



УДК 004.92

DOI: 10.20535/2077-7264.4(74).2021.247639

© Н. Є. Кулішова, канд. техн. наук, доц.,  
Д. Р. Зуєвський, студент, ХНУРЕ, м. Харків, Україна

## СТВОРЕННЯ ТА ВИКОРИСТАННЯ ДИНАМІЧНИХ МАСОК ПРИ РОЗРОБЦІ ТРИВИМІРНИХ МОДЕЛЕЙ ДЛЯ МОБІЛЬНИХ ІГОР

**Розглянуто основні етапи процесу створення динамічної багаточислової текстури. За дослідженою методикою створено динамічні маски для ігрового мобільного додатку. Доведено ефективність підходу для зниження обсягу даних, що описують тривимірну модель, та кількості звернень до пам'яті пристрою під час використання додатку.**

**Ключові слова:** динамічна маска; відеоефекти; тривимірне моделювання; багаточислова текстура; анімація текстури; мобільний додаток; ігровий додаток.

### Постановка проблеми

Для анімації текстур у тривимірній графіці є два основних підходи, перший з яких — використання набору зображень, кожне з яких є кадром анімації [1]. Цей спосіб вирішує більшість завдань, але потребує створення покадрового набору текстур для анімації. Крім того, подібний підхід у режимі реального часу знижує швидкість обробки одного кадра анімації. Другим способом є рух текстури в UV просторі, що є імітацією лінійного руху поверхні. До цього способу також можна віднести flow текстури [2–5], які імітують рух, спотворюючи UV простір, але вони вимагають додаткових обчислень у шейдері. Основним недоліком існуючих підходів є їхня неуніверсальність, складність редагування та необхідність значних обчислювальних витрат.

### Аналіз попередніх досліджень

Останніми роками широкого поширення до створення текстур у тривимірній анімації набули так звані динамічні текстури [6, 7]. В основі динамічної текстури (або маски) лежить параметричний підхід: зображення текстури поверхні створюється таким чином, що його зовнішній вигляд може змінюватись залежно від значення деякого параметра. По суті, динамічна маска є графіком тривимірної функції. Кожному значенню параметра анімації кадрів відповідає певний рівень поверхні функції. Плавна зміна параметра анімації, чи інакше, порога, дозволяє плавно варіювати зовнішній вигляд поверхні; змішувати кількість зображень текстури; «обрізати» поріг не лише вище за одне значення, а й кількох, отримуючи більш



складні ефекти, наприклад, такі, як хвилі. Перевага такої маски в тому, що одна динамічна текстура містить інформацію для всієї анімації, при цьому немає необхідності зберігати в пам'яті цілий набір покадрових зображень.

Динамічні маски виявилися надзвичайно ефективним інструментом для створення повністю або частково синтезованих зображень під час відновлення тривимірних моделей із результатів сканування. Особлива привабливість підходу в тому, що його швидкодія задовольняє вимогам функціонування у режимі реального часу [8, 9].

### Мета роботи

Розробити динамічну текстуру для тривимірних моделей в мобільному додатку, що зменшує обсяг потрібних графічних даних та кількість звернень до пам'яті пристрою.

### Результати проведених досліджень

У цифровому кольоровому зображенні  $l$  за змістом можна виділити кілька шарів  $L_i$ ,  $i = 1, \dots, N$ , кожен з яких містить кольорову

складову  $C_i$ , складову прозорості  $\alpha_i$  та композиційний порядок  $z_i$ , що пов'язує складові у суцільний шар. Динамічна текстура [4] — це генеративна модель, в якій розділяється візуальний компонент та його динаміка. Цю модель можна представити системою рівнянь:

$$\begin{cases} x_t = Ax_{t-1} + Bv_t \\ y_t = Cx_t + \sqrt{r}w_t \end{cases}, \quad (1)$$

де  $x_t \in R^n$  — стан моделі, яка розвивається в часі,  $y_t \in R^N$  — поточний кадр анімації,  $A \in R^{n \times n}$  — матриця переходу системи із стану  $(t-1)$  в стан  $t$ ,  $C \in R^{N \times n}$  — матриця перетворення стану системи в зображення кадру,  $Bv_t \sim \text{iid}N(0, Q)$

і  $\sqrt{r}w_t \sim \text{iid}N(0, rI_N)$  — шумові процеси стану та спостереження, параметризовані за  $B \in R^{n \times n}$  і  $r \in R$ , а початковий стан  $x \in R^n$  є константою.

