

Сегментация изображений

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i,$$

Обнаружение точек

-1	-1	-1
-1	8	-1
-1	-1	-1

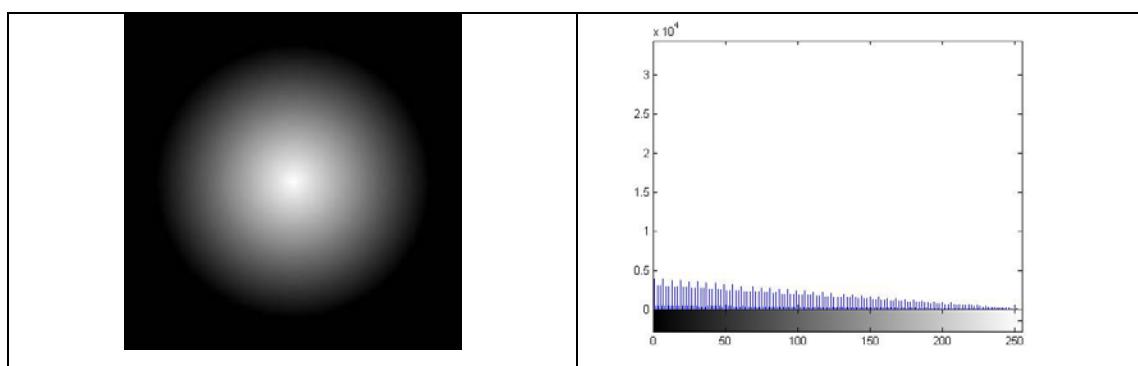
Рис. 10.1. Маска для обнаружения точек

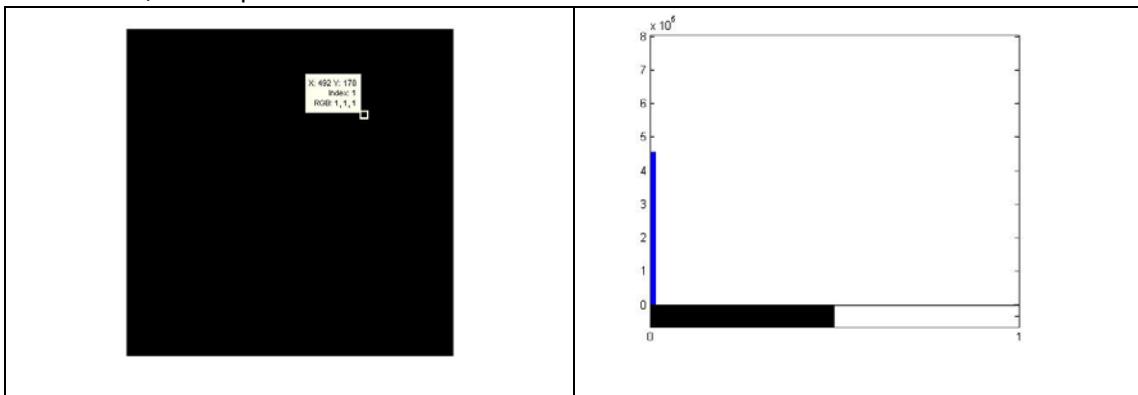
R>T

```
f = imread('Fig1002(a) (test_pattern_with_single_pixel).tif');
imshow(f);
imwrite(f, 'g.1002.a.jpg', 'jpg');
figure, imhist(f);

w = [-1 -1 -1; -1 8 -1; -1 -1 -1];
g = abs(imfilter(double(f), w));
figure, imhist(g);
T = max(g(:));
g = g >= T;
figure, imshow(g);
imwrite(g, 'g.1002.b.jpg', 'jpg');
figure, imhist(g);

m = 10; n = 10;
g1 = imsubtract(ordfilt2(f, m*n, ones(m, n)), ...
    ordfilt2(f, 1, ones(m, n)));
g1 = g1 >= T;
figure, imshow(g1);
figure, imhist(g1);
imwrite(g1, 'g.1002.c.jpg', 'jpg');
```





Обнаружение линий



Рис. 10.3. Маски для обнаружения линий

```
f = imread('Fig1004(a) (wirebond_mask).tif');
imshow(f);
imwrite(f, 'g.1004.a.jpg', 'jpg');
figure, imhist(f);

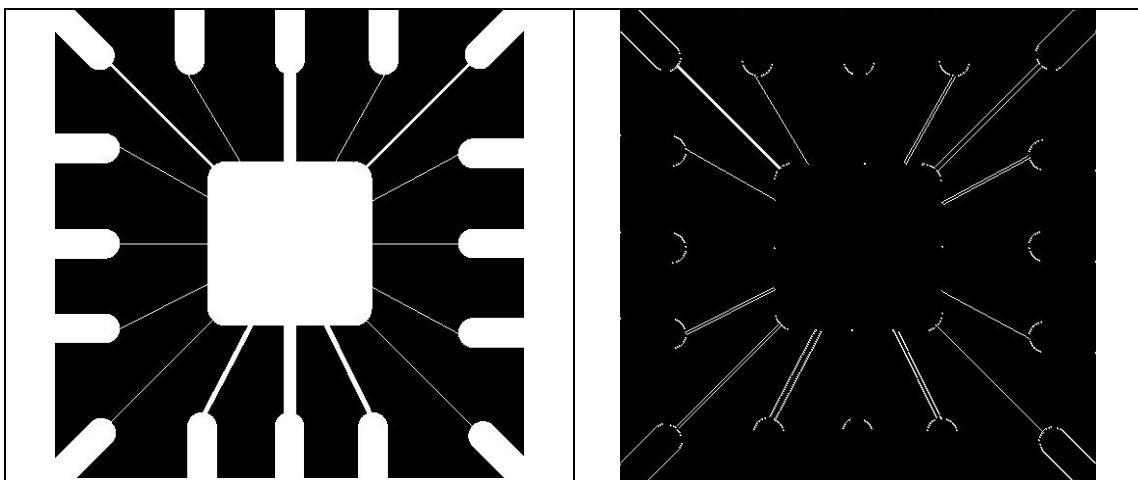
w=[2 -1 -1; -1 2 -1; -1 -1 2];
g=imfilter(double(f),w);
figure, imshow(g);
imwrite(g, 'g.1004.b.jpg', 'jpg');

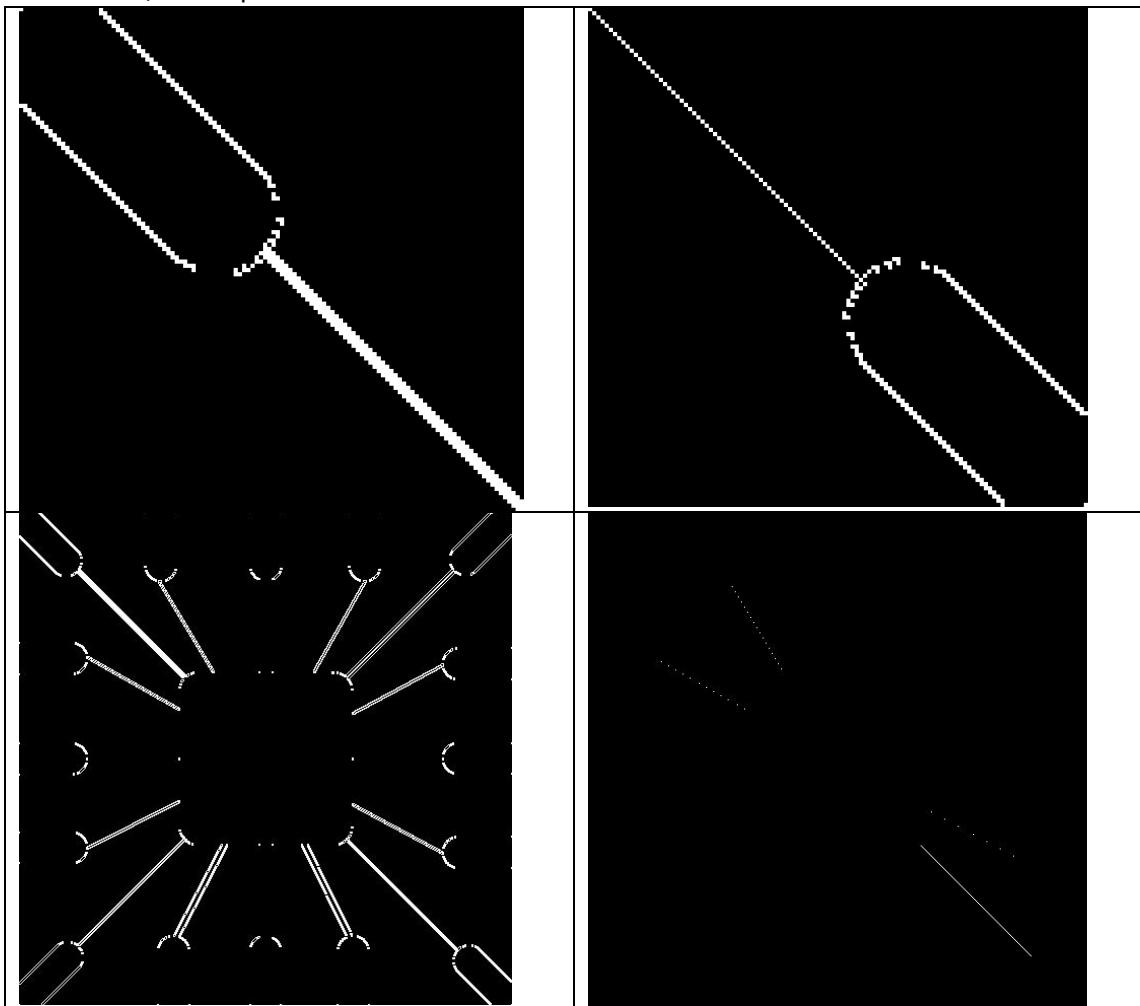
gtop=g(1:120, 1:120);
gtop=pixeldup(gtop,4);
figure, imshow(gtop);
imwrite(gtop, 'g.1004.c.jpg', 'jpg');

gbot=g(end-119:end, end-119:end);
gbot=pixeldup(gbot,4);
figure, imshow(gbot);
imwrite(gbot, 'g.1004.d.jpg', 'jpg');

g=abs(g);
figure, imshow(g);
imwrite(g, 'g.1004.e.jpg', 'jpg');

T=max(g(:));
g = g>=T;
figure, imshow(g);
imwrite(g, 'g.1004.d.jpg', 'jpg');
```





