

Обработка изображений пламени горелки

Отчет №2 о проделанной работе

Выполнил: Ульмаскулов Эрик Маратович

ФИАН
Москва, 2025

Введение

В работе приведен краткий отчет о проделанных действиях по обработке изображений пламени горелки и уже предобработанных изображений фронта. Основной задачей являлось выбор методики для вычисления самого фронта пламени. Программа написана на языке python.

Изображения фронта

В качестве изображений, с которыми проведена работа, выступили полученные изображения фронта (результат по их получению описан в предыдущем отчете). Для того, чтобы определить, что мы будем подразумевать под фронтом пламени, взглянем на исходное изображение на Рис.1.

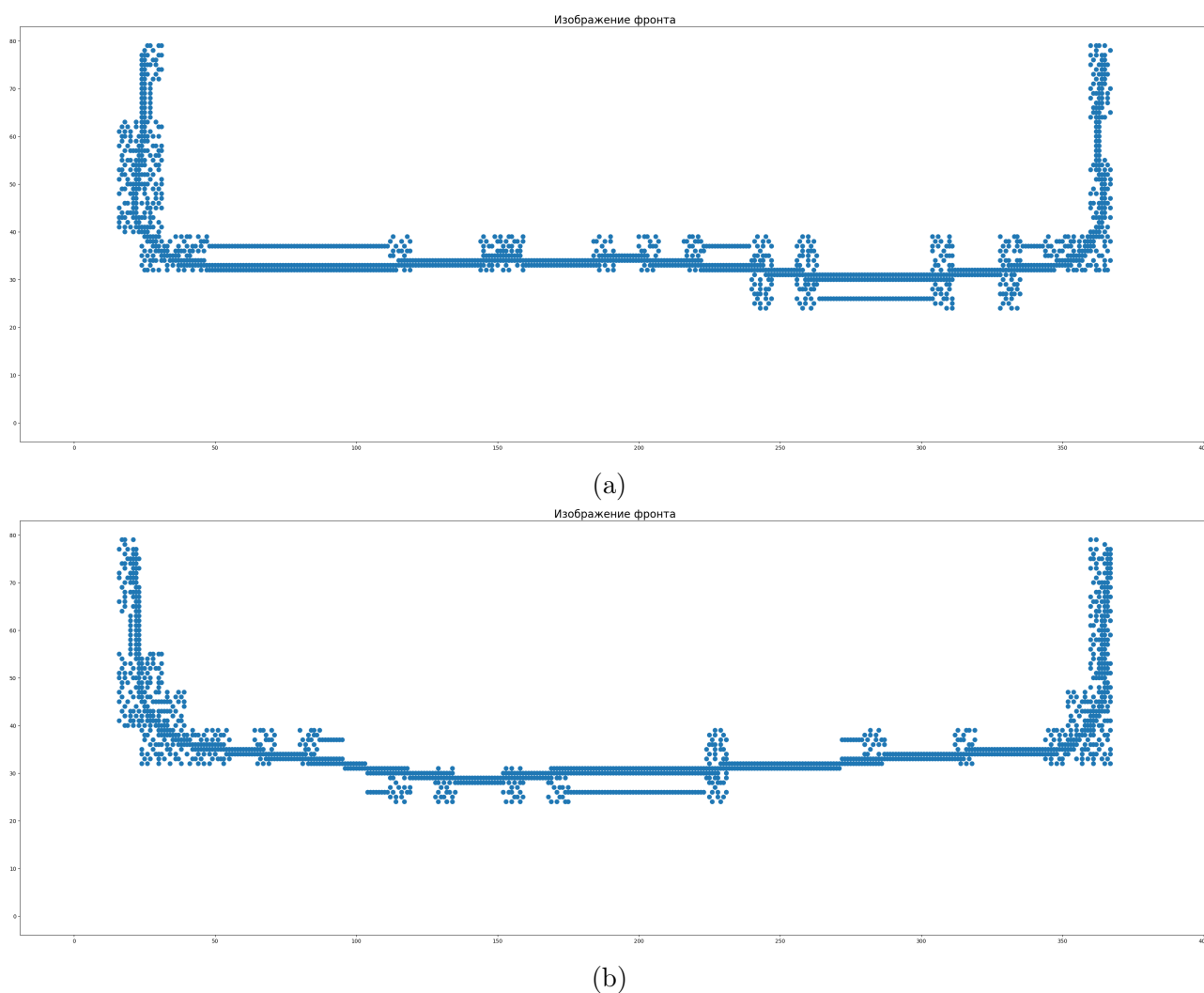


Рис. 1: Изображения фронта в разный момент времени

На рассмотренных изображениях можно заметить толстую линию в два пиксела, которая образует "скелет" фигуры, а так же участки с зашумлением, где хаотично расположены пиксеты. В качестве фронта пламени будем, для начала, подразумевать именно "скелет".

