

Determinant of Hessian Local Maxima Detector.

Задание.

Реализовать детектор локальных максимумов определителя Гессияна (*Determinant of Hessian Local Maxima Detector*). Гессиян (*Hessian*) - оператор, результатом применения которого является 2×2 матрица вторых частных производных, найденная для каждого пикселя. Значения вторых производных необходимо вычислить, используя операцию свертки изображения (*convolution*) с вторыми производными Гауссиана (*Gaussian*). В данном случае ядром свертки (*kernel*) является соответствующая производная оператора Гаусса. Локальные максимумы определителя Гессияна ищутся по области 3×3 . Далее ненадежные максимумы отфильтровываются по нижнему пороговому значению. Необязательным (но считающимся большим преимуществом) является нахождение локальных максимумов с субпиксельной точностью. Субпиксельное положение максимума уточняется при помощи аппроксимации двумерной функции поверхностью второго порядка. Поверхность второго порядка находится, используя разложение функции в ряд Тейлора. Коэффициенты ряда можно определить применяя дифференциальные операторы размера 3×3 . Вся обработку необходимо проводить над черно-белым (*grayscale*) изображением (цветное изображение предварительно конвертируется в *grayscale*). Детектор должен быть реализован на языках C/C++, без использования сторонних библиотек обработки изображений. Крайне желательно реализовать сам алгоритм на чистом C. Допускается использование контейнеров STL или Qt. Параметрами детектора являются размер Гауссиана (*scale*) и нижнее пороговое значение определителя Гессияна для локальных максимумов. Детектор возвращает массив или контейнер с координатами всех найденных максимумов выше порогового значения. По возможности он должен работать в отдельном потоке. Желательно, чтобы приложение могло работать со сжатыми изображениями (jpeg). Время на выполнение задания неограниченно.

Обязательные требования.

- Реализация детектора на C/C++ (желательно на чистом C), без использования сторонних библиотек. Допускается использование контейнеров STL или Qt.
- Нахождение положения максимума с субпиксельной точностью *не является* обязательным условием.

Преимущества реализации.

- Высокая производительность детектора.
- Использование любых технологий параллельных вычислений (OpenCL, OpenMP, CUDA, DirectCompute, C++ AMP, TBB) является *особым* преимуществом.
- Использование наиболее компактного типа данных для хранения промежуточных результатов, позволяющего обеспечить необходимую точность (порядок погрешности промежуточных результатов сравним с погрешностью на исходном изображении). Например, для хранения производной не обязательно использовать 32-битный float.
- Нахождение положения локального максимума с субпиксельной точностью.
- Реализация работы детектора в отдельном потоке.
- GUI-приложение с возможностями задания параметров детектора и отображения результатов его работы, наподобие предоставленного примера реализации.
- Реализация GUI используя Qt.
- Кросс-платформенная реализация.
- Понятный код с достаточным количеством комментариев.