

Спецкурс
"Основы цифровой обработки
изображений."
Часть 1: Математическая морфология"

А.В. Тузиков.

Объединенный институт проблем информатики
Национальной академии наук Беларуси
Факультет прикладной математики и информатики Белгосуниверситета
2024 г.

Содержание

1	Базовые операции математической морфологии для бинарных изображений	3
2	Операции математической морфологии для полутоновых изображений	7
2.1	Свойства полутоновых операций	9
2.2	Бинарный структурирующий элемент	10
2.3	Полутоновой структурирующий элемент	12
3	Геодезические эрозия и расширение	13
4	Дистанционное преобразование и морфологический скелет	15
4.1	Максимальные шары и представление скелетов	15
4.2	Свойства морфологического скелета	18
4.3	Быстрый алгоритм построения скелета и восстановления изображения	19
4.4	Дистанционное преобразование	20
4.5	Двухпроходной алгоритм построения дистанционной функции . .	21
5	Утолщение и утоньшение	22
5.1	Утоньшение и алгоритмы скелетизации	23
6	Морфологический спектр	27
7	Гранулометрия и преобразование отмыкания	30
7.1	Бинарные изображения	30
7.2	Полутоновые изображения	32
8	Фильтры	33
8.1	Порядково-статистические и медианные фильтры	33
8.2	Стековые фильтры	36
8.3	Представление через ядро	37
9	Примеры алгоритмов	40
9.1	Быстрый алгоритм вычисления суммы Минковского	40
9.2	Сложение Минковского выпуклых многоугольников	41
9.3	Быстрое вычисление полутоновых морфологических операций с произвольным бинарным структурирующим элементом	45
9.4	Параллельная реализация морфологических операций для линейных структурирующих элементов	49
9.5	Преобразование водораздела	56

1 Базовые операции математической морфологии для бинарных изображений

Математическая морфология оперирует с бинарными и полутоновыми изображениями. Бинарные изображения задаются подмножествами m -мерного пространства E^m , где E есть евклидово пространство \mathbb{R} либо дискретное пространство \mathbb{Z} и

$$E^m = \{a \mid a = (a_1, a_2, \dots, a_m), a_i \in E, i = 1, \dots, m\}.$$

Полутоновые изображения задаются функциями $f : D \rightarrow V$, где $D \subseteq E^m$ и $V \subseteq E$.

Определение 1 Пусть множества A и B содержатся в E^m . Расширением множества A множеством B называется множество $A \oplus B$ такое, что

$$A \oplus B = \{c \in E^m \mid c = a + b, a \in A, b \in B\}.$$

Операция сложения векторов понимается как покомпонентное сложение, т.е. $a + b = (a_1 + b_1, a_2 + b_2, \dots, a_m + b_m)$. Множество A задает исходное изображение, а множество B называется структурирующим элементом.

Операция расширения известна в геометрии как операция сложения Минковского.

Операция расширения является коммутативной и ассоциативной операцией, т.е. $A \oplus B = B \oplus A$ и $(A \oplus B) \oplus C = A \oplus (B \oplus C)$, в силу коммутативности и ассоциативности операции сложения.

Определение 2 Эрозией множества A структурирующим элементом B называется множество $A \ominus B$ такое, что

$$A \ominus B = \{c \in E^m \mid c + b \in A, \text{ для любого } b \in B\}.$$

Определение 3 Отмыканием и замыканием множества A структурирующим элементом B называются, соответственно, множества

$$A \circ B = (A \ominus B) \oplus B.$$

$$A \bullet B = (A \oplus B) \ominus B.$$

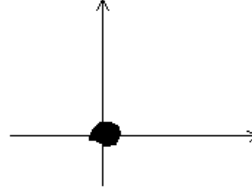
На Рис. 1 показаны результаты выполнения операций расширения, эрозии, отмыкания и замыкания для множеств в \mathbb{Z}^2 (бинарных изображений).

Определение 4 Сдвигом множества A на элемент $b \in E^m$ называется множество A_b , определяемое следующим образом

$$A_b = \{a + b \mid a \in A\}.$$



а)



б)



в)



г)



д)



е)

Рис. 1: Операции математической морфологии для бинарных изображений (множеств в \mathbb{Z}^2): а) исходное изображение A ; б) структурирующий элемент B , содержащий начало координат; в) расширение $A \oplus B$; г) эрозия $A \ominus B$; д) отмыкание $A \circ B$; е) замыкание $A \bullet B$.

Геометрически множество A_b есть сдвиг в E^m множества A на вектор $b \in E^m$.

Обозначим $\check{B} = \{-b \mid b \in B\}$, т.е. \check{B} есть множество, симметричное относительно начала координат множеству B . Дополнение множества A до пространства E^m будем обозначать $A^c = E^m \setminus A$.

Определение 5 Оператор $\psi : X \subseteq E^m \mapsto \psi(X) \subseteq E^m$ называется возрастающим, если из $X \subseteq Y$ следует $\psi(X) \subseteq \psi(Y)$.

Оператор ψ называется экстенсивным (антиэкстенсивным), если $\psi(X) \supseteq X$ (соответственно, $\psi(X) \subseteq X$) для любого $X \subseteq E^n$.

Приведем основные свойства операций математической морфологии.

1.

$$A \oplus B = \{x \mid \check{B}_x \cap A \neq \emptyset\}, \quad A \ominus B = \{x \mid B_x \subseteq A\},$$

$$A \oplus B = \bigcup_{b \in B} A_b, \quad A \ominus B = \bigcap_{b \in B} A_{-b}.$$

Первые две формулы иллюстрируются на Рис. 2. Последние формулы говорят о том, что расширение и эрозия множества A структурирующим элементом B есть, соответственно, объединение сдвигов множества A на элементы b из B и пересечение сдвигов A на элементы $-b$ из B .

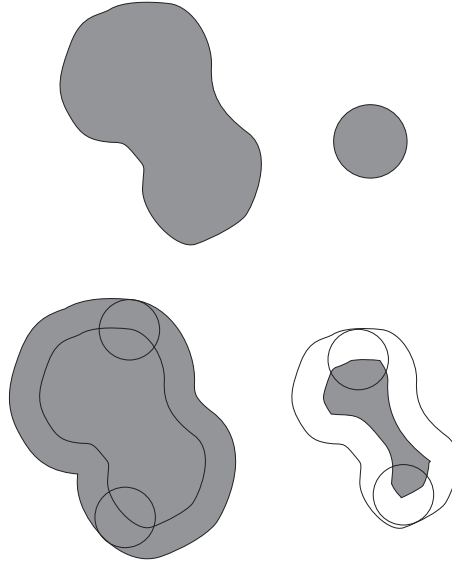


Рис. 2: Расширение и эрозия бинарного изображения круглым структурирующим элементом, содержащим начало координат в центре

2.

$$A_x \oplus B = (A \oplus B)_x,$$

$$A_x \oplus B_{-x} = A \oplus B,$$

$$A_x \ominus B = (A \ominus B)_x,$$

$$A \ominus B_x = (A \ominus B)_{-x}.$$

Следовательно, операция расширения является инвариантной относительно сдвига изображения и структурирующего элемента (в силу коммутативности расширения). Эрозия является инвариантной относительно сдвига изображения.

3. Если $0 \in B$, то $A \subseteq A \oplus B$ и $A \ominus B \subseteq A$.

4. Расширение и эрозия являются возрастающими операциями. Справедливы следующие соотношения для любого множества D :

$$\text{из } A \subseteq B \text{ следует } A \oplus D \subseteq B \oplus D,$$

$$\text{из } A \subseteq B \text{ следует } A \ominus D \subseteq B \ominus D,$$

$$\text{из } A \supseteq B \text{ следует } D \ominus A \subseteq D \ominus B.$$

5. Расширение является перестановочным с объединением, а эрозия – с пересечением изображений (множеств).

$$(A \cup B) \oplus C = (A \oplus C) \cup (B \oplus C),$$

$$(A \cap B) \oplus C \subseteq (A \oplus C) \cap (B \oplus C),$$

$$(A \cap B) \ominus C = (A \ominus C) \cap (B \ominus C),$$

$$(A \cup B) \ominus C \supseteq (A \ominus C) \cup (B \ominus C),$$

$$A \oplus (B \cap C) \subseteq (A \oplus B) \cap (A \oplus C),$$

$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C),$$

$$A \ominus (B \cup C) = (A \ominus B) \cap (A \ominus C),$$

$$A \ominus (B \cap C) \supseteq (A \ominus B) \cup (A \ominus C).$$

6. Отмыкание множества A структурирующим элементом B есть объединение всех сдвигов структурирующего элемента, при которых он содержится внутри A (см. Рис. 3), т.е.

$$A \circ B = \bigcup_{\{y \mid B_y \subseteq A\}} B_y.$$

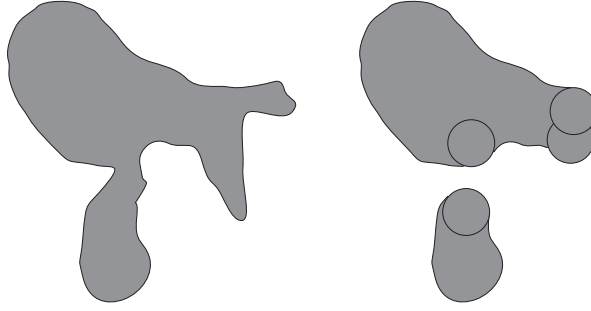


Рис. 3: Отмыкание бинарного изображения круглым структурирующим элементом

7.

$$A \bullet B = \{x \in E^m \mid x \in \check{B}_y \text{ влечет } \check{B}_y \cap A \neq \emptyset\} = \bigcap_{\{y \mid \check{B}_y \cap A = \emptyset\}} \check{B}_y^c.$$

8. Отмыкание множества A содержится в A , а замыкание содержит A , т.е.

$$A \circ B \subseteq A \subseteq A \bullet B.$$

Другими словами, отмыкание является антиэкстенсивной, а замыкание экстенсивной операцией.

9. Расширение и эрозия, отмыкание и замыкание являются двойственными операциями, т.е.

$$\begin{aligned} (A \ominus B)^c &= A^c \oplus \check{B}, \\ (A \bullet B)^c &= A^c \circ \check{B}. \end{aligned}$$

2 Операции математической морфологии для полутоновых изображений

Будем предполагать, что функция f определена на подмножестве пространства E^m , а множеством значений функции f является пространство $\bar{E} = E \cup \{-\infty, \infty\}$, где $E = \mathbb{Z}$ или $E = \mathbb{R}$. Бинарные изображения являются подмножествами из E^m , а полутоновые изображения – функциями из E^m в $E \cup \{-\infty, \infty\}$.

Будем использовать в дальнейшем два представления функций:

1. через сечения функции;
2. через подграфик (тень) функции.

Определим сечение $T_t(f)$, где $t \in E$, функции f следующим образом:

$$T_t(f) = \{x \in E \mid f(x) \geq t\}. \quad (1)$$

Рассматривая различные уровни $t \in E$, можем сопоставить функции f множество сечений $T_t(f)$. Множество сечений однозначно задает функцию f .

Теорема 1 1. Сечения $T_t(f)$, $t \in E$ функции f убывают с возрастанием t , т.е. для любых $t_1, t_2 \in E$

$$\text{из } t_1 < t_2 \text{ следует } T_{t_2}(f) \subseteq T_{t_1}(f). \quad (2)$$

2. Если $E = \mathbb{R}$, то сечения обладают свойством монотонной непрерывности, т.е.

$$T_t(f) = \bigcap_{r < t} T_r. \quad (3)$$

3. Функция $f(x)$ может быть восстановлена по сечениям следующим образом:

$$f(x) = \max\{t \in E \mid x \in T_t(f)\}, \quad x \in E^m. \quad (4)$$

4. Пусть $\{G_t \subseteq E^m, t \in E\}$ – любой набор множеств, удовлетворяющий условиям (2) и (3), если $E = \mathbb{R}$. Определим функцию $g(x) = \sup\{t \in E \mid x \in G_t\}$, $x \in E^m$. Тогда $T_t(g) = G_t$ для любого $t \in E$.

Формула (4) позволяет функции f принимать бесконечные значения. Так, если для некоторой точки x_0 справедливо $\{t \in E \mid x \in G_t\} = \emptyset$ или $\{t \in E \mid x \in G_t\} = V$, то мы полагаем $f(x_0) = \sup \emptyset = -\infty$ и $f(x_0) = \sup E = \infty$, соответственно.

Существует другое представление функции f через тень (подграфик) $U(f)$, определяемую следующим образом:

$$U(f) = \{(x, t) \in E^m \times E \mid f(x) \geq t\}. \quad (5)$$

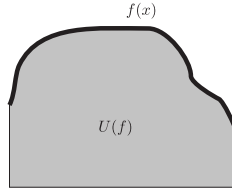


Рис. 4: Подграфик функции

Тень $U(f)$ однозначно характеризует функцию f , т.е.

$$f(x) = \sup\{t \mid (x, t) \in U(f)\}, \quad x \in E^m. \quad (6)$$

Будем говорить, что функция f не больше функции g , т.е. $f \leq g$, тогда и только тогда, когда $f(x) \leq g(x)$ для любых $x \in E^m$.

Справедливо соотношение, связывающее функцию, тень и сечения функции

$$f \leq g \Leftrightarrow U(f) \subseteq U(g) \Leftrightarrow T_t(f) \subseteq T_t(g) \text{ для любых } t \in E.$$

Введем операции расширения и эрозии для полутоновых изображений.

Определение 6 *Расширением и эрозией функции f функцией (структурирующим элементом) g назовем, соответственно, функции $f \oplus g$ и $f \ominus g$ такие, что*

$$U(f \oplus g) = U(f) \oplus U(g), \quad U(f \ominus g) = U(f) \ominus U(g).$$

В непрерывном случае данное определение является корректным только для функций, полунепрерывных сверху. Для произвольных f и g множество $U(f) \oplus U(g)$ может и не быть подграфиком функции. В этом случае за $U(f \oplus g)$ принимается по определению наименьший подграфик, содержащий $U(f) \oplus U(g)$. Аналогично определяются остальные базовые операции.

Обозначим через $S_{pt}(f) = \{x \in E^m \mid f(x) \neq -\infty\}$, а через $\check{f}(x)$ функцию такую, что $\check{f}(x) = f(-x)$, $x \in E^m$.

Из определения расширения и эрозии можно получить формулы, более удобные в вычислительном отношении.

$$(f \oplus g)(x) = \sup_{y \in (S_{pt}(f) \cap (S_{pt}(\check{g}) + x))} \{f(y) + g(x - y)\}, \quad x \in E^m. \quad (7)$$

$$(f \ominus g)(x) = \begin{cases} \inf_{y \in S_{pt}(g) + x} \{f(y) - g(y - x)\}, & \text{если } (S_{pt}(g) + x) \cap S_{pt}(f) \neq \emptyset, \\ -\infty & \text{в противном случае.} \end{cases} \quad (8)$$

2.1 Свойства полутоновых операций

Рассмотрим случай, когда множество значений функции f есть E (где $E = \mathbb{R}$ или $E = \mathbb{Z}$), т.е. исключаем точки $-\infty$ и ∞ и предположим, что все сечения функции являются замкнутыми множествами (если $E = \mathbb{Z}$, то для любой функции f сечения являются замкнутыми множествами). Известно, что соответствующие таким сечениям функции являются полунепрерывными сверху. В этом случае мы можем заменить супремум и инфимум на максимум и минимум и переписать формулы (7) и (8) в следующем виде.

Пусть функции $f(x)$ и $g(x)$ есть $f : F \rightarrow E$, $g : G \rightarrow E$, где $F, G \subseteq E^m$, тогда

$$(f \oplus g)(x) = \max_{z \in G, x-z \in F} \{f(x-z) + g(z)\}. \quad (9)$$

$$(f \ominus g)(x) = \min_{z \in G, x+z \in F} \{f(x+z) - g(z)\}. \quad (10)$$

Для расширения и эрозии справедливы следующие свойства:

1. коммутативность $f \oplus g = g \oplus f$,
2. ассоциативность $(f \oplus g_1) \oplus g_2 = f \oplus (g_1 \oplus g_2)$,
3. $(f \ominus g_1) \ominus g_2 = f \ominus (g_1 \oplus g_2)$,

4. сопряженность расширения и эрозии $g \leq f \ominus k \Leftrightarrow f \geq g \oplus k$,
5. двойственность $-(f \oplus g)(x) = ((-f) \ominus \check{g})(x)$, где $\check{g}(x) = g(-x)$.

Определение 7 Пусть $f : F \rightarrow E$ и $g : G \rightarrow E$. Отмыкание и замыкание функции f структурирующим элементом g обозначаются $f \circ g$ и $f \bullet g$, соответственно, и определяются следующим образом

$$f \circ g = (f \ominus g) \oplus g, \quad f \bullet g = (f \oplus g) \ominus g.$$

Для отмыкания и замыкания справедливы следующие свойства.

1. $(f \circ g)(x) \leq f(x)$ для любого $x \in F \circ G$,
2. $f(x) \leq (f \bullet g)(x)$ для любого $x \in F$,
3. $(f \circ g) \circ g = f \circ g$, $(f \bullet g) \bullet g = f \bullet g$.

Это свойство говорит о том, что операции отмыкания и замыкания являются идемпотентными, т.е. повторное применение отмыкания и замыкания не изменяют результат.

4. двойственность $-(f \circ g) = (-f) \bullet \check{g}$,
5. $(f \circ g)(x) = \max\{t \mid (x, t) \in \bigcup_{\{y \in E^m \times E \mid U(g)_y \subseteq U(f)\}} U(g)_y\}$.

Аналогично операциям отмыкания и замыкания для бинарных изображений можно дать геометрическую интерпретацию этим операциям для полутоновых изображений. Для получения отмыкания структурирующий элемент $f(x)$ располагаем как можно ближе к графику функции. При этом он скользит вдоль возрастания переменной x . Огибающая графика структурирующего элемента дает отмыкание функции $f(x)$. При этом, в зависимости от формы структурирующего элемента происходит сглаживание функции $f(x)$, если структурирующий элемент имеет достаточно гладкую форму. На основании свойства двойственности аналогичную геометрическую интерпретацию можно дать замыканию.

2.2 Бинарный структурирующий элемент

Рассмотрим случай, когда структурирующий элемент является бинарным, т.е. задается множество B (в этом случае можно положить $g(x) = 0$, если $x \in B$). Тогда формулы (9) и (10) можно переписать в виде

$$(f \oplus B)(x) = \max\{f(x - z), z \in B\}, \quad (11)$$

$$(f \ominus B)(x) = \min\{f(x + z), z \in B\}, \quad (12)$$

$$f \circ B = (f \ominus B) \oplus B, \quad (13)$$

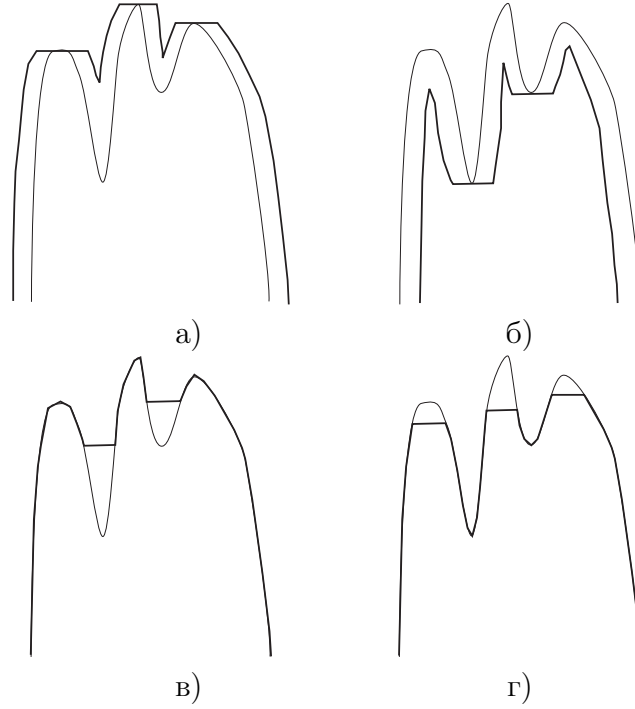


Рис. 5: Жирной линией выделены расширение а), эрозия б), замыкание в) и отмыкание г) функции (одномерного полутонового изображения) бинарным структурирующим элементом.

$$f \bullet B = (f \oplus B) \ominus B. \quad (14)$$

Применение операций показано на Рис. 5.