Выделение остова фигуры

Далее в качестве примера будет использовано изображение на Рис.1b. Для того, чтобы убрать шум, для пикселей фронта воспользуемся следующим ядром свертки:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

С его помощью будем определять количество значимых соседей для пиксела. Таким образом, нас будут интересовать следующие пиксели:

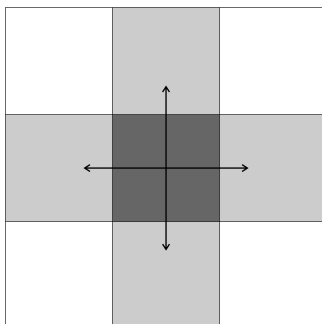


Рис. 2: 4x соседни

Выбирая порог для количества значимых соседей, получим различный результат. В качестве примера рассмотрим сначала порог в 2 значимых соседа. Остальные пиксели, не прошедшие пороговое значение, будут отброшены.

Тогда полученный результат изображен на Рис.3.

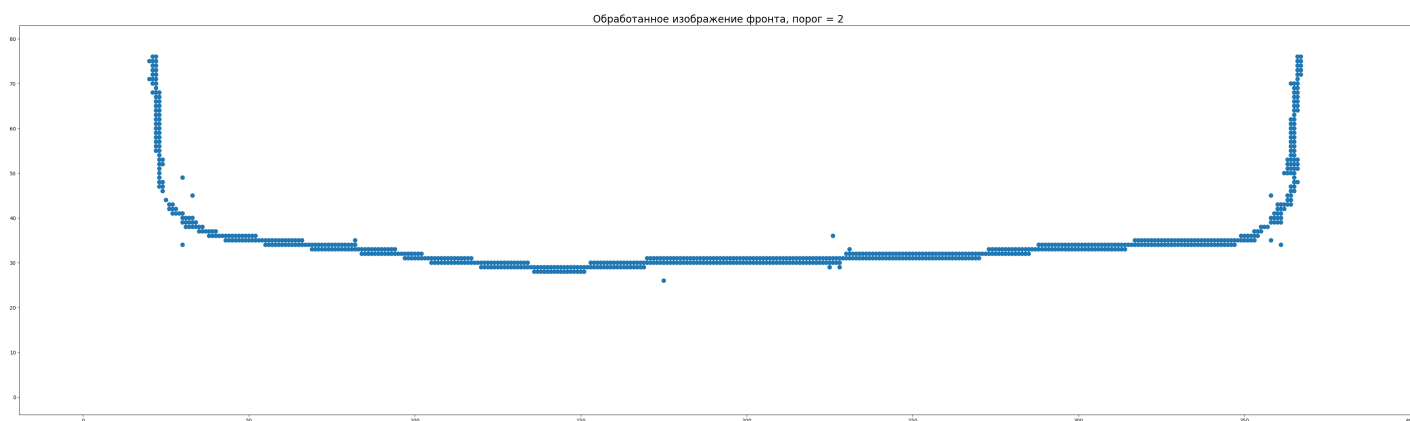


Рис. 3: Порог = 2

Как можно заметить, слева на изображении фронт обрывается. Т.е. в таком случае нет возможности найти длину фронта, т.к. рассматриваемая область более не является связной.

