

РОССИЙСКАЯ АКАДЕМИЯ НАУК  
СИБИРСКОЕ ОТДЕЛЕНИЕ  
А В Т О М Е Т Р И Я

№ 6

1995

УДК 681.3.04/05

А. Н. Колесников, Е. В. Тришина

(Новосибирск, Россия — Аделаида, Австралия)

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ УТОНЬШЕНИЯ  
БИНАРНЫХ ИЗОБРАЖЕНИЙ

Разработан параллельный алгоритм утоньшения бинарных изображений, основанный на дистанционном преобразовании. Представленный алгоритм предназначен для реализации на многопроцессорных вычислительных системах с распределенной памятью. Проверка, проведенная на реальных изображениях, показала его работоспособность и эффективность.

**Введение.** Утоньшение (скелетизация) изображений находит широкое применение в обработке изображений и распознавании образов для структурного описания, измерения параметров, сокращения избыточности изображений [1—6]. В связи с большим объемом подлежащей преобразованию информации при решении практических задач скелетизация изображений требует значительных временных затрат. Один из возможных способов сокращения времени обработки — использование многопроцессорной вычислительной техники.

Цель данной работы — разработка параллельного алгоритма скелетизации бинарных изображений, предназначенного для реализации на многопроцессорных рабочих станциях с распределенной памятью.

В разд. 1 приведено краткое описание последовательного алгоритма скелетизации изображений, основанного на дистанционном преобразовании. В разд. 2 дано описание аппаратных средств, на которые ориентирован разрабатываемый параллельный алгоритм, а также модель распараллеливания. В разд. 3 изложено решение поставленной задачи для четырех процессоров. В разд. 4 описан параллельный алгоритм, предназначенный для реализации на двумерной решетке процессоров. В разд. 5 приведены результаты практической проверки разработанного алгоритма на транспьютерной рабочей станции.

**1. Последовательный алгоритм скелетизации изображений.** В качестве алгоритма скелетизации, подлежащего распараллеливанию, был выбран алгоритм, основанный на дистанционном преобразовании (ДП). Этот метод имеет два важных достоинства: скелет изображения может быть получен за фиксированное количество проходов по изображению, и каждая точка такого скелета содержит информацию о расстоянии до ближайшей точки фона, которую впоследствии можно пересчитать в локальную ширину объекта. Процедура получения скелета состоит из трех этапов:

- дистанционное преобразование;
- выделение скелетных точек;

— пересчет значений ДП в локальную толщину линий и утоньшение скелетных линий до единичной толщины (для линий четной ширины).

**Дистанционное преобразование.** Вид дистанционного преобразования определяется метрикой пространства. В данной работе использовалось дистанционное преобразование с метрикой «шахматной доски» [7, 8]. Дистанционное преобразование этого вида может быть осуществлено за два прохода по изображению: первый проход окном  $2 \times 3$  в лексикографическом порядке; второй проход — в обратном направлении; в дальнейшем будем называть эти проходы

Рис. 1. Относительное расположение точки  $p$  и ее соседей

$n_2$     $n_3$     $n_4$   
 $n_1$     $p$     $n_5$   
 $n_8$     $n_7$     $n_6$

прямой и обратной прогонкой. Будем полагать, что входное изображение окаймлено рамкой цвета фона шириной в один пиксел. Кроме того, будем полагать, что оно предварительно размечено для предстоящего ДП: точки фона обозначены нулями, точки объектов — некоторым достаточно большим числом.

Во время прямой прогонки для каждой точки объекта  $p$  вычисляется значение ДП по следующей формуле (рис. 1):

$$p = \min(p, n_1 + 1, n_2 + 1, n_3 + 1, n_4 + 1).$$

Во время обратной прогонки для каждой точки определяется значение ДП по аналогичной формуле:

$$p = \min(p, n_5 + 1, n_6 + 1, n_7 + 1, n_8 + 1).$$

На рис. 2 представлены результаты дистанционного преобразования для тестового объекта после прямой и обратной прогонки, иллюстрирующие сущность метода.