Далее мы также покажем, каким образом вычисление расширения и эрозии, а также ряда других преобразований для дискретных изображений можно свести к вычислению бинарных морфологических операций.

Пусть дискретное полутоновое изображение задается двумерной функцией $f(m, n)$, принимающей значения $t = 0, 1, 2, \dots, A$. При сечении функции f для возможных уровней амплитуды t , $0 \leq t \leq A$ получаем бинарные изображения

$$T_t(f)[m, n] = \begin{cases} 1, & \text{если } f(m, n) \geq t, \\ 0, & \text{если } f(m, n) < t \end{cases}$$

Нетрудно видеть, что для любых (m, n)

$$f(m, n) = \max\{t \mid T_t(f)[m, n] = 1\} = \sum_{t=1}^A T_t(f)[m, n].$$

Будем говорить, что преобразование ψ полутонового изображения в полутоновое изображение перестановочно с сечением, если $\psi[T_t(f)] = T_t(\psi(f))$ для любого уровня амплитуды t . Примерами преобразований, перестановочных с

сечением, являются (11), (12), (13), (14). Из условия перестановочности с сечением преобразования ψ следует, что

$$\psi(f) = \sum_{t=1}^A \psi(T_t(f)) \quad (15)$$

для любого входного изображения f . В этом случае для вычисления преобразования $\psi(f)$ достаточно найти все сечения входного изображения f , вычислить преобразование ψ для этих сечений и найти сумму. При вычислении расширения и эрозии для каждого сечения необходимо выполнить бинарные операции расширения и эрозии, что существенно проще этих операций для полутоновых изображений.

Приведем примеры преобразований, перестановочных с сечением.

1. Морфологическое выделение границ.

Пусть B – двумерное симметричное структурирующее множество, содержащее начало координат. Пусть

$$ED_1(f) = f - (f \ominus B), \quad ED_2(f) = (f \oplus B) - (f \ominus B),$$

где минус обозначает поточечную разность двух функций. Преобразования $ED_1(f)$ и $ED_2(f)$ являются простыми, но достаточно эффективными детекторами границ для полутоновых изображений.

2. Выделение пиков и впадин

Пусть

$$PE(f) = f - (f \circ B),$$

где B – двумерный структурирующий элемент. Это преобразование выделяет пики входной функции f , основания которых не содержат B . Остальные пики функции f удаляются.

Аналогично для нахождения впадин функции f применяется

$$PV(f) = (f \bullet B) - f.$$

Линейная комбинация преобразований, перестановочных с сечением, является преобразованием, для вычисления которого можно использовать формулу (15).

2.3 Полутоновый структурирующий элемент

Рассмотрим операцию расширения полутонového изображения полутоновым структурирующим элементом $f \oplus g$. Это преобразование не является перестановочным с сечением и для его вычисления нельзя применить формулу (15).

Однако в этом случае существует алгоритм, сводящий вычисление полутонového расширения для дискретных изображений к выполнению достаточно большого числа бинарных операций.

1. Определим наибольшее значение в изображении и структурирующем элементе. Пусть они соответственно равны $P \geq 1$ и $Q \geq 1$.
2. Декомпозируем изображение на P бинарных изображений, используя сечения для каждого уровня от 1 до P . Декомпозируем структурирующий элемент на $Q + 1$ бинарных структурирующих элементов, используя сечения для каждого уровня от 0 до Q .
3. Выполним бинарное расширение для пар сечений изображения и структурирующего элемента в следующей последовательности:
для сечения изображения с номером k , $k = P, P - 1, \dots, 1$ выполняются расширения сечениями структурирующего элемента с номерами $Q, Q - 1, \dots, 0$, т.е.

$$T_{k,l} = T_k(f) \oplus T_l(g), \quad l = Q, Q - 1, \dots, 0.$$
Полученные результаты $T_{k,l}$ для сечения с номером k поточечно складываются и к ненулевым значениям добавляется k . В результате будет получено изображение f_k .
4. На основе результатов предыдущего шага формируем выходное изображение $f \oplus g$, значением которого в каждой точке x является $\max_k \{f_k(x)\}$.

3 Геодезические эрозия и расширение

Пусть X – подмножество из \mathbb{R}^m , $x, y \in X$. Предположим, что в \mathbb{R}^m определены понятия пути и длины пути между выбранными точками. Геодезическим расстоянием $d_X(x, y)$ в X между точками x и y называется длина наиболее короткого пути $l(x, y)$, лежащего в X и соединяющего эти точки:

$$d_X(x, y) = \inf \{l(x, y)\}.$$

Пусть Y – некоторое множество, содержащееся в X . Тогда

$$R_X(Y) = \{x \in X : \exists y \in Y, d_X(x, y) < \infty\}$$

называется X -реконструкцией множества Y (см. Рис. 6).

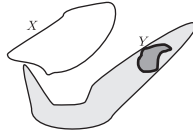


Рис. 6: Реконструкция $R_X(Y)$ множества Y в множестве X , состоящем из двух связных компонент, показана светло-серым цветом.

Предположим, что Y состоит из n связных компонент Y_i , $i = 1, 2, \dots, n$. Геодезической зоной влияния $Z_X(Y_i)$ компоненты Y_i называется множество

$$Z_X(Y_i) = \{x \in X : d_X(x, y) < \infty, \forall j \neq i, d_X(x, Y_i) < d_X(x, Y_j)\}. \quad (16)$$

Границы между различными зонами влияния дают геодезический скелет $SKIZ_X(Y)$ зон влияний Y в X :

$$IZ_X(Y) = \bigcup_i Z_X(Y_i),$$

$$SKIZ_X(Y) = X \setminus IZ_X(Y).$$

Геодезические расширение (Рис. 7) множества $Y \subseteq X$ и эрозия множества $Y \supseteq X$ структурирующим элементом B определяются следующим образом:

$$\delta_X(Y) = (Y \oplus B) \cap X, \quad \epsilon_X(Y) = X \cup (Y \ominus B).$$

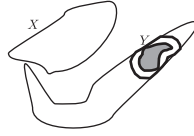


Рис. 7: Геодезическое расширение $\delta_X(Y)$ круглым структурирующим элементом увеличивает множество Y внутри множества X .

Геодезические расширение функции $f \leq g$ и эрозия функции $f \geq g$ с помощью плоского структурирующего элемента B определяются как:

$$\delta_g(f) = \min\{f \oplus B, g\}, \quad \epsilon_g(f) = \max\{f \ominus B, g\}.$$

Введем обозначения

$$\delta_f^{(n)}(g) = \delta_f(\delta_f(\dots \delta_f(g)) \dots),$$

$$\epsilon_f^{(n)}(g) = \epsilon_f(\epsilon_f(\dots \epsilon_f(g)) \dots)$$

(здесь геодезическое расширение и эрозия применяются, соответственно, n раз).

Теперь мы можем определить еще одну реконструкцию. Пусть имеются две функции $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$, такие что, $f \geq g$, тогда

$$r_f(g) = \lim_{n \rightarrow \infty} \delta_f^{(n)}(g).$$

Например, преобразование

$$\phi(f) = r_f(f \ominus H)$$

позволяет выделять светлые области на полутоновых изображениях.

4 Дистанционное преобразование и морфологический скелет

Скелет является одной из форм геометрического представления изображений. Скелет представляет из себя фигуру, содержащую информацию об изображении (размере, ориентации, форме, связности). Скелет содержит обычно существенно меньше элементов, чем само изображение, но позволяет полностью восстанавливать изображение. Поэтому использование скелетов в сочетании с эффективным кодированием обеспечивает значительное сжатие информации.

4.1 Максимальные шары и представление скелетов

Пусть $X \subseteq \mathbb{R}^m$ – непрерывное изображение. Шар $rB_a \subseteq X$ радиуса r с центром в точке $a \in X$ называется максимальным, если он не содержится полностью в другом шаре, содержащемся в X (т.е. если не существует шара sB_b такого, что $rB_a \subset sB_b \subseteq X$). Здесь B – шар единичного радиуса в некоторой метрике.

Можно рассматривать шары в обобщенном смысле, взяв в качестве B открытый выпуклый структурирующий элемент, содержащий начало координат.

Максимальный шар должен касаться границы объекта как минимум в двух различных точках.

Аналогичное понятие можно ввести для дискретных изображений, если определить понятие шара в дискретном пространстве. Это можно сделать посредством фиксированного структурирующего элемента. Пусть B – структурирующий элемент. Положим, что B имеет радиус, равный 1. Структурирующий элемент $B(n)$ (другое обозначение nB) радиуса n определим следующим образом

$$B(n) = \underbrace{B \oplus B \oplus \dots \oplus B}_{n \text{ раз}}.$$

Будем предполагать, что множество B ограничено и содержит начало координат 0. При $n = 0$ под $B(n)$ понимается $\{0\}$. Структурирующий элемент единичного радиуса задает единичную окрестность на дискретной решетке. В обработке изображений часто используются единичные окрестности, соответствующие четырех и восьмисвязным квадратным решеткам. Примеры шаров единичного и больших радиусов показаны на Рис. 8.

Существуют другие способы определения шаров в дискретном пространстве, связанные с использованием дискретной метрики. Примером служит чемферная метрика, которая дает хорошую аппроксимацию евклидовой метрики в дискретном пространстве.

Определение 8 Скелетом $SK(X)$ непрерывного или дискретного изображения $X \subseteq E^m$ называется множество центров максимальных шаров, содержащихся в X .

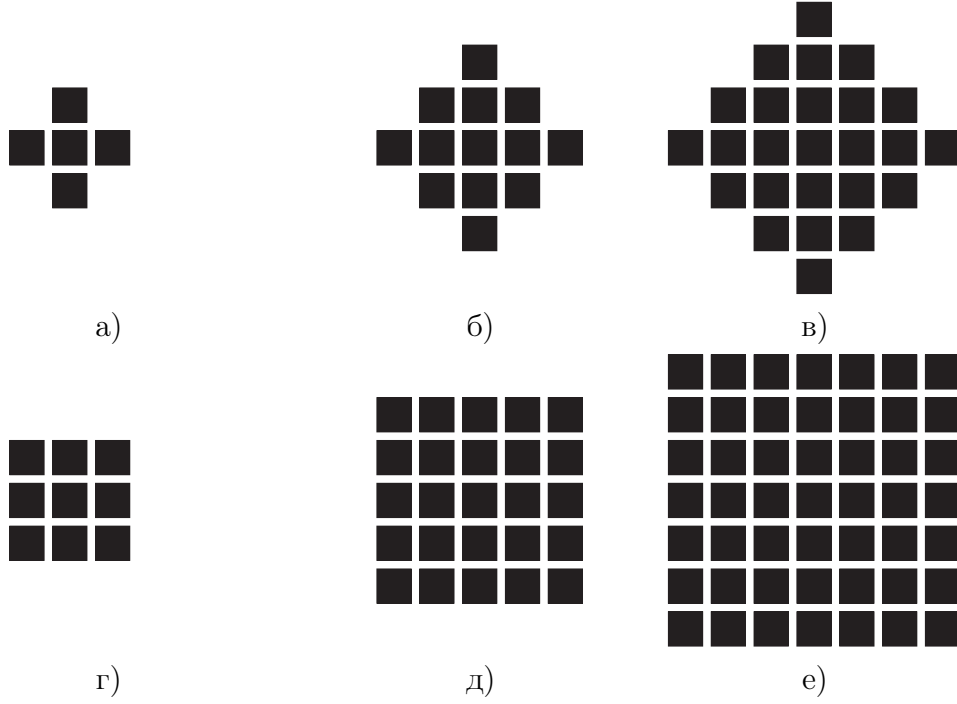


Рис. 8: Структурирующие элементы единичного радиуса, задающие локальную окрестность на четырехсвязной и восьмисвязной квадратной решетке а) и г); соответствующие структурирующие элементы радиуса 2 б) и д); структурирующие элементы радиуса 3 в) и е).

Пример максимального шара показан на Рис. 9. Обозначим через $S_r(X)$, $r > 0$, r -е скелетное подмножество, т.е. множество центров максимальных шаров радиуса r .

Скелет $SK(X)$ ограниченного открытого непустого множества $X \subset \mathbb{R}^m$ задается формулой

$$SK(X) = \bigcup_{r>0} S_r(X) = \bigcup_{r>0} [(X \ominus rB) - ((X \ominus rB) \circ drB)].$$

Здесь drB – “замкнутый шар бесконечно малого радиуса dr ”, а отмыкание множества Y структурирующим элементом drB есть $Y \circ drB = \bigcup_{s>0} (Y \circ sB)$.

Рассмотрим теперь дискретный случай. Пусть множество X представляет дискретное бинарное изображение из \mathbb{Z}^m . Будем предполагать, что множество X является непустым и ограниченным.

Алгоритм Серра построения морфологического скелета $SK(X)$ (или $SK(X, B)$ для большей определенности) дискретного бинарного изображения X структурирующим элементом B имеет вид

$$S_n(X) = (X \ominus B(n)) - ((X \ominus B(n)) \circ B), \quad n = 0, 1, \dots, N,$$

$$SK(X) = \bigcup_{n=0}^N S_n(X), \quad (17)$$

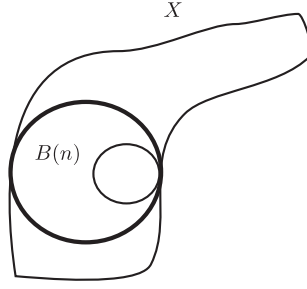


Рис. 9: Максимальный шар $B(n)$ содержит вписанный в X немаксимальный шар.

где $N = \max\{n \mid X \ominus B(n) \neq \emptyset\}$. Следует помнить, что скелетные подмножества имеют метки, равные номерам этих подмножеств. Знание номеров скелетных подмножеств позволяет восстанавливать исходное бинарное изображение. Изображение X восстанавливается по скелетным множествам следующим образом

$$X = \bigcup_{n=0}^N [S_n(X) \oplus B(n)]. \quad (18)$$

Этот алгоритм является непосредственным аналогом формулы для непрерывного случая.

Очевидно, что различные структурирующие элементы дают различные скелеты, при этом несимметричные структурирующие элементы дают скелеты, которые абсолютно не напоминают осей симметрии объекта. Но каждый скелет содержит достаточно информации для полного восстановления изображения. На Рис. 10 показаны структурирующие элементы, использованные для построения скелета одного и того же бинарного изображения Рис. 11а).

1	1	1	1	1	0	0	1	0	0	0	1	1	0	0	
1	1	1	1	1	0	1	1	1	0	0	1	1	1	0	
1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	
1	1	1	1	1	0	0	1	0	0	0	0	1	1	0	
B_1					B_2					B_3					

Рис. 10: Структурирующие элементы, использованные для построения скелета бинарного изображения, показанного на Рис. 11а). Жирным шрифтом выделен центр элемента.

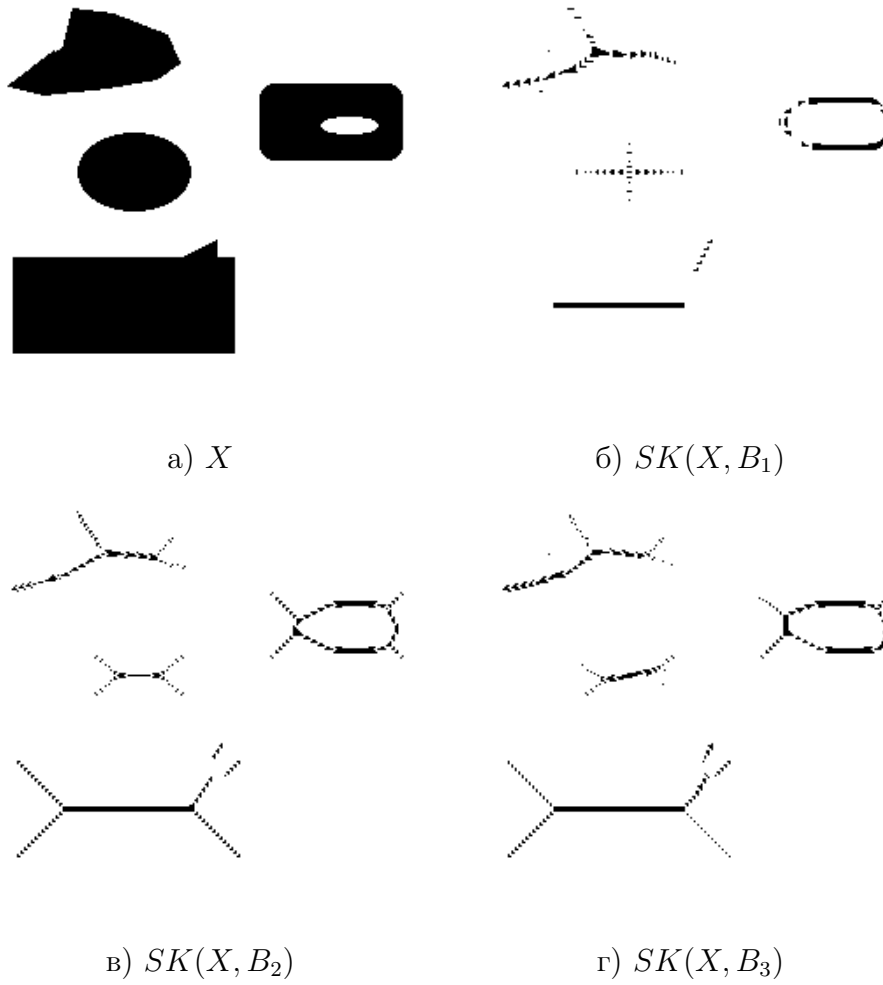


Рис. 11: Бинарное изображение и скелеты, полученные для структурирующих элементов, показанных на Рис. 10.

4.2 Свойства морфологического скелета

Предположим, что дискретный структурирующий элемент B содержит начало координат.

Свойство 1

1. Если $B = \{0\}$, то $SK(X) = \emptyset$;
2. Если $X \ominus B = \emptyset$, то $SK(X) = X$;
3. Различные скелетные множества $S_n(X)$ не пересекаются;
4. Преобразование получения морфологического скелета является инвариантным относительно сдвига изображения, антиэкстенсивным и идемпотентным.

Последние два свойства справедливы и для непрерывного случая.

Теорема 2 *Если скелет изображения X получен с помощью ограниченного выпуклого структурирующего элемента B , то точка z принадлежит n -му скелетному подмножеству $S_n(X)$ тогда и только тогда, когда шар $B(n)_z$ является максимальным в X .*

Теорема 3 *Если скелет изображения X получен с помощью ограниченного выпуклого структурирующего элемента B , то X эквивалентно отмыканию $X \circ kB$ тогда и только тогда, когда первые k скелетных подмножеств $S_n(X)$, $n = 0, 2, \dots, k-1$ являются пустыми.*

4.3 Быстрый алгоритм построения скелета и восстановления изображения

Опишем алгоритм, позволяющий строить скелет и восстанавливать изображение за линейное время относительно размера исходного изображения.

Пусть Y_1, Y_2, Y_3 – три множества.

Алгоритм 1 (Алгоритм построения скелета)

Шаг 1. $n := 0$, $Y_1 := X$.

Шаг 2. $Y_2 := Y_1 \ominus B$.

Шаг 3. Если $Y_2 = \emptyset$, то $N := n$, $S_N(X) = Y_1$ и стоп.

Шаг 4. $Y_3 := Y_2 \oplus B$.

Шаг 5. $S_n(X) = Y_1 - Y_3$.

Шаг 6. $n := n + 1$, $Y_1 := Y_2$ и переход к шагу 2.

Этот алгоритм требует выполнения только N эрозий и N расширений.

Алгоритм 2 (Алгоритм восстановления изображения)

Шаг 1. $n := N$, $A := \emptyset$.

Шаг 2. $A := A \cup S_n(X)$.

Шаг 3. Если $n = 0$, то стоп. В противном случае $A := A \oplus B$.

