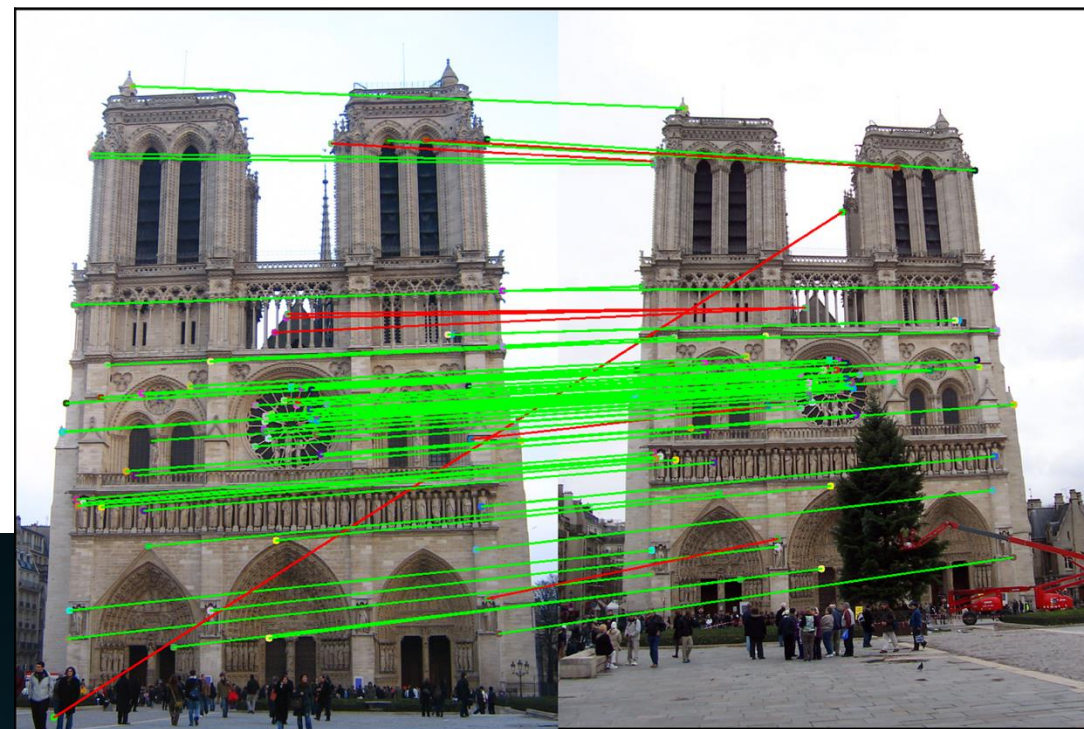


Споредба на Локални Дескриптори

Горазд Филиповски 223070



Мотивација & Преглед

- **Мотивација**

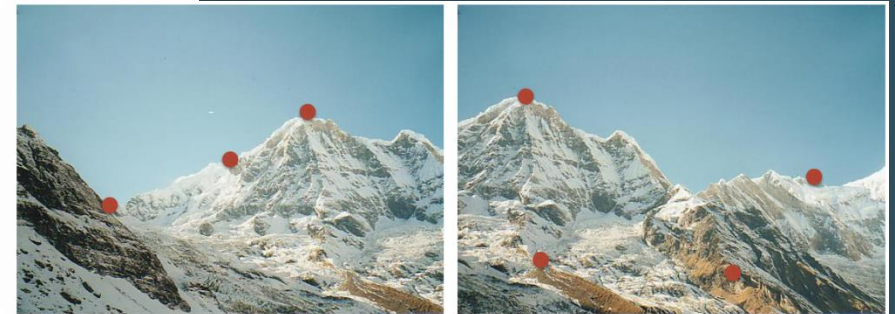
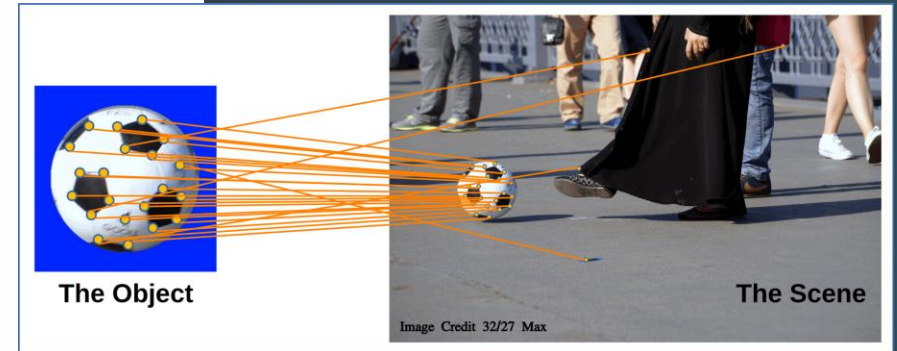
- Локалните дескриптори се клучни за совпаѓање и препознавање во компјутерска визија
- Изборот меѓу многуте (SIFT, ORB, итн.) е важен за оптимална перформанса/брзина