*Выделение скелетных точек* по локальным экстремумам значений ДП также осуществляется за два прохода по изображению окном  $3 \times 3$  [8, 9]. Во время прямой прогонки производится выделение скелетных точек на основе анализа информации о дистанционном преобразовании (рис. 3, а). Во время обратной прогонки осуществляется восстановление связности скелета (рис. 3, б). В связи с тем, что для линий четной толщины полученный скелет имеет толщину в два пиксела, необходим еще один проход по изображению для сокращения толщины линий до единицы. Одновременно для каждой точки объекта можно пересчитать дистанцию до ближайшей точки фона в локальную толщину линии.

*Пересчет значения ДП в ширину линии.* Как нетрудно видеть, для линий четной ширины получается скелет толщиной в два пиксела. Если для дальнейшей обработки необходим скелет единичной толщины, «лишние» скелетные точки удаляются за один проход по изображению с помощью стандартного алгоритма Арчелли. Во время этого прохода производится пересчет значения дистанционного преобразования  $D$  в локальную ширину линии  $W$  с помощью очевидного соотношения:  $W = 2D - d$ , где поправка  $d$  ( $d = 0, 1$ ) определяется конфигурацией скелета в окне  $3 \times 3$ .

a	b
0000000000011000000000000000	0000000000011000000000000000
0000000000011000000000000000	0000000000011000000000000000
011111111111111111111111111110	011111111111111111111111111110
01222222222222222222222222210	01222222222222222222222222210
01233333333333333333333333210	01233333333333333333333333210
01234444444444444444444443210	01234444444444444444444443210
01234555555555555555555543210	01234555555555555555555543210
01234566666666666666666543210	01234444444444444444444443210
0123456777777777777776543210	0123333333333333333333333210
012345678888888888876543210	012222222222222222222222210
012345678999999999876543210	0111111111111111111111111110
0000000000011000000000000000	0000000000011000000000000000
0000000000011000000000000000	0000000000011000000000000000

Рис. 2. Изображение тестового объекта после прямой (а) и обратной (б) прогонки дистанционного преобразования

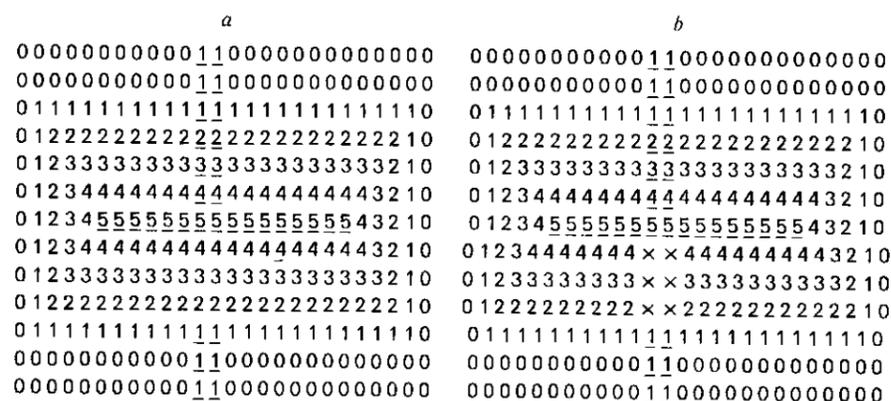


Рис. 3. Изображение тестового объекта после прямой (*a*) и обратной (*b*) прогонки алгоритма выделения скелетных точек:  
*a* – скелетные точки, зарегистрированные во время прямой прогонки, отмечены подчеркиванием; *b* – скелетные точки, зарегистрированные во время обратной прогонки, обозначены крестиком

Таким образом, для реализации описанного алгоритма скелетизации требуется несколько последовательных проходов по всему изображению в двух направлениях. Это обстоятельство делает задачу распараллеливания нетривиальной.

Как нетрудно видеть, дистанционное преобразование является определяющим этапом для всего алгоритма: если удастся создать параллельный алгоритм для дистанционного преобразования, то тем самым будет решена задача распараллеливания алгоритма выделения скелетных точек. В связи с этим в работе основное внимание уделяется описанию параллельного алгоритма дистанционного преобразования.

**2. Аппаратные средства и модель распараллеливания.** При разработке параллельного алгоритма скелетизации, пригодного для решения практических задач обработки изображений, предпочтение было отдано многопроцессорным вычислительным системам класса MIMD с распределенной памятью, поскольку они вполне доступны на рынке вычислительных средств и имеют высокое значение параметра «производительность/цена». Большинство MIMD-компьютеров имеет регулярную топологию с постоянной валентностью, их расширение не требует особых структурных перестроек.

