

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

**Факультет електроніки**

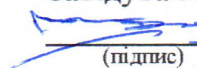
(повна назва інституту/факультету)

**Кафедра акустичних та мультимедійних електронних систем**

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

 С.А. Найда  
(підпис) (ініціали, прізвище)

“01” червня 2020 р.

**Дипломна робота**

на здобуття ступеня бакалавра

зі спеціальності 171 Електроніка  
(код і назва)

на тему: «Особливості використання технології “deep compose” в  
цифровому кіновиробництві»

Виконав: студент IV курсу, групи ДВ-61  
(шифр групи)

Ільченко Ілля Богданович

(прізвище, ім'я, по батькові)

  
(підпис)

Керівник

доц. Трапезон К.О.

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

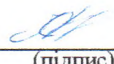
  
(підпис)

Консультант

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент доцент кафедри ЕПС, к.т.н., доцент Михайлов С.Р.  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ініціали)

  
(підпис)

Засвідчую, що у цьому дипломному  
проекті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент   
(підпис)

Київ – 2020 року

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

Факультет Електроніки

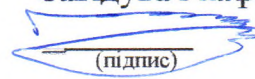
Кафедра акустичних та мультимедійних електронних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 171 Електроніка

ЗАТВЕРДЖУЮ

Завідувач кафедри

  
(підпис)

С.А. Найда  
(ініціали, прізвище)

« 25 » травня 2020 р.

**ЗАВДАННЯ  
на дипломну роботу студенту**

Ільченку Іллі Богдановичу  
(прізвище, ім'я, по батькові)

1 Тема роботи: «Особливості використання технології “deep compose”  
в цифровому кіновиробництві»

керівник роботи Трапезон Кирило Олександрович, к.т.н., доц.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «25»травня 2020 р. №1196-с

2 Термін подання студентом роботи 04 червня 2020 р.

3 Вихідні дані до роботи: 1) основний програмний продукт для аналізу – The Foundry NukeX non-commercial; 2) тип файлів для аналізу – open EXR; 3) формат зображення – HD 1080 (1920×1080 pix); 4) кількість об'єктів для роботи з deep-data – 2; 5) кількість зображень для аналізу – 2; 6) можливість роботи в режимі 3D – доступно.

4 Зміст роботи: 1) Розглянуто основні принципи композитингу з точки зору необхідності використання додаткової інформації про так звану “глибину” пікселів; 2) Проаналізовано базові операції цифрового композитингу на прикладі двох тестових зображень; 3) Визначено основні підходи до реалізації deer-композитингу при роботі з зображеннями; 4) Проведено моделювання схеми з використанням основних мод з групи DEEP програмного пакету NUKEX.

5 Перелік ілюстративного матеріалу: комплект презентації за матеріалами проведеного дослідження.

#### 6. Консультанти розділів роботи

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

7. Дата видачі завдання 11 березня 2020 р.

#### Календарний план

| № з/п | Назва етапів виконання дипломної роботи                            | Термін виконання етапів роботи | Примітка |
|-------|--------------------------------------------------------------------|--------------------------------|----------|
| 1     | Написання першого розділу                                          | 24.03.2020                     | виконано |
| 2     | Написання другого розділу                                          | 20.04.2020                     | виконано |
| 3     | Написання третього розділу                                         | 10.05.2020                     | виконано |
| 4     | Підготовка матеріалів до друку та оформлення пояснювальної записки | 30.05.2020                     | виконано |
| 5     | Підготовка та оформлення презентації для доповіді                  | 1.06.2020                      | виконано |

Студент

  
(підпис)

І. Б. Ільченко

(ініціали, прізвище)

Керівник роботи

  
(підпис)

К.О. Трапезон

(ініціали, прізвище)

## РЕФЕРАТ

Дипломна робота: 50 с., 31 рис., 1 дод., 8 джерел.

ЗОБРАЖЕННЯ, ФОРМАТ, ТРАНСФОРМАЦІЯ, ГЛИБИНА, КОМПОЗИТИНГ, НОДА, СХЕМА, СЕМПЛ, КАНАЛ, КАМЕРА, ОБ'ЄКТ, СЦЕНА.

Об'єкт дослідження – набір зображень формату HD\_1080 з розширенням open EXR з елементами графіки та наявні інструменти спеціалізованого програмного продукту з кіновиробництва.

Метою роботи є аналіз програмних інструментів програми з цифрового композитингу для роботи з деер-інформацією. Крім цього, необхідно визначити основні принципи та підходи, які дозволяють проводити усі основні операції з елементами зображення, яке містить додатковий канал глибини.

У результаті виконання дипломної роботи була створена нодова схема, яка на основі підключення двох зображень з елементами комп'ютерної графіки дозволяє визначити усі технічні нюанси роботи з деер інформацією. Крім цього, наведено алгоритм, за яким можна провести подвійне конвертування 2D-зображення та реалізувати базові операції цифрового композитингу.

Галузь застосування: кіно виробництво, цифровий композитинг, тобто створені схеми моделювання дозволять у майбутньому розробити ефективні методи з покращення зображення на етапі поствиробництва кінофільмів.

## **THE ABSTRACT**

Thesis of bachelor's: 50 p., 31 fig., 1 appendix, 8 refer.

IMAGE, FORMAT, TRANSFORMATION, DEPTH, COMPOSITING, NODE, SCHEME, SAMPLE, CHANNEL, CAMERA, OBJECT, SCENE.

The object of research is a set of HD\_1080 images with open EXR extension with graphic elements and available tools of a specialized software product for film production.

The aim of the work is to analyze the software tools of the digital compositing program for working with deep information. In addition, it is necessary to define the basic principles and approaches that allow you to perform all the basic operations with the elements of the image, which contains an additional depth channel.

As a result of the thesis, a node scheme was created, which, based on the connection of two images with elements of computer graphics, allows you to determine all the technical nuances of working with deep information. In addition, an algorithm is provided that can be used to double-convert a 2D image and perform basic digital compositing operations.

Field of application: film production, digital compositing, i.e. created modeling schemes will allow in the future to develop effective methods to improve the image at the stage of post-production of films.

## ЗМІСТ

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....                                                                 | 8  |
| ВСТУП.....                                                                                     | 9  |
| 1 ЦИФРОВИЙ КОМПОЗИТИНГ. ОСОБЛИВОСТІ РОБОТИ<br>З DEEP-ІНФОРМАЦІЄЮ.....                          | 10 |
| 1.1 Загальна інформація про цифровий композитинг. Історія<br>створення технічного прийому..... | 10 |
| 1.2 Основні принципи композитингу.....                                                         | 13 |
| 1.2.1 Базові операції композитингу.....                                                        | 14 |
| 1.2.2 Робота з маскою зображень.....                                                           | 17 |
| 1.3 Загальна інформація про необхідність створення каналу<br>глибини.....                      | 21 |
| Висновки до розділу 1.....                                                                     | 23 |
| 2 DEEP-КОМПОЗИТИНГ. БАЗОВІ ПІДХОДИ ПРИ РОБОТІ З<br>ЗОБРАЖЕННЯМ.....                            | 24 |
| 2.1 Підходи до deep-композитингу.....                                                          | 24 |
| 2.2 Основні програмні процедури deep-композитингу.....                                         | 26 |
| 2.2.1 Створення або зчитування зображень з даними глибини.....                                 | 27 |
| 2.2.2 Перетворення плоского зображення у Deep.....                                             | 27 |
| 2.2.3 Операція об'єднання двох Deep.....                                                       | 28 |
| 2.2.4 Корекція кольору у DeepImage.....                                                        | 29 |
| 2.2.5 Трансформація, переформатування, обрізання зображень.....                                | 29 |
| 2.2.6 Маскування.....                                                                          | 31 |
| 2.2.7 Запис deep-зображення.....                                                               | 33 |
| Висновки до розділу 2.....                                                                     | 34 |
| 3 АНАЛІЗ МОЖЛИВОСТЕЙ ІНСТРУМЕНТІВ ПРИ РОБОТІ<br>З DEEP-DATA.....                               | 35 |
| 3.1 Спеціальні ноди роботи з Deep-data.....                                                    | 35 |
| 3.2 Трансформація та об'єднання зображень з deep-data.....                                     | 43 |

|                                                   |    |
|---------------------------------------------------|----|
| 3.3 Аналіз отриманих результатів моделювання..... | 46 |
| Висновки до розділу 3.....                        | 47 |
| ВИСНОВКИ.....                                     | 48 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....                     | 50 |
| ДОДАТОК А. SUMMARY.....                           | 51 |

**ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

|      |                            |
|------|----------------------------|
| 2D   | - 2-dimensional;           |
| 3D   | - 3-dimensional;           |
| RGB  | - Red, Green, Blue;        |
| RGBA | - Red, Green, Blue, Alpha. |



## ВСТУП

Під час роботи з кіно- та фотоматеріалами в професійних студіях дуже часто виникає нагальна потреба об'єднання відзнятого матеріалу та даних, які зроблено в спеціальних програмах з комп'ютерної графіки. Добре, коли зйомка відбувалась на основі кінокамери з підтримкою 3D. Але так буває не завжди і матеріал з камери, наприклад Betacam, важко якісно об'єднати з картинкою, яка отримана, наприклад, з програми 3DMax. Для вирішення цієї задачі і присвячена особлива група нод в програмі Nuke під назвою Deerp.

Крім цього, іноді при об'єднанні різних об'єктів на етапі формування сцени кадру зображення маємо лише 2D-зображення і в цьому випадку може бути необхідна інформація про пікселі зображення. Іншими словами, на етапі обробки та компонування зображень може знадобитись інформація про взаємне розташування найменших елементів зображення і про віддаленість цих елементів від точки розташування об'єктиву камери під час проведення зйомки. І для отримання цієї інформації також може бути у нагоді спеціальний інструментарій програми Nuke від компанії The Foundry. Таким чином, виходячи з того, що сучасні кінофільми вже можуть бути навіть перенасичені різноманітними елементами комп'ютерної графіки, то можна вважати обрану тему дипломної роботи актуальною.

## **1 ЦИФРОВИЙ КОМПОЗИТИНГ. ОСОБЛИВОСТІ РОБОТИ З DEEP-ІНФОРМАЦІЄЮ**

Сучасні передові технології в цифровому кіно передбачають на етапі пост виробництва роботу з відзнятим кіно матеріалом. І інтеграція зображень з різних джерел може займати левову частку у часі при обробці відзнятих сцен. Особливо це відіграє роль, коли необхідно об'єднати в єдину сцену матеріали, які були отримані на основі різних кінокамер, або взагалі які були створені в різних програмах з комп'ютерної графіки. В рамках дослідження буде вивчено не лише, як можна провести цифровий композитинг, але й як врахувати взаємне розташування між собою об'єктів на сцені. Іншими словами, буде досліджено як інформація про глибину зображення (deep data) дозволяє передати об'ємність у випадку, коли саме зображення, або його складові є 2D-елементами.

### **1.1 Загальна інформація про цифровий композитинг. Історія створення технічного прийому**

Цифровий композитинг, як ми будемо його обговорювати, має справу з процесом інтеграції зображень з декількох джерел в єдине ціле. Хоча багато з цих методів початково були призначені до нерухомих зображень, буде розглянуто інструменти і методи, які також корисні для великих динамічних послідовностей зображень. Цифровий композитинг - цифрова маніпульована комбінація щонайменше з двох вихідних зображень для отримання інтегрованого результату. Безумовно, найскладнішою частиною цього процесу цифрового композитинга є отримання інтегрованого результату-зображення, яке не видає, що його створення було пов'язано з декількома вихідними елементами. Зокрема, ми зазвичай намагаємося створити послідовності, які могли б бути правдоподібно сфотографовані без використання будь-якої постобробки. У розмовній мові вони повинні

виглядати "реальними". Навіть якщо елементи в сцені явно нереальні (наприклад, величезні комахи, які стоять на гігантському персику), потрібно бути здатним повірити, що все в сцені було сфотографовано в один і той же час, однієї і тієї ж камерою. Телевізійні рекламні ролики швидше за все були скомпоновані разом з численних джерел. Проте, чи використовується процес цифрового композитинга для бізнес презентацій або для спеціальних ефектів художнього фільму, використовувані методи та інструменти слідує одним і тим же основним принципам. Справжня майстерність цифрового композитинга включає в себе як технічне, так і художні навички. Як і в будь-якому виді мистецтва, художник, безумовно, повинен мати гарне уявлення про техніку, вміння поводитися з інструментами, які будуть використовуватися. Ці інструменти потенційно можуть включати в себе будь-яку кількість різних апаратних і програмних композиційних систем. Але також потрібне розуміння всього процесу композитингу, а не тільки конкретних інструментів. Це включає в себе все, починаючи з розуміння того, як це візуальне дані представлені в цифровому форматі для отримання знань про те, як камера реагує на колір і світло. Поєднання декількох джерел для отримання нового зображення - це, звичайно, не нове, і робилося це задовго до того, як комп'ютери увійшли в кінематограф. Тож розглянемо основні моменти з історії композитингу. Це необхідно щоб зрозуміти які підстави були в цифровому кіно до переходу на деер-композитинг.

Шведський фотограф Оскар Рейландер працюючи у своїй студії в Англії вибірково об'єднав зображення з 32 різних скляних негативів, щоб отримати єдиний масивну палітру. Це один з найбільш ранніх прикладів того, що стало відомо як "комбінований знімок". Кінематографічна фотографія з'явилася в кінці 19 сторіччя і з часом були побудовані оптичні принтери, які могли вибірково комбінувати кілька частин плівки, і таким чином народився оптичний композитинг. Ряд ранніх прикладів оптичного композитинга можна знайти у фільмі "Кінг-Конг" 1933 року. Зображення, показане на рис. 1.1, насправді є складовим зображенням, яке було створено

в такий спосіб: спочатку була сфотографована гігантська мавпа, 16-дюймова мініатюра, яка анімована з використанням техніки стоп-руху.



Рисунок 1.1 – Фільм «Кінг-Конг»

Цей процес включає в себе фотографування моделі покадрово за раз, зміна пози або положення персонажа між кожним кадром. В результаті при відтворенні з нормальною швидкістю виходить безперервно анімований об'єкт. Після того, як ці кадри були розроблені і опрацьовані, вони були зпроектовані на великий задній проекційний екран, який був розташований на повнорозмірній сцені [1]. Потім, дія на передньому плані (актриса на дереві) фотографувалось, а фонові кадри проектувалися за ним, створюючи композиційне зображення. Цей конкретний тип композитингу відомий як ефект в камері, так як не було ніякої додаткової постпродакшн-роботи, необхідної для завершення зйомки. Інші сцени у фільмі були виконані з використанням ранньої форми синьою фотографії екрану, де цільне місце і фон були сфотографовані окремо, а потім об'єднані на етапі оптичної друку.