- **Цел на Проектот**

- Споредба на разни дескриптори под различни трансформации (ротација, скала, осветлување)
- Идентификација на јачините/слабостите и во 2D-планарни и во 3D-објекти

- **Практична употреба**

- Корисно за задачи како image stitching, 3D реконструкција, AR/VR



Алгоритми кои се обработени

- **Тестирани Методи**

1. SIFT (float-based, робустен на скала/ротација)
2. ORB (binary, ограничен број на клучни точки но брз)
3. BRIEF & BRISK (бинарни дескриптори со различни детектори)
4. AKAZE & KAZE (нелинеарен scale space, често детектираат многу клучни точки)
5. RootSIFT (SIFT со постпроцес за подобрена нормализација)

- **Кратка Споредба**

- Float vs. binary дескриптори
- Компромис меѓу брзина и точност

Парови слики користени за евалуација

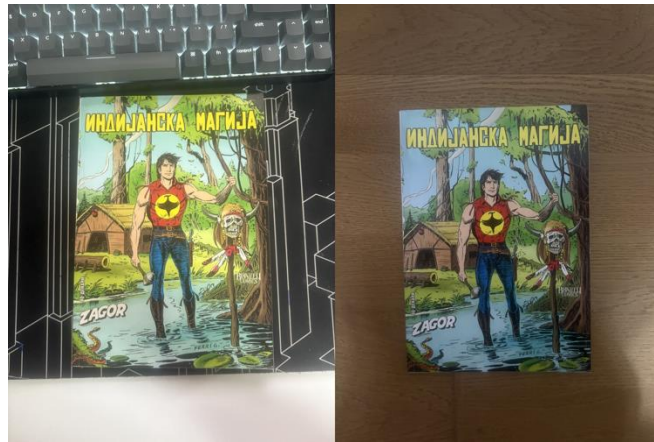
- **Шест Пара**

1. Книга со различна позадина
2. Книга со ротација
3. Книга со ротација + промена во осветлување
4. Книга со различна скала + ротација + додадени предмети
5. 3D статуа на Abraham Lincoln (различни агли)
6. 3D статуа на Статуата на Слободата (различни агли)

- **Трансформации**

- 2D-планарни (Пар 1–4) наспроти 3D објекти (Пар 5–6)
- Ротација, скала, агол на гледање, осветлување

