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

Abdumalikov Abduqodir Abdug'affor o'g'li

Andijon davlat universiteti, Axborot texnologiyalari kafedrasi o'qituvchisi.

Abduqodircoder@gmal.com

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

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

Abstract:

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

Keywords:

BRIEF, ORB, BRISK, SIFT descriptor, local descriptors.

Аннотация:

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

Ключевые слова:

BRIEF, ORB, BRISK, 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

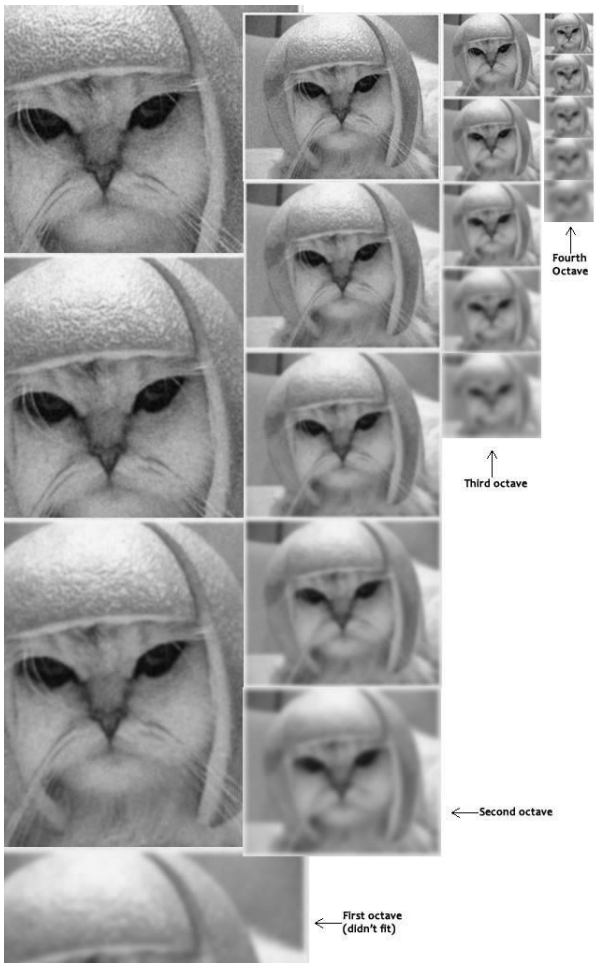
usullar, ularning xususiyatlariga ko'ra, bir vaqtning o'zida turli guruhlarga bo'linishi mumkin.

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, lekin ular SIFTga qaraganda tezroq va samaraliroq hisoblanadi.
SIFT ning asosiy afzalliklari quyidagilardan iborat:

- xarakteristikalar mahalliy, shuning uchun okklyuzion va buzilishlarga chidamli (dastlabki segmentatsiyasiz);
- individual xususiyatlarni obyektlarning katta ma'lumotlar bazasi bilan solishtirish mumkin;
- kichik obyektlar uchun ham ko'plab xarakteristikalar yaratilishi mumkin;
- Deskriptor real vaqtda ishlashga yaqin;

Deskriptor osonlik bilan har xil obyekt turlarining keng doirasiga o'zgartirilishi mumkin. Haqiqiy dunyo obyektlari faqat ma'lum bir miqyosda ma'noga ega. Odam stol ustidagi qand kupligini yoki yo‘lda mashinani osongina ko‘radi, lekin Somon yo‘lini ham, atom tuzilishini ham ko‘ra olmaydi. Obyektlarning bunday ko'p miqyosli tabiati tabiatda juda keng tarqalgan. Masshtab fazosi bu tushunchani raqamli tasvirlarda takrorlashga urinadi.Tasvir fazosi shkalasi L(x, y, s) funksiya bo‘lib, u Gauss yadrosini turli masshtablarda (loyqalik) kiritilgan tasvir bilan

birlashtirish orqali olinadi. Kosmik masshtab oktavalarga bo'linadi, oktavalar soni va masshtab asl tasvirning o'lchamiga bog'liq. Shunday qilib, biz asl tasvirning bir nechta oktavalarini hosil qilamiz. Har bir oktavaning tasvir hajmi avvalgisining yarmini tashkil qiladi. Fazoviy masshtabga misol quyidagi 1-rasmda keltirilgan.



