Захист <https://youtu.be/Y-1Ny7ZIxgM>

**ЛАБОРАТОРНА РОБОТА №4**

**ТЕМА:** Сегментація об’єктів на відеозображеннях

**МЕТА:** Засвоїти принципи створення процедур, призначених для маніпуляції об‘єктами на відеозображеннях.

**Короткі теоретичні відомості**

На цей час актуальною залишається тема автоматизованої сегментації, комбінуванні, переносу та трансформації зорових об‘єктів на відеозображеннях, Такі процедури широко застосовуються у кінематографі, у сфері розваг, а також у новітніх методах відстежування, які застосовуються, службами безпеки. Дана розрахунково-графічна робота присвячена здійсненню автоматизованого синтезу штучного зображення, який здійснюється шляхом переносу певного зорового об‘єкта на інший фон.

Як відомо, відеофайл, або відеопоток складається з окремих кадрів. Тож, щоб перемістити певний зоровий об‘єкт з однієї «сцени» на другу необхідно по-перше розділити обидва початкові відеофрагмента на кадри. Потім потрібно відокремити об‘єкт, що підлягає переносу, створивши для нього маску. Далі потрібно перенести об’єкт на новий фон, використовуючи процедури «згладжування» країв. І нарешті змонтувати послідовність зображень у відеофайл.

Зазвичай завданням сегментації є поділ зображення на області, що складаються з точок, що мають приблизно однаковий рівень яскравості (в разі напівтонових зображень) або схожі колірні характеристики (в разі кольорових зображень). Відповідно, контурами, або межами, даних областей є сукупність точок на зображенні, в яких відбувається зміна рівня яскравості або кольору. Для визначення меж або контурів областей розроблено багато різних методів, основою для яких, в більшості випадків, є побудова градієнтного зображення (рис 4.1), тобто, застосування до вихідного зображення оператора першої похідної для дискретної функції, визначеної на площині. Оператор градієнта можна розглядати як лінійний фільтр зображення. Лінійні фільтри зображень, як правило, задаються квадратною матрицею коефіцієнтів або маскою фільтра, яка застосовується для кожної точки зображення. Розмір маски оператора першої похідної, або, іншими словами, його масштаб, може бути різним. Часто відповідний масштаб оператора градієнта, який дає в результаті найкращу картину контурів об'єктів, підбирається шляхом експериментальних перевірок. Але зазвичай зображення, що зустрічаються на практиці, в більшості випадків містять контури з різними швидкостями зміни яскравості (для випадку напівтонових зображень) або кольору (для випадку кольорових зображень), тобто, як різкі, так і плавні. Отже, неможливо найкращим чином однозначно визначити всі існуючі на зображенні кордони, використовуючи лише оператор градієнта одного певного масштабу. Тож у більш складних випадках застосовують методи, що дозволяють побудувати картину контурів об'єктів зображення на основі інформації, одержуваної в результаті застосування оператора градієнта різних масштабів.

**Хід роботи**

1. Створіть за допомогою веб-камери, або іншим шляхом 2 відеофрагмента, один з яких повинен містити рухомий об ‘єкт(або об‘єкти) який ви будите переносити на новий фон, а інший – майбутьній фон. Один з цих відеофрагментів повинен містити ваше рухоме обличча.

2. Розбити ваш avi-відеофайл на посліфжовнысть кадрів bmp-формату 640х480 ( не меньш ніж 100 кадрів за допомогою VirtualDub).

3. При необхідності за допомогою градіетного методу та інших характерних ознак виділти об‘єкти та іх контури на кадрі. При необхідності застосувати функцію замикання контурівоб‘єктів.

5. Створити бинарну маску для усіх замкнених об‘єктів.

6. При необхідності видалити усі непотрібні об‘єкти меньшої пощи.

7. Перенести обраний об‘єкт шляхом його переносу на новий фон

8. Застосувати функцію зглажування меж обраного об‘єкту на новому фоні.

9. Застосувати послідовність перевворень (п.з 2 по 8) до усієї послідовності кадрів.

10. За допомогою VirtualDub нову послідовності кадрів змонтувати avi-відеофайл.

Було розроблено программу на мові програмування python. Програма захоплює відео з камери в режимі реального часу, де виділяє для себе контури об’єкта і фон за об’єктом перетворює на картинку.

В лістингу 1 зображено код програми.

Лістинг 1

import numpy as np

import mediapipe as mp

import os

import cv2

height = 1920

width = 1080

mp\_selfie\_segmentation = mp.solutions.selfie\_segmentation

selfie\_segmentation = mp\_selfie\_segmentation.SelfieSegmentation(

    model\_selection=1)

image\_path = 'images'

images = os.listdir(image\_path)

image\_index = 0

bg\_image = cv2.imread(image\_path+'/'+images[image\_index])

cap = cv2.VideoCapture(0)

while cap.isOpened():

    \_, frame = cap.read()

    frame = cv2.flip(frame, 1)

    height, width, channel = frame.shape

    RGB = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

    results = selfie\_segmentation.process(RGB)

    mask = results.segmentation\_mask

    condition = np.stack(

        (results.segmentation\_mask,) \* 3, axis=-1) > 0.6

    bg\_image = cv2.resize(bg\_image, (width, height))

    output\_image = np.where(condition, frame, bg\_image)

    cv2.imshow("Output", output\_image)

    cv2.imshow("Frame", frame)

    key = cv2.waitKey(1)

    if key == ord('q'):

        break

    elif key == ord('w'):

        if image\_index != len(images)-1:

            image\_index += 1

        else:

            image\_index = 0

        bg\_image = cv2.imread(image\_path+'/'+images[image\_index])

cap.release()

cv2.destroyAllWindows()

На рисунку 1 зображено результат роботи програми.



**Висновок:** засвоїв принципи створення процеду, призначених для маніпуляції об‘єктами на відеозображеннях.