

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет електроніки

(повна назва інституту/факультету)

Кафедра звукотехніки та реєстрації інформації

(повна назва кафедри)

«На правах рукопису»
УДК 004.2

«До захисту допущено»

Завідувач кафедри

(підпис)

(ініціали, прізвище)

“ ” _____ 20__ р.

Магістерська дисертація

зі спеціальності (спеціалізації) 171 Електроніка

(код і назва спеціальності)

на тему: «Дослідження можливостей програмного середовища Nuke для
забезпечення цифрового композитінгу у сучасному кіно»

Виконав студент VI курсу, групи ДВ-72мп

(шифр групи)

Орлов Богдан Валерійович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник доцент., к.т.н., доцент Трапезон К.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант _____

(назва розділу)

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2018 року

**Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»**

Інститут (факультет) _____ Факультет електроніки _____
(повна назва)

Кафедра _____ Кафедра звукотехніки та реєстрації інформації _____
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (спеціалізація) 171 Електроніка (Електронні та інформаційні технології кінематографії та аудіовізуальних систем)
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ (підпис) _____ (ініціали, прізвище)

«__» _____ 20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Орлова Богдана Валерійовича
(прізвище, ім'я, по батькові)

1. Тема дисертації «Дослідження можливостей програмного середовища Nuke

для забезпечення цифрового композитінгу у сучасному кіно». _____

науковий керівник дисертації Трапезон Кирило Олександрович, к.т.н.,
доцент _____,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «07» листопада 2018р. №4114-с

2. Строк подання студентом дисертації 10.12.2018р. _____

3. Об'єкт дослідження: набір візуального матеріалу, що є основою при створенні сучасного кінофільму.

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-кількість елементів для побудови 3D сцени – 4; 2) розширення файлу імпорту сцени – зовнішнє програмне забезпечення – FBX; 3) спосіб створення 3D сцени в середовищі Nuke – ручний ; 4) кількість віртуальних камер для налаштування 3D сцени – 1; 5) кількість елементів для проведення 3D рендерингу – 2; 6) максимальна кількість глибинних семплів – 10;

5. Перелік завдань, які потрібно розробити 1) Визначення технічних труднощів які спіткають розробників кінофільмів при роботі з 3D стерео матеріалом; 2) Побудова алгоритму в середовищі Nuke зі створення 3D сцени; 3) Ааналіз налаштувань які пов'язані з глибинним композитингом в середовищі Nuke.

6. Перелік графічного (ілюстративного) матеріалу 56 рис., 19 табл., 1 презентація, 10 слайдів.

7. Орієнтовний перелік публікацій Орлов Б. В. Дослідження програмних особливостей створення 3d-елементів для сучасного цифрового кіно *Elconf-2018.* – 2018.- Секція №7. – С 359 – 362; Орлов Б. В. Особливості використання багатокомпонентного композитингу *Сучасні проблеми застосування електронних та інформаційних технологій в телекомунікаціях, телебаченні та цифровому кінематографі: тези доповідей.* – 2018 – Секція А. – С 30 - 31.

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 10.09.2017

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
	Написання першого розділу: «Труднощі композитингу при роботі в 3d-стерео».	10.10.2017	
	Написання другого розділу: «3d-композитинг в nuke».	15.12.2017	
	Написання третього розділу: «Deep compositing».	01.05.2018	
	Написання четвертого розділу: «Розроблення стартап-проекту».	09.11.2018	
	Підготовка матеріалів до друку та оформлення пояснювальної записки	25.11.2018	
	Підготовка та оформлення презентації для доповіді	30.11.2018	

Студент _____
(підпис)

Б.В. Орлов
(ініціали, прізвище)

Науковий керівник дисертації _____
(підпис)

К. О. Трапезон
(ініціали, прізвище)

УДК 004.2

РЕФЕРАТ

Магістерська дисертація: 106 с., 15 рис., 19 табл., 7 джерел.

ВІЗУАЛЬНІ ЕФЕКТИ, ЦИФРОВИЙ КОМПОЗИТИНГ, КАНАЛ КОЛЬОРУ, МАСКА, ГЛИБИНА, РОЗМИТТЯ, ПРОЗОРИСТЬ.

Актуальністю дослідження полягає у розробленні послідовності дій яка дозволяє за наявності відзнятого візуального матеріалу провести налаштування з різних спеціальних видів цифрового композитингу.

Об'єктом дослідження набір візуального матеріалу, що є основою при створенні сучасного кінофільму.

Метою роботи є розроблення моделей в програмному середовищі Nuke, які дозволяють описати технічні особливості різних видів цифрового композитингу при створенні сучасних кінофільмів.

Для досягнення поставленої мети необхідно виконати такі завдання:

- Проаналізувати основні труднощі композитингу, які виникають у розробників 3D матеріалу. Визначення ролі глибини 3D сцени при створенні відповідного контенту;
- Розгляд основних проблем 3D композиції на основі аналізу особливостей програмного забезпечення Nuke. Побудова алгоритму дій зі створення 3D композитингу елементів. Розробка моделі в середовищі Nuke яка, ілюструє можливості реалізації 3D композитингу;
- Визначення головних функцій нод, які складають групу глибокого композитингу в програмі NukeX. Розробка послідовності дій яка ілюструє функціональні можливості обробки глибинних зображень.

SUMMARY

Master's dissertation: 106 pp., 15 rice, 19 tables, 7 sources.

VISUAL EFFECTS, DIGITAL COMPOSITING, COLOR CHANNEL, MASK, DEPTH, BLUR, TRANSPARENCY.

The urgency of the study is to develop a sequence of actions, which allows, in the presence of captured visual material, to make settings from various special types of digital compositing.

The object of research is the collection of visual materials, which is the basis for creating a modern movie.

The aim of the work is to develop models in the Nuke software environment, which allows you to describe the technical features of different types of digital composites when creating modern movies.

To achieve the goal you must accomplish the following tasks:

- Analyze the major compositing difficulties that developers in the 3D material. Determine the role of the depth of the 3D scene when creating the relevant content;
- Consider the main problems of 3D compositions based on the analysis of features of the software Nuke. Building an algorithm for creating 3D composite elements. Development of the model in the Nuke environment, which illustrates the possibilities of implementing 3D compositing;
- Define the main non functions that make up the deep composite group in the NukeX program. Developing a sequence of actions that illustrates the functionality of processing deep images.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	8
ВСТУП.....	9
1 ТРУДНОЩІ КОМПОЗИТИНГУ ПРИ РОБОТІ В 3D-СТЕРЕО	11
1.1 Стерео 3D	11
1.2 Вертикальна розбіжність.....	13
1.3 Особливості модуля Ocua в Nuke.....	15
1.4 Відмінності між ракурсами та їх врахування в Nuke.....	18
1.5 Установка глибини і налаштування конвергенції.....	21
1.5 Конвергенція.....	24
1.6 Дивергенція.....	26
1.7 Розмір екрану.....	27
1.8 Гіпер гіпостереоскопія.....	28
1.9 За межами кадру.....	28
1.10 Колірна відмінність.....	29
2 3D-КОМПОЗИТИНГ В NUKE.....	32
2.1 3D інсталяція сцени.....	32
2.1.1 Налаштування Nuke 3D сцени.....	34
2.1.2 Переміщення в 3D сцені.....	36
2.2 Використання Nuke для зберігання групи інструментів.....	37
2.3 Перегляд 2D зображення в якості 3D об'єкта.....	40
2.3.1 Використання пасу світового положення для створення 3D-об'єкта.....	42
2.3.2 Імпорт камери.....	43
2.4 Маніпулювання 3D ДЕРЕВА NODE в 3D просторі.....	46
2.4.1 Імпорт системи частинок.....	46
2.4.2 Суміщення 3D-об'єктів в просторі.....	50
2.5 Переміщення 3D-об'єктів в часі.....	52
2.6 Переведення 3D об'єктів в 2D пікселі.....	53

2.7 Застосування матеріалів до об'єктів.....	57
2.8 Корекція кольору зображення.....	61
3 DEEP COMPOSITING.....	64
3.1 Про Deep Compositing.....	64
3.2 Читання Deep Footage.....	66
3.3 Створення Deep Data.....	67
3.3.1 Перетворення послідовності 2D зображення в глибокі зображення використовуючи фрейми введення.....	68
3.3.2 Перетворення 2D-зображення в Deep Image.....	69
3.3.3 Перефарбування глибоких зображень.....	70
3.3.4 Використання ScanlineRender для генерації даних глибини.....	71
3.4 Злиття Deep зображень.....	74
3.5 Створення масок.....	75
3.5.1 Створення стиснутого зображення за допомогою DeepHoldout Node.....	75
3.5.2 Створення глибокої маски за допомогою DeepMerge Node.....	76
3.6 Створення 2D і 3D елементів з глибоких зображень.....	77
3.6.1 Створення 2D зображення з Deep Image.....	77
3.6.2 Створення хмари точок з глибокого зображення.....	77
3.7 Зміна Deep Data.....	79
3.7.1 Зміна кольору глибоких зображень.....	79
3.7.2 Налаштування ефекту Deep Color Correction.....	80
3.7.3 Зміна глибоких зображень за допомогою виразів.....	80
3.8 Зміна розміру, переформатування і перетворення Deep Images.....	81
3.8.1 Зміна розміру Deep Images.....	81
3.8.2 Переформатування Deep Images.....	82
3.8.3 Перетворення глибоких семплів.....	84
3.9 Відбір семплів глибокого зображення.....	84

3.10 Запис Deep Data.....	85
4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ.....	86
4.1 Опис ідеї проекту	86
4.2 Технологічний аудит ідеї проекту.....	88
4.3 Аналіз ринкових можливостей запуску стартап-проекту	89
4.4 Розроблення ринкової стратегії проекту	93
4.5 Розроблення маркетингової програми стартап-проекту	95
ВИСНОВКИ.....	98
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	10
ДОДАТОК А. ABSTRACT.....	101

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПЗЗ	–	Прилад з зарядовим зв'язком
CG	–	Computer generated;
CGI	–	Computer generated graphics;
DAG	–	Define a node graph;
DOF	–	Depth of field;
FOV	–	Field of view;
FBX	–	Flimbox;
OBJ	–	Object;
SFX	–	Special effects;
VFX	–	Visual effects;

ВСТУП

Актуальність теми. Швидкий розвиток кіновиробництва потребує зручних, доступних та зрозумілих інструментів що дозволяють значно скоротити час на створення кінофільмів зі спецефектами. Одним з напрямів кіновиробництва можна вважати створення 3D кіно з елементами комп'ютерної графіки. Для реалізації такого виду кінопродукту розроблені модулі та додатки програмного середовища Nuke. Безумовно складність цієї програми дещо обмежує особливості роботи але інструментарій нівелює ці обмеження. Це зумовило створення цифрових методів обробки відео контенту.

З ростом використання візуальних ефектів у фільмах, телесеріалах і рекламних роликах, стає необхідною висока продуктивність при створенні динамічних зображень та 3D сцен з підтримкою різних видів композитингу під час роботи в стислих термінах кіно виробництва. Наприклад, глибокий композитинг зображень вводить нові можливості, які збільшують гнучкість і вирішують старі проблеми композитинга на основі глибини.

Саме з цих обставин вважаю обрану тему дослідження актуальною.

Обґрунтування необхідності проведення дослідження. Розробка нових підходів та принципів які закладено в основу створення сучасних кінофільмів з комп'ютерними спеціальними ефектами.

Мета та завдання дослідження. Метою роботи є розроблення моделей в програмному середовищі Nuke, які дозволяють описати технічні особливості різних видів цифрового композитингу при створенні сучасних кінофільмів.

Для досягнення поставленої мети необхідно виконати такі завдання:

- Проаналізувати основні труднощі композитингу, які виникають у розробників D матеріалу. Визначення ролі глибини 3D сцени при створенні відповідного контенту;

- Розгляд основних проблем 3D композиції на основі аналізу особливостей програмного забезпечення Nuke. Побудова алгоритму дій зі створення 3D композитингу елементів. Розробка моделі в середовищі Nuke яка, ілюструє можливості реалізації 3D композитингу;
- Визначення головних функцій нод, які складають групу глибокого композитингу в програмі NukeX. Розробка послідовності дій яка ілюструє функціональні можливості обробки глибинних зображень.

Методом дослідження є програмні алгоритми та функції, які описують особливості з'єднання та взаємодії різних нод в режимах роботи програми Nuke.

Об'єктом дослідження набір візуального матеріалу, що є основою при створенні сучасного кінофільму.

Предметом дослідження є алгоритми та підходи які реалізують різні види цифрового композитингу в середовищі Nuke.

Новизна дослідження полягає у розробленні послідовності дій яка дозволяє за наявності відзнятого візуального матеріалу провести налаштування з різних спеціальних видів цифрового композитингу.

Практична цінність полягає у розробленні моделей в середовищі Nuke, результати роботи яких можна використати, в інших програмах комп'ютерної графіки.

1 ТРУДНОЩІ КОМПОЗИТИНГУ ПРИ РОБОТІ В 3D-СТЕРЕО

3D-кіно пройшло довгий шлях з тих перших днів, коли застосовувалися хитромудрі пристосування для отримання зображень які викликають головний біль, незмінно асоційованих в масовій свідомості з легко впізнаваними червоно-синіми анагліфічними окулярами, необхідними для перегляду перших стереозображень. Згодом досягнення в технологіях зйомки і проєкції дозволили усунути багато проблем, пов'язаних з ранніми експериментами в цій галузі. Однак, якщо не вжити певні дії на стадії композитинга, все ще можна створити стереоскопічний 3D-фільм, дивитися який буде некомфортно, а часом і просто неможливо для аудиторії. В даному розділі розглядається ряд проблем, з якими наймовірніше доведеться зіткнутися при роботі над стереофільмами, і окреслено варіанти вирішення цих проблем за допомогою широких 3D-можливостей системи Nuke.

1.1 Стерео 3D

Поza кіно наші візуальні системи визначають глибину простору навколо нас на основі різниці між сприйняттям світу правим і лівим оком. Ця різниця обумовлена відстанню між очима і найбільш очевидна при погляді на об'єкти розташовані близько до нас, оскільки кути, під якими бачить кожне око, злегка розрізняються. Коли ми дивимося на віддалені об'єкти, відстань між нашими очима (наша «міжокулярна» відстань) куди менша, ніж відстань до об'єктів, а тому обидва ока бачать ці об'єкти практично однаково. Стало бути, їх положення для кожного ока однакове, на відміну від об'єктів на близькій відстані.

Стосовно до кінематографа це означає, що ми можемо «обманути» наш зір і змусити його бачити обсяг на плоскому екрані шляхом демонстрації

кожному оку злегка різних зображень. Сьогодні, як і в 3D-кінотеатрах 1950-х років, це зазвичай досягається шляхом демонстрації зображень для кожного ока з застосуванням поляризованого світла. Аудиторія повинна вдягти поляризаційні окуляри, що забезпечують показ того чи іншого зображення тому чи іншому оку, фільтруючи альтернативний ракурс. Найбільш оптимальний результат - з мінімальним ризиком головного болю або втоми зору - може бути досягнутий, якщо гарантувати, що різниця між ракурсами максимально точно відповідає тому, що ми бачимо в реальному житті, а тому перегляд зображення не змушує наш зір перенапружуватися. Проте, на практиці це навряд чи можливо, і не тільки внаслідок таких фізичних обмежень, як розміри і положення кіноекрана, а й з-за творчих компонентів, необхідних для розповіді кіноісторії, а також необхідності зробити приємність аудиторії, для чого, в кінці кінців, вона і приходить в кінотеатр. Наприклад, найбільш реалістична репрезентація сцени може бути досягнута шляхом зйомки двома камерами, рознесеними на відстань, рівну відстані між очима середньостатистичної людини, а потім відображення знятого матеріалу зі збереженням цієї умови. На жаль, в більшості випадків це призведе до створення вкрай стомлюючого 3D-фільму.

Уявіть, зокрема, кадр ландшафту, на якому найцікавішим об'єктом є купа далеких гір. Внаслідок значної відстані зйомка цієї сцени з рознесенням камер (міжосьова відстань) на величину, рівну відстані між очима людини, призведе до отримання двох практично ідентичних ракурсів гір. В результаті, при проєкції на кіноекран гори будуть виглядати абсолютно плоскими – так, можливо, реалістичними, але від цього не менш розчаровують кожного, хто прийшов у кіно насолодитися вражаючим шоу. Замість цього, щоб максимізувати вплив на глядача від нашого гірського пейзажу, міжосьова відстань між камерами при зйомці буде, швидше за все, збільшеною. І глядачеві запропонують не звичайний ракурс, властивий людському сприйняттю, а ракурс

для велетня - з пропорційно збільшеною межокулярною відстанню, але для тієї ж самої сцени. І тепер, при перегляді як би широко розставленими очима велетня, навіть віддалені гори будуть бачитися тривимірними.

Ще одна причина, за якою максимально точна емуляція дійсності не завжди бажана, полягає в фіксованому положенні кіноекрану. У реальному житті, коли ми дивимося на об'єкт, наші очі сходяться на ньому одночасно з наведенням на різкість, щоб чітко бачити об'єкт. Коли ми дивимося на кіноекран, наші очі завжди сфокусовані саме на екрані, весь час знаходиться на одній і тій самій відстані від нас. Однак, коли ми стежимо за дією в кадрі, наші очі часто повинні фокусуватися на точці, що знаходиться десь за екраном, а часом навіть перед ним. Утримання точки конвергенції постійно на екрані, на якому ми фокусуємося – призведе до жорстких обмежень і не дасть отримати той повноцінний 3D-ефект, який повинен бути.

В обох випадках кращим рішенням буде не спроба відтворити реальність, а скоріше представити глядачеві поліпшену версію оптимальну для отримання 3D-ефекту, свого роду «гіперреальність». Очевидно, що чим більше розсуваються межі того, що очікує побачити глядач, тим важче доведеться працювати зору при перегляді, а це збільшує ймовірність втоми очей або того гірше.

1.2 Вертикальна розбіжність

Різниця в положенні однієї і тієї ж точки сцени на зображеннях для правого і лівого ока зазвичай називають «стереоневідповідністю» (stereo disparity) для цієї точки. Внаслідок горизонтального рознесення наших очей ці невідповідності практично завжди мають чіткий горизонтальний характер. Вертикальні невідповідності, з іншого боку, як правило, не зустрічаються в

реальному житті, і в результаті нашим очам може бути вкрай важко впоратися з ними.

На жаль, вертикальні невідповідності між лівим і правим зображеннями, з яких і створюється 3D фільм, є скоріше правилом, ніж винятком. До винаходу цифрової проекції, що дозволила за допомогою одного проектора відображати обидва зображення, вертикальні невідповідності були основним джерелом дискомфорту для кіноглядачів, оскільки в більшості випадків використовувалося два не надто точно суміщених проектора, що часто призводило до вертикальних невідповідностей між ракурсами. Однак навіть зараз, коли «головний біль» з суміщенням проекторів усунено, вертикальні невідповідності можуть виникати навіть на стадії зйомки фільму. Найбільш вирогідним є не дуже точне вертикальне суміщення двох камер при зйомці сцени, більше того, одна камера може бути злегка повернутою щодо іншої камери. Є і другий, менш очевидний випадок, при якому може виникати вертикальна невідповідність. 3D-картини часто знімаються двома камерами, поверненими в бік одне одного так, що їх ракурси сходяться в одній точці - це імітація того, як наші очі зазвичай дивляться на об'єкт. Коли камери розташовані таким чином, фізичні площини, на яких формується зображення - площа кадру плівки або прилад з зарядовим зв'язком (ПЗЗ) в цифровій камері - теж будуть знаходитися під кутом один до одного, а також до сцени, яку треба буде згодом відобразити. Такий кутовий ракурс призводить до легких перспективним спотворень, які для кожного з двох ракурсів НЕ будуть однаковими (рис. 1.1). Цей тип спотворення відомий як трапеція і призводить до деякої вертикальної невідповідності між двома ракурсами. Різниця може здаватися незначною, але її цілком достатньо для того, щоб викликати проблеми при проекції на великий екран.

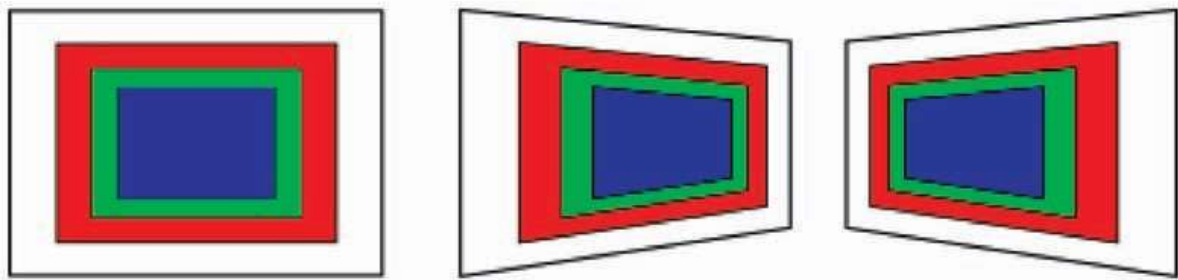


Рисунок 1.1 – Трапеційдальне спотворення: оригінал, зображення з лівої камери, зображення з правої камери

Саме тут доречна перша поява модуля *Ocula* для *Nuke*. *Ocula* - це набір засобів для корекції загальних проблем зі стереоматеріалами.

