**Сравнительный анализ алгоритмов заполнения многоугольников и заливки области**

Заливка изображений — часто нужная на практике задача, суть которой — заполнить некоторую область изображения, ограниченную контуром, заданным цветом. Помогут выполнить данную задачу известные многим алгоритмы.

В этом анализе будет рассмотрено три группы алгоритмов:

* Алгоритм закраски с затравкой
* Алгоритмы со списком рёберных точек
* Алгоритмы XOR

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Описание | Плюсы | Минусы |
| Алгоритм закраски с затравкой | Суть метода заключается в следующем: мы берём затравочную точку и закрашиваем её. Для каждого незакрашенного соседа мы выполняем аналогичную процедуру.  | С помощью этого алгоритма можно закрашивать любые замкнутые области. | Медленный и расходует много памяти для стека |
| Алгоритмы со списком рёберных точек | Подходит только для тех случаев, когда закрашиваемая область может быть задана в виде многоугольника | * Оптимизированный по расходу памяти
* Каждый пиксель обрабатывается строго один раз
 | Доступ к видео-памяти занимает сравнительно много времени. |
| Построчная XOR обработка | После того, как границы построены, закрашивание сводится к заполнению в каждой строке промежутков между двумя закрашенными точками. | Преимущество этого алгоритма в его предельной простоте и высокой скорости | Недостаток в том, что алгоритм не может работать, при наличии посторонних изображений. |
| Алгоритм XOR для граней с перегородкой | Идея заключается в том, чтобы инвертировать область не между ребром и границей экрана, а между ребром и специальной вертикальной линий (т.н. перегородкой). Чаще всего перегородка проводится так, чтобы она пересекала многоугольник | К достоинствам данного алгоритма можно отнести его простоту и оригинальность, а также отсутствие дополнительных структур данных | Недостатком является необходимость выполнения большого числа операций с пикселями (до N операций с каждым пикселем), в том числе и вне многоугольника. В частности, чем больше расстояние между многоугольником и правой границей экрана, тем больше будет совершено "лишних" операций. |

**Заключение**

Каждая группа алгоритмов имеет свои достоинства и недостатки, также и особенности использования. Используйте ту группу алгоритмов для которая наиболее подходит для поставленной цели, также стоит сказать, что для каждой ситуации нужно думать над реализацией, опираясь на задачу, которую нужно решить, тем самым сократив время на разработку приложений.