

## Содержание

Введение . . . . .	4
1 Аналитический раздел . . . . .	5
1.1 Случайный дизеринг . . . . .	5
1.2 Шаблонный дизеринг . . . . .	5
1.3 Упорядоченный дизеринг . . . . .	6
1.4 Дизеринг при помощи диффузии ошибок . . . . .	7
1.5 Вариации алгоритма дизернга при помощи диффузии ошибок . . . . .	7
1.5.1 Фильтр(маска??) Флойда-Стейнберга . . . . .	7
1.5.2 "Ложный"фильтр Флойда-Стейнберга . . . . .	7
1.5.3 Фильтр Джарвиса,Джунка и Нинка . . . . .	7
1.5.4 Фильтр Стаки . . . . .	8
1.5.5 Фильтр Бурка . . . . .	8
2 Конструкторский раздел . . . . .	9
3 Технологический раздел . . . . .	10
4 Экспериментальный раздел . . . . .	11
5 Организационно-экономический раздел . . . . .	12
6 Промышленная экология и безопасность . . . . .	13
Заключение . . . . .	14
Список использованных источников . . . . .	15
А Картинки . . . . .	16
Б Еще картинки . . . . .	17

## Глоссарий

**Artifacts** — Phenomena like contouring, which are not present in the source image but produced by the digital signal processing, are called artifacts. The most common type of artifact is the Moire' pattern. If you display or print an image of several lines, closely spaced and radiating from a single point, you will see what appear to be flower-like patterns. These are not part of the original image but are an illusion produced by the jaggedness of the display.[1] Error Noise

Returning to our fixed-thresholded (and badly-rendered) image, how could we document what has taken place to make this image so inaccurate? Expressing it in technical terms, a relatively large amount of error "noise" is present in the fixed-thresholded image. The error value is the difference between the image's original intensity at a given dot and the intensity of the displayed dot. Obviously, very dark values like 1 or 2 (which are almost full black) incur very small errors when they are rendered as a 0 value (black) dot. On the other hand, a gross error is incurred when a 129 value dot (a medium gray) is displayed at 255 value (white), for instance.[1]

### Fixed Thresholding

A good place to start is with the example of performing a simple (or fixed) thresholding operation on our grayscale image in order to display it on our black and white device. This is accomplished by establishing a demarcation point, or threshold, at the 50% gray level. Each dot of the source image is compared against this threshold value: if it is darker than the value, the device plots it black, and if it's lighter, the device plots it white.

What happens to the image during this operation? Well, some detail survives, but our perception of gray levels is completely gone. This means that a lot of the image content is obliterated. Take an area of the image which is made up of various gray shades in the range of 60-90%. After fixed thresholding, all of those shades (being darker than the 50% gray threshold) will be mapped to solid black. So much for variations of intensity.

Another portion of the image might show an object with an increasing, diffused shadow across one of its surfaces, with gray shades in the range of 20-70%. This gradual variation in intensity will be lost in fixed thresholding, giving way to two separate areas (one white, one black) and a distinct, visible boundary between them. The situation where a transition from one intensity or shade to another is very conspicuous is known as contouring.[1]

ШУМЫ(нужно тоже дать определение)

## Обозначения и сокращения

- ⊗ — Искомый пиксель
- — Пустой пиксель
- — Закрашенный пиксель
- ⊞ — Произвольный пиксель

## Введение

Обычно изображения, хранимые в цифровом виде, представляются как массив из значений атрибутов; при этом для представления полноцветных фотографий используется диапазон из нескольких миллионов значений на каждый атрибут. Но часто количество выводимых отображающим устройством оттенков ограничено. Если графическое устройство не способно воссоздавать достаточное количество цветов, тогда используют растривание — независимо от того, растровое это устройство или нерастровое. В полиграфии растривание известно давно. Оно использовалось несколько столетий тому назад для печати гравюр. В гравюрах изображение создается многими штрихами, причем полутонные градации представляются или штрихами разной толщины на одинаковом расстоянии, или штрихами одинаковой толщины с переменной густотой расположения. Такие способы используют особенности человеческого зрения и в первую очередь — пространственную интеграцию. Если достаточно близко расположить маленькие точки разных цветов, то они будут восприниматься как одна точка с некоторым усредненным цветом. Если на плоскости густо расположить много маленьких разноцветных точек, то будет создана визуальная иллюзия закрашивания плоскости определенным усредненным цветом. Однако, если увеличивать размеры точек и (или) расстояние между ними, то иллюзия сплошного закрашивания исчезает — включается другая система человеческого зрения, которая обеспечивает способность различать объекты, подчеркивать контуры. В компьютерных графических системах часто используют эти методы. Они позволяют увеличить количество оттенков цветов за счет снижения пространственного разрешения растрового изображения (иначе говоря — это обмен разрешающей способности на количество цветов) или подмешивание в исходное изображение случайного шума. В литературе по КГ такие методы растривания получили название dithering (разрежение, дрожание).

