Дослідження в математиці і механіці. – 2023. – Т. 28, вип. 1–2 (41–42). – С. 77–102

УДК 519.6(045)

А. Л. Максимов, С. В. Прокоф'єва
Одеській національний університет ім. І. І. Мечнікова

ПІДХІД ДО ЕФЕКТИВНИХ МЕТОДІВ РОЗВ'ЯЗАННЯ БАГАТОМІРНИХ ЕЙКОНАЛЬНИХ РІВНЯНЬ

У статті представлено розширений метод розв'язування ейконального рівняння в чотиривимірному просторі зі слабкими деформаціями. Ейкональне рівняння, поєднує хвильову оптику з геометричною оптикою та має різні фізичні інтерпретації, включаючи задачи пошуку найкоротших шляхів та обчислення електромагнітних або гравітаційних потенціалів. Запропонований метод розширює техніку трассування сфери до просторів багатьох вимірів з деформаціями і продемонстровано для задачі в просторі чотирьох вимірів. Метод використовує неявні функції для опису границь об'єктів, що побудовані з скінченого або нескінченого числа багатомірних примітивів. Нелінійне трассування сфери досягається генерацією на кожному кроці трассування звичайних (багатомірних) диференціальних рівнянь першого порядку з використанням гібрідного методу розв'язання, що поєднує метод Ейлера, коли сфера знаходиться близько до границі, з методами вищого порядку, коли сфера знаходиться далеко від границь. Вплив нелінійних перетворень на процес трасування реалізується за допомогою матриці Якобі деформації. Підхід реалізовано як шейдерну программу на мові GLSL, а вплив нелінійних перетворень визначається за допомогою параметра перетворення, який впливає на матрицю Якобі. Обчислювальна продуктивність методу оцінюється через середню та максимальну частоти кадрів для різних значень параметра. Запропонований підхід може знайти застосування в таких галузях, як комп'ютерна графіка, часозалежна комп'ютерна томографія, сейсмічна томографія та астрофізичне моделювання, оптимальне керування.

MSC: 39-08, 53-08, 78M34.
Ключові слова: ейконал, рівняння, деформація, трассування, матриця Якобі, метод Ейлера, шейдер, GLSL.
DOI: https://doi.org/10.18524/2519-206X.2023.1-2(41-42).305260.

Вступ

У поданій статті розглянуто питання побудови моделі та ефективного розв'язання окремого випадку ейконального рівняння у чотиривимірному просторі (який за бажання можна інтерпретувати як три виміри + час) для випадку, коли у цьому просторі є слабкі деформації (нелінійності).

Ейкональне рівняння - це нелінійне диференціальне рівняння, що виводиться із рівнянь Максвелла і пов'язує хвильову оптику з геометричною

Надійшла 21.03.2023
© Максимов А. Л., Прокоф'єва С. В., 2023

оптикою. Це рівняння має декілька фізичних інтерпретацій, серед яких задача пошуку найкоротшого шляху та електромагнітний потенціал.

Досліджуваний випадок ейконального рівняння можна розглядати з точки зору завдань варіаційного обчислення, де ми можемо інтерпретувати деформований простір з границями як чотиривимірний многовид, в якому найкоротші (геодезичні) лінії не є прямими. У лінійному випадку (відсутність деформації) це рівняння можна розв'язати геометрично та ефективно запрограмувати. Такий підхід потребує відносно низьку обчислювальну потужність, тому застосовується в багатьох сферах, наприклад, у комп'ютерній графіці. Як правило, ейкональне рівняння використовується для 3-вимірного простору. Однак у разі деформованого простору розмірності більше 2, загальновизнаний ефективний метод розв'язання невідомий.

Області застосування представленого підходу — спеціалізована комп'ютерна графіка, часозалежні комп'ютерна томографія, сейсмографія в геології, моделювання космічних нелінійних просторово-часових ефектів в астрофізиці, оптимальне керування.

Основні результати

1. Крайова задача для ейконального рівняння. Нелінійну крайову задачу для ейконального рівняння можна сформулювати так:

задано область Ω і функцію $F : \Omega \to \mathbb{R}_+$, яка представляє локальний опір руху (наприклад, показник оптичного заломлення). Ця функція керує системою від початкового стану $\mathcal{P}_s \subset \Omega$ до цільового стану $\mathcal{P}_g \subset \delta\Omega$ з мінімально можливим значенням u (наприклад, відстані або часу).

Завдання — знайти розв'язок $u(p)$ ейконального рівняння:

$$|\nabla u(p)| = F(p), \tag{1}$$

$$F(p) > 0 \ \text{ for } \ p \in \Omega \subset \mathbb{R}^N,$$

$$u(p) = 0 \ \text{ for } \ p \in \mathcal{P}_g$$

Після розв'язання $u(p)$ представляє поле відстаней (або часів прибуття), що містить значення (наприклад, час), необхідне для переходу від будь-якої точки p до найближчої точки в $\mathcal{P}_s$ за умови місцевого опору $F(p)$ руху.

Існує варіант рішення $\overline{u}(p)$, що доповнює $u(p)$, і яке представляє поле відстані, що містить значення, необхідні для переходу від будь-якої точки p до найближчої точки в $\mathcal{P}_g$ за зворотним локальним опором $F(p)$ руху.

2. Дискретний підхід до розв'язування ейконального рівняння

На сьогоднішній день існує багато алгоритмів розв'язування ейконального рівняння для випадку 2d дискретного кодування нелінійностей (а в деяких випадках і для 3d сіток). Серед них можна виділити наступні популярні алгоритми: FMM (Fast Marching Method) - метод швидкого маршу, UFMM (Untidy Fast Marching Method)- "брудний"метод швидкого маршу, GMM (Group Marching Method)- груповий метод швидкого маршу, LSM (Lock Sweeping Method)- метод разблокуючого замітання , DDQM (Double Dynamic Queue Method)- метод подвійної динамічної черги [7]. Ці алгоритми використовуються в таких галузях, як комп'ютерний зір, медична томографія, геологія тощо. Загальна їх базова риса - обчислювальна динаміка (хвильовий метод) на регулярних сітках в просторах малих розмірів.

Дискретні підходи показують досить хороші результати для особливих випадків - малорозмірних (2D і, набагато рідше, 3D) просторів, де необхідно обирати потрібні алгоритми для кращої продуктивності. Однак цей підхід має кілька проблем - чим вище число розмірності простору, тим істотніше збільшується час обчислення алгоритмів. Крім того, навіть у просторі малої розмірності, але у якому відсутні явні межі простору, дискретний підхід не працює просто тому, що неможливо зберігати таку кількість вузлів сітки у пам'яті комп'ютера.

3. Використання функції знакової відстані SDF для рівняння ейконала

Поле відстані зі знаком (SDF) для рівняння ейконала є неявним представленням граничних умов і може представляти поле потенціалу для простору без явних меж. Такі поля дозволяють розв'язувати завдання у постановці (1) навіть з нескінченними просторами. Збільшення кількості вимірів простору не погіршує продуктивність так сильно, як при дискретному підході. Крім того, дослідження [11] демонструє можливість роботи з частково деформованими просторами.

Якщо Ω є підмножиною метричного простору P з метрикою d, тоді функція відстані зі знаком f визначається як (2) та іллюструється на рис.

1.

$$f(p) = \begin{cases} d(p, \partial\Omega) & p \in \Omega \\ -d(p, \partial\Omega) & p \in \Omega^c \end{cases}, \tag{2}$$

де $\partial\Omega$ позначає межу Ω, і відстань $d(p, \partial\Omega) = \inf_{p_0 \in \partial\Omega} d(p, p_0)$.