Специфика задачи и аппаратных средств обусловила выбор геометрической модели распараллеливания. В данном случае это означает следующее: входное изображение разбивается на фрагменты, которые распределяются между процессорами, объединенными в двумерный массив. Каждый процессор обрабатывает свою часть данных, время от времени обмениваясь необходимой информацией с соседними процессорами. По завершении обработки фрагменты изображения собираются в управляющем компьютере и сохраняются во внешней памяти.

**3. Параллельный алгоритм скелетизации для четырех процессоров:**  $N_p = 2 \times 2$ . Рассмотрим решение поставленной задачи для случая четырех процессоров. Это частное решение впоследствии будет использовано при нахождении общего решения для двумерного массива процессоров.

Для распараллеливания алгоритма дистанционного преобразования прежде всего учтем то обстоятельство, что два этапа этого преобразования обладают свойством коммутативности, т. е. последовательность проходов по изображению (а именно: «слева направо, сверху вниз», а затем «справа налево, снизу вверх») может быть изменена на обратную. Более того, можно учесть симметричность алгоритма и в горизонтальном направлении: операция обработки «слева направо» перестановочна с операцией «справа налево».

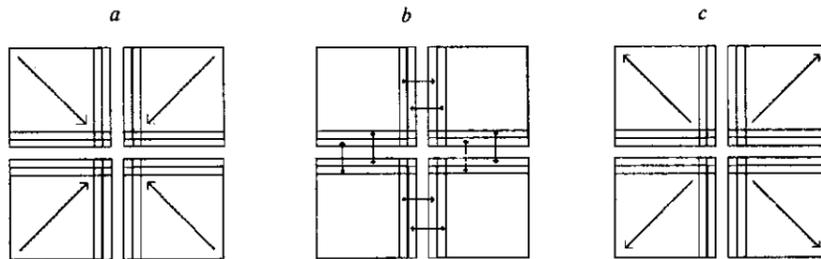


Рис. 4. Схема обработки фрагментов изображения для четырех процессоров:  
*a* — направление прямой прогонки; *b* — схема обмена данными между процессорами; *c* — направление обратной прогонки

Таким образом, рассматриваемый алгоритм дистанционного преобразования может быть распараллелен для исполнения на четырех процессорах. Согласно этому подходу, исходное изображение делится на четыре фрагмента (четыре квадранта). Поскольку все операции дистанционного преобразования фактически осуществляются в окне  $3 \times 3$ , при разбиении изображения на фрагменты необходимо обеспечить их взаимное перекрытие в полосе шириной в два пиксела (рис. 4, *a*).

**Прямая прогонка.** В четырех процессорах осуществляется дистанционное преобразование фрагментов изображения по сходящимся направлениям (см. рис. 4, *a*). При достижении границы раздела производится обмен данными: каждый процессор отдает двум соседям последний обработанный столбец и последнюю обработанную строку, взамен от одного из соседей он получает столбец, от другого — строку.

**Обратная прогонка.** Процессоры производят дистанционное преобразование в обратном направлении (рис. 4, *b, c*).

После обратной прогонки в обратном направлении происходит обмен данными на границе, после чего совершаются два прохода по изображению (в прямом и обратном направлении) для выделения скелетных точек; естественно, между прогонками также необходимо произвести обмен данными между процессорами.

**4. Параллельный алгоритм скелетизации для двумерного массива процессоров:  $N_p = N_x \times N_y$ .** Выше отмечалось, что алгоритм по своей сути является последовательным и его распараллеливание — задача нетривиальная. Учет симметрии алгоритма позволил распараллелить обработку на четыре процессора.

Что касается параллельного алгоритма для двумерного массива процессоров, то рассмотрим следующее решение проблемы согласно принятой нами модели распараллеливания: разобьем изображение на фрагменты, которые распределим по процессорам, после чего будем осуществлять прямую и обратную прогонку дистанционного преобразования с соответствующим обменом информации на границах фрагментов. Совершив прогонки достаточное количество раз, получим значения ДП для всего изображения. Очевидно, что этот алгоритм будет эффективным, если количество прогонок не будет зависеть от размеров изображений и, кроме того, не будет слишком большим.