Шаг 4. $n := n - 1$ и переход к шагу 2.

Алгоритм восстановления изображения кратко можно записать в следующем виде

$$X = [[\dots [[S_n(X) \oplus B] \cup S_{n-1}(X)] \oplus B] \cup S_{n-2} \dots] \oplus B] \cup S_0(X). \quad (19)$$

При этом быстроедействие приведенных алгоритмов можно еще повысить за счет декомпозиции структурирующего элемента на более простые элементы.

Использование всех скелетных подмножеств гарантирует точное восстановление исходного изображения. Если же некоторые скелетные подмножества опущены, то изображение будет частично восстановлено.

Справедливы следующие формулы

$$X \ominus B(k) = \bigcup_{n=k}^N [S_n(X) \oplus B(n-k)], \quad 0 \leq k \leq N \quad (20)$$

$$X \circ B(k) = \bigcup_{n=k}^N [S_n(X) \oplus B(n)], \quad 0 \leq k \leq N \quad (21)$$

4.4 Дистанционное преобразование

Дистанционное преобразование множества $X \subseteq \mathbb{Z}^m$ определяется следующим образом:

$$D(p) = \begin{cases} n, & \text{если } B(n-1)_p \subseteq X \text{ и } B(n)_p \not\subseteq X, \\ 0, & \text{если } p \in X^c. \end{cases}$$

Таким образом, $D(p)$ есть максимальный радиус n такой, что шар $B(n-1)$, сдвинутый в точку p , целиком содержится в X , а шар $B(n)_p$ не содержится целиком в X . Примеры вычисления дистанционного преобразования (функции) приведены на Рис. 12. Алгоритм вычисления дистанционной функции, приведенный в этом разделе, относится к случаю, когда расстояние задается бинарным структурирующим элементом. Случаи псевдоевклидова и евклидова расстояния не рассматриваются.

Скелет множества X может быть определен следующим образом

$$SK(X) = \{p \in X \mid B(D(p)-1)_p \not\subseteq B(D(q)-1)_q \subseteq X \text{ для любого } q \neq p\}.$$

Обозначим $B^* = B \setminus \{0\}$.

Справедливы следующие утверждения:

Утверждение 1

$$D(p) = \begin{cases} \min\{D(p+b), b \in B^*\} + 1, & \text{если } p \in X, \\ 0, & \text{если } p \in X^c. \end{cases}$$

При вычислении дистанционной функции в точке p достаточно, чтобы уже была вычислена дистанционная функция во всех точках $D(p+b)$, где $b \in B^*$. В зависимости от вида структурирующего элемента B иногда возможен такой выбор направления сканирования изображения, что вычисление дистанционной функции происходит за один просмотр области, содержащей изображение.

Утверждение 2 Пусть $B = B_1 \cup B_2, B_1 \cap B_2 = \emptyset, 0 \in B_1$, и $D_1(p)$ – дистанционное преобразование, соответствующее структурирующему элементу B_1 . Тогда имеем

$$D(p) = \min\{D_1(p), D(p+b) + 1, \text{ для всех } b \in B_2\}.$$

Утверждение 3

$$SK(X) = \{p \in X \mid \text{для любых } b \in B^*, D(p+b) \leq D(p)\}.$$

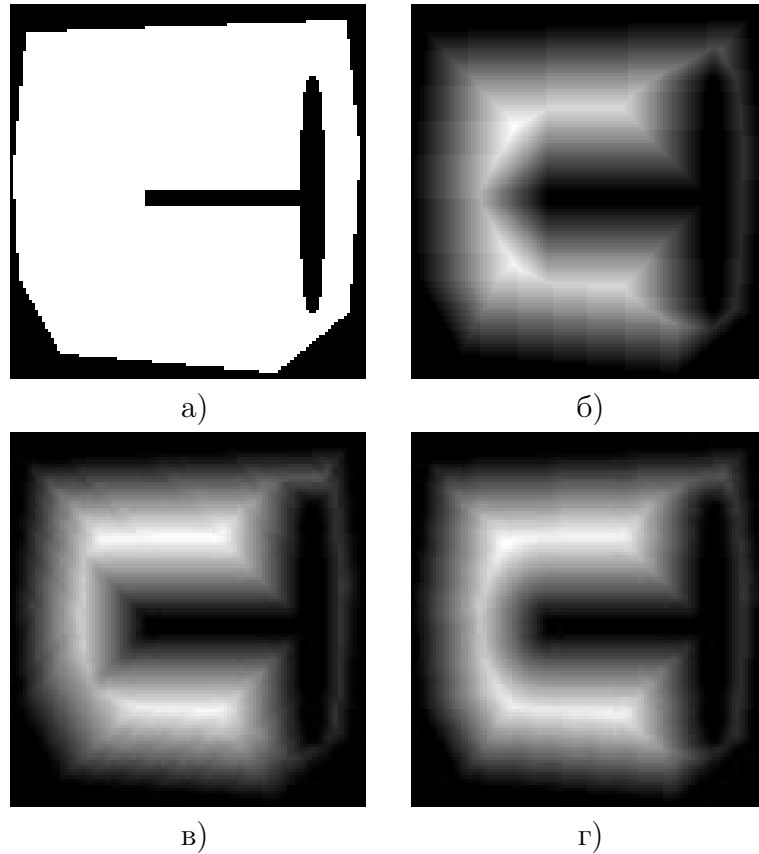


Рис. 12: Исходное бинарное изображение а) и дистанционная функция для четырехсвязного б), восьмисвязного в) евклидова расстояния.

4.5 Двухпроходной алгоритм построения дистанционной функции

Выберем направление сканирования изображения так, чтобы каждая его точка просматривалась один раз. Произведем вычисление дистанционной функции в точке p . Пусть структурирующий элемент B перенесен в точку p , т.е. в точку p перенесен центр 0 структурирующего элемента. Отнесем к $B_1(p)$ все точки из B_p , которые просматриваются до точки p , а также саму точку p .

Оставшиеся точки отнесем к $B_2(p)$. Дистанционная функция D_1 , соответствующая структурирующему элементу B_1 , может быть вычислена за один просмотр изображения. На следующем просмотре изображения, используя второе направление сканирования, вычисляем полную дистанционную функцию D .

Рассмотрим двумерный случай. Пусть координаты точки $b \in B$ есть (x_b, y_b) . При наиболее распространенном направлении сканирования (слева направо и сверху вниз) B декомпозируется следующим образом:

1. если $y_b > 0$, то $b \in B_1$;
2. если $y_b < 0$, то $b \in B_2$;

3. если $y_b = 0$ и $x_b \leq 0$, то $b \in B_1$, в противном случае $b \in B_2$.

В соответствии с утверждением 3 скелет изображения $SK(X)$ может быть построен из дистанционной функции D за один ее просмотр. Если число точек области изображения равно N^2 , а структурирующий элемент содержит m точек, тогда сложность алгоритмов вычисления дистанционной функции и скелета есть $O(N^2m)$.

В тоже время сложность алгоритма 1 быстрой скелетизации, описанного в разделе 4.3, есть $O(N^2mk)$, где k - максимальное значение дистанционной функции для изображения.

5 Утолщение и утоньшение

Рассмотренные ранее операции (расширение, эрозия, отмыкание, замыкание) являются возрастающими. Существует также большой класс морфологических преобразований, не являющихся возрастающими.

Определение 9 Пусть A и B - два непересекающихся структурирующих элемента. Определим операцию \otimes следующим образом:

$$X \otimes (A, B) = \{x \in E^m \mid A_x \subseteq X, B_x \subseteq X^c\}.$$

Нетрудно убедиться, что справедливо соотношение

$$X \otimes (A, B) = (X \ominus A) \cap (X^c \ominus B) = (X \ominus A) \setminus (X \oplus \check{B}).$$

Операция \otimes применяется тогда, когда нужно найти внутри объекта точки с заданными геометрическими свойствами, например, изолированные точки, граничные точки, условные точки. При этом выбор (A, B) определяется тем, что требуется найти в объекте.

Введем еще две операции: утолщение \odot и утоньшение \bigcirc .

Определение 10 Пусть $T = (A, B)$. Тогда

$$X \odot T = X \bigcup (X \otimes T), \quad X \bigcirc T = X \setminus (X \otimes T).$$

Нетрудно видеть, что $X \otimes T \subseteq X$, если $0 \in A$; $X \otimes T = \emptyset$, если $A \cap B \neq \emptyset$ и $X \otimes T \subseteq X^c$, если $0 \in B$. Поэтому операция утолщения является нетривиальной, если $A \cap B = \emptyset$ и $0 \notin A$, а операция утоньшения является нетривиальной, если $A \cap B = \emptyset$ и $0 \notin B$.

Операции утолщения и утоньшения являются двойственными и для них справедливо следующее соотношение

$$X^c \odot (A, B) = (X \bigcirc (B, A))^c.$$

Покажем в качестве примера, как операции утолщения и утоньшения можно применять при построении 45-выпуклой оболочки изображения и построения скелета (не морфологического) изображения.

Пусть изображение X задается множеством на прямоугольной решетке. Назовем 45-выпуклой оболочкой множества X пересечение всех полуплоскостей, содержащих X и имеющих угол наклона с осью Ox 0, 45, 90, 135, 180, 225, 270 градусов (если рассматривается пересечение всех полуплоскостей, то имеем выпуклую оболочку множества).

Рассмотрим структурирующие элементы T_1, T_2, \dots, T_8 , показанные на Рис. 13. Соответствующие каждому структурирующему элементу точки множества A обозначены черными квадратами, а точки множества B - белым квадратом.

Обозначим через

$$\psi(X) = (\dots((X \odot T_1) \odot T_2) \dots) \odot T_8.$$

Итерационное применение преобразования ψ , т.е. $\psi(\dots(\psi(X)))$ до тех пор, пока результат не изменяется при применении преобразования ψ , приводит к построению 45-выпуклой оболочки множества X . На Рис. 14 показано бинарное изображение, содержащее несколько объектов, и результат построения 45-выпуклой оболочки.

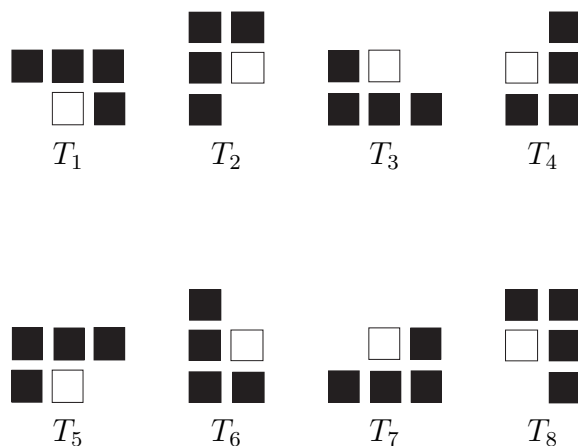


Рис. 13: Структурирующие элементы, используемые в алгоритме построения 45-выпуклой оболочки множества. Начало координат находится в белом квадрате.

5.1 Утоньшение и алгоритмы скелетизации

В отличие от морфологического скелета множества X , представляющего объединение центров максимальных шаров, вписанных в X , в литературе рассматриваются также скелеты множеств, полученные в результате последовательности преобразований утоньшения. В качестве примера рассмотрим алгоритм,

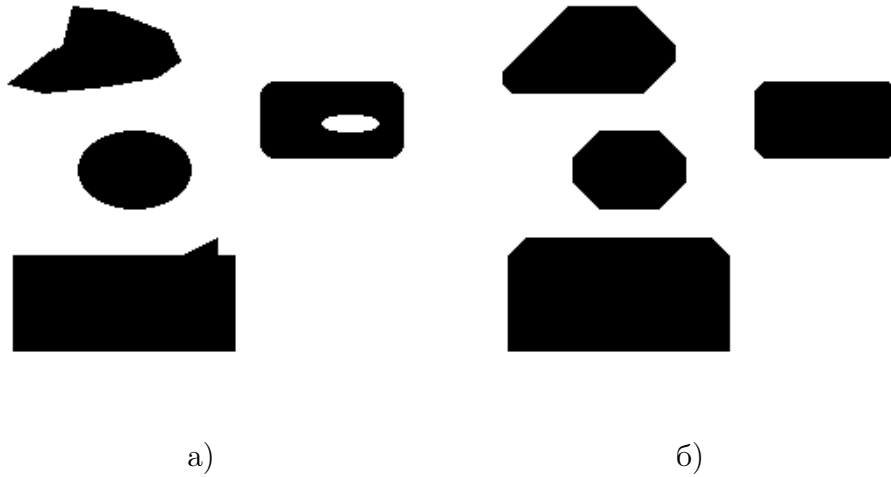


Рис. 14: Бинарное изображение а) и результат построения 45-выпуклой оболочки б) на основе преобразований утолщения со структурирующими элементами, показанными на Рис. 13.

один шаг которого имеет следующий вид:

$$\psi(X) = (\dots ((X \circ T_1) \circ T_2) \dots) \circ T_8.$$

Структурирующие элементы T_1, T_2, \dots, T_8 показаны на Рис. 15. Соответствующие каждому структурирующему элементу точки множества A обозначены черными квадратами, а точки множества B – белыми квадратами. Итерационное применение преобразования $\psi(X)$ приводит к получению скелета бинарного изображения. На Рис. 16 показано бинарное изображение, содержащее несколько объектов, и результат выполнения алгоритма скелетизации на основе преобразований утоньшения.

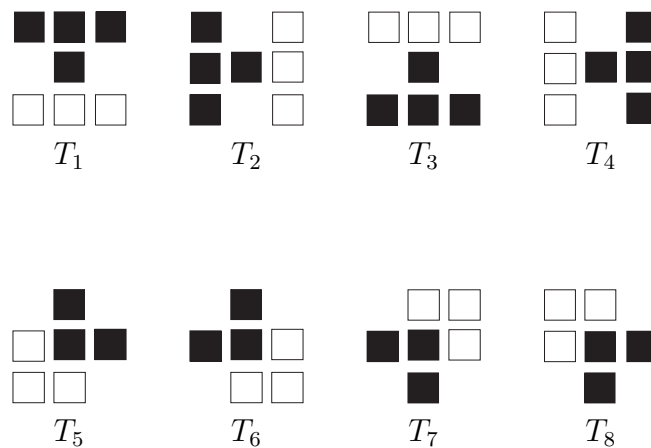


Рис. 15: Структурирующие элементы, используемые в алгоритме скелетизации на основе преобразований утоньшения. Начало координат находится в центральном элементе.

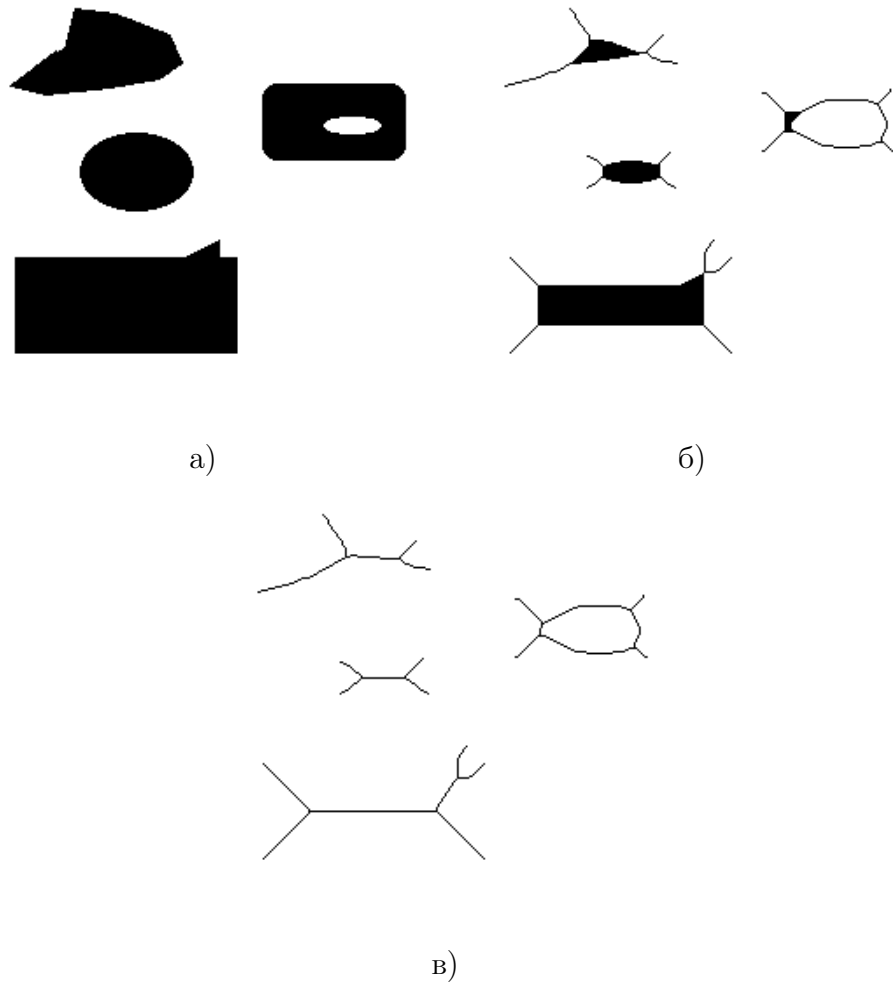


Рис. 16: Бинарное изображение а), результат выполнения 15 итераций алгоритма утоньшения б) и скелеты объектов в), полученные на основе преобразований утоньшения со структурирующими элементами, показанными на Рис. 15.

Приведем также другой алгоритм скелетизации бинарных изображений.

Алгоритм состоит из двух циклов. Первый цикл представляет собой итерационный процесс, причем на каждой итерации происходит 4 прохода изображения. На каждом проходе удаляются точки четырех направлений. Для этих четырех проходов соответственно применяются структурирующие элементы D_1, D_2, D_3, D_4 .

Второй цикл также представляет итерационный процесс, на каждой итерации которого выполняются четыре прохода изображения. На каждом проходе выполняется операция утоньшения. Применяются четыре структурирующих элемента E_1, E_2, E_3, E_4 . Используемые структурирующие элементы изображены на Рис 17.

Один шаг первого цикла алгоритма можно представить в виде

$$\psi_1\{X, D\} = \{X \circ D\} = (((X \circ D_1) \circ D_2) \circ D_3) \circ D_4, \quad (22)$$

0 0 x 0 1 1 x 1 x	x 0 0 1 1 0 x 1 x	x 1 x 1 1 0 x 0 0	x 1 x 0 1 1 0 0 x
D_1	D_2	D_3	D_4
x 0 x 1 1 1 x 1 x	x 1 x 1 1 0 x 1 x	x 1 x 1 1 1 x 0 x	x 1 x 0 1 1 x 1 x
E_1	E_2	E_3	E_4

Рис. 17: Структурирующие элементы, использованные для построения скелета бинарного изображения. Жирным шрифтом выделен центр элемента, а элементы, значения которых не играют роли (т.е. могут быть или 0, или 1), обозначены через x .

где $X \bigcirc D_i$ - утоньшение множества X .

Пусть после выполнения m шагов первого цикла наступает сходимость к множеству X' .

Один шаг второй цикла задается в виде

$$\psi_2\{X', E\} = \{X' \bigcirc E\} = (((X' \bigcirc E_1) \bigcirc E_2) \bigcirc E_3) \bigcirc E_4. \quad (23)$$

Второй цикл выполняется столько шагов, пока не наступает сходимость к множеству, которое и является скелетом исходного изображения X .

Рис. 18 иллюстрирует работу указанного алгоритма. Результат работы первого цикла показан на Рис. 18б), а окончательный скелет, полученный после завершения второго цикла показан на Рис. 18в).

Меняя порядок применения структурирующих элементов, можно получить различные алгоритмы скелетизации.

Например, в алгоритме скелетизации, описанном ниже, используются те же самые структурирующие элементы, но другой порядок выполнения операций утоньшения

$$\psi(X, T, E) = \{(X \ominus T_i) \bigcap (X \ominus T_{i+1}) \bigcap (X \ominus E_i)\},$$

т.е. на одном шаге выполняются три операции утоньшения.

Изображения, получаемые при применении указанных алгоритмов скелетизации, сохраняют связность исходного изображения. Порядок выполнения операций влияет на форму получаемого результата, на его симметрию.