Теперь рассмотрим порог равный 1. Тогда получим результат как на Рис.4.

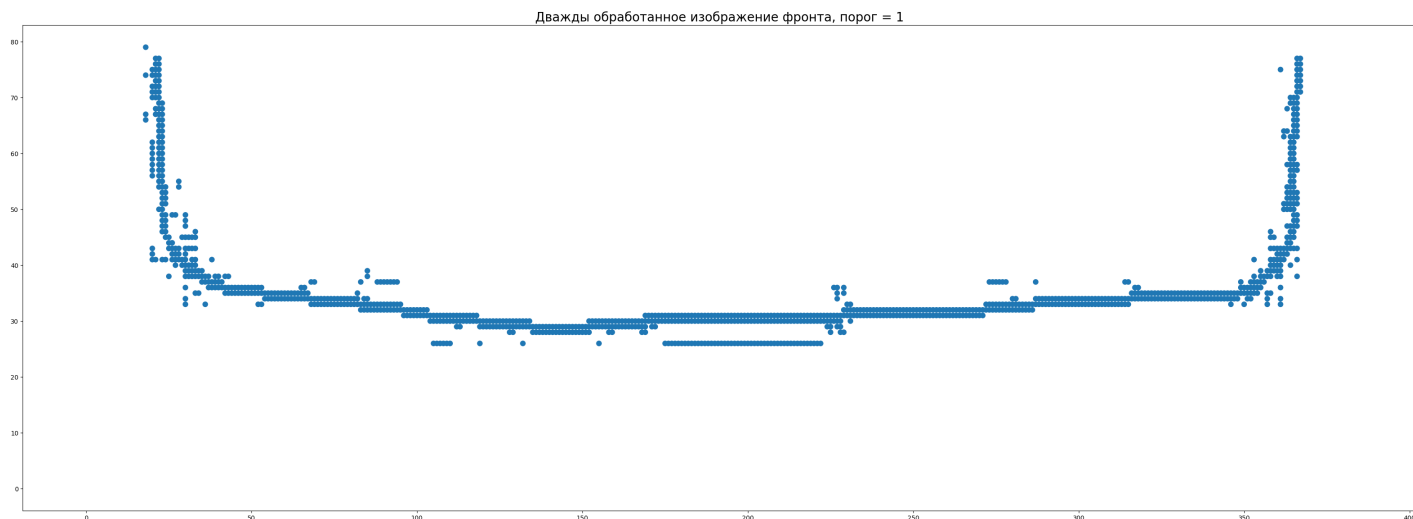
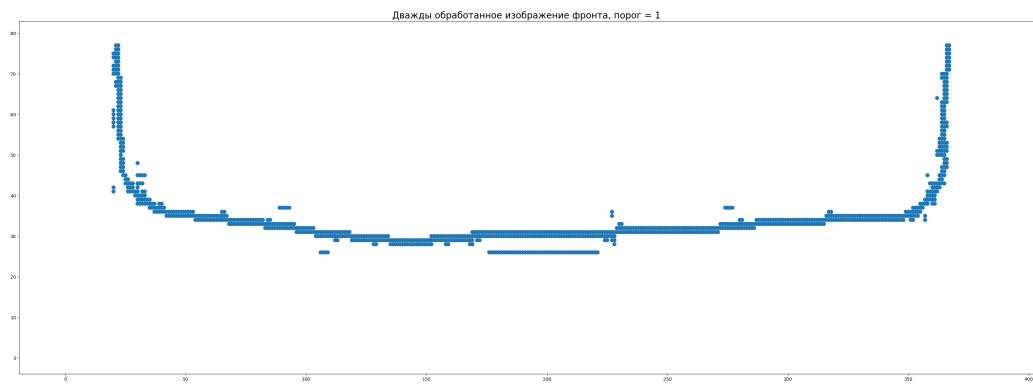