В даний час оптичний композитинг фактично замінений комп'ютерними системами загального призначення і деякими програмними засобами, але самі концепції практично не змінилися. І це фактично була перша спроба до переходу з створення глибини зображень при поєднанні з різних джерел.

## 1.2 Основні принципи композитингу

Взаємне розташування об'єктів в сцені можна закодувати через колір цих об'єктів. Тобто, наприклад домовимось, що два об'єкти які пофарбовано у різні відтінки одного і того самого кольору геометрично розташовані на різній відстані від знімальної камери.

Відомо, що одна з найбільш важливих речей, яку робить фахівець з композитингу, інтегруючи елементи в сцену - це збалансувати все з точки зору кольору, яскравості і контрастності. Здібності різних людей судити про колір можуть досить сильно відрізнятися, як правило, здібність людини правильно розпізнавати кольори може бути здобута через великий практичний досвід. Найважливішим фактом є те, що на сприйняття цих речей можуть істотно впливати зовнішні фактори. Класичний приклад цього принципу показаний на рис. 1.2 [2,3].

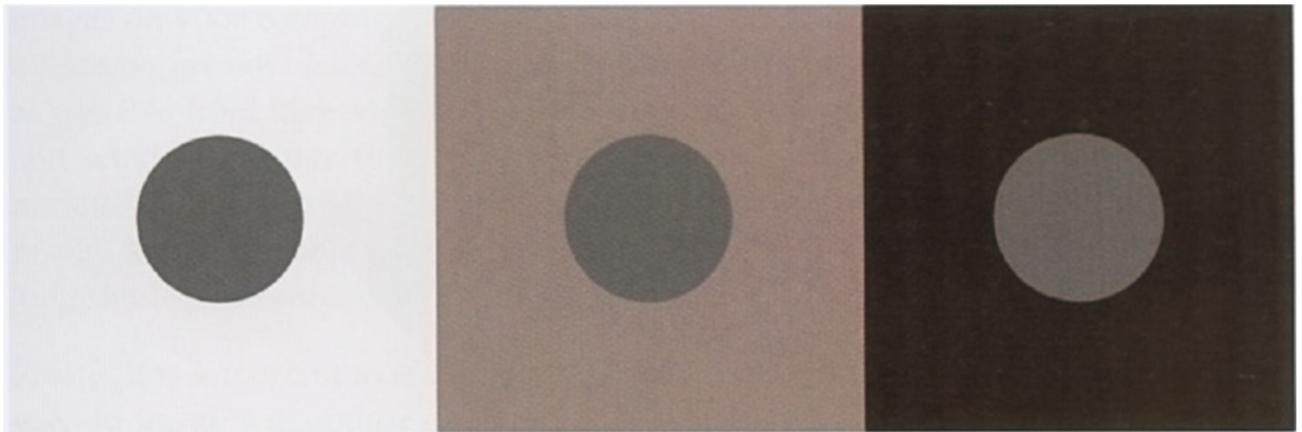


Рисунок 1.2 – Приклад одночасного контрасту

Хоча внутрішнє коло має один і той же колір у всіх трьох випадках, він здається більш яскравим, коли оточений більш темним фоном. Вчені, які вивчають сприйняття, знають цей феномен як "принцип одночасного контрасту". Те ж саме відноситься і до кольорових образів. Колір буде здаватися більш яскравим і насиченим, коли він оточений додатковими кольорами - наприклад, червоним, оточеним блакитним або зеленим, - і

взагалі будь-яке колірне сприйняття буде залежати від оточуючих кольорів, Ці моменти можна використати і при створенні глибини зображення.

Оскільки зорова система людини сприйнятлива до впливів навколишнього середовища, кожна людина робить все можливе, щоб спробувати контролювати середовище щоразу, коли судить про колір. Треба намагатися розглядати різні зображення в максимально схожому контексті. При порівнянні двох зображень на моніторі комп'ютера, треба переконатися, що оточення, як на екрані, так і за його межами, не спотворює судження про кольори. Також, існує ряд цифрових інструментів для вибірки кольору конкретного пікселя або групи пікселів. Важливо розуміти, що часто неможливо повністю уявити, як зображення буде виглядати на певному пристрої, якщо воно проглядається на іншому пристрої.

### **1.2.1 Базові операції композитингу**

Розглянемо ситуацію, коли необхідно об'єднати два або більше послідовностей зображень. Це вимагає використання спеціальних математичних операторів, які можуть працювати з більш ніж однією вхідною послідовністю, і тому ці оператори називаються операторами з декількома джерелами. Буде розглянуто лише деякі з найбільш важливих інструментів комбінування зображень. Зрозуміло, що такі самі підходи характерні і до так званих deer-зображень.

**Додавання.** Однією з найбільш поширених операцій роботи з зображеннями є процедура додавання. Додавання двох зображень разом включає в себе додавання кожного пікселя в першому зображенні до відповідного йому пікселя в другому зображенні. На рисунку 1.3 показані два зображення, які ми будемо використовувати для ілюстрації операції додавання.

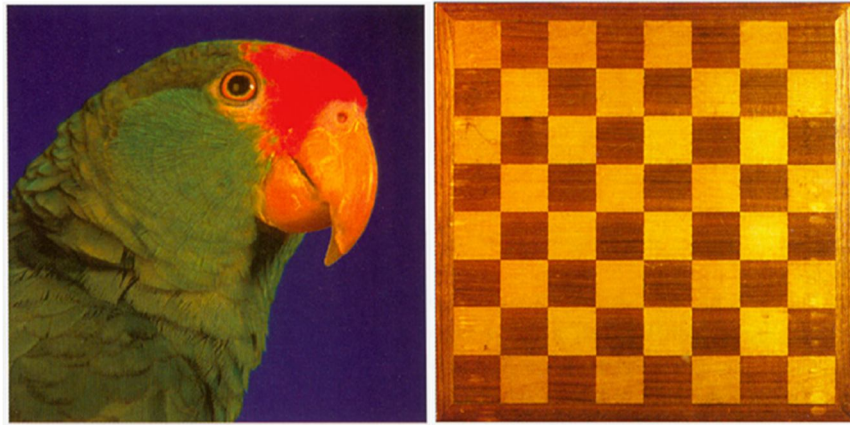


Рисунок 1.3 – Приклад зображень для об’єднання

Результат операції додавання зображень показано на рисунку 1.4.

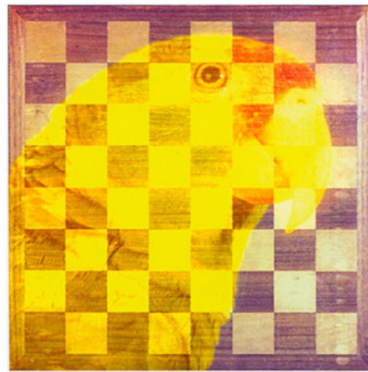


Рисунок 1.4 – Результат операції додавання

Як можна побачити з рисунку 1.4, результат приблизно схожий на фотографічну подвійну експозицію. Використовуючи стандартну нотацію, оператор “Add multi-source” буде просто представлений у вигляді:

$$O = A + B,$$

де  $A$  і  $B$  – це два вихідних зображення, а  $O$  – результуюче зображення.

Аналізуючи отриманий результат тим не менш не можна сказати, що шахівниця знаходиться попереду чи позаду відносно папуги. Відповідь на це питання можна отримати, використовуючи додаткову інформацію про

глибину результуючого зображення. Власне на це і покликана робота з deer-зображеннями [2,3].

**Віднімання.** Оператор віднімання призводить до того, що кожен піксель на зображенні  $A$  віднімається з відповідного пікселя на зображенні  $B$ . Слід відмітити, що віднімання не є симетричним оператором. Порядок, в якому вказані два зображення, важливий для отриманого результату. Таким чином,  $(B + A)$  дорівнює  $(A + B)$ , але  $(A - B)$  не дорівнює  $(B - A)$ . Це показано на рисунку 1.5.

Рисунок 1.5 визначає, що  $A$ -це шахова дошка, яка віднімається з зображення папуги  $B$ . Тоді як, на рисунку 1.6 – зображення птиці, віднімається від зображення шахової дошки. За замовчуванням оператор віднімання або вирізає всі значення, які йдуть нижче нуля, або (якщо ви працюєте в середовищі з плаваючою комою) створює негативні значення. Також корисно мати можливість вказати, що оператор повертає абсолютне значення результату, в якому негативні числа  $A$  повторно перетворюються в позитивні. Віднімання, яке повертає абсолютне значення результату, стає симетричною операцією, тому вам більше не потрібно турбуватися про порядок двох послідовностей зображень, які ви вказуєте.



Рисунок 1.5 – Результат операції віднімання зображень  $A-B$



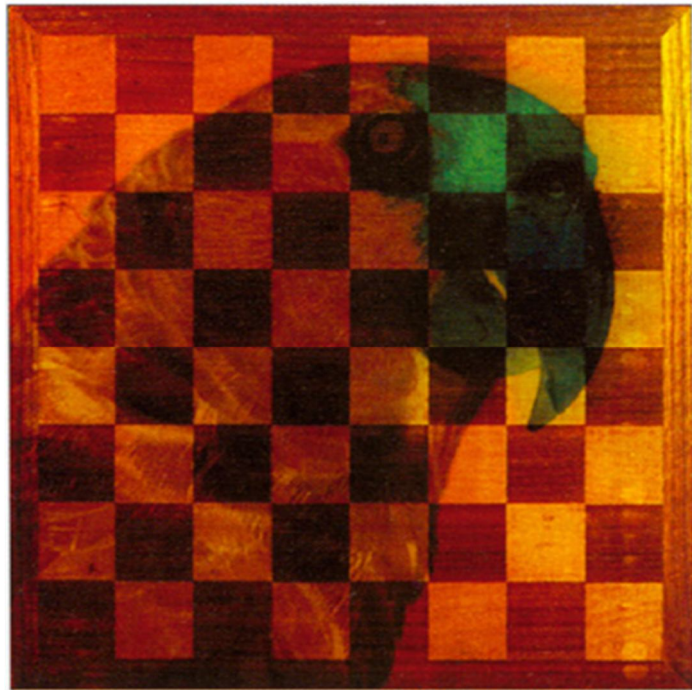


Рисунок 1.6 – Результат операції віднімання зображень В-А

### 1.2.2 Робота з маскою зображень

Об'єднання різних послідовностей зображень має бути процесом з максимально можливим контролем. Потрібно вміти контролювати прозорість різних шарів так, щоб вони не повністю приховували все, що вони покривають. Необхідно розробити метод визначення та управління цими атрибутами, який буде інтуїтивно зрозумілий і узгоджений з рештою обробкою зображень. Маска використовується під час композитингу, коли потрібно, щоб лише частина певного зображення була включена в результат [4]. Маскою зазвичай вважаються одноканальні зображення у відтінках сірого. Тут немає необхідності в трьох окремих каналах, як це буває при визначенні кольору, так як прозорість для будь-якого заданого пікселя може бути описана одним числовим значенням в діапазоні  $0 \div 1$ .

В ідеалі композиційна система буде зберігати одноканальну маску максимально ефективно, але при цьому дозволить фахівцю при необхідності

перетворювати її як звичайне RGB-зображення. Залежно від програмного пакету і обраного формату файлу маска також може бути об'єднана з трьохканальним кольоровим зображенням в якості дискретного четвертого каналу. Коли матове зображення є частиною чотириканального зображення, воно називається альфа-каналом.

Розглянемо дуже простий приклад використання маски, враховуючи вихідні два зображення (рис. 1.3) і маску, яка показана на рис.1.7. Ми будемо використовувати цю маску, щоб ізолювати або витягти частину переднього плану, а потім помістити його на задній план. Саме ці алгоритми використовуються і при роботі з деєр-інформацією. Отримане зображення показано на рис. 1.8.



Рисунок 1.7 – Маска зображення

Як можна побачити, області, які є білими (мають значення пікселя 1.0) в альфа каналі, використовуються для вказівки того, що відповідна область зображення переднього плану повинна бути збережена в повній непрозорості. Це, як кажуть, "тверда" область матерії. І навпаки, чорні

області маски використовуються для вказівки того, що відповідні пікселі на зображенні переднього плану будуть прозорими або ефективно віддалятися, коли воно розміщується поверх фону. Математично ця операція виглядає:

$$O = (A \times M) + [(1 - M) \times B]$$



Рисунок 1.8 – Результат операції маскування зображення

Аналізуючи отриманий результат роботи з маскою (рис.1.8) у глядача може скластись відчуття, що зображення папуги знаходиться на передньому фоні, тоді як зображення шахівниці розташовано на задньому фоні. Тобто в даному прикладі можна вважати, що є певні вказівки щодо “взаємного” псевдо розташування об’єктів. Для роботи з реальною глибиною зображень в сцені в рамках дослідження використаємо спеціалізований програмний продукт компанії NukeX, який доступний в некомерційному виконанні на сайт компанії – розробника (версія non-commercial).

Nuke – продукт компанії The Foundry, який відноситься до так званих «композерських» програм [2,3]. Програма застосовується для редагування

відео або послідовності зображень (секвенцій) і застосовується при обробці, накладенні ефектів, фінальній збірці знятого відео- і кіноматеріалу, і при створенні рекламних роликів, мультфільмів, кінофільмів, а також для інших завдань. Продукт є явним представником нодової архітектури, що пояснює його зрозумілість і логічність.

Nuke використовується в ряді найбільших студій візуальних ефектів світу, таких як: Digital Domain, Walt Disney Animation Studios, DreamWorks Animation, Sony Pictures Imageworks, Sony Pictures Animation, Framestore, Weta Digital and Industrial Light & Magic та ін.

Інтерфейс програми для користувача поділяється на декілька областей, вікна яких можна переміщувати. Задля забезпечення зручності роботи з програмою у структурі виведено окрему панель інструментів, де розміщені усі основні елементи програми Nuke і в якій в програмі поділено на групи. Ця вертикальна панель програми знаходиться з лівого боку робочого поля програми і називається Toolbar (рис.1.9).