Рассмотрим вышеперечисленные этапы получения ДП более детально.

**Разбиение изображения.** Итак, исходное изображение делится на прямоугольные фрагменты с перекрытием в два пиксела. Размеры фрагмента:

$$H_x \times H_y = (\text{size}_x / N_x + 1) \times (\text{size}_y / N_y + 1),$$

где  $\text{size}_x \times \text{size}_y$  — размеры изображения;  $H_x \times H_y$  — размеры фрагмента;  $N_x \times N_y$  — размеры двумерного массива процессоров. Каждые четыре фрагмента объединим в квадруплеты (четверки) фрагментов. На рис. 5 квадруплеты обозначены пунктиром. Для простоты будем полагать, что количество

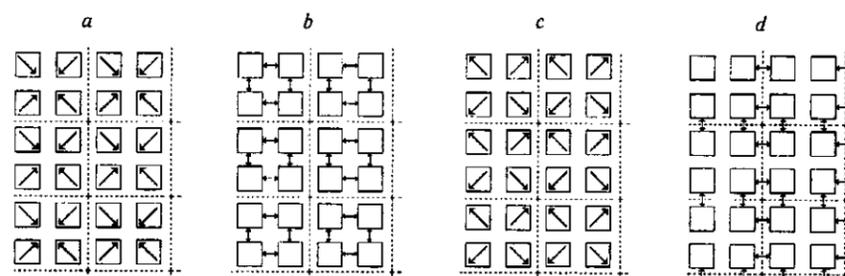


Рис. 5. Схема обработки для двумерного массива процессоров:

*a* – направление обработки при прямой прогонке; *b* – схема обмена данными после прямой прогонки (обмен внутри квадруплетов); *c* – направление обработки при обратной прогонке; *d* – схема обмена данными после обратной прогонки (обмен между квадруплетами)

процессоров кратно четырем; для большинства моделей рабочих станций это условие, как правило, выполняется.

Что касается вопроса оптимального выбора размеров фрагмента, если не принимать во внимание возможные ограничения из-за объема оперативной памяти процессоров, то решение получается путем минимизации объема данных при обменах между прогонками, т. е. минимизацией периметра прямоугольника при заданной его площади.

**Прямая прогонка.** В каждом фрагменте осуществляется прямая прогонка в соответствии со схемой, представленной на рис. 5, *a*; направление обработки определяется номером процессора в квадруплете.

После прямой прогонки производится обмен данными между фрагментами, принадлежащими одному квадруплету (рис. 5, *b*); каждый фрагмент отдает последние обработанные столбец и строку и принимает два соответствующих вектора, которые заносятся в последний столбец и последнюю строку фрагмента изображения (внутренний обмен в квадруплетах).

**Обратная прогонка.** Второй проход ДП осуществляется в обратном направлении (рис. 5, *c*), после чего фрагменты обмениваются данными с соседними фрагментами, принадлежащими соседним квадруплетам (обратная прогонка и внешний обмен между квадруплетами; рис. 5, *d*).

**Анализ ДП.** После обратной прогонки и внешнего обмена анализируются значения ДП в соответствующих строках и столбцах фрагмента. На основе этого анализа определяется количество строк  $n_y$  и столбцов  $n_x$ , подлежащих обработке в ходе предстоящей прямой прогонки, а также четыре условия завершения ДП после нее.

Количество строк  $n_y$  определяется выражением

$$n_y = \max_x \{0, \{(D(x, 1) - D(x, 0))/2\},$$

где  $D(x, y)$  — значения ДП на фрагменте;  $0 \leq x < \text{size}_x$ .

Процесс обработки может успешно завершиться после предстоящей прямой прогонки, если выполняются два условия:

- 1)  $n_y < H_y$ ,
- 2)  $\max_x \{D(x, 1)\} < 2H_y$ , где  $0 \leq x < H_x$ .

Аналогичным образом имеем выражение для количества столбцов  $n_x$ , подлежащих обработке в ходе предстоящей прямой прогонки, и последние два условия успешного завершения ДП после нее:

$$n_x = \max_y \{0, \{(D(1, y) - D(0, y))/2\}, \quad \text{где } 0 \leq y < \text{size}_y,$$

$$3) n_x < H_x,$$

$$4) \max_y \{D(1, y)\} < 2H_x, \text{ где } 0 \leq y < H_y.$$

Анализ значений ДП на двух других границах фрагмента после последующей прямой прогонки и внутреннего обмена проводится аналогичным образом.

