

АЛГОРИТМ МЕТОДУ ТИМЧАСОВО ПРОСТОРОВИХ ПЕРЕПОН “ЕКРАННОГО ПРОСТОРУ”

Анотація: У поточному часі пошук шляхів покращення алгоритмів глобального освітлення ведеться по багатьом напрямкам. В даній статті виділяється напрямок просторових перепон “екранного простору”. Вибір саме цього напрямку пов’язаний з широким використанням даного алгоритму в сучасних системах 3 вимірних візуалізацій в реальному часі.

Ключові слова: глобальне освітлення, просторові перепони “екранного простору”.

Вступ

У теперішній час обчислення глобального освітлення стало безкомпромійною вимогою до систем візуалізації 3 вимірних сцен реального часу. Для обчислення цього ефекту найчастіше використовується деякий різновид алгоритму просторових перепон “Екранного простору” (SSAO). Однак, щоб зберегти можливість обчислення даного алгоритму в реальному часі, необхідні деякі поступки і обмеження кількості зразків. В наслідок цього генероване глобальне освітлення характеризується наявністю поверхневого шуму, який може бути зменшений в пост обробці за допомогою перивчастого фільтру. В залежності від обраного фільтру можна зберегти різкість зображення, погодившись з наявністю деякого шуму або усунути шум, але отримати розмите зображення. Крім того для динамічно рухомих об’єктів шум може плавати на поверхні, що доволі відволікає. В даній статті описується рішення, яке дозволяє позбутися шуму отримавши при цьому чітке зображення, як в статичних сценах, так і в динамічних.

З фізичної точки зору, оклюзію оточення можна розглядати, як дифузне освітлення від неба [5]. Таким чином АО поверхневої точки p з нормаллю n_p обчислюється як [1]:

$$AO(p, n_p) = \frac{1}{\pi} \int V(p, \omega) D(|p - \xi|) n_p \omega d\omega, \quad (1)$$

де ω означає всі напрямки на півкулі і V є (зворотна) двійкова функція видимості, з $V(p, \omega) = 1$, якщо видимість в цьому напрямку блокується перешкодою, 0 в іншому випадку. D є монотонною спадною функцією від 1 до 0 на відстані від p до ξ , точка перетину з найближчою поверхні.

У простому випадку, D є ступінчастою функцією, з врахуванням перешкод в певному радіусі вибірки.

Методи SSAO намагаються апроксимувати оригінальний інтеграл AO у просторі екрана. Враховуючи існування декількох версій SSAO з різними припущеннями і компромісами.

Будемо вважати, що будь-який метод SSAO можна записати як середнє внесків C , які залежать від ряду зразків:

$$AO_n(p) = \frac{1}{n} \sum_{i=1}^n C(p, s_i). \quad (2)$$

Для того, щоб апроксимувати рівняння (1) використовуючи метод Монте-Карло, представимо відповідну функцію SSAO у вигляді запропонованому Рітшелем [9]:

$$C(p, s_i) = V(p, s_i) D(|s_i - p|) \max(\cos(s_i - p, n_p), 0). \quad (3)$$

На відміну від рівняння (1) напрямки були замінені реальними точками вибірки навколо p , отже $V(p, s_i)$ – можна вважати бінарною функцією видимості, яка дорівнює 0 якщо вона видима із точки p і дорівнює 1, якщо ні.

Такий метод називається – просторові перепони “екранного простору”, тому, що V обчислюється, шляхом перевірки видимості s_i на екранному просторі в залежності від буферу глибини (z-буферу). Варто звернути увагу на те, що деякі реалізації алгоритму SSAO не виконують тест глибини, і значення s_i вираховується за рахунок відстані і падаючого кута [3].

В запропонованій реалізації модифікованого алгоритму зразки s_i задалегідь вираховані і збережені в текстурі, яка являє з себе набір трьох вимірних точок, рівномірно розподілених в півкулі, які перетворюються на дотичний простір p для розрахунку C . Якби цей крок був опущений, то довелось би брати зразки з усієї сфери і значення s_i було в 2 рази більшим.