1-rasm – Tasvir maydoni masshtabi

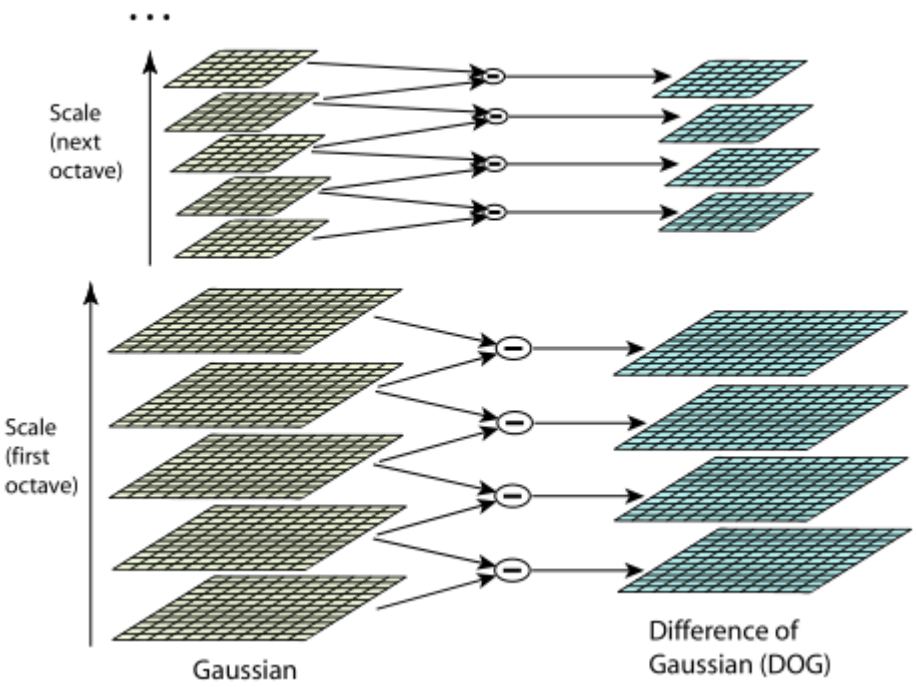
SIFT usulining asosiy xususiyati - Gauss piramidasining qurilishi va Gausslarning farqlari. Tasvir quyidagi formulaga mos keladigan Gauss filtri yordamida loyqalanadi:

$$L(x, y, o) = G(x, y, o) * I(x, y)$$

Formulada L(x,y,o) koordinatalari (x,y) va loyqa radiusi o bo'lgan nuqtadagi Gaussning qiymati, G(x,y,o) Gauss yadrosi, I(x, y) - asl tasvirning qiymati. Gauss yadrosi va asl tasvirning qiymati o'rtasida ko'paytirish emas, balki konvolyutsiya operatsiyasi sodir bo'ladi. Keyinchalik, Gausslarning farqi hisoblanadi. Tasvirning har bir pikselidan turli darajadagi loyqalik darajasiga ega tasvir pikseli chiqariladi:

$$D(x, y, o) = L(x, y, ko) - L(x, y, o)$$

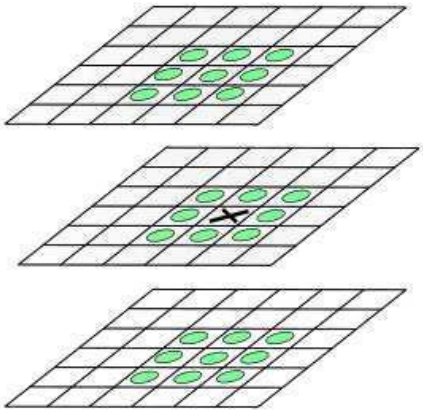
Keyinchalik masshtabli bo'shliqlar oktavalarga bo'linadi. Shundan so'ng, oktavadan tashqariga cho'zilgan yana ikkita Gauss tugallanadi. Shu bilan birga, Gauss farqlari piramidasi quriladi, unda Gauss piramidasiga qaraganda bitta kamroq tasvir bo'ladi. Piramidalarning yaratilishi quyidagi 2-rasmda ko'rsatilgan.



2-rasm - Gauss piramidalarining qurilishi va Gausslarning farqlari

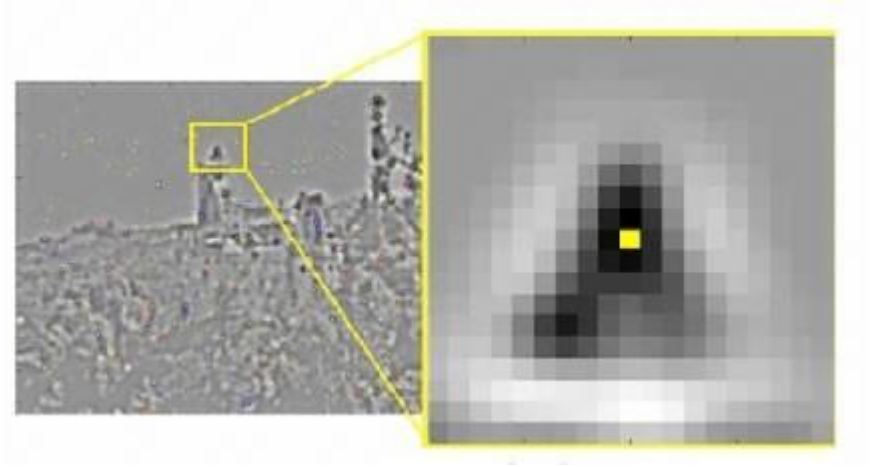
Keyingi qadam maxsus nuqtalarni aniqlashdir. Agar xochli nuqtadagi qiymat aylanali nuqtalarning qiymatlaridan sezilarli darajada farq qilsa, bunday nuqta Gausslar farqining mahalliy ekstremumi bo'ladi, ya'ni maxsus hisoblanadi. Yagona

