

Реферат

Титульный лист

Содержание

Введение	5
1 Аналитический раздел	6
1.1 Случайный дизеринг	6
1.2 Шаблонный дизеринг	6
1.3 Упорядоченный дизеринг	7
1.4 Дизеринг при помощи диффузии ошибок	7
1.5 Вариации алгоритма дизеринга при помощи диффузии ошибок	8
1.5.1 Фильтр Флойда-Стейнберга	8
1.5.2 "Ложный" фильтр Флойда-Стейнберга	8
1.5.3 Фильтр Джарвиса, Джунка и Нинка	8
1.5.4 Фильтр Стаки	9
1.5.5 Фильтр Бурка	9
1.6 Выбор оптимального класса алгоритма	9
2 Конструкторский раздел	10
2.1 Оценка качества изображений	10
2.1.1 PSNR	10
2.1.2 SSIM	10
2.2 Виды случайных распределений	11
2.2.1 Белый шум	11
2.2.2 Красный шум	11
2.2.3 Гауссовский шум	12
2.2.4 Розовый шум	13
2.2.5 Алгоритм Флойда-Стейнберга	13
3 Технологический раздел	15
4 Экспериментальный раздел	16
Заключение	17
Список использованных источников	18

Глоссарий

Артефакт — аномалия, возникающая во время визуального представления изображения[1]

Цифровой шум — дефект изображения, вносимый фотосенсорами и электроникой устройств, которые их используют вследствие несовершенства технологий. Цифровой шум заметен на изображении в виде наложенной маски из пикселей случайного цвета и яркости[1]

Фиксированная пороговая обработка (eng. fixed thresholding) — обработка пикселя на основе какого-то фиксированного порогового значения. В случае если текущее значения пикселя больше этого порогового значения, пиксель закрасивает одним фиксированным цветом, иначе - другим. [2]

Интенсивность цвета — степень отличия хроматического цвета от равно-го ему по светлоте ахроматического, «глубина» цвета. Два оттенка одного тона могут различаться степенью блёклости. При уменьшении насыщенности каждый хроматический цвет приближается к серому.[1]

Ахроматические цвета — оттенки серого (в диапазоне белый — чёрный).[1]

Монохромное изображение — изображение, содержащее свет одного цвета (длины волны), воспринимаемый, как один оттенок (в отличие от цветного изображения, содержащего различные цвета).[1]

Цвета шума — система терминов, приписывающая некоторым видам стационарных шумовых сигналов определённые цвета исходя из аналогии между спектром сигнала произвольной природы

Спектральная плотность $S(w)$ стационарного случайного процесса $x(t)$ — это частотная функция, характеризующая спектральный (частотный) состав процесса, и представляет собой частотную характеристику для средних значений квадратов амплитуд гармоник, на которые может быть разложен случайный процесс.

Обозначения и сокращения

- ⊠ — Искомый пиксель
- — Пустой пиксель
- — Закрашенный пиксель
- ▣ — Произвольный пиксель

Введение

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

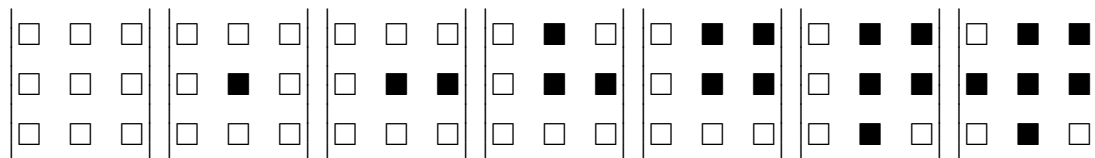
Нижеприведенные алгоритмы описываются для черно-белых изображений. Для цветных изображений алгоритмы аналогичны.

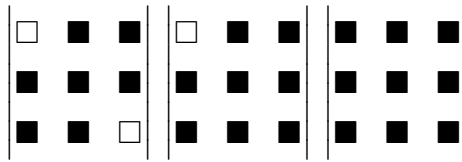
1.1 Случайный дизеринг

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.[2] Для каждого пикселя в нашем черно-белом изображении мы генерируем случайное число в диапазоне 0-255: если случайное число больше, чем значение в данной точке, то отображаем белый пиксель, иначе отображаем черн пиксель. Это создает изображение с большим количеством шумов. Хотя изображение выглядит неточным и зернистым, оно не содержит артефактов[2]]. Этот 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 не будет иметь артефактов после того, как к нему применят случайный дизеринг, а остальные методы дизеринга приведут к возникновению артефактов [2].