D є функцією відстані до точки зразка, яка може використовуватися для модуляції спаду.

Можна запропонувати наступний модифікований алгоритм SSAO, отже загальними кроками якого є:

1. генерація SSAO;
2. уточнення SSAO з плином часу;
3. виявлення помилкових оклюзій;
4. просторова фільтрація;

Уточнення SSAO з плином часу

Основною ідеєю модифікованого алгоритму є поширення розрахунку AO на декілька кадрів за рахунок використання повторної проекції. Кожного разу коли є можливість використання повторної

проекції пропонується рішення з попереднього кадру, яке відповідає пікселю кадру і уточнити його з урахуванням значень нових зразків, вирахованих в поточному кадрі.

У кадрі f обчислюється нове значення C_f від k нових зразків – повторна проекція:

$$C_f(p) = \frac{1}{k} \sum_{i=j_f(p)+1}^{j_f(p)+k} C(p, s_i). \quad (4)$$

де $j_f(p)$ кількість унікальних зразків, що вже були використані в даному рішенні.

Далі поєднується нове рішення з раніше обчисленим:

$$\begin{aligned} \omega_{f-1}(p_{f-1}) AO_{f-1}(p_{f-1}) + kC_f(p) \\ (\omega_{f-1}(p) + k), \\ AO_f(p) = \\ \omega_f(p) = \min(\omega_{f-1}(p_{f-1}) + k, \omega_{\max}), \end{aligned} \quad (5)$$

де вага ω_{f-1} кількість зразків, що вже були заздалегідь вираховані або заздалегідь визначений максимум після досягнення збіжності.

Теоретично, цей підхід може використовувати, як завгодно багато зразків. Але на практиці, це не рекомендовано, оскільки повторна проекція не є точною і потребує білінійної фільтрації для реконструкції, кожен крок повторної проекції вносить помилки, які поширюються з часом. Це проявляє себе у вигляді зростаючого розмиття зображення.

Крім того вплив тільки розрахованих зразків зводиться до нуля, і старі зразки ніколи не замінюються. Більш того виникає необхідність обмеження ω_f до заданого користувачем межового значення ω_{\max} , що з часом зменшує вплив старих зразків.

Таким чином,

$$\text{conv}(p) = \omega_f(p) / \omega_{\max}. \quad (6)$$

Зауваження, що до реалізації:

Значення ω_{f-1} зберігається в окремому каналі в буфері навколишньої оклюзії. Для досягнення швидкої збіжності використовується послідовність Халтона, яка відома своєю низькою невідповідністю [8], для генерації зразків.

Як, початковий індекс запропоновано використання значення j_f яке теж зберігається в буфері навколишньої оклюзії.

Таким чином, ціль RGBA з буфера навколишнього середовища зберігає наступні параметри:

- Розв'язання рівняння SSAO $C_{f-1}(p)$
- Вагу з попереднього рішення ω_{f-1}
- Початковий індекс j_{f-1}

- Глибина лінії очей d_{f-1}

Положення поточного індексу поширюється на наступний кадр за допомогою повторної проєкції, як і в SSAO, для того щоб запобігти інтерполяції початкового індексу апаратним забезпеченням і відхиленню значення в послідовності. Важливо завжди отримувати значення індексу із центру найближчого пікселя в буфері навколишньої оклюзії.

Центр пікселя може бути знайдений за допомогою розрахунку рівняння (7), з огляду на наявність координат повторної проєкції:

$$pixelcenter = \frac{[tex_{f-1}res_{x,y}] + 0,5}{res_{x,y}}. \quad (7)$$

Виявлення помилкових оклюзій

Перевірка на помилкові оклюзії проводиться шляхом порівняння поточної глибини фрагмента d_f з глибиною позиції фрагменту d_{f-1} збереженої в кеші [6]:

$$\left| 1 - \frac{d_f}{d_{f-1}} \right| < e. \quad (8)$$

Обґрунтованість зразка s_i для зафарбування пікселя p може бути оцінена шляхом обчислення зміни відносного положення зразка та пікселя:

$$\delta(s_i) = ||s_i - p| - |s_{if-1} - p_{f-1}||. \quad (9)$$

Просторова фільтрація

Методи SSAO застосовують просторову фільтрацію після розрахунку зафарбування для уникнення шумових артефактів пов'язаних з недостатньою частотою дискретизації. Тому в даній роботі теж використовується просторова фільтрація, але тільки до тих пір поки частота часової когерентності недостатня.