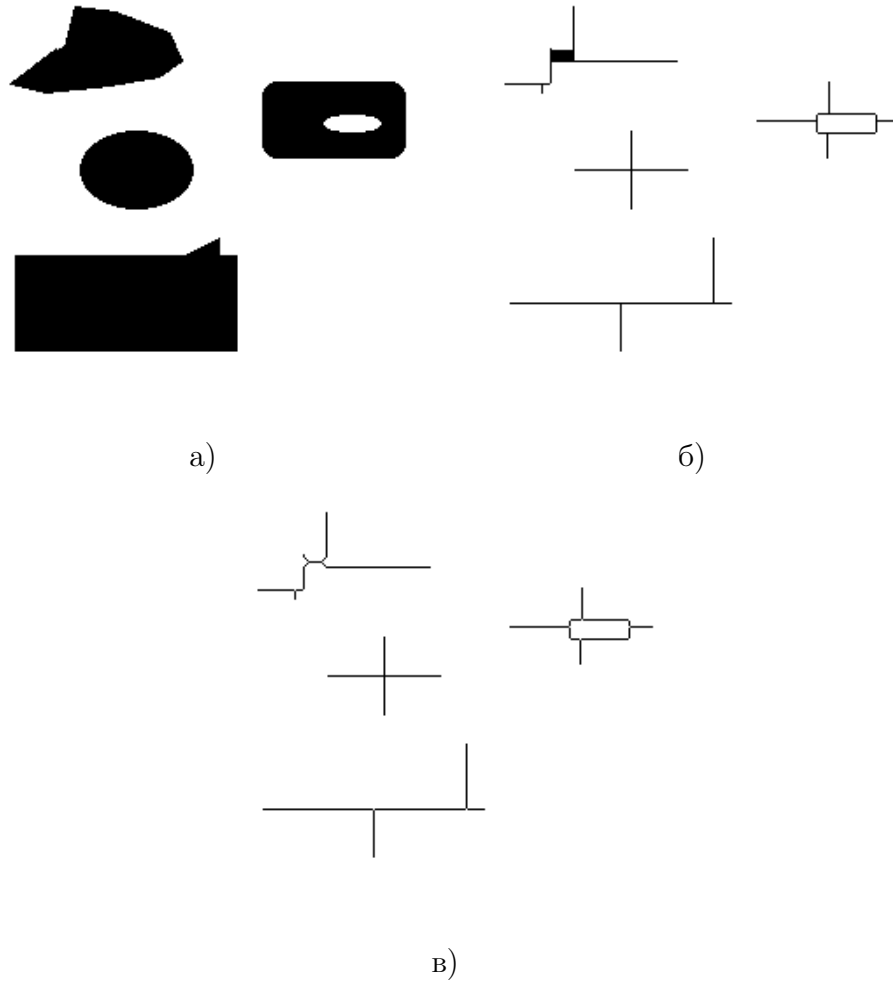


Рис. 18: Бинарное изображение а), результат выполнения итераций первого цикла алгоритма (22) б) и окончательный результат в), полученный после второго цикла алгоритма (23).

6 Морфологический спектр

Одной из важнейших проблем в машинном зрении и обработке изображений является проблема представления формы-размера изображения.

Непрерывные бинарные изображения. Определим спектр компактного бинарного изображения $X \subseteq \mathbb{R}^m$ относительно выпуклого бинарного изображения $B \subseteq \mathbb{R}^m$ следующим образом:

$$PS_X(r, B) = \frac{-dA(X \circ rB)}{dr}, \quad r \geq 0, \quad (24)$$

$$PS_X(-r, B) = \frac{dA(X \bullet rB)}{dr}, \quad r > 0. \quad (25)$$

Здесь $A(X)$ – площадь множества X , $X \circ rB$ – отмыкание множества X структурирующим элементом rB (т.е. элементом радиуса r , если радиус элемента

B является единичным), $X \bullet rB$ – замыкание множества X структурирующим элементом rB .

Выпуклость B гарантирует неотрицательность спектра для всех значений $r \in \mathbb{R}$, т.к. $X \circ rB \supseteq X \circ sB$, если $r < s$.

Поскольку X – компактное множество, то существует максимальное $p > 0$ такое, что $X \circ rB = \emptyset$ для любых $r > p$.

Обозначим $CH(X)$ выпуклую оболочку множества X , т.е. наименьшее выпуклое множество, содержащее X . Известно, что $X \bullet rB \subseteq X \bullet sB \subseteq CH(X)$ для любых $s > r \geq 0$.

Спектр содержит много полезной информации об изображении. Шероховатость границы изображения характеризует спектр для малых значений r . Наличие в изображении X больших выступов, сравнимых с элементами sB , влечет появление в спектре импульсов вокруг положительных значений $r = s$. Значения спектра для отрицательных r характеризуют наличие в X заливов и дыр.

Бинарные дискретные изображения.

Пусть $X \subseteq \mathbb{Z}^m$ – конечное бинарное изображение и $B \subseteq \mathbb{Z}^m$ – конечное множество.

Спектр множества X определяется следующим образом:

$$PS_X(n, B) = A\left((X \circ nB) \setminus (X \circ (n+1)B)\right), \quad n \geq 0,$$

$$PS_X(-n, B) = A\left((X \bullet nB) \setminus (X \bullet (n-1)B)\right), \quad n \geq 1.$$

Здесь $A \setminus B = \{x \in A, x \notin B\}$ означает разность множеств. Пусть N и K определяются следующим образом:

$$N = \max\{n \mid X \ominus nB \neq \emptyset\},$$

$$\lim_{n \rightarrow \infty} (X \bullet nB) = X \bullet KB.$$

Очевидно, что $PS(n) = 0$ для всех $n > N$ и $n < -K$.

Примеры морфологического спектра одного и того же изображения для структурирующих элементов, изображенных на Рис. 19, показаны на Рис. 20.

$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & \mathbf{1} & 1 \\ 1 & 1 & 1 \end{array}$	$\begin{array}{ccc} 0 & 1 & 0 \\ 1 & \mathbf{1} & 1 \\ 0 & 1 & 0 \end{array}$	$\begin{array}{ccccc} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \mathbf{1} & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{array}$
B_1	B_2	B_3

Рис. 19: Структурирующие элементы, использованные для построения морфологического спектра бинарного изображения, показанного на Рис. 20а). Жирным шрифтом выделен центр элемента.

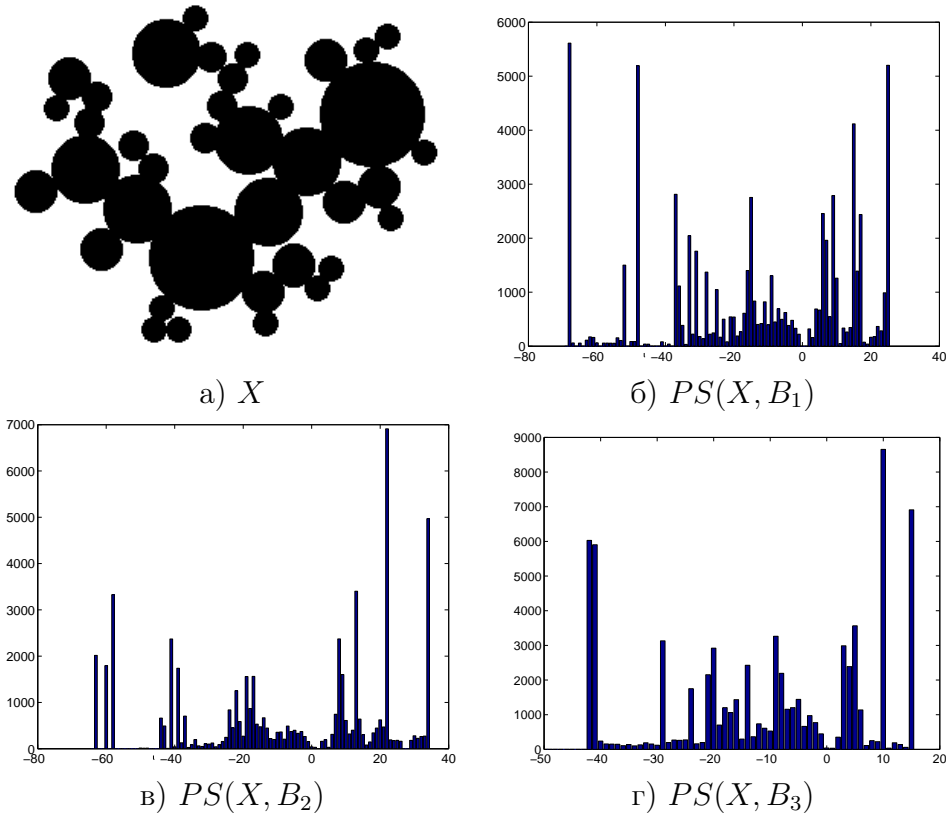


Рис. 20: Бинарное изображение и морфологические спектры, построенные для структурирующих элементов, показанных на Рис. 19.

Полутоновые дискретные изображения.

Пусть $f(x)$ и $g(x)$, $x \in \mathbb{Z}^m$ – неотрицательные полутоновые функции с дискретными конечными областями задания. Спектр функции f с помощью функции g определяется следующим образом

$$PS_f(+n, g) = A\left((f \circ ng) - f \circ (n+1)g\right), \quad 0 \leq n \leq N,$$

$$PS_f(-n, g) = A\left((f \bullet ng) - (f \bullet (n-1)g)\right), \quad 1 \leq n \leq K,$$

где N есть максимальное n , такое, что $f \ominus ng \neq -\infty$, K есть максимальное значение n такое, что $U(f \bullet Kg) = \lim_{n \rightarrow \infty} U(f \bullet ng)$ (здесь $U(f)$, как и ранее, подграфик функции f). Известно, что предел существует. Запись $(f_1 - f_2)$ означает функцию, принимающую в каждой точке x значение $f_1(x) - f_2(x)$, и $A(h) = \sum_{x \in H} h(x)$, где H – область задания функции h .

Очевидно, $PS_f(ng) = 0$ для всех $n > N$ и $n < -K$, а для остальных значений n выполняется $PS_f(ng) \geq 0$.

7 Гранулометрия и преобразование отмыкания

Гранулометрией (распределением размера) будем называть семейство операторов

$$\alpha_r(X) = X \circ B(r).$$

Здесь $B(r)$ – структурирующий элемент радиуса r , $0 \in B$. Вычислить гранулометрию дискретного или непрерывного изображения – значит найти отмыкание структурирующими элементами всех радиусов.

Гранулометрия связана с дистанционным преобразованием следующим образом:

$$\alpha_r(X) = \bigcup_{D_X(x) \geq r} B(D_X(x))_x.$$

7.1 Бинарные изображения

Преобразованием отмыкания $O_{X,B}$ множества $X \subset \mathbb{Z}^n$ структурирующим элементом B , $0 \in B$ будем называть полутоновое изображение:

$$O_{X,B}(x) = \max\{k \mid x \in X \circ B(k)\}.$$

Верно соотношение:

$$X \circ B(k) = \{x \in X \mid O_{X,B}(x) \geq k\}.$$

Таким образом, преобразование отмыкания содержит всю необходимую информацию для построения гранулометрии.

Идея алгоритма вычисления преобразования отмыкания заключается в выполнении следующих этапов:

1. Нахождение дистанционного преобразования множества X ;
2. Выделение точек, среди которых могут быть центры максимальных шаров;
3. Построение преобразования отмыкания по формуле:

$$O_{X,B}(x) = \max\{r \mid x \in B(y, r), r = \rho_{X,B}(y) - 1\}.$$

Пусть ρ_X – вычисленное дистанционное преобразование X . Выделим в множестве X точки, которые могут быть центрами максимальных шаров.

Пусть $\check{B} = \{x \mid -x \in B\}$. Если $y \in \{x\} \oplus \check{B}$ и $\rho_X(y) > \rho_X(x)$, то очевидно, $B(x, \rho_X(x)) \subset B(y, \rho_X(y))$. Таким образом, все центры максимальных шаров являются локальными максимумами дистанционного преобразования.

Алгоритм 3 Центры максимальных шаров относительно фиксированного структурирующего элемента.

1. Просматриваем все точки $x_i \in X$ в произвольном порядке. Для каждой x_i :

- Если для некоторой $y_j \in (\{x_i\} \oplus \check{B}) \cap X$ выполняется $\rho_X(y_j) > \rho_X(x_i)$, то отметить x_i как точку, не являющуюся центром максимального шара.

2. Зафиксируем множество M всех неотмеченных точек. Оно содержит все центры максимальных шаров.

Пусть M – найденное множество локальных максимумов дистанционного преобразования. Теперь можно восстановить преобразование отмыкания.

Алгоритм 4 Преобразование отмыкания относительно фиксированного структурирующего элемента.

1. Вычислить дистанционное преобразование X ;
2. Найти множество M локальных максимумов дистанционного преобразования;
3. Упорядочить точки M в порядке возрастания значений дистанционного преобразования;
4. Присвоить $O_i := 0$ для всех $x_i \in X$;
5. Для каждой $y_j \in M$ (в порядке возрастания $\rho_X(y_j)$) выполнить:
 - Для каждой $x_i \in B(y_j, \rho_X(y_j) - 1)$ присвоить $O_i := \rho_X(y_j) - 1$.

Итоговые значения O_i есть результат преобразования отмыкания.

Сложность этого алгоритма затруднительно оценить простой формулой. Самым объемным является последний его этап. Его сложность пропорциональна суммарному числу точек всех максимальных шаров (каждая точка считается столько раз, во сколько шаров она входит), имеет порядок больше, чем N и меньше, чем N^2 (N – число пикселей изображения) и, очевидно, тем больше, чем в большей степени максимальные шары перекрывают друг друга.

Алгоритм также обобщается на более сложные типы метрики, например, чемферную метрику.

На Рис. 21 показаны результаты преобразования отмыкания для бинарного изображения и структурирующих элементов, задающих четырехсвязное и восьмисвязное расстояние. Это же бинарное изображение использовалось на Рис. 12 для иллюстрации дистанционного преобразования. Различия в определениях преобразования отмыкания и дистанционной функции видны и на приведенных рисунках.

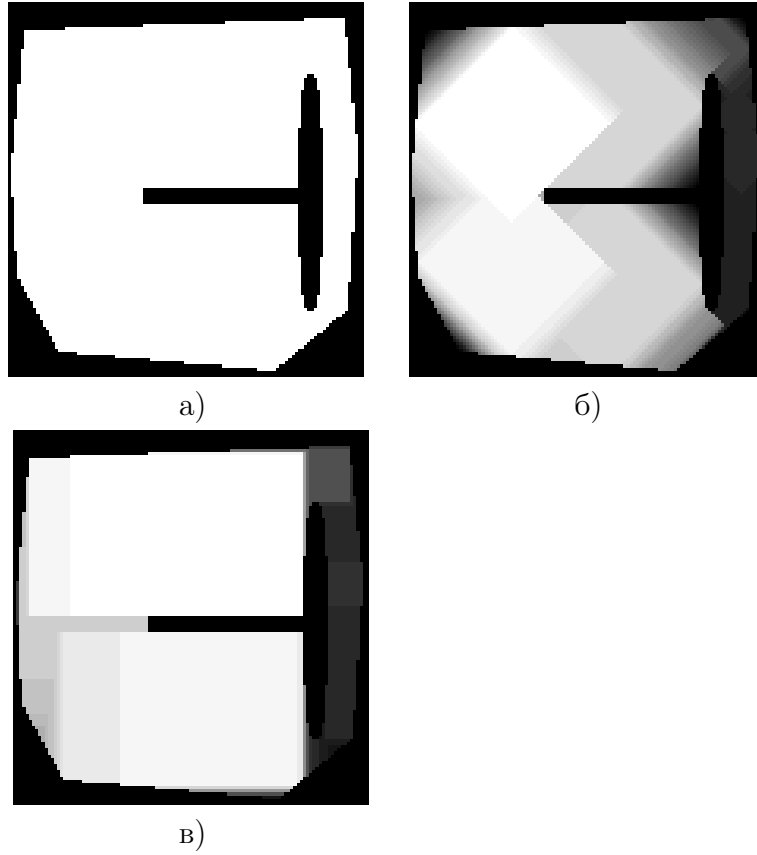


Рис. 21: Исходное бинарное изображение а) и преобразование отмыкания для четырехсвязного б) и восьмисвязного расстояния в).

7.2 Полутоновые изображения

Перейдем к случаю полутоновых изображений. Задача, которая нас интересует, состоит в следующем: найти отмыкание $f \circ g^{\oplus k}$ для всех $k \in \mathbb{N}$. Здесь $g^{\oplus k} = g \oplus \dots \oplus g$ (k слагаемых). Если $k = 0$, то полагаем, что функция $g^{\oplus 0}$ определена в единственной точке 0, и $g^{\oplus 0}(0) = 0$.

Для того чтобы задача допускала решение посредством конечного числа операций, будем предполагать, что $f, g : \mathbb{Z}^n \rightarrow \mathbb{Z}^+ \cup \{-\infty\}$ – функции с ограниченным носителем (областью определения). Кроме того, предполагаем $g(0) \geq 0$.

Переформулируем задачу в терминах бинарной морфологии, n -мерное полутоновое изображение будем представлять как $n + 1$ -мерное бинарное. Пусть $U = \{(x, t) \mid t \leq f(x)\}$, $B = \{(y, \tau) \mid \tau \leq g(y)\} \subset \mathbb{Z}^{n+1}$, $x, y \in \mathbb{Z}^n$, $t, \tau \in \mathbb{Z}$. Тогда $U \circ B^{\oplus k}$ – бинарное представление $f \circ g^{\oplus k}$.

Теперь вместо задачи с полутоновым изображением можно рассматривать задачу с бинарным изображением (большой размерности). Попытка непосредственно применить алгоритмы для бинарных изображений, однако, наталкивается на следующую трудность: подграфы функций являются неограниченными множествами, а алгоритмы имеют силу только для ограниченных. На

самом деле, однако, можно получить верный результат, используя ограниченные снизу подграфики (нижняя граница зависит от конкретных изображения и структурирующего элемента).

8 Фильтры

8.1 Порядково-статистические и медианные фильтры

Порядково-статистические фильтры и их частный случай – медианные фильтры – представляют собой класс нелинейных, инвариантных относительно сдвига фильтров. Эти фильтры находят широкое применение в цифровой обработке речевых сигналов и изображений.

Мы рассматриваем только дискретные порядково-статистические фильтры.

Пусть S – множество из n действительных чисел, в котором допускаются повторения чисел. Пусть числа упорядочены по невозрастанию их значений. Тогда k -е по номеру число в этом списке называется k -й порядковой статистикой множества S , $k = 1, 2, \dots, n$. Если n – нечетное число, то при $k = \frac{n+1}{2}$ имеем медиану множества S .

Определим понятие порядковых статистик множеств $X \subseteq \mathbb{Z}^m$ при помощи окна $W \subseteq \mathbb{Z}^m$, состоящего из n точек, т.е. $|W| = n$.

Назовем k -й порядковой статистикой множества X (или k -м порядково-статистическим фильтром) при помощи окна W множество $OS^k(X, W)$ (соответственно преобразование $OS^k(X, W)$), определяемое следующим образом

$$OS^k(X, W) = \{y \in \mathbb{Z}^m \mid |X \cap W_y| \geq k\},$$

где $k = 1, 2, \dots, n$, а W_y означает сдвиг окна (множества) W в точку y , т.е. $W_y = \{y + a, a \in W\}$. Если n нечетное число и $k = \frac{n+1}{2}$, то имеем медиану множества X или медианный фильтр с окном W .

Пусть функция $f(x)$ определена на некотором подмножестве из \mathbb{Z}^m , и $W \subseteq \mathbb{Z}^m$ состоит из n точек.

Назовем k -м порядково-статистическим фильтром функции $f(x)$ при помощи окна W преобразование функции $f(x)$ в функцию $os^k(f, W)$, определяемую следующим образом

$$os^k(f, W)(x) \text{ есть } k\text{-я порядковая статистика множества } \{f(y) \mid y \in W_x\},$$

где $x \in \mathbb{Z}^m$, $1 \leq k \leq n$, и W_x – сдвиг множества W в точку x . Будем обозначать медиану функции $f(x)$ при помощи окна W через $med(f, W)$.

Если функция $f(x)$ является бинарной (т.е. характеристической функцией множества), то нетрудно видеть, что определения порядковой статистики функции и соответствующего ей множества при помощи окна W являются эквивалентными.

Приведем некоторые свойства порядково-статистических фильтров.

Свойство 2 *Порядково-статистические фильтры для функций и множеств являются возрастающими, т.е.*

$$X \subseteq Y \Rightarrow OS^k(X, W) \subseteq OS^k(Y, W),$$

$$f \leq g \Rightarrow os^k(f, W) \leq os^k(g, W), \quad 1 \leq k \leq n.$$

Здесь $f \leq g$ означает, что $f(x) \leq g(x)$ во всех точках x .

Обозначим, как и ранее, через $T_t(f)$ сечение функции f на уровне t .

Свойство 3 *Порядково-статистические фильтры для функций перестановочны с сечением. Это означает, что для любой функции $f(x)$, любого конечного окна W , любого $t \in \mathbb{R}$ и $k = 1, 2, \dots, |W|$, справедливо*

$$T_t[os^k(f, W)] = OS^k[T_t(f), W].$$

Для медианного фильтра имеем

$$T_t[med(f, W)] = med[X_t(f), W].$$

Порядково-статистический фильтр восстанавливается по сечениям следующим образом

$$[os^k(f, W)](z) = \sup\{t \in \mathbb{R} \mid z \in OS^k[T_t(f), W]\}.$$

Для порядково-статистических фильтров справедливо свойство двойственности. Обозначим через X^c дополнение множества X в \mathbb{Z}^m .

Свойство 4 *Для любого множества X и любого конечного окна W , $|W| = n$, справедливо*

$$OS^k(X^c, W) = [OS^{n-k+1}(X, W)]^c, \quad 1 \leq k \leq n,$$

$$med(X^c, W) = [med(X, W)]^c.$$

Из определения порядково-статистических фильтров следует справедливость соотношений:

$$OS^{k+1}(X, W) \subseteq OS^k(X, W), \quad 1 \leq k \leq n-1,$$

$$OS^1(X, W) = X \oplus \check{W},$$

$$OS^n(X, W) = X \ominus W.$$

Здесь $\check{W} = \{-w \mid w \in W\}$.

Таким образом, расширение и эрозия являются специальными случаями порядково-статистических фильтров. Однако любые порядковые статистики можно представлять в виде объединения эрозий или пересечения расширений (а для функций – через максимумы эрозий либо минимумы расширений).

Справедлива следующая теорема.

Теорема 4 Для любого множества $X \subseteq \mathbb{Z}^m$ (любой функции $f(x), x \in \mathbb{Z}^m$) k -я порядковая статистика при помощи окна W , $k = 1, 2, \dots, |W|$, равна объединению (соответственно, максимуму) эрозий множества X (функции f) по всем структурирующим элементам, состоящим ровно из k точек окна W . Она также равна пересечению (соответственно, минимуму) расширений множества X (функции f) по всем структурирующим элементам, состоящим ровно из $n - k + 1$ точек окна \check{W} , т.е.

$$OS^k(X, W) = \bigcup_{P \subseteq W, |P|=k} X \ominus P = \bigcap_{Q \subseteq W, |Q|=n-k+1} X \oplus \check{Q},$$

$$os^k(f, W)(x) = \max_{P \in W, |P|=k} (f \ominus P)(x) = \min_{Q \in W, |Q|=n-k+1} (f \oplus \check{Q})(x).$$

Нетрудно видеть, что число подмножеств, состоящих из k точек окна равно числу сочетаний из n по k , т.е. $C_n^k = \frac{n!}{k!(n-k)!}$. Эрозия $(f \ominus P)$ и расширение $(f \oplus \check{Q})$ есть не что иное, как локальный минимум и локальный максимум функции f внутри окон P и \check{Q} , соответственно.

Известны соотношения, связывающие медианные фильтры с отмыканием и замыканием изображения одномерными и двумерными окнами. Приведем некоторые результаты для двумерных окон. Пусть изображения заданы в плоскости \mathbb{Z}^2 на гексагональной или прямоугольной решетках.

Обозначим через H и R следующие структурирующие элементы для указанных решеток, соответственно:

$$\begin{array}{ccccc} & 1 & & 1 & \\ 1 & & +1 & & 1 \\ & 1 & & 1 & \end{array} \quad \begin{array}{ccccc} & 1 & & 1 & \\ 1 & & +1 & & 1 \\ & 1 & & 1 & \end{array}$$

Пусть $med^k(X, H) = med(med^{k-1}(X, H))$ означает k -ю итерацию применения медианного фильтра к изображению X . Тогда справедлива следующая теорема.

Теорема 5 Для любого множества X или функции $f(x)$ на гексагональной решетке справедливо:

1.

$$X \circ H \subseteq med^k(X, H) \subseteq X \bullet H,$$

$$f \circ H \leq med^k(f, H) \leq f \bullet H \text{ для любого целого } k > 0.$$

2.

$$\text{из } X = X \circ H = X \bullet H \text{ следует } X = med(X, H),$$

$$\text{из } f = f \circ H = f \bullet H \text{ следует } f = med(f, H).$$

Аналогичная теорема справедлива и для прямоугольной решетки. В этом случае H нужно везде заменить на R . Доказательство теоремы 5 основано на следующих соотношениях

$$\text{из } z \in X \circ H \text{ следует } |H_z \cap X| \geq 4,$$

$$\text{из } z \in X \circ R \text{ следует } |R_z \cap X| \geq 4.$$

Здесь H_z означает сдвиг структурирующего элемента в точку z .

8.2 Стековые фильтры

Стековые фильтры определены следующим образом.

Рассмотрим входную функцию $f(z)$, $z \in \mathbb{Z}^m$ с конечным множеством значений $0, 1, \dots, M$. Для каждого $t \in \{0, 1, \dots, M-1\}$ находим сечения T_t функции f и преобразуем их с помощью возрастающего инвариантного относительно сдвига дискретного бинарного фильтра $\varphi(T_t)$. Предполагается, что результат преобразования с помощью фильтра $\varphi(X)$ в точке $z \in \mathbb{Z}^m$ определяется по значениям входного изображения внутри фиксированного конечного окна W , состоящего из n точек и помещенного в точку z .

Поскольку $\varphi(X)$ инвариантный относительно сдвига бинарный фильтр, то ему соответствует булева функция $\beta(x_1, x_2, \dots, x_n)$, где переменные x_i , $i = 1, 2, \dots, n$, представляют n значений входного изображения в точках $y \in W_z = \{z + w \mid w \in W\}$. Поскольку $\varphi(X)$ возрастающий фильтр, то $\beta(x_1, x_2, \dots, x_n)$ – положительная булева функция. Напомним, что булева функция называется положительной, если ее можно представить в виде конъюнктивной нормальной формы (к.н.ф.) или дизъюнктивной нормальной формы (д.н.ф.), не содержащей отрицания переменных.

Стековый фильтр $st(f)$ для входной функции $f(z)$ определяется следующим образом:

$$[st(f)](z) = \sum_{t=0}^{M-1} [\varphi(T_t(f))](z), \quad z \in \mathbb{Z}^m.$$

Это определение можно обобщить на непрерывный случай, когда функция $f(z)$ принимает значения из промежутка $I \subset \mathbb{R}$, т.е.

$$[st(f)](z) = \int_I [\varphi(X_t(f))](z) dt.$$

Стековый фильтр является по определению перестановочным с сечением и поэтому справедлива также следующая форма представления стекового фильтра

$$[st(f)](z) = \sup\{t \in V, z \in \varphi(T_t(f))\},$$

где V – область значений функции $f(z)$.

Опишем алгоритм представления любого стекового фильтра в терминах операций объединения и пересечения, либо максимума и минимума.

1. Пусть $W \subseteq \mathbb{Z}^m$ – окно, состоящее из n точек, а $\beta(x_1, x_2, \dots, x_n)$ – возрастающая булева функция. Пусть задано отображение $I : w \in W \longrightarrow \{1, 2, \dots, n\}$, задающее взаимно-однозначное соответствие между точками окна и аргументами функции β .
2. Построим минимальную д.н.ф. или к.н.ф функции β . Известно, что минимальная д.н.ф. или к.н.ф. возрастающей функции является единственной и положительной (т.е. не содержит отрицательных переменных).

3. Построим бинарный фильтр, оперирующий с входным множеством X , заменяя дизъюнкцию и конъюнкцию переменных, соответственно, объединением и пересечением множеств и каждую переменную x_i на X_{-w} , где $w \in W$ и $I(w) = i$. Предполагается, что точка $0 \in W$ и при движении окна помещается в точку, в которой вычисляется значение фильтра.
4. На основе полученной формулы построим стековый фильтр, заменяя множество X на сечения $T_t(f)$ входной функции $f(z)$.

Для получения функционального представления стекового фильтра через \max и \min функции $f(z)$ достаточно заменить в минимальной д.н.ф. или к.н.ф. конъюнкцию и дизъюнкцию, соответственно, на \max и \min , а каждую переменную x_i на $f(z + w)$, где $I(w) = i$.

Таким образом, каждый стековый фильтр представляется конечным числом максимумов и минимумов. Справедливо и обратное утверждение, т.е. любое конечное число максимумов и минимумов является инвариантным относительно сдвига, перестановочным с сечением фильтром. И этот фильтр является стековым.

Рассмотренные ранее морфологические фильтры с бинарным структурирующим элементом (расширение, эрозия, отмыкание и замыкание) и порядкостатистические фильтры являются специальными случаями стековых фильтров.

В то же время морфологические фильтры с полутоновым структурирующим элементом не перестановочны с сечением и, следовательно, не являются стековыми фильтрами.

8.3 Представление через ядро

Бинарные фильтры.

Пусть $P(E)$ – множество всех подмножеств E и $S \subseteq P(E)$ – класс входных множеств (бинарных изображений), замкнутый относительно сдвига, т.е. $X + z \in S$ для любых $X \in S$ и $z \in E$.

Пусть $\Psi : S \rightarrow P(E)$ инвариантный относительно сдвига (ИС) бинарный фильтр, т.е.

$$\Psi(X + z) = \Psi(X) + z \text{ для любых } X \in S, z \in E.$$

Ядро $\mathcal{K}(\Psi)$ ИС фильтра Ψ определяется следующим образом

$$\mathcal{K}(\Psi) = \{X \in S \mid 0 \in \Psi(X)\},$$

где 0 – нулевой вектор в E .

Фундаментальным свойством ядра является то, что по нему можно однозначно восстановить ИС фильтр, т.к.

$$\Psi(X) = \{z \in E \mid X - z \in \mathcal{K}(\Psi)\}.$$

Бинарный фильтр называется возрастающим, если

$$A \subseteq B \Rightarrow \Psi(A) \subseteq \Psi(B) \text{ для любых } A, B \in S.$$

Пусть Ψ – ИС возрастающий бинарный фильтр, заданный на S . Пусть $S^d = \{A^c \mid A \in S\}$, где A^c – дополнение A в E . Определим на S^d двойственный фильтр $\Psi^d(A) = [\Psi(A^c)]^c$, $A \in S^d$.

Очевидно, Ψ^d является ИС возрастающим бинарным фильтром только тогда, когда ψ является ИС возрастающим бинарным фильтром.

Справедлива теорема Матерона.

Теорема 6 Пусть $\Psi : S \rightarrow P(E)$ – инвариантный относительно сдвига возрастающий бинарный фильтр и множество $S \subseteq P(E)$ замкнуто относительно сдвига. Тогда для любого $X \in S$ справедливо

$$\Psi(X) = \bigcup_{A \in \mathcal{K}(\Psi)} X \ominus A = \bigcap_{B \in \mathcal{K}(\Psi^d)} X \oplus \check{B}.$$

Здесь $\check{B} = \{-b \mid b \in B\}$.

Полутоновые фильтры.

Пусть полутоновой фильтр ψ определен на подмножестве \mathcal{F} расширенных вещественно-значных функций $ERV(D)$ $f : D \rightarrow V \cup \{-\infty, +\infty\}$, где $D \subseteq \mathbb{R}^m$ (или \mathbb{Z}^m) и $V \subseteq \mathbb{R}$ (или \mathbb{Z}).

Будем предполагать, что множество \mathcal{F} замкнуто относительно переноса функций, т.е. если $f(x) \in \mathcal{F}$, то $f(x - y) + c \in \mathcal{F}$ для любых $(y, c) \in D \times V$.

Фильтр ψ называется инвариантным относительно переноса (ИП), если $\psi[f(x - y) + c] = \psi(f)(x - y) + c$.

Определим ядро $\mathcal{K}(\psi)$ инвариантного относительно переноса полутонового фильтра

$$\mathcal{K}(\psi) = \{f \in \mathcal{F} \mid \psi(f)(0) \geq 0\}.$$

Определим ядро полутонового фильтра ψ , перестановочного с сечением. Пусть Ψ – соответствующий фильтру ψ бинарный фильтр. Ядро полутонового фильтра, перестановочного с сечением, может быть найдено следующим образом

$$\mathcal{K}(\psi) = \{f \in \mathcal{F} \mid T_0(f) \in \mathcal{K}(\Psi)\}.$$

Утверждения, аналогичные теореме Матерона о представлении для бинарных фильтров, имеют место и для полутоновых фильтров. Для их формулировки нам необходимы понятия возрастающего фильтра и двойственного фильтра. Полутоновой фильтр ψ называется возрастающим, если

$$f \leq g \Rightarrow \psi(f) \leq \psi(g) \text{ для любых } f, g \in \mathcal{F}.$$

Обозначим через $\mathcal{F}^d = \{f, -f \in \mathcal{F}\}$, где $(-f)(x) = -f(x)$.

Двойственным фильтром ψ^d будем называть фильтр, определенный на \mathcal{F}^d следующим образом

$$\psi^d(f) = -\psi(-f), \text{ где } f \in \mathcal{F}^d.$$

Теорема 7

Полутоновые фильтры.

Пусть $\psi : \mathcal{F} \rightarrow ERV(D)$ – инвариантный относительно переноса возрастающий полутоновой фильтр, где множество $\mathcal{F} \subseteq ERV(D)$ замкнуто относительно переноса. Тогда ψ можно представить как поточечный супремум эрозий с помощью функций ядра, а также как поточечный инфимум расширений с помощью функций ядра двойственного фильтра ψ^d , т.е. для любой $f \in \mathcal{F}$

$$\psi(f) = \sup_{g \in \mathcal{K}(\psi)} f \ominus g = \inf_{h \in \mathcal{K}(\psi^d)} f \oplus \check{h}.$$

Полутоновые фильтры, перестановочные с сечением.

Пусть $\psi : \mathcal{F} \rightarrow ERV(D)$ – инвариантный относительно переноса полутоновой фильтр, перестановочный с сечением и Ψ – соответствующий ψ бинарный фильтр.

Тогда ψ можно представить как поточечный супремум эрозий с помощью ядра ψ , а также как поточечный инфимум расширений с помощью множества ядра двойственной системы ψ^d , т.е.

$$\psi(f) = \sup_{A \in \mathcal{K}(\Psi)} f \ominus A = \inf_{B \in \mathcal{K}(\Psi^d)} f \oplus \check{B}.$$

Теоремы 6 и 7 дают представления фильтров через морфологические преобразования. Тем не менее для практических применений они не являются пригодными, так как мощность ядра этих фильтров обычно является бесконечной. Естественный интерес представляет вопрос о том, когда можно получить конечные разложения фильтров.

Приведем в качестве примера следующий, достаточно очевидный, результат для бинарных фильтров.

Преобразование $\Psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ будем называть локально ограниченным, если существует ограниченная маска M такая, что

$$h \in \Psi(X) \Leftrightarrow h \in \Psi(X \bigcap M_h).$$

Таким образом, при вычислении локально ограниченного преобразования в некоторой точке h достаточно знать значения в некоторой окрестности M точки h .

Теорема 8 Пусть $E = \mathbb{Z}^m$ и Ψ – возрастающее инвариантное относительно сдвига, локально ограниченное преобразование. Тогда существует конечное подмножество \mathcal{K}_0 ядра $\mathcal{K}(\psi)$ такое, что

$$\psi(X) = \bigcup_{A \in \mathcal{K}_0} (X \ominus A).$$

Здесь $\mathcal{K}_0 = \{A \bigcap M \mid A \in \mathcal{K}(\psi)\}$. Так как $\mathcal{K}_0 \subseteq \mathcal{P}(M)$, то оно конечно.

Нетрудно видеть, что множество \mathcal{K}_0 можно уменьшить, оставив только минимальные по включению элементы.

Другой, более общий подход для сокращения элементов разложения фильтра состоит в переходе от ядра к базису, состоящему из минимальных элементов ядра.

Определим базис бинарного фильтра ψ и полутонного фильтра ψ следующим образом

$$\mathcal{B}(\Psi) = \{A \in \mathcal{K}(\Psi) \mid (B \in \mathcal{K}(\Psi) \text{ и } B \subseteq A) \Rightarrow A = B\},$$

$$\mathcal{B}(\psi) = \{f \in \mathcal{K}(\psi) \mid (g \in \mathcal{K}(\psi) \text{ и } g \leq f) \Rightarrow f = g\}.$$

Таким образом, базис состоит только из минимальных элементов ядра.

9 Примеры алгоритмов

9.1 Быстрый алгоритм вычисления суммы Минковского

Вычисление суммы Минковского дискретных множеств по определению требует значительных временных затрат. Предположим, например, что A и B – дискретные двумерные множества размера $N \times N$ на дискретной плоскости \mathbb{Z}^2 . Тогда реализация сложения Минковского по определению имеет временную сложность $O(N^4)$.

Пусть A и B – произвольные компактные множества в \mathbb{R}^2 .

Пусть $\chi[A]$ – характеристическая функция множества A , т.е.

$$\chi[A](x_1, x_2) = \begin{cases} 1, & \text{если } x = (x_1, x_2) \in A, \\ 0 & \text{в противном случае.} \end{cases}$$

Рассмотрим свертку

$$(\chi[A] \otimes \chi[B])(p_1, p_2) = \int_{\mathbb{R}} \int_{\mathbb{R}} \chi[A](x_1, x_2) \chi[B](p_1 - x_1, p_2 - x_2) dx_1 dx_2$$

и корреляцию

$$(\chi[A] \times \chi[B])(p_1, p_2) = \int_{\mathbb{R}} \int_{\mathbb{R}} \chi[A](x_1, x_2) \chi[B](x_1 - p_1, x_2 - p_2) dx_1 dx_2.$$

В случае дискретного пространства \mathbb{Z}^2 свертка определяется как

$$(\chi[A] \otimes \chi[B])(p_1, p_2) = \sum_{\mathbb{Z}} \sum_{\mathbb{Z}} \chi[A](x_1, x_2) \chi[B](p_1 - x_1, p_2 - x_2).$$

Предположим, что мощность множества B равна n ($|B| = n$), тогда k -й порядково-статистический фильтр $OS^k(A, B)$ преобразует бинарное изображение A следующим образом:

$$OS^k(A, B) = \{x \in \mathbb{Z}^2, \mid |A \cap B_x| \geq k\} \quad (26)$$

для любого $k = 1, 2, \dots, n$. При этом k называется номером фильтра. Очевидно, что для $k = 1$ имеем сложение Минковского $A \oplus \check{B}$ (расширение структурирующим элементом \check{B}) и для $k = n$ имеем вычитание Минковского $A \ominus B$ (эрозию структурирующим элементом B).

Обозначим через $V(B)$ площадь множества B .

Справедливо следующее утверждение.

Утверждение 4

$$\begin{aligned} p = (p_1, p_2) \in A \oplus B, & \quad \text{если и только если} \quad (\chi[A] \otimes \chi[B])(p_1, p_2) > 0, \\ p = (p_1, p_2) \in A \ominus B, & \quad \text{если и только если} \quad (\chi[A] \times \chi[B])(p_1, p_2) = V(B), \\ p = (p_1, p_2) \in OS^k(A, B), & \quad \text{если и только если} \quad (\chi[A] \times \chi[B])(p_1, p_2) \geq k. \end{aligned}$$

Доказательство утверждения непосредственно следует из определений расширения, эрозии и порядково-статистического фильтра.

На Рис. 22 приведен пример вычисления морфологических операций и порядково-статистического фильтра через соответствующие сечения функции корреляции. В данном примере мы рассматриваем симметричный относительно начала координат структурирующий элемент, поэтому свертка и корреляция совпадают.

Обозначим через $\mathcal{F}(f)$ преобразование Фурье действительнзначной функции $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. Также обозначим через \mathcal{F}^{-1} обратное преобразование Фурье. Принимая во внимание свойства преобразования Фурье, из утверждения 4 непосредственно получаем следующее предложение.

Утверждение 5

$$\begin{aligned} p = (p_1, p_2) \in A \oplus B, & \quad \text{если и только если} \quad (\mathcal{F}^{-1}(\mathcal{F}(\chi[A])\mathcal{F}(\chi[B])))(p_1, p_2) > 0, \\ p = (p_1, p_2) \in A \ominus B, & \quad \text{если и только если} \quad (\mathcal{F}^{-1}(\mathcal{F}(\chi[A])\mathcal{F}(\chi[\check{B}]))) (p_1, p_2) = V(B), \\ p = (p_1, p_2) \in OS^k(A, B), & \quad \text{если и только если} \quad (\mathcal{F}^{-1}(\mathcal{F}(\chi[A])\mathcal{F}(\chi[B])))(p_1, p_2) \geq k. \end{aligned}$$

Рассмотрим дискретную плоскость \mathbb{Z}^2 вместо евклидовой плоскости \mathbb{R}^2 . В этом случае можно применить быстрое преобразование Фурье (БПФ) для эффективного вычисления сложения Минковского. Для изображения размера $N \times N$ БПФ имеет временную сложность $O(N^2 \log N)$. Для получения корректных результатов вычисление сложения и вычитания Минковского посредством циклической свертки следует выполнять для окна $2N \times 2N$, и, следовательно, их временная сложность равна $O(N^2 \log N)$.

9.2 Сложение Минковского выпуклых многоугольников

В непрерывном случае для вычисления суммы Минковского выпуклых компактных множеств удобно использовать контурное представление таких множеств. Наиболее просто вычисляется сумма Минковского выпуклых многоугольников (в двумерном случае).

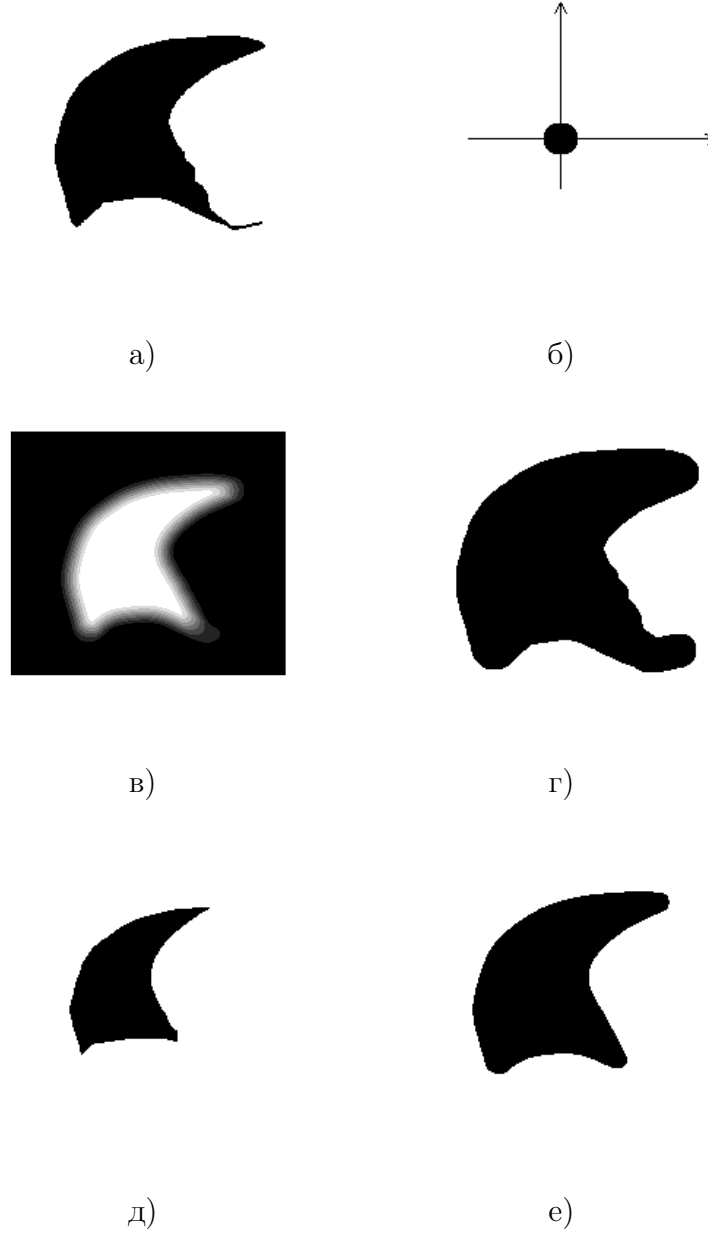


Рис. 22: Вычисление операций через сечения функции корреляции: а) исходное изображение A ; б) симметричный структурирующий элемент B , содержащий начало координат; в) корреляция $\chi[A] \times \chi[B]$; г) расширение $A \oplus B$; д) эрозия $A \ominus B$; е) медиана $OS^k(A, B)$ для $k = |B|/2$.

Многоугольник $P \subset \mathbb{R}^2$ может быть представлен единственным образом заданием одной из своих вершин, а также длин и направлений всех своих сторон. Обозначим через p_i длину i -й стороны, а через u_i – вектор, ортогональный этой стороне. Пусть $\angle u_i$ обозначает угол между положительным направлением оси Ox и u_i . Если известно положение точки P , то для задания многоугольника достаточно указать последовательность $(u_1, p_1), (u_2, p_2), \dots, (u_n, p_n)$, где $n = n_P$ – число вершин P . Будем называть эту последовательность *периметрическим представлением* P . Если многоугольник выпуклый, то достаточно задать множество $\{(u_i, p_i)\}$, поскольку в этом случае вектора нормалей к сторонам многоугольника упорядочены в порядке возрастания углов $\angle u_i$ (направление обхода против часовой стрелки). Иллюстрация периметрического представления дается на Рис. 23.

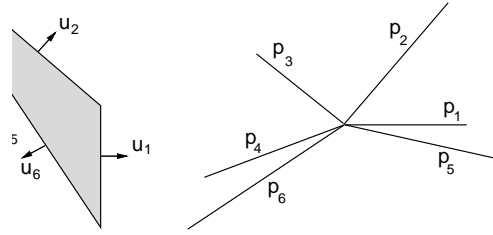


Рис. 23: Периметрическое представление невыпуклого многоугольника

В качестве другого представления может быть использована так называемая *периметрическая мера* M_P :

$$M_P(u) = \begin{cases} p_i, & \text{если } u = u_i, \\ 0 & \text{в других случаях.} \end{cases}$$

Для каждого выпуклого многоугольника P

$$\sum M_P(u)u = 0, \quad (27)$$

где сумма берется по всем u , для которых $M_P(u) \neq 0$. Более того, произвольная дискретная функция, заданная на единичной окружности, для которой выполняется условие (27), однозначно задает выпуклый многоугольник. Фактически это соотношение означает, что контур P замкнут.

Известно, что периметрическое представление суммы Минковского двух выпуклых многоугольников может быть вычислено через периметрические представления слагаемых, например

$$M_{P \oplus Q}(u) = M_P(u) + M_Q(u) \quad \text{для любого } u \in S^1. \quad (28)$$

Здесь S^1 обозначает единичную окружность. Это позволяет быстро вычислять сумму Минковского выпуклых многоугольников (см. Рис. 24).

Известно, что каждый выпуклый многоугольник может быть разложен в сумму Минковского отрезков и треугольников. Это разложение может быть легко построено посредством периметрического представления.

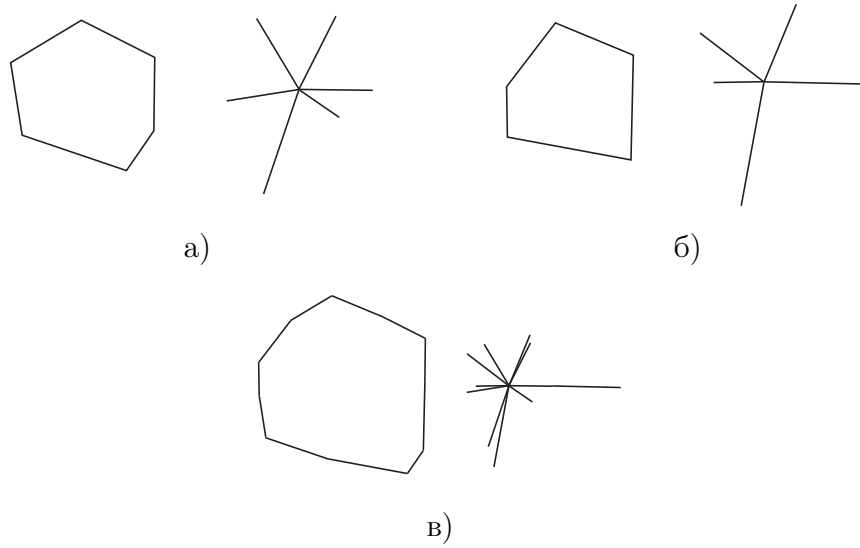


Рис. 24: Сумма Минковского двух выпуклых многоугольников и ее периметрическое представление: а),б) исходные многоугольники и их периметрические представления; в) сумма Минковского.

Выделение отрезка в разложении соответствует нахождению пары векторов, \vec{a} и \vec{b} , имеющих противоположные направления в периметрическом представлении многоугольника. Тогда вектор $\vec{a} + \vec{b}$ остается вместо векторов \vec{a} и \vec{b} . После выделения всех отрезков получаем периметрическое представление, не содержащее векторов с противоположными направлениями.

Выделение треугольника соответствует нахождению в периметрическом представлении тройки неколлинеарных векторов, для которых существует линейная комбинация с положительными коэффициентами, равная нулю. Это выделение всегда возможно для периметрического представления, содержащего более трех векторов. Кроме того, выделение следует осуществить так, чтобы после него роза содержала, по крайней мере, на один вектор меньше. Пусть \vec{a} , \vec{b} – два вектора, образующие друг с другом максимальный угол в периметрическом представлении. Этот угол меньше π (так как отрезки уже выделены). Найдем такой вектор \vec{c} , что вектора \vec{a} , \vec{b} , \vec{c} не содержатся ни в какой полуплоскости. Вектор \vec{c} всегда существует, так как сумма всех векторов в периметрическом представлении равна нулю. Тогда существуют числа p и q такие, что $p\vec{a} + q\vec{b} + \vec{c} = \vec{0}$. Пусть $r = \max\{p, q, 1\}$. Имеет место $\frac{p}{r}\vec{a} + \frac{q}{r}\vec{b} + \frac{1}{r}\vec{c} = \vec{0}$. Хотя бы один коэффициент среди p/r , q/r , $1/r$ равен 1. Поэтому после выделения векторов $\frac{p}{r}\vec{a}$, $\frac{q}{r}\vec{b}$, $\frac{1}{r}\vec{c}$ периметрическое представление будет содержать, по крайней мере, на один вектор меньше.

Таким образом, мы получили простое доказательство и алгоритм декомпозиции выпуклого многоугольника в сумму Минковского отрезков и треугольников.

Декомпозиция многоугольника в сумму Минковского отрезков и треуголь-

ников изображена на Рис. 25.

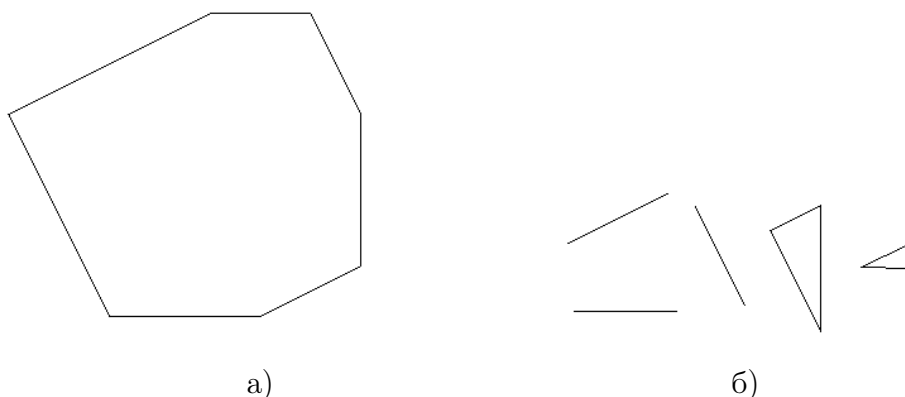


Рис. 25: Многоугольник а) и его декомпозиция в сумму Минковского отрезков и треугольников б)

9.3 Быстрое вычисление полутоновых морфологических операций с произвольным бинарным структурирующим элементом

Тривиальный алгоритм вычисления эрозии заключается в поиске для каждой точки p изображения f минимума среди всех значений $f(p+b)$, где $b \in W$. При вычислении расширения ищется максимум среди всех значений $f(p+b)$, $b \in \check{W}$, где $\check{W} = \{-b \mid b \in W\}$.

На практике, это означает, что в каждую точку p изображения f помещается структурирующий элемент B_p ($B = W$ для эрозии и $B = \check{W}$ для расширения) и среди точек, попавших в это окно находится минимальное значение (в случае эрозии) и максимальное (в случае расширения), которое и становится результатом выполнения морфологической операции в этой точке. На Рис. 26а),б) показана схема вычисления для двух последовательных точек p (темная точка) и q (точка с ромбом внутри). Закрашенная область соответствует структурирующему элементу B , перенесенному в точки p и q соответственно (окна B_p и B_q).

При сканировании изображения одни и те же точки посещаются несколько раз. Например, на Рис. 26 точка с окружностью внутри просматривается и при вычислении значения для точки p и для соседней точки q , так как она попадает в окна B_p и B_q . Если структурирующий элемент имеет большой размер, то таких точек будет больше.

На Рис. 27а) показан пример структурирующего элемента большого размера. При сканировании изображения слева направо (Рис. 27б)) окна B_p и B_q пересекаются: точки множества $B_p \cap B_q$ просматриваются при вычислении значения в p и q ; точки множеств $B_p \setminus (B_p \cap B_q)$ и $B_q \setminus (B_p \cap B_q)$ просматриваются при вычислении значений только для p и q соответственно. При вычислении

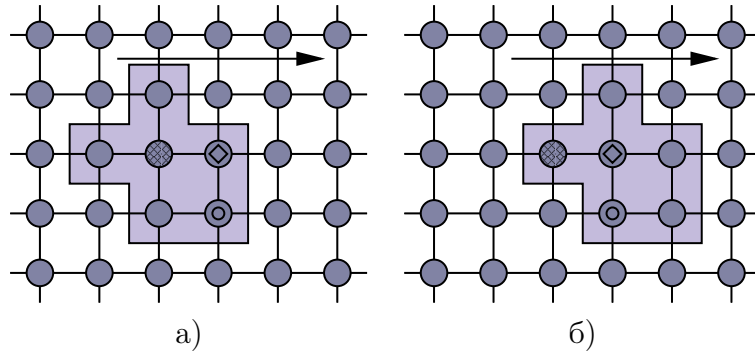


Рис. 26: Положение скользящего окна (структурирующего элемента) для двух последовательных точек изображения, p – темная точка, q – точка с ромбом внутри, закрашенная область – структурирующий элемент, перенесенный в текущую точку просмотра: а) окно в точке p ; б) окно в точке q .

эрозии минимальное значение внутри окна B_q можно получить из двух минимумов по множествам $B_p \cap B_q$ и $B_q \setminus (B_p \cap B_q)$. При этом, минимум по множеству $B_p \cap B_q$ уже найден при вычислении в точке p . Аналогично для расширения, ищутся максимальные значения. Чтобы эффективно вычислять экстремумы *быстрый* алгоритм использует гистограмму.

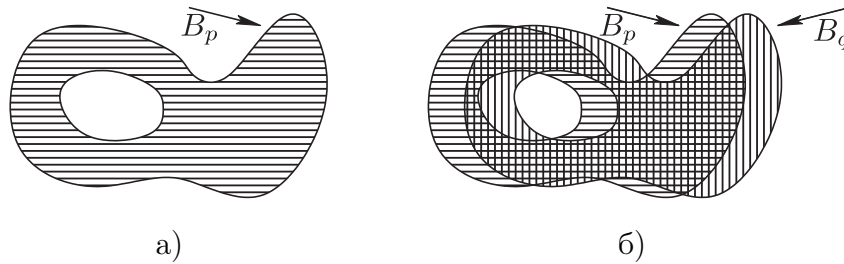


Рис. 27: Сдвиг скользящего окна: а) структурирующий элемент большого размера, перенесенный в точку p ; б) структурирующий элемент, перенесенный в две соседние точки p и q .

Гистограмма строится по точкам, попавшим в текущее окно B_p . При вычислении эрозии минимальное значение внутри окна находится из гистограммы. Гистограмма обновляется при каждом перемещении окна: из нее удаляются точки множества $B_p \setminus (B_p \cap B_q)$, не попадающие в окно B_q (область, закрашенная горизонтальными линиями на Рис. 27б)) и добавляются точки множества $B_q \setminus (B_p \cap B_q)$, попадающие в окно B_q , но не принадлежащие окну B_p (область, закрашенная вертикальными линиями на Рис. 27б)). Преимуществом использования гистограммы является то, что при перемещении окна значения для точек множества $B_p \cap B_q$ уже содержатся в гистограмме.

На Рис. 28 показана схема работы алгоритма. Гистограмма на Рис. 28б)

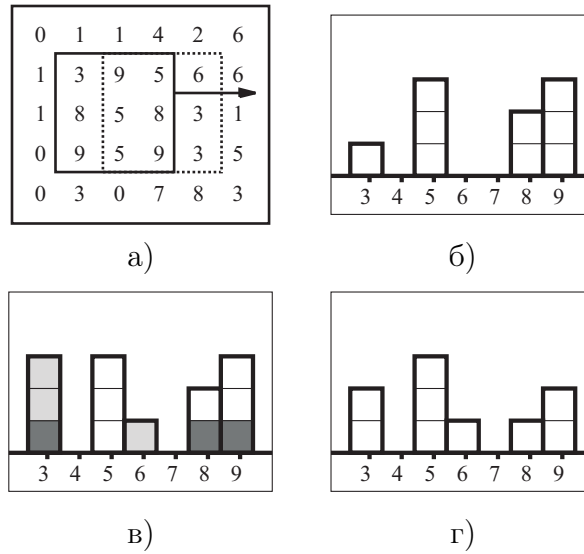


Рис. 28: Обновление гистограммы: а) фрагмент полутонового изображения со структурирующим элементом 3×3 ; б) гистограмма, построенная по окну первого рисунка; в) при обновлении гистограммы черные блоки удаляются, а серые блоки добавляются; г) результат обновления гистограммы при сдвиге окна на один элемент влево.

построена по точкам окна 3×3 (Рис. 28а)). Из гистограммы находится минимальное значение внутри окна $B_p - 3$, и максимальное – 9. При переходе окна от p к следующей точке q обновляется и гистограмма (Рис. 28в)): значения точек множества $B_p \setminus (B_p \cap B_q)$ удаляются из гистограммы (черные блоки), а значения точек множества $B_q \setminus (B_p \cap B_q)$ добавляются (серые блоки). Значение морфологической операции для точки q определяется по гистограмме на Рис. 28г), и это будет 3 в случае эрозии и 9 в случае расширения.

Сканирование всего изображения осуществляется в трех направлениях: слева направо (просматривается строка полностью), сверху вниз (переход на следующую строку), справа налево (просматривается строка полностью). Далее осуществляется переход на строку ниже и сканирование продолжается. При таком обходе гистограмма полностью строится лишь однажды – для самой первой точки (верхней левой), потом гистограмма обновляется при каждом перемещении окна.

Гистограмма обновляется значениями точек множеств $B_p \setminus (B_p \cap B_q)$ (горизонтальные линии на Рис. 27б)) и $B_q \setminus (B_p \cap B_q)$ (вертикальные линии на Рис. 27б)). Координаты обновляемых точек зависят лишь от формы элемента, а не от изображения, и поэтому их можно определить до начала сканирования изображения. Для каждого направления просмотра будут свои множества, однако множество $B_p \setminus (B_p \cap B_q)$ при сканировании справа налево совпадает с множеством $B_q \setminus (B_p \cap B_q)$ при сканировании слева направо. Это справедливо и для множеств $B_q \setminus (B_p \cap B_q)$ и $B_p \setminus (B_p \cap B_q)$. Таким образом, достаточно опре-

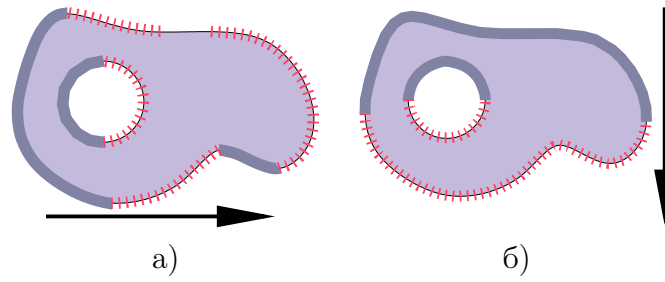


Рис. 29: Сплошной толстой линией отмечены точки структурирующего элемента, значения которых удаляются из гистограммы при перемещении окна, а прерывистой линией отмечены точки, значения которых добавляются в гистограмму; а) сканирование в направлении слева направо; б) сканирование в направлении сверху вниз.

делить координаты обновляемых точек для двух направлений сканирования изображения: слева направо и сверху вниз.

На Рис. 29а) условно сплошной толстой линией отмечены точки множества $B_p \setminus (B_p \cap B_q)$, а прерывистой линией – $B_q \setminus (B_p \cap B_q)$ при направлении сканирования слева направо. Аналогично, на Рис. 29б) показаны множества при сканировании сверху вниз. Эти множества определяются путем смещения структурирующего элемента в соответствующем направлении, а хранятся как список смещений относительно центра структурирующего элемента.

Псевдокод быстрого алгоритма эрозии

Описание переменных

- *imIn*: входное изображение f ;
- *imOut*: результат;
- B : бинарное изображение, содержащее структурирующий элемент;
- H : гистограмма;

Инициализация

Построение множеств $B_p \setminus (B_p \cap B_q)$ и $B_q \setminus (B_p \cap B_q)$ из B ;
 Построение гистограммы H для первой точки изображения *imIn*;

Основной код

```

For each нечетной строки  $l$  изображения imIn

  For each точки  $c$  строки  $l$ 
     $imOut[c,l] \leftarrow \min(H)$ ;
    Переместить окно вправо и обновить гистограмму;
  End For

   $imOut[c,l] \leftarrow \min(H)$ ;

If строка  $l + 1$  существует then

```



```

        Переместить окно на строку вниз и обновить гистограмму;
Else
    Exit;
End If

For each точки  $c$  строки  $l + 1$ 
     $imOut[c, l+1] \leftarrow \min(H)$ ;
    Переместить окно влево и обновить гистограмму;
End For

End For

```

Сложность алгоритма Наихудший случай – это когда множеству $B_p \cap B_q$ пусто. Тогда при сканировании изображения гистограмма будет строиться заново для каждой точки, так как содержимое окон будет каждый раз новое. В этом случае, сложность алгоритма эквивалентна сложности тривиального алгоритма. Отрезок (линейный сегмент) параллельный оси Y является примером "нехорошего" структурирующего элемента для выбранного направления сканирования.

Наилучший сценарий – количество обновляемых точек (точки множеств $B_p \setminus (B_p \cap B_q)$ и $B_q \setminus (B_p \cap B_q)$) во время перемещения окна мало по сравнению с общим количеством точек структурирующего элемента. В это случае, сложность алгоритма практически не зависит от размера структурирующего элемента. Отрезок, (линейный сегмент) параллельный оси X является примером "хорошего" структурирующего элемента.

Типичный структурирующий элемент имеет периметр намного меньше своей площади. В этом случае, данный алгоритм будет эффективнее тривиального.

Порядково-статистические и медианные фильтры Следует отметить, что описанный выше алгоритм применим не только для вычисления морфологических операций, его также можно применять для вычисления порядково-статистических фильтров (стр. 33) с окном произвольного вида и размера. При этом гистограмма позволяет получать любую статистику быстро и эффективно. Например, значение медианного фильтра для Рис. 28б) будет 8, а для Рис. 28г) – 5.

9.4 Параллельная реализация морфологических операций для линейных структурирующих элементов

Свойства базовых морфологических операций позволяют реализовать их параллельным образом и достичь, благодаря этому, высокой скорости обработки, используя параллельные компьютерные архитектуры.

В качестве примера рассмотрим задачу выделения линейных объектов на бинарных и полутоновых изображениях. Для этого будем применять фильтры,

которые удаляют объекты из плоскости изображения, имеющие размер вдоль некоторого заданного направления меньший, чем некоторое пороговое значение. Если ориентация таких объектов не известна заранее, то либо будем предполагать, что ее можно определить и выполнить последующую фильтрацию, либо будем применять вращательно инвариантные фильтры. Для фильтрации будем использовать преобразования, основанные на операциях математической морфологии, и порядково-статистические фильтры с линейными структурирующими элементами. Линейный структурирующий элемент аппроксимируется на дискретной решетке с помощью алгоритма Брезенхема. В этом случае фильтры оказываются эффективными при использовании специальной параллельной компьютерной архитектуры, описанной ниже. Дискретная аппроксимация структурирующего элемента по алгоритму Брезенхема происходит одновременно с реализацией используемых фильтров.

Пусть A и $f(x)$ будут, соответственно, бинарное и полутоновое изображение, и $B \in \mathbb{Z}^2$ - бинарный структурирующий элемент.

Порядково-статистические фильтры непосредственно связаны с морфологическими операциями. Предположим, что $|B| = n$. Тогда k -ый порядково-статистический фильтр $R_k[A, B]$ преобразует бинарное изображение A следующим образом:

$$R_k[A, B](x) = \{x \in \mathbb{Z}^2, |A \cap B_x| \geq k\}$$

для каждого $k = 1, 2, \dots, n$. Очевидно, для $k = 1$ имеем расширение симметричным структурирующим элементом $-B$ и для $k = n$ - имеем эрозию элементом B .

Порядково-статистический фильтр для полутонового изображения не реализуется эффективно в рамках предлагаемого подхода, поэтому для полутоновых изображений будем в дальнейшем применять фильтр средних значений $MV[f, B]$, определяемый следующим образом

$$MV[f, B](x) = (\sum_{z \in B} f(x + z)) / n.$$

Как было отмечено ранее, будем использовать только линейные структурирующие элементы. Пусть A - бинарное изображение и L - линейный структурирующий элемент длины l , образующий угол α с осью OX и имеющий начало координат своей первой точкой. Тогда следующие преобразования могут оказаться полезными для выделения объектов на бинарном изображении и удаления шумов:

$A \circ L$ - морфологическое отмыкание;

$(A \circ L) \bullet L$ - замыкание, следующее за отмыканием, с таким же структурирующим элементом. Этот фильтр позволяет не только выделять объекты, но и соединять части объектов;

$A \cap [(A \circ L) \bullet L]$ - алгебраическое отмыкание, т.е. антиэкстенсивное и идемпотентное преобразование;

$R_k[A, L]$ - порядково-статистический фильтр.

Для обработки полутонового изображения f будем применять следующие преобразования:

$f \circ L$ – морфологическое отмыкание, как и в предыдущем случае;

$f \bullet L$ – морфологическое замыкание;

$(f \circ L) \bullet L$ – замыкание, следующее за отмыканием;

$MV[f, L]$ – фильтр средних значений.

Заметим, что порядково-статистические фильтры и фильтр средних значений не так чувствительны к шуму по сравнению с морфологическим отмыканием и замыканием.

Все выше описанные преобразования могут быть эффективно реализованы с помощью параллельных алгоритмов на массиве процессоров матричного типа. Опишем кратко организацию и операции нижнего уровня такого компьютера.

Организация параллельного компьютера

В массиве процессоров матричного типа процессорные элементы точно соответствуют пикселям изображения. Каждый процессорный элемент (i, j) имеет регистр памяти для хранения значения пикселя (бинарного или полутонового), и он связан с 8-ю своими соседями, как это имеет место в случае квадратной 8-связной решетки. Для реализации описываемых алгоритмов необходимы следующие операции, выполняемые в параллельном режиме за один шаг:

1. Сдвиг всего изображения на 1 пиксел вдоль одного из 8 возможных направлений. Все возможные направления кодируются числами от 0 до 7;
2. Нахождение супремума SUP или инфимума INF двух изображений. Это означает выполнение логических операций И/ИЛИ для бинарных изображений, и операций MAX/MIN для полутоновых изображений за один шаг;
3. Суммирование двух полутоновых изображений;
4. Копирование одного изображения в другое;
5. Бинаризация входного полутонового изображения f . Эта операция формирует бинарное изображение, каждый пиксел (i, j) которого равен 1, если $(f(i, j) \geq Lbound) \& (f(i, j) < Hbound)$, и 0 в противном случае. Здесь целые переменные $Lbound$ и $Hbound$ являются, соответственно, верхним и нижним порогом, и $f(i, j)$ – значение яркости пикселя (i, j) ;
6. Деление полутонового изображения f на ненулевое целое число;
7. Умножение полутонового изображения f на целое число.

Реализация алгоритмов

Если направление линейного структурирующего элемента L совпадает с одним из 8 основных направлений квадратной решетки, то можно реализовать эрозию и расширение с помощью L точно в соответствии с их определениями, данными выше. Чтобы построить дискретную аппроксимацию произвольного линейного структурирующего элемента с заданным направлением и размером, применим алгоритм Брезенхема. На Рис. 30 показаны дискретные аппроксимации нескольких линейных отрезков.

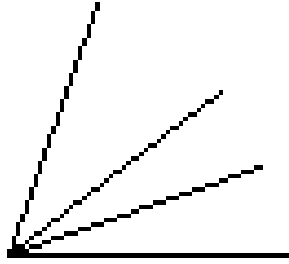


Рис. 30: Дискретные аппроксимации линейных отрезков.

Предположим, что линейный структурирующий элемент L задан следующей последовательностью пикселей $(i_0, j_0), (i_1, j_1), \dots, (i_k, j_k)$, где $i_0 = 0, j_0 = 0$. Тогда операции эрозии $A \ominus L$ и расширения $A \oplus L$ вычисляются следующим образом:

$$A \ominus L = \bigcap A_{(i_1, j_1)} \bigcap \dots \bigcap A_{(i_k, j_k)}, \quad (29)$$

$$A \oplus L = \bigcup A_{-(i_1, j_1)} \bigcup \dots \bigcup A_{-(i_k, j_k)}. \quad (30)$$

Здесь $A_{(i_1, j_1)}$ обозначает сдвиг всего изображения A на один пиксел в направлении, соответствующем пикселу (i_1, j_1) . Соответственно, $A_{(i_l, j_l)}$ есть сдвиг $A_{(i_{l-1}, j_{l-1})}$ в направлении пиксела (i_l, j_l) . Формулы (29) и (30) определяют описываемые ниже алгоритмы. В данных алгоритмах структурирующий элемент задается своей длиной и направлением, а его дискретная аппроксимация вычисляется по алгоритму Брезенхема одновременно с поиском направлений сдвига изображения.

Рассмотрим случай, когда линейный структурирующий элемент полностью расположен в первом октанте. Тогда для его аппроксимации достаточно выполнять пошаговые сдвиги вдоль оси OX и вдоль линии $X = Y$.

Для общего случая необходимо сначала определить октант, содержащий структурирующий элемент, и затем выполнить соответствующее преобразование системы координат.

Алгоритм 5 Эрозия линейным структурирующим элементом

Входные данные:

I_S – исходное изображение;

$dX, dY, SIZE$ – целые переменные, определяющие L :

dX, dY определяет ориентацию ($\text{tg}\alpha = dY/dX$) и

$SIZE$ определяет длину L ($dX > dY > 0$ для случая первого октанта).

Выходные данные:

I_R – изображение, содержащее результат операции.

Используемые внутренние переменные:

I_w – изображение для хранения промежуточных результатов, его тип зависит от типа I_S и I_R ;

$Error$ – константа, используемая в алгоритме Брезенхема.

```
{
/* инициализация */
Error ← 2 * dY - dX
I_w ← I_S
I_R ← I_S
/* аппроксимация L и эрозия */
for 1 to SIZE {
  if (Error < 0) {
    Error ← Error + 2 * dY
    Сдвиг изображения I_w вдоль оси OX
  }
  else {
    Error ← Error + 2 * (dY - dX)
    Сдвиг изображения I_w вдоль линии X = Y
  }
  I_R ← INF (I_w, I_R)
}
} /* конец */
```

Заменяя взятие инфимума на супрэмум получим расширение симметричным линейным структурирующим элементом L' , т. е. $L' = \{-x, x \in L\}$.

Теперь можно реализовать отмыкание и замыкание произвольным линейным элементом, используя их определения.

Морфологическая ковариация это эрозия парой точек, одна из которых - начало координат. Поэтому для ее реализации необходимо вычислить инфимум исходного изображения и изображения, сдвинутого в конечную точку.

Рассмотрим теперь порядково-статистический фильтр для бинарных изображений и фильтр средних значений для полутоновых изображений. Они имеют аналогичные реализации.

Пусть бинарное изображение задано своей характеристической функцией I , т.е. $I(i, j) = 1$, если пиксел (i, j) содержится в изображении, и $I(i, j) = 0$ в противном случае. Порядково-статистический фильтр ранга $Rank$ дает на выходе 1 для каждого пиксела (i, j) , если структурирующий элемент, сдвинутый в этот пиксел $L_{(i,j)}$, имеет не менее, чем $Rank$ общих пикселов с исходным изображением:

$$\begin{aligned} OutputValue(i, j) &= 1, \text{ если } \sum_{(i', j') \in L_{(i,j)}} I(i', j') \geq Rank, \\ OutputValue(i, j) &= 0 \text{ в противном случае.} \end{aligned}$$

Получаем следующий алгоритм реализации порядково-статистического фильтра.

Алгоритм 6 Порядково-статистический фильтр с линейным окном

Входные данные:

I_S – исходное бинарное изображение;

$dX, dY, SIZE$ – целые переменные, определяющие линейное окно L (см. описание для алгоритма 5);

$Rank$ – целое значение ранга.

Выходные данные:

I_R – выходное бинарное изображение.

Используемые внутренние переменные:

I_W – изображение для хранения промежуточных результатов;

$Error$ – константа, используемая в алгоритме Брезенхема.

```
{
/* инициализация внутренних переменных */
Error ← 2 * dY - dX
I_W ← I_S
/* Замечание: копирование бинарного изображения в полутоновое дает полу-
тоновое изображение со значениями пикселей 0 и 1 */
I_R ← I_S
/* аппроксимации L и фильтрации */
for 1 to SIZE {
    if (Error < 0) {
        Error ← Error + 2 * dY
        Сдвиг изображения I_R вдоль оси OX
    }
    else {
        Error ← Error + 2 * (dY - dX)
        Сдвиг изображения I_R вдоль линии X = Y
    }
    I_W ← I_W + I_R
}
I_R ← бинаризация I_W с Lbound = Rank, Hbound = (SIZE + 1)
}
```

Алгоритм реализации фильтра средних значений незначительно отличается от алгоритма 6. Для него достаточно заменить строку

$I_R \leftarrow \text{бинаризация } I_W \text{ с } Lbound = Rank, Hbound = (SIZE + 1)$

на

$I_R \leftarrow I_W / SIZE.$

В этом случае входное и выходное изображения являются полутоновыми.

В заключение рассмотрим более общий случай. Пусть L - дискретная аппроксимация заданного линейного структурирующего элемента, полученная с помощью алгоритма Брезенхема. Предположим, что все пиксели из L имеют номера $1, 2, \dots, SIZE + 1$, причем начало координат имеет номер 1 и последний пиксел имеет номер $SIZE + 1$. Рассмотрим набор k структурирующих элементов $L_i, i = 1, 2, \dots, k$. Каждое L_i содержит подмножество пикселей из L . Преобразование S бинарного или полутонного входного изображения I_R определяется следующим образом:

$$S[I_R, \{L_i\}] = \bigvee_{i=1}^k (I_R \ominus L_i).$$

Здесь супремум \bigvee заменяется объединением \bigcup , в случае бинарного изображения, и \max , в случае полутонного изображения.

Используя описанные выше алгоритмы, получаем параллельную реализацию фильтра $S[I_R, \{L_i\}]$. В этом случае необходимо использовать k временных изображений $I_{w1}, I_{w2}, \dots, I_{wk}$ одновременно, каждое из которых применяется для вычисления эрозии $I_R \ominus L_i, i = 1, 2, \dots, k$. Тогда главный цикл алгоритма имеет следующий вид:

```

for  $j = 1$  to  $SIZE$  {
  if ( $Error < 0$ ) {
     $Error \leftarrow Error + 2 * dY$ 
    Сдвиг изображения  $I_R$  вдоль оси  $OX$ 
  }
  else {
     $Error \leftarrow Error + 2 * (dY - dX)$ 
    Сдвиг изображения  $I_R$  вдоль линии  $X = Y$ 
  }
  for  $i = 1$  to  $k$  {
    if ( $j \in L_i$ ) {  $I_{wi} \leftarrow \text{инфимум } (I_{wi}, I_R)$  }
  }
}
 $I_R \leftarrow \text{супрэмум } (I_{w1}, I_{w2}, \dots, I_{wk})$ 

```

Отметим, что каждый стэковый фильтр с линейным окном может быть представлен в форме аналогичной той, которую имеет фильтр $S[I_R, \{L_i\}]$. Таким образом каждый стэковый фильтр с линейным окном допускает параллельную реализацию. Это возможно, если процессорные элементы позволяют использовать необходимое количество памяти для временных изображений.

Выделение прямолинейных отрезков

Сначала мы должны определить направление, вдоль которого ориентированы объекты изображения. Учитывая начальную информацию о распределении размеров объектов, можно оценить направление объектов, вычисляя ковариацию для некоторого набора направлений. Выбор последних зависит от желаемой точности выделения.

В качестве альтернативного подхода можно применить следующий вращательно-инвариантный фильтр:

$$\bigcup_{\alpha \in \Phi} F_{\alpha},$$

где Φ - множество углов вращения линейного структурирующего элемента, и F_{α} - морфологическое или алгебраическое отмыкание с помощью аппроксимации линейного структурирующего элемента, имеющего направление α . Как и в предыдущем случае, количество направлений зависит от необходимой точности выделения объектов.

Примеры фильтрации изображений линейными структурирующими изображениями показаны на Рис. 31 и 32 соответственно.

9.5 Преобразование водораздела

Пусть функция $f : D \in \mathbb{Z}^2 \rightarrow V \in \mathbb{Z}$ задает дискретное полутонное изображение. Напомним, что сечением функции f уровня i называется множество $T_i(f)$, определяемое как

$$T_i(f) = \{x \in \mathbb{Z}^2 | f(x) \geq i\}.$$

Аналогично определим множество $Z_i(f)$:

$$Z_i(f) = \{x \in \mathbb{Z}^2 | f(x) \leq i\}.$$

Одним из наиболее эффективных алгоритмов сегментации изображений на основе математической морфологии является преобразование водораздела (*watershed*-преобразование).

Рассмотрим изображение f как топографическую поверхность S и определим понятия собирающих бассейнов и линий водораздела на основе имитации процесса наводнения. Предположим, что поверхность S в каждом локальном минимуме $m_i(f)$ имеет отверстия. Поверхность погружается в воду вертикально с постоянной скоростью. Вода, проникающая через отверстия, заполняет поверхность S . Во время затопления два или более потоков воды, поступающих из различных минимумов, могут сливаться. Чтобы предотвратить сливание различных потоков воды в местах сливания на поверхности возводятся дамбы. Процесс погружения поверхности в воду заканчивается, когда вся поверхность будет заполнена водой. Построенные дамбы определяют линии водораздела функции f . Эти линии разделяют различные собирающие бассейны

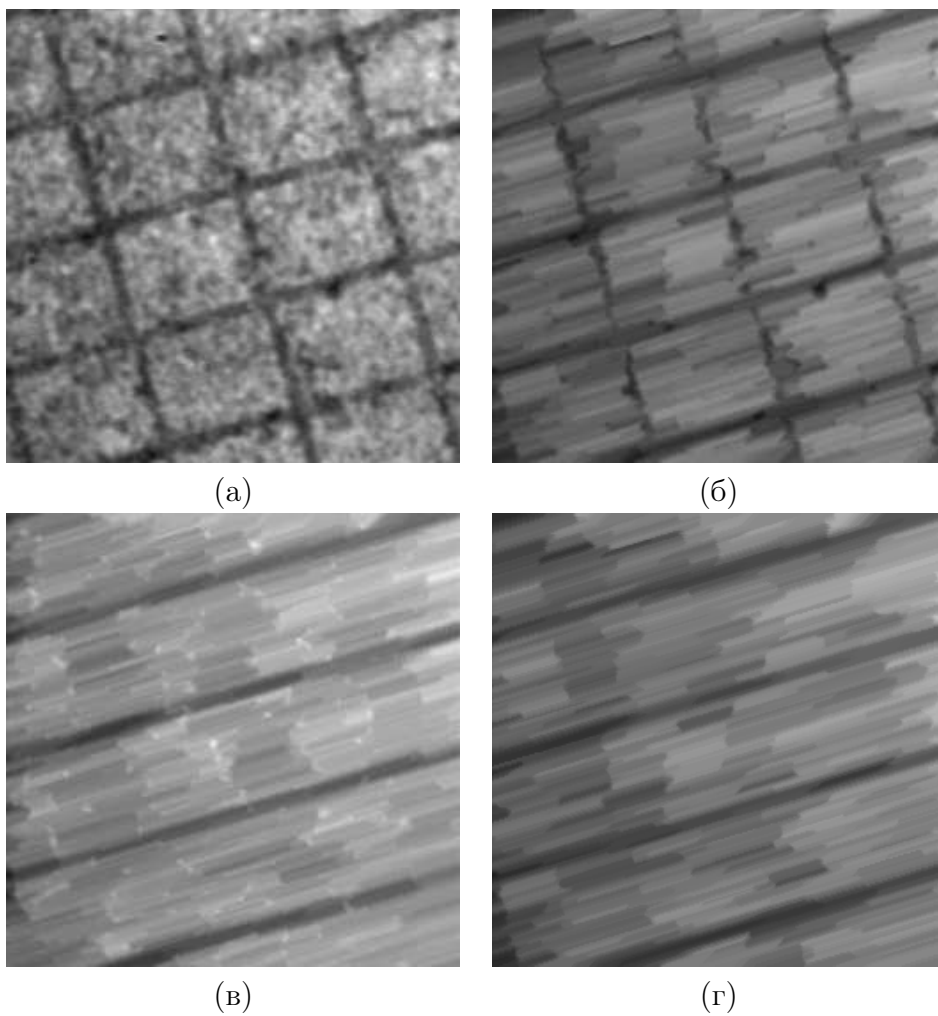


Рис. 31: Выделение линейных объектов на полутоновом изображении: (а) исходное изображение; (б) отмыкание структурирующим элементом длины 25; (в) замыкание структурирующим элементом длины 25; (г) отмыкание с последующим замыканием тем же элементом

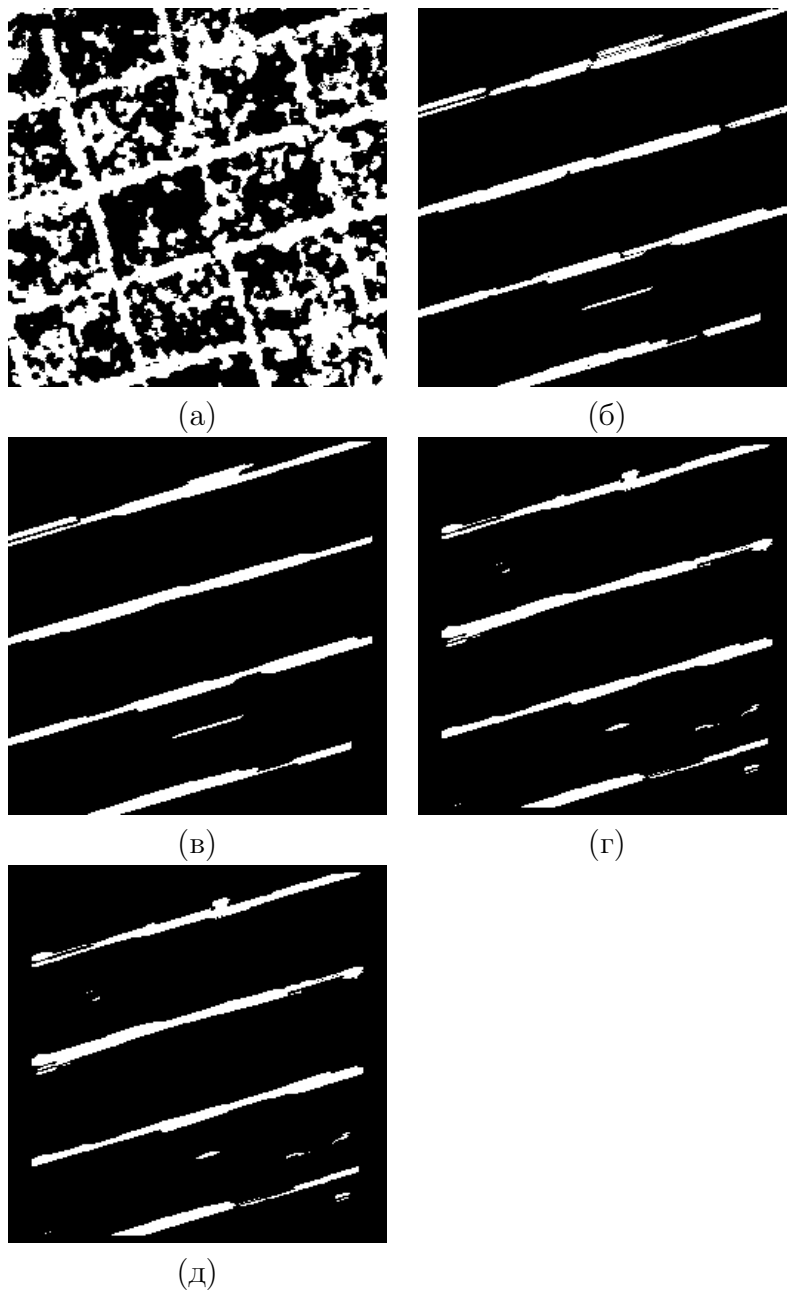


Рис. 32: Выделение линейных объектов на бинарном изображении: (а) исходное изображение; (б) отмыкание структурирующим элементом длины 25; (в) Отмыкание с последующим замыканием тем же элементом; (г) порядково-статистический фильтр с линейным окном длины 40 и рангом 37; (д) алгебраическое отмыкание линейным структурирующим элементом длины 25.

$CB_i(f)$, каждый из которых содержит только один локальный минимум $m_i(f)$ (см. Рис. 33).

Опишем простой алгоритм, реализующий преобразование водораздела на основе морфологических операций.

Рассмотрим множество $Z_i(f)$ функции f для некоторого уровня i (см. Рис. 34) и предположим, что уровень воды достиг значения i и залитые водой области образуют некоторое семейство связных множеств $\{C_k(i)\}$ из $Z_i(f)$. Рассмотрим процесс затопления на уровне $i + 1$. Залитые водой области на этом уровне будут формироваться либо из связных компонент множества предыдущего уровня, либо из локальных минимумов функции f , значение в которых равно $i + 1$. Обозначим множество локальных минимумов со значением $i + 1$ через \mathcal{M}_{i+1} . Область, залитая из связной компоненты $C_k(i)$ есть геодезическая зона влияния $Z_{Z_{i+1}(f)}(C_k(i))$ множества $C_k(i)$ в множестве $Z_{i+1}(f)$ (см. (16)). Область, залитая из локального минимума $M_l \in \mathcal{M}_{i+1}$ есть геодезическая зона влияния $Z_{Z_{i+1}(f)}(M_l)$. Аналогичным образом получается семейство связных множеств $\{C_k(i + 1)\}$ уровня $i + 1$. Из этого семейства формируется семейство связных множеств уровня $i + 2$ и т.д. Процесс начинается с уровня, равного минимальному значению функции f , а заканчивается на уровне n , равном максимальному значению функции f . Связные компоненты последнего уровня определяют собирающие бассейны $CB_i(f)$, а множество $D/\bigcup_k C_k(n)$ образует линии водораздела.

Выбор каждого регионального минимума в качестве центра некоторого бассейна приводит к пересегментации изображения. Однако следует заметить, что не все региональные минимумы имеют важное значение. Некоторые из них обусловлены наличием шума, другие – присутствием несущественных деталей в изображении. Чтобы избежать пересегментации, часто используются не все локальные минимумы, а только некоторые из них (либо просто некоторые начальные маркеры). В этом случае количество полученных областей равно количеству выбранных начальных маркеров. На Рис. 35 показан результат применения преобразования водораздела для полутонного изображения.

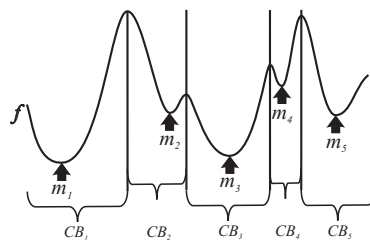


Рис. 33: Иллюстрация преобразования водораздела на топографической поверхности

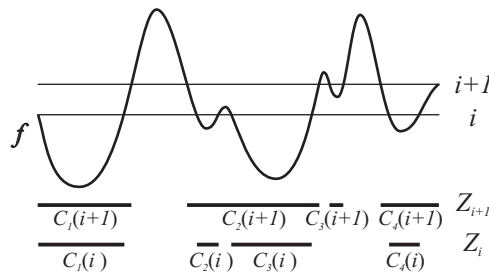


Рис. 34: Пример работы алгоритма водораздела

Алгоритм с упорядоченными очередями

Большинство существующих алгоритмов, реализующих преобразование водораздела имеют ряд недостатков, таких, как большая временная сложность вычислений либо недостаточная точность. Алгоритм, изложенный в данной работе, использует специальную структуру данных, которая позволяет эффективно реализовать преобразование водораздела. Этот алгоритм является легко переносимым на различные типы решеток: восьмисвязные, четырехсвязные, шестисвязные, 26-связные и т.д. Обычно на вход алгоритма водораздела подается градиентное изображение, каждый пиксел которого имеет яркость, равную абсолютному значению градиента в соответствующем пикселе исходного изображения.

Будем использовать структуру данных, называемую упорядоченной очередью. Ее достоинство состоит в способности запоминать точки изображения в произвольном порядке и восстанавливать их в порядке заливки рельефа.

Определим порядок поступления и извлечения элементов из упорядоченной очереди:

- Все элементы становятся в упорядоченную очередь в порядке их поступления.
- Все элементы с различными уровнями яркости извлекаются из упорядоченной очереди в порядке неубывания их яркостей.
- Элементы с одинаковым значением яркости обрабатываются по правилу *FIFO* (первый зашел - первый вышел).

Опишем саму упорядоченную очередь. Она может быть представлена в виде набора простых очередей. Входящий в очередь элемент (т.е. пиксел) запоминается в соответствии со своим приоритетом (яркостью). Каждый элемент помещается в конец очереди, номер которой соответствует его приоритету Pr . В каждый момент времени из очереди извлекается один элемент. Сначала элементы выбираются из очереди с номером 0, затем с номером 1 и т.д. Если в очереди с номером Pr не осталось ни одного элемента, то осуществляется переход на очередь с номером $Pr + 1$, а предыдущая очередь закрывается. В случае, когда после закрытия очереди с номером Pr необходимо записать в упорядоченную

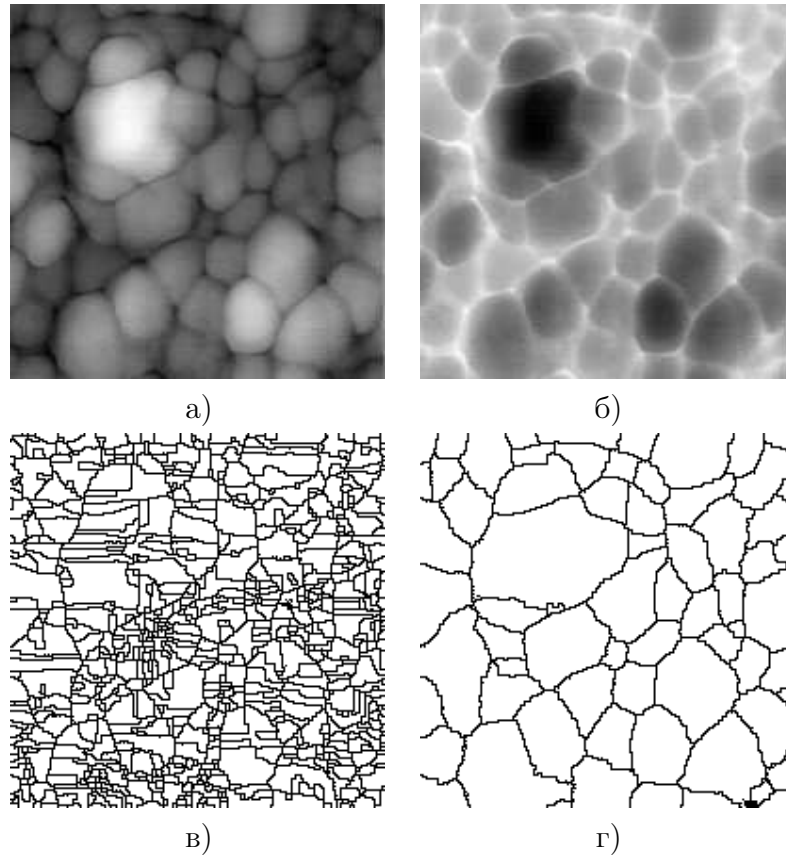


Рис. 35: Применение алгоритма водораздела для сегментации полутонового изображения: исходное изображение а); инвертированное исходное изображение б); избыточная сегментация, полученная в результате использования всех локальных минимумов в качестве маркеров в); использование меньшего числа начальных маркеров, полученных в результате обработки изображения г).

очередь элемент с приоритетом Pr , то он записывается в открытую очередь с минимальным номером. Этот механизм обеспечивает заливку бассейнов, локальные минимумы которых не были выбраны в виде начальных маркеров.

Кратко алгоритм преобразования водораздела, использующий упорядоченную очередь, можно описать следующим образом. Каждый пиксел изображения должен иметь флаг, приоритет и метку. Установленный флаг показывает, что элемент уже извлекался из очереди. Приоритет равен яркости пиксела, а метка показывает к какому собирающему бассейну относится пиксел. Вначале метки всех пикселов равны нулю.

На первом этапе каждый пиксел-маркер m изображения получает метку $\lambda(m)$ и заносится в очередь в соответствии со своим приоритетом (см. Рис. 36). Далее в цикле из очереди в соответствии с дисциплиной обслуживания выбирается один пиксел p и устанавливается его флаг. Каждый сосед q пиксела p , у которого не установлен флаг, получает метку $\lambda(q) = \lambda(p)$ и помещается в очередь в соответствии со своим приоритетом $f(q)$. Алгоритм завершает работу,

когда очередь окажется пустой.

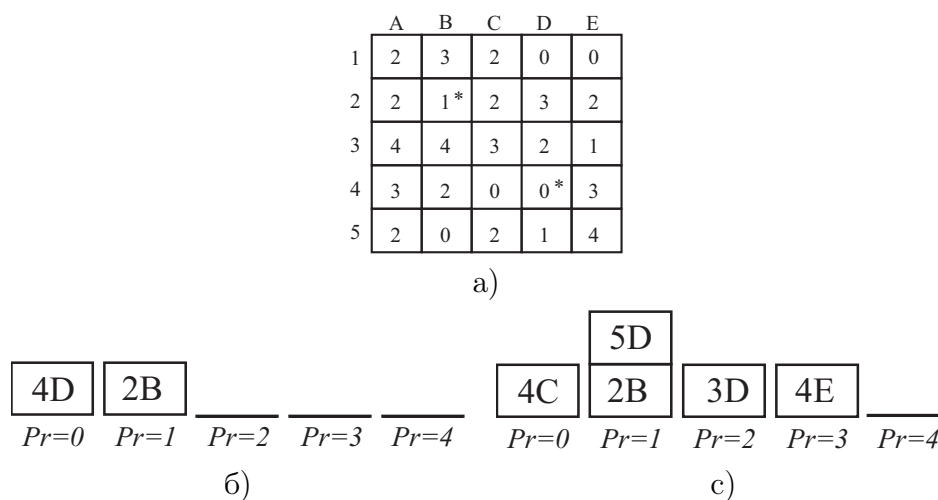
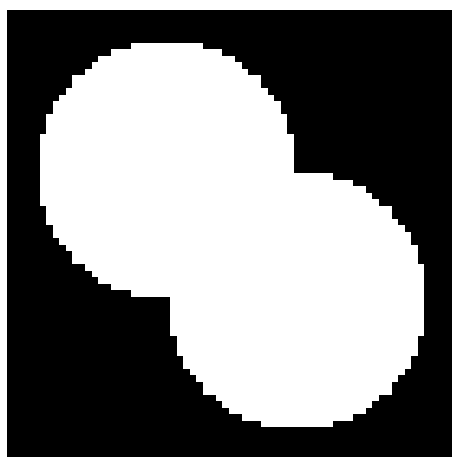
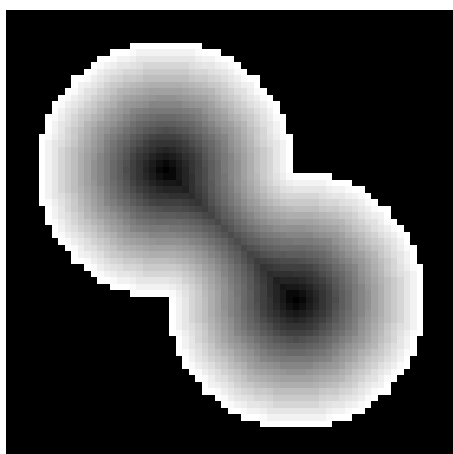


Рис. 36: Упорядоченная очередь: фрагмент изображения а), на котором маркеры выделены *; приоритетная очередь с занесенными маркерами б); очередь после обслуживания первого маркера и занесения его четырехсвязных соседей в).

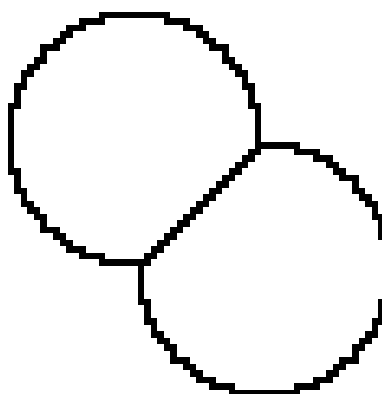
Приведенный алгоритм не генерирует линии водораздела, и собирающие бассейны непосредственно граничат друг с другом. Если необходимо получать и линии водораздела, то алгоритм можно модифицировать. Алгоритм водораздела позволяет, при определенных условиях, сегментировать даже перекрывающиеся объекты, что невозможно сделать, например, градиентными алгоритмами. На Рис. 37 показан пример разделения перекрывающихся объектов алгоритмом водораздела. Начальными маркерами являются локальные минимумы изображения, полученного с помощью дистанционного преобразования.



а)



б)



в)

Рис. 37: Сегментация перекрывающихся объектов алгоритмом водораздела: исходное бинарное изображение а); изображение инвертированной дистанционной функции б); результат алгоритма водораздела в).