

Санкт-Петербургский государственный университет

Направление Математическое обеспечение и администрирование
информационных систем
Профиль Технология программирования

Петров Егор Александрович

Параллельный алгоритм бинаризации изображений со скользящим окном

Бакалаврская работа

Научный руководитель:
к. ф.-м. н., доцент Макаров А. А.

Рецензент:
к. ф.-м. н., доцент Евдокимова Т. О.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
Technology in Programming

Egor Petrov

Parallel algorithm of image binarisation with sliding window

Bachelor's Thesis

Scientific supervisor:
associate professor Anton Makarov

Reviewer:
associate professor Tatyana Evdokimova

Saint-Petersburg
2017

Оглавление

Введение	4
1. Цель работы	5
2. Описание вычислительного комплекса и методики эксперимента	6
3. Экспериментальные результаты: оценки точности	9
4. Экспериментальные результаты: время работы	12
Заключение	13
Список литературы	14

Введение

Задачи автоматической обработки изображений сегодня необходимо решать повсеместно и они могут быть крайне трудоёмкими. Однако в большинстве случаев значимые результаты можно получить по бинарным изображениям (элементы которого могут принимать только два значения), формируемым по исходным многоградационным. При этом значительно сокращается объём обрабатываемой информации. Также бинаризация помогает выделить классы объектов на изображении, что используется в задачах распознавания образов. [3]

Рассматриваемый в данной работе алгоритм, основан на рандомизированном блочном алгоритме бинаризации, использующем условие равенства яркостей полутонового и бинарного изображений. [4] Рандомизированный блочный алгоритм позволяет улучшить визуальное качество бинарного изображения и дает хорошие результаты в полиграфии при печати для процесса растривания. Из-за особенностей растрового процессора изображений и размера печатных точек используются блоки определенного размера, в результате чего проявляется характерная пространственная структура на границах блоков, устранить которую можно за счет оконной пространственной фильтрации. [1] Ввиду того, что каждый блок может обрабатываться независимо, блочный алгоритм может быть эффективно распараллелен.

1. Цель работы

1. Реализация последовательного и параллельного алгоритмов фильтрации изображения на CPU и GPU.
2. Сравнение оценок качества бинаризованных блочным и оконным алгоритмом изображений.
3. Анализ времени работы параллельного алгоритма.

2. Описание вычислительного комплекса и методики эксперимента

Пусть $f = f(x, y)$ - исходное k -битовое полутоновое изображение размера $m \times n$. Пусть $g = g(x, y)$ - обработанное изображение (того же размера), полученное в результате бинаризации исходного изображения. Процесс бинаризации описывается уравнением

$$g(x, y) = B[f(x, y)], \quad (1)$$

где B - оператор бинаризации, сопоставляющий яркости элемента $f(x, y) \in [0, 2^k - 1]$ элемент $g(x, y) \in 0, 1$.

Под яркостью цифрового изображения здесь и далее будем понимать сумму значений всех его пикселей. Исходное полутоновое изображение $f(x, y)$ обрабатывается окном нечетного размера $h \times h$ таким образом, центральный элемент окна определяется однозначно. Опишем результат (отклик) бинаризации (пространственной фильтрации) в точке (x, y) .

1. Вычислим суммарную приведенную яркость окна, обозначаемого через f_l , которая определяется количеством целых единиц u_l в окне f_l :

$$\lfloor 2^{-k} \sum_{x, y \in f_l} f(x, y) \rfloor = u_l, \quad (2)$$

где скобки $\lfloor \rfloor$ - обозначают взятие целой части числа. Ясно, что $u_l \in [0, h^2]$.

2. Рассмотрим нулевую матрицу g_l размера $h \times h$. В матрице g_l поставим единицы на позиции, в которых находятся u_l наибольших значений яркостей $f(x, y)$ окна f_l , начиная с максимального значения. При этом если найдено несколько одинаковых наибольших значений, то среди всех вариантов один выбирается случайно в соответствии с некоторой вероятностью. Таким образом реализуется механизм рандомизации расстановки единиц, основанный на случайном правиле.