Застосовуються також варіанти поперечного двостороннього фільтра [2], там де фільтрація на краях не проводиться, шляхом врахування розходження глибини.

Зважаючи на те, що формально цей фільтр не використовується для роздільної фільтрації, в реальному часі він зазвичай окремо фільтрує координату x і y . На відміну від інших реалізацій в запропонованому підході є додаткова інформація, яка дозволяє суттєво зменшити зашумленість. Більше того в наданому алгоритмі ядро фільтру застосовується безпосередньо, до відстаней світових координат:

$$AO_{filt}(p) = \frac{1}{K(p)} \sum_{x \in F} g(|p-x|) conv(x) AO(x),$$

$$K(p) = \sum_{x \in F} g(|p - x|) \text{conv}(x). \quad (10)$$

Зі сходженням пікселів можна зменшити вплив фільтру використовуючи фактор зменшення s :

$$s(p) = \frac{\max(c_{\text{adaptive}} - \text{conv}(p), 0)}{c_{\text{adaptive}}}. \quad (11)$$

Впроваджені модифікації основного алгоритму дозволяють підвищити кількість кадрів в секунду і якість зображення завдяки збільшеній точності.

Випадковий шум. Як і в більшості підходів АО можна повертати шаблон зразків за допомогою випадкового вектора для кожного введеного пікселя. Ці маніпуляції обмінюють смугові артефакти на підвибірки для шуму. Однак це призводить до не очікувано великого підвищення продуктивності [7]. Більше того слід відключати обертання, як тільки сходження досягло граничного значення c_{rot} ;

Часовий простір. Щоб уникнути зменшення точності у великих сценах, необхідно зберігати значення позицій в місцевих координатах, з центром в поточній точці огляду. Ці значення можуть бути легко перетворені до світових координат, шляхом передачі попередньої та поточної точки огляду в якості параметрів шейдера.

Адаптивні вибірки

Хоча просторова фільтрація може зменшити шум, більш ефективним методом є забезпечення додаткових екземплярів вибірки. Або якщо сказати іншим чином, коли АО досягнув значного сходження, можна використати вже враховані значення, тим самим зменшивши кількість використаних зразків. Слід зазначити, що ці зразки відрізняються від зразків використаних при фільтрації. Рекомендовані параметри для використання в алгоритмі наведені в (табл. 1).

Таблиця 1

Рекомендовані параметри для алгоритму TSSAO

Ім'я параметру	Значення
Початкові зразки k_1	32
Конвергентні зразки k_2	8-16
Поріг C_{adaptive}	0.2
Поріг C_{spatial}	0.3
Поріг C_{rot}	0.5
Гладкий недійсний фактор	15-30
Максимальна вага ω_{max}	500-1500
Ширина фільтру F	5*5

Реалізація TSSAO

Дана реалізація заснована на методах Фокса \ Комптона [3] і Рітшеля [5].

Алгоритм Рітшеля

Він використовує 3 вимірне ядро вибірки і тест глибини для запиту видимості зразків, роблячи свій внесок в рівняння (3). Оригінальний алгоритм запропонований Crytek [4], не використовує випадковий кут для підрахунки ваги зразку. Для того, щоб отримати лінійний SSAO спад необхідно використовувати розподіл вибірки, що є лінійним по радіусу вибірки сфер. Слід зазначити що для реалізації використовується постійна функція спаду $D(x)=1$, через те, що спад викликається тільки через розподіл опорних точок. Відмінності від трасування променів в своїй основі викликані лише дискретизацією сцени екранного простору.