nuqtaga misol quyidagi 3-rasmda keltirilgan. Ekstremum xoch bilan belgilangan va maxsus nuqtadir.



3-rasm. Mahalliy ekstremum bo'yicha yagona nuqtani aniqlash

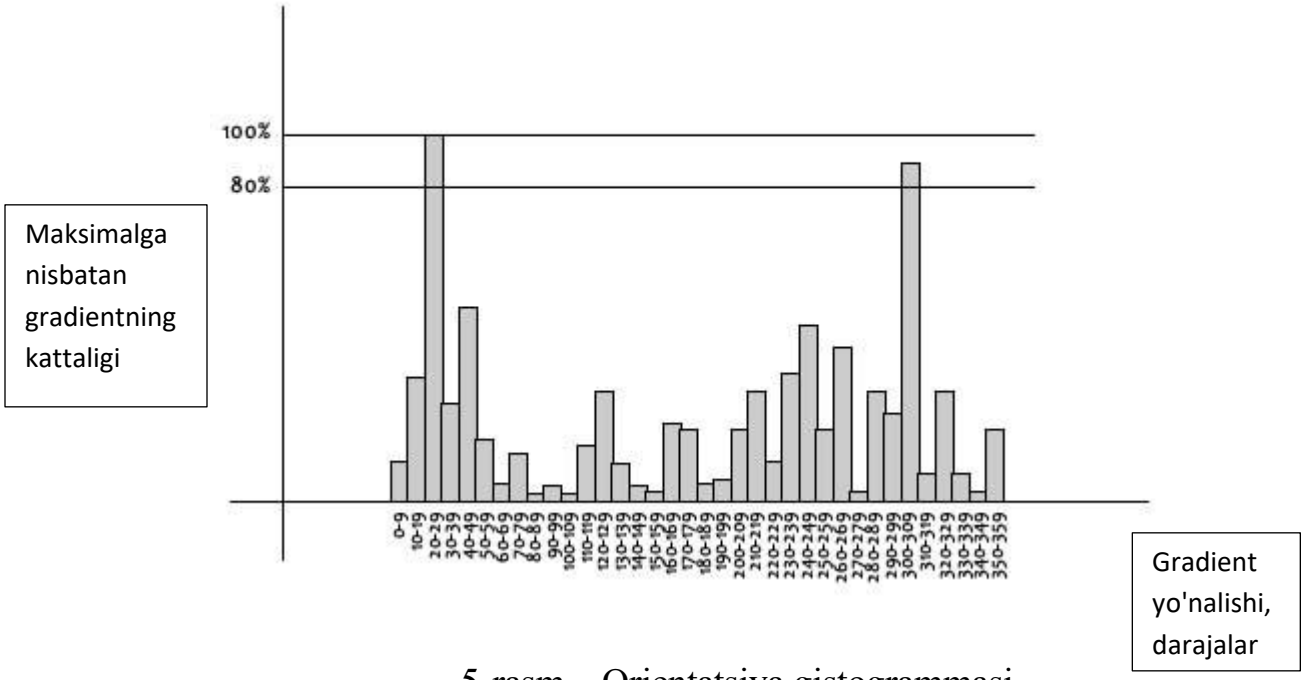
Keyin kalit nuqtalar aniqlangan shkala yordamida barqarorlik tekshiriladi (bu loyqa tasvir shkalasi bilan bir xil). Keyinchalik, har bir asosiy nuqtaga aylanishga chidamli bo'lish uchun yo'nalish beriladi. Rasmdagi yagona nuqtaga misol quyidagi 4-rasmda ko'rsatilgan.



