуються кубічні карти навколишнього середовища, що дають хороші результати як в статичних, так і в динамічних сценах. Реалістичність відображень, отриманих за допомогою таких карт, досягається особливим, відмінним від звичайного, процесом накладення карти навколишнього середовища на поверхню 3D-об'єкта.

Реалізація запропонованого методу розробки ефекту стала актуальною після виходу на ринок потужних графічних адаптерів з апаратною підтримкою багатьох елементів комп’ютерної графіки.

У цій роботі розглядається реалізація в реальному режимі часу кубічних карт і моделі освітлення. Її характерною особливістю є дешевий розрахунок і прийнятна реалістичність, якої достатньо для візуалізації та моделювання динамічних сцен.

Огляд літературних джерел

Процес візуалізації відбиття світла за законом Ламберта [1, 2] був запропонований у статті в'єтнамського інженера Ву Тонг Фонга [3] у 1975 р. Цей процес називається "Модель відбиття Бліна-Фонга" [4, 5]. Він є доволі популярним і поширеним у комп'ютерній графіці, попри те, що існують точніші фізичні моделі.

Для зображення поверхонь, що відбивають навколишнє середовище, використовують технологію кубічних карт [6]. Цю технологію запропонував Джим Блін у 1976 р. [7]. Обчислення вектора відбиття, який використовують як координати текстурних карт, запропонований у [8].

Такий підхід не є досконалим і потребує деякого коригування, що було реалізовано у цій статті.

Постановка задачі

Задача полягає в розробці візуального ефекту відображення навколишнього середовища на поверхнях 3D-об’єктів прикладної дизайнерської програми з метою покращення якості візуалізації та реалістичності 3D-об’єктів в ній. Програма повинна залишатись стійкою до будь-яких дій користувача та забезпечувати високу ефективність роботи в реальному режимі часу.

Процес візуалізації відбиття світла від поверхні об’єкта

Суть цієї ідеї полягає у такому: нехай задані точкове джерело освітлення, розміщене у деякій точці, поверхня, яка буде освітлюватись, і точка спостереження. Кожна точка поверхні має свої координати і в ній визначена нормаль до поверхні. Її освітлення складається із трьох компонент: рівномірне (фонове) освітлення, дифузне освітлення та освітлення відблиску. Властивості джерела визначають силу випромінення кожної з цих трьох компонент, а властивість матеріалу поверхні визначає її здатність сприймати кожен вид освітлення. Модель Бліна-Фонга розширено ще одним компонентом – самосвіченням. Цей компонент не залежить від будь-якого джерела світла і освітлює поверхню об'єкта рівномірно з визначеною інтенсивністю.

Рівномірне освітлення забезпечує постійне початкове освітлення для всієї сцени. Воно освітлює всі вершини об’єкта однаково, оскільки залежить тільки від власного кольору об’єкта та інтенсивності джерела світла у сцені.

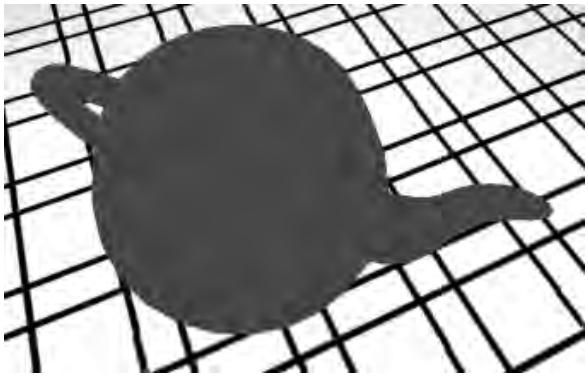


Рис. 1. Модель рівномірного освітлення

Формула для розрахунку моделі такого освітлення доволі проста. Вона містить лише добуток відповідного кольору матеріалу на кольори всіх джерел світла. Забарвлення рівномірного освітлення обчислюється за формулою

$$C_{ambient} = C_a \cdot I_a, \tag{1}$$

де C_a – власне забарвлення об’єкта, I_a – інтенсивність освітлення моделі.

