

МОДЕЛИРОВАНИЕ ОБЪЕКТОВ НА ФОНЕ ШУМОВ И АЛГОРИТМОВ ОБНАРУЖЕНИЯ С ПОСТОЯННЫМ ПОРОГОМ

Лабораторная работа № 3

по дисциплине

«Математическое моделирование радиотехнических устройств и систем»

Цель работы: Изучение методов представления, характеристик и алгоритмов формирования полей сигналов и помех, присущих радиотехническим системам и устройствам.

Общая структура программы моделирования наблюдений и алгоритмов обнаружения с постоянным порогом изображена на рис. 1. Программы позволяют сформировать тестовые изображения для исследования характеристик случайных полей, а также для анализа характеристик алгоритмов обработки, в частности, для обнаружения и распознавания объектов на изображениях. Блоки 2 и 3 включают специальные функции, вид которых зависит от используемой модели объекта и шума. Примеры формирования изображений объекта на фоне различных шумов и соответствующие гистограммы приведены на рис. 2.



Рис. 1. Общая структура программы моделирования объектов на фоне шумов и алгоритмов обнаружения с постоянным порогом

Сигнальная область представляет квадрат 49×49 . На рис. 2 слева использована экспоненциальная модель шума $E(30)$, в центре – гауссовский шум $N(30,30)$, справа – логнормальный шум $LN(21,1)$.

В сигнальной области параметр шума отличается так, что обеспечивается заданное значение отношения сигнал/шум в каждом пикселе. В данном случае решающая статистика совпадает с выборкой, поэтому ее дефлексия равна отношению сигнал/шум на входе. Дефлексия решающей статистики по определению есть изменение ее математического ожидания вследствие появления сигнала, отнесенное к среднеквадратическому значению шума.

Алгоритм обнаружения с постоянным порогом является простейшим алгоритмом, преобразующим исходное яркостное изображение в бинарное. В простейшем случае решение о наличии полезного сигнала принимается в каждой точке (пикселе) исходного изображения. Такие процедуры с глобальным порогом (одинаковым для всех пикселей изображения) существуют среди стандартных функций Matlab, например, функция `im2bw` позволяет конвертировать **grayscale** изображение в бинарное с использованием порога `level` со значениями в диапазоне $[0,1]$.