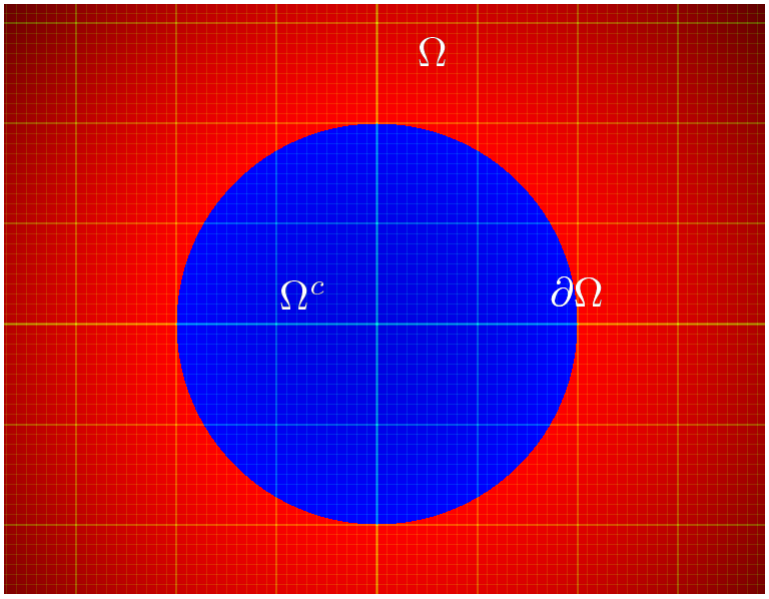

Рис. 1: Границя $\partial\Omega$ і підмножини Ω, Ω^c метричного простору P для визначення функції відстані зі знаком.

Якщо Ω є підмножиною $\mathbb{R}^n$ з кусково гладкою границею, то функція відстані зі знаком диференційована майже всюди, а для евклідової метрики її градієнт задовольняє рівнянню

$$|\nabla f| = 1 \tag{3}$$

У випадку тривимірного простору формула (3) має такий вигляд:

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial z}\right)^2} = 1 \tag{4}$$

Якщо межа $\partial\Omega$ знаходиться в C^k для $k \geq 2$, то $d(p)$ знаходиться в C^k у точках, досить близьких до межі $\partial\Omega$ [3]. Зокрема, на межі $\partial\Omega$ функція $f(p)$ задовольняє

$$\nabla f(p) = N(p) \tag{5}$$

де N — нормальне векторне поле (відносно до межі). Цей факт можна використовувати для терміналізації та оцінки плавності розв'язків в алгоритмах [13].

Алгоритми обчислення функції відстані зі знаком можуть бути, наприклад, загальними методами рівнів для підмножин простору. Будь-яка неявна функція $\Psi(p)$ задовольняє (2), але не завжди задовольняє (3). Але можна перетворити $\Psi(p)$ на $\tilde{f}(p)$, що задовольняє (2) і добре наближається до (3) використовуючи

$$\tilde{f}(p) = \frac{\Psi(p)}{|\nabla\Psi(p)|}, \tag{6}$$

4. Методи розв'язування рівняння ейконала за допомогою SDF

4.1. Маршування променя. Для розв'язання рівняння ейконала для полів відстаней в однорідному просторі можна використовувати метод маршування променя. З певної початкової позиції в багатовимірному просторі промінь просувається до тих пір, поки він не потрапить на граничну умову (гіперповерхню складеного багатовимірного об'єкта).

Якщо граничні умови неявно виконані за межами поля, то широко використована технологія розв'язання — це маршування променя від стартовоі точки в полі SDF доти, поки граничні умови не будуть виконані [14]. Точніше, треба знайти точку на промені $\mathbf{p} : \mathbb{R} \to \mathbb{R}^n$, де $\mathbf{p}(\mathbf{s}) = \mathbf{p_0} + s\,\mathbf{n}$, яка в той же час лежить на границі $\partial\Omega$, що представлена неявним рівнянням $f(p) = 0$. У цій нотації s - це відстань (скаляр) від стартовоі точки, і $\mathbf{n}$ це нормальний вектор променя.

Інформація про цю точку на поверхні $\partial\Omega$ може бути використана для подальших обчислень (нормаль, кривизна, і т.д.). Точку перетину променю з поверхнею $\partial\Omega$, заданою неявною граничною умовою, можна обчислити шляхом застосування методів знаходження коренів, що в загальному випадку є нетривіальною математичною задачею. Це було причиною розробки алгоритму маршування променя Перліном та Хоффертом [15]. Наприклад, точку перетину з неявною границею можна знайти шляхом обчислювання значення поля з фіксованими кроками, доки не буде досягнуто значення, близького до нуля.

Цей підхід можна проілюструвати псевдокодом:

```
Вибрати фіксований крок step > 0
Вибрати малий параметр eps > 0
  Для заданого одиничного вектора n:
     point = vector(0);
      while(SDF(samplingPoint) > eps)
            point += step * n;
```

4.2. Трасування сфер Техніка маршу променю має одну головну проблему, а саме фіксований розмір кроку. Якщо розмір занадто малий, тоді алгоритму знадобиться занадто багато часу, щоб досягти точки на граничній умові. Якщо розмір кроку занадто великий, на марші променю можна пропустити значення, близьке до нуля (граничну умову!). Трасування сфери є розширенням алгоритму маршування променю, у якому розмір кроку не є фіксованим, а динамічно змінюється вздовж променя, встановлюючи для нього значення з даних поля відстані. Це гарантує, що промінь не пропустить граничну умову, оскільки поле SDF повертає відстань до найближчої граничної умови (яка може не лежати на промені!). У той же час промінь може проходити через порожній простір швидше [14].

Це налаштування розміру кроку розв'язує проблему перевищення кроку під час маршу променів. Алгоритм не може пропустити малі межі, оскільки розмір кроку буде зменшено поблизу межі, і промінь ніколи не переступить через саму межу. Трасування сфер названо на честь необмежених сфер, створених уздовж променя.

У цьому алгоритмі кількість кроків залежить від початкової точки променя. Наприклад, якщо шлях променя паралельний границі, то розмір кроку не зростає на цій ділянці шляху. Це може призвести до значних витрат на обчислення. Тому необхідно визначити мінімальний поріг розміру кроку $0 < \epsilon << 1$ і максимальну кількість кроків.

Псевдокод для трасування сфери:

```
Вибрати малий параметер eps > 0
Для заданого одиничного вектора n:
    point=vector(0);
     step = SDF(point);
      while(abs(step) > eps)
```

```
step = SDF(point);
  point += step * n;
```

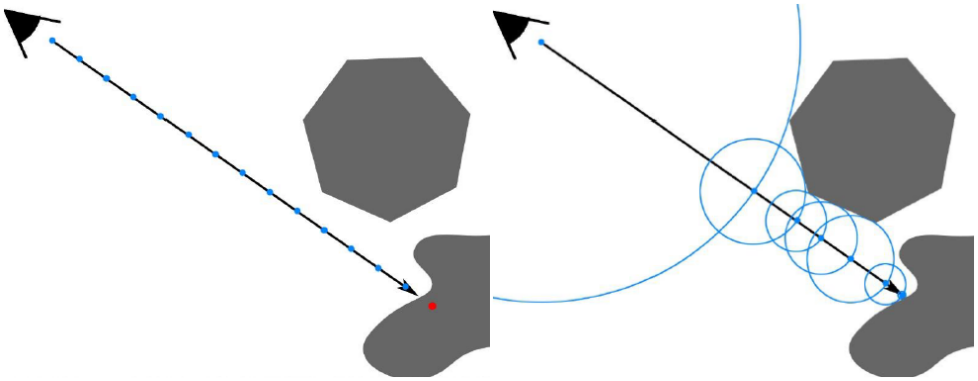

Рис. 2: Алгоритми маршування променю (ліворуч) і трасування сфери (праворуч).