В багатозаровій динамічній текстурі (рис. 1 [4], ліворуч) змінна призначення шару  $z_i$  призначає піксель  $y_i$  одному з станів (шарів) анімації і залежить від при-

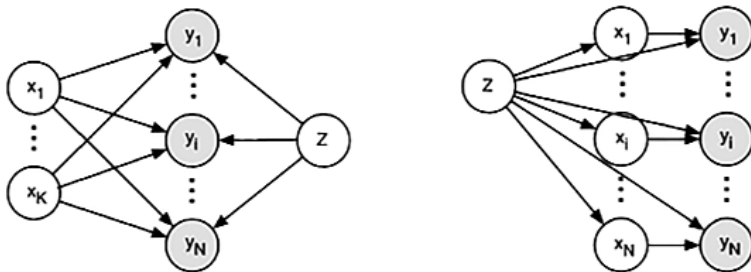


Рис. 1. Багатозарова динамічна текстура (ліворуч) і адаптивна шарувата динамічна текстура (праворуч);  $y_i$  — піксель, що спостерігається в часі,  $x_i$  — прихований стан анімації, а  $Z$  — множина змінних присвоєння шару  $z_i$ , які призначають кожен піксель одному з станів анімації



значення шару, пікселі в одному шарі моделюються як динамічна текстура. Крім того, сукупність призначень шарів  $Z = \{z_i\}_{i=1}^N$  моде-

люється як випадкове поле Маркова для забезпечення однорідності просторового шару. Як генеративна модель, шарувата динамічна текстура передбачає, що процеси стану  $X$  і призначення шару  $Z$  є незалежними, тобто рух шару не залежить від розташування шару, і навпаки.

В адаптивній динамічній текстурі (рис. 1 [4], праворуч) кожен піксель  $y_i$  пов'язаний зі своїм власним станом анімації  $x_j$  і для кожного пікселя визначено різну динамічну текстуру. Однак динамічні текстури, пов'язані з тим самим шаром, мають однаковий набір динамічних параметрів, які призначаються змінною призначення шару  $z_i$ . Знову, колекція призначень шарів  $Z$  моделюється як випадкове поле Маркова, але, на відміну від першої моделі, умова налаштування шарів створює всі пікселі незалежними.

За допомогою багатошарової динамічної текстури узгодженість динаміки анімації забезпечується за умови, щоб кожен піксель у шарі був пов'язаний з тим самим станом анімації. З іншого боку, для адаптивної динамічної текстури узгодженість у межах шару слабо забезпечується через те, що пікселі пов'язані з багатьма екземплярами станів (екземпляри, пов'язані з тим самим шаром із однаковими динамічними параметрами). Це слабша структура залежності забезпечує більш ефективний алгоритм модифікації текстури.

В межах одного шару значення змінних присвоєння шару  $z_i$  розподіляються у випадковому полі Маркова згідно виразу:

$$p(Z) = \frac{1}{Z} \prod_i \psi_i(z_i) \cdot \prod_{(i,j) \in \varepsilon} \psi_{i,j}(z_i, z_j), \quad (2)$$

де  $\varepsilon$  — набір ребер у сітці випадкового поля,  $Z$  — константа нормалізації (функція розподілу),  $\psi_i, \psi_{i,j}$  — потенційні функції виду

$$\psi_i(z_i) = \begin{cases} \alpha_1, & z_i = 1 \\ \dots & \\ \alpha_K, & z_i = K \end{cases}, \quad \psi_{i,j}(z_i, z_j) = \begin{cases} \gamma_1, & z_i = z_j \\ \gamma_2, & z_i \neq z_j \end{cases}. \quad (3)$$

Потенційна функція  $\psi_i$  визначає попередню ймовірність для кожного шару, тоді як  $\psi_{i,j}$  позначає більшу ймовірність для конфігурацій, де сусідні пікселі знаходяться в одному шарі. Параметри для потенційних функцій потрібно призначати для кожної моделі під час розробки завдяки, наприклад, картам зміщення.