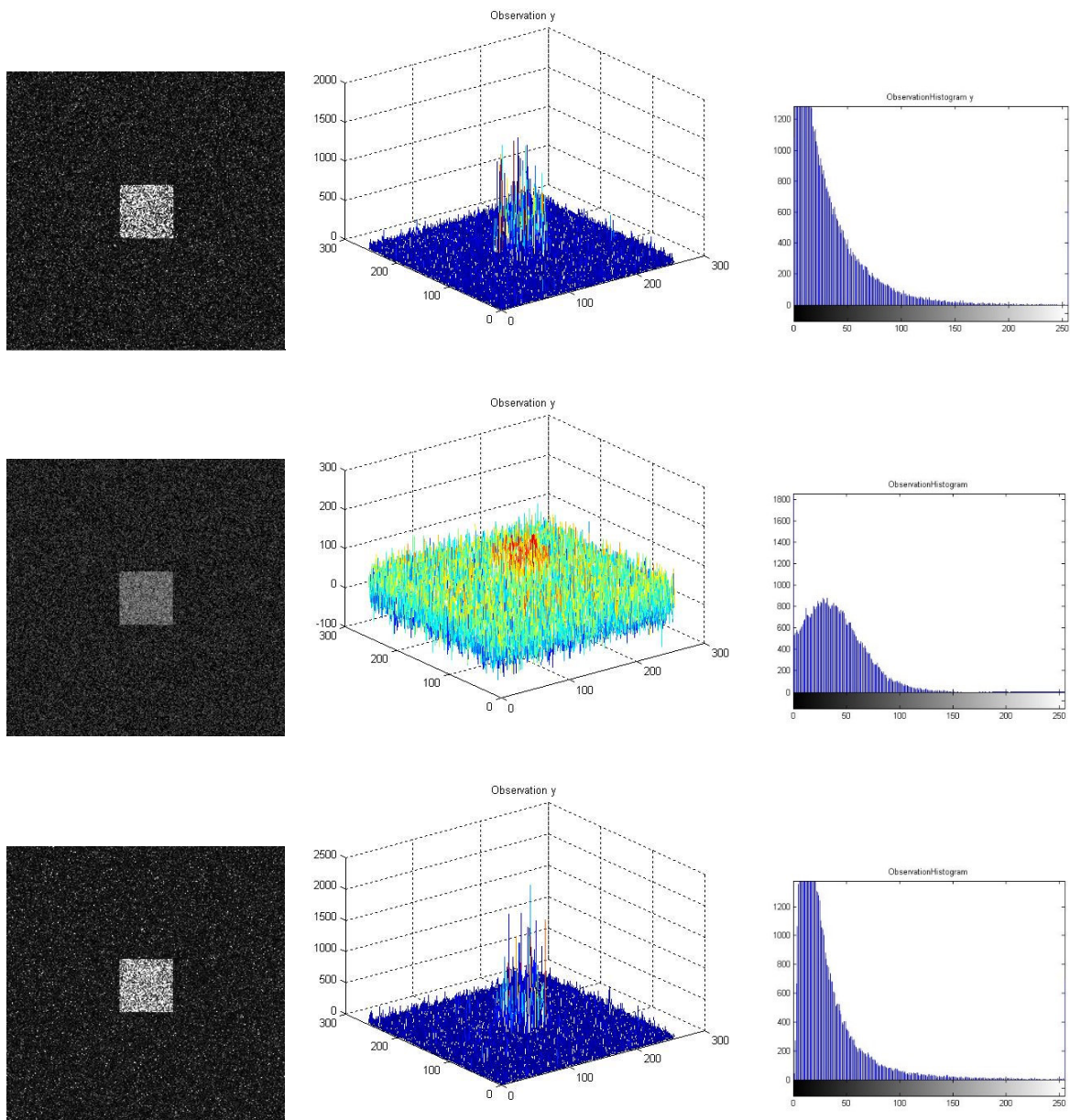


Рис. 2. Изображения объекта на фоне различных шумов и соответствующие гистограммы

Поскольку величина порога нормирована к максимальному значению на изображении, а оно может принимать случайное значение, то возникают трудности при попытке установить требуемое абсолютное значение порога.

В автоматическом режиме порог можно определить с использованием функции `graythresh`. В этом случае применяется метод Отсу, который основан на анализе гистограммы изображения. Предполагается, что любой из пикселей изображения может принадлежать одному из двух классов (*background* – фон или *foreground* – передний план). Значение порога, разделяющего пиксели на два класса по величине интенсивности, минимизирует взвешенную сумму выборочных дисперсий для двух классов (внутриклассовую дисперсию). Метод обладает универсальностью, однако он плохо применим к радиотехническим задачам, в которых требуется использовать критерии оптимальности и учесть априорную информацию о статистических свойствах сигналов и шумов.

В связи с необходимостью контроля и управления процессом обнаружения для радиотехнических задач предлагаются другие алгоритмы, использующие статистические характеристики процессов, а также методы инвариантности и адаптации в условиях априорной неопределенности [2, 3].