Алгоритм Фокса/Комптона

Цей алгоритм для затінення бере зразки для буфера глибини навколо пікселя і інтерпретує ці зразки, як невеликі плями, подібно до алгоритму Radiosity. Хоча алгоритм може бути фізично не точним, дуже часто він видає дуже гарний результат, за рахунок того, що зберігає малі деталі при використанні великих ядер. З іншого боку даний алгоритм має схильність до виявлення теселяції. Для реалізації алгоритму використовувалося рівняння:

$$C(p, s_i) = \frac{\max(\cos(s_i, p, n_p), 0)}{\max(e, |s_i - p|)}. \quad (12)$$

Основна відмінність від інших методів SSAO, полягає в тому що кожен зразок побудований на видимій поверхні, і інтерпретується як пластир, тоді як у рівнянні (12), зразки використовуються для оцінки функції видимості.

Генерування зразків s_i із 2 вимірних зразків Халтона виконується за допомогою наступного рівняння:

$$r = \sqrt{\zeta_y}, s_i = (r \cos(\alpha), r \sin(\alpha)), \quad (13)$$

$$\alpha = 2\pi\zeta_x,$$

$$r = \zeta_y.$$

Для методу Рітшеля запропоновано використовувати трьох вимірні зразки, створені в дотичному просторі поверхневого простору. З трьох вимірних зразків Халтона $\zeta_{x,y,z}$, напівсферичні трьох вимірні зразки створюються за допомогою використання рівняння:

$$s_i = (r \cos(\alpha) \sqrt{r - \zeta_z}, r \sin(\alpha) \sqrt{r - \zeta_z}, \sqrt{r - \zeta_z}). \quad (14)$$

Висновки

В роботі впроваджено алгоритм SSAO який використовує повторну проекцію, і тимчасову когерентність, для створення високо якісного ефекту АО для динамічних сцен. Запропонований алгоритм багаторазово використовує інформацію зразків з попередніх кадрів, якщо вона доступна, в той самий час генеруючи нові зразки і проводячи просторову фільтрацію тільки в тих місцях, де було виявлено недостатню кількість зразків. Також був реалізований ефективно новий тест валідації пікселів для алгоритмів зафарбовування, які впливають тільки на сусідні пікселі. Використання реалізованого алгоритму дозволяє отримати суттєву вигоду з тимчасової повторної проекції і в динамічних сценах.

Бібліографічний список

1. Robert L. Cook. "A Radiance Model for Computer Graphics."/ Robert L. Cook, Kenneth E. Torrance // ACM Trans. Graph. — 1982 — Vol.1. — P. 7–24.
2. Elmar Eisemann. "Flash Photography Enhancement via Intrinsic Relighting."/ Elmar Eisemann, Frederic Durand // ACM Trans. Graph. — 2004 — Vol. 23. — P. 673–678.
3. Megan Fox. "Ambient Occlusive Crease Shading."/ Megan Fox, Stuart Compton // Game Developer Magazine — 2008.
4. M. Mitrting. "Finding Next Gen – CryEngine 2." // ACM SIGGRAPH — 2007 — P. 97–121.
5. Tobias Ritschel. "Approximating Dynamic Global Illumination in Image Space."/ Tobias Ritschel, Thorsten Grosch, Hans-Peter Seidel // Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games — 2009 — P. 75–82.
6. Daniel Scherzer. "Pixel-Correct Shadow Maps with Temporal Reprojection and Shadow Test Consistency."/ Daniel Scherzer, Stefan Jeschke, Michael Wimmer // Proceedings of the Eurographics Symposium on Rendering — 2007, P. 45–50.
7. Niklas Smedberg. "Rendering Techniques in Gears of War 2."/ Niklas Smedberg, Daniel Wright // Proceedings of the Game Developers Conference — 2009.
8. Xiaoqun Wang. "Randomized Halton Sequences."/ Xiaoqun Wang, Fred J. Hickernell. // Mathematical and Computer Modelling — 2000 — P. 887–899.

Отримано 17.10.2013 р.