Анализу должно быть не меньше трех, сбор информации о значениях ДП в управляющем процессоре можно начинать после третьей прогонки.

После принятия решения о завершении ДП в каждом процессоре производится выделение скелетных точек, при этом осуществляется столько прямых и обратных прогонок процедуры выделения скелета, сколько было прямых и обратных прогонок ДП.

Таким образом, процесс дистанционного преобразования можно весьма эффективно контролировать путем анализа значений этого преобразования на границе фрагментов. Сам принцип дистанционного преобразования дает нам в руки механизм контроля.

Для того чтобы оценить количество прогонок, необходимое для обработки реальных изображений, следует рассмотреть наихудший случай: в качестве объекта, подлежащего скелетизации, взять прямоугольник размером  $P_x \times P_y$  (для определенности будем считать, что  $P_x < P_y$ ). Как показывает анализ, если ширина объекта меньше  $4H_x$  и по ширине он укладывается в два квадруплета, то его можно корректно обработать за три прохода ДП по изображению. Если ширина объекта меньше  $3H_x$ , он может быть корректно обработан также за три прохода, даже если по ширине принадлежит трем квадруплетам. Еще раз отметим, что речь идет о ширине объекта, а не о его длине.

Поскольку алгоритм скелетизации имеет смысл применять для обработки изображений карт, схем, чертежей, для которых маловероятно наличие широких объектов, то в большинстве практических случаев количество прогонок будет не более трех. Более того, третий проход, вероятнее всего, будет неполным, если количество подлежащих обработке строк и столбцов меньше размеров фрагмента. Слабая зависимость количества прогонок от количества процессоров для реальных изображений обуславливает эффективность разработанного алгоритма: производительность системы линейно зависит от количества процессоров.

Кроме того, разработанная методика контроля процесса дистанционного преобразования позволяет определить минимально необходимый объем вычислений для каждого фрагмента и тем самым избежать лишних затрат на обработку.

Для массива процессоров  $4 \times 4$  достаточно трех проходов независимо от размера объектов; для массива процессоров  $2 \times 2$ , как сообщалось выше, достаточно двух проходов. Разработанный алгоритм легко модифицируется для реализации на одномерной цепочке процессоров.

*Свойства параллельного алгоритма.* Отметим основные свойства разработанного параллельного алгоритма скелетизации для двумерного массива процессоров:

— баланс нагрузки оптимальный, поскольку каждый процессор получает равное количество данных; количество операций, приходящееся на каждый процессор, примерно одинаковое;

— отношение затрат на обмен к затратам на вычисления невелико, так как обмены относительно редки; объем обмена в процессе обработки пропорционален линейным размерам изображения, объем вычислений пропорционален его площади;

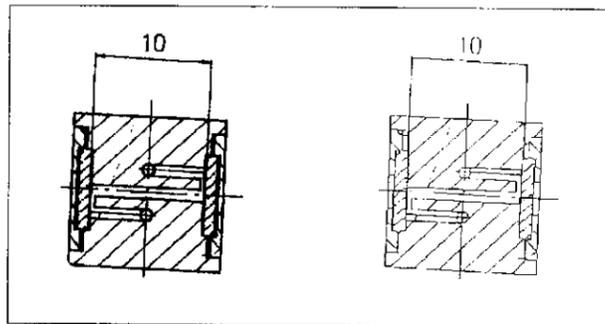


Рис. 6. Бинарное изображение (фрагмент чертежа) и его скелет

— эффективность алгоритма прямо пропорциональна количеству процессоров (если не принимать во внимание затраты на ввод/вывод) вследствие хорошего баланса нагрузки, а также слабой зависимости количества прогонок от количества процессоров.

**5. Программная реализация параллельного алгоритма.** Для проверки разработанного параллельного алгоритма и создания соответствующего математического обеспечения, пригодного для решения практических задач в области скелетизации изображений, этот алгоритм был программно реализован для параллельной рабочей станции PARStation фирмы "Transtech", в состав которой входило четыре транспьютера INMOS T800. В качестве управляющего компьютера использовалась рабочая станция SPARCstation 10. Входные данные представляли собой бинарные изображения в TIFF-формате.

