

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

**Журнал практики**

Институт № 8      «Компьютерные науки и прикладная математика»

Кафедра                      805                                      Учебная группа                      М8О-305Б-20

ФИО обучающегося                      Манташев Асадулла Уллубиевич

Направление подготовки/      01.03.04  
специальность

*шифр, наименование направления подготовки/специальности*

Вид практики                                      Научно-исследовательская практика  
*учебная, производственная, преддипломная или другой вид практики*

Оценка за практику                      \_\_\_\_\_ Волкова Т.Б.

Москва  
2023

**1. Место и сроки проведения практики:**

Наименование организации: МАИ, кафедра 805

Сроки проведения практики

дата начала практики: 09.02.2023

дата окончания практики: 05.06.2023

**2. Инструктаж по технике безопасности:**

\_\_\_\_\_/\_\_\_\_\_/09.02.2023г.  
подпись проводившего                      расшифровка подписи                      дата проведения

**3. Индивидуальное задание обучающегося:**

Анимация изображений с помощью эйлеровых полей движения

**4. План выполнения индивидуального задания обучающегося:**

№ п/ п	Место проведения	Тема	Период выполнения
1		Ознакомление с индивидуальным заданием.	
2		Поиск материалов, необходимых для решения задачи.	
3		Реализация решения.	
4		Оформление отчета. Подведение итогов.	

**Утверждаю**

\_\_\_\_\_  
подпись руководителя от МАИ

Волкова Т.Б./  
расшифровка подписи

9 февраля 2023\_г.  
дата утверждения\*

\_\_\_\_\_  
подпись руководителя от  
организации/предприятия

Пановский В.Н./  
расшифровка подписи

9 февраля 2023\_г.  
дата утверждения\*

**Ознакомлен**

\_\_\_\_\_  
подпись обучающегося

\_\_\_\_\_  
расшифровка подписи

9 февраля 2023\_г.  
дата ознакомления\*

\*Дата утверждения и ознакомления – дата начала практики

Все поставленные задачи выполнены

организации/предприятия

расшифровка подписи

data

## 6. Отчет обучающегося по практике:

В ходе выполнения работы была решена задача анимации изображения методом эйлеровых полей движения.

Для написания нейронной сети для распознавания изображений, можно использовать библиотеки для глубокого обучения, такие как TensorFlow, PyTorch.

Для решения поставленной задачи можно также использовать библиотеки OpenCV, matplotlib. OpenCV - это библиотека компьютерного зрения, которая также может использоваться для анимирования изображений. Она предоставляет широкий спектр функций для обработки изображений, включая возможность создания анимаций. Matplotlib - это библиотека для создания визуализаций на языке Python. Она также может быть использована для создания анимаций.

Было проведено сравнение некоторых методов анимирования изображений.

### Метод 1: Изменение цвета пикселей.

Метод изменения цвета пикселя в анимировании изображения может быть реализован путем применения различных фильтров или изменения значений цветовых компонент пикселей. Вот пример простого метода изменения цвета пикселя в анимации на языке Python:

```
import cv2
import numpy as np

# Загрузка исходного изображения
image = cv2.imread('image.jpg')

# Создание пустой последовательности кадров
colorized_frames = []

# Целевой цвет пикселя
target_color = (255, 0, 0) # Красный цвет

# Изменение цвета пикселя в анимации
for i in range(num_frames):
    # Создание копии изображения
    frame = image.copy()

    # Изменение цвета пикселей на целевой цвет
```

```

        frame[np.all(frame == target_color, axis=-1)] = (0,
255, 0) # Зеленый цвет

```

```

    colored_frames.append(frame)

```

```

# Сохранение анимации
for i, frame in enumerate(colored_frames):
    cv2.imwrite(f"frame_{i}.jpg", frame)

```

В этом примере мы создаем последовательность кадров, где каждый кадр представляет собой изображение, в котором цвет пикселей изменен на целевой цвет (в данном случае, смена красного цвета на зеленый). Используется функция `np.all()` для проверки, совпадает ли цвет пикселя с целевым цветом, и применяется изменение цвета, если это условие выполняется. Затем сохраняется каждый кадр в отдельном файле изображения.