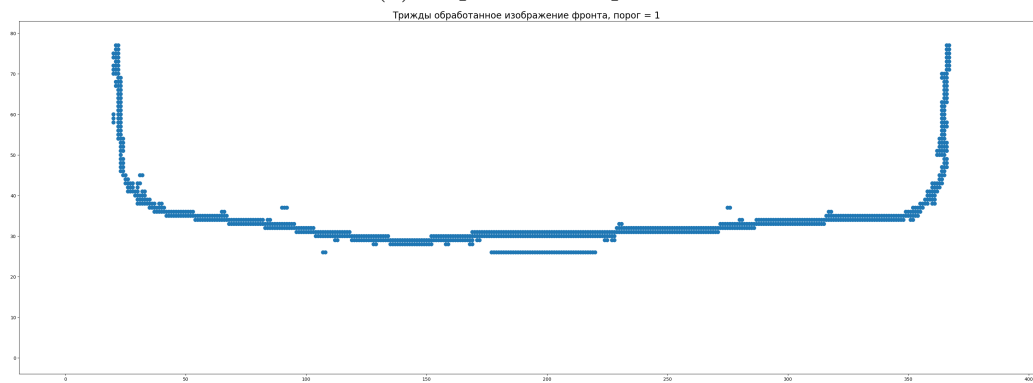
Рис. 4: Порог = 1

Можно заметить, что большинство шумов пропало, осталась, в основном, главная часть фронта. Проводя несколько таких фильтраций, можно уменьшить количество пикселей, не относящихся к исследуемой области.

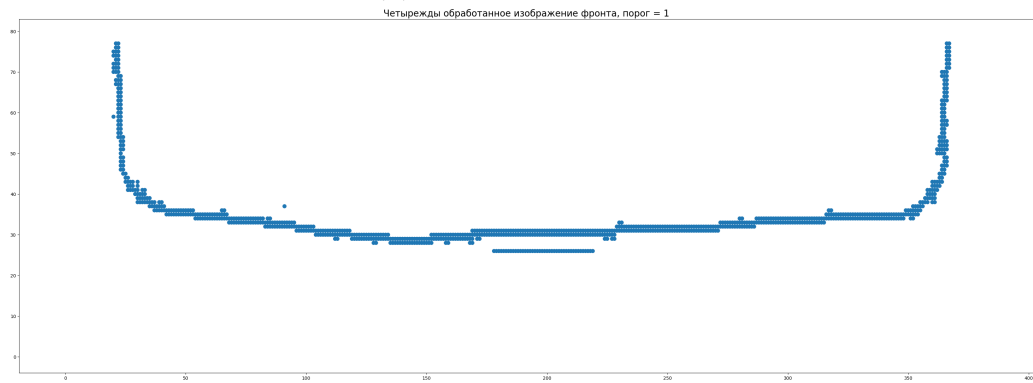
Рассмотрим результаты при последовательной повторяющейся обработке изображения вышеуказанным способом(на следующей странице):