При загрузке входного изображения прежде всего определяются его линейные размеры. Эти данные, а также сведения об общем количестве процессоров распространяются по сети с тем, чтобы подготовить их к приему фрагментов изображения. По мере загрузки строк изображения из внешней памяти они распространяются по сети. В каждом процессоре из потока строк выбираются те, которые предназначены для данного процессора, остальные передаются дальше по сети. Загрузка полученного фрагмента в локальную память процессора производится в определенном порядке, зависящем от номера процессора в квадруплете. Эта процедура делает возможным использование одной и той же программы для всех процессоров, независимо от направления первой прогонки. Во избежание конфликтов при обмене данными между прогонами очередность выдачи и приема данных в каждом процессоре определяется его положением в сети процессоров. После завершения обработки изображение возвращается по сети в управляющий процессор, где осуществляется его запись во внешнюю память в формате TIFF. На рис. 6 представлены бинарное изображение и его скелет — результат работы программы.

В связи с небольшим количеством процессоров, имевшихся в нашем распоряжении, измерение временных затрат проводилось только для кольцевой конфигурации сети (одномерный массив с одним, двумя и четырьмя процессорами). Замеры временных затрат на обработку изображений показали, что при увеличении количества используемых процессоров вдвое производительность возрастала в 1,7—1,8 раза. Отличие этого коэффициента от двойки объясняется, видимо, затратами на ввод/вывод, обмен и другими накладными расходами.

**Заключение.** Разработаны: параллельный алгоритм дистанционного преобразования и скелетизации бинарных изображений, предназначенный для реализации на многопроцессорных вычислительных системах с распределенной памятью; механизм контроля за ходом обработки, позволяющий корректно обработать бинарное изображение за минимально возможное время; метод пересчета дистанционного преобразования в локальную ширину линий. Проверка, проведенная на реальных изображениях, показала его работоспособность и эффективность.

Представленная методика может быть использована для разработки параллельных алгоритмов скелетизации, основанных на дистанционном преобразовании с иной метрикой пространства.

Данная работа выполнена в отделе информатики университета г. Йозенсуу (Финляндия) благодаря содействию и помощи проф. Мартти Пенттонена и Юхи Ойнонена.

#### СПИСОК ЛИТЕРАТУРЫ

1. Levie M. D. Vision in Man and Machine. N. Y.: McGraw-Hill, 1985. Ch. 10.
2. Darwish A. D., Jain A. K. A rule based approach for visual pattern inspection // IEEE Trans. on Pattern Analysis and Machine Intelligence. 1988. PAMI-10, N 1. P. 56.
3. Kaneko T. Line structure extraction from line-drawing images // Pattern Recognition. 1992. 25, N 9. P. 963.
4. Nihomiya T., Yoshimura K., Nomoto M., Nakagava Y. Automatic screen-printed circuit pattern inspection using connectivity preserving image reduction and connectivity comparison // Proc. of 11th International Conf. on Pattern Recognition. 1992. 1. P. 55.
5. Kasturi R., El-Masri W., Shan J., Gattiker J. R., Mokate U. B. A system for interpretation of line drawings // IEEE Trans. on Pattern Recognition and Machine Intelligence. 1990. PAMI-12, N 10. P. 978.
6. Suzuki S., Yamada T. MARIS: map recognition input system // Pattern Recognition. 1989. 23, N 8. P. 919.
7. Toriwaki J. I., Yokoi S. Distance transformations and skeletons of digitized pictures with applications // Progress in Pattern Recognition. Vol. 1 /Ed. L. N. Kanal, A. Rosenfeld. Amsterdam, The Netherlands: North-Holland, 1981. P. 187.
8. Arcelli C., de Baja G. S. A width-independent fast thinning algorithm // IEEE Trans. on Pattern Analysis and Machine Intelligence. 1985. PAMI-7, N 4. P. 463.
9. Arcelli C., de Baja G. S. The one-pass two operation process to detect the skeletal pixels on the 4-distance transform // IEEE Trans. on Pattern Analysis and Machine Intelligence. 1992. PAMI-13, N 4. P. 411.

*Поступила в редакцию 4 октября 1995 г.*