1. ***Описание проделанной работы***

*Была произведена работа над рядом функций из библиотеки OpenCV, а именно:*

* Модуль ***imgproc***
  + *Bilateral Filter*
  + *Filter 2D (Linear convolution)*
  + *Box Filter (Linear convolution with scalar kernel)*
  + *Morph Filter*
* Модуль***video***
  + *BackgroundSubtractorMOG2Impl::getBackgroundImage*
  + *BackgroundSubtractorMOG2Impl::apply*

Также во время работы, были обнаружены некоторые баги, тесты которых лежат в папке “bugs” в Git репозитории проекта. Каждый тест лежит в отдельном архиве.

1. ***Результаты работы***

При выборе приложений для оптимизации использовались следующие критерии:

* Кроссплатформенность
* Соотношение пользы/популярности приложения
* Наличие кода, который можно оптимально выполнить на GPU

Для оптимизации были выбраны следующие приложения:

* FFmpeg (была оптимизирована библиотека libavfilter, которую использует FFmpeg)
* OpenBR
* Gimp OpenCV Plug-in

На данный момент готово первое приложение, FFmpeg. А именно, было оптимизировано применение фильтра (Bilateral Filter) на видеоряд с помощью функций библиотеки OpenCV (в пункте 3 описан алгоритм сборки и тестирования данного приложения). Ниже представлены некоторые результаты замеров на процессоре Intel Core i5 3210M (Ivy Bridge), единица измерения секунды (меньше - лучше):

*Таблица с подробными результатами по тесту (тестовое видео лежит в репозитории)*

|  |  |  |
| --- | --- | --- |
|  | Gen with CPU\* | CPU |
| 3x3 | 8.9s | 8.9s |
| 5x5 | 7.2s | 13.7s |
| 7x7 | 10s | 26s |
| 9x9 | 15s | 42s |
| 11x11 | 23s | 62s |

\**Gen with CPU – здесь подразумевается параллельное исполнение на двух устройствах*

1. ***Сборка и тестирование***

Чтобы собрать пакет FFmpeg под Windows необходимо:

* 1. Для начала необходимо на машине установить и сконфигурировать MSYS2, его окружение, а так же загрузить некоторые Unix библиотеки. Для этого необходимо загрузить и установить пакет MSYS 2 с сайта <https://msys2.github.io>, а так же проделать инструкцию, предлагаемую на сайте.
  2. Далее необходимо склонировать репозиторий проекта FFmpeg с сайта <https://github.com/FFmpeg/FFmpeg>
  3. Загрузите с репозитория проекта (ссылка на него была выше) файл opencv.pc (он лежит на репозитории в папке “FFmpeg”). Откройте директорию с установленным пакетом MSYS2 (по умолчанию ставится в директорию C:\msys64), далее раздел \mingw64\lib\pkgconfig и поместите сюда ранее загруженный файл opencv.pc
  4. В корне диска C необходимо создать директорию с названием “opencv” (без кавычек, разумеется) и в ней создать директории “include” и “lib” соответственно. По данным директориям необходимо разложить файлы библиотеки opencv (после ее сборки). Они находятся в <build\_dir>/install и <build\_dir>/install/x64/vcXX соответственно.
  5. Далее необходимо открыть MinGW-w64 Win64 Shell (если у вас 64-битная машина), зайти в директорию с файлами проекта FFmpeg и выполнить следующую команду: ***./configure --disable-yasm --enable-libopencv***
  6. После того, как проект был сконфигурирован, необходимо вызвать в текущей директории утилиту ***make***
  7. FFmpeg необходимо для работы найти библиотеки из пакеты OpenCV, а именно opencv\_core300.dll и opencv\_imgproc300.dll, которые были получены после сборки OpenCV . Далее необходимо прописать в переменную PATH путь до данных библиотек.
  8. После компиляции, можно приступать к тестированию. Команда для теста имеет следующий синтаксис: ./ffmpeg -i <имя семпла> -y -vf "ocv=smooth:bilateral:5x5" temp.avi

В команде можно выбрать необходимый нам фильтр из библиотеки OpenCV из следующего списка:

* + - dilate
    - erode
    - smooth (приведен в примере) делится на:
      1. blur
      2. blur\_no\_scale
      3. median
      4. gaussian
      5. bilateral

Подробную справку про фильтры можно получить здесь <http://ftp.trinome.com/ffmpeg/doc/libavfilter.html#ocv>

В тесте также можно менять размер матрицы фильтра (в примере, например, стоит матрица размера 5x5)