Домашнее задание №2. ORB detector.

В данной работе производится реализация алгоритма ORB (Oriented FAST and Rotated BRIEF). ORB состоит из двух частей - детектора особых точек FAST и дескриптора BRIEF. В данной реализации рассматривается FAST-9 (функция fast) с дополнительными улучшениями:

1. отбор наиболее подходящих особых точек по метрике Харриса (Harris measure);
2. улучшение распознавания особых точек на разном масштабе при помощи поиска особых точек на пирамиде изображений (функция pyramid\_fast).
3. вычисление центроида инерции окрестности особой точки для устойчивости к вращению (функция calc\_centroid).

Для дескриптора BRIEF также используются дополнительные улучшения:

1. для каждой особой точки на основе базового набора точек для дескриптора выбирается набор точек, соответствующий ориентации центроида инерции особой точки (функция steered\_brief).

В итоге полная реализация ORB находится в функции orb.

В функции main реализован функционал для тестирования реализованной функции orb и сравнения ее с аналогичной функцией из библиотеки OpenCV. Для тестового изображения производится поворот, масштабирование, изменение перспективы и аффинное преобразование. Далее для этих изображений производится поиск особых точек и их сопоставление. Результаты представлены на следующих изображениях.

|  |
| --- |
|  |
|  |

Таблица 1. Тестирование алгоритма для масштабирования. Сверху - реализованная версия, снизу - библиотечная.

|  |
| --- |
|  |
|  |

Таблица 2. Тестирование алгоритма для поворота. Сверху - реализованная версия, снизу - библиотечная.

|  |
| --- |
|  |
|  |

Таблица 3. Тестирование алгоритма для изменения перспективы. Сверху - реализованная версия, снизу - библиотечная.

|  |
| --- |
|  |
|  |

Таблица 4. Тестирование алгоритма для аффинного преобразования. Сверху - реализованная версия, снизу - библиотечная.

Линиями соединены 10 наиболее похожих точек, красными кругами выделены особые точки, обнаруженные реализованной версией FAST. Можно заметить, что в библиотечной реализации больше совпадающих точек. Это может быть связано с более эффективной реализацией FAST в OpenCV (там отбрасываются особые точки, расположенные слишком близко друг к другу), а также с более эффективной версией BRIEF в OpenCV (там производится отбор наименее скоррелированных точек для составления дескриптора).

Также было измерено время работы алгоритма в библиотечной версии и в реализованной версии. Время для библиотечной/реализованной версии: 0.04/226.04 с. Такое существенное расхождение может быть объяснено несколькими факторами:

1. библиотечная реализация написана на C++;
2. в библиотечной реализации используется оптимизированный вариант FAST на базе дерева.