Обнаружение перепадов

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

Модуль вектора градиента равен

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}.$$

Таблица 10.1. Детекторы краев, доступные в функции `edge`

Детектор краев	Основные свойства
Собела	Обнаруживает края с помощью приближений Собела первых производных, заданных на рис. 10.5, б).
Превитта	Обнаруживает края с помощью приближений Превитта первых производных, заданных на рис. 10.5, в).
Робертса	Обнаруживает края с помощью приближений Робертса первых производных, заданных на рис. 10.5, г).
Лапласиан гауссиана	Обнаруживает края, выполняя поиск пересечений нулевого уровня после фильтрации $f(x, y)$ гауссианом.
Пересечения нулевого уровня	Обнаруживает края, выполняя поиск пересечений нулевого уровня после фильтрации $f(x, y)$ фильтром, заданным пользователем.
Канни	Обнаруживает края, выполняя поиск локальных максимумов градиента $f(x, y)$. Градиент вычисляется от гауссиана. Метод использует два порога для нахождения сильных и слабых краев. Слабые края включаются в выход, если они связаны с сильными. Следовательно, этот метод с большей вероятностью обнаруживает настоящие слабые края.

a)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Окрестностные маски детекторов

б)

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Детектор Собела

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

в)

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Детектор Превитта

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

г)

-1	0
0	1

0	-1
1	0

Детектор Робертса

$$G_x = z_9 - z_5$$

$$G_y = z_8 - z_6$$

Рис. 10.5. Некоторые маски детекторов и реализуемые ими формулы приближения для первых производных

```
f=imread('Fig1006(a)(building).tif');
imshow(f);
imwrite(f,'g.1006.a.jpg','jpg');
figure, imhist(f);
```

```
[gv, t]=edge(f, 'sobel', 'vertical');
figure, imshow(gv);
imwrite(gv,'g.1006.b.jpg','jpg');
t
```

```
gv=edge(f, 'sobel', 0.15, 'vertical');
figure, imshow(gv);
imwrite(gv,'g.1006.c.jpg','jpg');
```

```
gboth=edge(f, 'sobel', 0.15);
figure, imshow(gboth);
imwrite(gboth,'g.1006.d.jpg','jpg');
```

```
w45=[-2 1 0; -1 0 1; 0 1 2]
g45=imfilter(double(f), w45, 'replicate');
```

