

# Polynomial graphical approximation algorithm

---

## Спецификация графического формата PGA

Версия 1.0, 5 июля 2016г.  
Компания Q-Dex.

Формат PGA разрабатывался для хранения графических файлов, с последующим их применением на устройствах, основанных на микроконтроллерах с низкой и средней производительностью. Формат использует алгоритм сжатия с потерей качества, основанный на аппроксимации цветовых компонент растров.

### Содержание

Общие сведения .....	2
Описание заголовка файла.....	2
Принцип сжатия .....	3
Структура данных .....	4
Оценка эффективности.....	5

## Общие сведения

Формат PGA предназначен для сжатия растровых графических изображений. Изображение представляет собой набор квадратных блоков, каждый блок представляет собой набор линий, изменение цвета в которых описывается полиномиальной функцией (обычно линейной, либо квадратичной, однако стандарт допускает и старшие степени до 255). Алгоритм сжимает изображение с потерей качества, качество зависит от размеров блока (чем меньше, тем лучше) и от степени полиномов (чем больше, тем лучше). Максимальный размер изображения составляет 65535x65535. Общее описание стандарта допускает максимальный размер блока в 255 пикселей, однако все приложения, разработанные в Q-Dex, имеют ограничение на размер блока в 25 пикселей, связано это с нецелесообразностью дальнейшего увеличения размеров блока из-за значительной потери качества.

Преимуществами PGA по сравнению с другими форматами являются довольно эффективное сжатие графического файла и довольно простой алгоритм восстановления изображения. Эти преимущества позволяют эффективно использовать такой формат изображений на микроконтроллерах с невысокой производительностью.

## Описание заголовка файла

Заголовок PGA файла имеет размер в 18 байт, его структура приведена в таблице.

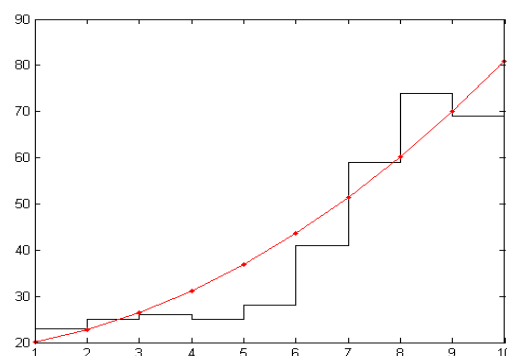
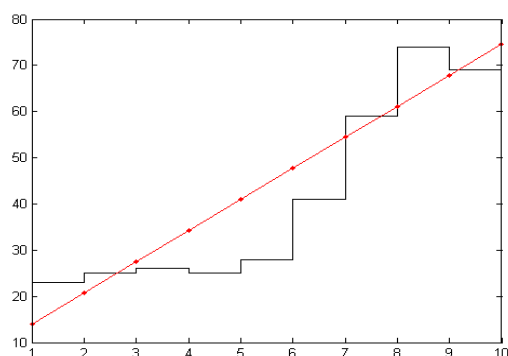
Поле	Смещение	Размер	Описание
ID	0	2	Идентификатор = "PG"
Width	2	2	Ширина изображения в пикселях
Height	4	2	Высота изображения в пикселях
BlockCountX	6	2	Количество блоков в ширину
BlockCountY	8	2	Количество блоков в высоту
BlockSize	10	1	Размер блока в пикселях
ApproxSize	12	1	Степень полинома
Reserved	14	6	Поле зарезервировано.

Для того чтобы идентифицировать формат PGA достаточно прочитать первые два байта, они должны содержать значение "PG".

## Принцип сжатия

Алгоритм представляет растровое изображение в виде набора квадратных блоков. Размер блоков выбирается при сжатии и записывается в заголовок файла. Также алгоритм на основе размера блоков вычисляет их количество по вертикали и горизонтали и заполняет соответствующие поля заголовка.

Идея метода основана на том, что любой набор численных значений можно описать некоторой функцией, значения которой будут близки к исходным численным значениям из набора. Записав эту функцию вместо самого набора значений можно сэкономить место на носителе, однако это справедливо только в том случае если описание функции состоит из меньшего количества значений, чем исходный набор. Формат версии 1.0 использует функции вида:  $f(x) = ax + b$  и  $f(x) = ax^2 + bx + c$ . Для описания первого полинома нужно два значения, а для второго три, соответственно при аппроксимации первой функцией размер блока должен быть не меньше 3 пикселей, а для второй не менее 4 пикселей. Аппроксимация такими функциями показана на рисунках ниже:



Алгоритм осуществляет аппроксимацию методом наименьших квадратов. Суть метода заключается в минимизации среднеквадратичной ошибки:

$$S = \sum_{i=1}^n (y_i - f(x_i))^2 \quad \text{для каждой линии внутри блока. Каждая цветовая}$$

плоскость анализируется независимо, таким образом, для каждого блока процесс сжатия выполняется трижды. Красная и синяя цветовые плоскости сжимаются вдоль горизонтальных линий, а зеленая вдоль вертикальной. Это связано с тем, что зеленая компонента обладает большей яркостью, и направив ее аппроксимированные линии перпендикулярно линиям других компонент можно минимизировать эффект смазанности цветов вдоль аппроксимированных линий.

## Структура данных

После заголовка PGA-файла в файле содержатся описания аппроксимирующих функций. Элементы записаны по блокам, блоки записываются сначала по горизонтали, затем по вертикали. В блоке цветовые плоскости записаны в следующей последовательности: красная, зеленая, синяя. Функции записаны по порядку начиная с первой линии в блоке. Каждый элемент аппроксимирующей функции имеет размер 1 байт.

Следует заметить, что для квадратичной функции значения коэффициентов могут превысить отметку в 2000. Это делает невозможным запись таких коэффициентов в 1 байт. Поэтому вместо самих коэффициентов формат PGA записывает в файл три точки параболы, первую в начале блока, вторую в середине, а третью в конце. Так как любую параболу однозначно можно описать тремя точками, это делает возможным последующее восстановление коэффициентов полинома второй степени. Для линейной функции свободный коэффициент имеет целочисленный формат, а для коэффициента при переменном формате с фиксированной запятой: один байт на знак, 5 байт на целую часть, 2 байта на дробную.

Алгоритм загрузки красной компоненты блока, независимо от степени аппроксимации, имеет следующий вид:

```
for j:= 1 to BlockSize do
begin
  for i:=1 to ApproxSize do
    BlockRead(f,RedB[i,j], 1);
end;
```

Чтение других компонент блока выполняется аналогично.

Процедура восстановления коэффициентов для квадратичной функции приведена ниже:

```
For i:=1 to ApproxSize do y[i]:=RedB[i,j];
a := ( 2/BlockSize * ( (y[3]-y[1])/(BlockSize-1) - 2*(y[2]-y[1])/(BlockSizeExtraX-2) ) );
b := ( 2*(y[2]-y[1])/(BlockSize-2) - (BlockSize/2+1) *x3 );
c := ( y[1] - x2 - x3 );
for x:=1 to BlockSize do
RedC := round( a*x*x + b*x + c );
```

## Оценка эффективности

Размер PGA зависит от размера блока, степени аппроксимации и размера исходного файла. Соотношение размеров исходного несжатого 24-х битного изображения к PGA файлу описывается соотношением:

$$\Delta = \frac{ApproxSize}{BlockSize}.$$

Так как метод сжатия основан на усреднении то алгоритм будет более эффективен для изображений с плавным переходом цвета, а разница между сжатым и исходным изображением будет наиболее заметна в местах резких цветовых переходов. Особенно сильно погрешность будет заметна при сохранении изображений содержащих текст.