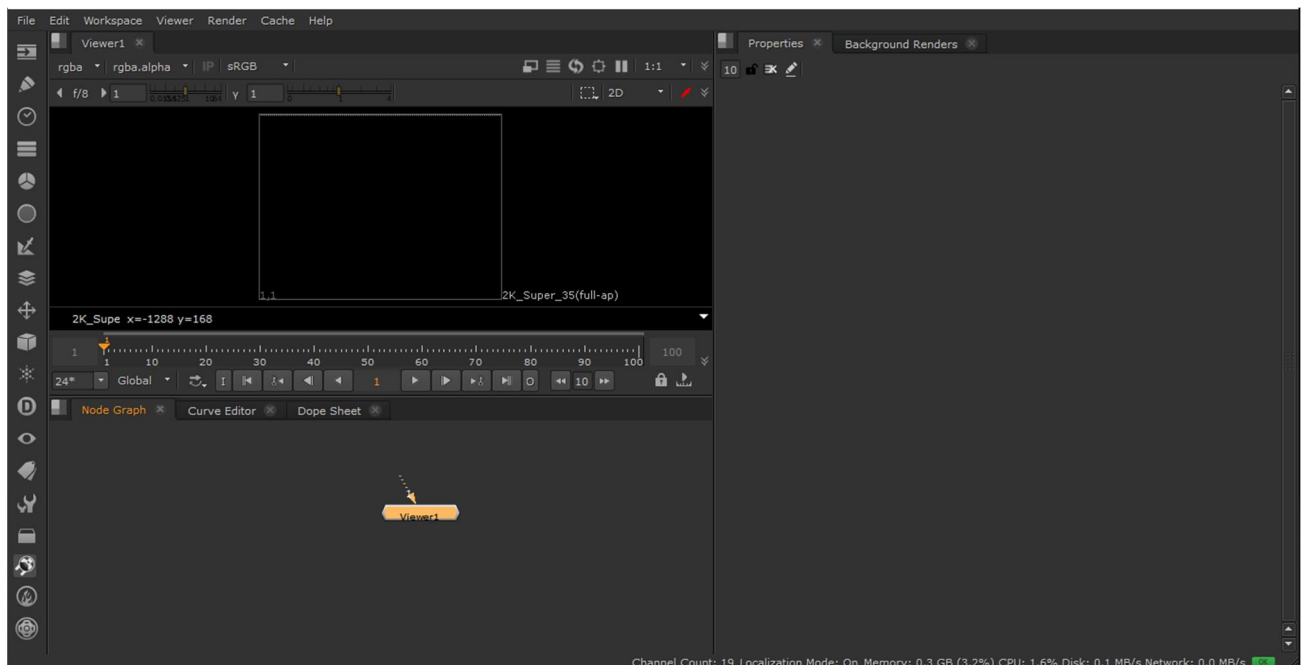


Рисунок 1.9 – Інтерфейс програми

Панель Node Graph – панель де розміщуються усі елементи створеного проекту. При цьому, ці елементи є графічними і пов'язані з компонентами

мультимедійного контенту. Прикладом такого зображення може бути зображення, відео файл або відео ефект. Зв'язок між елементами показують у програмі через лінії зі стрілками.

### **1.3 Загальна інформація про необхідність створення каналу глибини**

Як було вже відмічено вище, композітинг – це об'єднання двох або більше зображень для створення нової картини (Porter & Duff, 1984). Це може бути так само просто, як замінити небо, щоб зробити його схожим на хмари, що збираються для шторму, і так само складно, як живі і цифрові персонажі біжать по лісі, будучи частково прихованими гілками і листям. Сьогодні існує кілька способів об'єднання двох або більше зображень разом, щоб вони з'являлися на одному і тому ж кадрі. Ротоскопіювання [4], зелені екрани і автоматично згенеровані матові поверхні для цифрових 3D-об'єктів – лише деякі з сучасних методів в композітингу [5,6]. В міру ускладнення цифрових сцен художники часто використовують кілька програм для створення остаточного зображення. Деякі 3D-програми, такі як Autodesk Maya® і 3DS Max®, зазвичай інтегруються з композиторськими програмами, такими як Adobe After Effects® або The Foundry Nuke®. Частково це пов'язано з тим, що невеликі помилки моделювання не повинні призводити до необхідності повторного рендеринга всієї сцени. Композітинг також прагне уникнути згладжування і збереження м'яких країв. Кожен піксель складається з RGB, щоб визначити його колір. Також потрібно "коефіцієнт змішування", який управляє інтерполяцією двох зображень через непрозорість або альфа-канал, що визначає прозорість. Базовий формат для даного зображення полягає в тому, щоб кожен піксель позначався його червоними, зеленими і синіми значеннями. Це зазвичай представляється у вигляді  $(r, g, b)$  з числовими значеннями, які замінюють кожен аргумент, і кожен аргумент має інший "канал". Таким чином, основне зображення буде

мати червоний, зелений і синій канали. Виходячи з цього, можна додати четвертий канал, щоб дати інформацію про альфа-канал або про те, наскільки прозорий цей конкретний піксель. Це називається форматом RGBA, а не базовим форматом RGB [7]. Перевага формату RGBA полягає в тому, що обчислення колірного каналу і Альфа-каналу можуть виконуватися одночасно, на відміну від формату RGB, де альфа-канал не вбудований і легко доступний.

Наприклад, з огляду на те, що при об'єднанні двох зображень у вас є червоний колір, який тільки наполовину покриває піксель фону. Можна було б змусити програму обчислити новий колір, взявши червоний піксель переднього плану, помножуючи Альфа-канал на кольори, і створити новий піксель, який буде додано до якості. Однак це змушує комп'ютер робити непотрібні обчислення, зокрема множення, щоб отримати справжній колір. Замість цього можна попередньо помножити Альфа-канал на кольори, перш ніж вводити його в програмне забезпечення, і таким чином, маючи власний імпорт. Противники проти цього говорять, що це непотрібна кількісна оцінка колірної інформації, оскільки непрозорість наближається до нуля. Однак в композитингу множення має бути зроблено в якийсь момент процесу. Крім цього, Портер і Дафф також закріпили кілька функцій об'єднання зображень, таких як: "над, під, всередині, зовні, поверх".

Взявши ідею додавання більшої кількості каналів з більш глибокої інформацією, Портер і Дафф також були "зацікавлені в застосуванні концепції композітинга до алгоритмів Z-буфера, де інформація про глибину зберігається в кожному пікселі". Рік по тому Дафф продовжив цю конкретну дослідницьку тему і зосередився на створенні двох зображень з використанням Z-буфера і невеликих програм. Кажучи про Z-глибину, простіше за все думати про розташування уздовж осі  $OZ$ , яка існує в 3D-програмах. Ось  $OX$  спрямована з одного боку в інший або зліва направо, вісь  $OY$ -вгору-вниз, а вісь  $OZ$ -спереду-назад.

Тому, коли обговорюється  $Z$ -глибина, то це зазвичай ставиться до того, наскільки далеко від камери на осі  $OZ$  знаходиться об'єкт або точка. Слід також зазначити, що немає жодної абсолютної осі  $OZ$ . Замість цього існує нескінченне число  $Z$ -осей, і, кажучи про  $Z$ -глибину, найпростіше візуалізувати факт, що всі вони починаються з камери і виходять на сцену.

Згладжування може бути досягнуто шляхом об'єднання алгебраїчних рівнянь для композитинга і  $Z$ -буфера [8].

Кінематографічні системи, розроблені LucasFilm і штатом Огайо для композитінга, вимагають лінійно поділюваних поверхонь. В черговий раз збільшивши кількість каналів в зображенні, композери тепер мали можливість працювати з зображеннями, які мали червону, зелену, синю, альфа-і  $Z$ -глибину інформації, вбудованою для кожного пікселя. Для з'єднання двох зображень використовуються двійкові оператори. Хоча цей метод класифікується як "простір зображень грубої сили" і відкидається через непрактичність, він є одним з найпопулярніших алгоритмів для прихованих поверхонь. Дафф обчислив  $Z$ -буфери в кожному кутку пікселя і порівняв їх, щоб визначити, яка інформація повинна бути показана спереду. Це стало основою глибинного композитингу, який представляє собою композитинг з використанням даних глибини, вбудованих в зображення. Однак глибинний композитинг все ще стикається з проблемами щодо розмиття руху і згладжування.

**Висновки до розділу 1.** В першому розділі дипломної роботи розглянуто базові принципи та підходи, які лягли в основу сучасного цифрового композитингу. На початку розділу простежено історію створення оптичного композитингу, як першої спроби до створення глибинного композитингу. Додатково проаналізовано основні операції та відповідні оператори які використовується при об'єднанні зображення. Відмітимо, що подібні алгоритми простежуються і при роботі з *deep*-інформацією в спеціалізованому пакеті NukeX.



## 2 DEEP-КОМПОЗИТИНГ. БАЗОВІ ПІДХОДИ ПРИ РОБОТІ З ЗОБРАЖЕННЯМ

### 2.1 Підходи до deep-композитингу

Зображення, які містять інформацію про глибину об'єктів через створені в їх структурі окремий канал глибини (deep-channels, depth) в кіно та комп'ютерній графіці зазвичай використовуються для створення ефектів, і генерування складних сцен. Для прикладу проаналізуємо кадр з фільму «Планета мавп», у якій багато об'єктів перекривають один одного (рис.2.1). Завдяки технології глибокого композитингу можна змінювати розміщення об'єктів та персонажів, не виконуючи при цьому кожного разу повторного рендерінгу.

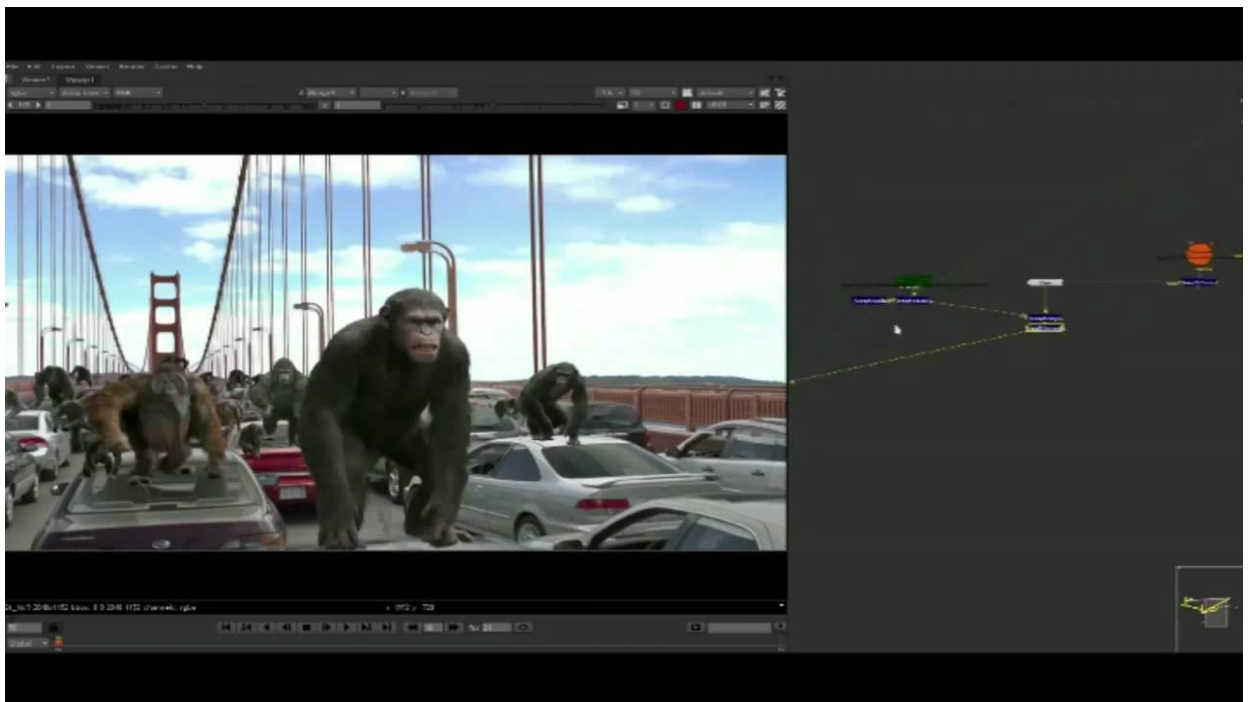


Рисунок 2.1 – Приклад використання deep-композитингу

Але також є і альтернативні методи використання цієї технології. Пристрої і технології, які відтворюють і записують глибину в реальному світі, стали більш популярними останнім часом. Цезар Паломо і Марсело



Гаттас у 2010 році запропонували ситуацію, в якій подію було знято з декількох ракурсів камери, але кут, який повинен бути показаний, існує між двома ракурсами. Їх дослідження запропонувало алгоритм, який оброблявся на графічному процесорі для створення нового зображення, де опорні види були зліва і праворуч від бажаного виду, враховуючи, що всі використовувані опорні види також мали “карти глибини”, характерні для них. Паломо і Гаттас визначили “зображення глибини” як кольорове зображення з одним значенням глибини на піксель. Карти глибин, отримані за допомогою нових технологій глибинного зондування, були лінеаризовані в півтонування. Метод, розроблений авторами, був заснований на тривимірній деформації, в якому, переміщаючись з системи координат камери в систему координат світу, 3D-сітка проектується з камери, а потім деформується в новий ракурс. Ця техніка сама по собі зазвичай зазнає невдачі через те, що області, які були закриті, стають відкритими і не мають ніякої інформації *RGBA* для них [8]. Щоб “змішати” два зображення між собою, кожен піксель з обох еталонних зображень порівнюється між собою. З огляду на граничне значення, якщо один піксель знаходиться ближче до нової камери, ніж інший (відповідно до їх інформацією про карту глибини), то цей піксель складається поверх фонового пікселя.

Інший метод застосування “глибоких” зображень був розроблений для імітації глибини різкості, загального ефекту, властивого камерам реального світу. Глибина різкості має в основі ідею, що є одна площина або зріз повністю в фокусі, і чим далі від цього зрізу ви потрапляєте в *Z*-глибину, тим більш розмитим є піксель. У 2004 році Будин опублікував статтю, присвячену алгоритму використання *Z*-глибини для створення штучної глибини різкості в цифрових зображеннях. Перший крок полягав у тому, що кожен піксель оцінювався і, в залежності від того, наскільки далеко від заданої *Z*-глибини він знаходиться, застосовувалося Гауссівське розмиття різної сили з центром на пікселі. Другий крок полягав у тому, що значення *RGBA* кожного пікселя нормалізувалося в залежності від того, наскільки

далеко від стандартного значення "1" вони перебували. Це було зроблено після того, як всі гаусові розмиття були застосовані. Для реалізації свого методу Будин розробив плагін для композиційної програми Shake. Його плагін також мав функцію попередньої візуалізації, яка затемнювала зображення в залежності від того, чи перебували вони всередині площин для фокусування. Одна з проблем його методу полягала в тому, що відсутність рендеринга фону або відсутність 3D-фону викликали проблеми з жорсткими краями. Глибина різкості-це властивість, властиве багатьом камерам і відеокамер, але це не означає, що не існує методів створення глибини різкості на нецифрових зображеннях після того, як вони були зроблені. Джейкобс, Бек і Лівої (2012) розробили методику, яка при наявності декількох фотографій на різних глибинах поля або фокусних діапазонах дозволяє користувачеві створювати власний фокусний діапазон. Необхідні карти, повідомляють програмному забезпеченню, як "у фокусі" знаходиться ця частина зображення. Варто відзначити, що це не зовсім карти глибини, так як найсильніше значення в ній може знаходитися в центрі зображення. Проте врешті-решт вони змогли створити зображення, в яких всі предмети були в фокусі, не втрачаючи при цьому якості.