1.2 Шаблонный дизеринг

Шаблонный дизеринг подразумевает то, что мы увеличиваем разрешение изображения. Так же, как и случайный дизеринг, это тривиальный алгоритм, но он гораздо более эффективен.[3] Для каждой точки изображения мы генерируем «шаблон» пикселей, который аппроксимирует эту точку. Тем самым имитируется больший набор оттенков, чем поддерживает наша глубина цвета. Например, шаблон 3x3. Он имеет 512 вариантов возможных расположений пикселей, но расположение пикселей не влияет на интенсивность цвета шаблона. Интенсивность 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. Для того, чтобы достичь этого, каждый пиксель в исходном изображении сопоставляется с пикселем на конечном изображении один-к-одному. Существуют два вида упорядоченного дизеринга: кластерный и дисперсный. Суть этих методов заключается в том, что исходное изображение разбивается на квадраты пикселей и значения маски в каждой точке квадрата выступает в роли порогового значения. Если значение цвета пикселя(отмасштабированное под интервал маски) в данной точке больше значения маски, то красим пиксель в черный цвет, иначе в белый. Кластерные паттерны выглядят следующим образом:

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 и подобное). Во многих исследованиях [4][2] было отмечено, что если устройство вывода позволяет применить дисперсный метод, то его применение является предпочтительным. Так же Байер [4] 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

Основным недостатком данного метода считается то, что в результате его работы формируется большое количество артефактов[3].

1.4 Дизеринг при помощи диффузии ошибок

Метод, обладающий наилучшим качеством среди представленных, - метод рассеивания ошибок. Но так же он, к сожалению, самый медленный.[2] Существуют несколько вариантов этого алгоритма, причем скорость алгоритма обратно пропорци-

онально качеству изображения.[2] 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 дизернга при помощи диффузии ошибок

Линия сканирование движется слева-направо. Когда линия сканирования доходит до конца горизонтальной строки пикселей, переходим к первому пикселю 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 $*7/16$*

1.5.1 Фильтр Флойда-Стейнберга

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

$$\begin{vmatrix} \boxminus & \boxtimes & 7 \\ 3 & 5 & 1 \end{vmatrix} (1/16)$$

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 чётные строки сканировать справа-налево, а нечетные строки сканировать слева-направо.

$$\begin{vmatrix} \boxtimes & 3 \\ 3 & 2 \end{vmatrix} (1/8)$$

1.5.3 Фильтр Джарвиса, Джунка и Нинка

В случае когда фильтры Флойда-Стейдберга дают недостаточно хороший результат, применяются фильтры с более широким распределением ошибки. Фильтр Джарвиса, Джунка и Нинка требует связи с 12 соседями, что очевидно ведет в большим затратам памяти и времени[2]:

$$\begin{vmatrix} \boxminus & \boxminus & \boxtimes & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{vmatrix} (1/48)$$

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.

$$\left| \begin{array}{ccccc} \square & \square & \boxtimes & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{array} \right| (1/42)$$

1.5.5 Фильтр Бурка

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

$$\left| \begin{array}{ccccc} \square & \square & \boxtimes & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \end{array} \right| (1/32)$$

Существует много различных вариантов фильтро дизеринга при помощи диффузии ошибок, здесь приведены наиболее популярные алгоритмы.[2]

1.6 Выбор оптимального класса алгоритма

Вышеприведенные методы упорядочены по качеству получаемого на выходе изображения, однако, такие соображения как время, экономия памяти и прочие являются определяющими при выборе алгоритма[2]. На основе вышеприведенных данных сравним классы алгоритмов дизеринга.

Характеристика Вид алгоритма	Скорость	Качество	Доп память
Случайный	+	-	-
Шаблонный	+-	-+	+
Упорядоченный	-+	+-	-
Диффузия ошибок	-	+	+

Самым

быстрым классом алгоритмов дизеринга является случайный дизеринг. Однако, качество получаемых при помощи изображений низко. Выдвинем гипотезу, что существуют алгоритмы случайного дизеринга не сильно уступающие по качеству алгоритму диффузии ошибок Флойда-Стейнберга.