1.3 Особливості модуля *Ocula* в *Nuke*

Можливості *Nuke* і *Ocula* в плані їх застосування при створенні 3D-фільмів використовували такі компанії, як *Weta Digital*, *Framstore* і *CafeFX*. *Nuke* і *Ocula* застосовувалися при роботі над кількома великими стереокартинами, включаючи фільм «Брама» (*The Hole*) - володар нагороди за кращий стереоскопічний фільм на фестивалі у Венеції в 2009 році, а також гучний «Аватар» Джеймса Кемерона.

І *Framestore*, і *Weta Digital* збирали «Аватар» в *Nuke*. У випадку з *Framestore* вибір *Nuke* був обумовлений складністю роботи над «Аватаром», що відзначив голова *Nuke Compositing* Крістіан Кайстнер (*Christian Kaestner*). «Без сумніву, «Аватар» змусив використовувати технології на межі їх можливостей, причому не тільки на рівні 3D або 2D цього проекту, але і під час композитингу, - говорить він. - Кількість шарів комп'ютерної графіки в кожній сцені, з яким повинен був працювати фахівець, було настільки великим, що нам раніше не доводилося стикатися з таким рівнем складності. Ми спостерігали, як *Nuke* справляється з цим, і переконувалися, що системі ця робота під силу. Все, що

ми поклали на Nuke - а це неймовірну складність фільму «Аватар» - система виконала бездоганно ».

А компанія Weta широко застосовувала Ocula. «Модулі Ocula для Nuke стали для нас ефективним, потужним набором засобів при роботі над «Аватаром», - говорить відповідальний за композітинг Ерік Уінквіст (Erik Winquist).

Розташована в США студія CafeFX також вибрала Nuke і Ocula для своєї роботи над «Пунктом призначення» (The Final Destination). На переконання відповідального за візуальні ефекти Тома Вільямсона (Tom Williamson), реальної альтернативи не було: «У порівнянні з іншими системами, які більш жорстко і куди менш елегантно обходяться з матеріалом в сенсі технологічного процесу, Nuke виявився на висоті, - говорить він. - Парадигма View, закладена в Nuke, а також весь процес роботи над лівим і правим ракурсами за допомогою одного комплекту вузлів, яким можна керувати кожним каналом окремо, але зберігаючи можливість спільного управління, виявилася вкрай вдалою. Ми були задоволені тим, що обрали Nuke і взяли участь в програмі бета-тестування Ocula».

Ще однією перевагою Nuke є простота, з якою можна наростити базову функціональність системи, будь то додавання програмних модулів, скрипти Python або можливість створення потужних Gizmo за допомогою ефективних інструментів. Технічні фахівці CafeFX застосовували можливості Gizmo для створення Nuke Gizmo, що забезпечила сумісність з 3D-моніторами Zalman Triton, що дозволило взаємодіяти з 3D-сценою в режимі реального часу. «Ми мали можливість вносити зміни в композицію в середовищі Nuke і тут же бачити результати, - каже Вільямсон. - Ми також могли перемикатися між правим і лівим ракурсами, розділяти вузли, щоб міняти індивідуальні налаштування, а потім відтворювати матеріал і переглядати результати в режимі реального часу».

Не менше враження на Вільямсона справив і набір модулів Osula, зокрема, модуль VerticalAligner і функція Correlate, яку він забезпечує в складі Nuke. «Вертикальне зведення - це прекрасна функція Osula, - говорить він. - Для виправлення планів, в яких була розбіжність правого і лівого ракурсів, або були трапецієподібні спотворення, ми пропускали їх через Osula. Не менше вразив модуль Correlate, що дозволив істотно зменшити витрати ручної праці в сфері розфарбовування і промальовування. Маніпуляції із зображенням, такі як розфарбовування, промальовування і створення полігональних форм, зроблені для одного ока, автоматично дублюються для іншого». На думку Вільямсона, це не тільки значно підвищує продуктивність, але і покращує остаточний результат: «Поки ми не отримали Osula, всю цю роботу доводилося виконувати окремо для кожного каналу, і це було справжньою катастрофою, - каже Вільямсон. - Око людини дуже добре вловлює невідповідність між двома зображеннями, але за допомогою Osula його можна знизити до рівня, недоступного людському зору, а саме, до субпіксельних значень».

Цю думку розділяє фахівець з візуальних ефектів Джей Бартон (Jay Barton) з компанії Digital Domain, яка використала Nuke і Osula для створення рекламного 3D-ролика SoBe Lifewater, що демонструвався під час фіналу з американського футболу в 2009 році. «Набір функцій Osula для Nuke значно прискорив процес роботи зі стереозображення. Робота в режимі перегляду стерео з миттєвим доступом до складних інструментів багат шарового монтажу зміщених одна відносно іншої потоків зображень і візуалізованими елементами дозволила нам отримати більш якісний результат за набагато менший час, ніж раніше».

Отже, як вихід для усунення трапеційдального спотворення можна виористати інструмент VerticalAligner. Він перший перевіряє два зображення, щоб визначити точки відповідності на кожному ракурсі, потім обчислює трансформацію, яку потрібно застосувати до кожного ракурсу, щоб зробити ці точки вирівняними по горизонталі з максимально доступною точністю. Результатом служить пара зображень, в яких вертикальні невідповідності прибрані, а тому вони не зможуть викликати у глядачів в кінотеатрах головний біль.

Дуже часто вертикальне зведення є першим завданням в технологічному процесі роботи з стереобладнанням. У багатьох випадках VerticalAligner зі складу Oscola може замінити трудомістке ручне вирівнювання - досить зробити кілька кліків миші.

1.4 Відмінності між ракурсами та їх врахування в Nuke

При виконанні багат шарового монтажу (compositing) матеріалу в режимі стерео важливо зберігати правильний взаємозв'язок між двома ракурсами. Будь-які відмінності, що з'являються на етапі обробки і монтажу, або ті, що були виправлені в цей час, згодом створять глядачам великі труднощі при сприйнятті цих двох ракурсів як єдиної 3D-сцени.

Технологічний процес роботи зі стерео в Nuke розроблений так, щоб зберігати відповідність між ракурсами було максимально просто. Ракурси для лівого і правого ока можна зчитувати як єдиний потік зображення і обробляти його в такій формі протягом всього процесу монтажу, якщо є така необхідність. Це означає, що глобальний ефект, такий як розмиття або кольоросинхронізація, може бути застосований до обох ракурсів одним клацанням миші і однаково вплине на обидва ракурсу. І навпаки, якщо потрібно виконати прив'язану до позиції операцію, таку як додавання комп'ютерного персонажа до знятому

наживо матеріалу, важливо зберегти різницю між ракурсами, щоб правильно виконати позиціонування. В цьому випадку вузли Split and Join, наявні в Nuke, дозволяють розділити ракурси в стереопотоці, обробити їх окремо, потім знову об'єднати. Далі ракурси будуть оброблятися як єдиний потік, або можуть бути знову розділені, а потім об'єднані в будь-який момент, коли це буде потрібно (рис. 1.2).

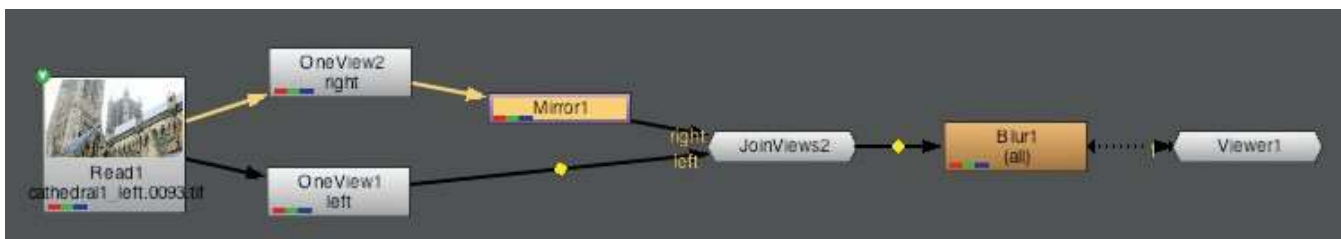


Рисунок 1.2 – Приклад обробки стереопотоку в Nuke

Два ракурси сцени зчитуються як єдиний потік, поділяються так, щоб трансформація була застосована тільки до правого ракурсу, потім знову об'єднуються. Розмиття застосовується до обох ракурсів одночасно.

Модуль Ocula також дуже зручний для точного, прив'язаного до ракурсу позиціонування нових елементів, що потрібно для гарантовано правильного їх появи в 3D-сцені. Ключовим компонентом Ocula є DisparityGenerator, здатний створювати карту попиксельної відповідності для кожної пари лівого і правого ракурсів. Карта невідповідності «зліва направо» говорить про те, як кожен піксель лівого ракурсу потрібно пересунути, щоб створити правий ракурс, і аналогічно діє карта невідповідності «справа наліво». Ці дві карти не є просто інверсією по відношенню один до одного - деякі частини лівого зображення можуть бути приховані в правому ракурсі і навпаки. Навколо меж поля зору будуть області, які видно тільки на одному із зображень.

Nuke може використовувати карту невідповідності для копіювання елемента, намальованого в одному ракурсі, такого як мазок пензля чи щось

інше, в відповідне місце в іншому ракурсі, завдяки чому істотно економиться час, а також забезпечується точність позиціонування скопійованого елемента. Внаслідок відмінності кутів огляду часто виникає необхідність деформування елемента, щоб надати йому правильну форму в другому ракурсі. Nuke може трансформувати елемент двома способами: по-перше, шляхом простого слідування вектору невідповідності від одного ракурсу до іншого. А по-друге, тому що в прихованих областях, наприклад, вектори невідповідності не завжди несуть повну інформацію, опція Correlate Using Ocula буде слідувати векторам невідповідності (як в першому випадку), щоб обчислити приблизне відповідне положення для елемента, потім вона виконає уточнення цього первинного припущення, щоб обчислити найкраще становище елемента у другому ракурсі.

Ocula також містить генератор NewView, здатний створювати нові проміжні ракурси в будь-якому місці між вихідними лівим і правим ракурсами. Як і описана вище кореляція, цей процес починається з формування карти невідповідності, але, крім того, «інтелектуально» працює з будь-якими прихованими областями, що дозволяє генерувати максимально реалістичні проміжні ракурси сцени. До того ж, якщо є карта невідповідності, NewView може використовуватися для створення лівого ракурсу з правого і навпаки. Це може бути корисно для корекції помилок - усунення тих невідповідностей між ракурсами, які можуть вплинути на можливість вірного сприйняття глядачем двох ракурсів як єдиного 3D-зображення, а не як двох різних зображень. Уявіть, наприклад, що ліва камера бачить яскравий відблиск в конкретній точці, а права камера практично нічого не бачить. Хоча фізично все правильно, якщо ліве зображення не буде оброблено, така різниця в освітленні створить серйозний дискомфорт при перегляді в кінотеатрі. У цій ситуації генератор NewView можна застосувати для трансформування ракурсу з правої камери - створити те, що бачила б в цій області ліва камера, якби там не було яскравого

відблиску. Скоригована область потім може бути прорахована і перенесена назад у вихідний лівий ракурс, замінюючи відблиск.



Рисунок 1.3 – Приклад того, як форма, намальована навколо вивіски (зліва внизу) в лівому ракурсі, автоматично копіюється засобами Nuke в правильне положення в правому ракурсі

1.5 Установка глибини і налаштування конвергенції

Вище вже згадувалося, що глибину стереоскопічного 3D-фільму додає міжосьова відстань між камерами. Стало бути, зміна міжосьової відстані дозволяє змінити глибину сцени, роблячи її більш дрібною або глибокою залежно від того, зменшується відстань між камерами або збільшується. Звичайно, це легко робити, коли камери віртуальні і застосовуються для візуалізації комп'ютерного фільму. У реальному житті це зробити важче, а після зйомки - під час монтажного процесу - взагалі неможливо (хоча з Ocula можна до цього наблизитися). Основна проблема полягає в тих самих прихованих областях, про які говорилося вище.

Коли камера переміщається щодо тривимірної сцени, яку вона знімає, різні області сцени стають видимими або невидимими, як показано на рис. 1.4.

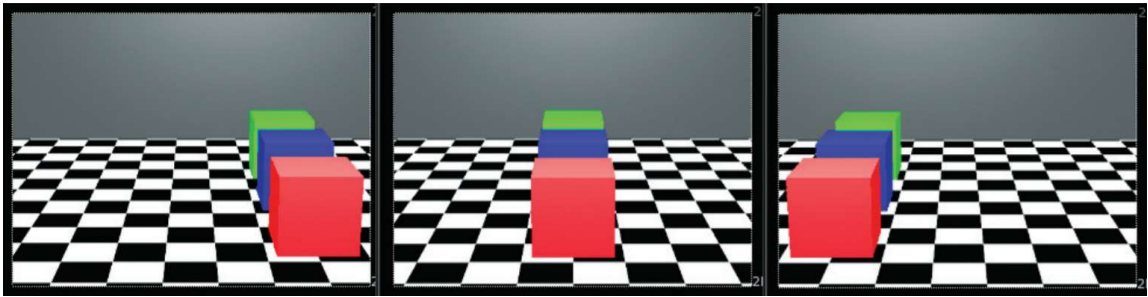


Рисунок 1.4 – Одна і та ж сцена з трьох різних позицій камери по мірі її переміщення зліва направо

При наявності стереопари зображень основна частина сцени буде видна на обох ракурсах, але деякі області біля кордонів предметів виявляться невидимими в одному ракурсі, але видимими в іншому. Якщо потім буде необхідно змістити лівий ракурс до правого, наприклад, зміниться не тільки кут, під яким ліва камера бачить сцену, а й області, які приховані: деякі частини сцени, невидимі раніше на лівому зображенні, тепер стануть видні. Ці ефекти найбільш сильні поблизу камери і будуть послаблюватися в міру віддалення від об'єктів зйомки.

Тому якщо режисер вирішує, змінити глибину сцени під час монтажу, складність проблеми визначається тим, для якої кількості видимих частин сцени потрібно індивідуально обчислити величину зсуву: шляхом обчислення невідповідності між лівим і правим зображеннями ми дійсно можемо обґрунтовано припустити, як все повинно бути. Справжньою проблемою є області недостатньої інформації, де частина сцени, яка раніше була прихована, стає видимою, а ми не знаємо, що камера побачила б з цієї нової точки. Однак відомо, як ці області виглядають в ракурсі з іншої камери. Також відомо, як ці області сцени змінюється від одного ракурсу до іншого. Як вже зазначалося, можна зробити обґрунтоване припущення про те, як має виглядати зображення, і це саме те, що робить модуль `InteraxialShifter` який входить до складу `Ocula`. Його можна застосовувати для зниження віртуальної глибини сцени шляхом

наближення двох ракурсів один до одного, як якщо б відстань між камерами, які використовувались для зйомки, виявилось менше, ніж було насправді. Через проблеми прихованих областей, а також з-за того, що передбачувана невідповідність не може мати стовідсоткову точність, два нових ракурсу не будуть ідеальні, але їх можна створити швидко і легко, а згодом буде потрібно лише незначна ручна корекція.

На жаль, навіть за допомогою InteraxialShifter не можна виконати дії в зворотному напрямку і дійсно збільшити міжосьову відстань камер на стадії монтажу. З однієї простої причини - ми не маємо інформації про невідповідність за межами вихідної пари зображень. Хоча, шляхом екстраполяції невідповідності, можливо прийти до обгрунтованого припущення, хоча б на невеликих дистанціях. Але і тут реальну проблему створюють приховані області: при віддаленні камер одна від одної починають відкриватися області, які не були видні ні в одному з вихідних ракурсів. В цьому випадку дійсно немає ніякої інформації про те, як заповнити ці відсутні області. Зробити обгрунтоване припущення про те, що там має бути, куди складніше, якщо взагалі можливо.

Проте, можливість зменшити міжосьову відстань під час монтажу і обробки не позбавлене цінності, коли мова заходить про створення комфортного для перегляду 3D-фільму. При зміні глибини сцени очам людини потрібен якийсь час, щоб перебудуватися на новий стан віртуального середовища. З цієї причини швидкі зміни планів з різною глибиною небажані, і їх потрібно уникати. Хоча зазвичай прийнято планувати зміну глибини для всього фільму ще до початку зйомок (куди має входити і поєднання сусідніх сцен по глибині, наскільки це можливо), порядок планів може змінюватися вже на стадії монтажу. У цьому випадку буде потрібно анімувати глибину так, щоб вона змінювалася поступово від одного плану до іншого. Якщо це відбувається

плавно, аудиторія не помітить зміни, але обов'язково негативно відреагує на різкі, дискомфортні зміни від сцени до сцени.

Може також виникнути бажання зменшити глибину в сценах з великою кількістю швидкого руху, у яких занадто сильний стереоефект може ввести в оману і зробити сцену складною для сприйняття.

1.5 Конвергенція

З конвергенцією все точно так само, як і з глибиною – раптовий її перехід від однієї сцени до іншої створює труднощі для зору. Якщо число таких стрибків велике протягом пари годин, поки йде 3D-фільм, це зробить його дуже некомфортним для перегляду.

Коли ми говоримо про конвергенцію в цьому контексті, то насправді маємо на увазі нульову стереорозбіжність, або нульовий паралакс. Багато 3D-картин зняті паралельними, а не збіжними камерами, що дає горизонтальну пару ракурсів сцени. В такій ситуації «конвергенція», або точка нульового паралакса, встановлюється на стадії монтажу і обробки шляхом накладання одного зображення на інше до тих пір, поки бажана фокальна точка сцени не опиниться в однаковому становищі в кожному з ракурсів, щоб стереорозбіжність виявилася рівною нулю. При перегляді в кінотеатрі ця частина сцени буде бачитися на глибині екрана. Що ж стосується всього іншого, то будь-яка частина сцени, положення якої в лівому ракурсі правіше її положення в правому ракурсі (негативний паралакс) буде бачитися перед екраном. Будь-яка точка, положення якої в лівому ракурсі лівіше її положення в правому ракурсі (позитивний паралакс) буде бачитися за екраном, як показано на рис. 1.5.

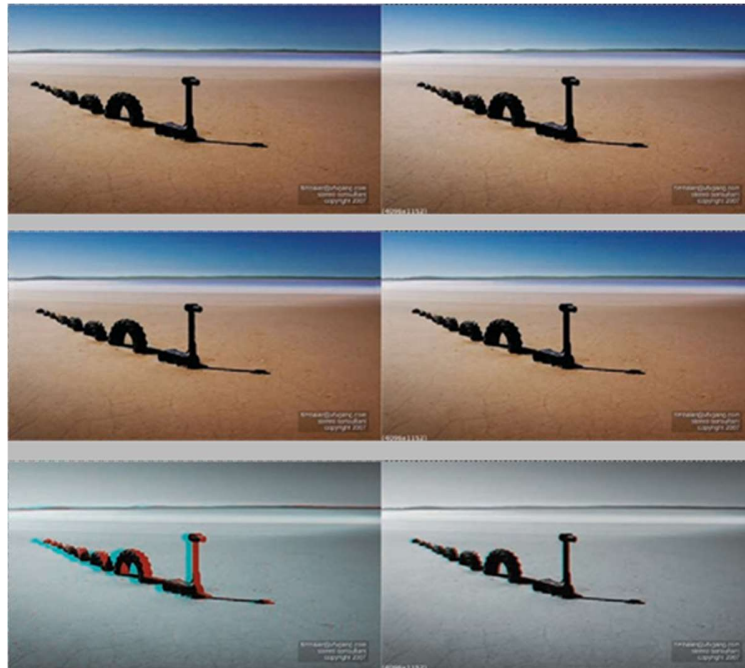


Рисунок 1.5 – Приклад зміни міжосьової відстані за допомогою Osula. Верхній ряд – вихідні лівий і правий ракурси. Середній ряд – лівий і правий ракурси після наближення одного до іншого на 40% стереоне відповідності. Нижній ряд – анагліфний вид пари зображень до і після міжосьового зміщення

Це означає, що зміна точки конвергенції з положення перед екраном на положення за ним стрибкоподібно змінить всю 3D-сцену – якщо до зміни вона бачилася за екраном, то після зміни буде бачитися перед ним. Такий стрибок не схожий ні на що з того, що ми бачимо в реальному світі, а тому його сприйняття може бути дуже складним для нашого зору (і мозку). Як і у випадку зі стрибком глибини, стрибки конвергенції можна згладити, анімуючи її зміну так, щоб вона відбувалася поступово.

На практиці міжосьова відстань і точка конвергенції сцени тісно пов'язані, і часто бажано міняти їх разом. При зміні одного з параметрів або обох слід, однак, потурбуватися про те, щоб уникнути появи дивергенції – гарантованого способу довести аудиторію до хворобливого стану.

1.6 Дивергенція

Дивергенція має місце, коли ступінь позитивного паралакса між двома ракурсами настільки велика, що глядач змушений розводити очі від центральної осі (осі симетрії), щоб сприймати сцену. Очевидно, такого в реальному світі не трапляється, і людський зір зовсім не розрахований на це. Хоча краще повністю уникати дивергенції, є загальноприйнята думка, що дуже мала дивергенція - не більше 1° - цілком прийнятна для більшості людей, а тому допустима і в 3D-кіно. Звідси впливає ще одна ситуація, в якій може знадобитися зменшення міжосьової відстані при зйомці. При переміщенні точки конвергенції вперед позитивний паралакс в напрямку заднього плану сцени буде збільшуватися, можливо, до моменту виникнення дивергенції на найвіддаленіших об'єктах. Якщо це відбувається, можна використовувати InteraxialShifter для зменшення загальної глибини сцени і приведення невідповідності в цих областях в розумні межі.