## **2.2 Основні програмні процедури deer-композицінгу**

Методика deer-композицінгу нерозривно пов'язана з операціями окремих пікселів зображень. В даному випадку закладається інформація, як один піксель у визначеному місці одного зображення розташований по відношенню пікселя іншого зображення, який розташований в цих самих координатах. Принципи deer-композицінгу полягають у наступних процедурах:

- імпорт зображень з даними глибини або створення каналу глибини у плоскому зображенні;
- об'єднання цих зображень з глибиною в одному кадрі;

- корекція кольору;
- перетворення, переформатування, обрізання зображень;
- маскуванню зображення;
- створення deer-зображення.

Розглянемо деякі найпростіші процедури, які пов'язані з deer-інформацією зображення або послідовності зображень. За основу розгляду візьмемо ноди які входять до групи Deer програми Nuke.

### **2.2.1 Створення або зчитування зображень з даними глибини**

Програма Nuke дозволяє зчитувати зображення з даними глибини у двох форматах DTEX та OpenEXR. DTEX – це формат зображень які були створені у програмі компанії Pixar, PhotoRealistic RenderMan. Формат OpenEXR – формат даних, які підтримує і значення даних глибини. Оскільки OpenEXR є значно більш поширений формат, буде описано як імпортувати саме його. Алгоритм процедури наступний:

1. Треба створити ноду DeerRead, і для цього натиснути Deer (вхід в групу) а потім з переліку обрати ноду DeerRead.
2. Знайти потрібний файл у форматі .exr та натиснути Open.

### **2.2.2 Перетворення плоского зображення у Deer**

Для цього потрібно використати ноду DeerFromFrames. DeerFromFrames створює семпли з даними про глибину з вхідної секвенції. Алгоритм наступний:

1. Приєднати DeerFromFrames до вхідної секвенції зображень. Зображення з інформацією про глибину створюється шляхом розміщення кожного кадру на збільшувальну глибину.
2. Щоб налаштувати результат потрібно скористуватися панеллю налаштувань. На рисунку 2.2 наведено приклад процедури.

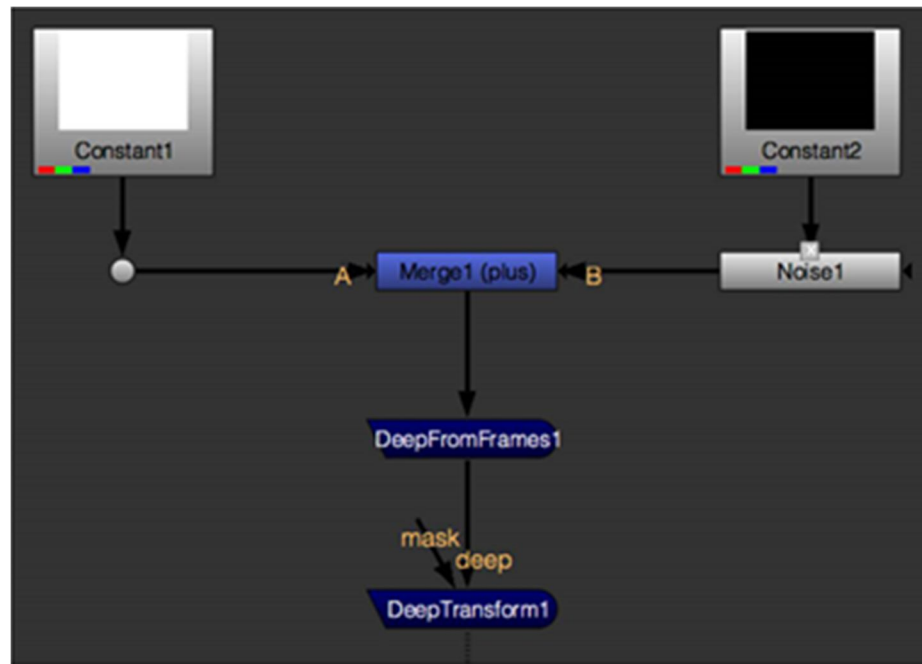


Рисунок 2.2 – Перетворення секвенції зображень, у зображення з даними глибини

### 2.2.3 Операція об’єднання двох deer

Операція об’єднання двох deer-зображень проходить на основі ноди DeerMerge. Таким чином, кожен вихідний піксель містить всі вибірки з одного і того ж пікселя кожного вхідного зображення.

Для того щоб об’єднати потрібно провести наступні дії:

1. Під’єднати вхідні зображення до ноди DeerMerge як показано на рисунку 7.2
2. У налаштуваннях ноди DeerMerge поставити операцію combine (рис.2.3).

Також є можливість встановити “прапорець”, відкинути приховані зразки, на панелі властивостей, щоб не вмикати зразки, які повністю закриті ближчими до камери зразками, альфа-значення котрих дорівнює одиниці.

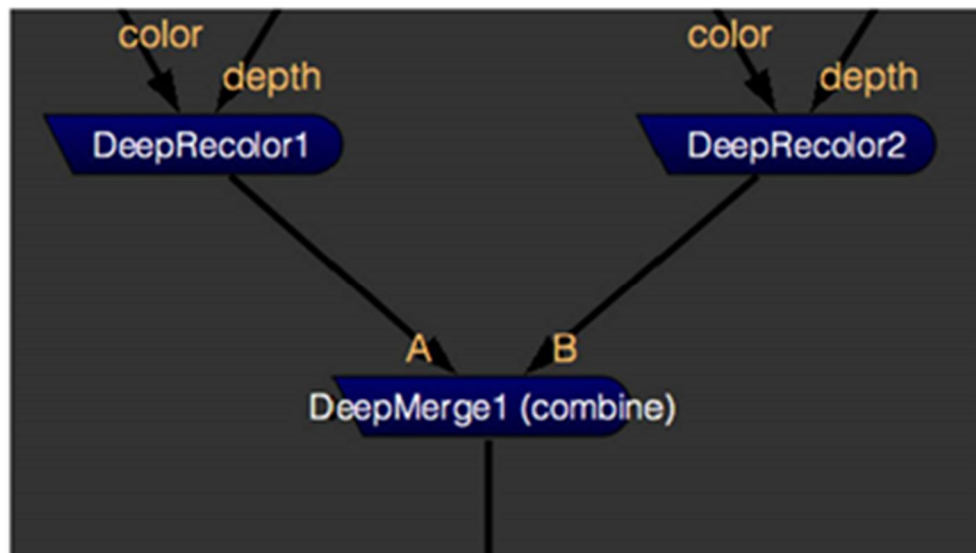


Рисунок 2.3 – Використання ноди DeepMerge для об’єднання двох deep-зображень

#### 2.2.4 Корекція кольору у DeepImage

Нода DeepColorCorrect (рис.2.4) дозволяє корегувати колір кожного пікселя у кожному семплі. Існують керуючі набори для налаштування тіней, півтону і відблисків, а також є функція налаштування всього цього відразу. Є можливість використовувати криві вкладці Ranges для управління діапазоном зображення, який має бути у тіні, півтонах і відблисках.

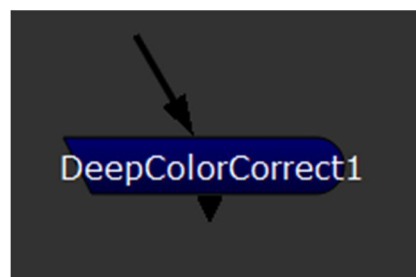


Рисунок 2.4 – Нода DeepColorCorrect

#### 2.2.5 Трансформація, переформатування, обрізання зображень

Трансформація deep-зображень виконується за допомогою ноди DeepTransform. За допомогою цієї ноди можна налаштовувати положення

зображення в кадрі та змінювати його масштаб глибини. В даному випадку процедура має наступний порядок:

1. Під'єднати вхідне зображення до ноди DeepTransform.
2. Використовуючи *translate x,y,z* можна змінювати місцеположення об'єкту
3. Масштаб глибини змінюється за допомогою *zscale*. Значення більше за 1 збільшують глибину, менше 1 навпаки зменшують.
4. Приєднавши маску у вхід *mask*, є можливість регулювати ефект трансформації від ноди у різних частинах кадру.

Нодою DeepReformat змінюється розмір, масштаб об'єкту в зображенні де є канал глибини. Алгоритм використання ноди наступний:

1. Під'єднати вхідне зображення до ноди DeepReformat.
2. У списку Тип виберіть пункт:
  - *to format* - встановити ширину і висоту вихідного обраного формату. Виберіть потрібний формат у списку формат виведення. Якщо формат ще не існує, ви можете вибрати новий, щоб створити новий формат з нуля.
  - *to box*-встановлює вихідну ширину і висоту для розмірів, які ви визначаєте в пікселях. Введіть значення в полях ширина, висота і піксельний аспект, щоб вказати розміри.
  - *Scale* – встановіть ширину та висоту вхідного зображення, щоб змінити його масштаб. Використовуючи слайдер можна обрати з якою кратністю потрібно відмасштабувати зображення.

Для обрізання зображення використовується нода DeepCrop. Порядок її використання наступний:

1. До DeepCrop під'єднати вхідне зображення.
2. Налаштувати crop box у Viewer на осях x та y для визначення області яка буде обрізана.
3. Використати *znear* та *zfar* для того щоб обрізати семпли на осі OZ.

### 2.2.6 Маскування

Використовуючи ноду DeepSample можна маскувати кожен піксель вхідного зображення у deer-зображення. Процедура наступна:

1. Під'єднати ноду DeepSample до іншої Deer-ноди.
2. Розмістити маркер pos на пікселі який потрібно замаскувати, у Viewer.
3. У вкладці властивосте ноди можна подивитися інформацію про піксель на якому знаходиться маска.

Deer-ноди – зображення, де в одному шарі, в одному каналі одного пікселя є інформація про декілька пікселів, які розтягнуто за глибиною. Тобто, пікселі будуть розташовані в глибину на різні відстані. Ці пікселі, які розташовані за глибиною в рамках основного пікселя отримали назву семплів. Іншими словами, всередині одного пікселя можуть бути декілька семплів (як контейнер інформації). Зрозуміло, що візуально сприймаються вони як один піксель в зображенні (їх проекція на піксель зображення дає один піксель). Різні зображення з різними семплами через ноди групи можна об'єднувати шляхом об'єднання семплів. Для перегляду семплів використовують ноду Deepsample, де вбудовано маніпулятор position (рис.2.5).

В завантаженому зображенні з deer-data через ноду DeepSample можна побачити семпл з відповідним вказівником (рис.2.5).

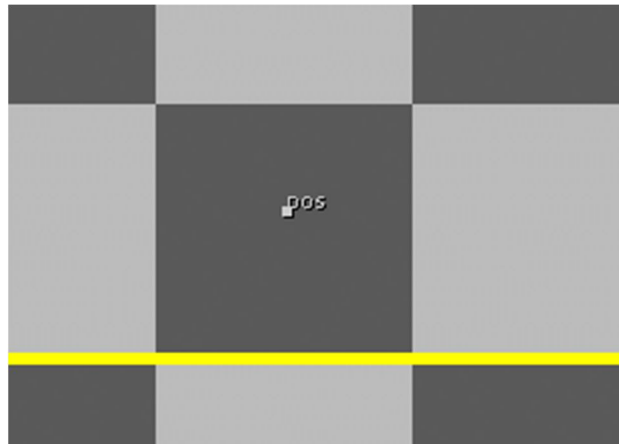


Рисунок 2.5 – Вказівник position

На рисунку 2.6 наведено властивості підключеної ноди DeepSample де показана інформація про семпл.

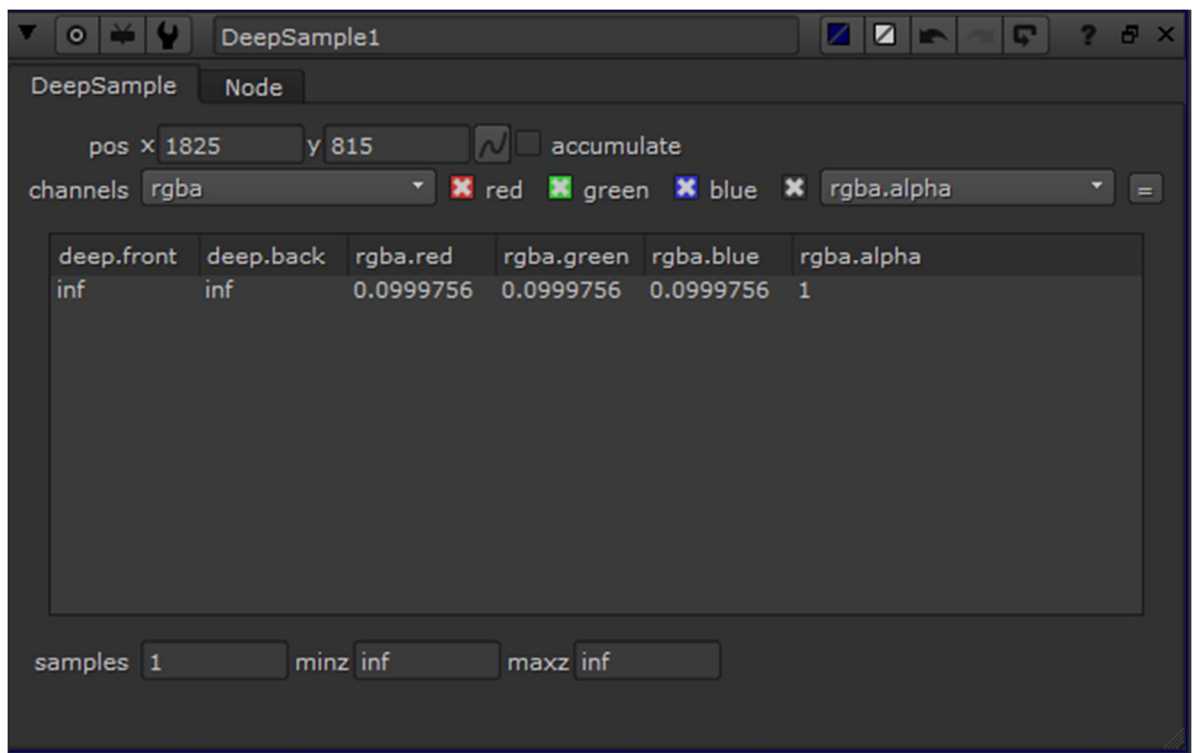


Рисунок 2.6 – Інформація про семпл

Кожен рядок – окремий семпл, який ототожнює піксел, що витягнутий за глибиною. Столпчики *deep.front*, *deep.back* визначають глибину семпла.