2 Конструкторский раздел

2.1 Оценка качества изображений

2.1.1 PSNR

Пиковое отношение сигнала к шуму (англ. peak signal-to-noise ratio) - соотношение между максимумом возможного значения сигнала и мощностью шума, искажающего значения сигнала.[5]

$$PSNR = 20 \log_{10} \frac{MAX_i}{\sqrt{MSE}} \quad (2.1)$$

Где MAX_i - это максимальное значение, принимаемое пикселем изображения, MSE - среднеквадратичное отклонение. Для двух монохромных изображений I и K размера $m \times n$, одно из которых считается зашумленным приближением другого, вычисляется так:

$$MSE = \frac{1}{m * n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i,j) - K(i,j)|^2 \quad (2.2)$$

2.1.2 SSIM

Индекс структурного сходства (SSIM от англ. structure similarity) — метод измерения схожести между двумя изображениями путем полного сопоставления. SSIM-индекс является развитием традиционных методов, таких как PSNR (peak signal-to-noise ratio) и метод среднеквадратичной ошибки MSE, которые оказались несовместимы с физиологией человеческого восприятия.

Отличительной особенностью метода, в отличие от MSE и PSNR, является то, что он учитывает «восприятие ошибки» благодаря учёту структурного изменения информации. Идея заключается в том, что пиксели имеют сильную взаимосвязь, особенно когда они близки пространственно. Данные зависимости несут важную информацию о структуре объектов и о сцене в целом. Особенностью является, что SSIM всегда лежит в промежутке от -1 до 1, причем при его значении равном 1, означает, что мы имеем две одинаковые картинки. Общая формула имеет вид

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2.3)$$

Тут μ_x среднее значение для первой картинки, μ_y для второй, σ_x среднеквадратичное отклонение для первой картинки, и соответственно σ_y для второй, σ_{xy} это уже ковариация. Она находится следующим образом:

$$\sigma_{xy} = \mu_{xy} - \mu_x\mu_y \quad (2.4)$$

c_1 и c_2 - поправочные коэффициенты, которые нужны вследствие малости знаменателя.

$$c_1 = (0,01 * d)^2 \quad (2.5)$$

$$c_2 = (0,03 * d)^2 \quad (2.6)$$

d - количество цветов, соответствующих данной битности изображения

2.2 Виды случайных распределений

2.2.1 Белый шум

Белый шумом называют сигнал с равномерной спектральной плотностью на всех частотах и дисперсией, равной бесконечности. Является стационарным случайным процессом. В качестве сигнала в задаче дизеринга рассматривается последовательность чисел, получаемых от генератора случайных чисел.

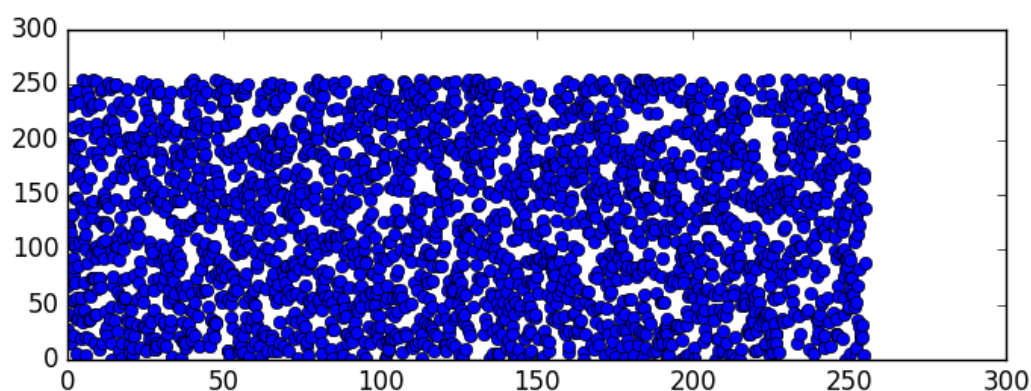


Рисунок 2.1 — Диаграмма белого шума//почему-то очень много места сверху

2.2.2 Красный шум

Спектральная плотность красного шума пропорциональна $1/f^2$, где f — частота. Это означает, что на низких частотах шум имеет больше энергии, чем на высоких. Коричневый шум можно получить при помощи алгоритма случайного блуждания. Выбирается наугад число в определенном диапазоне, прибавляются к значению предыдущего числа(тоже полученного случайно) и делится пополам. Например, если первое значение равно 103, а нам случайно выпало число 67, то следующее значение будет

