2D Feature Detection and Tracking

# Задача

С помощью библиотеки OpenCV реализовать детектирование и трекинг уголков на видео. См. пример по ссылке: <https://youtu.be/DbFovlQUltc>.

Нужно пользоваться кодом из репозитория <https://github.com/cvcourse/home-assignments-autumn-2022>.

# Подробное описание

* Необходимо реализовать детектирование и трекинг (отслеживание) уголков с помощью библиотеки OpenCV. Самостоятельно методы детектирования и трекинга писать не нужно.
* Основная цель задания — получить на выходе хорошее покрытие кадров уголками и длинные треки. Что это значит:
  + Уголки должны быть найдены/отслежены на всех кадрах видео (если они там есть, конечно).
  + Один и тот же уголок, отслеженный с помощью оптического потока на протяжении какого-то числа кадров, называется треком. Получаемые вами треки должны быть по возможности длинными, то есть чем дольше трек не обрывается, тем лучше.
  + Если на кадрах по ходу видео появляются новые уголки, они должны регистрироваться вашей программой. Треки не должны начинаться только, например, на первом кадре видео и затем постепенно пропадать.
  + Уголки должны покрывать кадр по возможности равномерно и полностью (то есть не должно быть такого, что на половине кадра уголки не отслеживаются, хотя они там очевидно есть).
    - Например можно жадно выбрать подмножество из лучших N уголков, так чтобы они были не ближе чем M пикселей друг от друга.
* Всевозможные константы, необходимые для настройки методов поиска и отслеживания уголков, подобрать самостоятельно.
* Домашнее задание нужно выполнять с использованием выдаваемого нами кода:
  + Реализацию поиска и трекинга уголков нужно писать в файле corners.py. (\_build\_impl). В нем есть какой-то минимальный код, его можно менять весь, но так, чтобы не изменить интерфейс модуля corners (подробнее см. в требованиях).
  + Файл \_corners.py содержит код, касающийся непосредственно уголков, но реализованный преподавателями. Туда можно и нужно смотреть, чтобы понимать, что происходит.
  + Если очень хочется или нужно, разрешается менять код в файле \_corners.py, но так, чтобы не сломать уже существующие интерфейсы (подробнее см. в требованиях).
  + Файл corners.py может использоваться как консольное приложение для запуска вашей реализации поиска уголков и отображения результатов.
  + Подробности по коду см. в комментариях в самом коде.

# Требования

* Ваша программа должна успешно запускаться с помощью testrunner.py и корректно проходить все тесты из датасета, расположенного на [Google Drive в директории курса](https://drive.google.com/drive/folders/1A7J7S4uzqk4TkvaM1mkWcayQmk3y5gCd). Для запуска используйте конфигурацию тестов dataset\_ha1.yml.
* Для детектирования уголков необходимо использовать именно функцию качества уголка по Ши — Томаси.
* Для отслеживания уголков необходимо использовать пирамидальный метод Лукаса — Канаде.
* Все дополнительные зависимости, которые вы, возможно, захотите подтянуть, должны быть прописаны в requirements.txt, без проблем ставиться с помощью pip и работать под Ubuntu.
* Запрещено менять существующий интерфейс модуля corners.py, но при этом разрешено его расширять. В частности:
  + запрещено удалять функции, видные наружу, но можно добавить свои, если очень хочется;
  + запрещено удалять или переименовывать свойства класса FrameCorners, но разрешено добавлять свои;
  + запрещено удалять или переименовывать методы класса CornerStorage, но можно добавлять свои.
* Запрещено менять консольный интерфейс модуля corners.py, но разрешено его расширять так, чтобы новые параметры не были обязательными.
* Программа должна показывать индикатор выполнения (progress bar), адекватно отображающий прогресс работы с видео.

# Подсказки

Часто OpenCV требует какой-то определенный тип данных на входе, хоть это и не указано в документации. Если вдруг вылетает какое-то исключение со странным текстом ошибки, попробуйте поискать по этому тексту — объяснение скорее всего найдется.

# Дополнительные задания

1. **(1 балл) Подсчет качества уголков и фильтрация**  
   ***Введение***  
   Можно различными способами попробовать оценить качество уголка. Например, просто взять минимальное собственное число из метода Ши – Томаси. Или после отслеживания уголка оптическим потоком посчитать евклидово расстояние между окнами вокруг уголка в предыдущем и текущем кадрах. Обычно подобные оценки качества так или иначе используются при поиске уголков или для отбрасывания плохо отслеженных уголков.  
   В нашем случае удобнее будет отслеживать уголки на видео с одними требованиями к качеству (задаются в качестве параметров при вызове функций OpenCV), а затем иметь возможность фильтровать их (т. е. оставлять наиболее качественные) уже на этапе поиска 3D-решения. Такой подход позволит экспериментировать с требованиями к качеству уголков не выполняя анализ видео каждый раз, когда захочется изменить пороги качества.  
   ***Задание***  
   В дополнение к уже существующим полям записать в структуру FrameCorners какие-нибудь метрики качества уголков (не менее двух). Реализовать возможность фильтрации уголков по качеству с помощью класса \_StorageFilter (для примера см. функцию without\_short\_tracks).
2. **(2 балла) Пирамиды**  
   Реализовать поиск уголков на нескольких масштабах пирамиды кадра. При этом нужно правильно отмасштабировать все параметры уголков так, чтобы пользователь не задумывался, на каком уровне пирамиды они были найдены, и считал, что размеры уголков и т. д. заданы в масштабе оригинального разрешения видео.