Лістинг 1. Шейдер моделі рівномірного освітлення

```
float4x4 mat_mvp;           // Світова, оглядова та
проекційна матриці.
float4   ambient_color;     // Забарвлення об’єкта.
float    ambient_intensity; // Інтенсивність
освітлення.
// Вхідні дані.
struct VS_INPUT_STRUCT
{
  float4 position: POSITION;
};
// Вихідні дані.
```

```
struct VS_OUTPUT_STRUCT
{
  float4 position: POSITION;
};
// Вершинний шейдер.
VS_OUTPUT_STRUCT VertexShader (VS_INPUT_STRUCT
In_struct)
{
  VS_OUTPUT_STRUCT Out_struct;
  Out_struct.position = mul (mat_mvp,
In_struct.position);
  return Out_struct;
}
// Фрагментний шейдер
float4 main(): COLOR0
{
    // Повертаємо результат  з формули AI*AC.
    return ambient_color * ambient_intensity;
}
```

Для наступних моделей освітлення визначимо три вектори:

• *Нормаль* (N) – це вектор, що спрямований перпендикулярно до поверхні об’єкта у кожній його точці.

• *Оглядовий вектор* (V) – це вектор, напрямлений з точки огляду сцени (камери) у конкретну точку поверхні об’єкта.

• *Світловий вектор* (L) – вектор, напрямлений з точки джерела світла у конкретну точку поверхні об’єкта.

Кути між векторами впливають на інтенсивність освітлення.

Дифузна модель освітлення – це модель освітлення, яка залежить від джерела світла та від нормалі поверхні об’єкта. Оскільки випромінювання світла є однаковим у всіх напрямках, оглядовий вектор не має значення, тобто V=0. Такий метод потребує великих обрахунків, оскільки зміни відбуваються на кожній вершині об’єкта. Але він дає можливість непогано затемнювати об’єкти і надавати їм об’єму. Світло падає на об'єкт, не заповнюючи всю його поверхню однаковим кольором (як у випадку з рівномірним освітленням), а створюючи враження, що світло спрямоване на певну поверхню.

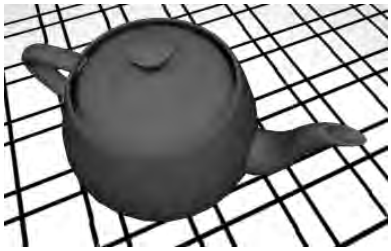


Рис. 2. Дифузна модель освітлення

Забарвлення дифузного освітлення обчислюється за формулою

$$C_{\text{diffuse}} = C_d \cdot I_d (\overline{N} \bullet \overline{L}), \quad (2)$$

де C_d – власне забарвлення об’єкта цієї моделі, I_d – інтенсивність освітлення, $\overline{N}$ – нормаль поверхні, $\overline{L}$ – напрямлений вектор від позиції джерела світла до поверхні, $\overline{N} \bullet \overline{L}$ – скалярний добуток векторів, що визначає кількість відбитого світла на конкретному фрагменті поверхні.

Лістинг 2. Шейдер дифузної моделі освітлення.

```
float4x4 mat_mvp;           // Світова, оглядова та
проекційна матриці.
float4x4 mat_world;         // Сівтова матрица.
float4   vec_light;         // Позиція джерела світла.
float    diffuse_intensity; // Інтенсивність
освітлення.
float4   diffuse_color;     // Забарвлення об’єкта
// Вхідні дані.
struct VS_INPUT_STRUCT
{
    float4 position: POSITION;
    float3 normal:   NORMAL;
};
// Вихідні дані.
struct VS_OUTPUT_STRUCT
{
     float4 position: POSITION;
     float3 normal:   TEXCOORD1;
};
// Вершинний шейдер.
VS_OUTPUT_STRUCT VertexShader (VS_INPUT_STRUCT
In_struct)
{
    VS_OUTPUT_STRUCT Out_struct;
    // Рахуємо позицію вершини.
    Out_struct.position =
mul(mat_mvp,In_struct.position);
    //Рахуємо нормаль поверхні і зберігаємо для
фрагментного шейдера.
    Out_struct.normal   =
normalize(mul(mat_world,In_struct.normal));
    return Out_struct;
}
// Фрагментний шейдер
float4 main(): COLOR0
{
    return diffuse_color * diffuse_intensity *
dot(normal, vec_light);
}
```

Модель освітлення відблиском. Важко уявити таку модель освітлення, не побачивши її. Насправді, вона трапляється доволі часто. Прикладом може слугувати чисто відполірована пряма металева поверхня, на яку напрямлене джерело світла, і на яку дивляться під кутом, не перпендикулярним до її поверхні. У результаті можна побачити відблиски на поверхні, які істотно збільшують реалістичність зображення. Ці відблиски є результатом відбиття джерела світла від поверхні.