(C) Александр Баженов 2009

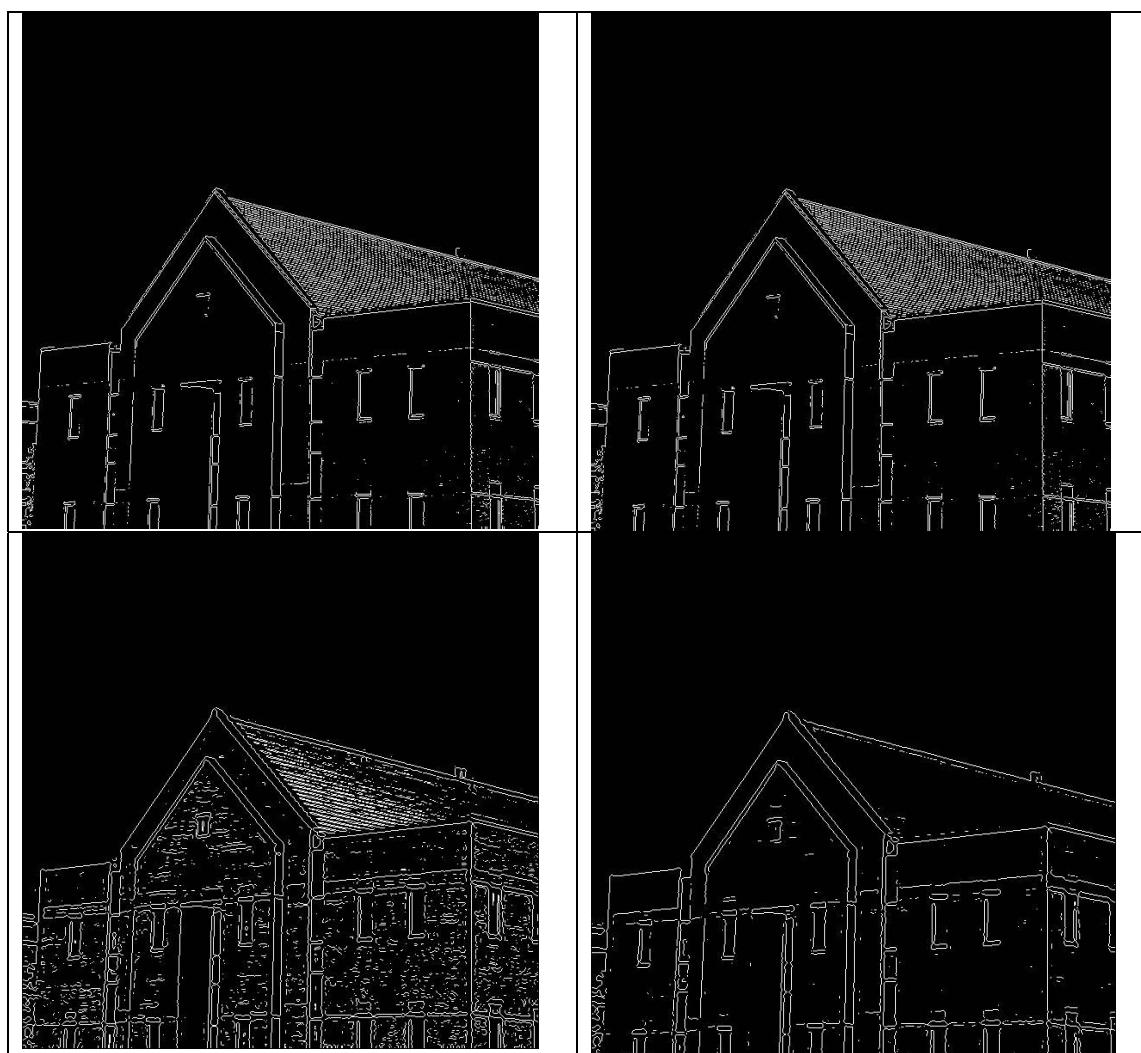
Сегментация изображений

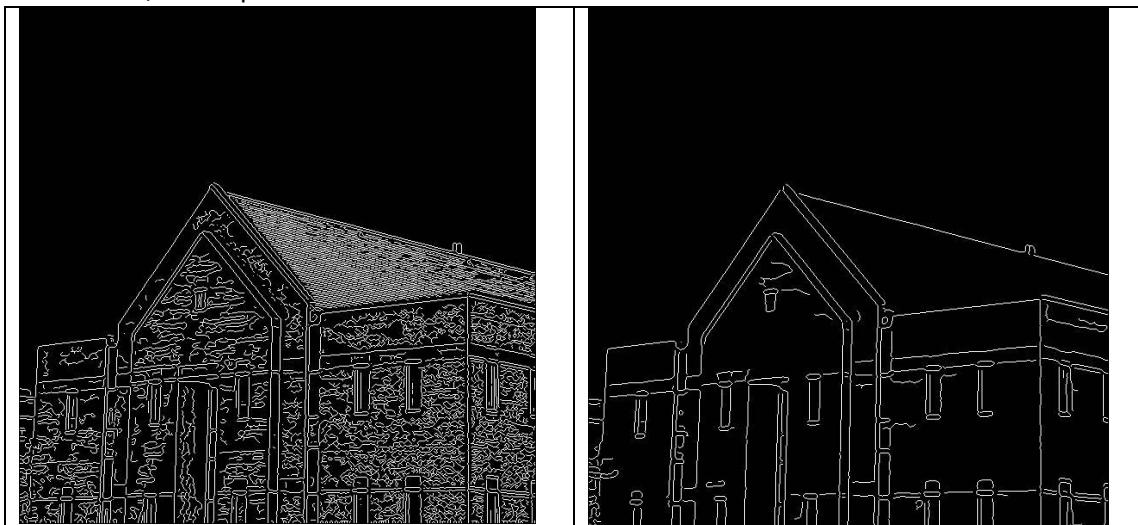
```
T=0.3*max(abs(g45(:)));
g45=g45>=T;
figure,imshow(g45);
imwrite(g45,'g.1006.e.jpg','jpg');
```

```
[g_sobel_default, ts]=edge(f, 'sobel');
figure,imshow(g_sobel_default);
imwrite(g_sobel_default,'g.1007.a.jpg','jpg');
```

```
[g_log_default, tlog]=edge(f, 'log');
figure,imshow(g_log_default);
imwrite(g_log_default,'g.1007.b.jpg','jpg');
```

```
[g_canny_default, tc]=edge(f, 'canny');
figure,imshow(g_canny_default);
imwrite(g_canny_default,'g.1007.c.jpg','jpg');
```





ts
tlog
tc

```
g_sobel_fit=edge(f, 'sobel', 0.05);  
figure,imshow(g_sobel_fit);  
imwrite(g_sobel_default,'g.1007.d.jpg','jpg');
```

```
g_log_fit=edge(f, 'log', 0.003, 2.25);  
figure,imshow(g_log_fit);  
imwrite(g_log_fit,'g.1007.e.jpg','jpg');
```

```
g_canny_fit=edge(f, 'canny', [0.04 0.10], 1.5);  
figure,imshow(g_canny_fit);  
imwrite(g_canny_fit,'g.1007.f.jpg','jpg');
```

Обнаружение линий с помощью преобразования Хафа

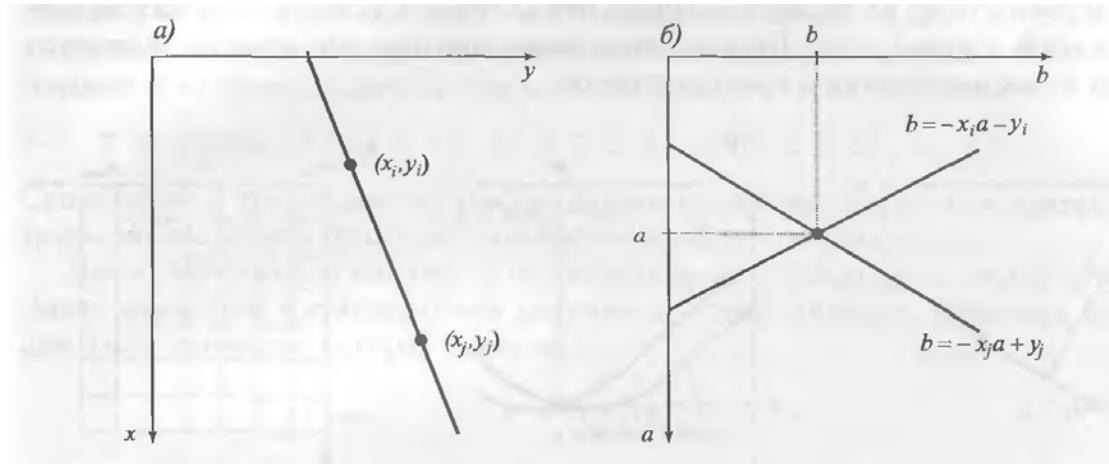
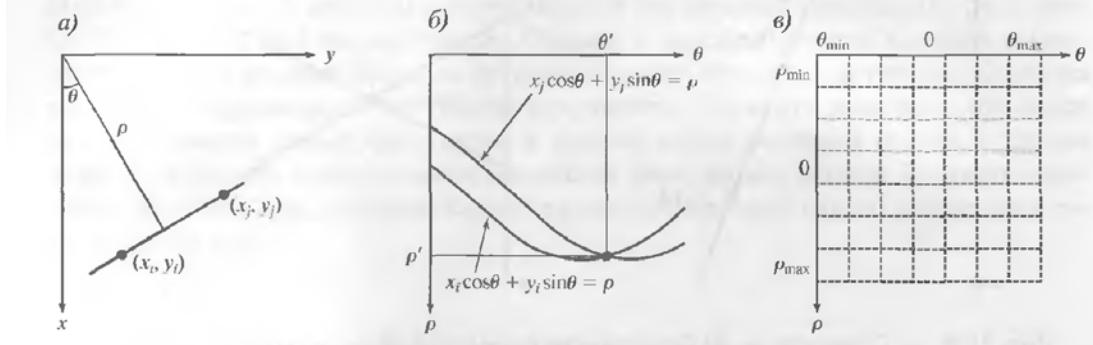


Рис. 10.8. а) Плоскость xy . б) Пространство параметров ab



```

clear;
f=zeros(101,101);
f(1,1)=1;
f(101,1)=1; f(1,101)=1; f(101,101)=1; f(51,51)=1;
imshow(f);
imwrite(f,'g.1010.a.jpg','jpg');
H=hough(f);
imshow(H);
imwrite(H,'g.1010.b.jpg','jpg');
[H, theta, pho]=hough(f);
imshow(theta, pho, H, [], 'notruesize');
axis on, axis normal
xlabel('theta'), ylabel('rho');
grid on

clear;
f=imread('g.1007.e.jpg','jpg');
imshow(f);
[H, theta, rho]=hough(f, 0.5);
figure,
imshow(theta, rho, H, [], 'notruesize');
axis on, axis normal
xlabel('theta'), ylabel('rho');

[r, c]=houghpeaks(H, 5);
hold on
plot(theta(c), rho(r), 'linestyle', 'none', ...
'marker', 's', 'LineWidth', 4, 'color', [1 0 0]);

```

(C) Александр Баженов 2009

Сегментация изображений

```
lines=houghlines(f, theta, rho, r, c);
figure, imshow(f), hold on
for k=1:length(lines)
    xy=[lines(k).point1; lines(k).point2];
    plot(xy(:,2), xy(:,1), 'LineWidth', 4, 'Color', [1 0 0]);
end
```

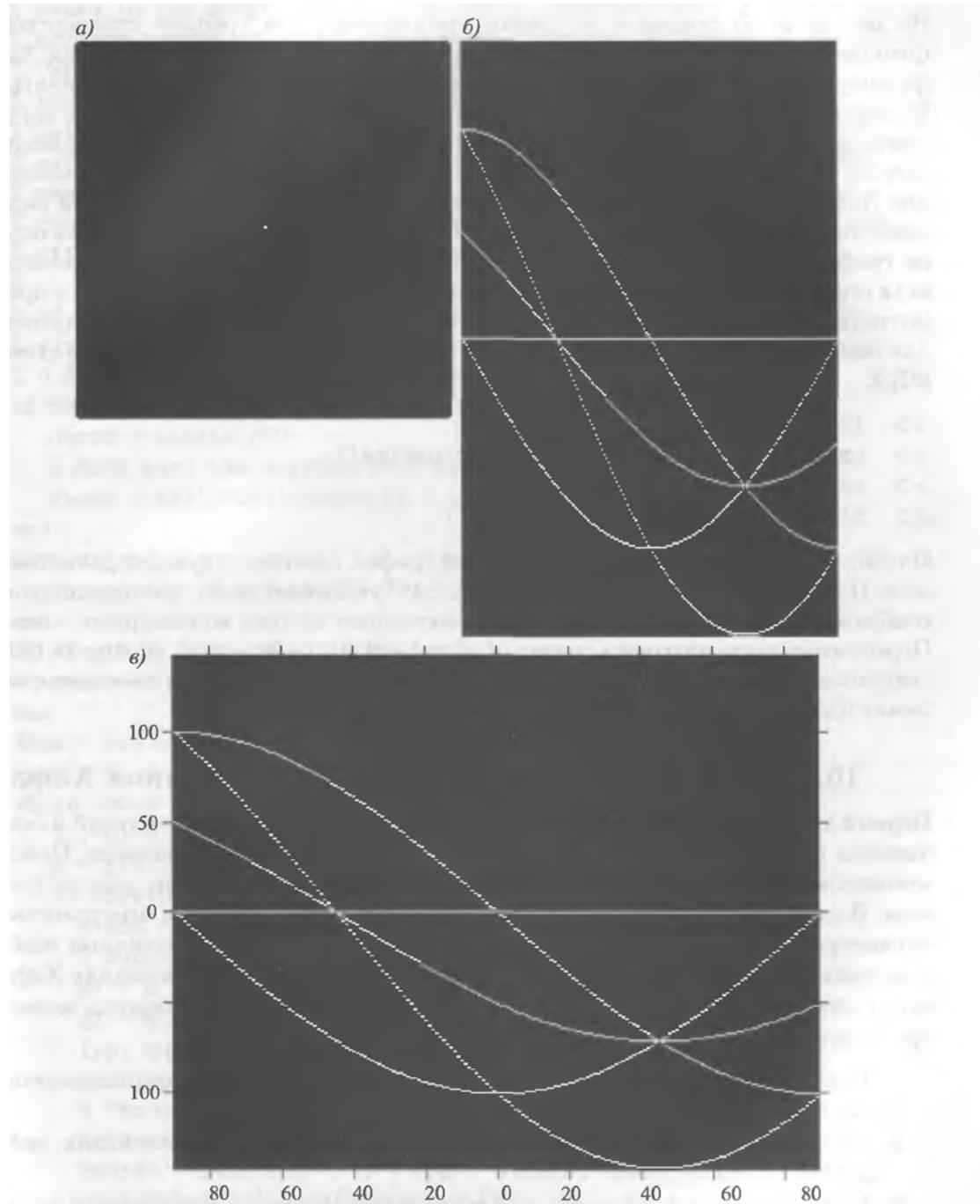


Рис. 10.10. а) Двоичное изображение с пятью точками (четыре точки находятся в углах изображения). б) Преобразование Хафа, показанное с помощью функции imshow. в) Альтернативный график с размеченным осями. (Точки на рис. а) были увеличен для лучшей видимости)

10.2.1. Нахождение максимумов преобразования Хафа

10.2.2. Преобразование Хафа при обнаружении линий и связывании

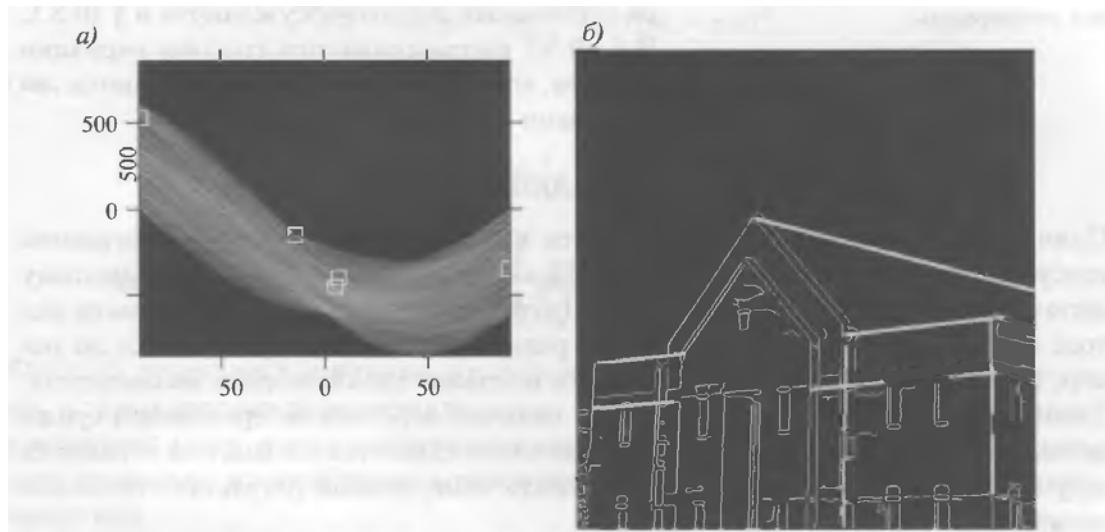
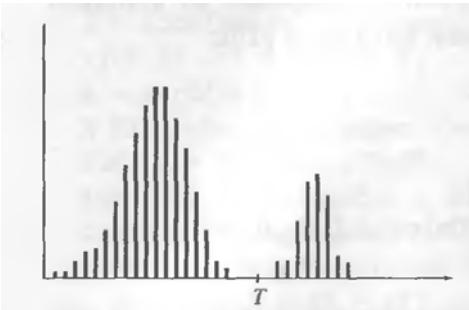


Рис. 10.11. а) Преобразование Хафа с пятью локальными максимумами. б) Сегменты линий, соответствующие максимумам преобразования Хафа

10.3. Пороговая обработка



$$g(x, y) = \begin{cases} 1 & \text{при } f(x, y) \geq T, \\ 0 & \text{при } f(x, y) < T. \end{cases}$$

Рис. 10.12. Выбор порога на основании визуального анализа бимодальной гистограммы

10.3.1. Обработка с глобальным порогом

Для автоматического выбора порога в работе [Gonzalez, Woods, 2002] предложена следующая интерактивная процедура:

1. Выбрать некоторую начальную оценку для значения порога T . (Предлагаемая величина равна среднему значению между максимумом и минимумом яркости изображения).
2. Совершить сегментацию с помощью порога T . В результате образуются две группы пикселов: G_1 и G_2 . Область G_1 состоит из пикселов, яркость которых больше или равно T , а яркость пикселов из G_1 меньше T .
3. Вычислить среднюю яркость пикселов μ_1 и μ_2 по областям G_1 и G_2 .
4. Вычислить новое значение порога:

$$T = \frac{1}{2} (\mu_1 + \mu_2).$$

5. Повторять шаги с 2-го по 4-ый до тех пор, пока разность порогов T для соседних итераций не станет меньше наперед заданного значения T_o .

В пакете IPT имеется функция `graythresh`, которая вычисляет пороги по методу Отсу из [Otsu, 1979]. Чтобы проверить этот метод, основанный на анализе гистограмм, мы представим нормализованную гистограмму в виде дискретной функции распределения вероятностей

$$p_r(r_q) = \frac{n_q}{n}, \quad q = 0, 1, 2, \dots, L - 1,$$

где n — это общее число пикселов изображения, n_q обозначает число пикселов, яркость которых равна величине r_q , а L — число различных уровней яркости изображения. Пусть теперь порог k выбран так, что C_0 обозначает множество пикселов с уровнями яркости из $[0, 1, \dots, k - 1]$, а C_1 — это множество пикселов с уровнями яркости из $[k, k + 1, \dots, L - 1]$. Метод Отсу заключается в выборе порога k так, что он максимизирует *межклассовую дисперсию* σ_B^2 , которая равна

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2,$$

где

$$\begin{aligned} \omega_0 &= \sum_{q=0}^{k-1} p_q(r_q), & \omega_1 &= \sum_{q=k}^{L-1} p_q(r_q), \\ \mu_0 &= \sum_{q=0}^{k-1} qp_q(r_q)/\omega_0, & \mu_1 &= \sum_{q=0}^{k-1} qp_q(r_q)/\omega_1, & \mu_T &= \sum_{q=0}^{L-1} qp_q(r_q)/\omega_0. \end{aligned}$$

```
clear;
f=imread('Fig1013(a)(scanned-text-grayscale).tif','tif');
imshow(f);
imwrite(f,'g.1013.a.jpg','jpg');
figure, imhist(f);
```

```
% thershould computing by Gonzales, Woods
T=0.5*(double(min(f(:)))+double(max(f(:)))) 
done=false;
while ~done
    g = f >= T;
    Tnext=0.5*(mean(f(g))+mean(f(~g)));
    done=abs(T-Tnext) < 0.5;
    T=Tnext;
end
T
```

```
figure, imshow(g);
imwrite(f,'g.1013.a1.jpg','jpg');
figure, imhist(g);
```

```
% thershould computing by Otsu
T2=255*graythresh(f)
g2 = f >= T;
figure, imshow(g2);
figure, imhist(g2);
```

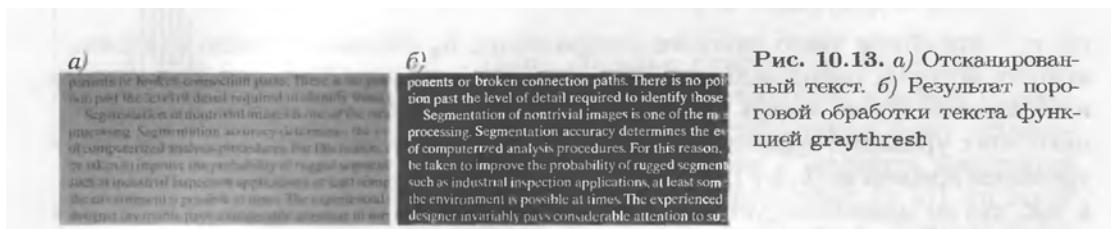


Рис. 10.13. а) Отсканированный текст. б) Результат пороговой обработки текста функцией `graythresh`

10.4. Сегментация на отдельные области

10.4.1. Постановка задачи

Пусть R обозначает всю область изображения. Сегментацией называется процесс разбиения области R на n подобластей R_1, R_2, \dots, R_n таких, что

- a) $\bigcup_{i=1}^n R_i = R;$
- б) R_i является связной областью для любого $i = 1, 2, \dots, n$.
- в) $R_i \cap R_j = \emptyset$ для всех i и j , $i \neq j$.
- г) $P(R_i) = \text{TRUE}$ при $i = 1, 2, \dots, n$.
- д) $P(R_i \cup R_j) = \text{FALSE}$ для любых двух смежных областей R_i и R_j ¹.

Здесь P — это некоторый логический предикат², определенный на точках множества R_i , а \emptyset обозначает пустое множество.

10.4.2. Выращивание областей

```
function [g, NR, SI, TI] = regiongrow(f, S, T)
%REGIONGROW perform segmentation by region growing
f = double(f);
% if S is a scalar, obtain the seed image
if numel(S)==1
    SI = f == S;
    S1 = S;
else
    % S is an array
    SI = bwmorph(S, 'shrink', Inf);
    J = find(SI);
    S1= f(J); % array of seed values
end
TI=false(size(f));
for K=1:length(S1)
    seedvalue=S1(K);
    S = abs(f-seedvalue) <= T;
    TI = TI | S;
end
[g, NR] = bwlabel(imreconstruct(SI, TI));
```

Выходом g служит сегментированное изображение, причем элементы каждой области помечаются одним и тем же целым числом. Параметр NR равен числу выделенных областей. Параметр SI является изображением, на котором отображены центры кристаллизации, а TI — это изображение, содержащее пиксели, которые прошли пороговый тест до их обработки на предмет связности. Оба изображения SI и TI имеют размеры, как у f .

```
clear;
f=imread('Fig1014(a)(defective_weld).tif','tif');
imshow(f);
figure, imhist(f);
```

S=255 % seed values close to max
T = 255-190 % 190 is intensity min close to S

[g, NR, SI, TI] = regiongrow(f, S, T);

```
figure, imshow(SI, []);  
imwrite(SI,'g.1014.b.jpg','jpg');
```

```
figure, imshow(TI, []);  
imwrite(TI,'g.1014.c.jpg','jpg');
```

```
figure, imshow(g, []);  
imwrite(g,'g.1014.d.jpg','jpg');
```

Рис. 10.14. а) Изображение дефектного сварного шва. б) Центры кристаллизации. в) Двоичное изображение, показывающее (белым цветом) все пиксели, прошедшие пороговый тест. г) Результат анализа всех пикселов на предмет 8-связности с центрами кристаллизации. (Исходное изображение предоставлено компанией X-TEK Systems Ltd)

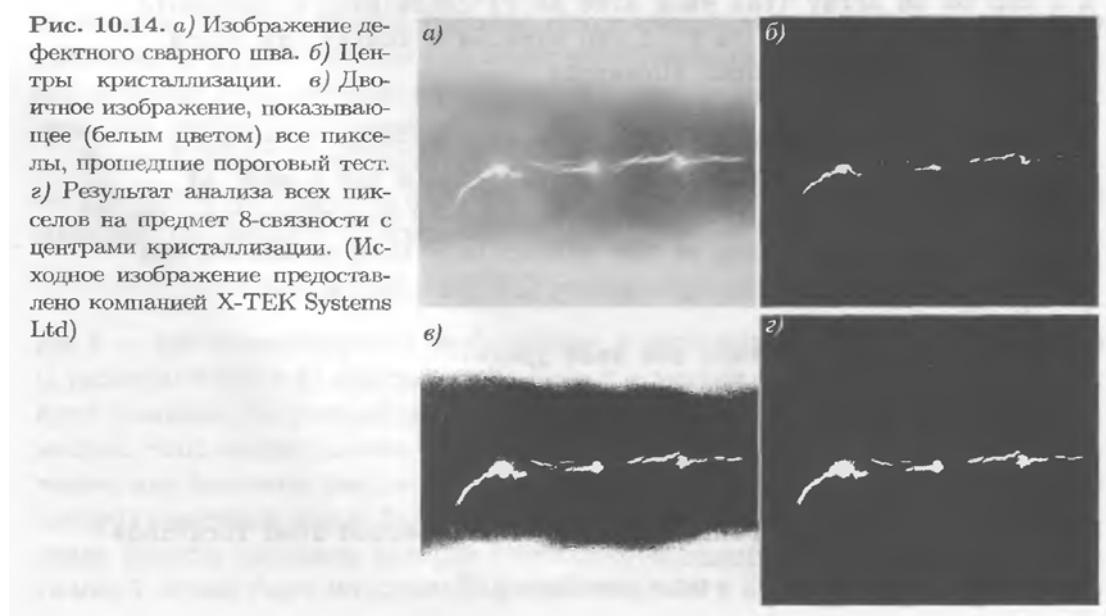
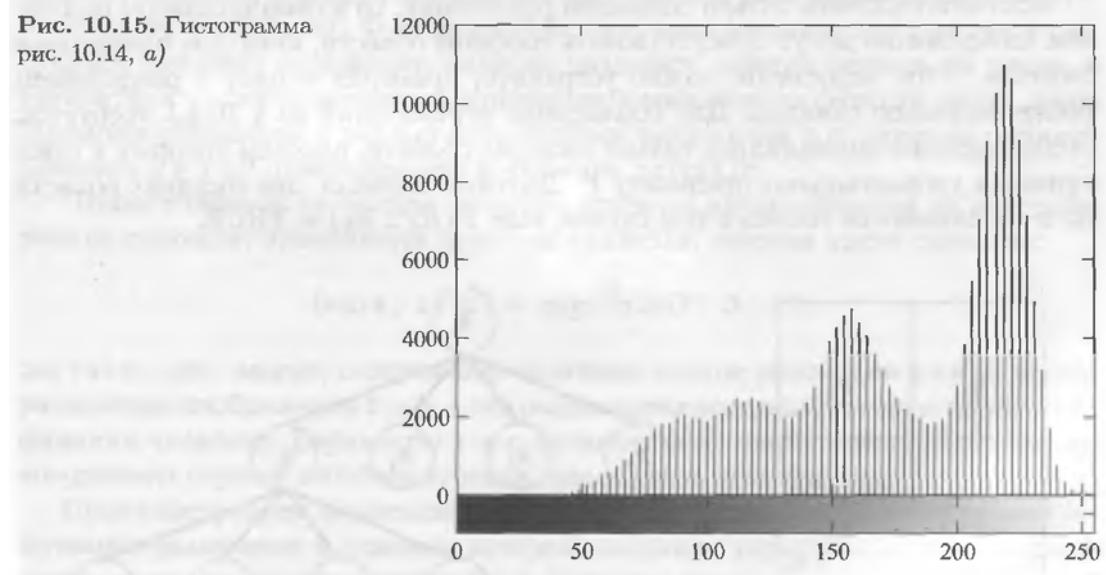


Рис. 10.15. Гистограмма рис. 10.14, а)



10.4.3. Разделение и слияние областей

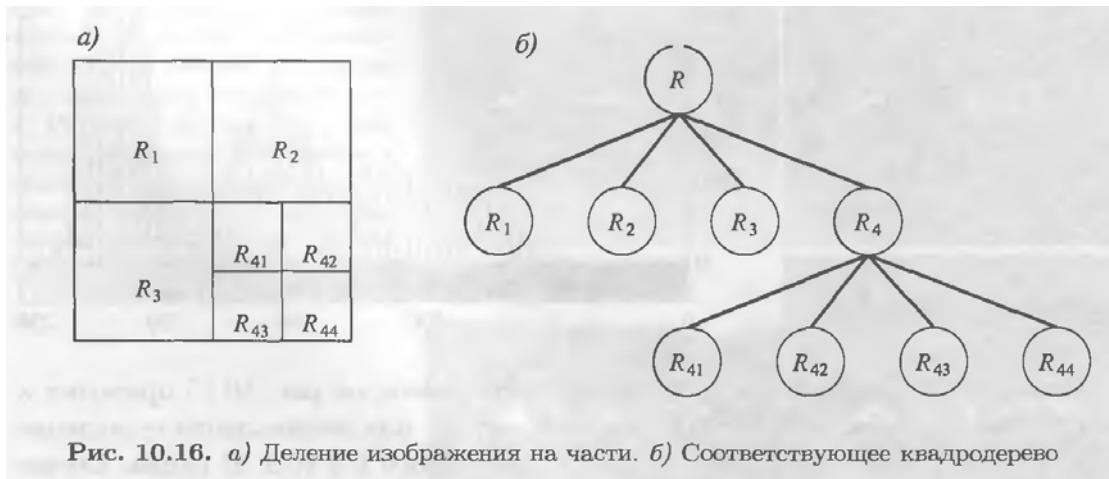


Рис. 10.16. а) Деление изображения на части. б) Соответствующее квадрдерево

- Любая область R_j , для которой $P(R_j) = \text{FALSE}$, разделяется на четыре непересекающиеся четверти.
- Любые две соседние области R_j и R_k , для которых $P(R_j \cup R_k) = \text{TRUE}$, объединяются в одну.
- Если невозможно выполнить ни одной операции слияния или разделения, то процедура завершается.

```

function g = splitmerge(f, mindim, fun);
%SPLITMERGE Segment an image using a split-and-merge algorithm
Q=2^nextpow2(max(size(f)));
[M, N] = size(f);
f = padarray(f, [Q-M, Q-N], 'post');
% Perform splitting first
S = qtdecomp(f, @split_test, mindim, fun);
% Merge
% Get size of largest block
Lmax = full(max(S(:)));
% Set output image to zero
g = zeros(size(f));
MARKER = zeros(size(f));
% Begin merging
for K=1:Lmax
    [vals, r, c] = qtgetblk(f, S, K);
    if ~isempty(vals)
        for I=1:length(r)
            xlow=r(I); ylow=c(I);
            xhigh=xlow+K-1; yhigh=ylow+K-1;
            region=f(xlow:xhigh, ylow:yhigh);
            flog=feval(fun, region);
            if flog
                g(xlow:xhigh, ylow:yhigh)=1;
                MARKER(xlow, ylow)=1;
            end
        end
    end
    % Obtain connected region and label
    g = bwlabel(imreconstruct(MARKER, g));
    % Crop
end

```

(C) Александр Баженов 2009

Сегментация изображений

```
g=g(1:M, 1:N);
%-----
function v = split_test(B, mindim, fun)
k=size(B, 3);
v(1:k)=false;
for I=1:k
    quadregion=B(:, :, I);
    if size(quadregion,1)<=mindim
        V(I)=false;
        continue
    end
    flag=feval(fun, quadregion);
    if flag
        v(I)=true;
    end
end
%-----

clear;
f=imread('Fig1017(a)(cygnusloop_Xray_original.tif');
imshow(f);
imwrite(f,'g.1017.a.jpg','jpg');
figure, imhist(f);

%-----

mindim=32
g = splitmerge(f, mindim, @predicate);
figure, imshow(g);
figure, imhist(g);
```