5. Розв'язування ейконального рівняння за допомогою SDF в просторі з нелінійними перетвореннями

Нелінійне перетворення можна виразити як деформацію $D(p)$.

Розв'язання

$$f(D^{-1}\mathbf{p}(s)) = 0 \tag{7}$$

- поширений підхід до введення малих деформацій D у SDF. Цей підхід дозволяє кожен крок для трасування сфери з нелінійним перетворенням переписати так, щоб під час крокування вздовж променя в деформованому просторі кожна точка перетворювалася назад у недеформований простір, щоб знайти мінімальну відстань до найближчої (інверсно-деформованої) границі. Кожен крок буде виглядати так:

$$\mathbf{p}_{i+1} = \mathbf{p}_i + |f(D^{-1}\mathbf{p}(s))|\mathbf{n}. \tag{8}$$

Тут $\mathbf{n}$ нормований вектор напрямку променю.

Однак існує серйозне обмеження через складність обчислення D^{-1}. Це обчислення має бути доступним і ефективним. D^{-1} можна легко обчислити для афінних і деяких нелінійних перетворень, але це, як правило, складна операція. Крім того, має бути виконана умова $||\nabla(f \circ D^{-1})|| \leq 1$, щоб задовольнити умову використання трасування сфери. Проблема в тому,

що f повертає відстань у недеформованому просторі, але промінь крокує у деформованому просторі. Якщо перетворення D є неперервними за Ліпшицем і асоційована константа Ліпшица $\lambda \geq ||\nabla(f \circ D^{-1})||$ відома, тоді можна крокувати відповідно до $|f(\mathbf{p}_i)|/\lambda$. Для більш складних деформацій, ніж афінні перетворення, константи Ліпшица може бути важко або неможливо обчислити аналітично [11].

Всі труднощі викликані тим, що промінь трасується в деформованому просторі, а відстань обчислюється в недеформованому. У разі трасування в недеформованому просторі немає необхідності враховувати межі Ліпшица. Однак прямий промінь у деформованому просторі деформується в деяку криву в недеформованому просторі

Згідно з [12] для деформування параметричних кривих, можна переписати рівняння променя в деформованому просторі як інтеграл:

$$\mathbf{p}(s) = \mathbf{p}(0) + s\mathbf{n} = \mathbf{p}(0) + \int_0^s \mathbf{n}\, d\sigma = \mathbf{p}(0) + \int_0^s \mathbf{p}'(\sigma)\, d\sigma. \tag{9}$$

В інтегралі $\mathbf{p}'(\sigma) = \mathbf{n}$ є першою похідною $\mathbf{p}(s)$ і є константою в недеформованому просторі. Щоб отримати рівняння променя в деформованому просторі, необхідно застосувати зворотну деформацію D^{-1} до попереднього рівняння:

$$\widehat{\mathbf{p}(s)} = D^{-1}(\mathbf{p}(0)) + \int_0^s J_D^{-1}(\mathbf{p}(\sigma))\mathbf{n}\, d\sigma. \tag{10}$$

Досить обчислити якобіан $J_{D^{-1}}$ зворотної деформації D^{-1} всередині інтеграла. Згідно з теоремою оберненої функції, якобіан зворотної деформації в деякій точці $\mathbf{p}$ є оберненим якобіаном прямої деформації у відповідній точці $\widehat{\mathbf{p}}$. У цьому випадку якобіан є $n \times n$ матрицею. Після параметризації інтеграла (10) на довжину дуги в недеформованому просторі ми отримаємо інтеграл променя

$$\widehat{\mathbf{p}}(\widehat{s}) = D^{-1}(\mathbf{p}_0) + \int_0^{\widehat{s}} \widehat{\mathbf{n}}(\widehat{\mathbf{p}}(\sigma))\, d\sigma \tag{11}$$

де

$$\widehat{\mathbf{n}}(\widehat{\mathbf{p}}) = \frac{J_D^{-1}(\widehat{\mathbf{p}})\mathbf{n}}{||J_D^{-1}(\widehat{\mathbf{p}})\mathbf{n}||} \tag{12}$$

Параметр $\widehat{s}$ і поле відстані зі знаком $f(\widehat{\mathbf{p}})$ визначені в одному метричному просторі. Але все одно потрібно один раз обчислити D^{-1} для початкової точки $\mathbf{p_0}$ променя в деформованому просторі.

5.1. Задачі для необмеженого нелінійного ейконального рівняння

Нескінченні періодичні або квазіперіодичні функції можна використовувати в багатьох областях. Наприклад, для опису поверхні моря, архітектурних споруд тощо. Основною проблемою необмеженого нелінійного рівняння ейконала є великий обчислювальний час, отже, погана продуктивність. SDF може розв'язувати необмежене ейконального рівняння завдяки обчислюванні по модулю (операція mod()) без різкого зниження продуктивності.

6. Метод розв'язування необмеженого нелінійного ейконального рівняння Трасування сфери з деякими доповненнями може бути використано для розв'язування необмеженого нелінійного ейконального рівняння.

6.1. Базові функції та операції відстані зі знаком Трасування сфери засноване на використовує функції відстані зі знаком. Для кожного базового примітиву [17] має існувати функція відстані зі знаком. Функції (як точні, так і наближені) можна розширити для роботи з n-вимірним евклідовим метричним простором.

Формула (6) дає наближення першого порядку функції відстані зі знаком, яка є чудовою поблизу границі та хорошою для асимптотики для великих відстаней [18]. Однак використання формули (6) може погіршити поведінку деяких алгоритмів, заснованих на градієнті функції відстані зі знаком. Обчислювальні експерименти показують хорошу евристику для таких ситуацій. Вона складається з використання формули (6) для оцінки відстані до межі та водночас із використанням нормалізованого градієнта $|\nabla\Psi(p)|$ як хорошого наближення справжнього градієнта поля відстані.

Для певного класу базових фігур було розроблено хороші формули наближення як для відстані, так і для градієнта. Наприклад, використовують (14), тобто модифікацію (6) для узагальнених 2-вимірних кіл (13) (див. рис. 3):

$$\Psi(x,y)_l = x^l + y^l - r^l \tag{13}$$

$$\tilde{f}(p) = \frac{\Psi(p)}{|\nabla\Psi(p)|_q}, \tag{14}$$

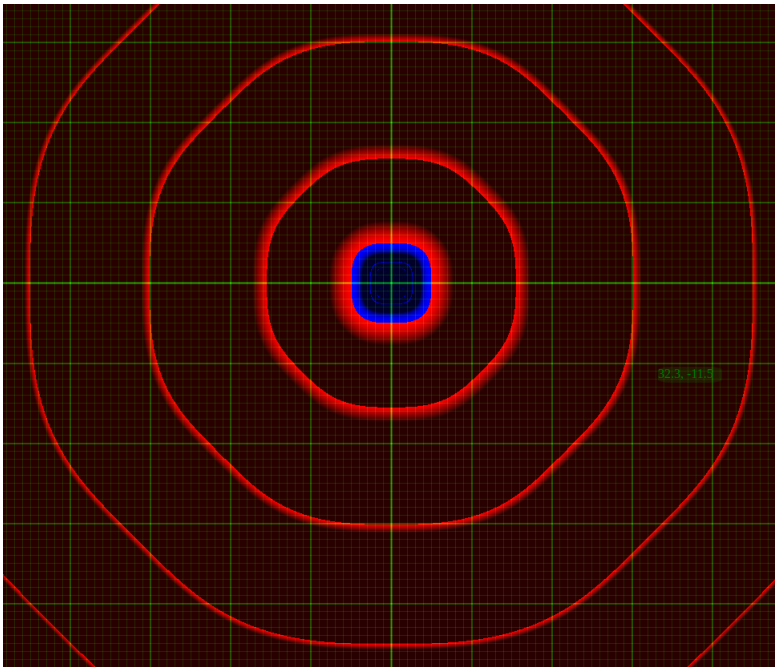

Рис. 3: Апроксимовано полем відстані (6) для узагальнених кіл.