У такій моделі освітлення, крім векторів позиції джерела світла і нормалі (як у випадку з дифузною моделлю), використовується ще два вектори: оглядовий вектор і вектор відображення.

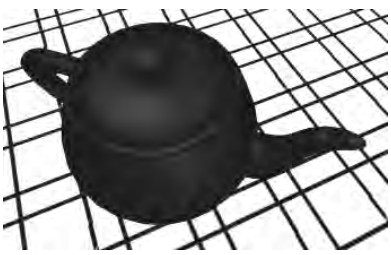


Рис. 3. Модель освітлення відблиском

Формула визначення забарвлення відблиском має вигляд:

$$C_{specular} = C_s \cdot I_s (\overline{L} \bullet \overline{R})^n, \quad (3)$$

де C_s – власне забарвлення об'єкта моделі, I_s – інтенсивність освітлення моделі, $\overline{V}$ – оглядовий вектор, $\overline{L}$ – вектор джерела світла, $\overline{R}$ – відбитий від поверхні вектор, що обчислюється за формулою

$$\overline{R} = (-\hat{V}) - 2 \cdot \overline{N} \cdot ((-\hat{V}) \bullet \overline{N}), \quad (4)$$

$$\hat{v} = \frac{\overline{v}}{\|\overline{v}\|}. \quad (5)$$

Лістинг 3. Шейдер моделі освітлення відблиску

```
float4x4 mat_mvp;             // Світова, оглядова та
проекційна матриці.
float4x4 mat_world;           // Сівтова матрица.
float4   vec_light;           // Позиція джерела світла.
float3   vec_view;       // Оглядовий вектор
float    specular_intensity; // Інтенсивність
освітлення.
float4   specular_color;      // Забарвлення об'єкта
// Вхідні дані.
struct VS_INPUT_STRUCT
{
    float4 position: POSITION;
    float3 normal:   NORMAL;
};
// Вихідні дані.
struct VS_OUTPUT_STRUCT
{
     float4 position: POSITION;
     float3 normal:   TEXCOORD1;
};
// Вершинний шейдер.
VS_OUTPUT_STRUCT VertexShader (VS_INPUT_STRUCT
In_struct)
{
    VS_OUTPUT_STRUCT Out_struct;
    // Рахуємо позицію вершини.
    Out_struct.position =
mul(mat_mvp,In_struct.position);
    //Рахуємо нормаль поверхні і зберігаємо для
фрагментного шейдера.
    Out_struct.normal   =
normalize(mul(mat_world,In_struct.normal));
    return Out_struct;
}
// Фрагментний шейдер
float4 main(): COLOR0
{
    float power = 16;
    float3 reflect_vec = reflect(-normalize(vec_view),
In.normal);
    return specular_color * specular_intensity
           * pow(dot(reflect_vec, vac_light), power);
}
```

Модель освітлення самосвіченням є найпростішою із наведених моделей. Вона визначається більшим значенням кольору матеріалу та сумою кольорів усіх моделей. Цю модель застосовують для об'єктів, що мають властивість самосвічення, таких як джерела світла.

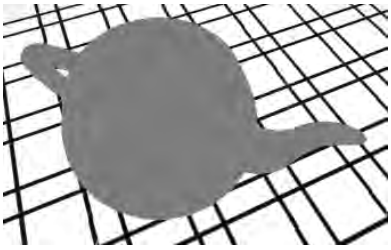


Рис. 4. Модель освітлення самосвіченням

Визначення забарвлення самосвіченням

$$C = \max(C_{emissive}, C_e) \tag{6}$$

Комбінування моделей освітлення.

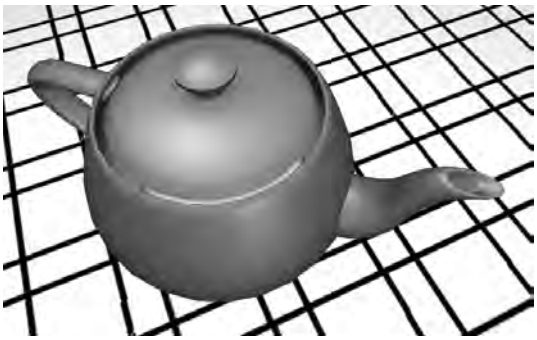


Рис. 5. Комбінація чотирьох моделей

Після обчислення кожної з моделей освітлення об'єкт отримує забарвлення за формулою

$$C = \max(C_{emissive}, (C_{ambient} + C_{diffuse} + C_{specular})) \tag{7}$$

Процес візуалізації відображення навколишнього середовища на поверхнях 3D-об'єкта із використанням кубічних карт

Попередній обрахунок Кубічних карт. Для зображення на поверхні об'єкта інших об'єктів, що оточують його, тобто відображення навколишнього середовища, використовується технологія кубічних карт. Кубічні карти складаються з шести різних текстур (рис. 6). Кожна із цих текстур являє собою промальовану сцену з місця розташування об'єкта, для якого генерується ця карта, у певному напрямку.

