

Проекты

В рамках курса предлагается поработать в командах до 3х человек над проектом. На команды разбиваемся по собственному желанию (в том числе разрешается команда из разных учебных групп).

Код должен храниться в репозитории, ссылки выдаются при регистрации команды.

Свой проект нужно будет защищать Виктории. Защищать можно как полностью готовую реализацию, так и по частям. Защиты проводятся только во время занятий.

Задание 0 Подготовка

В рамках проекта вы реализовываете свой графический редактор, который должен в обязательном порядке поддерживать определенный функционал.

Требуемый функционал будет выкладываться по мере изучения лекционного материала.

Начать следует с определением состава команды, в которой вы будете работать, и изучения фреймворков по выбранным вами языкам программирования.

Задание 1 Представление изображений

Цель работы: изучить способ хранения серых и цветных изображений в формате PNM (P5 и P6).

Описание

Реализация функционала по чтению и записи изображений формата PNM (P5 и P6).


Реализация приложения с графическим интерфейсом для просмотра изображений формата PNM.

Поддержка функций открытия (чтения) изображений из файла и сохранения в файл через пользовательский интерфейс.

Работа с PNM файлами должна быть реализована полностью самостоятельно (без использования каких-либо библиотек по работе с изображениями).

Должна быть реализована базовая обработка ошибок с информированием пользователя: файл не удалось открыть или сохранить, формат файла не поддерживается.

[UPD 17.09]

1. В PNM не обязательно поддерживать комментарии. При тестировании вашего проекта будут подаваться файлы, в которых гарантировано нет комментариев.
2. Для просмотра изображений под Windows можно использовать wic codec ( WIC Codec), pictureview (плагин для far manager)

или GIMP/Photoshop и прочие приличные графические редакторы.

Задание 2 Цветовые пространства

Цель работы: реализовать функционал, который позволяет проводить преобразования между цветовыми пространствами.

Описание

Необходимо реализовать следующие возможности:

- сохранение в файл и загрузку из файла изображений в выбранной в программе цветовой модели;
- преобразование одной цветовой модели в другую;
- разложение представления изображения в выбранной цветовой модели на отдельные каналы с возможностью визуализации одного или всех сразу каналов для заданной цветовой модели.

Поддерживаемые цветовые пространства:

- RGB;
- HSL;
- HSV;
- YCbCr.601;
- YCbCr.709;
- YCoCg;
- CMY.

Входные и выходные файлы везде 8-битные данные и полный диапазон (0..255, PC range).

Цветовое пространство должно выбираться в вашей программе. Если есть загруженное изображение – оно должно преобразовываться при изменении пользователем этого параметра.

При сохранении и загрузке интерпретировать цветные PNM файлы как содержащие данные в текущем цветовом пространстве.

Если выбрано отображение одного канала, то при сохранении должен быть сохранён один этот канал как PNM в оттенках серого.

Обратите внимание: для минимизации ошибок округления и артефактов переполнения, начиная с этого задания, изображение в памяти следует хранить в более точном формате (например, float) и преобразовывать из/в 8-бит на канал только во время загрузки/сохранения/отображения.

Материал по цветовым пространствам был в лекции 1.1:
<https://t.me/c/1714867481/9>

Задание 3 Гамма-коррекция

Цель работы: реализовать поддержку гамма-коррекции.

Описание

Необходимо добавить возможность задания коэффициента гамма-коррекции (неотрицательное вещественное значение) с последующим преобразованием изображения по заданной гамме. Коэффициент равный 0 соответствует гамме sRGB.

Гамма-коррекция в программе должна поддерживаться в следующих режимах:

1. *Преобразование в назначенную гамму* (convert gamma): пересчет гаммы из текущей в то, что ввел пользователь (в пользовательском интерфейсе должна быть возможность ввода значения гаммы). Внешний вид изображения не должен меняться.
2. *Назначение гаммы* (assign gamma): изменить параметр вашей программы, отвечающий за интерпретацию значений пикселей (без изменения самих значений пикселей). В результате изменится внешний вид изображения.

