Индивидуальные задания по модулю «Проектирование многопоточных приложений в Qt»

Разработка многоплатформенного ПО

- Необходимо разработать программу просмотра изображений в форматах JPEG и всех остальных, поддерживаемых стандартными средствами Qt.
- Необходимо реализовать применение одного из фильтров или других алгоритмов анализа/обработки изображения (см. ниже) к просматриваемому изображению в памяти с использованием библиотеки OpenCV: `http://docs.opencv.org/3.1.0/d7/dbd/group__imgproc.html`. Результат преобразования должен быть выведен после обработки.
- Обработку изображения необходимо выполнять в отдельном потоке, при этом на время его работы соответствующие элементы пользовательского интерфейса должны быть деактивированы.
- Для описания проекта необходимо использовать систему CMake вместе с командой `find_package()` для поиска набора библиотек OpenCV.
- Для преобразования форматов изображения между Qt и OpenCV можно использовать любые готовые решения (например, см. `https://asmaloney.com/2013/11/code/converting-between-cvmat-and-qimage-or-qpixmap/`).

1. Применить фильтр Шарра (Scharr) к заданному изображению.
2. Применить фильтр Собеля (Sobel) к заданному изображению.
3. Применить фильтр сглаживания `medianBlur` к заданному изображению.

4. Найти все окружности в заданном изображении, используя функцию `HoughCircles()`, преобразовав предварительно цветность изображения к оттенкам серого. Выделить все найденные окружности в исходном изображении.

5. Найти все отрезки прямых линий в заданном изображении, используя функцию `HoughLinesP()`, преобразовав предварительно изображение к оттенкам серого. Выделить все найденные отрезки в исходном изображении.

6. Найти все углы в изображении, используя функцию `preCornerDetect()`, преобразовав предварительно цветность изображения к оттенкам серого. Выделить все найденные углы небольшими окружностями в исходном изображении.

7. Найти заданный прямоугольный фрагмент в заданном изображении, используя функцию `matchTemplate()`. Выделить его границы в изображении.

8. Выполнить удаление шума в изображении при помощи функции `fastNlMeansDenoisingMulti()`.

9. Найти гистограмму заданного изображения при помощи функции `calcHist()`. Изобразить полученную гистограмму и вывести изображение в одном или разных окнах.