Рис. 6. Кубічна карта деякого навколишнього середовища

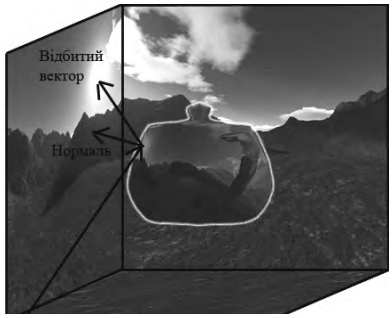


Рис. 7. Наочний приклад ефекту відбиття навколишнього середовища

Обчислюють відбитий вектор за формулою

$$\overline{R} = \hat{P} - 2 \cdot \overline{N} \cdot (\hat{P} \bullet \overline{N}), \tag{8}$$

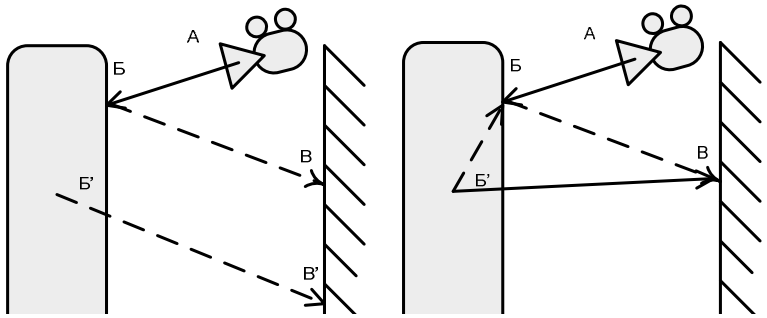
де $\overline{P}$ – позиція об’єкта, що відображає навколишнє середовище; нормалізація цього вектора ($\hat{P}$) здійснюється згідно з формулою (5); $\overline{N}$ – нормаль поверхні об’єкта; $\overline{T}$ – відбитий від поверхні об’єкта вектор, який одночасно і є координатами кубічної текстури для конкретного фрагменту поверхні об’єкта.

Обчислення текстурних координат для кубічних карт відбувається множенням оберненої матриці спостереження, що є одним із параметрів ефекту, на відбитий вектор:

$$\overline{T} = \overline{R} \times V^{-1}, \tag{9}$$

де V^{-1} – обернена матриця спостереження.

Корекція відбитого вектора. Після обчислення відбитого вектора можна отримати забарвлення в кожній точці поверхні об’єкта, що відбиває навколишнє середовище. Цей ефект можна застосовувати для об’єктів будь-яких геометричних форм. Проте такий алгоритм одержання тривимірних координат буде ідеальним тільки для об’єктів, форма яких близька до сфери. У решти випадках, особливо коли об’єкт має дещо плоску форму, цей алгоритм обчислюватиме спотворене відображення навколишнього середовища. Проблема полягає в тому, що вираховується вектор, який прямує з позиції огляду сцени (А) в конкркетну точку поверхні об’єкта (Б) та підставляється у кубічну текстуру. Текстура обраховувалась в точці центру об’єкта (Б'), а не у поточній позиції поверхні (Б), де обчислюється цей вектор. Отриманий вектор буде дещо неправильно визначати координати (В'), за якими вибиратиметься забарвлення.

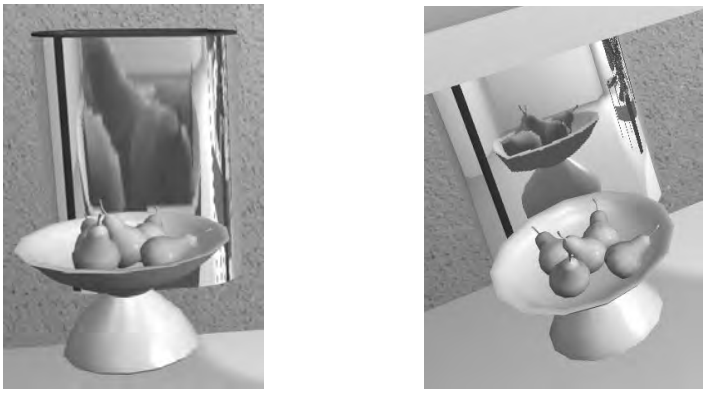


а *б*

Рис. 8. Зображення відбитого вектора (а) та його коригування (б)