Рассмотрим программу `main_exp_const_thr` для моделирования алгоритма обнаружения объекта на изображении при действии шума с экспоненциальным распределением:

```
clear all    %main_exp_const_thr
close all

    STARTING_PROCESS=1    %1_INPUT_IMAGE
m=256; n=256; M=49; N=49; x=0; y=0; %Windows
s=func1_rect(m,n,M,N,x,y);
figure(1), imshow(s,[]), title('InputBinaryImage');
inds=find(s); Ns=size(inds);%NumberOfPoints
    [ms ns]=size(s); %m-vertical
maxs=max(max(s)); %max value of the signal=1
    t=input('STOP_1_InputBinaryImage_Fig1','s');
        STEP=2    %2_ExpThrCalculation
F0=0.01; D0=0.5; beta0=30;
[thr0,d0]=func2_exp_const_thr_calc(F0,D0,beta0);
    thr0
    d=d0    %d=d0 - theoretical value
        t=input('STOP_2_InsertTheorValueDeflection','s');
            STEP=3    %3_EXP NOISE GENERATION
    [z0 zd]=func3_exp_noise_gen(m,n,beta0,d);
%z0, zd - matrices (m,n), m-vertical
z01=uint8(z0); z02=z01*5;
zd1=uint8(zd); zd2=zd1;
% figure(3), imshow(z02,[]), title('ExpNoise z0');
% figure(31), imhist(z01);
% figure(32), imshow(zd2,[]), title('ExpSignal zd');
% figure(33), imhist(zd1);
        STEP=4    %4_OBSERVATION
for i=1:m
```

```

        for j=1:n
            y(i,j)=zd(i,j)*s(i,j)+z0(i,j)*(1-s(i,j));
        end;
end;          %whos y
% may=max(max(y))      %max value of observation
% miy=min(min(y))      %min value of observation
%figure(41), imshow(y,[]), title('Observation');
%figure(42), imhist(y);
y2=uint8(y); % whos y2      %get image format
% mmy2=max(max(y2))
figure(40), imshow(y2,[]); title('Observation y');
figure(43), imhist(y2), title('ObservationHistogram y');
figure(401), mesh(y), title('Observation y');
    t=input('STOP_4_GettingObservation_Fig40','s');
        STEP=5 %5_FalseAlarmConstThreshold
        thr=thr0 %theoretical value
        t=input('STEP_5_InsertTheorValueConstThr','s');
    [F I H A]=func4_bin(z0,thr,0); %whos I
F %FalseAlarms
A %number of detections
H %cell number of window
figure(5), imshow(I,[]), title('FalseAlarmConstThresh');
    t=input('STOP_5_FalseAlarmConstThresh','s');
        STEP=6 %RightDetectionConstThr
    [Ds Ps Bs As]=func5_det_sign(y,thr,s);%whos P
Ds % Detections
figure(6), imshow(Ps,[]),
title('SignalDetConstThrSignalField');
        FINAL_STEP=7 %DetectionConstThr
    [D P B C ]=func4_bin(y,thr,0); %whos P1;
    [D,B,C]
figure(7), imshow(P,[]), title('DetConstThresh');
    t=input('FINAL_STOP_7_NP_DetConstThr_Fig7','s');
imwrite(s,'s.jpg'); imwrite(y2,'y.jpg');
imwrite(I,'I.jpg'); imwrite(P,'P.jpg');

```

На первом шаге формируется бинарное изображение объекта с использованием функции `func1_rect`. На втором шаге программы по заданным вероятностям ложной тревоги F_0 и правильного обнаружения D_0 производится вычисление теоретического значения порога обнаружения `thr0` с использованием критерия Неймана–Пирсона. Также вычисляется пороговое значение дефлексии `d0` решающей статистики `y`. Эти величины вычисляются функцией `func2_exp_const_thr_calc` по формулам, приведенным в [1]:

```

function [ thr,d ] = func2_exp_const_thr_calc(F,D,beta0)
%const threshold and thr deflection calculation
a=-log10(F); e1=exp(1); c=log10(e1);
thr=a*beta0/c; b=-log10(D); d=a/b-1;

```

На третьем шаге программы генерируются два массива случайных чисел, один из которых `z0` соответствует экспоненциальному шуму с параметром b_0 ,

а другой массив **zd** – шуму с параметром $b_1 = b_0(d+1)$, где значение отношения сигнал/шум **d** может быть задано произвольно. Если оно задается равным пороговой дефлексии **d0**, то алгоритмом будут обеспечены заданные характеристики обнаружения. Для генерирования шумов используется функция **func3_exp_noise_gen**:

```
function [ z0,zd ] = func3_exp_noise_gen(n,m,beta0,d)
%exp noise matrix (m,n) generation for two beta
%m-vertical
r0=rand([n m]); r1=rand([n m]); beta1=beta0*(d+1);
z0=-beta0*log(1-r0); zd=-beta1*log(1-r1);
```

На четвертом шаге формируется массив наблюдений **y**, в котором фоновая область соответствует массиву **z0**, а область объекта – сигнальная область – массиву **zd**.

