KAZE DESKRIPTORLARI YORDAMIDA YUZNI TANIB OLISH DASTURIY TA'MINOTINI ISHLAB CHIQISH

Mahkamov Azizbek Shuhratbek o'g'li
Andijon davlat chet tillari instituti Ijtimoiy gumanitar fanlar, pedagogika va psixologiya kafedrasi o'qituvchisi.
darsalgoritm@gmail.com
Andijon davlat chet tillari instituti
Filologiya va tillarni o'qitish rus tili yo'nalishi 1-kurs talabasi
Ortiqaliyeva Ruxshonaxon Abdupatto qizi
Andijon davlat chet tillari instituti
Filologiya va tillarni o'qitish rus tili yo'nalishi 1-kurs talabasi
Mamadjanova Mubina Axmadjonovna

Annotatsiya.Bu maqolada KAZE deskriptorlari muhimligi. KAZE deskriptoridan foydalanib python 3.9.6 opencv-python 3.4.11.39 vositasida dastur tuzish ko'rib o'tilgan.

Kalit so'zlar: BRIEF, KAZE, ORB, BRISK, SIFT, deskriptor, mahalliy deskriptorlar,

Abstract: The importance of KAZE descriptors in this article. Python 3.9.6 opencv-python 3.4.11.39 program creation using the KAZE descriptor is considered.

Keywords: BRIEF, ORB, BRISK, KAZE descriptor, local descriptors.

Аннотация:Важность дескрипторов KAZE в этой статье. Рассмотрено создание программы на Python 3.9.6 opencv-python 3.4.11.39 с использованием дескриптора KAZE.

Ключевые слова:BRIEF, ORB, BRISK, KAZE, SIFT, дескриптор, локальные дескрипторы

Deskriptor - bu xususiyatlar to'plamiga asoslangan holda tasvirning ba'zi bir mintaqasini aniqlaydigan usul. Ko'pincha tasvirlardagi obyektlarni izlash deskriptorlar yordamida solishtirishga asoslanadi. Ikki o'lchovli tasvir deskriptorlarining quyidagi guruhlari ajratiladi gradientga asoslangan deskriptorlar, lokal ikkilik deskriptorlar, spektral tasvirga asoslangan deskriptorlar, bazis funksiyali deskriptorlar, shakl deskriptorlari. Shuni ta'kidlash kerakki, ba'zi usullar, ularning xususiyatlariga ko'ra, bir vaqtning o'zida turli guruhlarga bo'linishi mumkin.

Mahalliy ikkilik identifikatorlar ikkilik vektorlar ko'rinishidagi tasvirning kichik maydonining tavsifidir. Eng mashhur mahalliy deskriptorlar mahalliy binar naqshlar va ularning modifikatsiyalari (Local Binary Patterns - LBP). Bu guruhga shuningdek BRIEF (Binary Robust Independent Elementary Features) , ORB (Oriented BRIEF),

BRISK (Binary Robust Invariant Scalable Keypoints) va boshqalar kiradi. SIFT (Scale-Invariant Feature Transform) – bu tasvirlardan xususiyatlar (kalit nuqtalar) olish uchun ishlatiladigan algoritm bo'lib, u har xil o'lcham va orientatsiyalardagi tasvirlardan barqaror va ishonchli xususiyatlar ajratib olish imkonini beradi. SIFT kengaytmasi sifatida SURF (Speeded-Up Robust Features) va ORB (Oriented FAST and Rotated BRIEF) algoritmlari mavjud bo'lib, ular ham tasvirlarni tanish va kalit nuqtalarni aniqlashda foydalaniladi.

KAZE (Accelerated KAZE) deskriptori tasvirlarni qayta ishlash va kompyuter ko‘rish sohasida muhim ahamiyatga ega. U tasvirdagi xususiyatlarni samarali va aniq aniqlash hamda taqqoslash imkonini beradi.