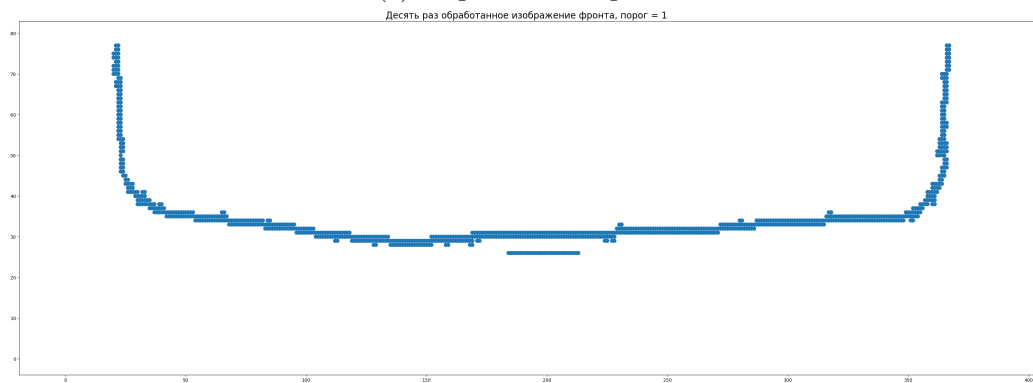
(a) Обработано 2 раза



(b) Обработано 3 раза



(c) Обработано 4 раза



(d) Обработано 10 раз

Рис. 5: Попрог = 1

Как можно заметить, на Рис.5 все "шумовые" пиксели пропали, а протяженные последовательности толщиной в 1 пикселе, не относящиеся к фронту, с каждым разом уменьшаются в длине стягиваясь в точку. Таким образом 4 раза уже достаточно для того, чтобы основная связная область содержала только пиксели фронта.

Вычисление длины фронта

Теперь, полностью выделив пиксели фронта на изображении, займемся вопросом о нахождении его длины. В качестве длины фронта выберем внешний контур. Начальной и конечной точкой будут левый и правый самые верхние пиксели соответственно. Алгоритм нахождения будет следующим: будем запоминать предыдущее направление движения обхода по фронту (для простоты назовем это направление "вперед"). В зависимости от него, будем брать пиксеты в следующем приоритете: направо, направо-вперед, вперед, влево-вперед и т.д. как на Рис.6.

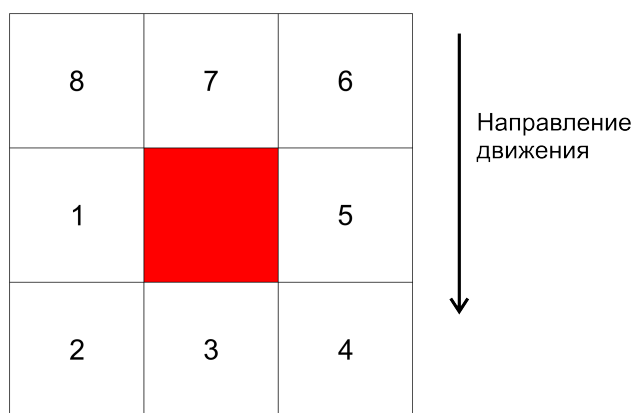
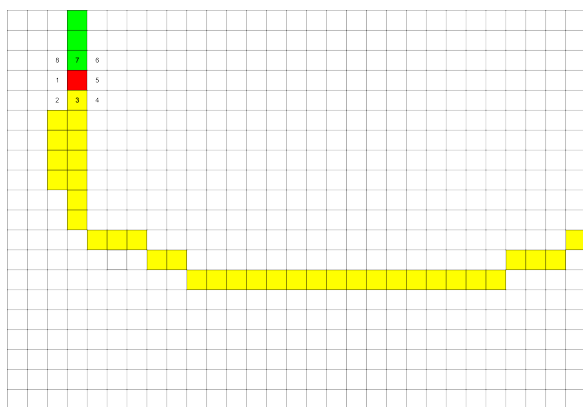


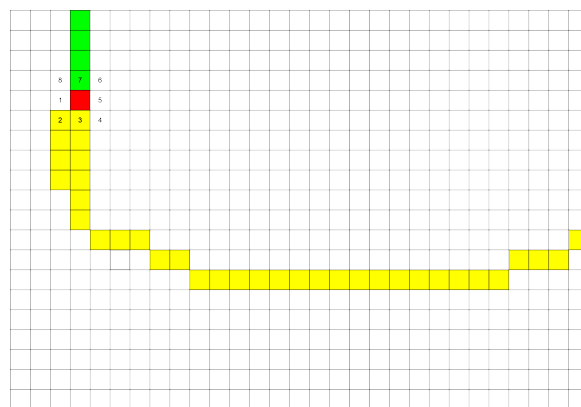
Рис. 6: Приоритет обхода пикселей при данном направлении движения

Причем, если движение происходит по диагонали, то направление движения мы не будем изменять, и оно будет оставаться прежним.