Карта зміщення  $d(p, t)$  є функцією піксельних координат  $p$  і часу  $t$ . Застосування її безпосередньо до шару зображення  $L_i$  призводить до переміщення деформованого шару зображення  $L'_i$  такого, що [3]:

$$L'_i(p + d(p, t)) = L_i(p). \quad (4)$$



Однак, оскільки пряме відображення загрожує такими проблемами, як хибні контури та дірки, фактично використовується зворотне деформування, визначене як

$$L_i(p) = L(p + d(p, t)). \quad (5)$$

Позначимо цю операцію

$L' = L \otimes d'$ ,  $d'$  — обернена карта зміщення, яку отримано з  $d(p, t)$ . Для її визначення використовується процедура, яка, згідно з [3–7], включає кілька операцій:

- художник створює потокову карту динамічної текстур;
- нормаль поділяють на два шари;
- використовують шумові компоненти  $Bv_t \sim \text{iid}N(0, Q)$  і  $\sqrt{r}w_t \sim \text{iid}N(0, rI_N)$ , щоб зменшити артефакти як пульсації візерунка на стиках фрагментів текстур;
- видаляють не потрібні фази анімації, щоб зменшити ефект дублювання;

— виконують рендеринг. Під час процесу візуалізації для кожного моменту часу  $t$  і шару  $L_i$  синтезується карта зміщення  $d_i(t)$  (тут відкинуто залежності  $L_i$  і  $d_i$  від  $p$  для стислого позначення). Потім ця карта зміщення застосовується до  $C_i$  та  $\alpha_i$ , щоб отримати

$$L'_i(t) = L_i(0) \otimes d'_i(t). \quad \text{Це зміщення}$$

оцінюється як абсолютне зміщення вхідного зображення  $I(0)$ , а не відносно зміщення попереднього зображення  $I(t-1)$ . Це дозволяє уникати повторної передвибірki і накопичення числових помилок. Нарешті, всі модифіковані шари з'єднуються разом від фону до фронту для синтезу кадру в момент  $t$ ,  $I(t) = L_1(t) \otimes L_2(t) \otimes \dots \otimes L_n(t)$ ,

де  $z_1 \geq z_2 \geq \dots \geq z_n$  і  $\otimes$  є стандартним оператором перекривання.

У цій роботі смарт-об'єктом для динамічної текстур, створеної за допомогою пакету тримірного моделювання Blender [10], вибрано градієнт від білого до чорного (рис. 2). Глибина кольо-

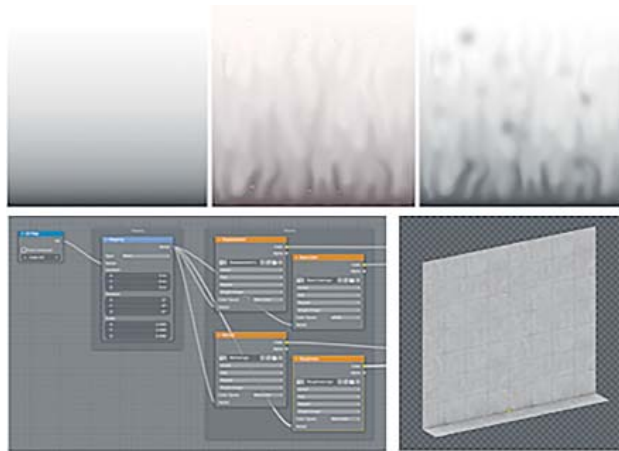


Рис. 2. Створення маски та базового матеріалу

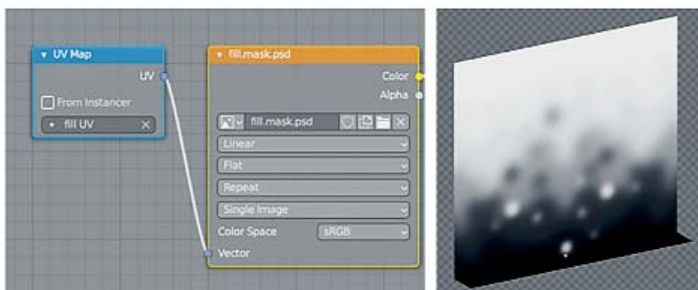


Рис. 3. Маска використовує окрему систему координат UV

ру визначає плавність маски. Для більш плавної зміни карти краще використовувати 16 або більше біт.