Однак є просте вирішення цієї проблеми за допомогою застосування кількох операцій із векторми. Коригують відбитий вектор за формулою

$$\overline{БВ`} = (А - Б') + \overline{АБ} \tag{10}$$



а *б*

Рис. 9. Відображення навколишнього середовища у разі некоригованого (а) та коригованого (б) векторів

Комплексне врахування процесів

Для обрахунку кінцевого вигляду поверхні застосовують як обчислення поверхні непрозорого об'єкта, так і обчислення поверхні об'єкта, що відбиває навколишнє середовище. А їх вплив на забарвлення поверхні встановлюється властивістю типу поверхні, що має назву *інтенсивність відбиття*. Чим більше значення цієї опції, тим більша поверхня об'єкта матиме відбитий, а не власний, колір і навпаки.

Рис. 10. Об'єкти з різною величиною відбиття.
Кожний наступний об'єкт має більшу за попередню величину відбиття

Для візуалігації 3D-об'єктів, що відбивають на своїй поверхні навколишнє середовище з неабсолютною величиною відбиття, розроблено механізм післяопрацювання кубічної карти. Під час виконання цього механізму, попередньо згенерована кубічна карта піддається фільтру розмиття. Величина розмиття регулюється властивістю глянцевості.

Опробація та застосування методу

Запропонований метод успішно інтегровано в комерційний програмний продукт ViSoft Premium версії 2010-1. Він призначений для проектування та 3D-візуалізації ванних кімнат. Серед його переваг можна виділити широкі можливості проектування ванних кімнат, використання великої бази даних продукції європейських виробників сантехніки та плитки, вивід на друк креслень та даних про кімнату, що проектується.

Недоліком ViSoft Premium вважалась дещо застаріла 3D-візуалізація кімнати та об'єктів у ній.

Розроблений метод покращив якість продукту та зробив ViSoft Premium конкурентоспроможнішим. Зразки застосування методу для хромованих і керамічних поверхонь наведені на рис. 11 та 12.

Рис. 11. Використання методу
для візуалізації хромованих поверхонь

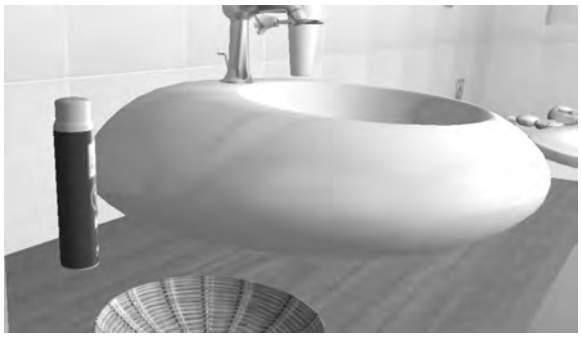


Рис. 12. Використання методу
для візуалізації керамічних поверхонь

Висновки

Запропонований у роботі вдосконалений метод розробки ефекту відображення навколишнього середовища на поверхнях 3D-об'єктів забезпечує контроль та якість процесу проектування. Ефективність оцінюється на кожному етапі проектування з погляду глибини аналізу вхідних параметрів. Цей метод рекомендується використовувати для оцінювання можливості розв'язання поставленої задачі у разі обмеження часу проектування та фіксованої глибини аналізу параметрів.

1. J. D. J. Ingle and S. R. Crouch, Spectrochemical Analysis, Prentice Hall, New Jersey (1988). 2. Houghton, J.T. The Physics of Atmospheres 2nd ed. Chapter 2. 3. G. Bisob, D. Weirner, Fast Phong Shading, SIGGRAPH 86, Computer Graphics, v20, n4, cm527-536, 1986. 4. Закон Ламберта. Модель отражения Фонга. Модель отражения Блинна-Фонга (www.compgraphics.info) 5. Программирование шейдеров на HLSL. Модели освещения. (www.gamedev.ru) *6. Кубическая текстура. (www.ru.wikipedia.org) 7. Blinn K., Models of Light Reflection for Computer Sythesized Pictures, SIGGRAPH 77, Computer Graphics, v11, n4, cm273-536, 1977. 8. HDRCubeMap Direct3D Sample (www. msdn.microsoft.com).*