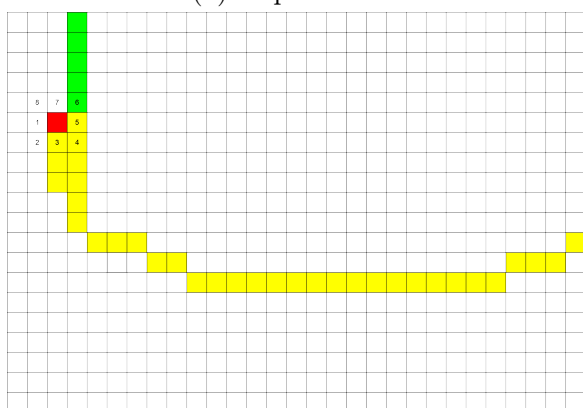
Рассмотрим результат движения шаблона (Рис.6) на следующей последовательности изображений:



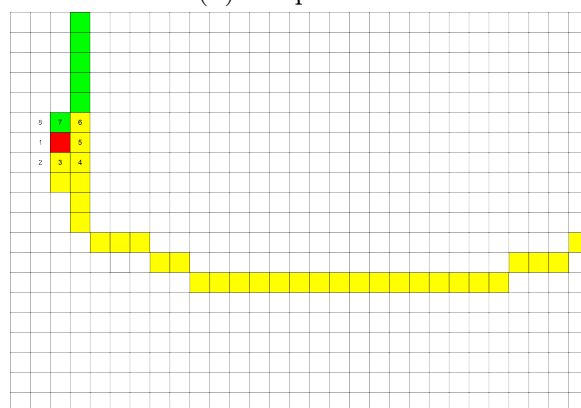
(a) Первый шаг



(b) Второй шаг



(c) Третий шаг

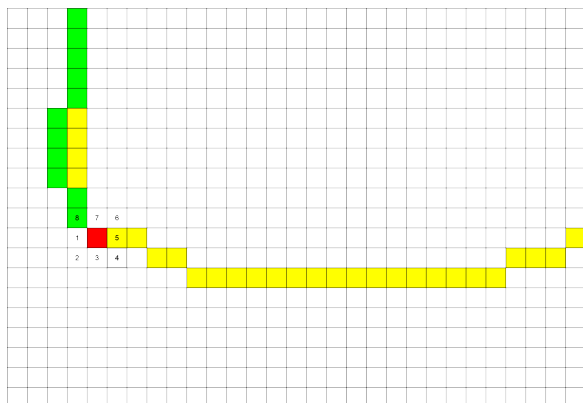


(d) Четвертый шаг

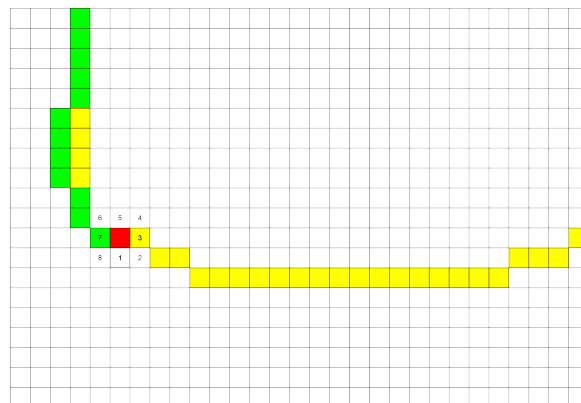
Рис. 7: Пример движения шаблона. Желтые пиксели относятся к пикселеам фронта; зеленые пиксели - пиксели, выбранные для вычисления фронта; красный пиксел - текущий пиксле

Как можно заметить, на Рис. 7с и Рис. 7d приоритет выбора следующего пиксела не изменился. Это иллюстрирует приятую договоренность о том, что при движении по диагонали приоритет движения не меняется.