## Метод 2: Изменение яркости.

Метод изменения яркости в анимировании изображения может быть реализован путем наложения яркостного фильтра на каждый кадр анимации. Вот пример простого метода изменения яркости в анимации на языке Python:

```

import cv2
import numpy as np

# Загрузка исходного изображения
image = cv2.imread('image.jpg')

# Создание пустой последовательности кадров
brightness_adjusted_frames = []

# Изменение яркости пикселей в анимации
for i in range(num_frames):
    # Создание копии изображения
    frame = image.copy()

    # Изменение яркости пикселей в кадре
    alpha = 1.5 # Коэффициент яркости (1.0 - без изменений)
    beta = 50    # Коэффициент контрастности (0 - без
изменений)

```

```
adjusted_frame = cv2.convertScaleAbs(frame,  
alpha=alpha, beta=beta)
```

```
brightness_adjusted_frames.append(adjusted_frame)
```

```
# Сохранение анимации
```

```
for i, frame in enumerate(brightness_adjusted_frames):  
    cv2.imwrite(f"frame_{i}.jpg", frame)
```

В данном примере создается последовательность кадров, где каждый кадр представляет собой изображение с измененной яркостью. Используется функция `cv2.convertScaleAbs()` для изменения яркости и контрастности каждого пикселя в кадре. Параметры `alpha` и `beta` задают коэффициенты яркости и контрастности, соответственно. Затем сохраняется каждый кадр в отдельном файле изображения.

### **Метод 3: Метод морфинга.**

Метод морфинга в анимировании изображения позволяет создавать плавные переходы между двумя изображениями, создавая впечатление метаморфозы или плавного перемещения объектов. Вот пример простого метода морфинга в анимации на языке Python:

```
import cv2
```

```
import numpy as np
```

```
# Загрузка исходных изображений
```

```
image1 = cv2.imread('image1.jpg')
```

```
image2 = cv2.imread('image2.jpg')
```

```
# Создание последовательности промежуточных кадров
```

```
morphed_frames = []
```

```
# Количество промежуточных кадров
```

```
num_frames = 10
```

```
# Морфинг изображений
```

```
for i in range(num_frames + 1):
```

```
    # Вычисление весового коэффициента
```

```
    alpha = i / num_frames
```

```
    # Создание промежуточного кадра
```

```
    morphed_frame = cv2.addWeighted(image1, 1 - alpha, image2,  
alpha, 0)
```

```

morphed_frames.append(morphed_frame)

# Сохранение анимации
for i, frame in enumerate(morphed_frames):
    cv2.imwrite(f"frame_{i}.jpg", frame)

```

В данном случае создается последовательность промежуточных кадров, где каждый кадр представляет собой смешение двух исходных изображений. Используется функция `cv2.addWeighted()` для создания промежуточного кадра с помощью весового коэффициента `alpha`. Значение `alpha` изменяется от 0 до 1, что позволяет создать плавный переход между изображениями. Затем сохраняется каждый кадр в отдельном файле изображения.

#### **Метод 4: Метод эйлеровых полей движения.**

```

# Создание пустой последовательности кадров
eulerian_frames = []

# Создание пустого поля движения
motion_field = np.zeros_like(image, dtype=np.float32)

# Шаг перемещения
step = 10

# Количество кадров анимации
num_frames = 30

# Реализация эйлеровых полей движения
for i in range(num_frames):
    # Создание копии изображения
    frame = image.copy()

    # Накопление поля движения
    motion_field += step

    # Применение поля движения для смещения пикселей
    warped_image = cv2.remap(image, motion_field[:, :, 0],
motion_field[:, :, 1], cv2.INTER_LINEAR)

    # Отрисовка смещенного изображения
    frame = cv2.putText(warped_image, 'Moving Image', (100,
100), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)

```

```
eulerian_frames.append(frame)

# Сохранение анимации
for i, frame in enumerate(eulerian_frames):
    cv2.imwrite(f"frame_{i}.jpg", frame)
```

Подробнее о задаче анимирования изображения методом эйлеровых полей движения:

Задача **анимирования изображения** состоит в создании эффекта движения или изменения визуальных элементов на изображении, чтобы оживить его и сделать более привлекательным для зрителя. Анимации могут быть простыми, как изменение позиции или размера объекта, или более сложными, включая применение специальных эффектов, изменение цветовой гаммы или создание иллюзии трехмерности.