Використання кількох каналів розгортки спрощує роботу з накладанням текстури. Так, розгорткою для маски може бути проекція моделі або її частин з певного ракурсу. Таким чином, маска використовує власну систему локальних координат UV (рис. 3).

Швидкість зміни динамічної текстури регулюється значенням порога, який налаштовується для маски (рис. 4).

Для матеріалів, що змішуються, при цьому залишається можливість робити більш складні розгортки незалежно від маски (рис. 5, 6). За необхідності можна «запекти» маску для основної розгортки.

Отриману маску можна комбінувати з іншими картами для

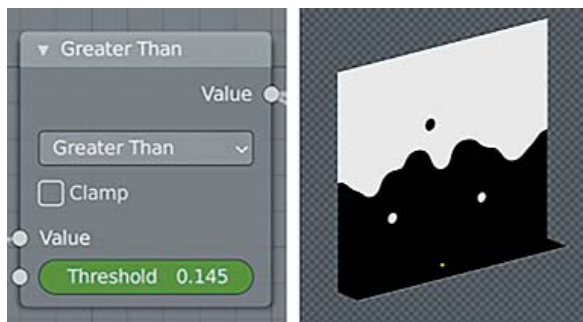


Рис. 4. Математичний вузол, що обробляє маску

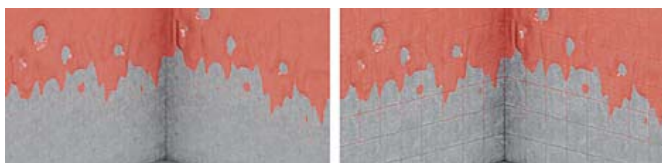


Рис. 5. Дві розгортки дозволяють змінювати текстури незалежно одна від одної

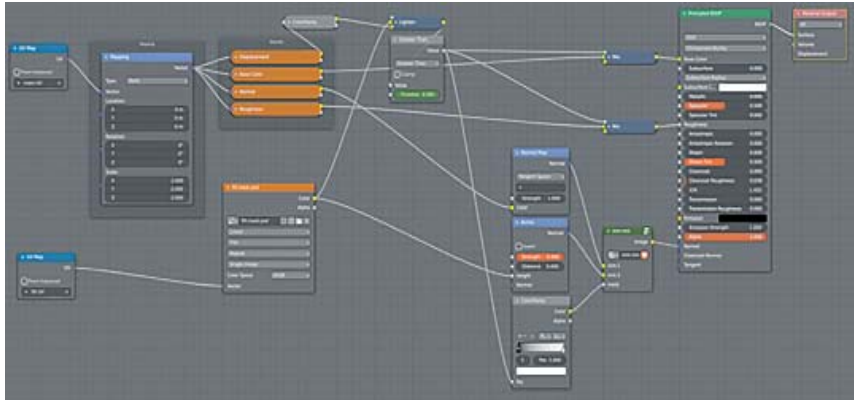


Рис. 6. Налаштований матеріал

досягнення більш реалістичного розповсюдження маски та її руху поверхню об'єкта (рис. 7).

При створенні складніших текстур для більш точного моделювання особливостей матеріалів використовується перемноження отриманої картки з іншими. Як й інші карти, цю можна поєднувати з іншими чорно-білими картами каналами [3], для ігор це зменшить кількість звернень під час завантаження карт. Картами для змішування можуть бути AT, bump, displacement чи різні канали карт нормалей (рис. 8).

Одна така динамічна текстура розміром 867 kb замінює 256 масок загальним розміром 5370 kb. Але головне в такому підході — це різниця не в обсязі даних, а в кількості викликів під час малювання кадру, а також звернень до пам'яті пристрою. Ці показники для динамічних текстур значно нижчі, ніж набору статичних масок (рис. 9).