На пятом шаге производится бинаризация фонового изображения **z0**, для этого в функции **func4_bin** задается порог **thr**. При использовании критерия Неймана-Пирсона значение **thr** берется равным теоретическому значению **thr0**. Функция позволяет получить бинаризованное изображение **I**, и вычислить количество превышений порога **A**, размер изображения **H**, а также относительное число превышений порога $F = A/H$. Входной параметр **Nshr** используется при изменении (усечении) исходного размера изображения для удаления краевых эффектов, и в данном случае полагается равным нулю.

Выходная величина **F** является оценкой вероятности ложной тревоги по всему изображению:

```
function [ F,I,H,A ] = func4_bin(Y,T,Nshr)
%const threshold comparator for matrix Y
%output matrix I, exceedings number F
W=size(Y); H=W(1)*W(2);
for i=1:1:W(1),
    for j=1:1:W(2)
        I(i,j)=0;
        if (Y(i,j)>T), I(i,j)=1; end;
    end;
end;
A=sum(sum(I));
if Nshr>0 F=A/Nshr; else F=A/H; end;
```

Для определения числа правильных превышений порога в сигнальной области на шестом шаге используется функция **func5_det_sign**

```
function [ Ds,Ps,Bs,As ] = func5_det_sign(Y,T,S)
%right detection counting
%I=1 for detection Y>T, Q=1 is signal indication
%P=1 for right detection in signal field
W=size(S);
for i=1:1:W(1),
    for j=1:1:W(2)
```

```

        I(i,j)=0;
    if (Y(i,j)>T), I(i,j)=1; end;
        Ps(i,j)=0; Q(i,j)=0;
    if S(i,j)>0, Q(i,j)=1; end;
    if (S(i,j)>0)&&(I(i,j)>0), Ps(i,j)=1; end;
    end;
end;
Bs=sum(sum(Q)); As=sum(sum(P)); Ds=As/Bs;

```

Входная величина s определяет сигнальную область, по которой производится оценивание числа превышений порога. Выходная величина Ds является оценкой вероятности правильного обнаружения по сигнальной области.

На седьмом шаге работает функция `func4_bin`, которая оценивает число превышений порога d по всему изображению.

В случае нормального шума в программе изменятся шаги 2 и 3:

```

        STEP=2      %2_NormThrCalculation
F0=0.01; D0=0.5;
mu0=30   %background level
sigma0=30 %background level
    t=input('STOP_InputData','s');
[thr0,d0]=func2_norm_const_thr_calc(F0,D0,mu0,sigma0);
    thr0
    d0%theoretical threshold deviation
    d=d0;% d=d0 - thoretical value
    t=input('STOP_2_InsertTheorValueDeflection','s');
        STEP=3      %3_Normal_NOISE_GENERATION
    d      %deflection value
    gamma=0 %relative var change
[z0 zd]=func3_norm_noise_gen(m,n,mu0,sigma0,d,gamma);
%z0, zd - matrices (m,n), m-vertical
z01=uint8(z0); z02=z01*5; zd1=uint8(zd); zd2=zd1;
figure(30), imshow(z02,[]), title('NormNoise z0');
figure(301), imhist(z01);
figure(31), imshow(zd2,[]), title('NormSignalPlusNoise zd');
figure(311), imhist(zd1);