де $|.|_q$ – метрика Мінковського з $q = 1 - 2/l$ (Fig. 4)

Точна формула (15) для знакового поля відстані сфери є уніфікованою для будь-якої кількості вимірів. Тут $\mathbf{p}$ - точка в n-мірному просторі, r - радіус сфери.

$$\text{sdSphere}(\mathbf{p}, r) = |\mathbf{p}/r| - 1 \tag{15}$$

Формула (16) для еліпсоїда наближена. Тут $\mathbf{p}$ - точка в n-мірному просторі, r - значення великих півосей еліпсоїда. Операція ./ - це паралельне ділення компонентів вектора чисельника на компоненти вектора знаменника.

$$\text{sdEllipsoid}(\mathbf{p}, \mathbf{r}) = (|\mathbf{p}./\mathbf{r}| - 1); \tag{16}$$

Найпростіша операція об'єднання реалізована як мінімальна відстань до найближчої межі:

$$\text{sd}_1(\mathbf{p}) \bigcup \text{sd}_2(\mathbf{p}) = \min(\text{sd}_1(\mathbf{p}), \text{sd}_2(\mathbf{p})) \tag{17}$$

Операція (18) повторення використовується для створення нескінчен-

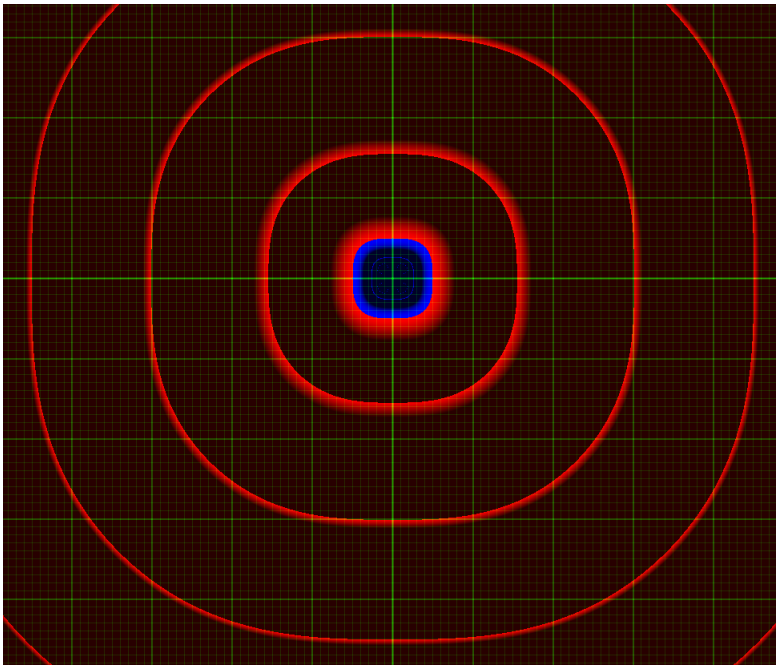

Рис. 4: Виправлене на (14) поле відстані для узагальнених кіл.

них меж із заданим періодом c:

$$\text{opRep}(\mathbf{p}, c) = \mathbf{p} \mod c - 0.5 \cdot c \tag{18}$$

7. Розширення нелінійного трасування сфери

Розширення алгоритму нелінійного трасування сфери можна записати як

$$\widehat{\mathbf{p}}_{i+1} = \widehat{\mathbf{p}}_i + |f(\widehat{\mathbf{p}}_\mathbf{i})|\widehat{\mathbf{n}}(\widehat{\mathbf{p}}_i) \tag{19}$$

Однак цей підхід дає велику похибку, якщо промінь сильно викривлений (рис. 5 a). Зменшення розміру кроку покращує точність, але погіршує продуктивність (рис. 5 b). Тому було запропоновано обчислювати підкроки без використання SDF для мінімальної відстані до найближчої межі (рис. 5 c). Кожен крок i від $\widehat{\mathbf{p}}_\mathbf{i}$ до $\widehat{\mathbf{p}}_{\mathbf{i+1}}$ можна розглядати як звичайне диференціальне рівняння першого порядку $\mathbf{y}'(\widehat{\mathbf{p}}) = \widehat{\mathbf{n}}(\widehat{\mathbf{p}})$ з початковою умовою $\mathbf{y_0} = \widehat{\mathbf{p}}_\mathbf{i}$.

Тоді наступний крок можна обчислити як $\widehat{\mathbf{p}}_{\mathbf{i+1}} = R(\widehat{\mathbf{p}}_\mathbf{i}, f(\widehat{\mathbf{p}}_\mathbf{i}))$, де R — будь-який розв'язувач звичайного диференціального рівняння, який приймає початкову умову та мінімальну відстань до найближчого об'єкта як

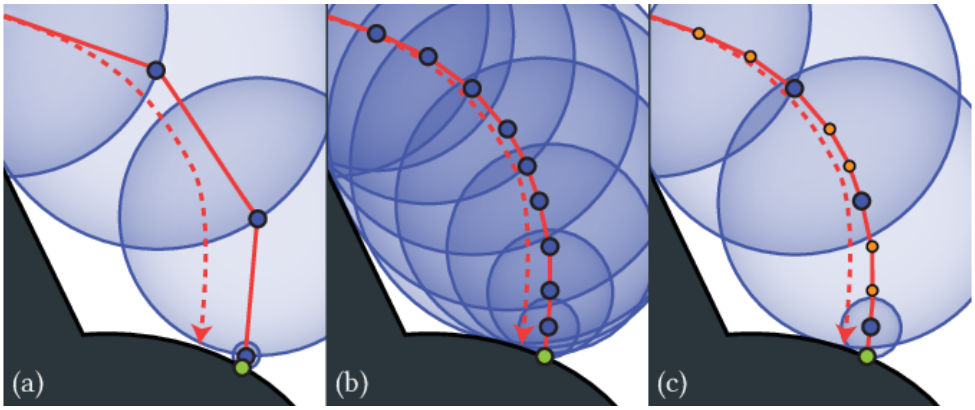

Рис. 5: З сильно вигнутим променем розмір кроку може бути занадто великим для хорошої точності (a). Хоча збільшення кількості кроків покращує точність, це негативно впливає на продуктивність обчислень (b). Представлений метод додає невелику кількість кроків і дає таку ж точність (c).

параметри. Вибраний розв'язувач може ділити $f(\widehat{\mathbf{p}}_\mathbf{i})$ на підмножини, щоб досягти заданого допустимого відхилення, але йому не потрібно перераховувати $f(\widehat{\mathbf{p}})$. Завдяки параметризації довжини дуги будь-який правильний розв'язувач гарантує, що кожна підмножина залишається всередині сфери навколо $\widehat{\mathbf{p}}_\mathbf{i}$ (рис. 6) і, таким чином, задовольняє умову відстеження сфери [11].

Розв'язувач звичайних диференціальних рівнянь слід вибирати ретельно, оскільки він сильно впливає на продуктивність обчислень. Композиції деформацій сильно змінюють промінь у деяких локаціях простору. Крім того, чим ближче промінь до границі, тим меншою стає мінімальна відстань $f(\widehat{\mathbf{p}})$. У цьому випадку простіший інтегратор Ейлера, такий як (19), часто є достатньо хорошим. Уніфікований розв'язувач або не зможе правильно обчислювально відтворити промінь, або, для дуже частого простого випадку, коли промінь незначно змінюється, розв'язувачу знадобиться багато часу обчислення.