Об'єднання зображень з `deep-data`, це фактично упорядкування семплів один відносно іншого. Результат – об'єднання яскравостей семплів з урахуванням напівпрозорості (стовпчик `rgba.alpha`).

### 2.2.7 Запис `deep`-зображення

Записати `deep`-зображення можна у форматі `OpenEXR 2.3` або вище використовуючи ноду `DeepWrite`. Для цього слід провести наступні операції:

1. У вкладці `Deep` обрати `DeepWrite`.
2. На панелі налаштування обрати *file* або *proху* і директорію куди потрібно зберегти файл.
3. Після визначення місця для зберігання файлу, надання файлу імені яке повинно включати в себе формат файлу `.exr`, слід натиснути `OK`. У випадку виведення секвенції зображень також потрібно обрати кількість кадрів, яка буде виведена в імені файлу.
4. Використовуючи `datatype` потрібно обрати бітність глибини для рендерингу секвенції, 16 біт або 32 біти.
5. У пункті `compression` обрати тип компресії з яким буде рендеритись файл.
6. У меню `metadata` потрібно обрати чи будуть метадані включені у інформацію відрендереного файлу. Метадані використовуються для подальшої роботи з цим файлом. Пункти меню `metadata`:
  - `no metadata`- метадані не будуть включені у файл;
  - `default metadata`- стандартні метадані, опціонально можна обрати тайм код, частота кадрів:
    - `default metadata and exr/*`
    - `all metadata except input/*`
    - `all metadata`
7. Натиснути кнопку `Render` в налаштуваннях ноди `Deepwrite`.

**Висновки до розділу 2.** В другій частині дипломної роботи розглянуто передумови які необхідно враховувати при роботі з z-координатою зображень. Додатково виділено базові програмні процедури по використанню основних нод з групи Deer програми Nuke. Наведені положення дозволяють перейти до моделювання схеми, яка описує основні можливості deer-композитингу.

### **3 АНАЛІЗ МОЖЛИВОСТЕЙ ІНСТРУМЕНТІВ ПРИ РОБОТІ З DEEP-DATA**

Композитинг зображень з залученням додаткової інформації про глибину пікселів можна вважати одним з перспективних способів обробки зображень. Загальновідомо, що при роботі з зображенням кожен найменший елемент цього зображення описується, як правило через інформацію, яка розміщена в кольорових каналах та каналі прозорості. Але виявляється, що коли необхідно працювати з комп'ютерною графікою цієї інформації про елементи зображення можуть бути недостатньо. І в цьому випадку створюють для 2D-зображення додатковий канал *depth*, який визначає наскільки даний найменший елемент зображення віддалений від умовної точки розташування камери зйомки. В цьому розділі в межах практичної частини дослідження розглянемо які можливості надає спеціалізована програма NukeX при роботі з глибиною зображення. В якості практичної доцільності розглянемо особливості спеціальних нод програми з групи Deep на прикладі створеної завчасно простої композиції з двох окремих об'єктів, як результат імпорту з програми 3DMax – сплюснутої сфери (шару) (рис.3.1) та вертикальних стовпчиків (колони) (рис.3.2) у форматі зображення HD\_1080 (1920×1080 пікс). І спробуємо на основі наявного інструментарію дослідити як можна працювати з Deep-інформацією.

#### **3.1 Спеціальні ноди роботи з Deep-data**

Як вже згадувалось у попередньому розділі роботи, Deep compose – група нод, які працюють з особливим типом інформації зображення – глибиною. Тобто, Deep-інформація це поєднання даних з плоского зображення та інформації з 3D-простору. Натомість, *depth*-канал – плоска картинка в оригінальному зображенні, яка відображає один канал, де яскравість пікселя визначає відстань за якою цей піксель розташований від

камери зйомки. Тобто, depth – інформація про відстань до пікселя в глибину від камери.

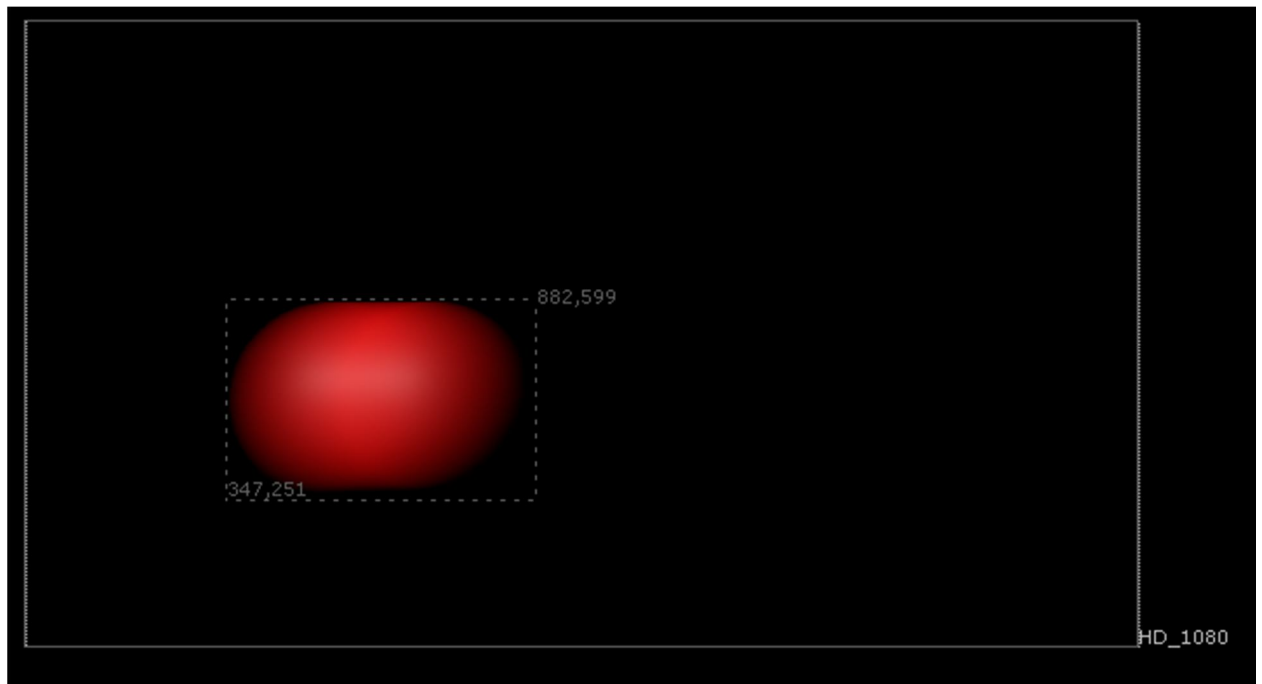


Рисунок 3.1 – Об'єкт для аналізу

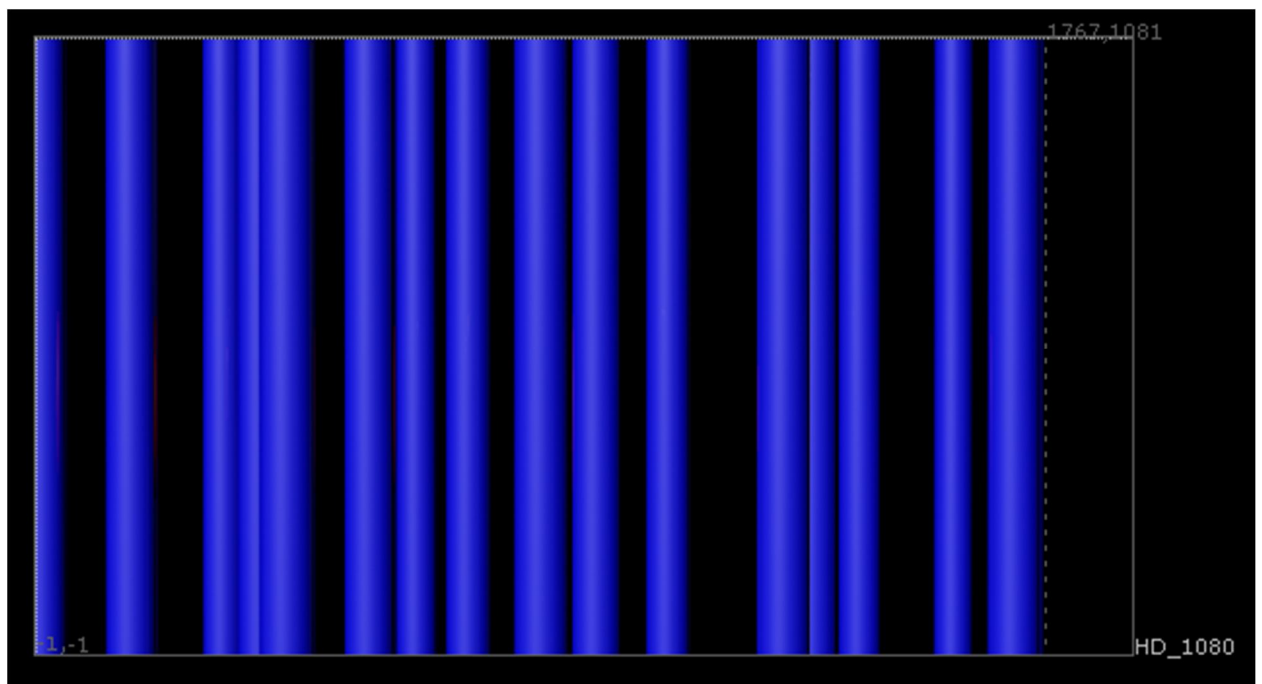


Рисунок 3.2 – Об'єкт для аналізу

На основі даних з другого розділу дипломної роботи (п.2.2), які дозволяють проаналізувати програмні процедури з deer-композитингу була сформована нодова схема в програмі Nuke об'єднання двох зображень, які містять deer-канали в одне зображення. В фінальному результаті шар повинен знаходитися між колонами. Інформацію про глибину для шару та колон буде взята з попередньо відрендерених зображень у форматі .exr. Графічне зображення цієї схеми в блоці backdrop “Deer operations” наведено на рисунку 3.3. Нода DeepReformat використовується для зміни масштабу, розміру.

Нода DeepCopyBBox – це сервісна нода, яка призначена для копіювання координати прямокутного кордону зображення. До входу Deep під'єднується нода DeepRecolor, з якої береться інформація про глибину, а до входу image під'єднується вхідне зображення

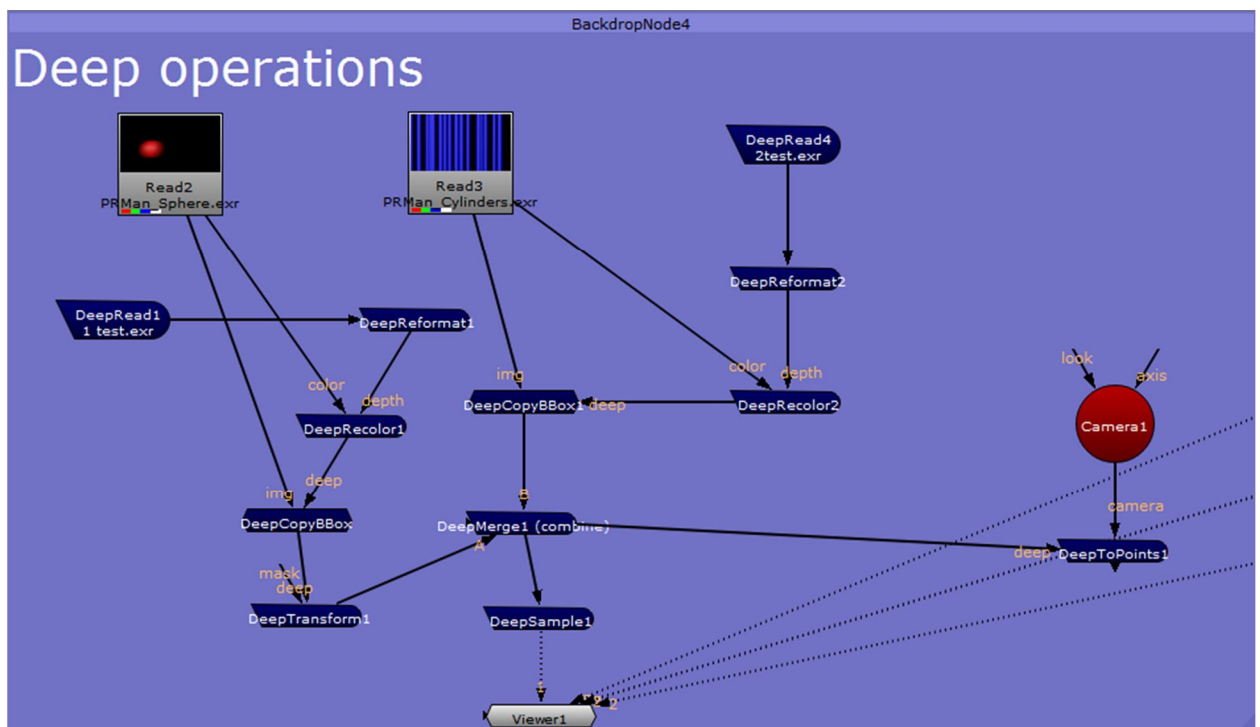


Рисунок 3.3 – Базова нодова схема

В результаті у вікні Viewer можна отримати зображення яке показано на рис.3.4. Аналізуючи отримане зображення з вмістом deer-data постає

питання про управління окремих пікселів, зокрема може цікавити в рамках дослідження як можна змінювати розташування цих елементів між собою та по відношенню до віртуальної камери зйомки. Інше питання може виникнути про те, як з отриманого зображення вилучити окремі крапки для аналізу в режимі просторового розміщення. Розглянемо ці особливості далі.