```

На этих шагах используются функция `func2_norm_const_thr_calc` для вычисления теоретического порога и пороговой дефлексии и функция `func3_norm_noise_gen` для формирования двух массивов случайных чисел:

```

function [ thr,d ] = func2_norm_const_thr_calc(F,D,mu,sigma)
%const threshold and thr deflection calculation for norm noise
    x=icdf('Normal',1-F,0,1); %percentile
    thr=mu+x*sigma;
    y=icdf('Normal',D,0,1);   d=(x+y);

```

Входной параметр γ задает изменение дисперсии, в данном случае при неизменной дисперсии он равен нулю:

```

function [ z0,zd ] =
func3_norm_noise_gen(n,m,mu0,sigma0,d,gamma)
%norm noise matrix (m,n) generation for two mu values
%m-vertical
    g0=randn([n m]); g1=randn([n m]);
    z0=g0*sigma0+mu0;
    mu1=mu0+d*sigma0;
    sigmalsq=sigma0.^2*(gamma+1);
    signal=sqrt(sigmalsq);
    zd=g1*signal+mu1;

```

В случае логнормального шума второй шаг программы:

```

                STEP=2      %2_LognormThrCalculation
F0=0.01;  D0=0.5;
med0=21
kv=1      %med>0    kv>0
mu0=0;  sigma0=1;
t=input('STOP_InsertInputParameters','s');
[thr1,d,d1,E0,stdev0,E1,stdev1,med1] =
func2_Lognorm_const_thr_calc(F0,D0,med0,kv);
[thr1,d,d1,E0,stdev0,E1,stdev1,med1]
t=input('STOP_2_InsertTheorValueDeflection','s');
                STEP=3  %LognormNoiseGeneration
[z0,zd,ro,mul0,mul1,signal,E0,stdev0,E1,stdev1]
=func3_Lognorm_noise_gen(m,n,med0,kv,d1);
ZD=uint8(zd);
figure(3), imshow(ZD,[], title('LognSignPlusNoise'));
figure(31), imhist(zd), title('LognSignPlusNoiseHistogram');
    [ro,mul0,mul1,signal] %ro=E/med
%N(mul,signal)-generic normal distribution, signal=sqrt(2ln(ro))

```

СООТВЕТСТВУЮЩИЕ функции func2_Lognorm_const_thr_calc и
func3_Lognorm_noise_gen ИМЕЮ ВИД:

```

function [thr1,d,d1,E0,stdev0,E1,stdev1,med1] =
func2_Lognorm_const_thr_calc(F,D,med0,kv)
%const threshold and thr deflection calculation for Lognorm
noise, med0-noise median, kv- coeff. var
ro=sqrt(kv.^2+1); mul0=log(med0);
E0=ro*med0;          %NoiseExpectation
stdev0=E0*kv;       %NoiseStDeviation
signal=sqrt(2*log(ro));
x=icdf('Normal',1-F,0,1); %percentile
thr=mul0+x*signal; thr1=exp(thr);%LognormThreshold
y=icdf('Normal',D,0,1); d=(x+y);
mul1=mul0+d*signal; med1=exp(mul1);
E1=ro*med1; %SignPlusNoiseExpectation
stdev1=E1*kv; %SignPlusNoiseStdev
d1=(E1-E0)/stdev0; %deflection

```

```

function[z0 zd ro mul0 mul1 sigmal E0 stdev0 E1
stdev1]=func3_Lognorm_noise_gen(m,n,med0,kv,d1)
% med-median, kv- coeff. var
ro=sqrt(kv.^2+1); mul0=log(med0);
E0=ro*med0; %NoiseExpectation
stdev0=E0*kv; %NoiseStDeviation
sigmal=sqrt(2*log(ro)); E1=E0+d1*stdev0;
stdev1=E1*kv; med1=E1/ro; %SignPlusNoiseMedian
mul1=log(med1);
g0=randn(m,n); g0=g0*sigmal+mul0;
g1=randn(m,n); g1=g1*sigmal+mul1;
z0=exp(g0); zd=exp(g1);

```

Результаты обработки различных полей представлены на рис. 3. Тестовые изображения формировались с использованием моделей экспоненциального шума $E(30)$, нормального шума $N(30,30)$ и логнормального шума $LN(21,1)$ в фоновой области. В сигнальной области 49×49 эти распределения имели другие параметры, определяемые отношением сигнал/шум.