Гібридний підхід полягає у використанні інтегратора Рунге-Кутта, коли $f(\widehat{\mathbf{p}})$ є відносно великим у порівнянні з $\widehat{\epsilon}$. У цій роботі розглядалися наступні розв'язувачі ОДУ: метод інтегрування Ейлера, метод 4-го порядку Хойна та Рунге-Кутта.

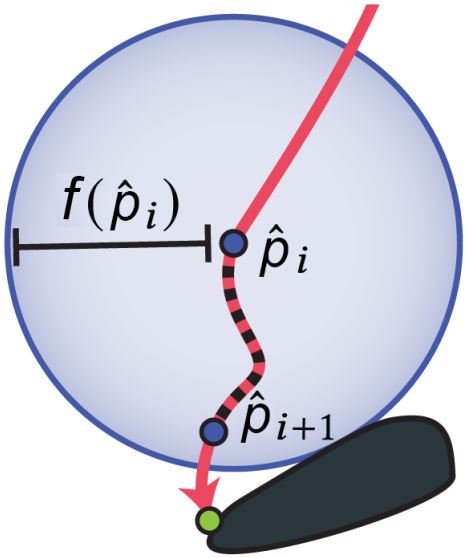

Рис. 6: Додаткові підкроки.

Інтегратор Ейлера — чисельна процедура першого порядку для розв'язання звичайних диференціальних рівнянь із заданим початковим значенням. У випадку трасування сфери ми могли б зробити це все ще крокуючи вздовж променя в деформованому просторі, трансформуючи кожну точку назад у недеформований простір, щоб обчислювати поле відстані. Коли промінь знаходиться близько до межі та мінімальна відстань менша за $a \cdot \widehat{\epsilon}$, де a — це порогове значення, вибране користувачем, достатньо простого інтегрування Ейлера. Формула інтегратора Ейлера виглядає наступним чином:

$$\hat{\mathbf{p}}_{\mathbf{i+1}} = \hat{\mathbf{p}}_{\mathbf{i}} + h \cdot \mathbf{rd}(\hat{\mathbf{p}}_{\mathbf{i}}) \tag{20}$$

де,

- $\mathbf{rd} = \nabla f(D \cdot \mathbf{p}_i)$ – нормалізована функція напрямку променя, змінювана нелінійною деформацією D ,
- $\hat{\mathbf{p}}_{\mathbf{i}}$ - це точка в просторі об'єкта,
- $h = |\hat{\mathbf{p}}_{\mathbf{i}} - \hat{\mathbf{p}}_{\mathbf{i-1}}|$ - це параметр кроку метода,

Метод Хойна або метод середнього нахилу дає більш точне наближення, ніж інтегратор Ейлера. Він дає явну формулу для обчислення y_{i+1}.

Основна ідея полягає у виправленні деяких помилок оригінального методу Ейлера[20]:

$$y_{i+1} = y_i + h(\frac{1}{2}k_1 + \frac{1}{2}k_2) \tag{21}$$

де $k_1 = g(x_i, y_i)$, $k_2 = g(x_i + h, y_i + h \cdot k_1)$.

Модифікована версія методу Хойна для нелінійного трасування сфери:

$$\mathbf{\hat{p}_{i+1}} = \mathbf{\hat{p}_i} + h(\frac{1}{2}\mathbf{k_1} + \frac{1}{2}\mathbf{k_2}) \tag{22}$$

де $\mathbf{k_1} = \mathbf{rd}(\mathbf{\hat{p}_i})$, $\mathbf{k_2} = \mathbf{rd}(\mathbf{\hat{p}_i} + h \cdot \mathbf{k_1})$.

Метод четвертого порядку Рунге-Кутта найчастіше використовується для розв'язування ОДУ. Потім за допомогою ітераційної формули [21] обчислюється наближене значення в наступних точках:

$$\begin{aligned} y_{i+1} &= y_i + \Delta y_i \\ \Delta y_i &= \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \tag{23}$$

де
$k_1 = g(x_i, y_i)$,
$k_2 = g(x_i + \frac{h}{2}, y_i + \frac{hk_1}{2})$,
$k_3 = g(x_i + \frac{h}{2}, y_i + \frac{hk_2}{2})$,
$k_4 = g(x_i + h, y_i + h \cdot k_3)$.

Цей метод має четвертий порядок точності. Це означає, що помилка на одному кроці становить $O(h^5)$, а загальна помилка за кінцевий інтервал інтегрування становить $O(h^4)$.

Модифікована версія кроків для класичного методу Рунге–Кутта для нелінійного трасування сфери: [21]:

$$\mathbf{\hat{p}_{i+1}} = \mathbf{\hat{p}_i} + \Delta\mathbf{\hat{p}_i}, \tag{24}$$

де $\Delta\mathbf{\hat{p}_i} = (\mathbf{k_1} + 2\mathbf{k_2} + 2\mathbf{k_3} + \mathbf{k_4})/6$,
$\mathbf{k_1} = \hat{\mathbf{rd}}(\mathbf{\hat{p}_i})$,
$\mathbf{k_2} = \hat{\mathbf{rd}}(\mathbf{\hat{p}_i} + h \cdot \mathbf{k_1}/2)$,
$\mathbf{k_3} = \hat{\mathbf{rd}}(\mathbf{\hat{p}_i} + h \cdot \mathbf{k_2}/2)$,
$\mathbf{k_4} = \hat{\mathbf{rd}}(\mathbf{\hat{p}_i} + h \cdot \mathbf{k_3})$.

Algorithm 1 Гібрідний алгоритм трассування променю

Require: $r0, rd, maxd$
  $p \leftarrow ro + rd * t$
  for $i = 0$ to $i = RAY_STEP$ **do**
    if $abs(h) \geq epsilon$ **OR** $t \leq maxd$ **then**
      break
    end if
    $res \leftarrow map(p)$
    $h \leftarrow res.x$
    if $abs(h) < epsilon * a$ **then**
      $p = euler(rd, h)$
    else
      $p = R(rd, h)$
    end if
    $t+ = h$
  end for
  if $t >= maxd$ **then return** $(t, -1)$
  else return $(t, 1)$
  end if

8. Алгоритмічна реалізація та тестування

8.1 Алгоритмічна реалізація

Основою реалізації є гібрідний алгоритм трассування променю, який трасує промінь у заданому напрямку.

Гібрідний алгоритм трасування променя приймає початкову точку променя `r0`, напрямок променя `rd` і максимальну довжину променя `maxd`. Максимальна довжина використовується для економії часу на обчислення, якщо промінь не може досягти межі на своєму шляху. Для першого кроку промінь залишається прямим, щоб можна було знайти наступну точку. Наступними кроками є цикл, доки довжина променя `t` не перевищує максимальну довжину `maxd` або відстань до найближчої точки `h` менше заданого порогу `epsilon`. Всередині циклу спочатку перевіряється, якщо відстань `h` до найближчої точки менше, ніж порогове значення `epsilon*a`, то використовується інтегратор Ейлера для скорочення часу обчислення. Якщо відстань до найближчої границі досить велика, то ця відстань використовується в ODE розв'язувачі `R`, щоб знайти наступну точку на промені. Тоді `h` додається до довжини променя `t`. Алгоритм повертає `t` та -1, якщо `t` більше максимальної довжини `maxd`, інакше 1 .