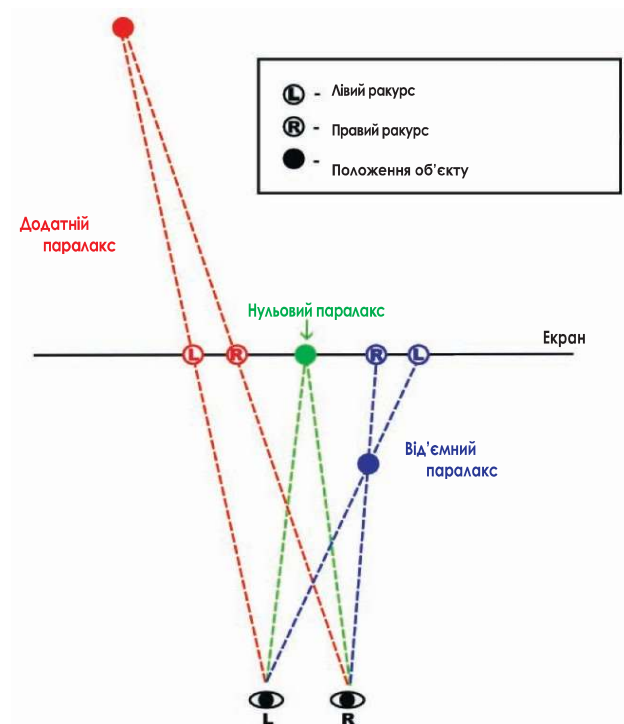


Рисунок 1.6 – Вплив паралакса на удаване положення об'єкта

1.7 Розмір екрану

Є ще дещо, що впливає на уявну глибину 3D-сцени. Це розмір екрану, на якому здійснюється відображення. А точніше, глибина, яку сприймає глядач, буде визначатися відношенням розміру екрану до відстані глядача до нього. Це відношення не буде однаковим для дизайнера перед комп'ютерним монітором і людини в кінотеатрі. Отже, вони по-різному будуть сприймати той же діапазон глибини, якщо стереосцену, створену на комп'ютері, просто збільшити до розмірів кіноекрану. Фактично, сцена, сприймається як практично плоска і нецікава на екрані комп'ютерного монітора, може бути більш вражаючою при показі в кінотеатрі. Тим не менш, це також означає, що в кінотеатрі може виникнути дивергенція в сценах, в яких вона не була видна при перегляді на малих екранах під час комп'ютерної обробки. Тому вплив розміру екрану треба завжди мати на увазі.

1.8 Гіпер гіпостереоскопія

Ще однією причиною для обережності при варіаціях з міжосьовою відстанню є те, що занадто велике відхилення від середньої міжокулярної відстані може викликати небажаний ефект. Дуже велика міжосьова відстань - «гіперстереоскопія» - замість сприяння в додаванні глибини при сприйнятті віддалених об'єктів, як говорилося вище, може викликати так звану мініатюризацію: глядачеві здається, що він дивиться не на реальну сцену, а на її масштабну модель. Це є результатом діяльності мозку - так він реагує на паралакс, при якому далекі об'єкти виглядають більшими, ніж зазвичай очікує мозок: він вважає, що оскільки наше міжокулярна відстань не збільшилася раптом, то ці об'єкти повинні бути менше (і ближче), ніж могли б бути в реальному житті. Аналогічно, дуже мала міжосьова відстань - «гіпостереоскопія» - створить у глядача відчуття, що він зменшився в порівнянні зі сценою, і те, що зазвичай бачиться маленьким - комаха, наприклад, буде здаватися гігантським. Неважко помітити, що ці ефекти іноді можуть бути корисними з творчої точки зору, а часом навіть застосовуватися для розповіді частини історії в таких класичних дитячих фільмах, як «Джеймс і гігантський персик» і «Аліса в Країні Чудес». Однак поза сферою дитячого кіно ці ефекти будуть швидше відразливими, а не привабливими, тому потрібно подбати про те, щоб уникнути їх небажаної появи.

1.9 За межами кадру

Коли щось або хтось - актор, скажімо - виходить за межі кадру з одного боку екрану, то він зникне спочатку в одному ракурсі, а тільки потім - в іншому. Це не завжди проблема - якщо увагу аудиторії сконцентровано на чомусь в центрі екрану, наприклад, то вони можуть не помітити того, що відбувається на

його кордонах. Якщо ж глядачі дивляться на актора в міру того, як він іде, то вони можуть бути збиті з пантелику, коли актор зникає в одному з ракурсів. Більш того, це може навіть призвести до руйнування 3D-ефекту. Традиційним вирішенням цієї проблеми є додавання чорного поля уздовж кордону одного ракурсу з метою формування віртуальної «рамки» для сцени і відрізання частини сцени там, де одне око може побачити більше, ніж інше. Альтернативний спосіб прийшов з комп'ютерного світу і полягає в використанні різних міжосьових відстаней для різних частин сцени. У фільмі «Монстри проти прибульців», компанія Dreamworks ефективно впоралася з цією проблемою, зменшуючи стереоефект на персонажах у міру того, як вони виходили за межі кадру, забезпечивши їх видимість для обох очей настільки довго, наскільки можливо. Зрозуміло, це неможливо при живій зйомці дій, але в деяких обставинах може бути досягнуто під час монтажу за допомогою модуля InteraxialShifter і деяких прийомів грамотного багатопланового монтажу.

1.10 Колірна відмінність

Колірна відмінність між лівим і правим ракурсами також ускладнює глядачеві сприйняття їх як єдиної 3D-сцени. Щоб зробити два ракурси застосовуються різні камери і об'єктиви, і навіть якщо теоретично вони ідентичні, все одно можуть дати легку колірну відмінність між двома ракурсами, що формують стереопару. Колірна відмінність стає ще більш помітним при використанні знімальних платформ зі світло роздільним блоком, що дозволяє отримати менші міжосьові відстані, ніж це можливо при паралельній установці камер з великими об'єктивами, що не дають зменшити цю відстань нижче певного значення, що визначається габаритами обладнання. При використанні таких платформ світло, що формує два зображення, має різну поляризацію, внаслідок чого виникає локальна і глобальна колірна відмінність -

яскраві світила, можуть бути присутніми в одному ракурсі і повністю відсутність в іншому.

Наявний в Ocula модуль ColourMatcher забезпечує рішення для різних ситуацій. По-перше, алгоритм глобальної відповідності кольору спробує усунути колірну невідповідність між двома ракурсами і може бути ефективним для корекції невеликої глобальної різниці колірних спектрів, що виникають через розбіжність характеристик камер і оптики. А по-друге, ColourMatcher також містить метод на базі блоків, який більш ефективний при роботі з локальною - часто більш істотною - невідповідністю кольору, причиною якої є різна поляризація. В цьому режимі ColourMatcher задіє карту невідповідності для відома відповідних пікселів в кожному з ракурсів. Потім модуль ділить сцену на прямокутні області і намагається забезпечити збіг кольору кожної області одного зображення з відповідною областю іншого. Розмір області можна регулювати, якщо необхідно отримати найкращий результат. Цей локальний метод може бути дуже ефективним, навіть для перенесення таких деталей, як відблиски, з одного ракурсу в інший, в якому їх спочатку не було.

Висновки

Багатошаровий монтаж стереоскопічного 3D-зображення створює багато нових проблем навіть для дуже досвідчених фахівців. Наявність відповідних інструментів для роботи, таких як Nuke і Ocula, може допомогти подолати ці складності і досягти найкращого результату. Технологічний процес Nuke для роботи в режимі стерео може позбавити від зайвої роботи, пов'язаної з окремою обробкою двох потоків зображення, а не одного, а інструменти Ocula, такі як VerticalAligner і ColourMatcher здатні автоматизувати процедури, що раніше виконувалися вручну і забирали багато часу і сил. Крім того, технологія генерації карт невідповідності Ocula допомагає зберегти точну кореляцію між

лівим і правим ракурсами, завдяки чому аудиторія зможе насолоджуватися 3D-зображенням, а не страждати від нього.

2 3D-КОМПОЗИТИНГ В NUKE

Одна з найпотужніших функцій в Nuke це його 3D-модуль. Nuke має майже повнофункціональну 3D-систему, яка дозволяє імпортувати камери, створювати прості об'єкти, редагувати і імпортувати об'єкти, виконувати проєкції камер, витягувати різні типи 2D даних з 3D контенту та багато іншого. Важливо відзначити, що 3D модуль Nuke не є повноцінною заміною 3D програм. Nuke не має інструментів для анімації персонажів чи серйозного освітлення і рендерингу. Хоча можливості 3D Nuke є чудовими для системи композитинга, вони все ще далекі від того, щоб замінити повноцінні 3D пакети.

3D можливості Nuke створені для спрощення зв'язку відділу композитингу з 3D відділом. Крім того, якщо в сцені наявні 3D об'єкти, композер може отримати дані 3D трекінгу для використання у своїх цілях. Також 3D модуль Nuke використовують, щоб легше виконувати 2D трекінг інформації, спростити ротоскопінг і багато, багато іншого. Nuke добре працює із зовнішніми 3D-додатками і може обмінюватися даними з ними, використовуючи формати файлів, зазвичай використовуваних в 3D-додатках: Alembic, OBJ і FBX. Розширення .obj для файлів, які мають лише геометрію. Alembic і FBX типи файлів можуть містити практично все, що генерується в 3D сцені.

2.1 3D інсталяція сцени

У Nuke, 3D сцени побудовані з чотирьох основних елементів: камери, геометрії, вузла сцени (при необхідності), і вузла ScanlineRender для візуалізації даних 3D в 2D-зображення.

Камера: Через елемент камери, вузол рендеринга розглядає сцену. Також можна імпортувати властивість камери з FBX файлів.

Геометрія: Елемент геометрії може бути примітивною (простою геометрією) створеною в Nuke, такі як куб чи сфера, або може бути імпортовано з іншої програми у вигляді файлу OBJ, послідовності OBJ файлів (для анімації об'єктів), або файл FBX.

Сцена: Вузол Scene з'єднує всі частини, які складають сцену. Включає в себе всі частини геометрії, світло і камери. Вузол сцени говорить, що всі ці елементи знаходяться в одному просторі.

ScanlineRender: Вузол ScanlineRender приймає дані 3D і робить 2D зображення з нього. Без нього не можливо отримати 2D зображення.

Використовуючи ці елементи разом (рис 2.1), і деякі інші вузли програмного забезпечення Nuke, можна вирішити більшість проблем програм 3D моделювання. Інструменти для роботи з 3D-матеріалами знаходяться у вкладці 3D на панелі інструментів.

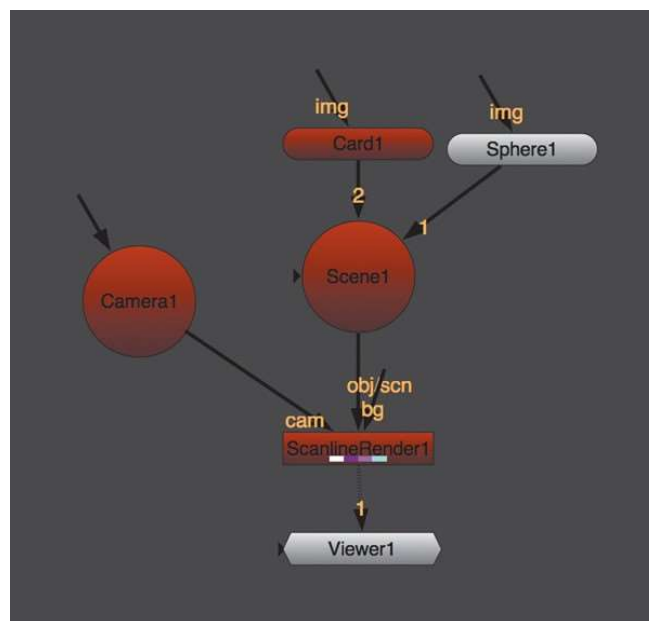


Рисунок 2.1 –Основні налаштування 3D сцени

2.1.1 Налаштування Nuke 3D сцени

Алгоритм створення віртуальної камери в Nuke:

1. Коли нічого не обрано у вікні, створити вузол камери з 3D-інструментів.
2. Коли нічого не обрано у вікні, створити вузол сцени з 3D-інструментів.
3. Обрати CAMERA1 і Scene1 і вставити вузол ScanlineRender з 3D-інструментів.

Вузол ScanlineRender з'єднає себе з вузлами камери і сцени в правильних входах (рис 2.2).

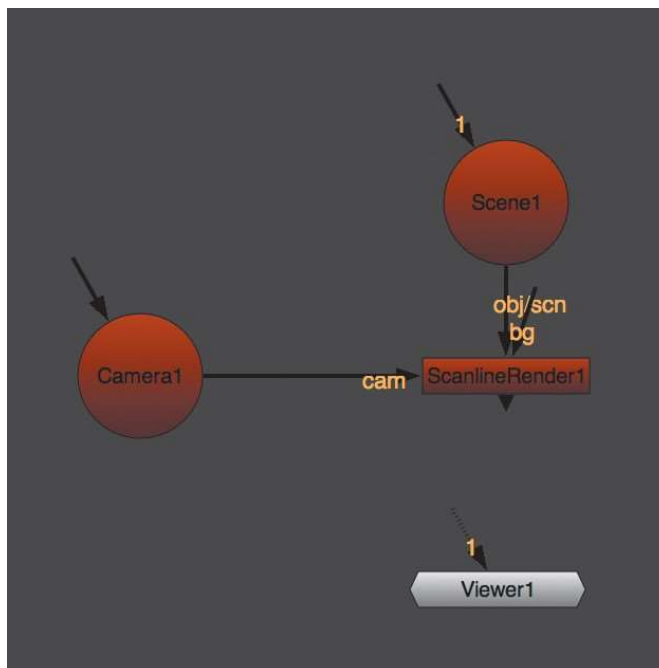


Рисунок 2.2 – Початок налаштування 3D сцени

4. Обрати ScanlineRender1 і натиснути клавішу 1, щоб переглянути його.

На даний момент у вікні перегляду нічого не видно в режимі 2D зображення. Проте, в 3D-світі вже існує камера. В Nuke існують функції як 2D і 3D Viewer. Між режимами перегляду 2D і 3D можна легко перемикатися при наведенні курсору миші на вікно перегляду і натиснувши клавішу Tab. Крім

того, можна обрати 2D або 3D (або інші види, такі як зверху і спереду) з меню, що випадає View Selection в правому верхньому куті вікна перегляду, як показано на рис 2.3.

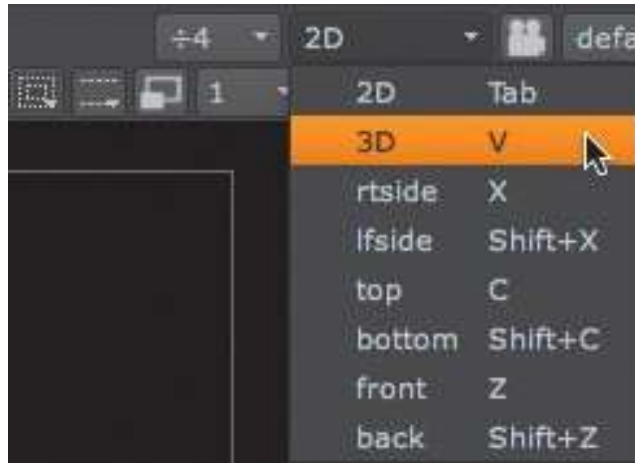


Рисунок 2.3 –Зміна видів у вікні перегляду

5. Навести курсор миші на вікні перегляду і натиснути клавішу Tab для переходу в 3D режим (рис 2.4).

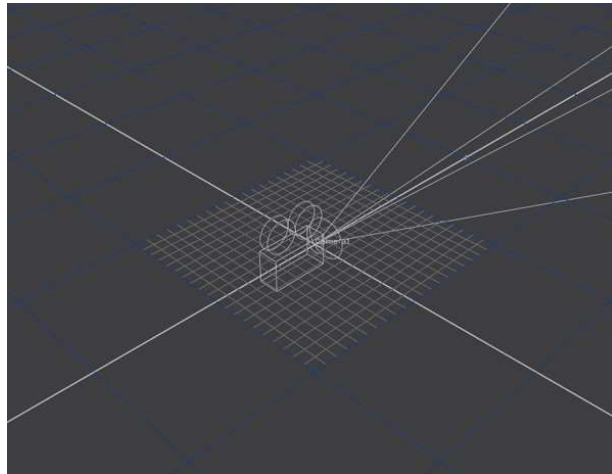


Рисунок 2.4 –Перегляд 3D камери в Nuke

Камера знаходиться всередині віртуального світу в “0, 0, 0” координатах. Всі 3D елементи формуються на цій позиції. Саме цією камерою виконується перегляд 3D сцени.

2.1.2 Переміщення в 3D сцені

Наступні гарячі клавіші і методи використовуються для навігації в 3D вигляді:

Ctrl / Cmd + ліва кнопка миші для повороту в сцені.

Alt / Option + ліва кнопка миші для переміщення в сцені.

Колесо прокрутки (якщо є) або клавіші +/- для збільшення і зменшення масштабу.

Переміщення камери в сцені здійснюється за рахунок переміщення її осі координат. Двічі клацнувши по камері активується вісь в режимі перегляду. Осі мають червоний, зелений і синій кольори, відображаються в центрі камери і дозволяють переміщати його (рис 2.5).

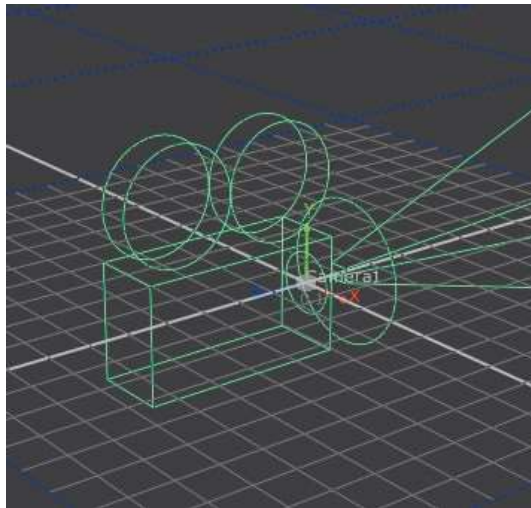


Рисунок 2.5 –Переміщення 3D камери в Nuke

Як правило, червоний, зелений, синій відповідає осям X , Y , Z . Іншими словами, червоний колір контролює напрямок X , зелений контролює напрямок Y , а синій Z відповідно.

Аналогічна ситуація справедлива для обертання камери. У вікні перегляду, утримуючи Ctrl / Cmd відобразиться елемент керування обертання (рис 2.6).

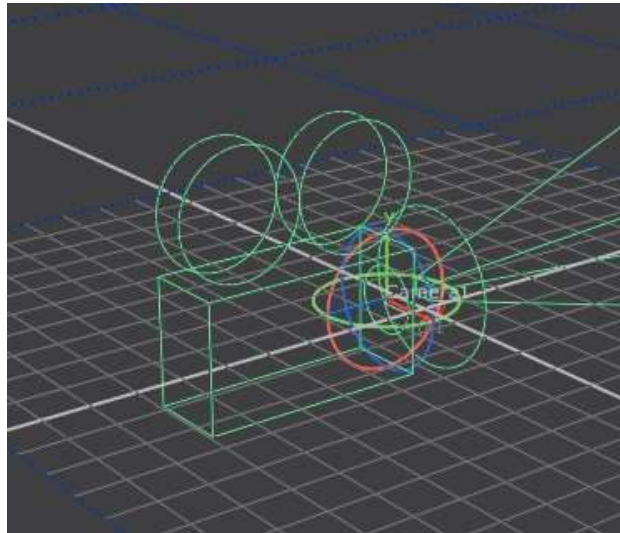


Рисунок 2.6 – Обертання камери

Кола, які з'являються на осі при утриманні клавіши Ctrl дозволяють обертати камеру навколо однієї з трьох осей.

2.2 Використання Nuke для зберігання групи інструментів

В Nuke є можливість створити власну панель інструментів в якій будуть зберігатися не тільки вузли, але й зв'язки між вузлами, і кожна властивість в кожному з вузлів (а також анімації, вирази...).

Алгоритм створення власного ToolSet.

1. Клацнути правою кнопкою миші на панелі властивостей CAMERA1 і обрати Set Knobs to Default для перезапуску камери.
2. Створити область виділення за допомогою миші, обрати

CAMERA1, СЦЕНА1 і ScanlineRender1 (рис 2.7).

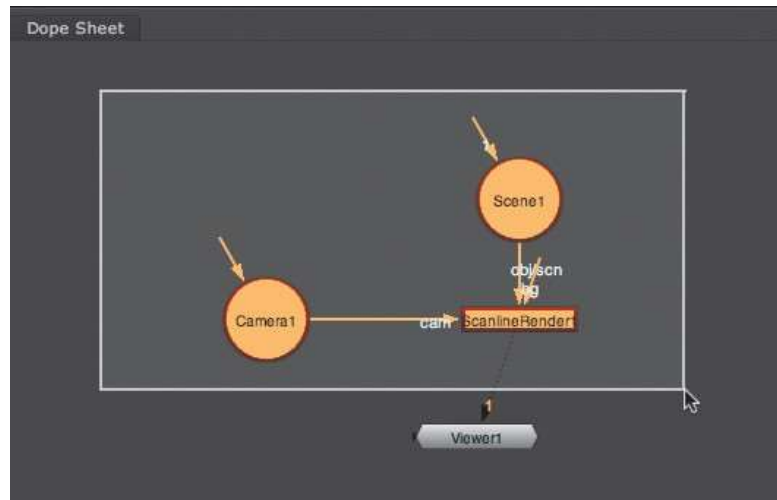


Рисунок 2.7 –Виділення елементів для ToolSet

3. Обрати «Створити» з панелі інструментів.