Нода `PositionToPoints` використовується для того, щоб перетворити зображення з даними про глибину у хмару точок у 3D вигляді. Цей вузол використовується для орієнтації за позицією. Підключимо отримане зображення з рисунку 3.3 до цієї ноди, щоб визначити її призначення в програмі (рис.3.5).

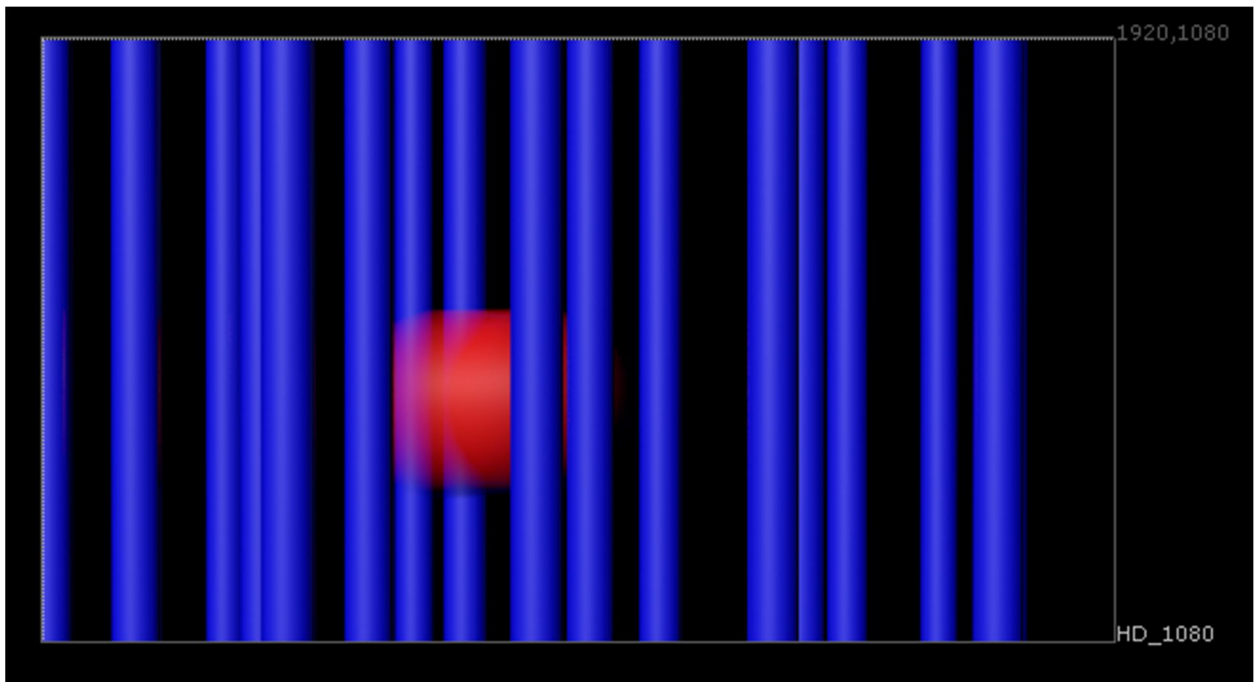


Рисунок 3.4 – Результат об'єднання

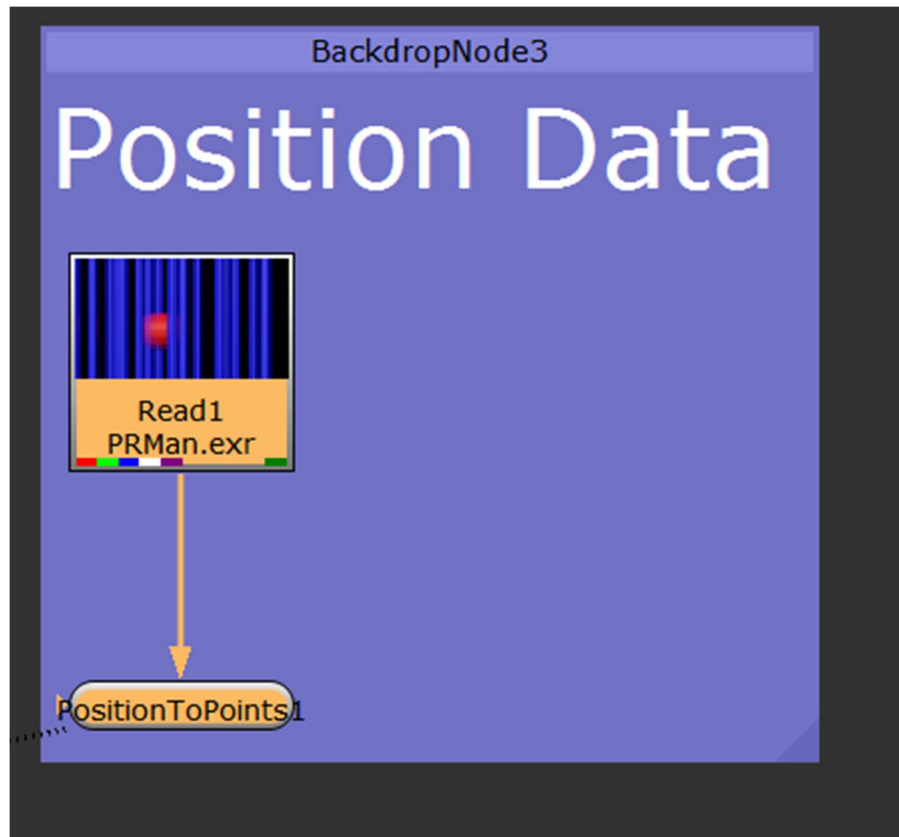


Рисунок 3.5 – Тестування ноди

В результаті отримуємо вихідне зображення (рис. 3.6) у вигляді хмари точок (рисунок 3.6) в режимі 3D програми Nuke.

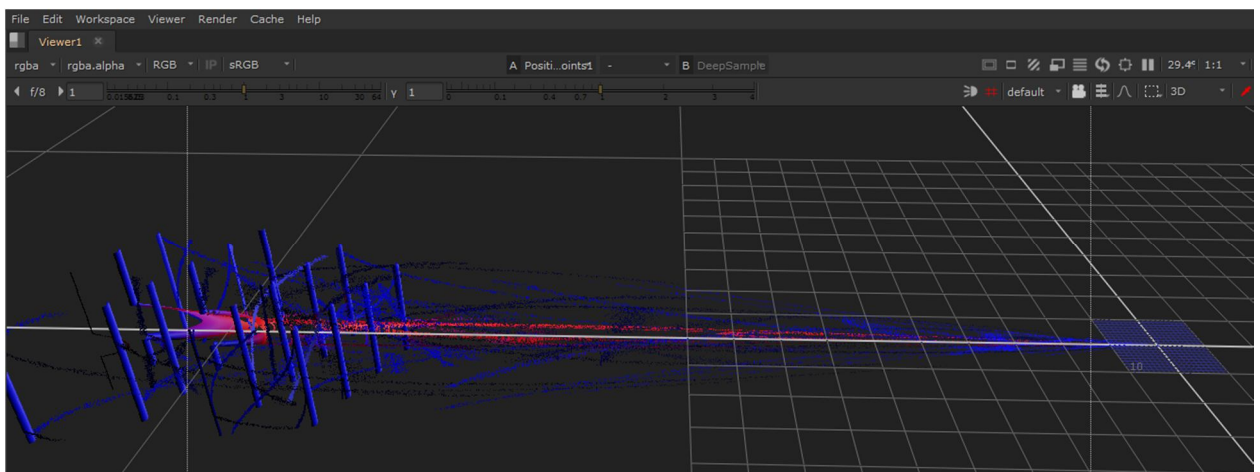


Рисунок 3.6 – Результат роботи схеми

Аналізуючи отриману хмару крапок можна візуально визначити як між собою розташовані окрему крапки зображення з рисунку 3.4. Тобто,

фактично ми можемо тепер описати глибину зображення в результаті об'єднання двох об'єктів.

Нода `DepthToPoints` в програмі Nuke є своєрідним генератором точок, тобто можна отримати набір точок з об'ємної поверхні-фігури. Підключимо зображення з рисунку 3.4 до цієї ноди (рис.3.7). Але для того, щоб отримати, результат попередньо слід налаштувати камеру. Оскільки дана нода має відповідний вхід. Крім цього, в налаштуваннях ноди `DepthToPoints` є відповідні поля “display” та “render” які пов'язані з підключеною камерою (рис.3.8).

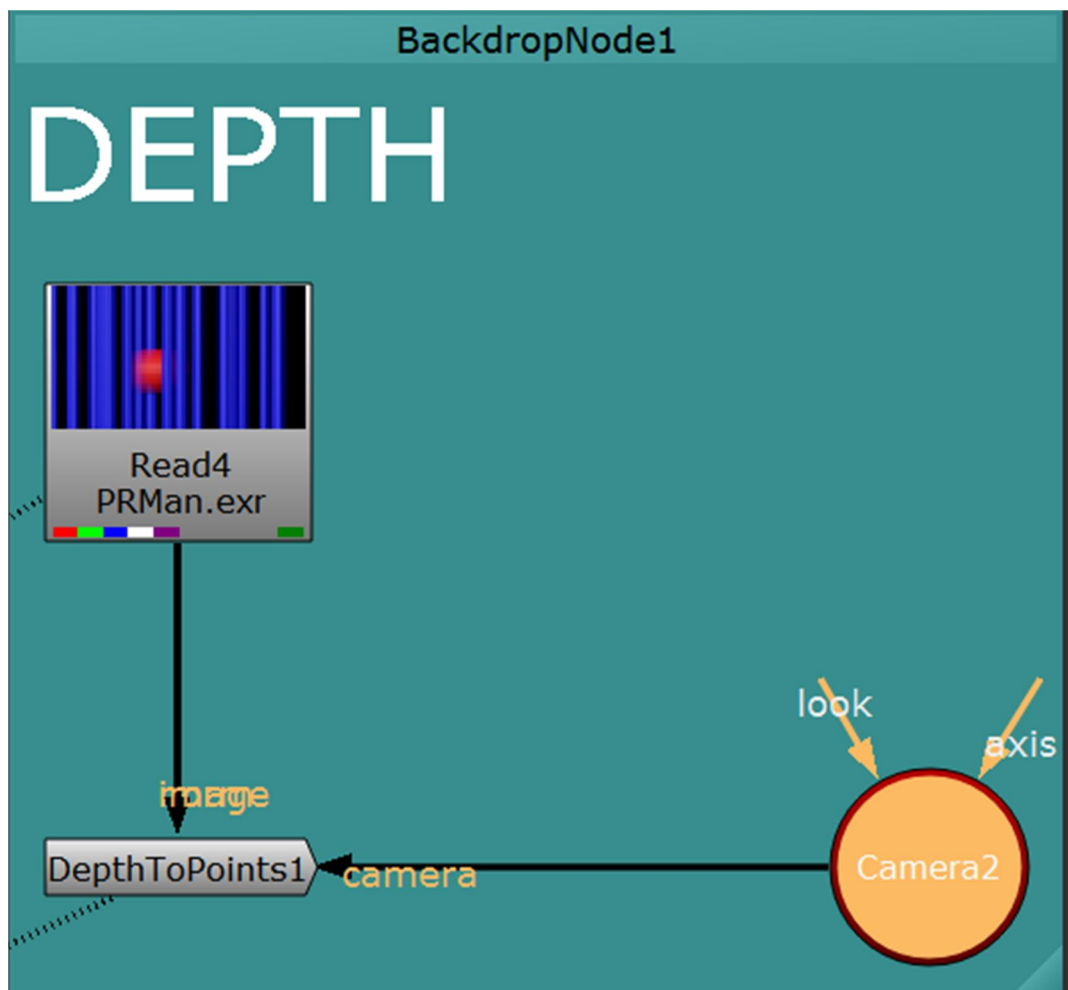


Рисунок 3.7 – Тестова схема



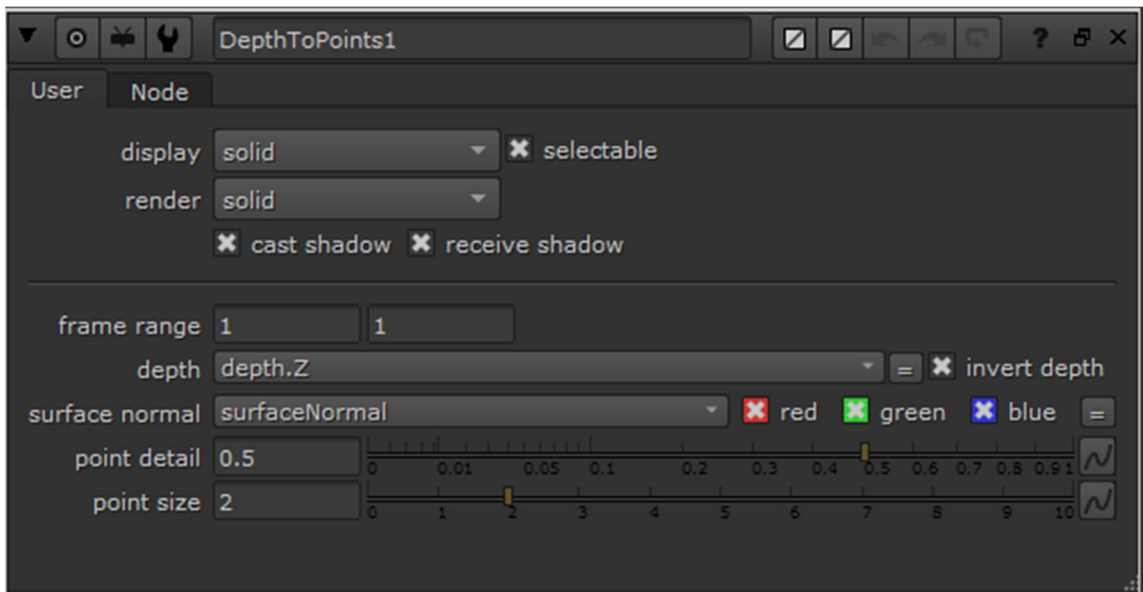


Рисунок 3.8 – Налаштування ноди

Поле display – це те що буде відображено у вікні Viewer програми Nuke, а поле render – це результат який буде отримано після рендерингу.