Розв'язувач ODE використовує нормалізований вектор на кожному кроці. Щоб знайти його, спочатку обчислюється обернена матриця Якобі `inverseJacobian`, а потім за допомогою оберненої матриці модифікується напрямок променя `rd` і далі нормалізується вектор результату:

Algorithm 2 Алгоритм отримання модифікованого нормалізованого вектору

Require: $jacobian, rd, point$
  $inverseJacobian \leftarrow inverse(jacobian)$
  $normVector \leftarrow normalize(inverseJacobian * rd)$
  return $normVector$

8.2. Реалізація та тестування

Програма трасування сфери в 4-вимвірному просторі була реалізована як шейдер з використанням мови GLSL. Шейдер використовує для візуалізації графічний процесор. Усі обчислення проводилися на графічному процесорі Nvidia GeForce GTX 1060. Чотиривимірні об'єкти в процесі візуалізації проектувалися спочатку у тривимірний простір зі спеціально

підібраними кольорами та освітленням, а потім стандартні методи відображення обчислювали двовимірні кадри тривимірних проекцій. Візуалізація обчислень дозволяє відстежувати (і навіть відчувати) час обчислення, а також досить легко контролювати помилки обчислень.

Трасування сфери було реалізовано з параметром, який визначає ефект від нелінійного перетворення. Було розглянуто 4 значення цього параметру: 0, 0,2, 0,45, 0,65. Коли параметр дорівнює 0, то нелінійне перетворення взагалі не впливає (рис. 7). З параметром 0,2 важко побачити різницю між лінійним і нелінійним трасуванням сфери (рис. 8). Параметри 0,45 і 0,6 дають більшу деформацію границям (рис. 9 і 10).

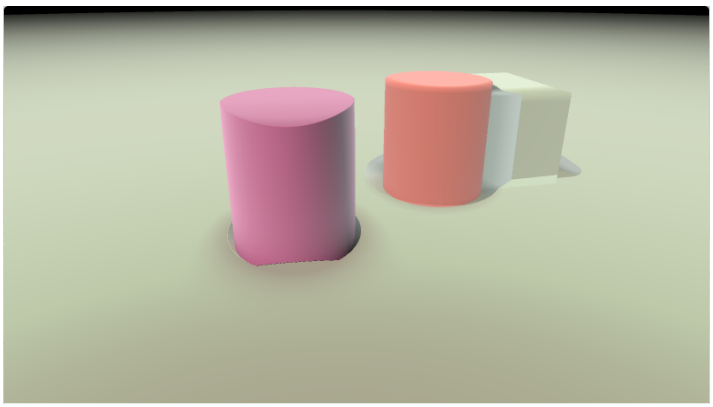

Рис. 7: Лінійне трасування сфери

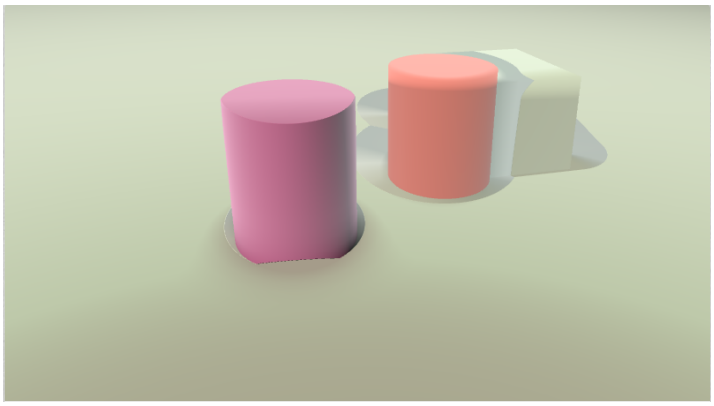

Рис. 8: Нелінійне трасування сфери з параметром = 0.2

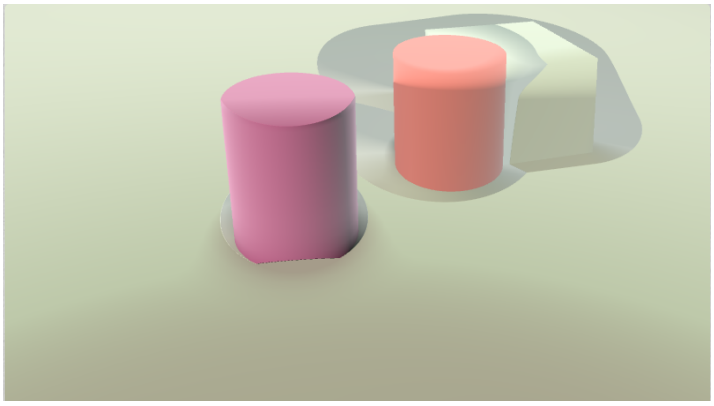

Рис. 9: Нелінійне трасування сфери з параметром = 0.45

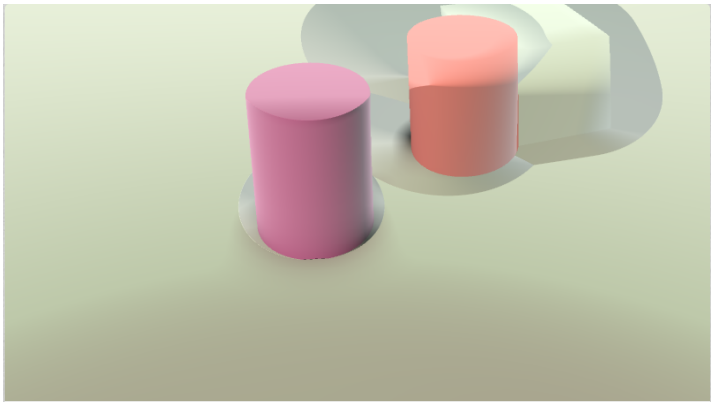

Рис. 10: Нелінійне трасування сфери з параметром = 0.65

Результати представлені в таблиці 1. Для кожного значення параметру розраховувалися середній FPS (кадрів в секунду) і максимальний FPS. Для трасування лінійної сфери як середній, так і максимальний FPS досить високі, це забезпечує плавну якість відео. При нелінійній трасуванні сфери середній FPS значно знижується. Чим більше значення параметра, тим нижчий середній FPS. Це впливає на продуктивність шейдера, що створює враження перегляду слайд-шоу. Слід зазначити, що обчислення виконувались для чотиривимірного простору, для якого стандартні розв'язувачі видавали би результати за час на порядки більше (якби вони взагалі могли бути застосовані).

	Parameter			
	0.0	0.20	0.45	0.65
Avg FPS	51.3	20.6	16.4	12.9
Max FPS	60.1	31.9	27.1	22.5

Табл. 1: Трасування сфери з різним параметром нелінійного перетворення.

Зміни в візуалізації однакові для нескінченних меж. Починаючи з параметру = 0,45, зміни стають більш помітними (рис. 11-14).