4-rasm – Rasmdagi maxsus nuqtaga misol

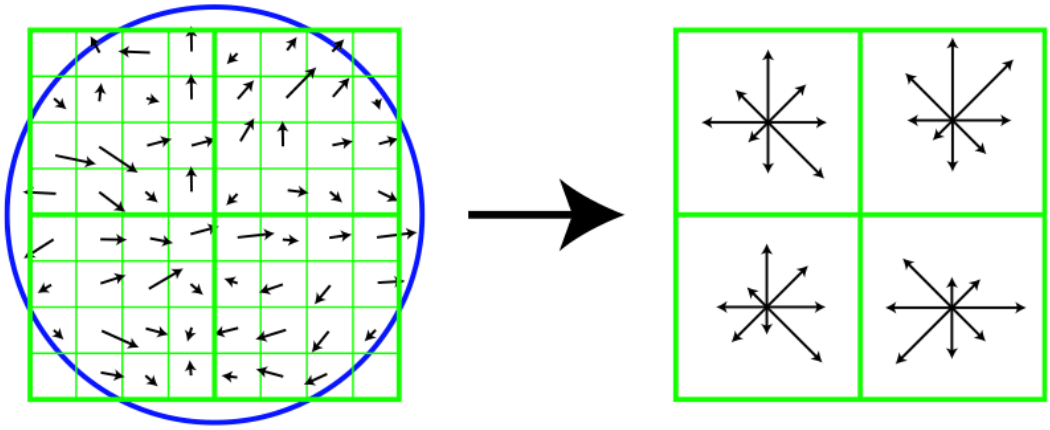
Asosiy nuqtaning joylashgan joyi atrofida masshtabga bog'liq qo'shnichilik olinadi va bu sohada gradientning kattaligi va yo'nalishi hisoblanadi. Orientatsiya gistogrammasi 360 gradusni o'z ichiga olgan 36 quti bilan yaratilgan. Misol uchun,

ma'lum bir nuqtada gradientning yo'nalishi (orientatsiyani yig'ish sohasida) 18,759 daraja, keyin u 10-19 darajali qutiga o'tadi. Va savatga qo'shilgan miqdor o'sha nuqtadagi gradientning kattaligiga mutanosibdir. Ushbu amalni asosiy nuqta atrofidagi barcha piksellar uchun takrorlagandan so'ng, gistogramma bir nuqtada cho'qqiga ega bo'ladi.



5-rasm – Orientatsiya gistogrammasi

5-rasmdagi misol gistogrammadagi eng yuqori cho'qqini oladi va uning qiymatining 80% dan yuqori bo'lgan har qanday cho'qqi ham orientatsiyani hisoblash uchun hisobga olinadi.Bu yondashuv joylashuvi va masshtablari bir xil, lekin turli yo‘nalishdagi asosiy nuqtalarni yaratishga yordam beradi va asosiy nuqta moslashuvining barqarorligiga hissa qo‘shadi.SIFT deskriptori masshtab bo‘yicha yagona nuqtaga imkon qadar yaqin bo‘lgan Gauss bo‘yicha aniqlanadi. . Deskriptorning ishlashi 6-rasmda ko'rsatilgan.



6-rasm - Rasm gradientlarining sxematik ko'rinishi (chapda) va ulardan olingan deskriptor (o'ngda) 6-rasmdan ko'rinib turibdiki, deskriptor oynasidagi har bir gradientga uchta koordinata berish mumkin: x,y,n. Ushbu gradientlar asosida tavsiflovchi tuziladi. SIFT deskriptori tasvirni qayta ishlash uchun yaxshi tanlovdir, lekin uning bir qator kamchiliklari bor. Nuqtalar va ularning tavsiflovchilari har doim ham kerakli talablarga javob bermaydi va har doim ham yechim topib bo'lmaydi. Agar tasvir takrorlanuvchi fragmentlardan iborat bo'lsa (masalan, devor), u holda bu deskriptor to'g'ri ishlamaydi

SIFT dan foydalanishga doir Python dasturlash tilida tuzilgan dastur.

```
import cv2
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
sift = cv2.SIFT_create()
saved_images = ['inson1.jpg', 'inson2.jpg']
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 = sift.detectAndCompute(roi_gray, None)
        saved_descriptors.append(descriptor)
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    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 = sift.detectAndCompute(roi_gray, None)
        if descriptor is not None:
            bf = cv2.BFMatcher(cv2.NORM_L2, crossCheck=True)
            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(".")[0], (x, y
- 10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
    cv2.imshow('frame', 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 SIFT deskriptoridan foydalanib ushbu dasturlardan foydalaniladi.

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.Bhandari, A., & Rai, A. (2017). "Face Detection and Recognition Using SIFT in Python." International Journal of Advanced Research in Computer Science, 8(5), 100-103.

5.Rafique, M., & Zia, A. (2018). "A Comprehensive Review of SIFT Algorithm and Its Applications in Image Processing." Journal of Computer Science, 14(8), 1234-1241.

6.Zhang, J., & Zhang, Y. (2019). "Python-Based Implementation of Image Matching Using SIFT." Journal of Imaging Science and Technology, 63(5), 1-10.

7.Ge, S., & Wu, Y. (2020). "Using SIFT Features for Object Detection in Python." International Journal of Computer Vision, 128(3), 542-556.