На рис. 2.8. показано панель створення ToolSet. В ній потрібно задати ім'я своєму Toolset, а також можна створити підменю. Створення підменю називатиметься 3D і в ньому називають цю установку Toolset.



Рисунок 2.8 –Створення ToolSet панелі

4. В поле меню обрати пункт 3D / Setup.
5. Натиснути кнопку Створити.
6. Тепер панель інструментів має такий вигляд (рис. 2.9).

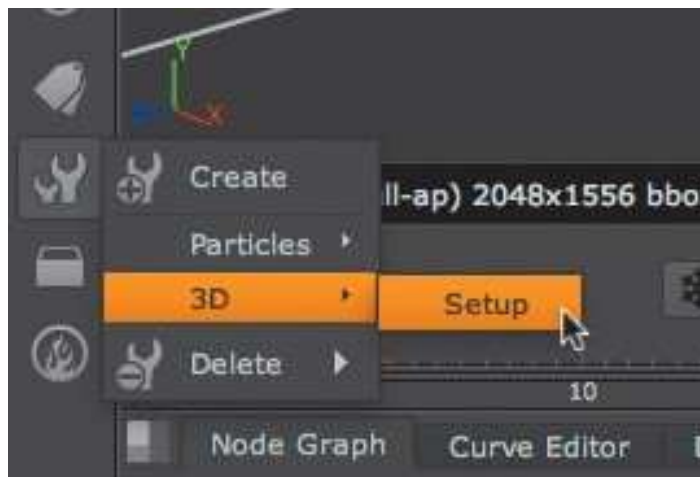


Рисунок 2.9 –Новий, власний ToolSet всередині нового підменю

Для того щоб видалити Toolset, можна використати підміню Delete, показане на рис. 2.10.

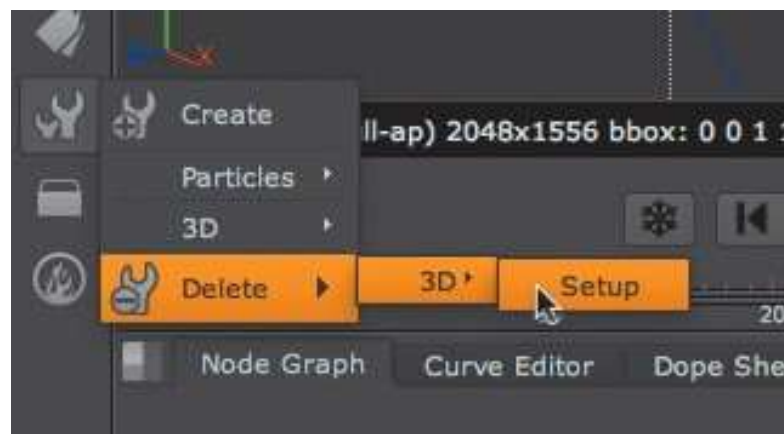


Рисунок 2.10 –Видалення Toolset

Тепер данну будову вузлів можливо створити завдяки одному клікові миші.

7. У меню Файл виберіть команду Очистити.

Команда Clear просто починає новий порожній скрипт, не ставлячи ніяких питань.

2.3 ПЕРЕГЛЯД 2D зображення в якості 3D об'єкта

Для прикладу була використана відрендерена 3D сцена. Коли істота вставляє свою голову між прутами решітки, додамо деяких магічних метеликів які вискочать з-під зазначеної ґратки. Для реалізації цього ефекту була використана система частинок. Система частинок являє собою метод, який відтворює хаотичні системи, випускаючи багато об'єктів і управління ними з фізичними похідними силами. Системи частинок досить просунутий матеріал, вони доступні тільки для власників NukeX.

Для того щоб помістити систему частинок в 3D-світі, повертаючи 2D зображення, яке було додано в 3D-об'єкт потрібно використовувати пас світового положення.

2.3.1 Використання пасу світового положення для створення 3D-об'єкта

Пас світового положення - це пас, який рендериться в 3D програмному забезпеченні який не відображає візуальної частини зображення, натомість він має певні переваги. Пас світового положення відображає де знаходиться кожен піксель зображення відповідно до нульової координати (0,0,0). Використовуючи цей пас і ноду `PositionToPoints` можна створити хмару пікселів які будуть розташовані на своїх правильній позиція в 3D світу.

1. У вікні перегляду, переключитися на перегляд безлічі каналів `WorldPos` (рис. 2.11).

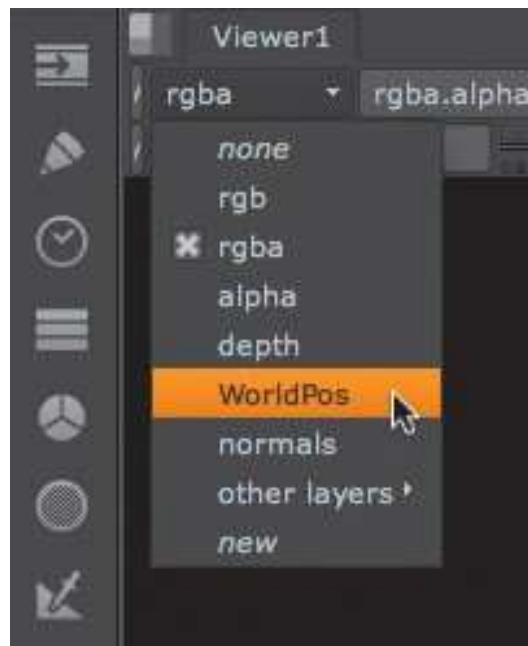


Рисунок 2.11 –Перегляд набору каналів у вікні перегляду

2. Навести покажчик миші у вікно перегляду, в правому нижньому кутку побачити значення пікселів (рис. 2.12).

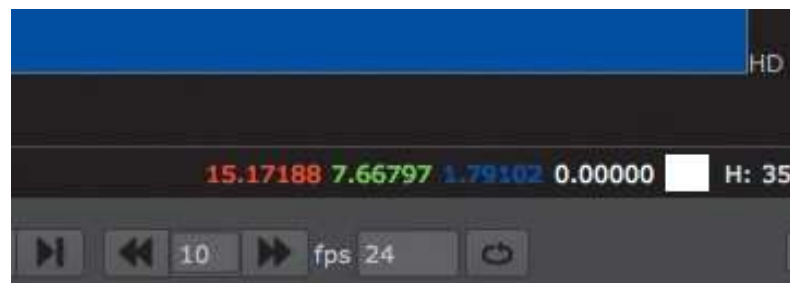


Рисунок 2.12 –Значення пікселів у вікні перегляду

Прохід світового положення створює дуже яскраве зображення. Ці кольори не мають нічого спільного з візуальною частиною зображення. Значення кольорів показують положення пікселів в 3D-просторі. Значення RGB відповідають координатам XYZ відповідно. Курсор на рис. 2.12; розташований на $X = 15.17$, $Y = 7,67$ і $Z = 1,79$.

Ще один пас, який є менш важливим в цьому випадку, називається пасом нормалей. Пас нормалей дуже схожий на пас світової позиції з відмінністю в тому, що пас позицій являє собою пікселі в просторі, а нормалі являє собою кут пікселя в світових координатах.

3. Перейти на перегляд RGBA каналів у вікні перегляду.
4. З READ1 обраний в DAG, вставити вузол PositionToPoints з панелі інструментів 3D / геометрії.

Глядач не перейде до 3D виду, так як більше не дивиться на зображення. Зображення, яке було у вікні перегляду тепер 3D-об'єкт, це буде видно якщо змінити пару властивостей.

5. На панелі PositionToPoints Properties обрати WorldPos з меню, що випадає Surface Point. Для нормалей поверхні, обрати нормалі.

Можете відразу побачити, що деякі пікселі з'явилися у вікні перегляду. Це насправді не пікселі, але малі 2D площину, як шматочки паперу, що плавають в 3D-просторі (рис 2.13).

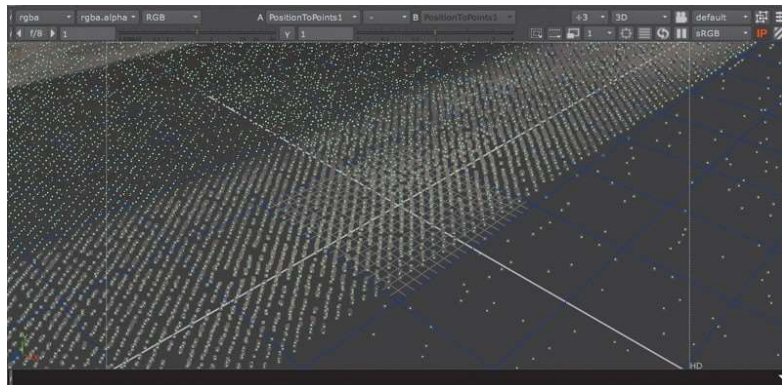


Рисунок 2.13 – Положення пікселів в 3D-просторі

Переміщуючи контролер в 3D просторі, ймовірно, можна побачити зображення цілком досить добре. Але в будь-якому випадку, це зображення повинно бути видно з конкретної камери.

2.3.2 Імпорт камери

Для виконання 3D композитингу необхідно мати камеру, яка б вела себе аналогічно до тієї яка знімала цю сцену. Для імпорту камери зі сторонніх джерел необхідно виконати такий алгоритм дій. Перш за все, необхідна налаштована 3D сцена.

1. Нічого не вибрано в DAG, обрати 3D / Налаштування на панелі інструментів ІНСТРУМЕНТІВ (рис. 2.14).

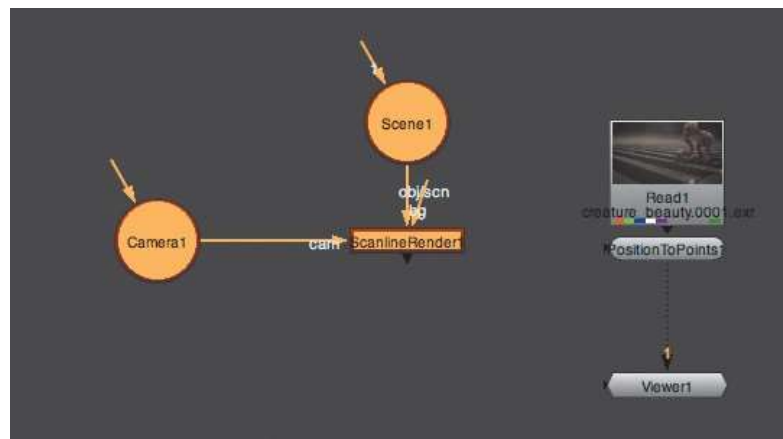


Рисунок 2.14 – Використання власного набору інструментів.

2. Двічі клацнути CAMERA1, щоб відобразити її панелі властивостей. Найвища властивість називається читанням з файлу. Необхідна щоб використати вкладку Файл виконується читання розташування камери і інші властивості з файлу (рис. 2.15).

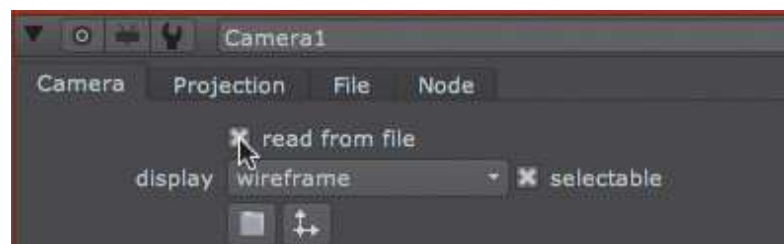


Рисунок 2.15 – При натисканні на це вікно дозволяє читати Файл

3. Обрати Читання з файлу.
4. Перейти на вкладку Файл в панелі властивостей CAMERA1 (рисунок 2.16).

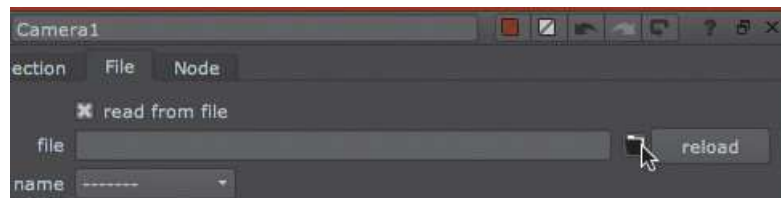


Рисунок 2.16 – CAMERA1 Вкладка Файл

5. Щоб імпортувати файл камери, необхідно клацнути значок папки праворуч від властивості файлу, щоб завантажити браузер файлів.
6. У браузері файлів перейти в каталог з Camera.fbx.
7. З'явиться діалогове вікно із запитом, щоб знищити дані поточної камери; натисніть кнопку Так.

Файл FBX може містити багато властивостей камер в різних дублях. З двох розкритих меню, необхідно обрати ім'я вузла. Вони, як правило, включають багато пресетів за замовчуванням, які зазвичай не використовують, в залежності від того, яка програма використовується при експорті файлу камери.

8. З меню, що випадає Take Name, обрати Take 001.
9. З меню, що випадає Ім'я вузла, обрати S26A_Camera (рис. 2.17).

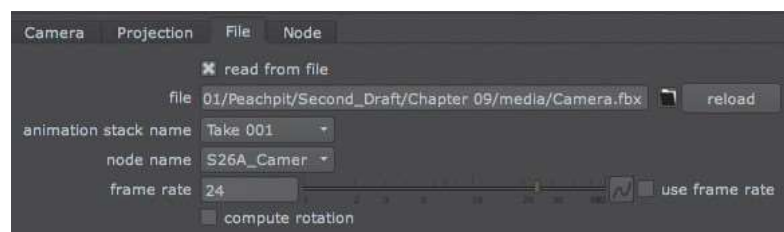


Рисунок 2.17 – Меню імпорту камери

10. Перейти на вкладку Camera.

Можна бачити, що поля переміщення і повертання вже заповнені значеннями, і всі вони неактивні та недоступно для зміни. Так здійснюється анімація камери. Якщо файл на диску зміниться значення в цих полях також зміняться. Камера зникла з вікна перегляду. Для того, щоб побачити її знову, необхідно виконати наступне:

11. Обрати CAMERA1 в вузлі Graph.

12. Навести курсор миші на вікно перегляду і натиснути клавішу F, щоб обрізати перегляд для обраного елемента (рис. 2.18).

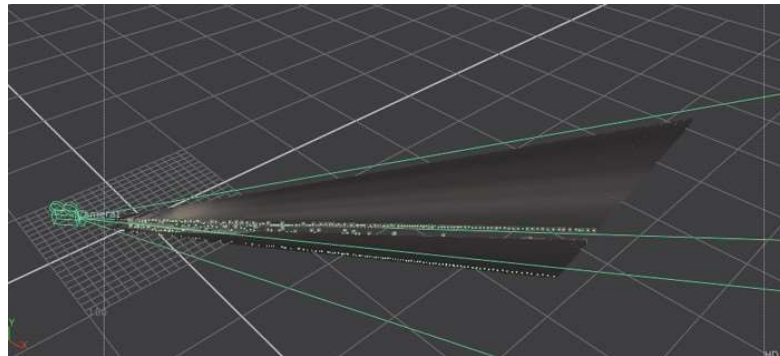


Рисунок 2.18 –Камера після імпорту

Існує можливість візуально побачити зв'язок між камерою і точками, створеними з вузлом PositionToPoints. Частинки існують тільки в області поля зору камери. Для перегляду сцени з імпортованої камери.

13. З перегляду камери в спадному меню в правому верхньому куті вікна перегляду, як видно на рис. 2.19 обрати CAMERA1.

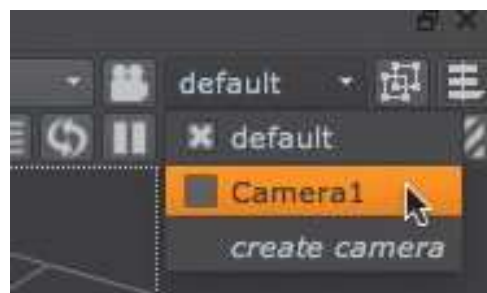


Рисунок 2.19 –Вибір камери для перегляду

Зараз видно точки зображення під правильним кутом з CAMERA1.
(рис. 2.20).



Рисунок 2.20 – Всі ці точки разом складають зображення

В цьому зображенні з'являється багато прогалин, це нормально адже за допомогою цього методу можуть бути відтворені точки які знаходяться в полі зору камери. Далі отриману інформацію можна використати для розташування об'єктів у просторі.

2.4 Маніпулювання 3D ДЕРЕВА NODE в 3D просторі

2.4.1 Імпорт системи частинок

Система частинок зберігається як інший сценарій Nuke, його можна об'єднати за допомогою команди Import Script в меню Файл.

1. У меню Файл обрати команду Імпорт сценарію.
2. Перейти в папку з частинками і двічі клацнути по Particles.nk.
3. В програмі Nuke, можна отримати повідомлення про те, що ви

використовуєте інструменти NukeX. (рис. 2.21).



Рисунок 2.21 –Запуск інструментів NukeX в Nuke генерує це повідомлення

NukeX не має такого повідомлення і можна змінити налаштування будь-якого з вузлів частинок пізніше.

Імпортовані вузли можна з легкістю переміщати оскільки вони одразу являються обраними в сцені.

4. Переміщення вузлів з розташуванням інших вузлів (рис. 2.22).

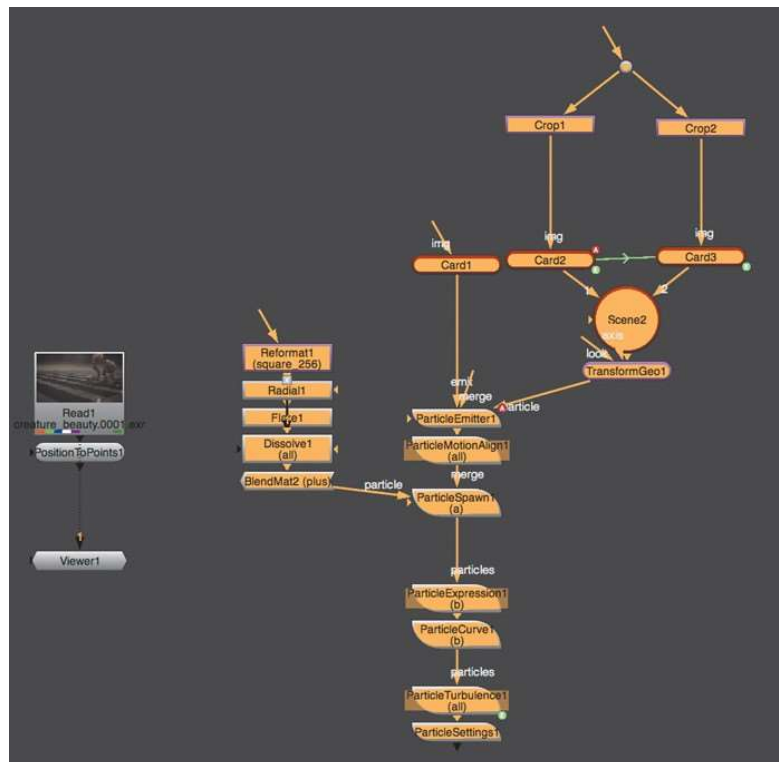


Рисунок 2.22 –Дерево вузлів що генерує частинки

В даному дереві вузлів не вистачає одного елемента, а саме зображення частинки.

5. На самому верху імпортованого дерева, над двома вузлами є вузол точки. Підключити вихід read2 до входу цієї точки (рис 2.23).

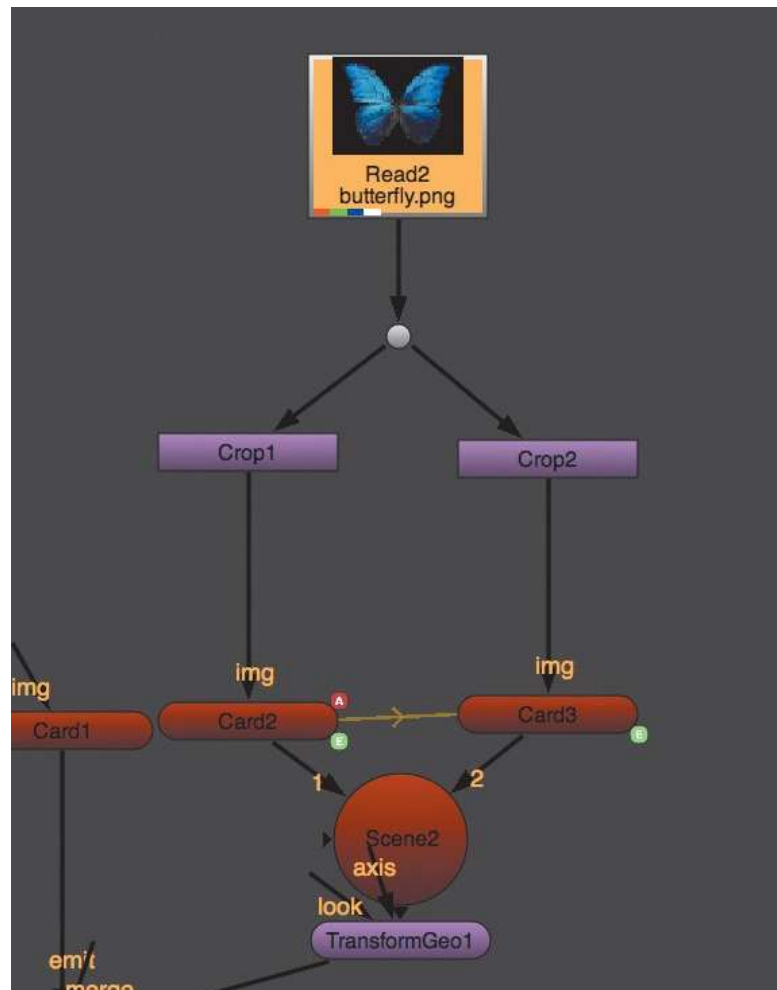


Рисунок 2.23 –Підключення зображення метелика

Основні властивості і можливості дерева вузлів для генерації частинок в 3D просторі програми Nuke.