Користени парови слики



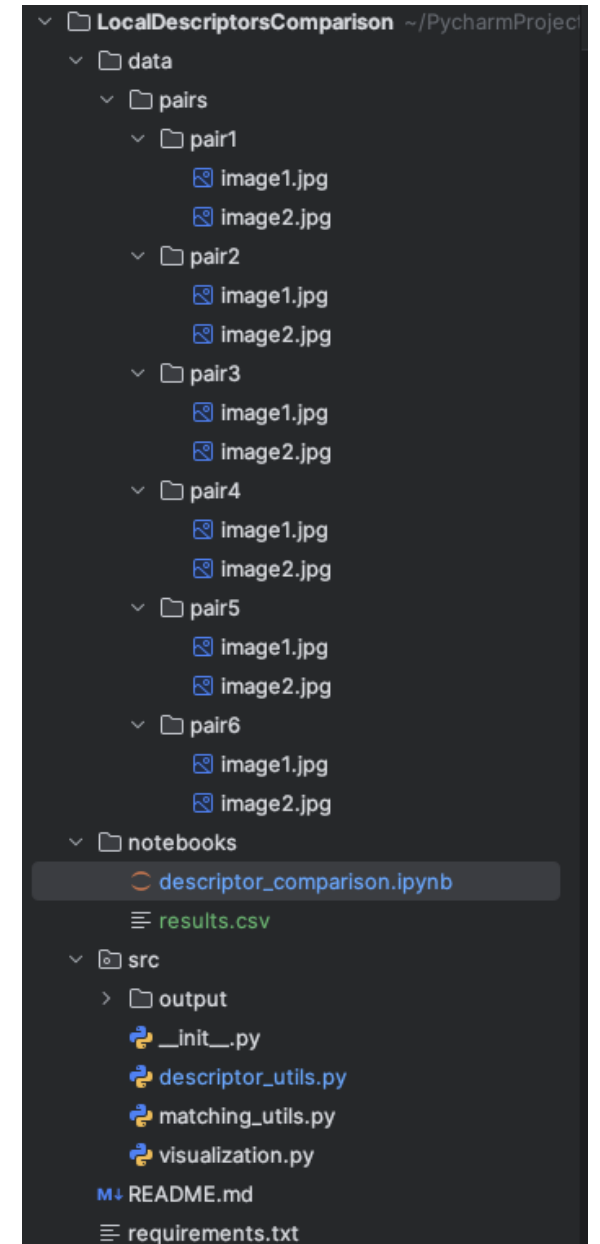
Методологија

- **Работен Тек во Јупитер Тетратката (notebook)**

1. Вчитување слики од data/pairs
2. Екстракција на карактеристики (SIFT, ORB, итн.) во descriptor_utils.py
3. Совпаѓање на дескриптори во matching_utils.py (BFMatcher + ratio test)
4. Пресметка на метрики: број на клучни точки, број на добри совпаѓања
5. Визуелизација на совпаѓања во visualization.py

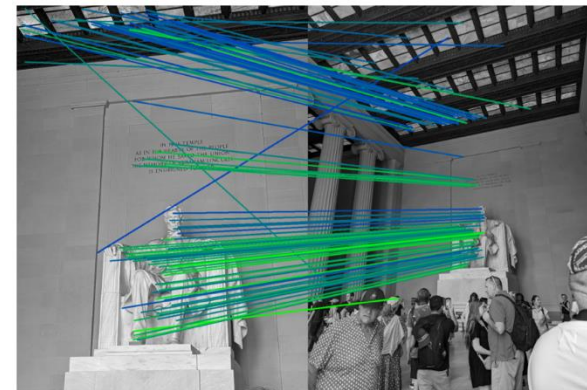
- **Главни Скрипти**

- descriptor_utils.py
- matching_utils.py
- visualization.py



Визуелни Примери

- **Преклопување на Совпаѓања**
 - Слики со линии што ги поврзуваат клучните точки (на пр., `draw_colored_matches()`)
 - SIFT vs. ORB кај Pair 1 (многу линии наспроти малку)
 - KAZE кај Pair 4 (голем број клучни точки, густо совпаѓања)
- **Заклучоци од овие визуелизации**
 - Дескриптори со повеќе клучни точки имаат повеќе можности за совпаѓање
 - 3D аглите предизвикуваат потешкотии кај сите методи