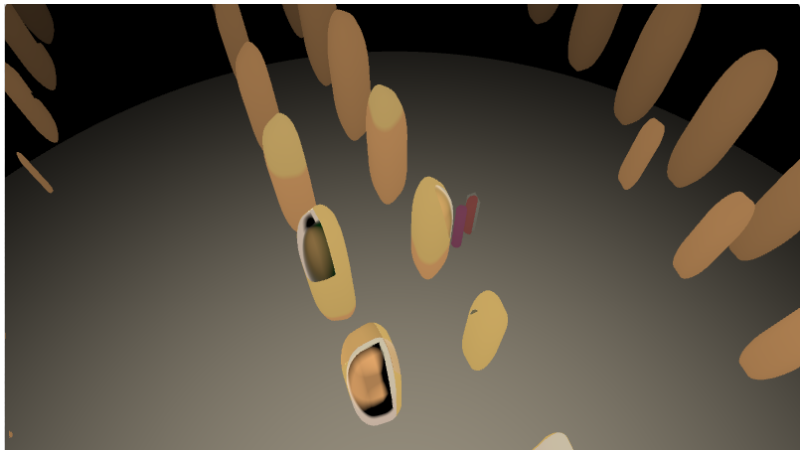

Рис. 11: Лінійне трасування сфери

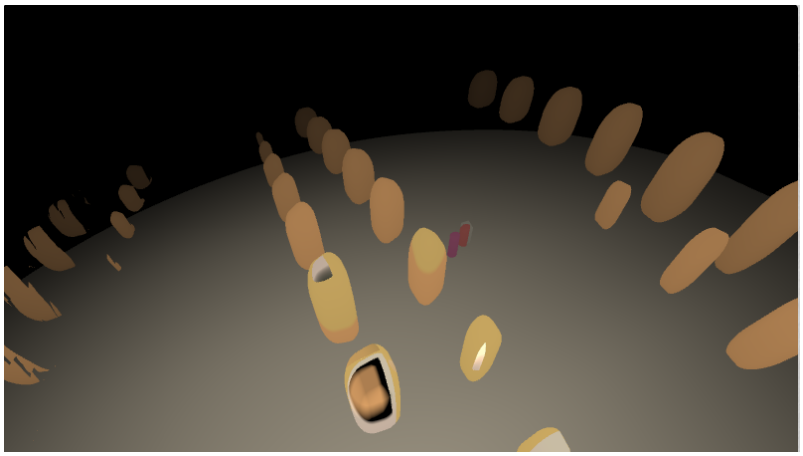

Рис. 12: Нелінійне трасування сфери з параметром = 0.2

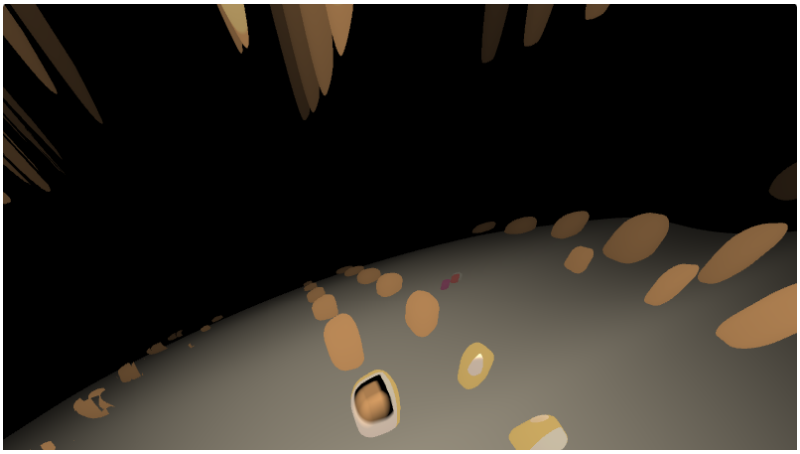

Рис. 13: Нелінійне трасування сфери з параметром = 0.45

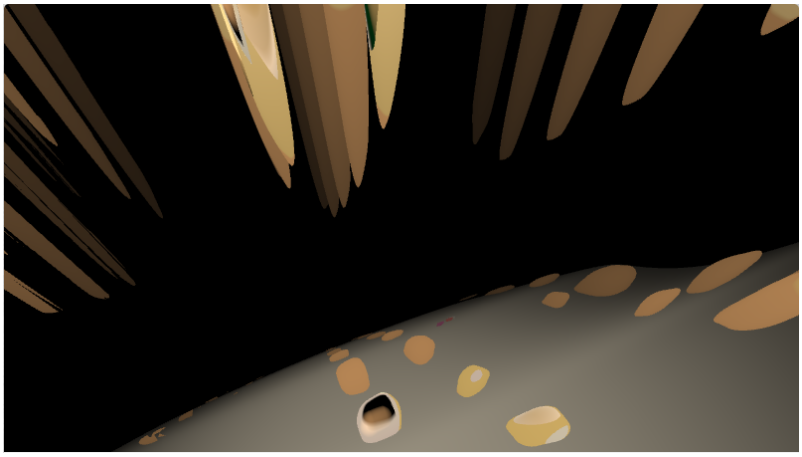

Рис. 14: Нелінійне трасування сфери з параметром = 0.65

Середній FPS і максимальний FPS підраховувалися також для меж нескінченного повторення з однаковими параметрами нелінійного перетворення - 0, 0,2, 0,45 і 0,65 (Таблиця 2). Середній FPS для трасування лінійної сфери з нескінченними періодичними межами вдвічі нижчий, ніж для нескінченних меж. Застосування нелінійного перетворення також сильно знижує FPS.

	Parameter			
	0.0	0.20	0.45	0.65
Avg FPS	29.6	6.9	6.1	5.8
Max FPS	30.1	9.1	8.6	8.3

Табл. 2: Sphere tracing for infinite periodic boundaries.

Висновки

У рамках цієї роботи був розглянутий частковий випадок ейконального рівняння та проаналізовані існуючі методи його розв'язання. Також був розглянутий розширений метод трасування сфер (sphere tracing) для нелінійних трансформацій. У роботі запропоновано спосіб використання розширеного способу для 4-мірного простору та продемонстровано на практиці.

Для розширення методу трасування сфер до 4-мірного використовуються відповідні вдосконалені версії неявних функцій, які описують примітивні границі.

Для нелінійного трасування сфер на кожному кроці ітерації розглядається крайова задача, де кожен крок є звичайним диференціальним рівнянням першого порядку з початковою умовою результату попереднього кроку. Диференціальне рівняння виглядає як функція від точки, в якій перераховується значення нормального вектора. Для цього використовується добуток зворотньої матриці Якобі та вектора, який задає направлення променю. Універсальний підхід не є доцільним, оскільки виконує забагато обчислень у випадку, коли промінь знаходиться доволі близько до границі або не може відтворити викривлення променю достатньо точно. Тому було обрано змішаний підхід – за замовчуванням використовувати будь-який доволі точний метод розв'язання звичайних диференціальних рівнянь та у випадку, коли відстань між границею та променем є доволі малою – використовувати метод Ейлера.

Розширений метод трасування сфер був реалізований як шейдер на мові GLSL. Для матриці Якобі використовувався спеціальний параметр, який задає вплив нелінійної трансформації на промінь. Для різних значень цього параметра було визначено середню та максимальну кількість кадрів

за секунду під час роботи шейдеру.

Список літератури

1. Javaid M. 4D printing applications in medical field: A brief review / M. Javaid, A. Haleem // Clinical Epidemiology and Global Health. — 2019. — V.7, №3. — P. 317—321.
2. Kwong Y. Four-dimensional computed tomography (4DCT): A review of the current status and applications / Y Kwong., A. O. Mel, G. Wheele, J. M. Troupis // Journal of Medical Imaging and Radiation Oncology. — 2015. — P. 545—554.
3. Polacci M. Crystallisation in basaltic magmas revealed via in situ 4D synchrotron X-ray microtomography / M. Polacci, F. Arzilli, G. La Spina, N. Le Gall, B. Cai, M. E. Hartley, D. Di Genova, N. T. Vo, S. Nonni, R. C. Atwood, E. W. Llewellin, P. D. Lee, M. R.Burton // Scientific Reports. — 2018.
4. Irmak E. Concept of 4th Dimension for Databases / Irmak E., Kurtuldu . // 14th International Conference on Machine Learning and Applications (ICMLA). — 2015. — P. 1159—1162.
5. Raytracing 4D fractals, visualizing the four dimensional properties of the Julia set [Електронний ресурс]. — 2008. — Режим доступу до ресурсу: \http://www.codinginstinct.com/2008/11/raytracing-4d-fractals-visualizing-four.html
6. Prokofieva S. Methods for solving partial types of the eikonal equation / S. Prokofieva // Madrid: Barca Academy Publishing, Madrid, Spain, 2021. — P. 192—194.
7. Sethian, J. A. A fast marching level set method for monotonically advancing fronts / J. A.Sethian // Proceesings of the National Academy of Sciences. — 1969. — V.93, №4. — P. 1591—1595.
8. Gmez J. V. Fast Methods for Eikonal Equations: An Experimental Survey / J. V.Gmez, S. Garrido, D. Alvarez, L. E. Moreno // IEEE Access. — 2019.
9. Dijkstra E. A note on two problems in connection with graphs / E. Dijkstra // Numerische Mathematik — 1959. — V.1 — P. 269-271.
10. Zhao H. Fast sweeping method for eikonal equations / H. Zhao // Mathematics of Computation — 2005. — V.74 — P. 603-627.
11. Seyb D. Non-linear sphere tracing for rendering deformed signed distance fields / D. Seyb, A. Jacobson, D. Nowrouzezahrai, W. Jarosz // ACM Trans. Graph. — 2019. — V.38, №6. — P. 12
12. Barr A. H. Global and Local Deformations of Solid Primitives / A. H. Barr // Computer Graphics (Proceedings of SIGGRAPH). — 1984. — V.18, №3. — P. 21–30.
13. Signed distance function [Електронний ресурс]. — Режим доступу до ресурсу: \https://en.wikipedia.org/wiki/Signed_distance_function
14. Tomczak L. J. GPU Ray Marching of Distance Fields / L. J. Tomczak // Technical University of Denmark. — 2012.