## 1 Аналитический раздел

Алгоритмы дизеринга подразделяются на следующие категории:

- Случайный дизеринг(Random dither);
- Шаблонный дизеринг(Patterning);
- Упорядоченный дизеринг(Ordered);
- Дизеринг рассеивания ошибок(Error-diffusion).

Каждый следующий метод лучше по качеству предыдущего, однако, такие соображения как время, экономия памяти и прочие могут склонить наш выбор в пользу более простого алгоритма. Для того, чтобы применить любой из первых трех методов в цвете, нужно просто применить алгоритм отдельно для каждого цвета отдельно и смешать полученные значения. Для простоты описания, будем рассказывать только о черно-белых изображениях.

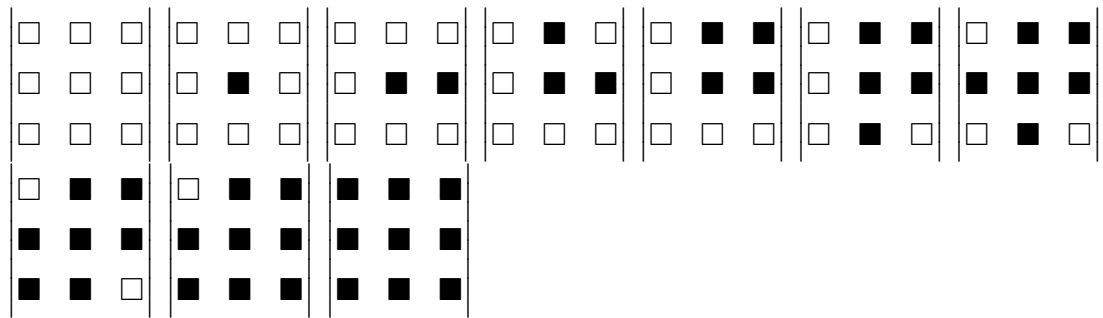
### 1.1 Случайный дизеринг

Этот алгоритм тривиален. По сравнению с остальными алгоритмами, его качество слишком низко, поэтому он применяется лишь там, где необходима высокая скорость работы в ущерб качеству. Для каждой точки в нашем черно-белом изображении мы генерируем случайное число в диапазоне 0-255: если случайное число больше, чем значение в данной точке, то отображаем белую точку, иначе отображаем черную точку. Это создает изображение с большим количеством «белого шума», который выглядит как «снег». Хотя изображение выглядит неточным и зернистым, оно не содержит артефактов. К слову, это метод дизеринга полезен при воспроизведении низкокачественных изображений, где отсутствие артефактов более важно, чем наличие шумов. Например, изображение содержит градиент всех уровней от черного до белого. Это изображение будет отлично выглядеть после того, как к нему применят случайный дизеринг, а остальные методы дизеринга приведут к возникновению артефактов.

### 1.2 Шаблонный дизеринг

Шаблонный дизеринг подразумевает то, что мы увеличиваем разрешение изображения для того, чтобы оно смотрелось более красиво для человеческого глаза. Так же, как и случайный дизеринг, это простой алгоритм, но он гораздо более эффективен. Для каждой точки изображения мы генерируем «шаблон» пикселей, который аппроксимирует эту точку. Тем самым мы можем имитировать больший набор оттенков, чем поддерживает наша глубина цвета. Например, шаблон 3x3. Он имеет 512 вариантов возможных расположений пикселей, но нас интересует только интенсивность цвета. Она формируется на основе количества черных пикселей,

содержащихся в шаблоне. То есть возможных вариантов 10.



### 1.3 Упорядоченный дизеринг

Значительным недостатком шаблонного дизеринга является пространственное увеличение картинки (и увеличение её разрешения). Упорядоченный дизеринг позволяет избежать этого пространственного искажения. Очевидно, что для того, чтобы достичь этого, каждая точка в исходном изображении должна быть сопоставлена с пикселем на конечном изображении один-к-одному. Существуют два вида упорядоченного дизеринга: кластерный и дисперсный. Суть этих методов заключается в том, что мы разбиваем исходное изображение на квадраты пикселей и значения маски в каждой точке квадрата выступают в роли порогового значения. Если значение пикселя (отмасштабированное под интервал маски) в данной точке больше значения маски, то красим пиксель в черный цвет, иначе в белый. Кластерный паттерны выглядят вот так:

8	3	4	1	7	4
6	1	2	5	8	3
7	5	9	6	2	9

Кластерные паттерны применяются в случаях когда понятие «конкретный пиксель» у устройства вывода информации отсутствует (ЭЛТ-мониторы подобное). Во многих исследованиях [ссылка на Байера] было отмечено, что если устройство вывода позволяет применить дисперсный метод, то его применение является предпочтительным. Так же Байер [2] показал, что для матриц порядков степени двух существует оптимальная структура дисперсных точек, которая приводит к наименьшему количеству шумов (для матрицы 2x2 и 4x4 соответственно):

1	3
4	2

1	9	3	11
13	5	15	7
4	12	2	10
16	8	14	6

В конструкторской части показан рекурсивный метод для создания

масок большего размера. Основным недостатком данного метода считается то, что в результате его работы формируется большое количество артефактов.

## 1.4 Дизайн при помощи диффузии ошибок

Наиболее качественным методом является метод рассеивания ошибок. Но так же он, к сожалению, самый медленный. Существуют несколько вариантов этого алгоритма, причем скорость алгоритма обратно пропорционально качеству изображения. Суть алгоритма: для каждой точки изображения находим ближайший возможный цвет. Затем мы рассчитываем разницу между текущим значением и ближайшим возможным. Эта разница и будем нашим значением ошибки. Это значение ошибки мы распределяем между соседними элементами, которые мы ещё не посещали. Для последних точек ошибка распределяется между уже посещенными точками.

## 1.5 Вариации алгоритма дизайна при помощи диффузии ошибок

Линия сканирования движется слева-направо. Когда линия сканирования доходит до конца горизонтальной строки пикселей, переходим с первого пикселя следующей строки и повторяем необходимые действия.

*Примечание: числа на схемах - это доли от значения ошибки. Например,  $7/16$  на схеме выглядит как 7. То есть 7 обозначает некую величину, равную значению ошибки  $\times 7/16$*

### 1.5.1 Фильтр(маска??) Флойда-Стейнберга

Каждый пиксель распределяет свою ошибку Флойд и Стейнберг тщательно выбрал этот подбирали коэффициенты таким образом, что в районах с интенсивностью  $1/2$  от общего количество оттенков, изображение выглядело похожим на шахматную доску.

$R \times 7$

3 5 1 ( $1/16$ )

### 1.5.2 "Ложный" фильтр Флойда-Стейнберга

В случае сканирования слева-направо этот фильтр порождает большое количество артефактов. Чтобы получить изображение с меньшим количеством артефактов, нужно четные строки сканировать справа-налево, а нечетные строки сканировать слева-направо.

$\times 3$

3 2 ( $1/8$ )

### 1.5.3 Фильтр Джарвиса, Джунка и Нинка

В случае когда фильтры Флойда-Стейнберга дают недостаточно хороший результат, применяют фильтры с более широким распределением ошибки. Фильтр Джарвиса, Джунка и Нинка требует связи с 12 соседями, что очевидно ведет в боль-

шим затратам памяти и времени:

Р Р \* 7 5

3 5 7 5 3

1 3 5 3 1 (1/48)

#### 1.5.4 Фильтр Стаки

Стаки переработал фильтр Джарвиса, Джунка и Нинка. После такого как мы вычислим 8/42 ошибки, остальные значения можно получить при помощи побитовых сдвигов, тем самым сокращая время работы алгоритма.

Р Р \* 8 4

2 4 8 4 2

1 2 4 2 1 (1/42)

#### 1.5.5 Фильтр Бурка

Бурк переработал фильтр Стаки. Результат можно получить чуть быстрее за счет использования побитовых операций.

Р Р \* 8 4

2 4 8 4 2 (1/32)



## **2 Конструкторский раздел**

Тут что-то будет(возможно)

### **3 Технологический раздел**

Тут что-то будет(возможно)

## 4 Экспериментальный раздел

В данном разделе проводятся вычислительные эксперименты. А на рис. 4.1 показана схема мыслительного процесса автора...

Рисунок 4.1 — Как страшно жить

## 5 Организационно-экономический раздел

## **6 Промышленная экология и безопасность**

## Заключение

В результате проделанной работы стало ясно, что ничего не ясно...

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Crocker, Lee Daniel*. Digital Halftoning / Lee Daniel Crocker. — 1989.
2. *Bayer, B.E.* An Optimum Method for Two-Level Rendition of Continuous Tone Pictures / B.E. Bayer. — IEEE International Conference on Communications, Conference Records, 1973.

## Приложение А    Картинки

Рисунок А.1 — Картинка в приложении. Страшная и ужасная.



## Приложение Б    Еще картинки

Рисунок Б.1 — Еще одна картинка, ничем не лучше предыдущей. Но надо же как-то заполнить место.