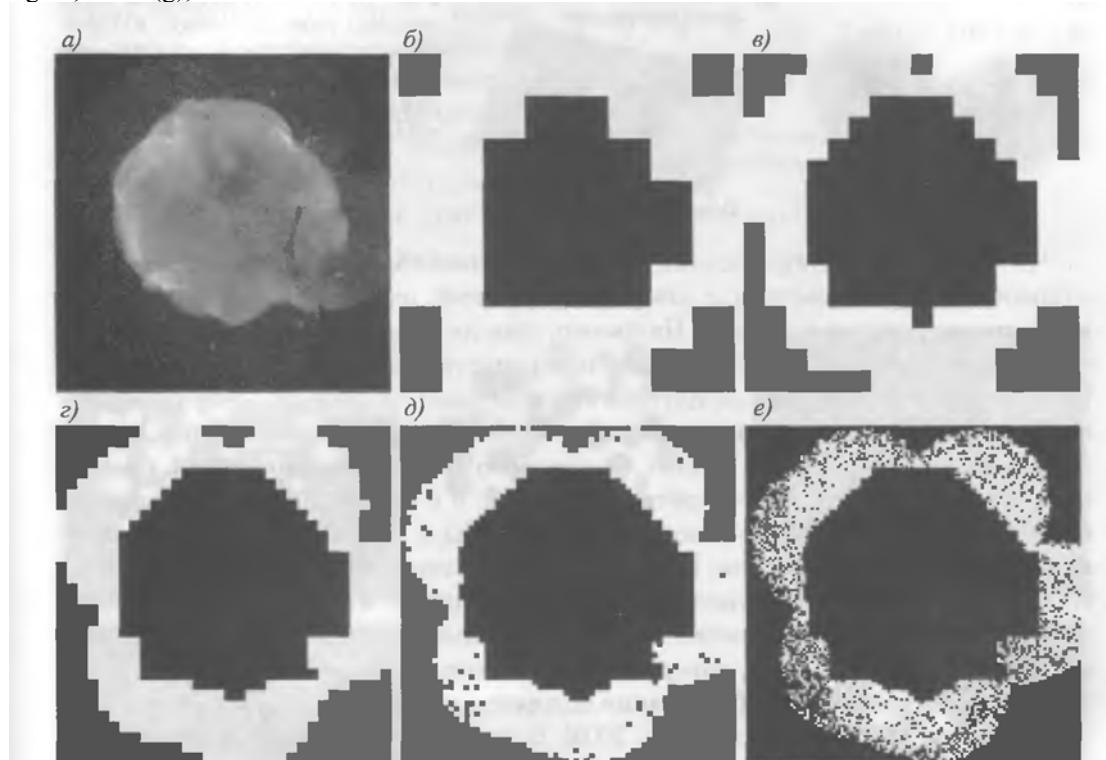


Рис. 10.17. Сегментация изображения методом разделения и слияния. а) Исходное изображение. б)-е) Результаты сегментации функцией `splitmerge` со значениями `mindim`, которые равны 32, 16, 8, 4 и 2 соответственно. (Исходное изображение предоставлено Агентством NASA)

10.5. Сегментация преобразованием водораздела



а)					б)				
1	1	0	0	0	0.00	0.00	1.00	2.00	3.00
1	1	0	0	0	0.00	0.00	1.00	2.00	3.00
0	0	0	0	0	1.00	1.00	1.41	2.00	2.24
0	0	0	0	0	1.41	1.00	1.00	1.00	1.41
0	1	1	1	0	1.00	0.00	0.00	0.00	1.00

Рис. 10.19. а) Маленькое двоичное изображение. б) Преобразование расстояния

```
f = imread('g.929.b.jpg','jpg');
imshow(f);
figure, imhist(f);

% сегментация по водоразделам с помощью преобразования расстояния
g = im2bw(f, graythresh(f));
figure, imshow(g);
imwrite(g,'g.1020.a.jpg','jpg');
%figure, imhist(g);

gc = ~g;
figure, imshow(gc);
imwrite(gc,'g.1020.b.jpg','jpg');

D = bwdist(gc);
figure, imshow(D);
imwrite(D,'g.1020.c.jpg','jpg');
figure, imhist(D);

L = watershed(~D);
w = L == 0;
figure, imshow(w);
imwrite(w,'g.1020.d.jpg','jpg');

g2 = g & ~w;
figure, imshow(g2);

f2 = f;
f2(w) = 255;
figure, imshow(f2);
```

(C) Александр Баженов 2009

Сегментация изображений

imwrite(f2,'g.1020.f.jpg','jpg');

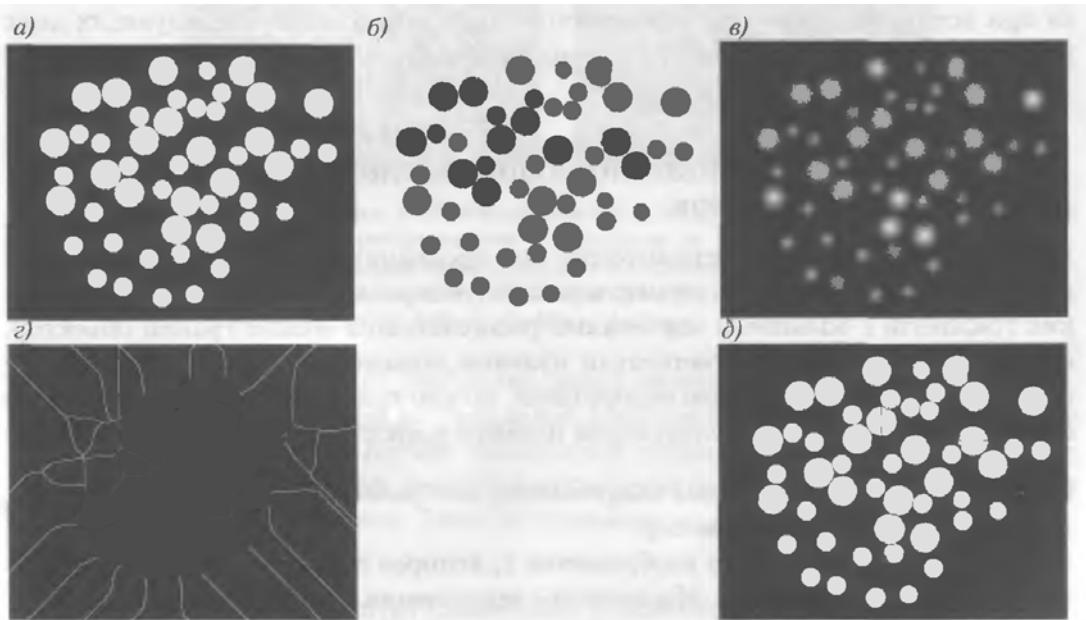


Рис. 10.20. а) Двоичное изображение. б) Дополнение изображения а). в) Преобразование расстоянием. г) Линии водораздела преобразования расстояния от дополнения. д) Линии водоразделов, нанесенные черным цветом на исходное двоичное изображение. Видна некоторая избыточная сегментация

10.5.2. Сегментация по водоразделам с помощью градиентов

```
clear;
f=imread('Fig1021(a)(small-blobs).tif','tif');
imshow(f);
imwrite(f,'g.1021.a.jpg','jpg');
figure, imhist(f);

% сегментация по водоразделам с помощью градиентов
h = fspecial('sobel');
fd=double(f);
g= sqrt(imfilter(fd, h, 'replicate').^2+ ...
    imfilter(fd,h,'replicate').^2);
imshow(g);
imwrite(g,'g.1021.b.jpg','jpg');
figure, imhist(g);

L=watershed(g);
wr=L==0;
figure, imshow(wr);
imwrite(wr,'g.1021.c.jpg','jpg');

g2=imclose(imopen(g, ones(3,3)), ones(3,3));
L2=watershed(g2);
wr2=L2==0;
f2=f;
f2(wr2)=255;
figure, imshow(f2);
imwrite(f2,'g.1021.d.jpg','jpg');
```