На Рис.8 же показан процесс смены приоритета.



(a) Первый шаг



(b) Второй шаг

Рис. 8: Пример смены приоритета при движении шаблона.

Стоит обратить внимание на то, что на Рис. 8a движения происходит по диагонали и направление движения сохраняется, а на Рис.8b направление движения уже изменилось.

Используя вышеописанный алгоритм, проведем вычисления для одного из изображений фронта. Для большей точности проведем проверку на изображении обработанном шаблоном Рис. 2 с порогом в 1 значимого соседа. На Рис. 9. приведен результат.

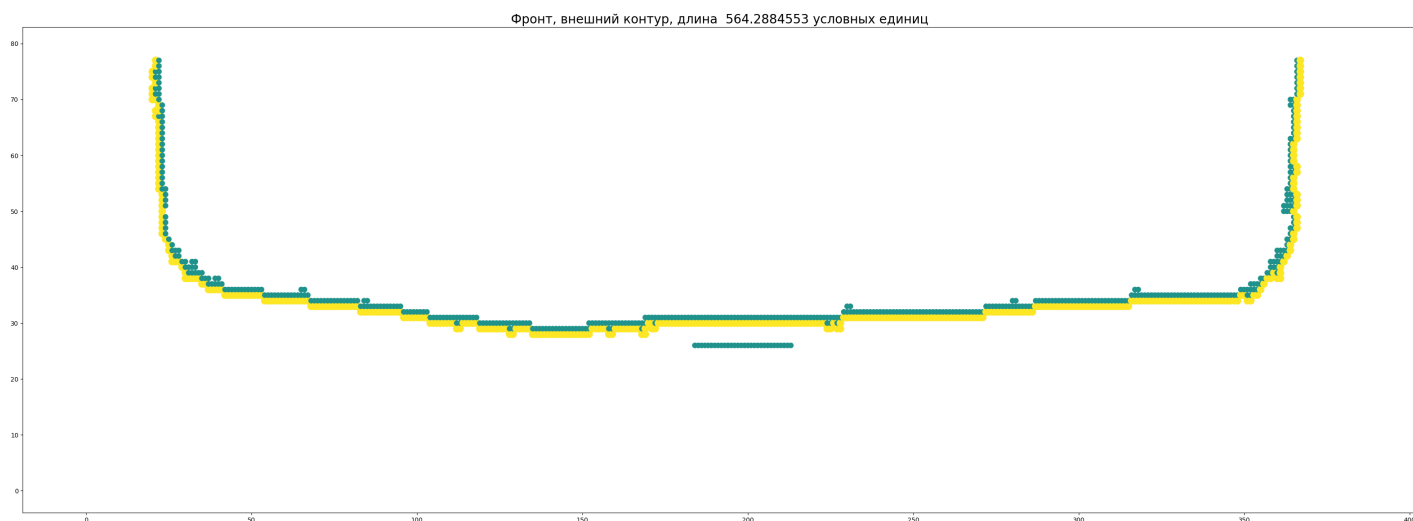


Рис. 9: Пример работы алгоритма

Важно: на Рис. 9 указана длина полученного фронта - 564.2884553. Эта длина указана в **УСЛОВНЫХ ЕДИНИЦАХ**. Чтобы перевести ее в миллиметры, необходимо в программе в функции для определения длины указать множитель, который соответствует длине пиксела в миллиметрах.

Заключение

В работе описаны методы финальной обработки изображений для выделения фронта и последующего вычисления его длины. Благодаря этому имеется возможность обучить нейронную сеть для выполнения этой задачи(скорее в исследовательских целях, чтобы узнать точность получаемых от нее результатов).