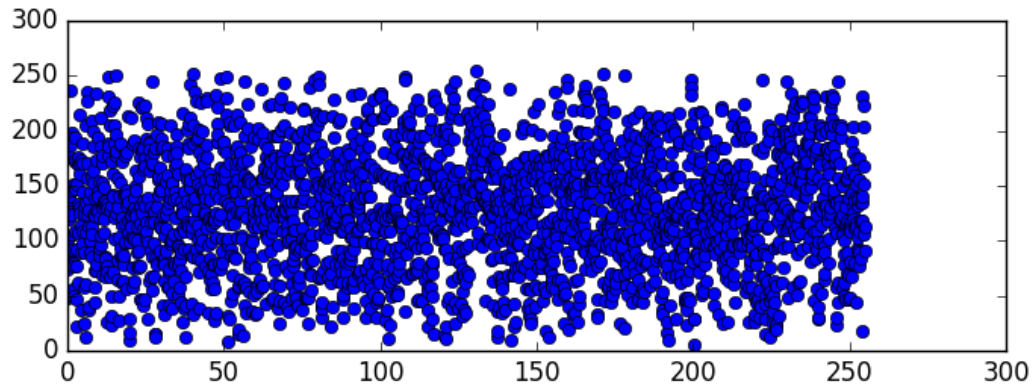


Рисунок 2.2 — Диаграмма красного шума

170. Если построить так длинную последовательность значений, получится красный шум.

Листинг 2.1 — Получение коричневого шума

```

1  def smoother(noise):
2      output = []
3      for i in range(len(noise) - 1):
4          output.append(0.5 * (noise[i] + noise[i+1]))
5  return output

```

2.2.3 Гауссовский шум

Гауссовский шум - шум, имеющий функцию плотности вероятности (PDF), равную нормальному распределению, которое также известно как гауссово распределение.

$$p_g(z) = \frac{1}{\sigma\sqrt{2 * \pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (2.7)$$

z - количество цветов, μ - среднее значение, σ - стандартное отклонение.

2.2.4 Розовый шум

Розовый шум представляет собой что-то среднее между коричневым шумом и белым шумом. Получение его аналогично получению коричневого шума.

Листинг 2.2 — Получение розового шума

```
1 def rougher(noise):
2     output = []
3     for i in range(len(noise) - 1):
4         output.append(0.5 * (noise[i] - noise[i+1]))
5 return output
```

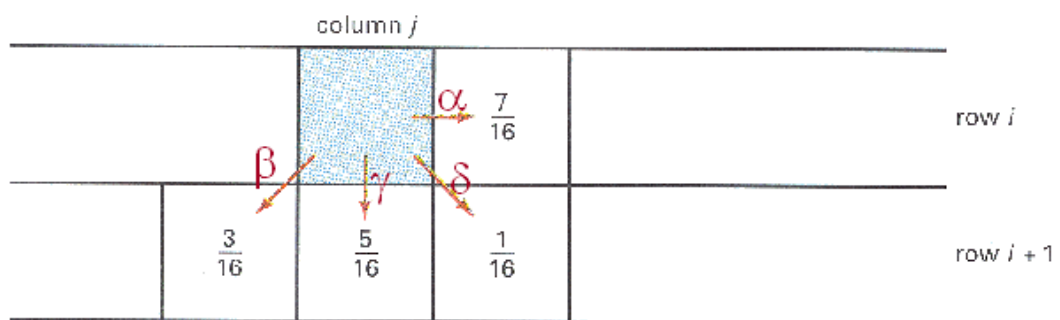
2.2.5 Алгоритм Флойда-Стейнберга

Рассмотрим более детально алгоритм Флойда-Стейнберга.

Листинг 2.3 — Алгоритм Флойда-Стейнберга

```
1 for (x = 0; x < width; x++)
2     for (y = 0; y < height; y++){
3          $P(x, y) = \text{trunc}(I(x, y) + 0.5);$ 
4          $e = I(x, y) - P(x, y);$ 
5          $I(x, y+1) += \backslash alpha * e;$ 
6          $I(x+1, y-1) += \backslash beta * e;$ 
7          $I(x+1, y) = \backslash gamma * e;$ 
8          $I(x+1, y+1) += \backslash sigma * e; }$ 
```

Рисунок 2.3 — Диаграмма гауссовского шума



$$\alpha + \beta + \gamma + \delta = 1.0$$

Рисунок 2.4 — Схема рассеивания ошибки

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

Заключение

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

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Википедия. — <https://en.wikipedia.org>.
2. *Crocker, Lee Daniel*. Digital Halftoning / Lee Daniel Crocker. — 1989.
3. *Ulichney, R.* Digital Halftoning / R. Ulichney. — The MIT Press, 1987.
4. *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.
5. *Инсаф, Ашрапов*. PSNR и SSIM или как работать с изображениями под С / Ашрапов Инсаф. — <https://habrahabr.ru/post/126848/>, 2011.