### Висновки

Динамічна маска для деяких завдань є більш простим та ефек-

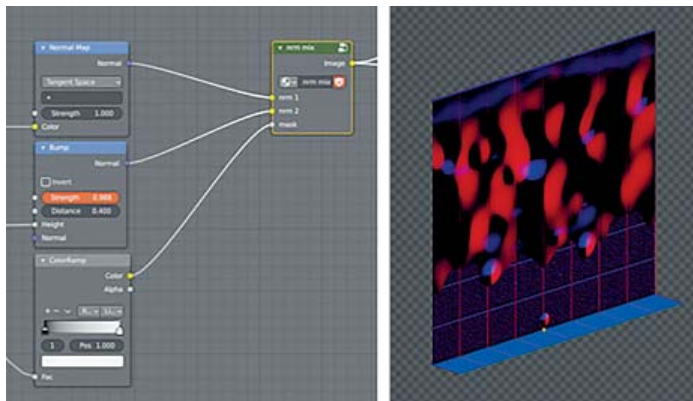


Рис. 7. На основі маски змішуються карти об'єму



Рис. 8. Змішування маски з displacement map

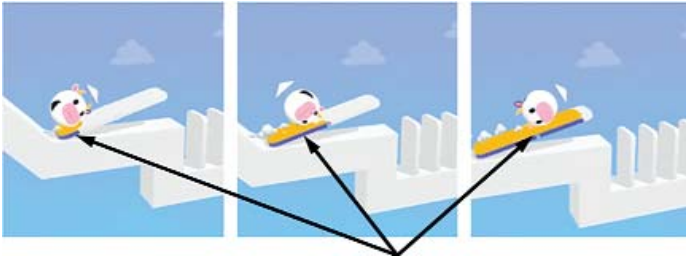


Рис. 9. Приклад практичного використання маски у мобільній грі

тивним рішенням. Її можна використовувати для створення ефектів намочання, фарбування, руху стилізованої води тощо. Перевагою є відносно малий розмір, який замінює покадровий набір

масок при невеликих обчислювальних витратах. Також цей підхід анімації може просто комбінуватися з класичними методами для складніших та комплексних завдань.

#### Список використаної літератури/References

1. Flavell, L. (2011). *Chapter 4: Lighting and Procedural Textures*. In: *Beginning Blender: Open Source 3D Modeling, Animation, and Game Design*. New York: Apress, 69–96.
2. *Texture Distortion. Faking Liquid*. Retrieved from <https://catlikecoding.com/unity/tutorials/flow/texture-distortion>.
3. *SIGGRAPH 2010 Water Flow in Portal 2*. Retrieved from <https://www.slideshare.net/alexvlachos/siggraph-2010-water-flow-in-portal-2>.
4. *Making Blood VFX in Assassin's Creed 3*. Retrieved from <https://80.lv/articles/001agt-behind-the-scenes-blood-in-assassins-creed/>.
5. *ChannelPacking*. Retrieved from <http://wiki.polycount.com/wiki/ChannelPacking>.
6. Chuang, Y.-Y., Goldman, D. B., Zheng, K. C., Curless, B., Salesin, D. H., & Szeliski, R. (2005). Animating pictures with stochastic motion textures. *ACM Transactions on Graphics*, Volume 24, Issue 3, 853–860. doi.org/10.1145/1073204.1073273.
7. Chan, A. B., & Vasconcelos, N. (2009). Layered dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 31, Issue 10, 1862–1879. DOI: 10.1109/TPAMI.2009.110.



8. Nagano, K., Seo, J., Xing, J., Wei, L., Li, Z., Saito, Sh., Agarwal, A., Fursund, J., & Li, H. (2018). Pagan: Real-time avatars using dynamic textures. *ACM Transactions on Graphics*, Volume 37, Issue 6, November, Article No.: 258, 1–12. doi.org/10.1145/3272127.3275075.

9. Habermann, M., Liu, L., Xu, W., Zollhoefer, M., Pons-Moll, G., & Theobalt, C. (2021). Real-time deep dynamic characters. *ACM Transactions on Graphics*, Volume 40, Issue 4, August, Article No.: 94, 1–16. doi.org/10.1145/3450626.3459749.

10. Blender 2020. *Blender*. Retrieved from <https://www.blender.org/>.

**The main stages of the development of a dynamic multilayer texture are considered. According to the investigated method, dynamic masks for a gaming mobile application are created. The effectiveness of the approach to reduce amount of data describing a three-dimensional model and the number of calls to device memory when using the application has been proven.**

**Keywords: dynamic mask; video effects; 3D modeling; multilayered texture; texture animation; mobile application; game application.**

Надійшла до редакції 17.11.21