**Эйлеровы поля движения** являются математическим методом моделирования движения частиц в поле, вдохновленных работами швейцарского математика Леонарда Эйлера. Они основаны на векторном поле, которое определяет направление и скорость движения частиц. При использовании эйлеровых полей движения можно создать реалистичные и органичные анимации, имитирующие различные естественные явления, такие как волны, дым, огонь, водопады, движение травы и многое другое.

Для анимации изображения объекта с использованием эйлеровых движений необходимо определить начальную ориентацию объекта и задать параметры его движения. Для этого можно использовать углы поворота вокруг каждой из осей координат.

Далее, с помощью математических операций, можно вычислить новую ориентацию объекта в каждый момент времени, исходя из его начальной ориентации и параметров движения. Новая ориентация объекта может быть использована для отображения в каждом кадре анимации.

Метод эйлеровых движений позволяет моделировать разнообразные типы движений, такие как вращение вокруг оси, перекачивание, наклон, вращение



вокруг произвольной оси и др. Он является эффективным и универсальным способом создания реалистичной анимации объектов в трехмерном пространстве, так как учет ориентации объекта позволяет достичь более естественного и плавного движения.

**Преимущества** использования эйлеровых полей движения в анимации изображений включают:

- **Реалистичность:** Эйлеровы поля движения позволяют создавать анимацию, которая выглядит естественно и реалистично, что делает ее более привлекательной для зрителя.
- **Гибкость:** С использованием эйлеровых полей движения можно создавать разнообразные эффекты движения, которые могут быть адаптированы под различные цели и требования проекта.
- **Эффективность:** Эйлеровы поля движения могут быть реализованы с помощью сравнительно небольшого количества вычислительных ресурсов, что делает их эффективными и доступными для использования в различных приложениях и средах.
- **Интерактивность:** Эйлеровы поля движения позволяют создавать анимацию, которая может реагировать на внешние воздействия или действия пользователя, что делает ее более интерактивной и цепляющей для зрителя.

**Процесс создания эйлеровых полей** движений обычно включает несколько шагов:

- **Определение системы дифференциальных уравнений:** Исходя из физической системы или задачи, необходимо сформулировать соответствующую систему дифференциальных уравнений Эйлера-Лагранжа. Уравнения могут быть линейными или нелинейными, и зависят от конкретной задачи.
- **Нахождение интеграла движения:** Интеграл движения - это функция на многообразии, которая сохраняется вдоль траекторий системы дифференциальных уравнений. Нахождение интеграла движения может быть сделано методами аналитики или численными методами.

- Построение эйлерова поля движений: После нахождения интеграла движения, эйлерово поле движений может быть построено путем определения векторного поля, где каждому вектору в каждой точке многообразия соответствует направление движения.
- Анализ и интерпретация результатов: Полученные эйлеровы поля движений могут быть проанализированы и интерпретированы с помощью графиков, изображений или других методов визуализации. Они позволяют понять динамику системы и предсказать ее будущее движение.

Анимация изображений методом эйлеровых полей движения представляет собой способ создания движущихся изображений путем применения математических алгоритмов к статическим изображениям. Эйлеровы поля движения основаны на принципе векторного поля или эйлеровых уравнений, которые описывают движение каждой точки внутри изображения.

По результатам исследования можно заключить, что выбор метода анимирования изображений зависит от конкретной задачи и требуемого эффекта.

Также было выяснено, что анимирование изображений с использованием эйлеровых полей движения является эффективным и гибким методом создания реалистичных и качественных анимаций. Применение этого подхода может быть полезно в таких областях, как компьютерное видео, игровая индустрия, визуализация данных и другие.

Таким образом, был изучен вопрос анимации изображений методом эйлеровых полей движения, его цель, задачи, а также прочие методы позволяющие реализовать анимацию, такие как: метод морфинга, метод изменения цвета пикселя, метод изменения яркости, метод эйлеровых полей перемещения. Были получены навыки разработки приложений, полезные для дальнейших исследований и разработок в области анимации и компьютерной графики.