Семинар №3
Обработка изображений

Задание №1
Простой детектор границ

Реализация алгоритма:

1. Сглаживание

$$K = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

2. Поиск градиентов

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

3. Пороговая фильтрация

$$G_{lower} < G < G_{upper}$$

Операция свёртки:

$$\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \right) =$$

$$= (a*9) + (b*8) + (c*7) + (d*6) + (e*5) + (f*4) + (g*3) + (h*2) + (i*1)$$

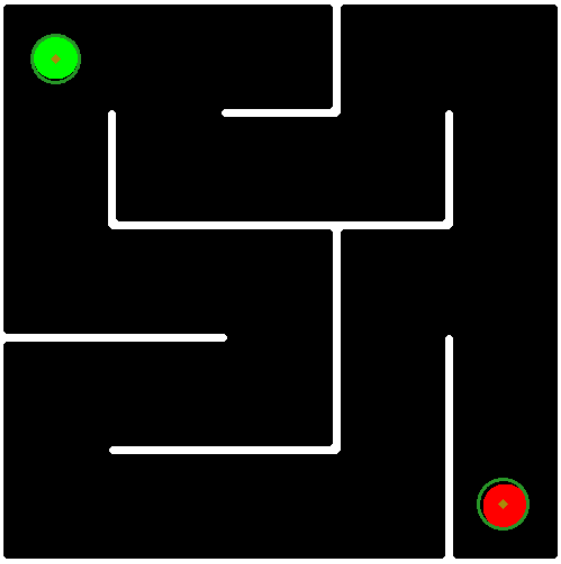
Задание:
Реализовать алгоритм обнаружения границ на изображении с использованием GPU(CUDA).
Для преобразования цветного изображения в массив можно воспользоваться программой img_to_text.py
Для визуализации полученного результата можно воспользоваться программой img_show.py

Задание №2
Поиск пути в лабиринте

Реализовать программу поиска пути в лабиринте.

На вход программы поступает изображение лабиринта. Красным маркером отмечено начало пути, зелёным - конец.

Для обнаружения стартовой и конечной точек можно воспользоваться функцией cv2.HoughCircles (пример circles_detection.py)



В лабиринте могут находиться маркеры синего цвета. Если такие маркеры найдено, то нужно найти путь, содержащий максимальное количество синих маркеров.