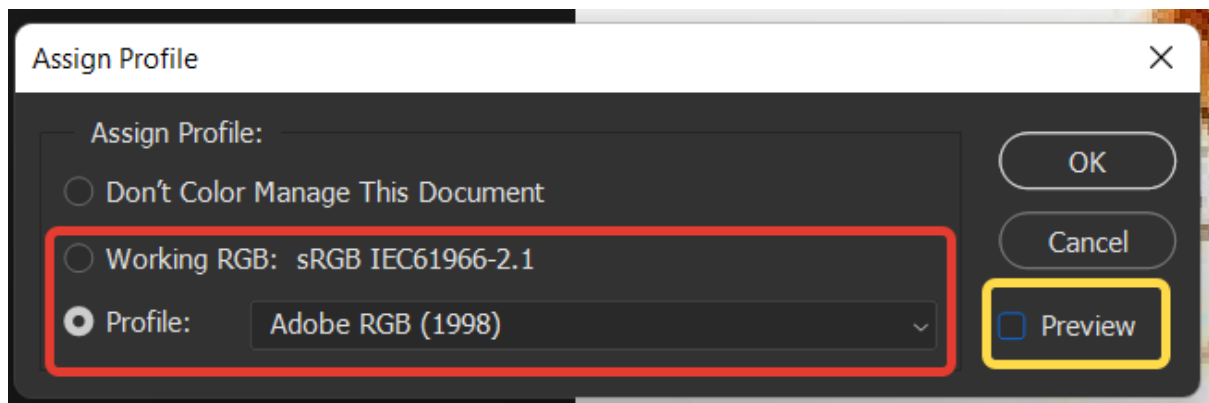
Замечание: Изменение гаммы должно влиять только на видимую яркость, но не на цветовые оттенки изображения.

Считаем, что все входные и выходные изображения имеют гамму sRGB.

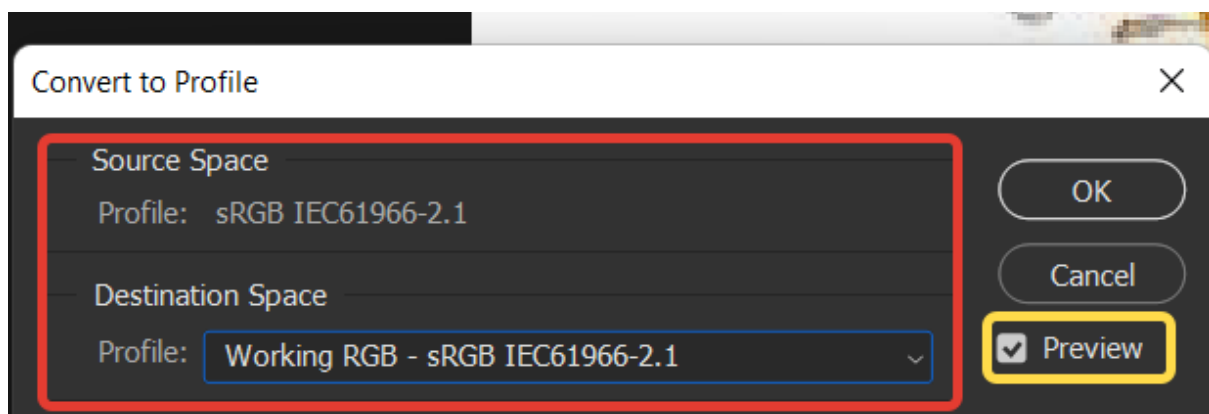
Пример реализации подобного функционала в Photoshop (вкладка

Edit):

- Назначение гаммы (Assign Profile)



- Преобразование в гамму (Convert to Profile)



Выделение красным – обязательно к реализации (только часть программы, не весь функционал цветовых профилей), желтым – по желанию.

Задание 4 Отрисовка растровых линий с применением сглаживания

Цель работы: реализовать функционал, позволяющий рисовать линию на изображении.

Описание

Необходимо построить растровую прямую произвольной толщины, которая рисуется поверх изображения с антиалиасингом (учитывать, что линия может покрывать часть пикселя) и заданной прозрачностью.

Начало и конец линии выбирается пользователем посредством нажатия на пиксели изображения. Также необходимо предусмотреть возможность задания толщины, цвета и степени прозрачности линии пользователем через интерфейс.

Замечание: Если вы реализуете функционал цветовых пространств, то рисование происходит в соответствии с выбранными там параметрами (выбранный цветовой канал, если выбран один; значения цвета интерпретируются в текущем цветовом пространстве).

Задание 5 Псевдотонирование изображений

Цель работы: реализовать функционал псевдотонирования (дизеринга) изображений.

Описание

Необходимо реализовать функционал, применяющий выбранный пользователем алгоритм дизеринга при сохранении изображения, с возможностью предварительного просмотра.

Необходимо предусмотреть возможность задания алгоритма дизеринга пользователем через интерфейс, битность результата дизеринга 1..8 (“битность” здесь означает количество различных уровней в каждом канале изображения, сами значения по-прежнему имеют диапазон 0..255).

Поддерживаемые алгоритмы дизеринга:

- Ordered (8x8);
- Random;
- Floyd-Steinberg;
- Atkinson.

Для упрощения проверки корректности реализации также необходимо реализовать возможность генерации изображения (вместо открытия), содержащего плавный горизонтальный градиент (от 0 слева до максимума справа) заданных размеров во внутренней точности представления изображения в вашей программе (вероятно, float).

Примеры преобразований 2 (random) и 3 (floyd-steinberg) при разных значениях гаммы и битностей: [Dithering_samples](#)

Задание 6 Масштабирование изображений

Цель работы: реализовать поддержку масштабирования изображений.

Описание

Необходимо реализовать функционал, позволяющий пользователю масштабировать загруженное изображение.

Для масштабирования пользователь задает:

- Ширину и высоту результирующего изображения.
- Смещение центра результата относительно центра исходного изображения (X и Y: вещественные числа в единицах результирующего изображения, по умолчанию 0).
- Способ масштабирования:
 - ближайшая точка (метод ближайшего соседа),
 - билинейное,
 - Lanczos3,
 - BC-сплайны (должна быть возможность указать два параметра: B и C, по умолчанию 0 и 0.5 (Catmull-Rom)).

На частичный балл допустимо реализовать:

- результирующее изображение выровнено относительно исходного по левому верхнему краю, без указания смещения,
- масштабирование только на увеличение,
- любые 3 из 4 указанных способов масштабирования.

Подробнее про BC-сплайны:

- <https://de.wikipedia.org/wiki/Mitchell-Netravali-Filter>
- http://mentallandscape.com/Papers_siggraph88.pdf

Задание 7 Формат изображений PNG

Цель работы: реализовать чтение и запись изображений в формате PNG.

Описание

Должны поддерживаться палитровые (только на чтение), серые и цветные изображения (Color Type 3, 0 и 2 соответственно), 8 бит на канал.

Поддержка прозрачности, interlacing и цветовой коррекции (цветовые профили и пр.) не требуется.

Если выполнено задание 3 (гамма-коррекция), то требуется поддержка гаммы (блок gAMA), иначе поддержка не обязательна. При отсутствии этого блока использовать гамму sRGB.

Контрольные суммы при чтении проверять не обязательно, но при записи их следует писать корректно.

Должна быть реализована базовая обработка ошибок с информированием пользователя: файл не удалось открыть или сохранить, формат файла не поддерживается.