3. Отклик в точке (x, y) – это центральный элемент теперь уже бинарной матрицы g_i .

Особенность расстановки нулей и единиц приводит к тому, что в бинарном изображении частично воспроизводится пространственное распределение яркости полутонового изображения. Для визуального восприятия это означает следующее: 1) более точно воспроизводятся детали и границы объектов полутонового изображения; 2) выделение слабоконтрастных структур, которые могут быть неразличимы глазом или плохо передаваться техническими средствами, формирующими изображение.

Бинаризация полутоновых изображений является примером квантования изображения на два уровня, при котором могут возникать потери информации, ведущие к снижению качества изображения. Объективные оценки весьма ненадежны и неуниверсальны. Однако они необходимы. Рассмотрим некоторые из них.[2]

Расстояние между двумя изображениями $f = f(x, y)$ и $g = g(x, y)$, где $x = 1...m$, $y = 1...n$, определяется через норму Гёльдера, усредненную на количество элементов изображения

$$\|f - g\|_p = \left(\frac{1}{mn} \sum_{x=1}^m \sum_{y=1}^n |f(x, y) - g(x, y)|^p \right)^{1/p}, p \geq 1. \quad (3)$$

При $p = 1$ возникает средняя разность, а при $p = 2$ – корень из среднеквадратического отклонения (RMSE) (показатель рассеивания случайной величины относительно её математического ожидания).

На основе введенного расстояния строится пиковое отношение сигнал/шум (PSNR) между двумя изображениями, которое определяется следующим образом

$$PSNR = 20 \lg \frac{2^k - 1}{\|f - g\|_2} \quad (4)$$

Пусть μ_f - это среднее значение яркости изображения f

$$\mu_f = \frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x, y), \quad (5)$$

через σ_f^2 обозначим вариацию изображения f

$$\sigma_f^2 = \frac{1}{(m-1)(n-1)} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (f(x, y) - \mu_f)^2, \quad (6)$$

а через σ_{fg} обозначим ковариацию изображений f и g

$$\sigma_{fg} = \frac{1}{(m-1)(n-1)} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} (f(x, y) - \mu_f)(g(x, y) - \mu_g). \quad (7)$$

Хорошее приближение к человеческому зрению дает индекс структурного сходства

$$SSIM(f, g) = (l(f, g))^\alpha (c(f, g))^\beta (s(f, g))^\gamma, \quad (8)$$

где $l(f, g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1}$ - яркостная компонента, $c(f, g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2}$ - компонента, отвечающая за контраст, $s(f, g) = \frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3}$ - структурная компонента. Вспомогательные константы $C_1 = (K_1K)^2$, $C_2 = (K_2K)^2$, $C_3 = C_2/2$, $K = 2^k - 1$, $K_1 = 0.01$, $K_2 = 0.03$, необходимы для устойчивости вычислений. Показатели степени $\alpha, \beta, \gamma \geq 0$, $\alpha + \beta + \gamma > 0$ используются для корректировки влияния каждой компоненты на индекс в целом. Далее будем использовать значения $\alpha = \beta = 1, \gamma = 0.1$.

3. Экспериментальные результаты: оценки точности

Результаты работы блочного алгоритма бинаризации с размером блока 4×4 и оконных алгоритмов с размерами окна 3×3 и 5×5

Таблица 1: Оценки качества

	MAD (p=1)	RMSE (p=2)	PSNR	SSIM
оконный алгоритм 3×3	0.2277	0.2908	10.7258	0.7218
оконный алгоритм 5×5	0.2413	0.3132	10.0838	0.7429
блочный алгоритм	0.2599	0.3418	9.3232	0.7299