- Два вузли Crop під Read3 розділяють метелика на ліву половину і праву половину.
- Card2 і Card3 дві плоскі 3D поверхні. Вони являють собою

паперовий тонкий об'єкт в 3D-просторі. Зображення метелика (ліве і праве) текстури цих поверхонь. До цих нодів підключені вирази які змушують їх рухатися в синхронному порядку.

- СЦЕНА2 з'єднує дві частини метелика і TransformGeo1, який являє собою 3D-вузол трансформації, який дозволяє обертати комбінацію з двох вузлів карти, щоб краще відображалися перед камерою.

6. Щоб переглянути це в вікні перегляду необхідно двічі клацнути TransformGeo1, і натиснути 1.

- Card1 є джерелом частинок, що випромінюються. Також його називають випромінювачем або пістолетом.

- ParticleSettings1 останній вузол в цьому дереві який являється виходом з цього дерева.

7. Для перегляду анімації руху метеликів необхідно натиснути Play (рис. 2.24).

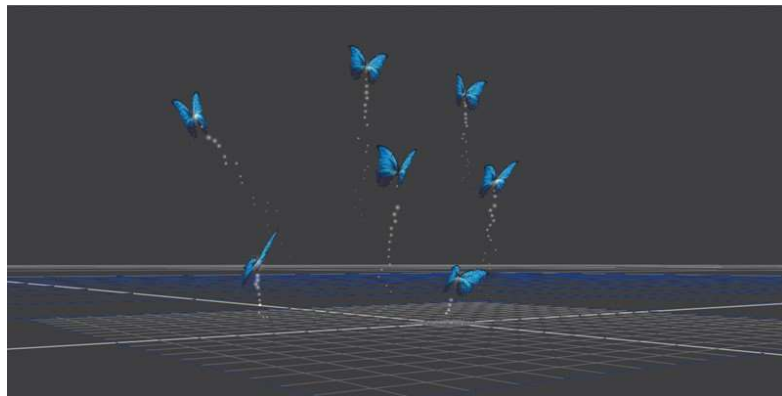


Рисунок 2.24 –Анімація метеликів створена за допомогою системи частинок

Важливо відзначити кілька речей: система частинок створюються в світового центру і починають випускатися з кадру 1.

2.4.2 Суміщення 3D-об'єктів в просторі

Вузол TransformGeo так само, як вузол 2D Transform дозволяє переміщувати об'єкти в просторі. Єдина відмінність полягає в тому, що він призначений для переміщення об'єктів або геометрії, а не 2D-зображень. Це також означає, що він рухається не тільки по двох осях, як вузол 2D Transform, але по трьох. Для суміщення 3D- об'єктів необхідно:

1. Обрати ParticleSettings1 і вставити вузол TransformGeo після нього.

Даний вузол створює вісь, яка переміщує всю систему частинок в 3D-просторі. Елементи управління є такими ж, як для камери. Можливо використовувати осі для переміщення, і утримуючи Ctrl / Cmd, для повороту.

2. Двічі клацнути PositionToPoints1 і CAMERA1.

3. З меню перегляду камери в правому верхньому куті вікна обрати CAMERA1.

4. Обрати TranformGeo2 в DAG.

Тепер можна бачити вісь управління системою частинок в засобі перегляду відносно істоти (рис 2.25). Необхідно помістити систему частинок саме там, де голова істоти знаходиться між прутами решітки.

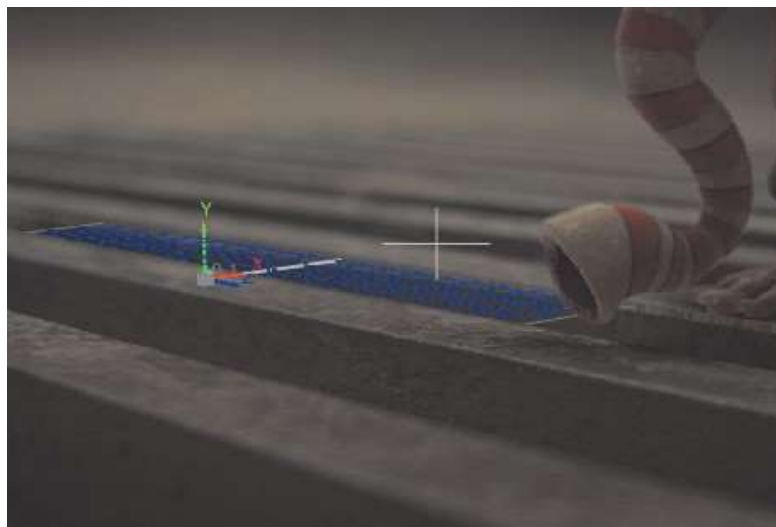


Рисунок 2.25 –Положення вісі частинок відно 3D простору.

5. Перейти до кадру 40 в TimeBar.

Голова істоти вставляється в цю решітку. Необхідно розташувати TransformGeo2 прямо в цьому кадрі.

6. Перемістити TransformGeo2 туди, де знаходиться голова істот.

Такі координати голови істоти $X = 2.2$ і $Z = 11,7$. Проте, метелики повинні випускатися з-під ґратки, тому необхідно перемістити систему частинок вниз.

7. Використання панелі властивостей TransformGeo2 для зміни Y координати на -2.

8. Перейти до кадру 90 в TimeBar.

За замовчування елементи системи частинок досить крихітні, щоб їх було добре видно в кадрі необхідно збільшити їх розмір.

9. Повернутися до кадру 40 в TimeBar.

10. В панелі властивостей TransformGeo2, змінити Uniform Scale 5.

Результат зміни розміру частинок можна спостерігати на (рис 2.26).

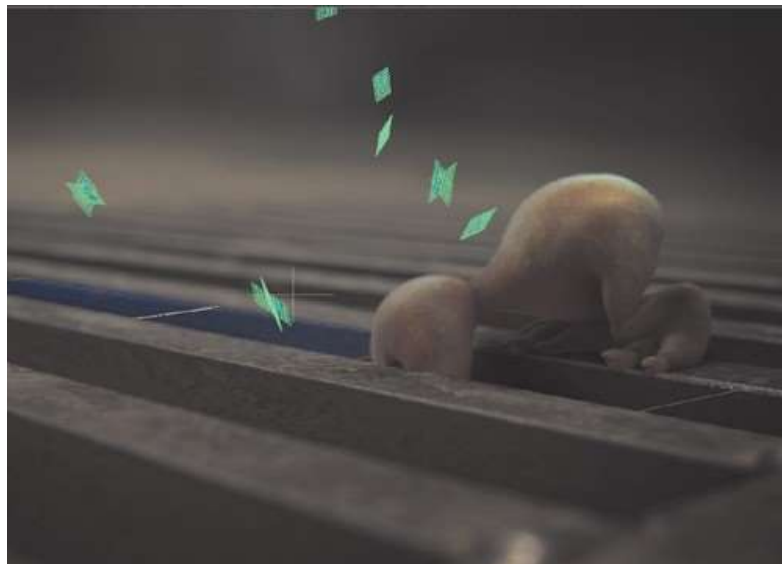


Рисунок 2.26 – Метелики в правильному місці і розмірі.

2.5 Переміщення 3D-об'єктів в часі

Тепер необхідно розташувати систему частинок в правильному місці в часі. Всі вузли пов'язані з часом знаходяться в Time інструментів. Деякі з цих вузлів не тільки працюють над кліпами зображень, вони також працюють з геометрією.

Вузли, які діють на геометрію FrameRange, Ofset і FrameHold. Для переміщення 3D фігур в часі застосовують Ofset. Обрати TransformGeo2 в DAG і вставити вузол Ofset після нього від часу набору інструментів (рис 2,27).

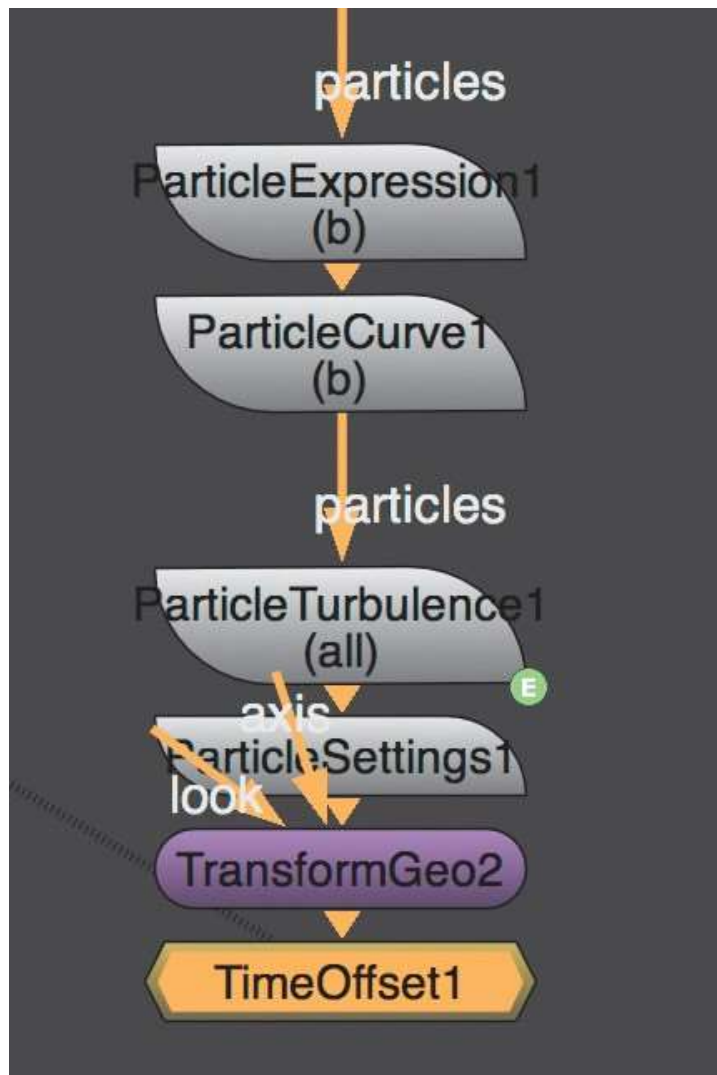


Рисунок 2.27 – Offset вставляється після TransformGeo2

В даному випадку необхідно змістити початок створення частинок на 30 кадрів вперед. Для цього в панелі властивостей TimeOffset1, в розділі Time Offset необхідно задати значення введіть 30 в поле введення. Тепер елементи частинок створюються в правильний час.

2.6 Переведення 3D об'єктів в 2D пікселі

В даний момент в полі DAG існує три основні дерева композиції. Для зручності вони згруповані у фігури як показано на рис. 2.28.

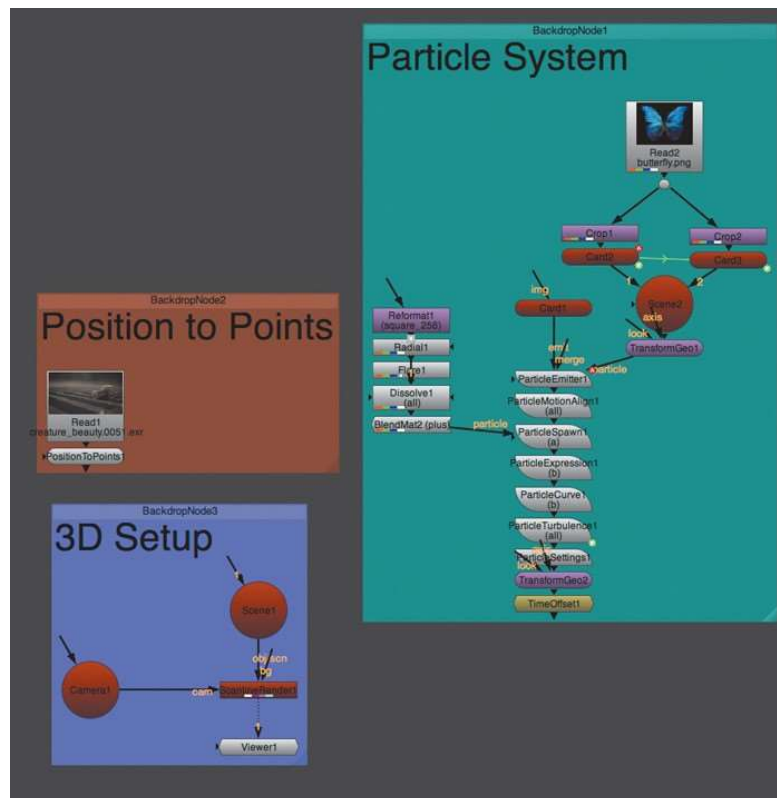


Рисунок 2.28 –Три дерева вузлів

Область по праву сторону це система частинок. Область зліва є посилання на об'єкт що генерується вузлом PositionToPoints. Область що знаходиться в

нижній частині є налаштуванням 3D сцени. Наявність окремих дерев дає багато свободи дій і гнучкості. Для підключення частинок необхідно:

1. Перетягнути вихід TimeOffset1 на вхід СЦЕНИ1.
2. Обрати ScanlineRender1 і натисніть 1, щоб переглянути його у вікні перегляду.

Вузол ScanlineRender виконує рендеринг 3D сцени за таким же принципом що зовнішнє 3D програмне забезпечення, 3D-геометрію за допомогою камери і 2D-зображення, яке можна маніпулювати в Nuke з будь-яким 2D вузлом, як Merge, Grade і Blur.

Можна виконати композитинг композиції з метеликами поверх зображення фону, це було б не можливо якщо б не перетворили геометрію в 2D-зображення.

3. Обравши ScanlineRender1, натиснути M для вставки вузла Merge.
4. Підключити вхід Merge1 в В для READ1 і перегляду Merge1 у вікні перегляду.
5. Поверніться до кадру 1 і натисніть кнопку відтворення, щоб подивитися систему частинок, як 2D-зображення (рис 2.29).

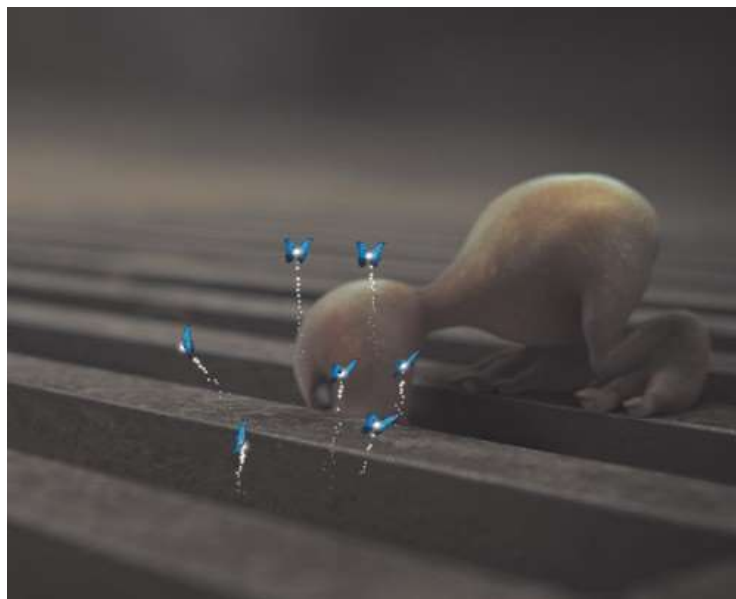


Рисунок 2.28 –Метелики як 2D-зображення, скомпоновані над фоном

В цьому зображенні необхідно виправити три пункти. По перше метелики мають зовсім іншу кольорову гаму зображення. По друге вони мають з'являтися з під ґратки люку. І по третє деякі з них мають леті за істотою. Для виправлення цих недоліків необхідно:

6. Натиснути кнопку Стоп і перейти в кадр 50.
7. Обрати нічого, натиснути Р, щоб створити вузол RotoPaint.
8. Помістити курсор миші у вікні перегляду і натиснути V, щоб почати малювати фігуру Безьє. Намалювати уздовж решітки, як показано на рис 2.30.

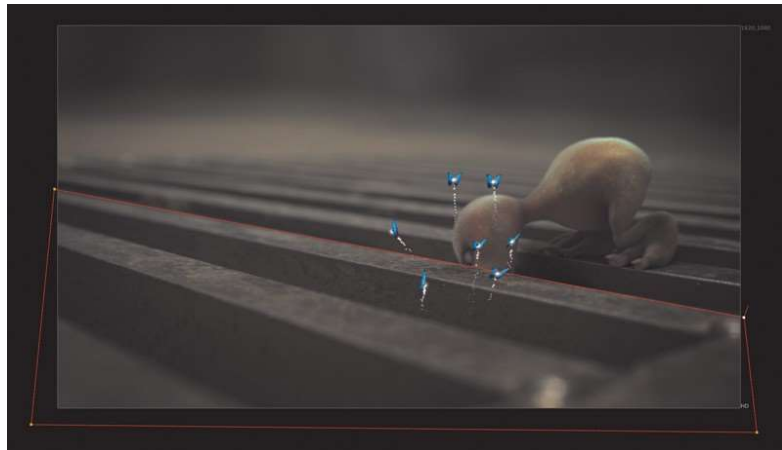


Рисунок 2.30 – Створена маски вузлом RotoPaint.

Тепер застосуємо RotoPaint вузол з маскою, для маскування метеликів.

9. Обрати RotoPaint1 натиснути М, щоб вставити інший вузол Merge.
10. Помістити Merge2 зв'язку між ScanlineRender1 і Merge1.
11. З меню, операцій що випадає Merge2, обрати команду трафарет.

Рис. 2.31 показує метеликів які маскуються за ґраткою і вже здається, що вони надходять з-під підлоги. Край маски є трохи різким.



Рисунок 2.31 –Замасковані метелики

12. Обрати RotoPaint1 і натиснути кнопку B, щоб вставити вузол Blur.
13. Встановити значення Blur1 на 3 пікселі.

Тепер край зображення не виглядає таким різким. Тепер дерено нодів даної композиції виглядає так як показано на рис. 2.32.

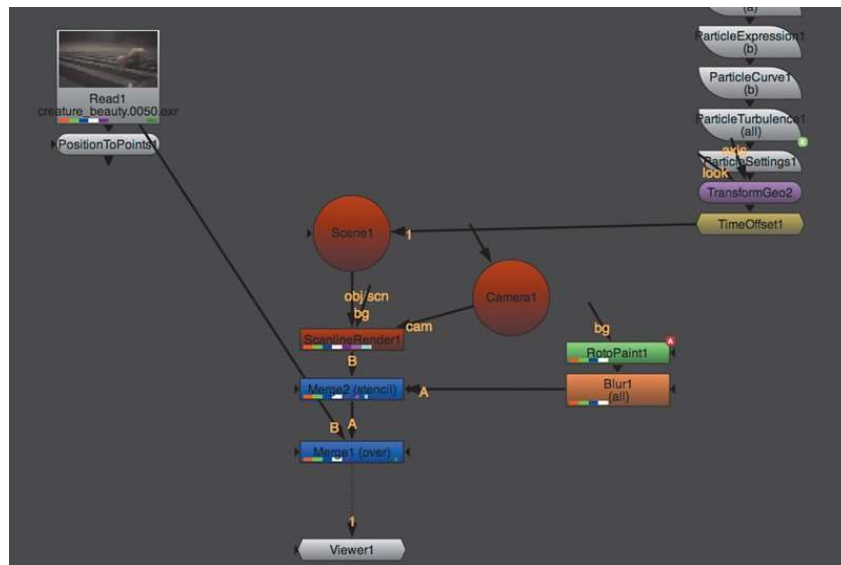


Рисунок 2.32 –Дерево нодів після маскування метеликів

2.7 Застосування матеріалів до об'єктів

Помістити частину метеликів перед об'єктом, а іншу частину за об'єктом є досить складною задачею. Для цього необхідно виконати наступне. В сцені вже існує об'єкт де сидить істота і його координати можуть бути використані в якості маски-істоти зроблені з точок. Для цього:

1. Підключити вихід з PositionToPoints1 до СЦЕНИ1 (рис 2.33).