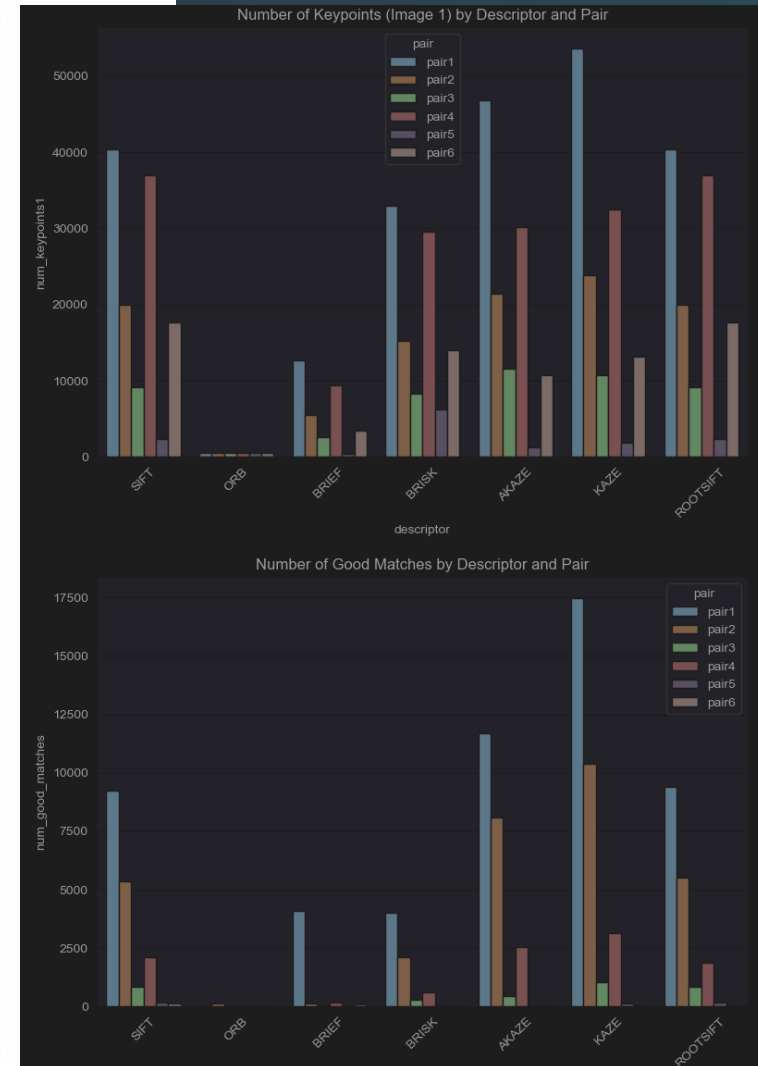
Резултати – Број на Клучни Точки & Добри Совпаѓања

• Клучни Забелешки

- KAZE/AKAZE детектираат десетици илјади клучни точки во некои сцени
- ORB останува на 500, што го ограничува бројот на совпаѓања
- Кај 3D статуите има помал број клучни точки и совпаѓања кај сите дескриптори

• Добри Совпаѓања

- KAZE најчесто води; AKAZE, SIFT и RootSIFT не се многу позади во планарните сцени
- BRIEF, BRISK, ORB покажуваат понизок број совпаѓања при комплексни трансформации



Кога & Кој алгоритам да се користи

- **KAZE**

- **Јачина:** Детектира многу клучни точки, робустен на промена во скала/осветлување.
- **Корисен Кога:** Ви треба максимален опфат на карактеристики за детални, планарни сцени и можете да прифатите побавно пресметување.

- **AKAZE**

- **Јачина:** Сличен на KAZE но побрз; нуди и бинарни и флоат дескриптори.
- **Корисен Кога:** Сакате робуственост налик на KAZE со нешто помала пресметковна цена.

- **SIFT**

- **Јачина:** Добра инваријантност на скала/ротација, многу истражуван и стабилен.
- **Корисен Кога:** Можете да прифатите умерено време на пресметка и ви требаат сигурни совпаѓања, особено при умерени трансформации.

- **RootSIFT**

- **Јачина:** SIFT со подобрена нормализација, често подобар кај репетитивни шари.
- **Корисен Кога:** Сакате малку подобра прецизност од SIFT кај одредени услови.

- **ORB**

- **Јачина:** Многу брз, бинарен дескриптор со вградена ориентациона корекција.
- **Корисен Кога:** Имате строги услови за апликации кои функционираат во реално време и 500 клучни точки ви се доволни.

- **BRIEF**

- **Јачина:** Исклучително брз, мала мемориска употреба.
- **Корисен Кога:** Ви треба едноставен, брз дескриптор за средини со мали ротации/скали и можете да го споите со соодветен детектор.

- **BRISK**

- **Јачина:** Бинарен дескриптор со вградена скала, умерено брз.
- **Корисен Кога:** Сакате некаква инваријантност на скалирање, но претпочитате бинарен дескриптор поради ефикасност.

Заклучок

- „Најдобриот“ дескриптор зависи од приоритетите: брзина или точност

- KAZE/AKAZE лидерски во број на совпаѓања во 2D сцени; SIFT/RootSIFT исто така силни

- ORB брз, но ограничен; погоден за real-time апликации

Благодарам на
вниманието!