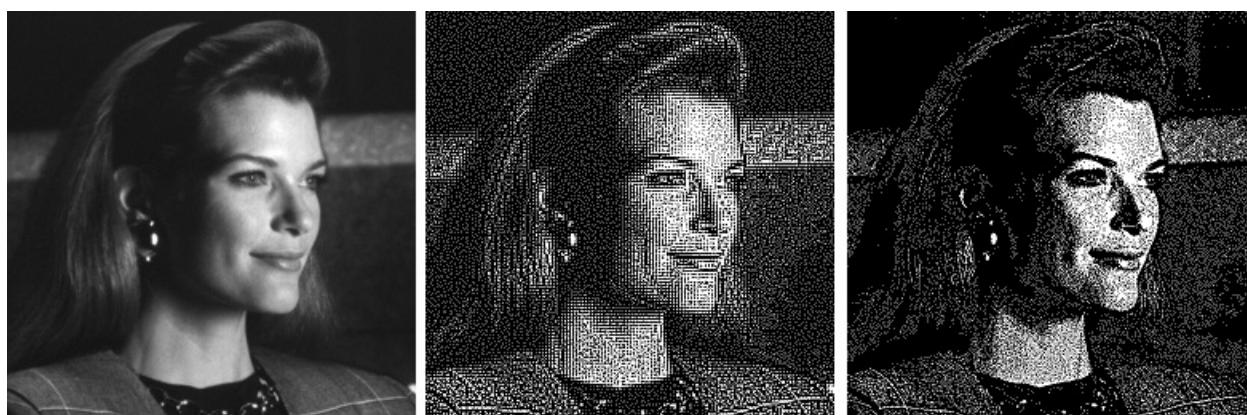


Рис. 1: Оригинал, Блочный алгоритм, Оконный алгоритм 3×3 (512×512)

Таблица 2: Оценки качества

	MAD (p=1)	RMSE (p=2)	PSNR	SSIM
оконный алгоритм 3×3	0.2943	0.3341	9.5220	0.6841
оконный алгоритм 5×5	0.3012	0.3441	9.2660	0.7001
блочный алгоритм	0.3329	0.3875	8.2326	0.6830

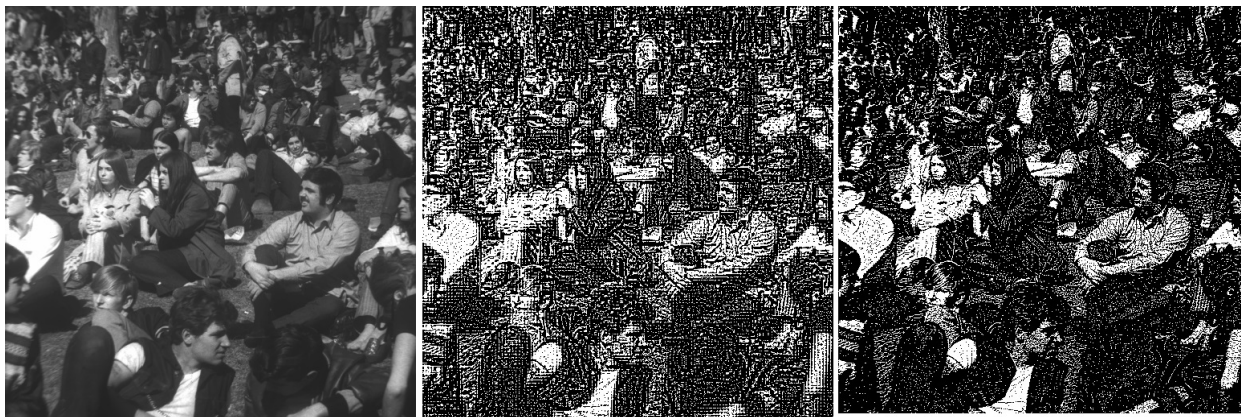


Рис. 2: Оригинал, Блочный алгоритм, Оконный алгоритм 3×3 (512×512)

Таблица 3: Оценки качества

	MAD (p=1)	RMSE (p=2)	PSNR	SSIM
оконный алгоритм 3×3	0.2315	0.3229	9.8162	0.9132
оконный алгоритм 5×5	0.2218	0.3076	10.2389	0.9182
блочный алгоритм	0.2422	0.3393	9.3881	0.9106