15. Perlin K. Hypertexture / K. Perlin, E. M. Hoffert // In ACM SIGGRAPH Computer Graphics. — 1989. — V23, №3. — P. 253-262.

16. Wright D. Dynamic Occlusion with Signed Distance Fields. Advances in Real-Time Rendering / D. Wright // SIGGRAPH. — 2015.

17. Distance functions [Електронний ресурс]. — Режим доступу до ресурсу: \https://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm

18. Taubin G. Distance Approximations for Rasterizing Implicit Curves / G. Taubin // ACM TransactIons on Graphics. — 1994. — V.13, 1. — P. 3—42.

19. Euler's method [Електронний ресурс]. — Режим доступу до ресурсу: \https://en.wikipedia.org/wiki/Euler_method

20. Heun's method [Електронний ресурс]. — Режим доступу до ресурсу: \https://en.wikipedia.org/wiki/Heun

21. Runge-Kutta's method [Електронний ресурс]. — Режим доступу до ресурсу: \https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods

Maximov A. L., Prokofieva S. V.

Approach to Effective Methods for Solution of Multidimensional Eiconal Equations

Summary

The article presents an extended method of solving the eiconal equation in four-dimensional space with weak deformations. The eiconal equation combines wave optics with geometric optics and has various physical interpretations, including the task of finding the shortest paths and calculation of electromagnetic or gravitational potentials. The proposed method extends the sphere tracing technique to spaces of many dimensions with deformations and demonstrated for the problem in the space of four dimensions. The method uses implicit functions for describing the boundaries of objects built from a finite or an infinite number of multidimensional primitives. Nonlinear sphere tracing is achieved by generation at each trace step ordinary (multidimensional) differential equations of the first order using a hybrid solution method combining the Euler method, when the sphere is close to the boundary, with higher order methods, when the sphere is far from the boundaries. Influence of non-linear transformations the tracing process is implemented using the Jacobian deformation matrix. The approach is implemented as a shader program in the GLSL language, and the effect of nonlinear transformations is determined using of the transformation parameter that affects the Jacobian matrix. The computational performance of the method is evaluated through average and maximum frame rates for different parameter values. The proposed approach can find application in such fields, as computer graphics, time-dependent computer tomography, seismic tomography, astrophysical modeling and optimal control.

Key words: eiconal, equation, deformation, tracing, Jacobi matrix, Euler method, shader, GLSL.

References

1. Javaid M. 4D printing applications in medical field: A brief review / M. Javaid, A. Haleem // Clinical Epidemiology and Global Health. — 2019. — V.7, №3. — P. 317—321.

2. Kwong Y. Four-dimensional computed tomography (4DCT): A review of the current status and applications / Y Kwong., A. O. Mel, G. Wheele, J. M. Troupis // Journal of

Medical Imaging and Radiation Oncology. — 2015. — P. 545—554.

3. Polacci M. Crystallisation in basaltic magmas revealed via in situ 4D synchrotron X-ray microtomography / M. Polacci, F. Arzilli, G. La Spina, N. Le Gall, B. Cai, M. E. Hartley, D. Di Genova, N. T. Vo, S. Nonni, R. C. Atwood, E. W. Llewellin, P. D. Lee, M. R.Burton // Scientific Reports. — 2018.
4. Irmak E. Concept of 4th Dimension for Databases / Irmak E., Kurtuldu . // 14th International Conference on Machine Learning and Applications (ICMLA). — 2015. — P. 1159—1162.
5. Raytracing 4D fractals, visualizing the four dimensional properties of the Julia set [Електронний ресурс]. — 2008. — Режим доступу до ресурсу: \http://www.codinginstinct.com/2008/11/raytracing-4d-fractals-visualizing-four.html
6. Prokofieva S. Methods for solving partial types of the eikonal equation / S. Prokofieva // Madrid: Barca Academy Publishing, Madrid, Spain, 2021. — P. 192—194.
7. Sethian, J. A. A fast marching level set method for monotonically advancing fronts / J. A.Sethian // Proceesings of the National Academy of Sciences. — 1969. — V.93, №4. — P. 1591—1595.
8. Gmez J. V. Fast Methods for Eikonal Equations: An Experimental Survey / J. V.Gmez, S. Garrido, D. Alvarez, L. E. Moreno // IEEE Access. — 2019.
9. Dijkstra E. A note on two problems in connection with graphs / E. Dijkstra // Numerische Mathematik — 1959. — V.1 — P. 269-271.
10. Zhao H. Fast sweeping method for eikonal equations / H. Zhao // Mathematics of Computation — 2005. — V.74 — P. 603-627.
11. Seyb D. Non-linear sphere tracing for rendering deformed signed distance fields / D. Seyb, A. Jacobson, D. Nowrouzezahrai, W. Jarosz // — ACM Trans. Graph. — 2019. — V.38, №6. — P. 12
12. Barr A. H. Global and Local Deformations of Solid Primitives / A. H. Barr // Computer Graphics (Proceedings of SIGGRAPH). — 1984. — V.18, №3. — P. 21–30.
13. Signed distance function [Електронний ресурс]. — Режим доступу до ресурсу: \https://en.wikipedia.org/wiki/Signed_distance_function
14. Tomczak L. J. GPU Ray Marching of Distance Fields / L. J. Tomczak // Technical University of Denmark. — 2012.
15. Perlin K. Hypertexture / K. Perlin, E. M. Hoffert // — In ACM SIGGRAPH Computer Graphics. — 1989. — V23, №3. — P. 253-262.
16. Wright D. Dynamic Occlusion with Signed Distance Fields. Advances in Real-Time Rendering / D. Wright // SIGGRAPH. — 2015.
17. Distance functions [Електронний ресурс]. — Режим доступу до ресурсу: \https://www.iquilezles.org/www/articles/distfunctions/distfunctions.htm
18. Taubin G. Distance Approximations for Rasterizing Implicit Curves / G. Taubin // ACM TransactIons on Graphics. — 1994. — V.13, 1. — P. 3—42.

19. Euler's method [Електронний ресурс]. — Режим доступу до ресурсу: \https://en.wikipedia.org/wiki/Euler_method

20. Heun's method [Електронний ресурс]. — Режим доступу до ресурсу: \https://en.wikipedia.org/wiki/Heun

21. Runge-Kutta's method [Електронний ресурс]. — Режим доступу до ресурсу: \https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods