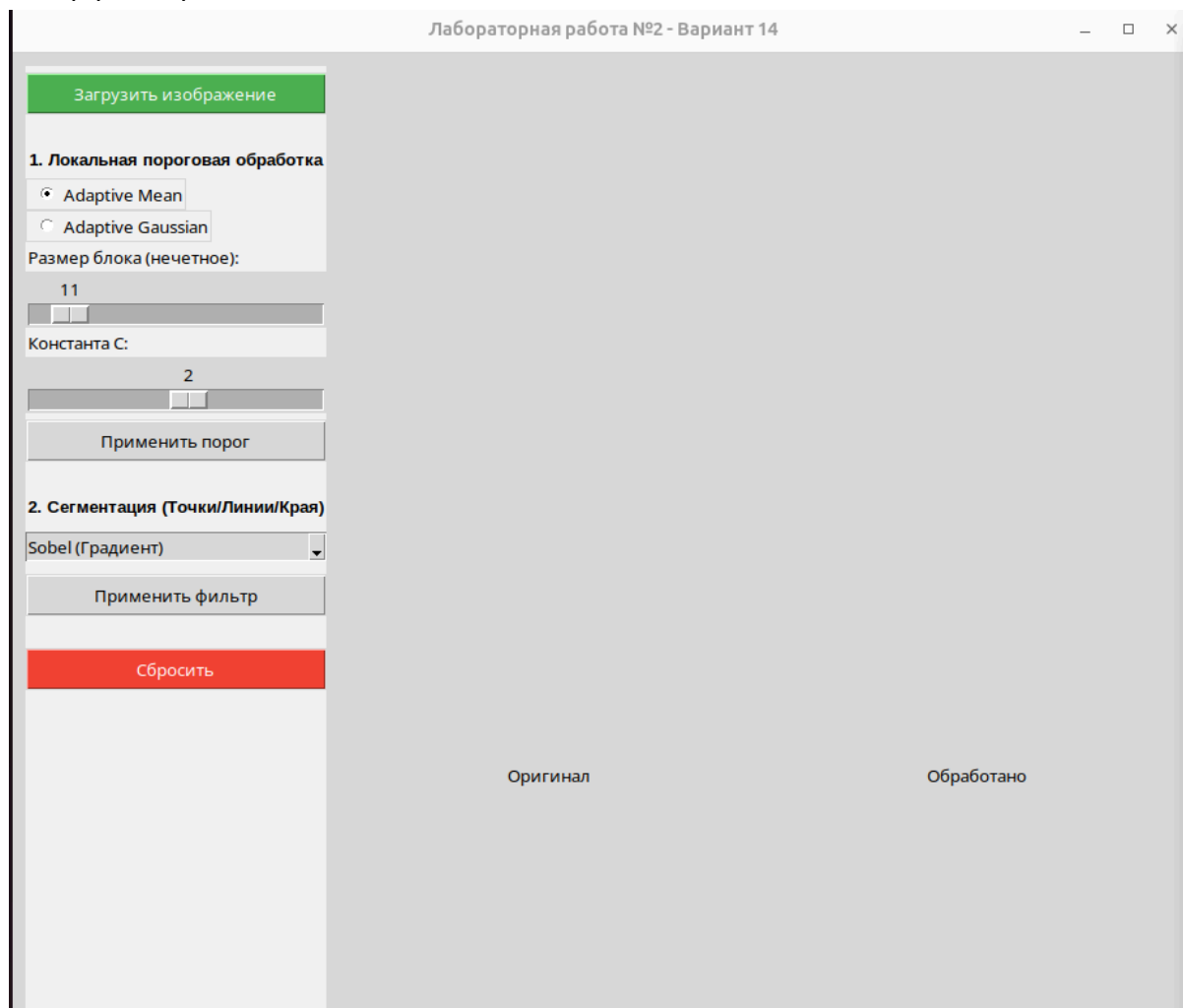


Лабораторная 2. Вариант 14

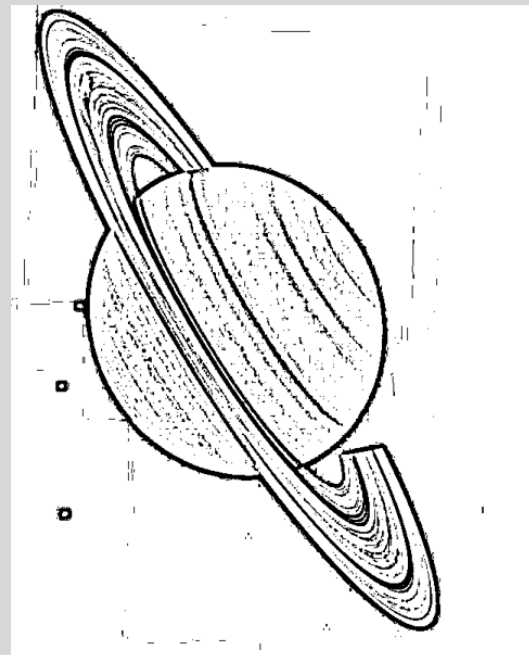
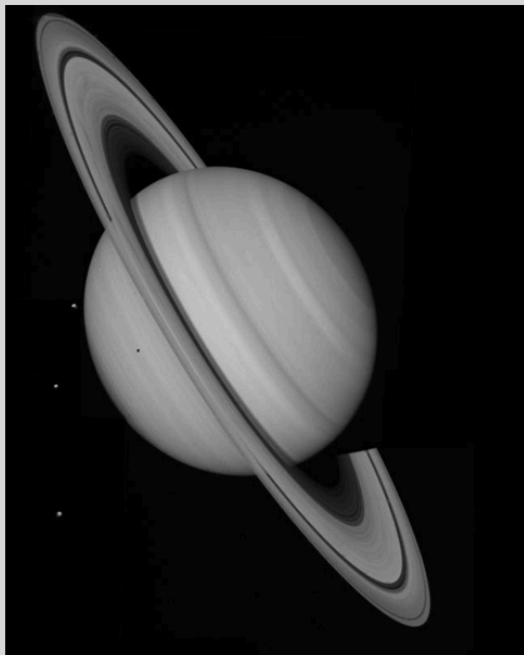
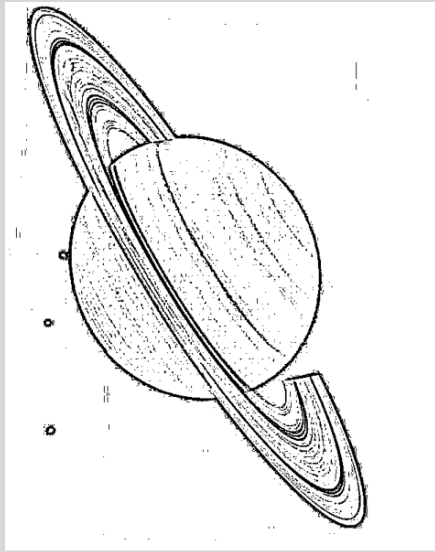
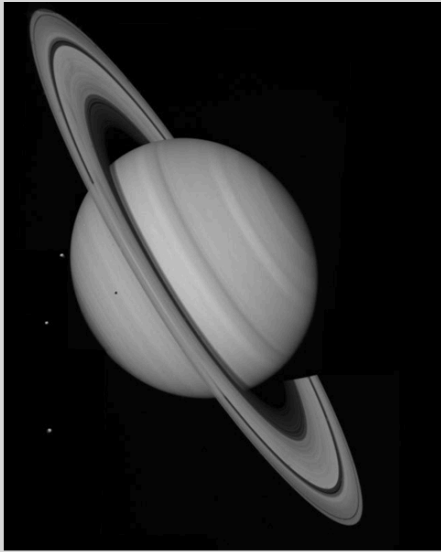
Поликарпов Михаил

Приложение реализовано с помощью библиотеки tkinter для python. Есть возможность загрузить изображение, выбрать вид преобразования, а также настроить константы для пороговой обработки.

Интерфейс приложения:



Для пороговой обработки были выбраны алгоритмы Adaptive Mean и Adaptive Gaussian и библиотека opencv для python.



Оба алгоритма хорошо справляются с задачей пороговой обработки. Однако Adaptive Mean визуально даёт лучшие результаты за счёт того, что на нём почти отсутствует шум от невидимых глазу изменений в чёрном цвете, и сама поверхность планеты менее размыта. Также при более детальном рассмотрении видно, что звёзды в обработанном изображении лучше прорисованы также в первом методе.

Для сегментации были применены алгоритмы из той же библиотеки: Sobel и Canny.



В сегментации с точки зрения выделения границ однозначным лидером является метод Canny. Изображение преобразованное с помощью Sobel получилось размытое, хотя результат работы алгоритма является неплохим. Метод Canny выделил границы очень чётко, полностью очертив границы монет, цифр, текста и арнамента. Однако обработка этим методом гораздо более жёсткая и потери информации явно больше. При детальном рассмотрении видно, что артефактов гораздо больше, граница цифры 2 частично стерлась.

Вывод:

На картинке Сатурна метод Adaptive mean показал себя немного лучше за счёт большей устойчивости к артефактам и резким, неожиданным переходам. При этом оба метода в общем и целом хорошо справляются с задачей.

Алгоритмы сегментации Sobel и Canny, судя по картинке с монеткой могут использоваться для несколько разных задач. Canny очень хорошо выделяет границы, но может оставить артефакты или затереть важную информацию, если она недостаточно выделена. Sobel более консервативен, однако тоже выделяет детали изображения, при этом добавляя небольшую размытость.