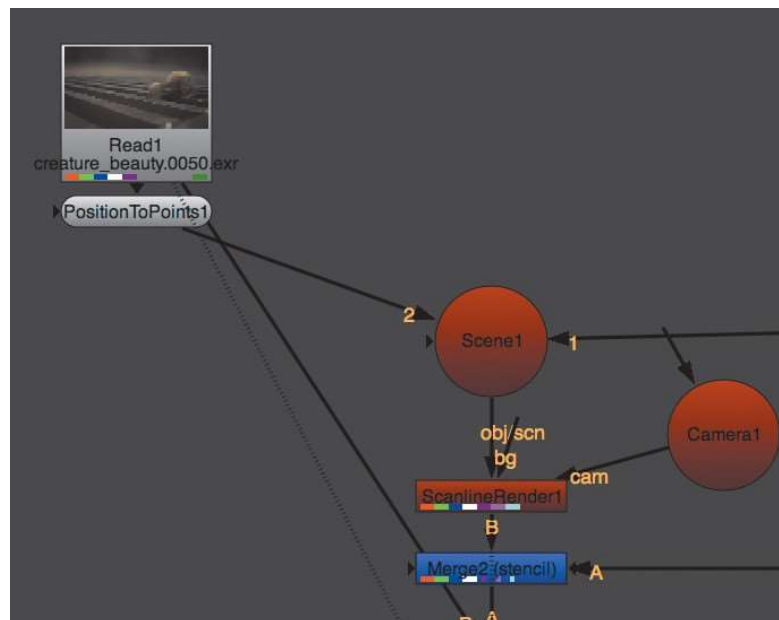


Рисунок 2.33 – Підключення іншого об'єкта до сцени

Тепер, коли PositionToPoints1 підключений до СЦЕНИ1, як точки об'єкта істоти і система частинок існує в тому ж просторі, вони будуть рендеритися разом через вузол ScanlineRender.

2. Обрати READ1 і натиснути 2, щоб переглянути його в другому вході Viewer.
3. Обрати ScanlineRender1 і натиснути 1, щоб переглянути його в першому вході Viewer.

4. Перемикання між входами 1 і 2 відбувається при наведенні покажчика миші у вікно перегляду і натисканні клавіш 1 і 2.

Після виконання цих дій деякі метелики дійсно зникають позаду голови істоти. Виникає одна проблема, що все відображене на вході 1 є товстішим ніж потрібно. Це відбувається тому, що вузол `PositionToPoints` не дає точної геометрії. Точки, створюють не правильний розмір об'єктів. Можна отримати точніше зображення, використовуючи `Point Size` властивість `PositionToPoint` (рис. 2.34), але `PositionToPoints` просто не розраховані і не призначені для цього. Можна і надалі використовувати необроблений `READ1` в якості фону, але потрібно використовувати `PositionToPoints1` як маскою в 3D-просторі.



Рисунок 2.34 –Розмір `PositionToPoints`

Об'єкт, як правило, має матеріал, примінений до нього. Вузли карт, складові метелики мають зображення метелика покладені на них в якості матеріалу; зображення метелика підключене до входу `Img` вузлів карт (рис. 2.35). Якщо не призначити матеріал, на них буде пусте зображення, що можна побачити в 3D-перегляді, але нічого не буде рендеритися через вузол `ScanlineRender`.

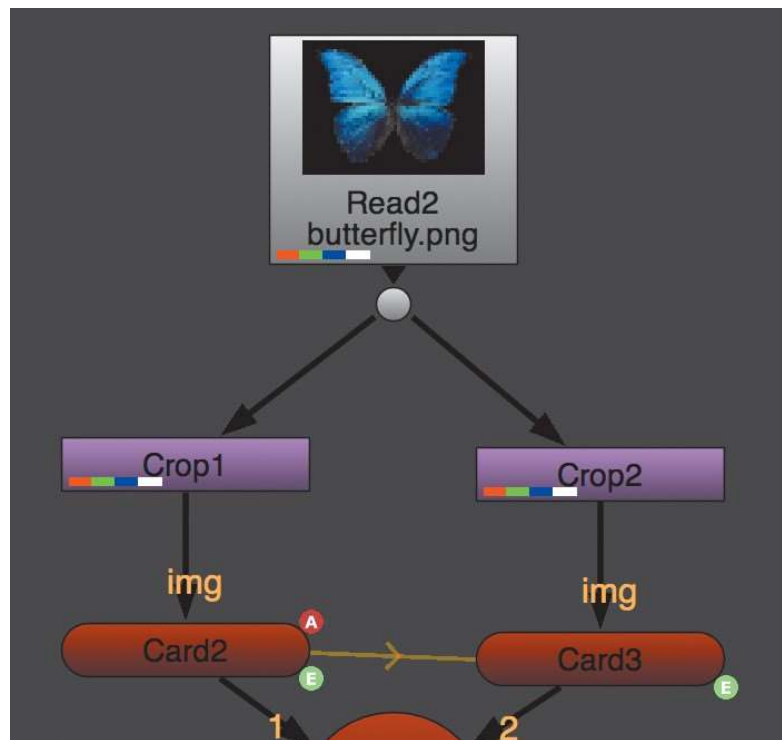


Рисунок 2.35–Ноди зображень підключені в якості матеріалів

Крім простих матеріалів, таких як зображення, деякі більш складні матеріали мають більший контроль над тим, як вони реагують на світло; вони можуть бути блискучими або випромінювати світло, і вони можуть контролювати колір своєї тінь, наприклад. Всі ці матеріали можна знайти в 3D> Shader (Шейдер це ще один термін для матеріалу).

Одним з таких матеріалів називається FillMat. Сенс цього матеріалу або шейдера повернути геометрію до якої він призначений у маску для іншої геометрії в сцені. Оскільки PositionToPoints не має входу Img, необхідно використати інший вузол з 3D> Shader інструментів під назвою ApplyMaterial.

5. Обрати PositionToPoints1 і з панелі інструментів 3D> Shader обрати ApplyMaterial.
6. Обрати нічого і створити FillMat з 3D> Shader.
7. Підключити вихід FillMat1 до Mat входу ApplyMaterial (рис. 2.36).

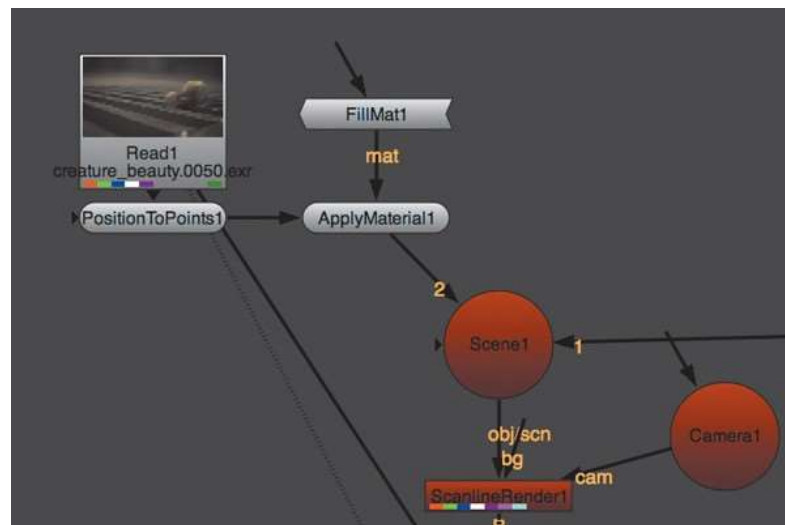


Рисунок 2.36 – Підключення матеріалу за допомогою вузла ApplyMaterial

8. Перегляд ScanlineRender1 у вікні перегляду (рис 2.37).



Рисунок 2.37 – Істота маскує деяких метеликів

В даний момент не видно істоти на виході ScanlineRender1.

9. Перегляд Merge1 у вікні перегляду (рис. 2.38).



Рисунок 2.38 –Вигляд композиції з маскою істоти

2.8 Корекція кольору зображення

Таким чином, вирішені майже всі проблеми 3D-композиції. Необхідно зробити метеликів максимально наближеними по відношенню кольору сцени і зробити їх не такими різкими, для цього:

1. Вставити вузол Grade між ScanlineRender1 і Merge2.
2. Для входу ScanlineRender1 обрати RGBA.alpha з (Un) premult.
3. Змінити колір метелика на більш теплий і можливо, трохи яскравіший. Вихідні значення: $R = 1,425$, $G = 1,18$, $B = 0,85$.
4. Вставити вузол Blur після Grade1.
5. В панелі властивостей Blur2, змінити розмір до 2.
6. Закрити всі панелі властивостей, натиснути кнопку відтворення у вікні перегляду. (рис. 2.39).

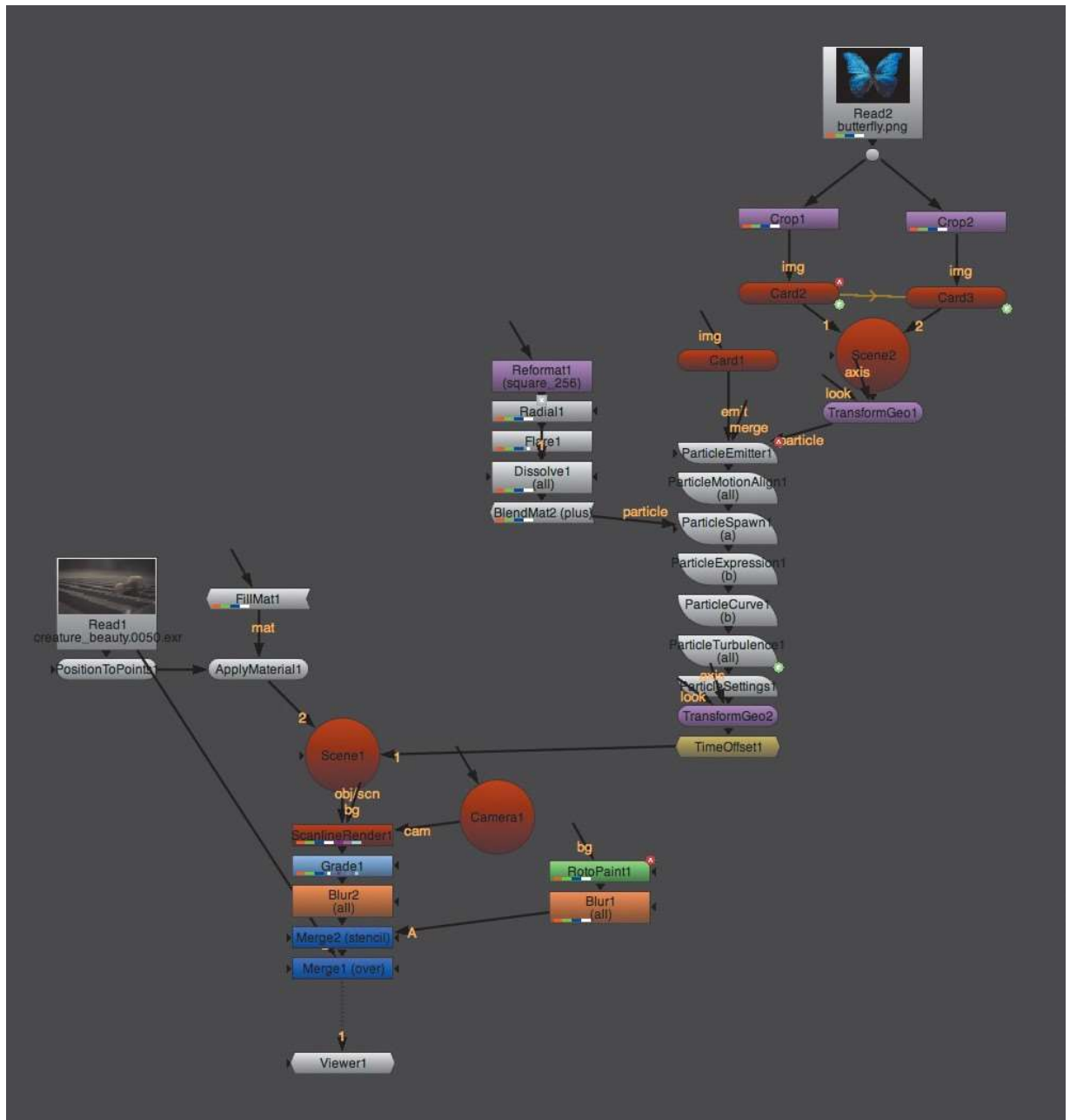


Рисунок 2.39 –Готове дерево нодів 3D сцени

Висновки

В другому розділі дослідження наведені наочно основні проблеми, які пов'язані зі створенням 3D-композиції в програмному забезпеченні Nuke. Так окремо, наведено не лише такі традиційні операції, як вибір камери в 3D-режимі, інсталяція сцени, рух в сцені, суміщення 3D об'єктів в просторі, але й показано у детальному розрізі весь алгоритм дій таких процедур як, наприклад,

переведення 3D-об'єктів в 2D пікселі. Окремо в розділі побuована модель в середовищі Nuke, яка дозволяє розробникам в сфері кіно провести тестування можливостей програми при створенні фільмів.

3 DEEP COMPOSITING

Глибокий композитинг являє собою спосіб композитинга цифрових зображень з використанням даних в іншому форматі стандартної «плоскої» композиції. Як випливає з назви, глибокий композитинг використовує додаткові дані про глибину. Це зменшує необхідність повторного рендеринга, дозволяє отримати високу якість зображення, і допомагає вирішити проблеми, пов'язані з артефактами по краях об'єктів.

3.1 Про Deep Compositing

Стандартне 2D зображення містить одне значення для кожного каналу кожного пікселя. На противагу цьому, глибокі зображення містять безліч вибірок на піксель на різних глибинах, і кожен зразок містить попіксельну інформацію, таку як колір, прозорість і глибину відносно камери.

Наприклад, створення масок об'єктів, які були додані в сцену пізніше вимагало повторного рендеринга фону і часто викликало купу проблем з прозорими пікселями і згладжуванням. З набором Деер композитинга вузлів Nuke, з'являється можливість візуалізувати фон один раз, а потім перемістити об'єкти в різні місця глибини, без необхідності повторної візуалізації фону. Будь-які прозорі пікселі також працюють з розмитістю і можуть бути візуалізовані без проблем, тому використання глибокого композитингу не тільки швидше, а й дає можливість отримати більш високу якість зображення рис. 3.1.

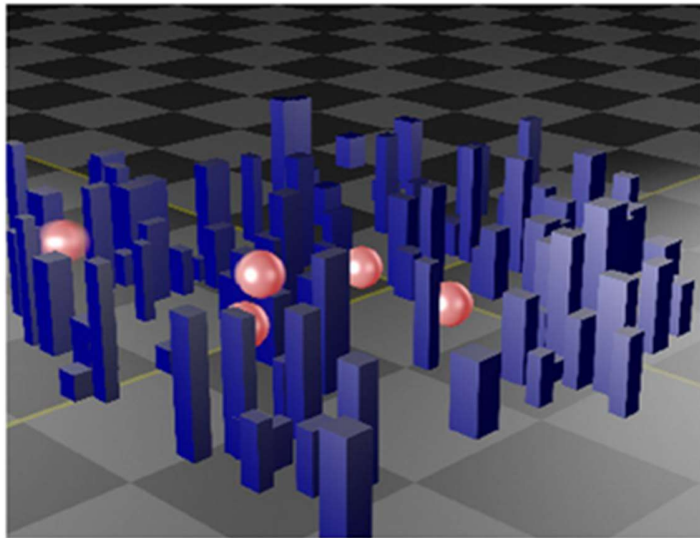


Рисунок 3.1 – Глибокий композит з м'ячем серед об'єктів синіх будівель

Вузли Nuke з розділу глибокого композитингу дозволяють виконати наступне:

- Прочитайте глибоке зображення за допомогою вузла DeepRead.
- Об'єднувати глибокі дані за допомогою DeepMerge
- Створювати маски з пари глибоких зображень за допомогою вузла DeepHoldout.
- Переводити глибокі зображення в звичайні 2D або створювати хмари точок з них.
- Отримувати інформацію по даному пікселю, використовуючи вузол DeepSample.
- Обрізання, переформатування і трансформування глибоких образів таким же чином, як звичайні зображення, використовуючи вузли DeepCrop, DeepReformat і DeepTransform.
- Створення глибоких зображень з використанням DeepFromFrames, DeepFromImage, DeepRecolor, і ScanlineRender вузлів.

3.2 Читання Deep Footage

Читання глибоких зображеннях Nuke відбувається за допомогою вузла DeepRead, який працює як читання в будь-яких інших зображень з вузлом читання. Nuke дозволяє імпортувати глибокі зображення в двох форматах:

- DTEX (генерується з фотореалістичних RenderMan® Pro Server від Pixar).

- Scanline q 2,2 або вище (плиткові файли OpenEXR не підтримуються).

Імпорт файлів DTEX.

Перед імпортом DTEX файлів, необхідно налаштувати RenderMan Pro Server від Pixar 20, на комп'ютері.

Для цього необхідно виконати наступні дії:

Встановити RenderMan Pro Server на комп'ютері і встановити необхідне змінне оточення, яке дозволяють Nuke працювати з ним.

Створити DeepRead натиснувши Deep> DeepRead.

Перейти до .dtx зображення і натисніть кнопку Відкрити.

За замовчуванням, Nuke намагається автоматично визначити тип файлу .dtx, подивившись на ім'я зображення. Якщо ім'я закінчується Deep Shadow або закінчується на .deeropacity, Nuke обробляє файл як глибокий файл непрозорості. Проте, якщо вручну змінити назву зображення при візуалізації файлу, Nuke може не бути в змозі правильно визначити тип файлу. Якщо це так, то необхідно встановити одну з наступних ознак:

- deeropacity - Це змушує Nuke розуміти файл як накопичувач глибоких файлів зображень непрозорості, що відповідає конфігурації RenderMan Display Driver в:

Дисплей "Filename.dtx" "deepshad" "deeropacity"

• альфа - Змушує Nuke розуміти файл в якості нової точки дискретизації альфа каналу або кольору, відповідні конфігурації RenderMan Display Driver: або:

Дисплей «Filename.dtex» «deepshad» «a» або

Дисплей "Filename.dtex" "deepshad" "RGBA"

Імпорт Scanline OpenEXR файлів

Алгорит імпорту Scanline OpenEXR файлів:

Створити DeepRead натиснувши Deep> DeepRead.

.

Перейти до глибоких .exr зображення і натиснути Open.

.

За замовчуванням префікс EXR додаються до метаданих ключів, щоб зробити їх відмінними від інших метаданих в дереві. Якщо не потрібно читати метадані можна не долучати префікс.

3.3 Створення Deep Data

- Перетворення послідовності 2D зображень в глибокі зображення використанням фреймів введення. А саме

- Перетворення 2D-зображення в Deep Image

- Перефарбування глибина даних

- використання ScanlineRender для генерації даних глибинних

Можна створювати глибокі дані в Nuke по:

• вибірці регулярної 2D послідовності зображень на кількох кадрах, щоб створити кілька зразків для кожного пікселя в одному кадрі глибокої композиції.

- перетворенням регулярних 2D-зображень до глибокого зображення за допомогою одного зразка для кожного пікселя на глибині, який визначається каналом depth.Z.
- зразка глибини перефарбування за допомогою звичайного 2D кольорового зображення.
- додавання вузла ScanlineRender до 3D сцени і підключення Deep після нього.

3.3.1 Перетворення послідовності 2D зображення в глибокі зображення використовуючи фрейми введення

Можна використовувати вузол DeepFromFrames для створення зразків глибини від вхідних кадрів для цього необхідно:

Підключити вузол DeepFromFrames до відеоматеріалу. Глибоке зображення створюється шляхом розміщення кожного кадру при збільшенні глибини.

Для коригування результатів, використовують елементи керування на панелі властивостей:

- зразки - число вибірок для створення кожного пікселя в вихідному глибокому зображенні.
- діапазон кадрів - діапазон кадрів, що використовується для одного глибокого зображення. Так, наприклад, зі значенням за замовчуванням зразків (5) і значення діапазону кадру (1-9) DeepFromFrames зразки створяться в момент часу 1, 3, 5, 7 і 9.
- premult - перевірка для попереднього множення зразків.
- альфа-режим розділення - обрати **additive** для виконання прямого розподілу на кілька зразків або мультиплікативний розподіл альфа, так що він може бути повернений в його початкове значення, якщо виконати поєднання

пізніше (за допомогою вузла DeepToImage, наприклад). При виборі **additive**, альфа не може бути повернутий до свого первісного значення.

- Z_{min} - глибина для призначення першого зразка кожного пікселя глибокого зображення, що відповідає першому кадру в діапазоні.
- Z_{max} - глибина для призначення останньої вибірки кожного пікселя глибокого зображення, відповідна останнього кадр в діапазоні.

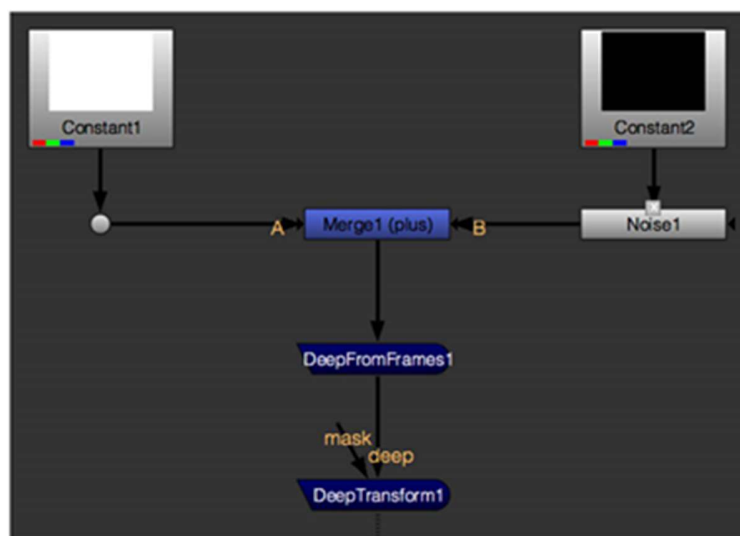


Рисунок 3.2 – Налаштування сцени створення глибокого туману

3.3.2 Перетворення 2D-зображення в Deep Image

Використовуючи вузол DeepFromImage, можна перетворити звичайне 2D зображення до глибокого зображення за допомогою одного зразка для кожного пікселя на глибині, який визначає канал depth.Z.