В случае экспоненциального и нормального распределений математические ожидания в сигнальной области увеличивались в соответствии с пороговой дефлексией, т. е. отношением сигнал/шум, при котором обеспечивались заданные значения вероятностей ложной тревоги и правильного обнаружения: $F = 0,01$, $D = 0,5$.

Для экспоненциального распределения пороговая дефлексия равнялась 5,644, для нормального – 2,326. В случае логнормального распределения задавалось отношение сигнал/шум для порождающего нормального распределения равным 2,326. В результате пороговая дефлексия для каждой выборки логнормального распределения была равна 5,937.

Во всех трех случаях выбирался оптимальный порог по критерию Неймана-Пирсона. Из этого следует, что для обнаружения сдвига распределения в сигнальной области с заданной вероятностью D наименьшая пороговая дефлексия требуется для нормального распределения, а другие модели требуют увеличения ее более чем в два раза.

Нижний ряд рисунков показывает результаты пороговой обработки тестовых изображений с использованием функций **graythresh** и **im2bw**. Вследствие различия гистограмм значения порогов по методу Отсу сильно зависят от вида распределения. Для экспоненциальной модели шума относительный порог был равен 0,282, в случае нормального шума 0,157, для логнормального шума – 0,341. Во всех трех случаях метод Отсу давал более низкие пороги, чем те, которые получались по критерию Неймана-Пирсона, что влекло увеличение число ложных тревог на изображении. Для нормального шума порог Отсу оказался слишком низким даже по сравнению с порогом по байесовскому критерию максимального правдоподобия, значение которого равно 0,5.

Программы позволяют сформировать тестовые изображения для исследования характеристик случайных полей, а также для анализа

характеристик алгоритмов обработки, в частности, для обнаружения и распознавания объектов на изображениях. Они дают возможность исследовать влияние параметров основных модельных распределений на характеристики обнаружения для обнаружителя с постоянным порогом. Таким образом, можно определить значимость тех или иных параметров в случае априорной неопределенности [1, 2].

Рассмотренные алгоритмы используют для бинаризации изображения глобальный порог, который вычисляется для всего изображения. При этом никак не учитываются размеры и форма объекта в сигнальной области, поскольку для этого требуется априорная информация о структуре объекта на изображении, и соответствующие алгоритмы предварительного сглаживания.

В связи со сложной и неоднородной структурой изображений обычно применяются локальные пороги, при этом используются методы инвариантности, робастности и адаптации [2, 3]. При неизвестной форме и размерах объектов переходят к универсальным методам выделения кромок (перепадов интенсивности), границ текстур, пятен и других так называемых «точек интереса» или «областей интереса».

СПИСОК ЛИТЕРАТУРЫ

1. Волков, В. Ю. Обнаружение и различение сигналов в радиотехнических задачах: учеб. пособие / В. Ю. Волков ; СПбГУТ. – СПб., 2000.
2. Волков, В. Ю. Адаптивные, инвариантные и робастные методы обнаружения и различения сигналов: учеб. пособие / В. Ю. Волков ; СПбГУТ. – СПб., 2005.
3. Волков, В. Ю. Адаптивные, инвариантные и робастные методы обнаружения и различения сигналов : учеб. пособие : часть 2 / В. Ю. Волков ; СПбГУТ. – СПб., 2008.
4. Волков, В. Ю. Адаптивные, инвариантные и робастные методы обнаружения и различения сигналов : учеб. пособие : часть 3 / В. Ю. Волков ; СПбГУТ. – СПб., 2012