KAZE dan foydalanishga doir Python dasturlash tilida tuzilgan dastur.

```
import cv2
import os
import tkinter as tk
from tkinter import simpledialog
# Yuzni aniqlash uchun Haar kaskadini yuklash
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
# KAZE deskriptorini yaratish
kaze = cv2.KAZE_create()
# Rasm saqlanadigan papka
images_folder = 'tasvir'
# Papka mavjud bo'lmasa, uni yaratish
if not os.path.exists(images_folder):
    os.makedirs(images_folder)

# Papkada mavjud bo'lgan rasmlarni yuklash va KAZE deskriptorlarini hisoblash
saved_images = [os.path.join(images_folder, file) for file in os.listdir(images_folder) if file.lower().endswith(('.png', '.jpg', '.jpeg'))]
saved_descriptors = []
for image_path in saved_images:
    img = cv2.imread(image_path, 0)
    faces = face_cascade.detectMultiScale(img, 1.3, 5)
    for (x, y, w, h) in faces:
        roi_gray = img[y:y + h, x:x + w]
        keypoints, descriptor = kaze.detectAndCompute(roi_gray, None)
        if descriptor is not None:
            saved_descriptors.append(descriptor)
capture_button_pressed = False
```

```
button_x, button_y, button_w, button_h = 10, 400, 150, 50
def draw_capture_button(frame):
    cv2.rectangle(frame, (button_x, button_y), (button_x + button_w, button_y + button_h), (0, 255, 0), -1)
    cv2.putText(frame, 'tasvir', (button_x + 30, button_y + 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
def mouse_callback(event, x, y, flags, param):
    global capture_button_pressed
    if event == cv2.EVENT_LBUTTONDOWN:
        if button_x < x < button_x + button_w and button_y < y < button_y + button_h:
            capture_button_pressed = True
def ask_for_filename():
    root = tk.Tk()
    root.withdraw()
    file_name = simpledialog.askstring("Input", "Ism Familya kiriting:")
    root.destroy()
    return file_name
cap = cv2.VideoCapture(0)
cv2.namedWindow('KAZE deskriptori uchun dastur')
cv2.setMouseCallback('KAZE deskriptori uchun dastur', mouse_callback)

while True:
    ret, frame = cap.read()
    if not ret:
        print("Kamera bilan muammo bor!")
        break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
        roi_gray = gray[y:y + h, x:x + w]
        keypoints, descriptor = kaze.detectAndCompute(roi_gray, None)
        if descriptor is not None:
            bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)  # KAZE NORM_L2 talab qiladi
            best_match_index = None
            best_match_count = 0
            for i, saved_descriptor in enumerate(saved_descriptors):
                if saved_descriptor is not None:
```

```
                matches = bf.match(descriptor, saved_descriptor)
                if len(matches) > best_match_count:
                    best_match_count = len(matches)
                    best_match_index = i
            if best_match_index is not None:
                cv2.putText(frame,          saved_images[best_match_index].split(os.sep)[-
1].split(".")[0], (x, y - 10),
                            cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    draw_capture_button(frame)

    if capture_button_pressed:
        file_name = ask_for_filename()
        if file_name:
            file_path = os.path.join(images_folder, f'{file_name}.jpg')
            success = cv2.imwrite(file_path, frame)
            if success:
                print(f"Rasm saqlandi: {file_path}")
            else:
                print("Rasmni saqlashda xato yuz berdi!")
        capture_button_pressed = False

    cv2.imshow('KAZE deskriptori uchun dastur', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Dasturda python versiyasi 3.9.6 opencv-python 3.4.11.39 dan foydalanilgan. inson1 va inson2 avvaldan mavjud rasmlar yuzni aniqlash uchun. Yuzni tanib olishda KAZE deskriptoridan foydalanib ushbu dasturlar ishlaydi.

Foydalanilgan adabiyotlar ro'yxati

1. Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints." International Journal of Computer Vision, 60(2), 91-110.
2. Mikolajczyk, K., & Schmid, C. (2005). "A Performance Evaluation of Local Descriptors." IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10), 1615-1630.

3. Bai, X., & Liu, J. (2015). "Real-Time Face Recognition with SIFT Features." International Journal of Computer Applications, 118(5), 1-5.
4. https://www.embarcadero.com/ru/product s/delphi - Delphi muhiti haqida rasmiy ma’lumot
5. Mahkamov , A. . (2024). Delphi dasturlash muhitida tuzilgan dasturdan foydalanib microsoft word xujjatlarida ko’p qatorli matnli test variantlarini bilan ishlash afzallik va kamchiliklari. Наука и инновация, 2(19), 34–36. извлечено от https://in-academy.uz/index.php/si/article/view/34528