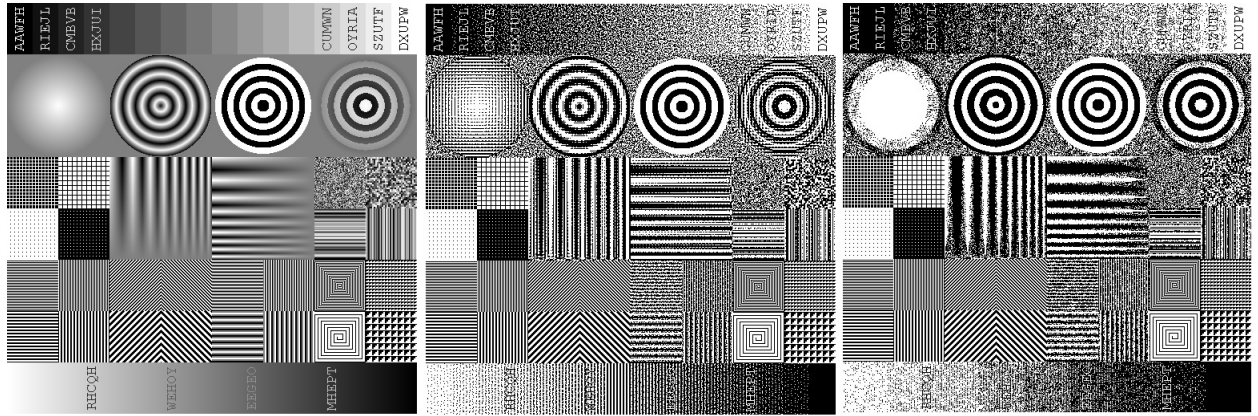


Рис. 3: Оригинал, Блочный алгоритм, Оконный алгоритм 3×3 (512×512)

Из результатов вычислений выбранных мер искажений (см. табл. 1 - 3) следует, что в большинстве случаев предложенный оконный алгоритм бинаризации с размером окна 3×3 показывает лучшие результаты по сравнению с блочным алгоритмом. Только по критерию SSIM оконный и блочный алгоритмы дают примерно одинаковые значения.

4. Экспериментальные результаты: время работы

Результаты распараллеливания на CPU (OpenMP) на компьютере с процессором Intel Core i7-3610QM (4 ядра):

Таблица 4: Время работы бинаризации изображения размера 3180x2700

количество потоков	время работы (мс)
1	3200
2	1900
4	1200
8	900

Результаты распараллеливания на GPU (OpenCL) на видеокарте AMD Radeon HD 7600M:

Таблица 5: Время работы бинаризации изображения размера 3180x2700

вид распараллеливания	время работы (мс)
CPU (1 поток)	3200
CPU (8 потоков)	900
GPU	700

Распараллеливание на CPU производилось разбиением строк изображения между потоками. Попытки распараллелить вычисления в самом окне (вычисление суммы яркостей) только увеличивали время работы.

При распараллеливании на GPU каждое окно обрабатывалось отдельным потоком.

Результаты распараллеливания на 4-х ядерном CPU и GPU показывают, что обе реализации являются эффективными, однако GPU немного превосходит CPU.

Заключение

В процессе работы были реализованы параллельные алгоритмы бинаризации изображений со скользящим окном на CPU и GPU. Доказано превосходство оконного алгоритма над блочным с помощью вычисления оценок точности для изображений. Было получено ускорение в 4,5 раза относительно последовательной реализации.

Список литературы

- [1] Daniel L. Lau Gonzalo R. Arche. Modern digital halftoning. — 2001.
- [2] Макаров А.А. Максимов В.В. О бинаризации гидролокационных изображений. — 2017.
- [3] Я.А. Фурман. Цифровые методы обработки и распознавания бинарных изображений. — Красноярск : Изд-во Краснояр. ун-та, 1992.
- [4] Яковлева Е.С. Макаров А.А. О свойствах блочного алгоритма бинаризации цифровых изображений. — Компьютерные инструменты в образовании., 2015.