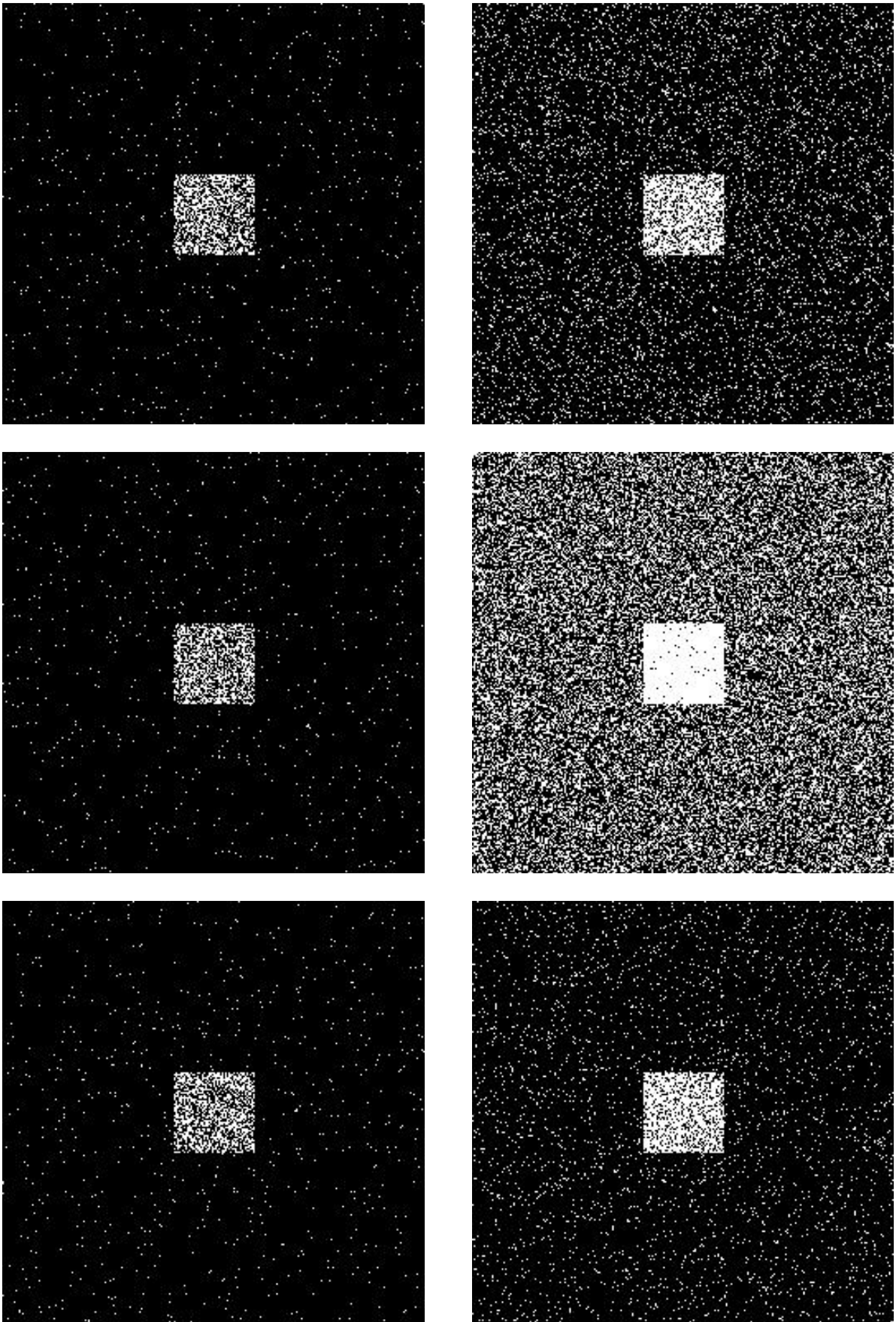


Рис. 3. Результаты обнаружения объекта на фоне различных шумов:
слева – по критерию Неймана–Пирсона, справа – с порогом по методу Отсу