Підключити до DeepFromImage кадрів які необхідно перетворити в глибокий образ.

Використати поле введення premult в панелі властивостей, щоб обрати, чи вхідні канали повинні бути попередньо помноженим чи ні.

Знімати альфа поле Keep 0, якщо необхідно відмовитися від будь-яких зразків з нульовим значенням альфа каналу з отриманого глибокого зображення. За замовчуванням прапорець встановлено, в результаті глибоке зображення містить нульове значення семплів альфа.

Також можна вказати глибину за допомогою елемента управління в панелі властивостей. В цьому випадку необхідно вказати ZBOX, що буде використовуватися глибина каналу від входу.

3.3.3 Перефарбування глибоких зображень

Використати вузол DeepRecolor для злиття глибоких файлів зображень, які містять непрозорість для кожного зразка за допомогою стандартного 2D кольорового зображення. DeepRecolor змінює колір в кожному пікселі вхідного 2D зображення в усіх зразках відповідного пікселя в глибокому вході.

Підключити глибоке зображення до входу вузла DeepRecolor і 2D-зображення в colorinput. Тоді можна додати вузол unpremultiply між входом кольору і DeepRecolor якщо 2D зображення множитья.

В панелі властивостей, можна обрати, які канали необхідно використовувати для входу кольору зображення.

У наведеному нижче прикладі, DeepRecolor приймає unpremultiplied .exr зображення і використовує його, щоб змінити колір глибокі зразки в .dtxfile рисунку 3.3.

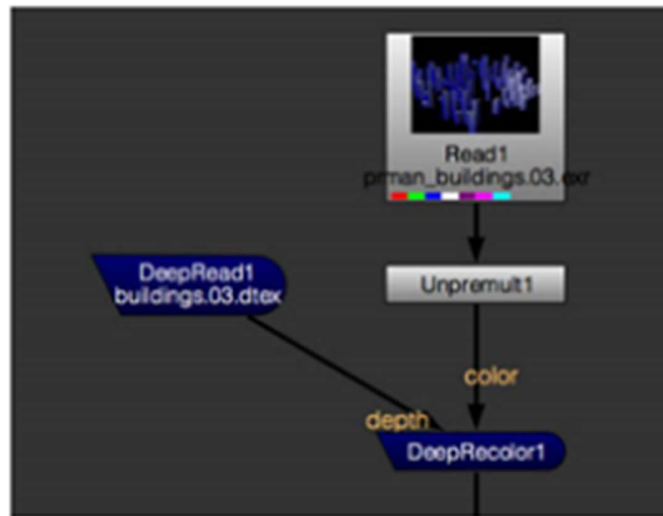


Рисунок 3.3 – Зміна кольору глибокого зображення

Якщо в цей момент альфа в кінцевій високій якості плоскої візуалізації не відповідає альфі представлений в глибоких семплах(наприклад, в результаті стиснення, яке зазвичай відбувається з глибокими файлами на диску або будь-які зміни в шейдері) можна перевірити цільовий вхід альфа. Це означає, що альфа-колірної Вхідний розподіляються серед глибинних проб, так що кінцевий результат отриманий після того, як альфа глибоких даних відповідає альфі коліру вхідних даних.

Якщо залишити цільовий альфа вхід необраним, Nuke розподіляє колір кожного зразка unpremultiplying альфа кольорового зображення, а потім перемножує на альфа кожного зразка. В цьому випадку, альфа від DeepRecolor може не збігатися з альфа введеного кольору.

3.3.4 Використання ScanlineRender для генерації даних глибини

Вузел ScanlineRender може генерувати глибокі дані, якщо існує інший глибокий вузол в дереві нодів.

Створити 3D-сцену і прикріпити вузол ScanlineRender до неї, щоб

. візуалізувати сцену у вигляді 2D зображення.

Deep композитинг підтримує лише режим змішування «над». В результаті, якщо є вузол BlendMat в 3D сцені, результат його роботи завжди встановлений в режим накладання в «над» при перетворенні в Deep.

Додати вузол з Deep меню нижче від ScanlineRender.

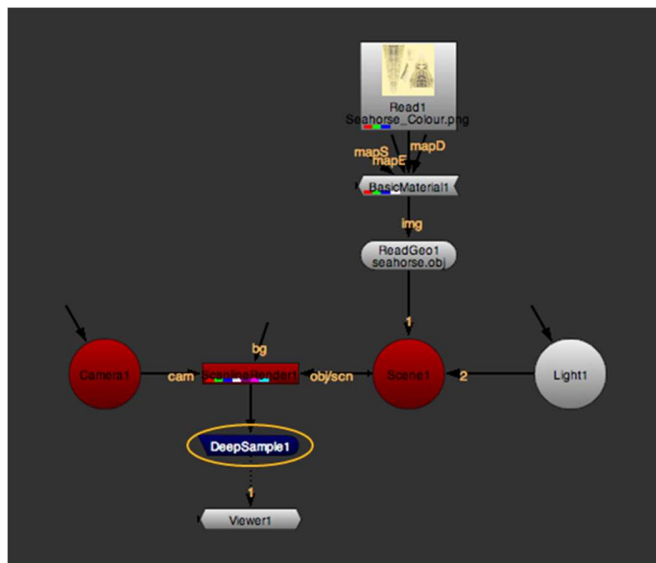


Рисунок 3.4 – Додавання вузлу DeepSamples1

Якщо не потрібно використовувати глибокі зразки з альфа-значенням 0, необхідно відкрити властивості ScanlineRender і переконатися, що ввімкнено значення альфа вище нуля.

Налаштувати інші властивості ScanlineRender як зазвичай. Наприклад:

- Якщо видно будь-які артефакти згладжування в візуалізації, перейти на вкладку MultiSample і збільшити кількість зразків. Це збільшує кількість глибоких семплів на піксель. Крім того, можна встановити згладжування з низьким, середнім або високим значенням на вкладці ScanlineRender.рис3.5



Рисунок 3.5 – Низьке значення і високе значення згладжування

- Якщо необхідно додати розмиття руху до 3D-сцени, потрібно збільшити значення зразків для вибірки зображення в декілька разів протягом часу витримки.

В поле затвора ввести кількість кадрів, затвор залишається відкритим, коли відбувається розмиття руху. Якщо рендеринг стає дуже повільним, можна апроксимувати рендеринг кільку проб і скоротити час рендеринга шляхом збільшення стохастичних зразків.



Рисунок 3.6 – Без розмиття руху та з розмитістю

Можна використовувати вузол DeepToPoints після ScanlineRender для створення 3D хмари точок, яка буде являти собою рух в сцені(рис. 3.7)

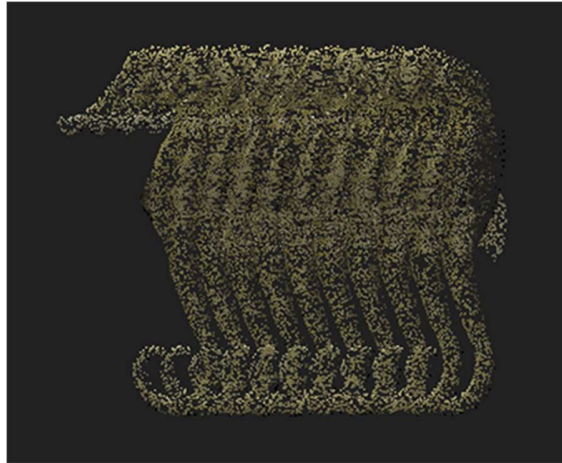


Рисунок 3.7 – DeepToPoints Motion Blur

3.4 Злиття Deep зображень

Для злиття глибоких зображень використовують вузол DeepMerge, щоб об'єднати зразки з кількох глибоких зображень, так що кожен вихідний піксель містить всі зразки з того ж пікселя кожного входу.

Підключити дані, які необхідно об'єднати у пронумеровані входи вузла . DeepMerge.

У властивостях DeepMerge переконатися, що операція встановлюється . для об'єднання.

Перевірити падіння прихованих зразкі в панелі властивостей, щоб не включати в себе зразки, які повністю поглинається зразками альфа-значення яких близьке до одиниці.

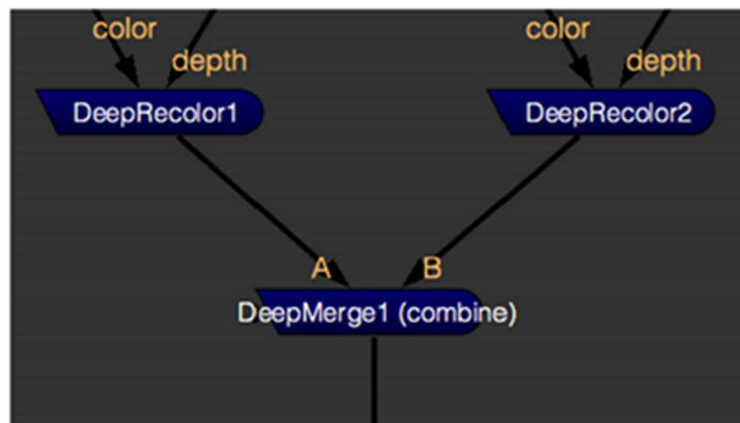


Рисунок 3.8 – Об'єднання двох результатів DeepRecolor.

3.5 Створення масок

Є два способи створення масок, використовуючи або DeepHoldout або вузли DeepMerge. Основна відмінність полягає отриманні вузлу для отримання стиснутого вихідного зображення або глибокого кінцевого зображення після використання маски.

3.5.1 Створення стиснутого зображення за допомогою DeepHoldout Node

Вузол DeepHoldout видаляє або затирає зразки в головному вході, які накладаються на вході маски. Результатом є зведене зображенням головного вхідного зображення, з отворами, де об'єкти у вхідній масці зображення мають непрозорі частини. Для отримання глибокого кінцевого зображення, необхідно використовувати holdoutoperation в вузлі DeepMerge.

Для того, щоб створити маску за допомогою вузла DeepHoldout:

- 1 Підключити глибоке зображення, яке хочете видалити або замаскувати частини від основного входу.

2. Підключити глибоке зображення з яке буде маскою на вхід маски.
3. Тепер можна переглянути результат маскуванню червоного, зеленого, синього і альфа-каналів.

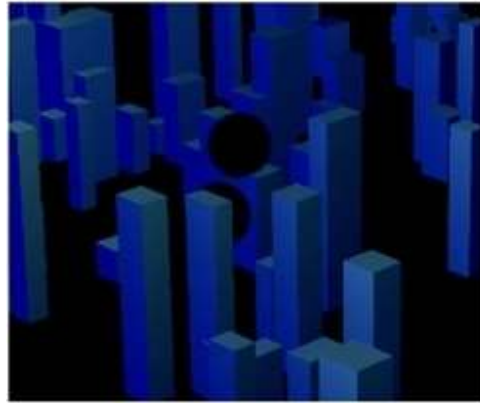


Рисунок 3.9 – Маскування будівель за допомогою кулі

3.5.2 Створення глибокої маски за допомогою DeepMerge Node

Подібно вузлу DeepHoldout, операція маскуванню вузла DeepMerge видаляє або затирає зразки на вході B, які накладаються зразками у вході A. Однак, на відміну від вузла DeepHoldout, DeepMerge НЕ компресує дані, але зберігає глибоке вихідне зображення.

Для того, щоб створити маску за допомогою вузла DeepMerge:

1. Підключити глибоке зображення, яке потрібно видалити до входу B.
2. Підключити глибоке зображення фону до входу A.
3. У властивостях DeepMerge, встановити операцію маскуванню.
4. У зв'язку з плаваючою точкою неточності, деякі зразки з дуже низькими значеннями альфа можуть бути неправильно збережені. Для того, щоб видалити їх з маски, необхідно збільшити падіння нульового порогу. За замовчуванням його встановлено в 0,0000001. Результатом є зображення з маскою на червону, зелену, синьому і

альфа-каналі, а також це зображення є глибоким зі збереженням всієї інформації.

3.6 Створення 2D і 3D елементів з глибоких зображень

Nuke надає можливість створити 2D-зображення або 3D хмару точок з глибокого зображення.

3.6.1 Створення 2D зображення з Deep Image

Для цього необхідно використати вузол DeepToImage для переведення в плоске зображення, інші слова, об'єднати всі зразки в глибокому зображенні в звичайне 2D-зображення.

- 1 Підключити вузол до глибокого зображенню (або DeepMerge з об'єднаними глибокими даними), які потрібно вирівняти.
- 2 В панелі властивостей, прапорець об'ємного композитингу увімкнутий за замовчуванням, але якщо його зняти, Nuke обчислює лише передню глибину кожного зразка і передбачає, що зразки не перекривають один одного. Якщо зняти цей прапорець розрахунок займає менше часу, але якщо існують зразки які перетинаються зображення не може представляти кожен піксель як було очікувано.

3.6.2 Створення хмари точок з глибокого зображення

Можна використовувати вузол DeepToPoints для перетворення глибоких зразки пікселів в точки у 3D-просторі, яке як можна побачити в 3D вигляді Nuke, вони відображаються як хмара точок. Цей вузол є корисним

лише для зрозуміння положення.

- 1 Підключити глибокий вхід цього вузла DeepToPoints до глибокого зображенню, яке потрібно переглянути в 3D. Якщо в сцена і є камера через яку необхідно спростерігати хмару точок то її необхідно також підключити до вузла.
2. Щоб ознайомитись з результатами зміни в 3D вигляді натиснути Tab.
- 3 В панелі властивостей, ви можна використовувати повзунок що регулює кількість точок, щоб відрегулювати щільність хмари.

Рис 3.10

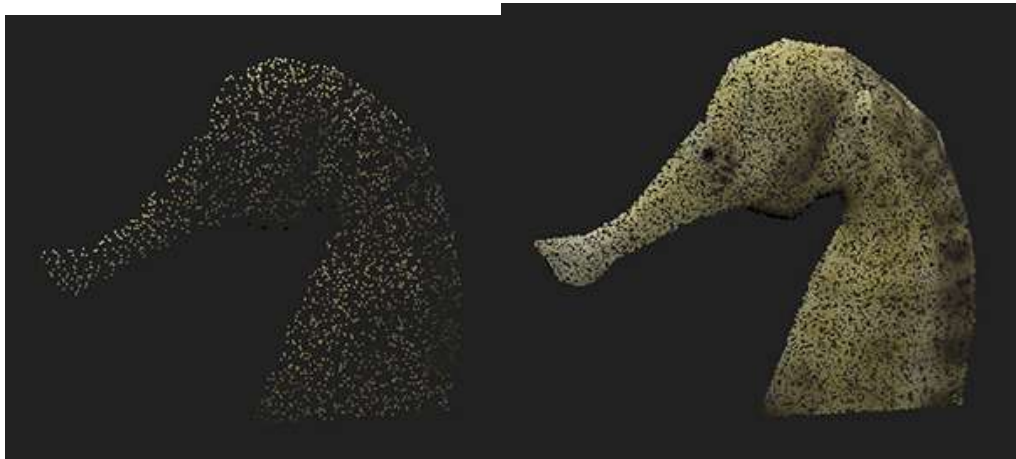


Рисунок 3.10 – Деталізація точок 0,005. детальність точок 0,05

Налаштування розміру точки для зміни розміру хмари. Також можна використовувати вибір 3D об'єктів на хмарі точок, і наприклад, клацнути на DeepToPoints для результату рис. 3.11



Рисунок 3.11 – розмір точки 2 і розмір точки 6

3.7 Зміна Deep Data

3.7.1 Зміна кольору глибоких зображень

Nuke дозволяє виконувати корекцію глибоких зображень, а також змінювати їх за допомогою виразів.

Вузол `DeepColorCorrect` застосовує корекцію кольору для кожного зразка у кожного пікселя. Є набори керування для регулювання тіні, півтонів і відблисків, а також майстер-набір для налаштування всього цього відразу. Можна використовувати криві керування на вкладці `Ranges` щоб контролювати діапазон зображення, який знаходиться в тіні, півтони і відблиски.

Зробити маску можна встановивши значення посилення для альфа-каналу на 0 і встановивши значення зміщення для альфа-каналу 1 в діапазоні де має знаходитися маска.

3.7.2 Налаштування ефекту Deep Color Correction

На вкладці маскування можна встановити точку серед глибинних семплів, де ефект колірної корекції починається і закінчується.

1. Встановити прапорець `limit_z`, щоб активувати інструмент `zmap`.
2. Налаштувати трапецію, так що `A` відзначає глибину, де потрібно почати корекцію кольору, `B` і `C` відзначають довжину повного ефекту і обмежувачем `D` вказує на те, де ефект припиняється. вісь у `Zmap` в інструменті, вказує на кількість ефекту, а вісь `x` діапазон зразків глибини.
3. За допомогою регулятора суміші, необхідно регулювати загальне змішування між кольором виправленим результатом і вихідним зображенням. Нульове значення це вихідне зображення, а значення 1 є повним результатом корекції кольору.

3.7.3 Зміна глибоких зображень за допомогою виразів.

Можна використовувати вузол `DeepExpression` для запуску `Nuke` виразів до глибоких даних. За допомогою елементів управління в панелі властивостей:

1. Є чотири поля для тимчасових виразів, так само, як в звичайному вузлі `Expression`. Це може бути корисно, якщо потрібно використовувати довгий вираз в кількох областях і необхідно привласнити цей вираз тимчасовій змінній. Ввести ім'я змінної зліва від знака рівності (`=`), а вираз справа. Потім можна використати ім'я змінної, щоб представити весь вираз у виразі поля для каналів.

- 2 У chans0 - chans3 випадючих меню можна вказати, які канали будуть використовувати вирази. Це додає або видаляє вираз поля що знаходиться нижче.
- 3 Після цього можна ввести вирази для різних каналів в області визів каналу.

3.8 Зміна розміру, переформатування і перетворення Deep Images

Існує можливість обрізати, переформатувати і трансформувати глибокі образи багато в чому таким же чином, як із звичайним зображенням, використовуючи відповідні глибокі вузли.

Оскільки зразки в кожному пікселі можуть бути розташовані на довільних глибинах, зміна положення семплів в ході трансформації може призвести до непередбачуваних результатів, так як там не може бути зразків при тій же глибині в сусідніх пікселях.

3.8.1 Зміна розміру Deep Images

Можна використовувати вузол DeepCrop підрізати глибоке зображення, так само, як нормальний вузол Crop:

- 1 Підключити вузол DeepCrop до глибокого зображення, яке змінити.
- 2 Налаштувати рамку зміни зображення у вікні перегляду X і Y напрямках, щоб визначити площу вирізу. З іншого боку, визначити область кадрування за допомогою поля BBox в панелі властивостей. Якщо потрібно зберегти зразки глибини за межами кадру, потрібно поставити помітку напроти keep outside bbox.

3 Використати `znear` і `zfar` управління в панелі властивостей, щоб обрізати зразки в глибину. Якщо не потрібно використовувати будь-якого з цих елементів управління, можна їх відключити, прибравши прапорець «використовувати» поруч з ними. Якщо потрібно зберегти зразки глибини за межами діапазону `z`, який визначається цими елементами управління, потрібно позначити поле тримати поза `zrange` вікна.

3.8.2 Переформатування Deep Images

DeepReformat є вузлом переформатування для глибоких даних. Можна використовувати його для установки глибокого зображення в потрібних розмірах, масштабах, і так далі. Для того, щоб переформатувати глибоке зображення:

1. Підключити вузол DeepReformat до глибокого зображення як змінити.
2. У списку тип обрати:
 - формувати - задає ширину і висоту вихідного зображення в обраному форматі. Обрати формат в списку OUTPUTFORMAT. Якщо формат ще не існує, ви можете вибрати новий, щоб створити новий формат з нуля. Значення за замовчуванням, `root.format`, змінює розмір зображення у форматі, вказаному в діалоговому вікні «Параметри проекту».
 - поле - задає ширину виводу і висоту до розмірів, які визначають в пікселях. Ввести значення ширини, висоти і піксельний аспект полів для задання розмірів.
 - шкала - задає ширину і висоту вихідного до кратного розміру вхідного зображення. За допомогою шкали повзунка визначається коефіцієнт. Масштабний коефіцієнт злегка заокруглений, тому що вихідне

зображення являє собою ціле число пікселів в напрямку обраному відповідно до типу зміни розміру.

3 Можна вказати, які зміни розміру потрібно задати в типі зміни розміру списку. Обрати:

- пусто - не змінювати розмір оригіналу.
- ширина - масштаб оригіналу, поки його ширина не відповідає ширині вихідного розміру. Висота потім масштабується таким чином, щоб зберегти початкове співвідношення сторін.

- висота - масштабувати оригінал так, щоб він заповнив висоту виведення. Ширина потім масштабується таким чином, щоб зберегти початкове співвідношення сторін.

- підібрати - масштабувати оригінал так, щоб його маленька сторона заповнює ширину вихідний зображення або висоту. Довга сторона потім масштабується таким чином, щоб зберегти початкове співвідношення сторін.

- заповнити - масштабувати оригінал так, щоб його довга сторона заповнила ширину вихідного зображення або висоту. Найменша сторона потім масштабується таким чином, щоб зберегти початкове співвідношення сторін.

- спотворювати - масштабувати оригінал так, щоб обидві сторони заповнити розміри вихідного зображення. Цей параметр не зберігає вихідне співвідношення сторін, тому можуть виникнути спотворення.