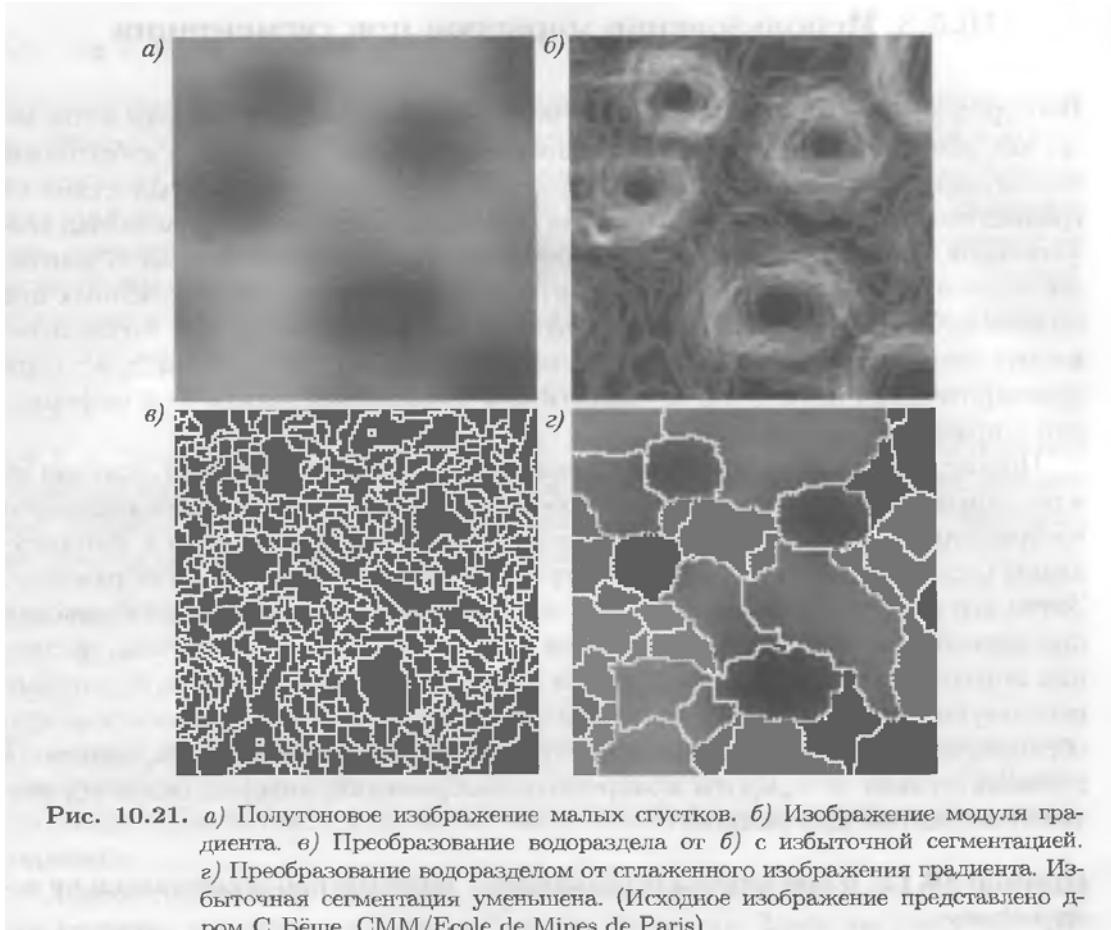


Рис. 10.21. а) Полутоновое изображение малых сгустков. б) Изображение модуля градиента. в) Преобразование водораздела от б) с избыточной сегментацией. г) Преобразование водоразделом от сглаженного изображения градиента. Избыточная сегментация уменьшена. (Исходное изображение предоставлено д-ром С. Бёше, CMM/Ecole de Mines de Paris)

10.5.3. Использование маркеров при сегментации по водоразделам

```
clear;
f=imread('Fig1022(a)(gel-image).tif','tif');
imshow(f);
imwrite(f,'g.1022.a.jpg','jpg');
figure, imhist(f);

% сегментация по водоразделам с помощью градиентов
h = fspecial('sobel');
fd=double(f);
g= sqrt(imfilter(fd, h, 'replicate').^2+ ...
    imfilter(fd,h,'replicate').^2);
L=watershed(g);
wr=L==0;
figure, imshow(wr);
imwrite(wr,'g.1022.b.jpg','jpg');

rm=imregionalmin(f);
imshow(rm);
imwrite(rm,'g.1022.c.jpg','jpg');

im=imextendedmin(f, 2);
fim=f;
fim(im)=175;
imshow(fim);
```

(C) Александр Баженов 2009

Сегментация изображений

imwrite(fim,'g.1022.d.jpg','jpg');

```
Lim=watershed(bwdist(im));
```

```
em=Lim==0;
```

```
figure, imshow(em);
```

```
imwrite(em,'g.1022.e.jpg','jpg');
```

```
g2=imimposemin(f, im | em);
```

```
figure, imshow(g2);
```

```
imwrite(g2,'g.1022.f.jpg','jpg');
```

```
L2=watershed(g2);
```

```
f2=f;
```

```
f2(L2==0) = 255;
```

```
figure, imshow(f2);
```

```
imwrite(f2,'g.1022.h.jpg','jpg');
```

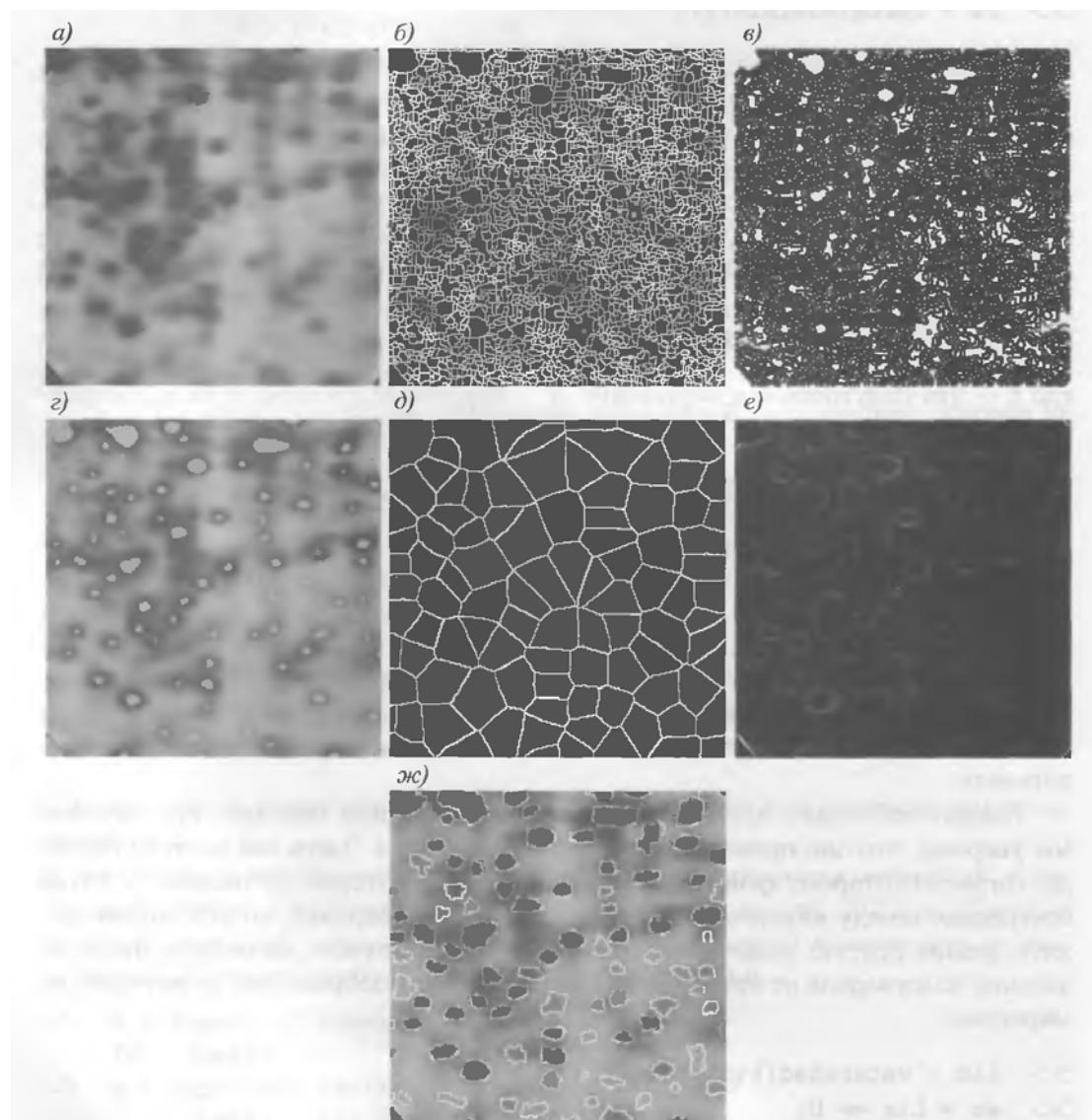


Рис. 10.22. а) Изображение гели. б) Избыточная сегментация в результате применения преобразования водораздела к изображению модуля градиента. в) Локальные минимумы модуля градиента. г) Внутренние маркеры. д) Внешние маркеры. жс) Модифицированный модуль градиента. з) Результат сегментации. (Исходное изображение представлено д-ром С. Бёше, CMM/Ecole de Mines de Paris)