Весь функционал необходимо реализовать самостоятельно, без использования сторонних библиотек, за исключением работы с deflate-сжатием: здесь использование сторонних библиотек допустимо. Библиотеки, реализующие deflate:

- [zlib](#) (по ссылке можно найти порты под широкий набор языков программирования)
- [libdeflate](#) (C/C++, C#, Go, Java, Python, Ruby и другие)

- [isa-l](#) (C/C++, могут возникнуть проблемы на Apple M1)

Возможно использование и других библиотек сжатия, по согласованию.

На частичный балл допустимо реализовать только чтение.

Спецификация: [PNG \(Portable Network Graphics\) Specification, Version 1.2](#)

Для просмотра структуры png по блокам (chunks) можно использовать tweakPNG (под Windows) и pngcheck (Unix/macOS).

Задание 8 Формат изображений JPEG

Цель работы: реализовать возможность чтения изображений в формате JPEG (Baseline).

Описание

Поддержка серых и цветных изображений (3 канала), включая поддержку субдискретизации цветности (Chroma subsampling) с билинейной интерполяцией.

Если выполнено задание 2 (Цветовые пространства), то цветные изображения должны декодироваться в родное цветовое пространство (например, в YCbCr.601).

Должна быть реализована базовая обработка ошибок с информированием пользователя: файл не удалось открыть или сохранить, формат файла не поддерживается.

Весь функционал необходимо реализовать самостоятельно, без использования сторонних библиотек.

На частичный балл допустимо реализовать поддержку только серых изображений.

Задание 9 Гистограмма изображения

Цель работы: реализовать отображение гистограммы изображения и алгоритм автоматической коррекции интенсивности.

Описание

Необходимо реализовать визуализацию гистограммы изображения.

Если выбран 1 канал изображения, то показывается одна гистограмма этого канала. В случае отображения всех каналов (3), то показываются 3 отдельных гистограммы по каждому каналу.

Изображение может иметь плохую контрастность: используется не весь диапазон значений, а только его часть, что хорошо видно на гистограмме. Например, если самые тёмные места изображения имеют значение 20 вместо 0. Необходимо реализовать алгоритм автоматической коррекции, которая изменяет значения пикселей таким образом, чтобы получить максимальную контрастность (полный диапазон значений).

Пользователь должен иметь возможность задать долю игнорируемых пикселей (значение в диапазоне $[0; 0.5)$) - часть самых светлых и самых тёмных пикселей, которые следует не учитывать при вычислении минимального и максимального значения в каждом канале. Это необходимо для устойчивой обработки изображения с незначительным шумом.

При обработке нескольких каналов одновременно коэффициенты расширения и сдвига должны быть одинаковые для всех каналов. В частности это означает, что и при обработке в RGB не изменится цветность (оттенки).

Задание 10 Фильтрация изображений

Цель работы: изучить алгоритмы фильтрации и их приложения.

Описание

Необходимо реализовать следующие алгоритмы фильтрации изображений:

№	Фильтр	Параметры, задаваемые через пользовательский интерфейс
1	Пороговая фильтрация (2 класса)	Порог фильтрации в диапазоне (0; 255)
2	Пороговая фильтрация методом Оцу (2 класса)	-
3	Медианный фильтр	Радиус ядра
4	Фильтр Гаусса	Параметр σ (радиус ядра вычисляется по правилу 3σ)
5	Линейный усредняющий фильтр (box blur)	Радиус ядра
6	Фильтр Собеля	-
7	Contrast Adaptive Sharpening (включая better diagonals)	Sharpness [0.0; 1.0]

Радиус ядра – неотрицательные положительные числа. Диаметр ядра вычисляется как $\text{радиус} * 2 + 1$.

Обработка границ: при выходе за границы изображения считать значение равным ближайшему граничному.

На защите нужно подготовить изображения, применение фильтров к которым проиллюстрирует применение этих фильтров.

Про Оцу можно почитать в [документе](#).