4 Перевірити центральне поле, щоб визначити, чи вхідні пікселі повинні бути перетворені до нового розміру або бути відцентровані на виході. Якщо не відцентрувати, нижній лівий кути входу і виходу збігаються.

5 Для подальшого налаштування макету зображення можна

- . відповідні прапорці:
 - flip - поміняти місцями верх і низ зображення.
 - flip - поміняти місцями ліві і праві зображення.
 - turn - повернути зображення на 90 градусів.
 - black outside - встановити пікселі за межами формату чорного кольору.
 - зберегти обмежувальну рамку - зберегти пікселі за межами вихідного формату, а не відсікати їх.

3.8.3 Перетворення глибоких семплів

Можна використовувати вузол DeepTransform, щоб змінити глибокі семпли.

- 1 Підключити вузол до глибокого відеоматеріалу який перетворити.
- 2 Використати перетворення X, Y, і Z елементи управління, щоб перевести зразки.
- 3 Шкала глибини z зразків може бути збільшеною за допомогою zscale елементу управління. Значення вище 1 збільшить глибину, в той час як значення нижче 1 зменшить його.
- 4 Якщо підключити маску до входу вузла маски, можна використати його, щоб регулювати, кількість ефекту перетворення глибини існує в різних частинах кадру.

3.9 Відбір семплів глибокого зображення

Можна використовувати вузол DeepSample до зразків будь-якого заданого пікселя в глибокому зображенні. Вузол DeepSample переводить

дані про глибину в якості значень.

1. Підключити вузол DeepSample до іншого вузла глибокого зображення.
2. Помістити індикатор позиції на піксель, який необхідно побачити в засобі перегляду.
3. Перегляд інформації глибокого семплу відбувається в прикладі таблиці на панелі властивостей DeepSample.
4. Також можна переключити вікно вибору джерела зображення щоб побачити окремі значення семплів з вибірки пікселя (не переключений), або остаточне значення скомпоновані (переключений).

3.10 Запис Deep Data

Глибокі образи можна записувати в формат файлів OpenEXR 2,2 або вище, використовуючи вузол DeepWrite, який має багато подібного до стандартного Write вузла. Виконати наступні дії:

1. Обрати Deep> DeepWrite вставити DeepWrite вузол в сценарій.
2. В панелі властивостей обрати файл або папку в меню вибору і перейти до каталогу, в якому потрібно зберегти глибоке зображення.
3. Після введення шляху, ввести ім'я для глибокого зображення, включаючи розширення .exr, а потім натисніть кнопку ОК. Якщо виконується рендеринг послідовності зображень, включити в назву змінний номер кадру (наприклад, #####).
4. Використовуючи меню типу даних, що випадає, обрати бітову глибину для наданого файлу: 16 бітове зображення або 32 біт з плаваючою точкою.

5. Встановити стиснення типу стиснення, щоб застосувати до наданого файлу.

6. З меню метаданих, обрати метадані включені наданим файлом:

- немає метаданих – Ніякі дані користувача не створюються, тільки метадані, які заповнюють необхідні поля заголовка записуються.

- метадані за замовчуванням - Опціональний тимчасовий код, край коду, частота кадрів і заголовки заповнені з використанням значень метаданих.

- метадані за замовчуванням і EXR / *

- всі метадані, окрім даних введення / *

- всі метадані

7. За замовчуванням невідомі ключі метаданих мають префікс нюк який додається до них, коли вони записуються в файл. Якщо потрібно записати файл «як є», без префікса, обрати пункт «не приєднувати префікс».

8. Налаштувати інші елементи управління в міру необхідності.

9. Натиснути кнопку Render. Nuke запитає діапазону кадрів, за замовчуванням діапазон береться з поля діапазону кадрів.

10. Змінити початкові і кінцеві кадри, а потім натиснути кнопку ОК.

Nuke пише глибокі дані в рядки розгортки OpenEXR 2,2 (плиткові файли OpenEXR не підтримуються).

Висновки

Проаналізовано основні функції та можливості нодів, які складають групу глибинного композитингу. Зокрема, визначено такі характерні особливості ноїв, які визначають створення ефекту глибокого туману. Окремо описано основні операції які можна провести з глибинними зображеннями в середовищі Nuke.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту 3D metro

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
3D Metro – це простий і зрозумілий сервіс для пошуку і стеження за рухом метрополітену. Сервіс дозволяє прокладати найбільш зручний та зрозумілий маршрут руху.	1. Онлайн пошук і стеження за вагонами метро на різних станціях.	Зручність та економія часу.
	2. Онлайн новини та довідкова інформація.	Проінформованість.
	3. Візуалізація географічних об'єктів.	Вивчення карти міста.

Опис до таблиці 4.2:

W – слабка сторона;

N – нейтральна сторона;

S – сильна сторона.

Таблиця 4.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економічні характе- ристики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		3D Metro	Київ міськ довідка	Телефонн а довідка	Мобільні послуги			
1	Програмне забезпечен ня	Nuke	Текстові данні	Автовідп овідач\ Оператор	Додаток androind			+
2	Автоном- ний час роботи	24 години	24 години	12 годин	4 години		+	
3	Підтримка візуалізації 3D	+	-	-	+			+
4	Об'єкти застосува- ння	Метро, місто	Місто	Місто та передміст о	Скрізь		+	
5	Контакт- центр	-	+	+	-	+		
6	Вартість послуги	5 \$	0 \$	0 \$	10 \$		+	

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Модернізація станцій метро	Nuke	наявна	доступна
2	Встановлення руху переміщення людини через об'єкт розумного міста	GSM модуль, SIM карта	наявна	доступна
3	Використання площі для ретрансляції корисних порад	Проектори та LED панелі	наявна	доступна
4	Персональний онлайн сервіс	Програмне забезпечення для ОС: Windows, Android, Mac	необхідно розробити	доступна

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.4 - Попередня характеристика потенційного ринку

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5
2	Динаміка ринку (якісна оцінка)	Зростає
3	Наявність обмежень для входу (вказати характер обмежень)	відсутні
4	Середня норма рентабельності в галузі (або по ринку), %	90%

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Орієнтування в місті, телефони спеціальних служб, зручність переміщення в метро та місті.	Пішоходи міста	Немає	<ul style="list-style-type: none"> - гнучкість - компактність - доступність - простота - зручність - швидкість

Таблиця 4.6 - Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Незацікавленість клієнтів	Внаслідок невдалого маркетингу клієнт може не зацікавитись послугами	Внесення додаткових сервісних послуг та зниження цін
2	Втрата монополії	Втрата рангу єдиного гаранту якості технології	Якісне та кількісне нарощування інтенсивності

Таблиця 4.7 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Монополія	Інноваційний тип послуг	Стандартизація на високому рівні
2. Локальний	Відсутність єдиного національного постачальника послуг	Окремий підхід до кожної локальної ділянки
3. Міжгалузєва	Конкуренція з іншими галузями (постачальниками апаратної частини)	Необхідність співробітництва в окремих сегментах
4. Товарно-видова	Подолання розсинхронізації відбувається за схожими технологіями, що реалізовані апаратно	За необхідності, використання приладів схожого типу

Продовження таблиці 4.7 - Ступеневий аналіз конкуренції на ринку

5.Цінова	Можливість заощадити за допомогою діагностики	Гнучка політика цін
6.Марочна	Кожна діагностика має бути стандартизованою	Отримання монополії над стандартом синхронізації

Таблиця 4.8 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товаризамінники
	Технологічні постачальники	Необхідність пошуку постачальників	Залучення малопопулярних постачальників	Незалежність у прийнятті клієнтських рішень	Надання переваги більш авторитетним технологічним рішенням
Висновки:	Незначна	Можливість виходу на ринок є	Постачальники диктують цінову політику на обладнання	Клієнти диктують вимоги до якості	Обмеження існують лише у разі відмови від діагностики

Таблиця 4.9 - Обґрунтування факторів конкурентноспроможності

№	Фактор конкурентноспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Раціональніший ціновий показник	Можливість більш раціонально використати ресурсів
2	Надання персональних сервісних послуг 24/7	Сервісна підтримка апаратної та програмної частини
3	Синхронізованість	Синхронізація з усіма ОС.
4	Спектр застосувань	Використання для ряду потреб користувачів.

Таблиця 4.10 - Порівняльний аналіз сильних та слабких сторін 3D Metro

№	Фактор конкурентноспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні						
			-3	-2	-1	0	1	2	3
1	Раціональніший ціновий показник	13			+				
2	Надання персональних сервісних послуг 24/7	15			+				
3	Синхронізованість	20	+						
4	Спектр застосувань	17		+					

Таблиця 4.11 - SWOT- аналіз стартап-проекту

Сильні сторони:, надання персональних сервісних послуг 24/7, синхронізованість	Слабкі сторони: раціональніший ціновий показник
Можливості: використання для ряду потреб користувачів	Загрози: незацікавленість клієнтів, втрата монополії

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.12 - Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Пішоходи та туристи	Середня	Високий	Високий	Середня
2	СМІ	Середня	Високий	Середній	Середня
3	Сім'ї	Середня	Висока	Низький	Середня
4	Web ресурси (сайти, клуби, соц.мережі)	Висока	Висока	Середній	Середня
Які цільові групи обрано: пішоходи та туристи					

Таблиця 4.13 - Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Створення гаранту якості державного рівня	Встановлення єдиного універсального стандарту	Розробка і випуск власних пристроїв	Стратегія диференціації
2	Дешевизна проекту	Раціональніші витрати на обладнання, та послуги	Маловідомі партнери з постачання обладнання	Стратегія лідерства по витратах

Таблиця 4.14 - Визначення базової стратегії конкурентної поведінки

№	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	ні	Забирати існуючих та шукати нових	Характеристики апаратної частини	Стратегія виклику лідера

Таблиця 4.15 - Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Висока якість послуг	Стратегія диференціації	Синхронізованість	Якість, надійність, сервісність
2	Мінімальні витрати	Стратегія лідерства по витратах	Широкий спектр застосування	Дешевизна, раціональність, тех. підтримка

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.16 - Визначення ключових переваг концепції

потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Якість	Висока якість, сервісність	сервісність
2	Дешевизна	Раціональне використання коштів	дешевизна

Таблиця 4.17 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Дешевий якісний товар та послуги, стандартизована якість послуг та обладнання		
II. Товар у реальному виконанні	Властивості/характеристики:	М/Нм	Вр/Тх /Тл/Е/Ор
	1) Варстість обслуговування, 2) Кількість елементів	1) М 2) М	1)Е 2) Пр
	3) Строк безвідмовної праці 4) Технологічна собівартість товару	3) М 4) М	3)Нд 4)Тх
	Якість: дерстандарт якості, високоякісні технології		
III. Товар із підкріпленням	До продажу – діагностика, обладнання, кріплення, дод.елементи живлення Після продажу – персональний онлайн сервіс		

Таблиця 4.18 - Визначення меж встановлення ціни

№	Рівень цін на товари замітники	Рівень цін на товарианалоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на послугу
1	1000 у.о./од	250 у.о./од	Різний	Н.5 у.о. – В.10 у.о.

Таблиця 4.19 - Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рек-ламного пові-домлення	Концепція рекламного звернення
1	Зацікавленість в якісному продукті з раціональним використанням ресурсів	Мережні ресурси	Синхронізованість з будь-якими ОС	Зацікавити у покращеннях пов'язаних із зростаючою популярністю товару та послуг	Представлення продукції відправною точкою на шляху до безпеки
2	Зацікавленість у великій кількості продукту із дотриманням умов якості	Мережні ресурси	Широкий спектр застосування	Зацікавити у позитивних сторонах	Представлення якісної роботи з клієнтами

ВИСНОВКИ

В рамках магістерської дисертації було проведено дослідження програмних особливостей різних режимів налаштування цифрового композитингу в середовищі Nuke для сучасного кіно. Зокрема, в рамках дослідження:

1. Визначено основні технічні труднощі, які можуть з'являтися, або які можуть виникнути при створенні стерео кінофільмів. Зокрема, технологічний процес Nuke для роботи в режимі стерео може позбавити від зайвої роботи, пов'язаної з окремою обробкою двох потоків зображення, а не одного, а інструменти Osula, такі як VerticalAligner і ColourMatcher здатні автоматизувати процедури, що раніше виконувалися вручну і забирали багато часу і сил. Крім того, технологія генерації карт невідповідності Osula допомагає зберегти точну кореляцію між лівим і правим ракурсами, завдяки чому аудиторія зможе насолоджуватися 3D-зображенням, а не страждати від нього.

2. У другій частині дослідження розглядалися основні елементи які слід враховувати при створенні 3D матеріалу. Серед них особливу важливість відіграє налаштування віртуальної камери та ноди яка відповідає за рендеринг 3D сцени.

Проведено створення моделі 3D сцени та проаналізовані різні інструменти щодо її управління. В окремій частині дослідження показано особливості використання «пасу» світового положення для створення маски 3D об'єкта.

3. В третьому розділі магістерського дослідження проаналізовано технічні особливості роботи з глибоким зображенням. Показана модель глибокого туману та досліджено можливості роботи з альфа каналом. Окремо показано особливості роботи з елементами рендерингу для обраної об'ємної моделі.

Створено 2D і 3D елементи з вихідного зображення на основі глибокого композитингу.

4. Запропонована ідея стартап проекту яка дозволяє використовувати отримані результати досліджень в рамках концепції розумного міста, надані переваги щодо реалізації ідей стартап проекту зокрема на станція метро.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Goh S. Digital Compositing with Nuke 101. Hong Kong, 2010. 160 p., ISBN: 978-988-18465-1-8.
2. Ganbar R. Nuke 101. Professional Compositing and Visual Effects. USA, 2011. 404 p., ISBN: 978-0-321-73347-4
3. Трудности композитинга при работе в 3D-стерео. URL: <http://www.mediavision-mag.ru/> (дата звернення: 26.10.2018).
4. Okun A., Susan Z. Visual Effects Society. China, 2010. 923 p., ISBN 978-0-240-81242-7
5. John G., [digital] VISUAL EFFECTS & COMPOSITING. USA, 2015. 423 p., ISBN 13: 978-0-321-98438-8.
6. Орлов Б. В. Дослідження програмних особливостей створення 3d-елементів для сучасного цифрового кіно *Elconf-2018*. – 2018.- Секція №7. – С 359 – 362.
7. Орлов Б. В. Особливості використання багатокomпонентного композитингу *Сучасні проблеми застосування електронних та інформаційних технологій в телекомунікаціях, телебаченні та цифровому кінематографі: тези доповідей*. – 2018 – Секція А. – С 30 - 31.

ДОДАТОК А
ABSTRACT

The purpose of this work is to explore a specific area of visual effects production - Digital Compositing. The work explains the origin of this craft and clarifies reasons for its demand in the field.

The theoretical section explains the role of Digital Compositing in the VFX pipeline, as well as introduces a reader to the major compositing operations, which digital artists in the field have to be able to perform. Among those are compositing CGI materials into live-action plates, rotoscoping, keying, clean up, tracking and recreation of realistic lens effects.

The practical part demonstrates how those Digital Compositing operations are combined on practice to achieve a desirable illusion in a shot.

Based on the ability to produce realistic illusions and implement complex visual ideas, the work concludes, that Digital Compositing is an important and inseparable part of the VFX pipeline, and a demanded craft in the industry.

Digital Compositing has come to the industry of VFX production to stay. Not only does it save computer powers, time and money to companies and freelance artists, it raises the limits of what ideas are possible to implement in a tight schedule. Being a part of the pipeline, Digital Compositing supplements 3D applications, as well as offers compelling solutions for 2D imagery challenges.

From a point of view of an artist, Digital compositing is a fascinating form of art. It is an astonishing fusion of technical and artistic skills. Replacing backgrounds, eliminating objects from a shot, integrating fantastic CG characters into realistic worlds, creating fantasy environments and whole new worlds for a viewer's entertaining – those are the challenges that a digital compositor deals with.

Digital Compositing is a powerful and highly demanded craft. The need to create a believable illusion on the screen is an integral part of modern imagery production. Constantly evolving specialized software provides more and more tools and possibilities for digital artists to create those illusions, making the most unexpected ideas come true, at least on the screen. However, all the possibilities of

this digital art can't be brought to a conclusion. As long as the expectations of the audience grow, as far as its fantasies go, Digital Compositing will follow. Evolving, surprising and inspiring the audience.

With the increasing use of visual effects in feature films, TV series and commercials, flexibility becomes essential to create astonishing pictures while meeting tight production schedules.

Deep image compositing introduces new possibilities that increase flexibility and solve old problems of depth based compositing. The following thesis gives an introduction to deep image compositing, illustrating its power and analyzing its use in a modern visual effects pipeline.

In computer graphics when rendering images from 3D software, one ends up with a temporal flow of two-dimensional images. These are usually further treated in compositing to achieve the desired final look. In compositing it can, however, be useful to have more information than the afore mentioned information in x and y. More precisely, the needed information to composit or alter elements in the right way or to achieve certain effects is information on the third dimension, the depth, also referred to as z.

This information is already used in the 3D application for the rendering of the picture and should therefore be fairly easily retrieved and handed over to compositing. Existing concepts and methods already use this so called "zBuffer" and render it into a separate channel or image mostly referred to as zDepth.

Due to the conceptual nature of an image as a two-dimensional array of (color) values, the depth information stored is not accurate enough and often leads to problematic edge artefacts. As for volumetric imagery, depth throughout the volume can not be represented by traditional methods, and thus does not allow for further depth based operations on it.

The concept of deep compositing has two primary ways to face the afore mentioned problems:

For volumetrics, occluding objects placed inside the volume usually needed to be rendered with holdout mattes¹. An animation change meant the need to also re-render the volumetric and therefore led to higher render times and thus a longer turn around time. Deep compositing is facing this problem with a more flexible solution.

In addition, every z based compositing operation which relies on the traditional zDepth channel faces some fairly big trade-offs and usually results in edge artefacts. Due to the inability to accurately store differentiated depth information in overlapping semitransparent areas, good results were sometimes impossible to achieve, depending on the exact case. Motion-blurred renderings, where a fairly big amount of transparent edges occur, were nearly impossible to use with traditional z based compositing operations. Tackling this issue is one of the most fundamental aims of deep compositing.

The following thesis is going to concentrate on the use of deep compositing in the visual effects industry. Similar concepts might also be used in medical technology or for industrial assembly purposes. Due to the different needs I'm going to concentrate on high end visual effects work. With growing use of visual effects in feature films, commercials and other media formats, quick turnarounds and tight production schedules go hand in hand with last minute change orders and tight budgets. For a visual effects company, it is vital to have flexible structures and a pipeline supporting quick changes and adaptations. As rendering is one of the main time critical factors, re-rendering should be avoided wherever possible. Rendering times are much higher for 3D renderings². Thus, giving compositing the power to change aspects of the rendering afterwards helps to quickly adjust for changes. This concept is already largely adopted³. With multi pass renderings and additional utility passes, compositing artists can quickly alter shading and lighting and achieve computationally expensive operations such as motion blur or depth of field after the rendering. Deep compositing embraces this concept of providing additional information to the compositing department and pushes it even further.

Delving deeper into deep compositing reveals even more possibilities. The logical conclusion of which is a combination of deep compositing with other already existing techniques in the visual effects industry.

In the following document I am going to investigate the use of deep compositing in a modern visual effects pipeline. To start off, I will give an insight into the basic concept behind deep compositing. I will then compare it to traditional concepts like the zDepth pass and holdout mattes. I will illustrate these with some examples, showing the advantages of deep compositing over existing methods. Further, the implementation in Nuke and V-Ray/3ds Max will be explained. This will lead us to another important point for the real life use of deep images, their performance in a production infrastructure. This will lead to a look at real production necessities in terms of tools and workflow. To finish off, I will give an outlook on what else deep compositing could theoretically allow for and what I think would be the logical technical conclusion. To conclude, I will summarize the important points to keep in mind when working with deep images and give a personal recommendation.

Deep compositing brings many benefits, two of them particularly worth mentioning. The first is increased flexibility which is demonstrated by its capability to render elements independently and combine them in compositing. The second is the unbiased depth and color information that deep images present. With this, artefacts caused by inaccurate depth information are eliminated. As with every good tool, deep compositing allows the artist to become even more creative, removing technical headaches posed by structural restrictions. However, these advantages come at a price – decreased performance. Especially for volumetric renderings, file sizes can explode and present a big processing challenge. Therefore, deep images should not be used blindly. Not every situation needs the additional depth information. The relation of file size to information is linear, the larger the files are, the more information they contain.

However, more information does not always mean a better result. Some situations do not require as much depth resolution, so it is better to keep the file sizes reasonable. Additionally, handling of large deep images can be optimized by using concepts such as proxies and region of interest. Volumetric renderings may result in the largest files but they also bring a huge advantage. And although non-volumetric renderings don't have as big an impact on performance, they still offer reasonable benefits over traditional images in semi-transparent areas.

Before deep compositing is used in production, the infrastructure must be tested and, if necessary, adapted to facilitate a flawless adoption of deep compositing. Otherwise, the use of deep images can cause serious decline in performance and a potential breakdown of the network infrastructure. Still, this is only the beginning. There is great potential in deep images but the tools to handle them are sparse. This will hopefully change with the official release of a standard format, namely Open EXR 2.0. This essential information is needed by many tools, and should be clearly and distinctly embedded into the image itself. The possibilities offered by deep images will increase greatly once the tools evolve. The lines between 3D and compositing will get ever more blurred, introducing more and more possibilities and flexibility. This will potentially lead to a future of voxel based compositing, where the image is not the intermediate, but only the final product.