Типи відображення для хмари точок:

- off – приховування крапок;
- wireframe – відображення тільки контурів крапок;
- solid – відображення усіх крапок суцільними;
- solid + wireframe – відображення точок у суцільному вигляді з їх контурами;
- textured – відображення лише текстуру поверхні;
- textured+wireframe – відображає каркас і текстуру поверхні.

Інші параметри з налаштування ноди DepthToPoints визначають наступні функції:

Point detail – частота крапок, тобто їх кількість, і це налаштування відповідає за деталізацію зображення.

Point size – розмір крапок.

Таким чином, налаштувавши камеру, її фокус, апертуру, розміщення отримуємо в режимі 3D зображення, яке показано на рисунку 3.9. Результат безпосередньо з камери Cam1 показано на тому ж рисунку 3.9 знизу.

Аналізуючи отримане зображення можна в режимі 3D оцінити взаємне розташування об'єктів по відношенню до місця розташування камери.

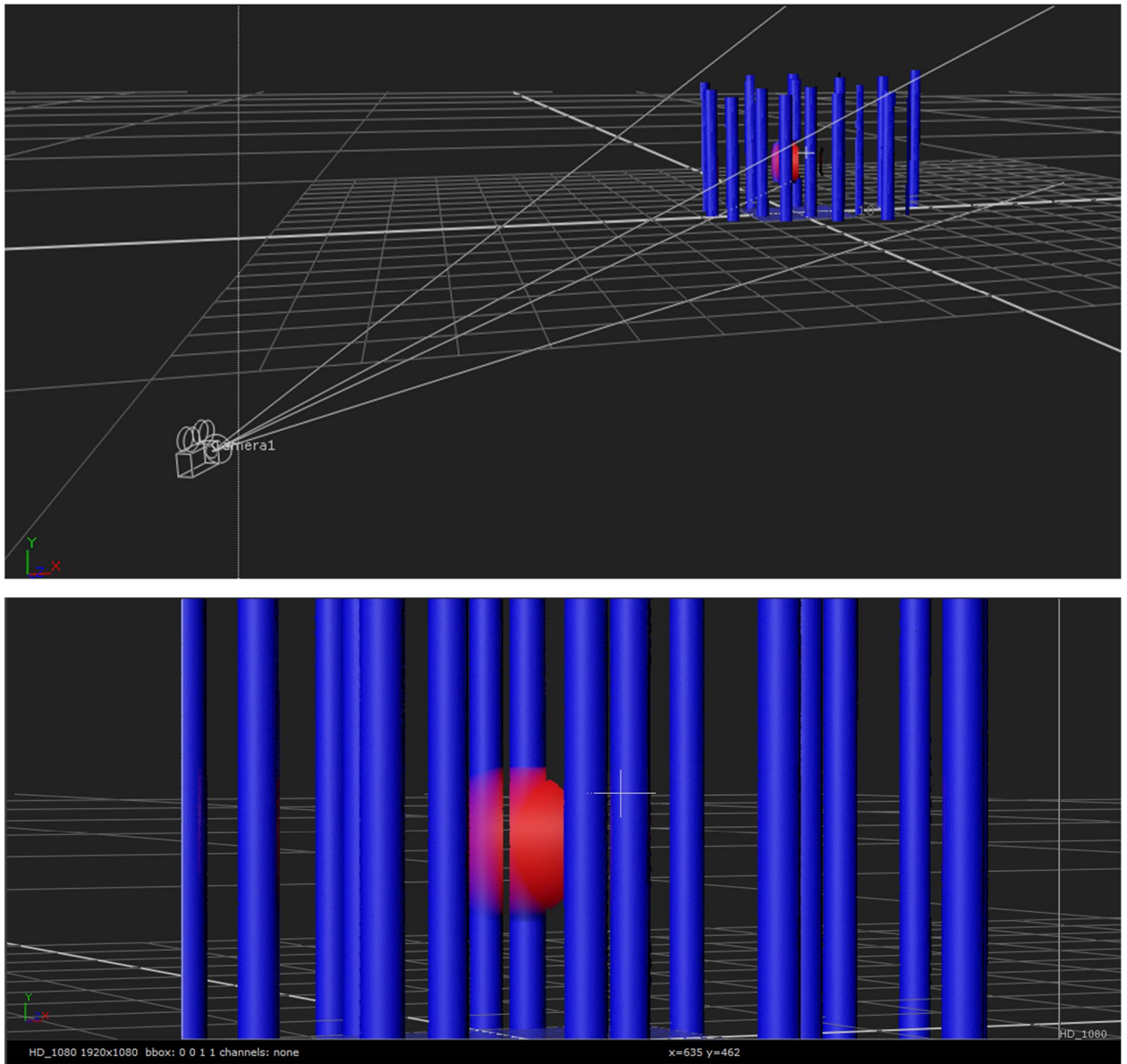


Рисунок 3.9 – Результат роботи ноди

Для створення файлу з інформацією про глибину (створення *deerp-data*) можна змоделювати нодову схему конвертування (конвертування плоских пікселів в *deerp-зображення*), яка показана на рисунку 3.10. Тобто, можна взяти 2D-зображення і для нього створити власний канал глибини.

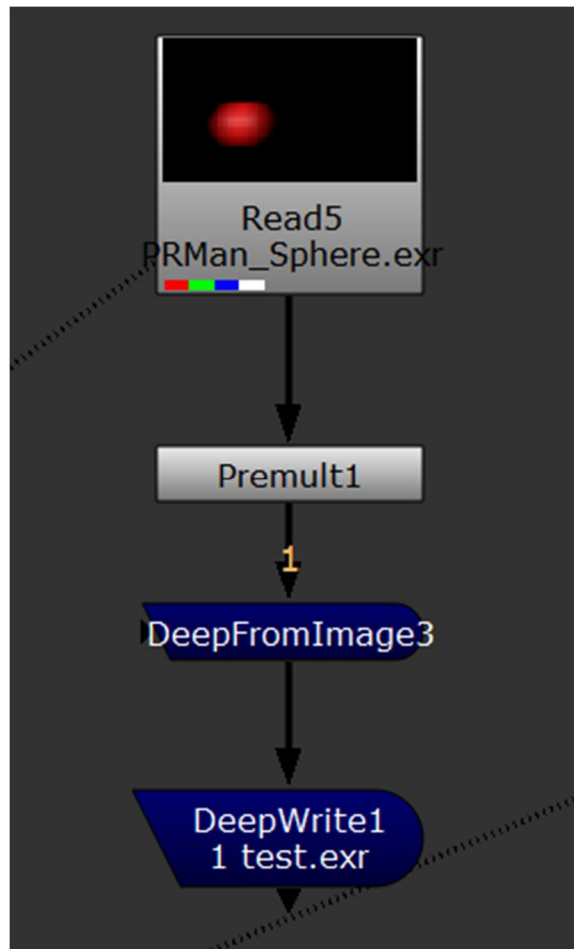


Рисунок 3.10 – Схема зі створення deep-data

В налаштуваннях ноди `DeepFromImage` існує параметр, *specify z* – це параметр визначення даних координати суто на основі depth-каналу, якщо він є в зображенні. Тобто, ноди та файли з deep-data не дозволяють підключати звичайні ноди, і треба інформацію конвертувати. Лише в цьому випадку можна буде використовувати розглянуті вище ноди з групи `Deep`.

### 3.2 Трансформація та об'єднання зображень з deep-data

В межах практичної частини дослідження розглянемо інші частини створеної схеми моделювання з рисунку 3.3. Для змінення положення шару в просторі використовується нода `DeepTransform`. За допомогою цієї ноди

можна перемістити шар у потрібне місце. Зазначимо, що в даному випадку програма аналізує саме інформацію, яка розміщена в каналі глибини. Як можна побачити з налаштувань цієї ноди, можна змінювати положення за координатами  $x$ ,  $y$ ,  $z$  (рис.3.11). Також існує налаштування *zscale*, яке дозволяє змінювати масштаб через глибину  $z$  всіх створених семплів у кожному пікселі. Значення вище 1 зменшує глибину, тоді як значення нижче 1 збільшує її. Також до ноди DeepTransform можна підключити маску.

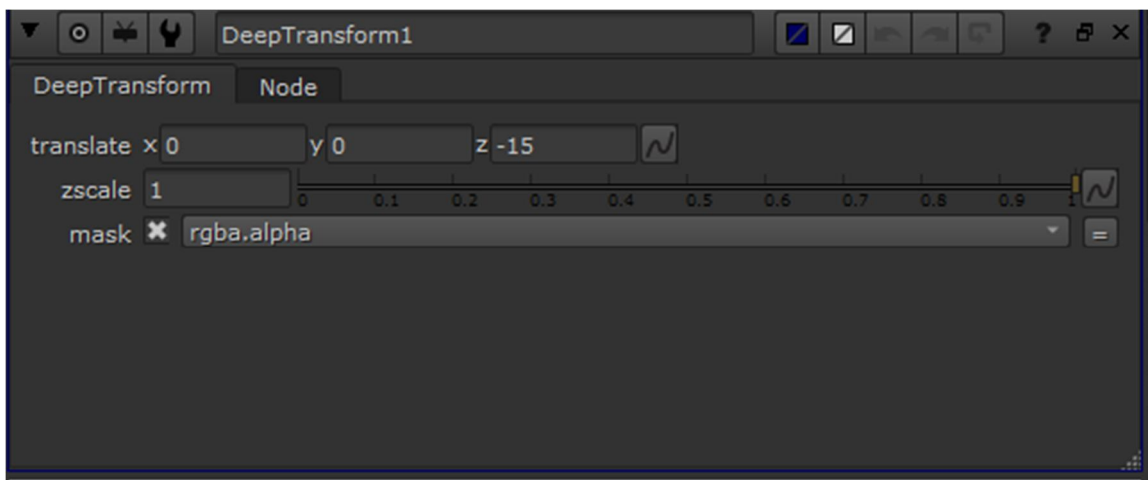


Рисунок 3.11 – Налаштування ноди

В рамках експерименту підключимо до входу *mask* ноду Roto і створимо маску (рис.3.12). В результаті отримуємо ефект, коли частина шару буде позаду колон, а інша попереду як показано на рисунку 3.12.

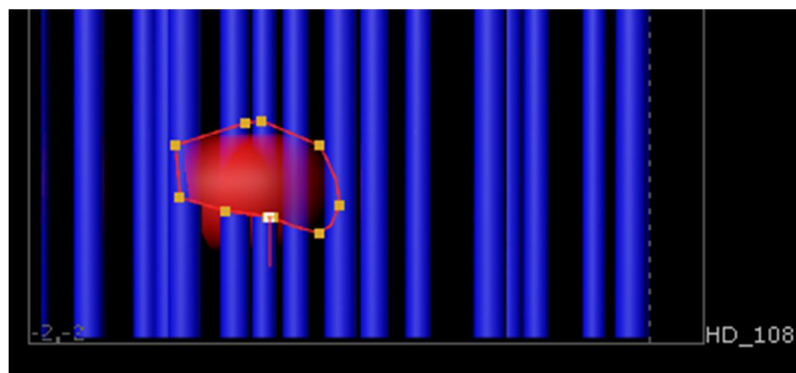


Рисунок 3.12 – Результат підключення маски

Для об'єднання двох зображень які містять *deep-data* використати ноду *DeepMerge*. Об'єднавши ці зображення маємо зображення, яке показано на рисунку 3.13.

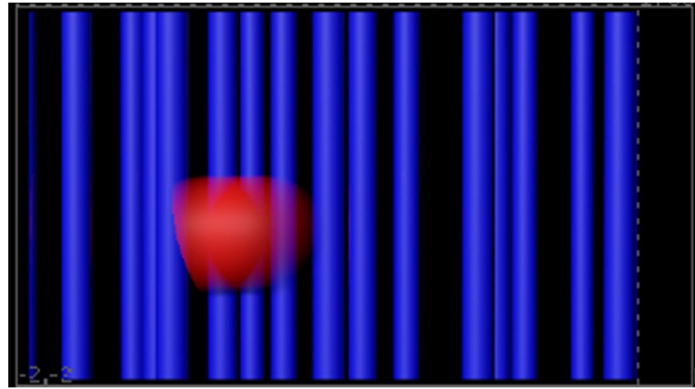


Рисунок 3.13 – Результат об'єднання зображень

Підключивши до виходу ноди *DeepMerge* ноду *DeepSample* є можливість отримати інформацію про кожен піксель. За допомогою вказівника *pos* є можливість обрати з якого саме пікселя буде отримана інформація. Наведемо рисунок, де вказано місце розташування цього вказівника (рис.3.14). А на рисунку 3.15 покажемо які в цій точці зображення створить семпли.

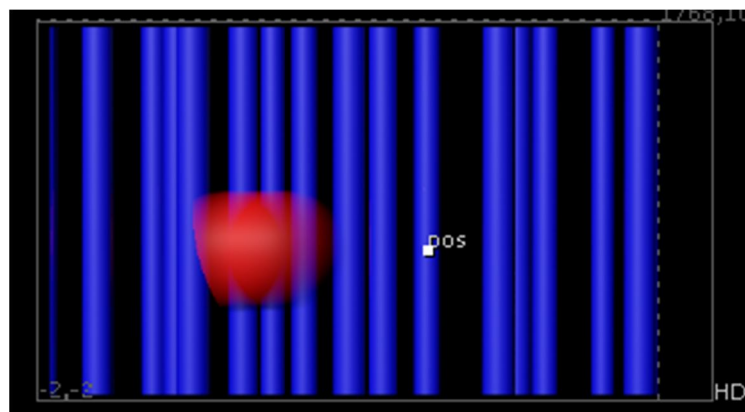


Рисунок 3.14 – Розташування вказівника *pos*

| deep.front | deep.back | rgba.red   | rgba.green | rgba.blue | rgba.alpha |
|------------|-----------|------------|------------|-----------|------------|
| 2.102      | 2.102     | 0.0231323  | 0.0231323  | 0.638672  | 1          |
| 2.102      | 2.102     | 0.0115662  | 0.0115662  | 0.319336  | 0.5        |
| 2.102      | 2.102     | 0.00770889 | 0.00770889 | 0.212839  | 0.333252   |
| 2.102      | 2.102     | 0.00578308 | 0.00578308 | 0.159668  | 0.25       |
| 2.102      | 2.102     | 0.00462534 | 0.00462534 | 0.127703  | 0.199951   |
| 2.102      | 2.102     | 0.00385445 | 0.00385445 | 0.106419  | 0.166626   |
| 2.102      | 2.102     | 0.00330381 | 0.00330381 | 0.0912166 | 0.142822   |
| 2.102      | 2.102     | 0.00289154 | 0.00289154 | 0.079834  | 0.125      |
| 2.102      | 2.102     | 0.00256963 | 0.00256963 | 0.0709462 | 0.111084   |
| 2.102      | 2.102     | 0.00231267 | 0.00231267 | 0.0638516 | 0.0999756  |

Рисунок 3.15 – Інформація про семпли в заданій точці

Аналізуючи дані отриманої таблиці з рис.3.15 можна дійти висновку, що в координаті розташування вказівника pos (рис.3.14) – (deer\_front;deer\_back) знаходиться інформація про кольорові компоненти 10 крапок. Тобто, фактично за рахунок інформацію про глибину зображення було створено своєрідний контейнер з набором deer-data.

### 3.3 Аналіз отриманих результатів моделювання

На основі наявних інструментів програми Nuke створена нодова схема (рис.3.3) з елементів, які дозволили виявити головні особливості роботи з deer-інформацією. Так, на основі підключення двох тестових об'єктів можна відмітити, що ноди дозволяють не лише сформувати контейнер з набору значень кольорових каналів кожного пікселя зображення, тобто створити набір семплів, але й візуально це простежити в режимі 3D. Серед розглянутих інструментів найбільш корисними з групи Deer-нод можна вважати ноди PositionToPoints (рис.3.5) та DepthToPoints (рис.3.7). Якщо для першої ноди, з'являється можливість відстежити розташування крапок в зображенні, де є інформація про глибину пікселів. То при використанні другої ноди, можна проаналізувати, як ці пікселі віддаленні в просторі від умовної камери зйомки. Крім цього, за отриманими результатами можна відмітити, що створення семплів є ключом до розширення можливостей зображення, як елементу обробки. Оскільки, в даному випадку,

використовуючи принцип об'єднання можна поєднати фактично растрову графіку (після процедури створення каналу depth) та векторну графіку. Зрозуміло, що без такого інструменту роботи з Деер в програмі Nuke, досі це зробити якісно і достовірно було неможливо використовуючи інші програми нелінійного монтажу та композитингу.

**Висновки до розділу 3.** В практичній частині дослідження була створена схема, яка дозволила на практиці перевірити можливості основних нод з групи Деер програмного пакету Nuke. Виявлено, що переважно ноди в цій групі корелюються за принципами роботи звичайних нод програми. Проте, в цьому випадку слід враховувати, що основною координатою для елементів зображення є саме координата  $z$ .

## ВИСНОВКИ

В дипломній роботі проведено аналіз основних програмних інструментів програмного пакету Nuke для роботи з deer-інформацією. Зокрема, визначено базові принципи класичного композитингу та deer-композитингу на основі правил об'єднання двох обраних тестових 2D-зображень відповідно.

В рамках дипломної роботи проведено практичне дослідження основних нод, які дозволяють не лише прочитати deer-інформацію, але й її проаналізувати, змінити (обробити) та записати на зовнішній носій інформації. На основі проведених досліджень отримано такі результати:

1. Визначено базові принципи та підходи, які лягли в основу програм з цифрового композитингу. Так знайдено, що найбільш розповсюдженою операцією при роботі з зображеннями є операція маскування. Відмічено, що алгоритми роботи з deer-інформацією, які відображено через відповідні ноди, є дуже схожими з класичними алгоритмами композитингу та алгоритмами, які викорисовуються при роботі зі стерео-зображеннями в спеціалізованому пакеті NukeX.

2. Сформульовано програмні передумови, які необхідно враховувати при роботі з z-координатою зображень. Додатково виділено базові програмні процедури з використання основних нод групи Deer програми Nuke.

3. На основі інструментів програми Nuke створена нодова схема з 15 елементів та двома підключеними зображеннями у форматі exr, які дозволили під час дослідження виявити головні особливості роботи з deer-інформацією. Так, на основі підключення двох тестових об'єктів можна відмітити, що ноди дозволяють не лише сформувати контейнер з набору значень кольорових каналів кожного пікселя зображення, тобто створити набір семплів, але й візуально це простежити в режимі 3D самої програми. Серед розглянутих інструментів найбільш корисними з групи Deer-нод



можна вважати ноди `PositionToPoints` (рис.3.5) та `DepthToPoints` (рис.3.7). Якщо для першої ноди, з'являється можливість відстежити розташування крапок в зображенні у просторі, де є інформація про глибину пікселів. То при використанні другої ноди, можна проаналізувати, і як ці пікселі віддаленні в просторі від умовної камери зйомки.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Основи цифрового композитингу в сучасному кіновиробництві : навч. посіб. для студ. спеціальності 171 «Електроніка», освітньої програми «Електронні системи мультимедіа та засоби Інтернету речей» / К. О. Трапезон, Г. Г. Власюк. – К. : ФОП Ямчинський О.В., 2019. – 330 с.
2. R. Gambar, Nuke 101. Professional Compositing and Visual Effects (1st Edition)., Berkeley: Peachpit Press, 2011, p. 404. ISBN-13: 978-0321733474.
3. R. Gambar, Nuke 101. Professional Compositing and Visual Effects (2nd Edition)., Berkeley: Peachpit Press, 2014, p. 380. ISBN-13: 978-0321984128.
4. B. Bratt, Rotoscoping. Techniques and Tools for the Aspiring artist., London: Elsevier, 2011, p. 283. ISBN: 978-0240817040.
5. S. Wright, Digital Compositing for Film and Video., Burlington: Focal Press, 2013, p. 475. ISBN-13: 978-0240813097.
6. Райт С. Цифровой композитинг в кино и видео / Райт Стив; пер. с англ. Медведникова М.М. – изд. 2-е, доп. и перераб. – М. : НТ Пресс, 2009. – 448 с. : ил. – (Секреты профессионалов). ISBN 978-5-477-00621-2.
7. P. Shirley, Fundamentals of Computer Graphics., Taylor Group: CRC Press, 2009, p. 785. ISBN-13: 978-1439865521.
8. Красильников Н. Н. Цифровая обработка 2D- и 3D-изображений / Красильников Н. Н. – СПб. : БХВ-Петербург, 2011. – 608 с. ISBN:978-5-9775-0700-4.

ДОДАТОК А  
SUMMARY

Digital compositing, as we will discuss it, deals with the process of integrating images from several sources into a single whole. Although many of these methods were originally designed for still images, tools and techniques that are also useful for large dynamic image sequences will be considered. Digital compositing is a digitally manipulated combination of at least two original images to obtain an integrated result. Undoubtedly, the most difficult part of this process of digital compositing is to obtain an integrated result-image, which does not show that its creation was associated with several source elements. In particular, we usually try to create sequences that could be plausibly photographed without the use of any post-processing. In colloquial language, they must look "real". Even if the elements in the scene are clearly unreal (for example, huge insects standing on a giant peach), you need to be able to believe that everything in the scene was photographed at the same time, by the same camera. Television commercials were most likely composed from numerous sources. However, whether the digital compositing process is used for business presentations or for special effects of a feature film, the methods and tools used follow the same basic principles. The true skill of digital compositing includes both technical and artistic skills. As with any art form, the artist definitely needs to have a good idea of the technique, the ability to handle the tools to be used. These tools can potentially include any number of different hardware and software composite systems. But it also requires an understanding of the whole compositing process, not just specific tools. This includes everything from understanding how this visual data is presented in digital format to gaining knowledge of how the camera responds to color and light. Combining several sources to get a new image is certainly not new, and it was done long before computers entered cinema. So let's look at the main points in the history of compositing. This is necessary to understand what the reasons were in digital cinema before the transition to deep-compositing.

Swedish photographer Oscar Raylander working in his studio in England selectively combined images from 32 different glass negatives to obtain a single massive palette. This is one of the earliest examples of what became known as the

"combo shot". Cinematic photography appeared in the late 19th century and over time, optical printers were built that could selectively combine several parts of the film, and thus optical compositing was born. A number of early examples of optical compositing can be found in the 1933 film *King Kong*. The image shown in, is actually a composite image, which was created as follows: first, a giant monkey was photographed, a 16-inch miniature, which is animated using the technique of stop motion.

This process involves photographing the model frame by frame at a time, changing the pose or position of the character between each frame. As a result, when playing at normal speed, a continuously animated object is obtained. Once these frames were designed and processed, they were projected onto a large rear projection screen that was located on a full-size stage. Then, the action in the foreground (actress on a tree) was photographed, and the background shots were projected behind it, creating a composite image. This particular type of compositing is known as the effect in the camera, as there was no additional post-production work required to complete the shooting. Other scenes in the film were made using an early form of blue screen photography, where the whole place and background were photographed separately and then combined at the optical printing stage.

Currently, optical compositing has actually been replaced by general-purpose computer systems and some software, but the concepts themselves have not changed. And this was actually the first attempt to move from creating depth to images when combined from different sources.

The relative positions of objects in a scene can be encoded by the color of those objects. That is, for example, we agree that two objects that are painted in different shades of the same color are geometrically located at different distances from the camera.

It is known that one of the most important things a composer does when integrating elements into a scene is to balance everything in terms of color, brightness and contrast. The ability of different people to judge color can vary

greatly, as a rule, a person's ability to correctly recognize colors can be gained through extensive practical experience. The most important fact is that the perception of these things can be significantly influenced by external factors.

Although the inner circle has the same color in all three cases, it seems brighter when surrounded by a darker background. Scientists who study perception know this phenomenon as the "principle of simultaneous contrast." The same applies to color images. The color will appear brighter and more saturated when it is surrounded by additional colors - such as red, surrounded by blue or green - and in general any color perception will depend on the surrounding colors. These points can be used to create the depth of the image.

Because the human visual system is susceptible to environmental influences, everyone does their best to try to control the environment every time they judge color. You should try to look at different images in the most similar context. When comparing two images on a computer monitor, you need to make sure that the environment, both on the screen and outside, does not distort the judgment of color. Also, there are a number of digital tools for selecting the color of a particular pixel or group of pixels. It is important to understand that it is often impossible to fully imagine what an image will look like on a particular device if it is viewed on another device.

Images that contain information about the depth of objects through deep-channels, depth created in their structure in film and computer graphics are usually used to create effects and generate complex scenes. For example, let's analyze a frame from the movie "Planet of the Apes", in which many objects overlap. With deep compositing technology, you can change the placement of objects and characters without having to render each time.

But there are also alternative methods of using this technology. Devices and technologies that reproduce and record depth in the real world have become more popular recently. Caesar Palomo and Marcelo Gattas in 2010 proposed a situation in which the event was filmed from several angles of the camera, but the angle that must be shown exists between the two angles. Their research proposed an

algorithm that was processed on a graphics processor to create a new image, where the reference views were to the left and right of the desired view, given that all reference views used also had "depth maps" characteristic of them. Palomo and Gattas defined a "depth image" as a color image with one depth value per pixel. Depth maps obtained using new deep sounding technologies were linearized into halftones. The method developed by the authors was based on three-dimensional deformation, in which, moving from the coordinate system of the camera to the coordinate system of the world, the 3D grid is projected from the camera, and then deformed into a new angle. This technique in itself usually fails because the areas that have been closed become open and have no RGBA information for them. To "mix" two images with each other, each pixel of both reference images is compared with each other. Given the limit value, if one pixel is closer to the new camera than the other (according to their depth map information), this pixel is stacked on top of the background pixel.

Another method of using "deep" images was designed to simulate the depth of field, the overall effect inherent in real-world cameras. Depth of field is based on the idea that there is one plane or slice completely in focus, and the farther from this slice you get to the Z-depth, the more blurred the pixel. In 2004, Budin published an article on the algorithm for using Z-depth to create artificial depth of field in digital images. The first step was that each pixel was evaluated and, depending on how far from a given Z-depth it is, Gaussian blurring of different strengths centered on the pixel was applied. The second step was that the RGBA value of each pixel was normalized depending on how far from the standard value of "1" they were. This was done after all Gaussian blurs had been applied. To implement his method, Budin developed a plugin for the composition program Shake. Its plug-in also had a pre-visualization feature that darkened the images depending on whether they were inside the focusing planes. One of the problems with his method was that the lack of background rendering or the lack of a 3D background caused problems with hard edges. Depth of field is a feature common to many cameras and camcorders, but that doesn't mean that there are no methods

for creating depth of field on non-digital images after they've been taken. Jacobs, Beck, and Left (2012) have developed a technique that allows a user to create their own focus range when there are multiple photos at different depths of field or focal ranges. Necessary maps tell the software how this part of the image is "in focus". It is worth noting that this is not exactly a depth map, as the strongest value in it may be in the center of the image. However, in the end they were able to create images in which all the subjects were in focus without losing quality.

The technique of deep-compositing is inextricably linked with the operations of individual pixels of images. In this case, the information is laid down as one pixel in a certain place of one image is located relative to the pixel of another image, which is located in the same coordinates. The principles of deep-compositing are the following procedures:

- import images with depth data or create a depth channel in a flat image;
- Combine these images with depth in one frame.
- color correction;
- conversion, reformatting, cropping images;
- image masking;
- deep image creation.

Let's look at some of the simplest procedures that involve deep image information or image sequences. As a basis of consideration we will take nodes which are a part of group Deep of the Nuke program.

Nuke allows you to read images with depth data in two formats, DTEX and OpenEXR. DTEX is an image format created in Pixar's PhotoRealistic RenderMan program. OpenEXR format is a data format that also supports depth data values. Since OpenEXR is a much more common format, it will be described how to import it. The algorithm of the procedure is as follows:

1. You need to create a DeepRead node, and to do this, click Deep (join the group) and then select the DeepRead node from the list.
2. Find the desired file in .exr format and click Open.



To do this, use the DeepFromFrames node. DeepFromFrames creates samples with depth data from the input sequence. The algorithm is as follows:

1. Attach DeepFromFrames to the input image sequence. An image with depth information is created by placing each frame at a magnifying depth.
2. Use the settings panel to adjust the result. Figure 2.2 shows an example of the procedure.

The operation of merging two deep-images is based on the DeepMerge node. Thus, each output pixel contains all samples from the same pixel of each input image. To merge, follow these steps: 1. Connect the input images to the DeepMerge node as shown in Figure 7.2 2. In the DeepMerge node settings, set the combine operation (Fig. 2.3). It is also possible to check the box, discard hidden samples, on the properties panel, so as not to include samples that are completely closed by samples closer to the camera, the alpha values of which are equal to one.

DeepColorCorrect node allows you to adjust the color of each pixel in each sample. There are control kits for adjusting shadows, halftones and highlights, and there is a function to adjust all this at once. It is possible to use the curves of the Ranges tab to control the range of the image, which should be in shadows, halftones and glare.