

## **РАСПОЗНАВАНИЕ ОБЪЕКТОВ И ОПРЕДЕЛЕНИЕ КООРДИНАТ В СРЕДЕ MATLAB**

*Цель работы:* Познакомиться со средствами обработки изображений для распознавания графических объектов и определения координат объектов в среде MATLAB.

*Задача работы:* Отслеживание координат наблюдаемого объекта.

*Приборы и принадлежности:* Персональный компьютер. Среда MATLAB. Веб-камера.

### **ВВЕДЕНИЕ**

Системы распознавания широко используются в системах наблюдения и для определения пространственных координат объекта.

Тестирование систем, имеющих доступ к программным или аппаратным каналам вывода информации пользователя не требует распознавания данных. Однако, когда такое подключение к данным объекта отсутствует, его можно выполнить при помощи бесконтактной системы распознавания, которая может обеспечить длительное наблюдение за состоянием объекта в автоматическом режиме.

В этой работе обсуждаются средства распознавания среды MATLAB (без использования нейронных сетей) и средства вычисления координат объекта.

Раздел ОБЩИЕ СВЕДЕНИЯ содержит следующую информацию.

- Библиотечные функции обработки изображений MATLAB
- Характеристики веб камеры, ее подключение к среде MATLAB, настройка режимов камеры.
- Распознавание символов с использованием корреляционных функций.
- Интерфейс пользователя действующей системы распознавания

### **ОБЩИЕ СВЕДЕНИЯ**

#### **Библиотечные Функции Обработки Изображений MATLAB**

Примечание: Более подробное описание функций можно найти в Help MATLAB по названию функций.

MATLAB имеет библиотеки функций для работы с графическими файлами и видеосигналами. Ниже даны используемые варианты библиотечных функций.

#### **Считывание изображения графического файла**

```
>> pct = imread('DC_OS.jpg');
```



Рис. 1. RGB изображение [1] JPG файла в формате <196x259x3 uint8>

Поддерживаемые форматы входных файлов BMP, CUR, GIF, HDF4, ICO, LPEG, JPEG 2000, PBM, PCX, PGM.

### Добавление графического примитива к изображению

Imf = insertShape(pct, 'Rectangle', [32, 27, 220, 153], 'Color', 'green'); % к исходной картинке pct добавлена прямоугольная рамка зеленого цвета.



Рис. 2. Изображение pct с прямоугольной рамкой зеленого цвета, <240x320x3 uint8>.

m-код добавления примитивов к видеокдрам:

```
clear all
```

```
pos_triangle = [183 297 302 250 316 297];
```

```
pos_hexagon = [340 163 305 186 303 257 334 294 362 255 361 191];
```

```
%webcamlist;
```

```
cam = webcam(1);
```

```
for i = 1:100
```

```
    %считывание кадра веб камеры
```

```
    I = snapshot(cam);
```

```
    % добавление круга
```

```
    RGB = insertShape(I, 'circle', [150 280 35], 'LineWidth', 5);
```

```
    % добавление треугольника и многоугольника
```

```
    RGB = insertShape(RGB, 'FilledPolygon', {pos_triangle, pos_hexagon}, 'Color', {'white', 'green'}, 'Opacity', 0.7);
```

```
    figure(2), imshow(RGB); % вывод результата на графопостроитель
```

```
end
```

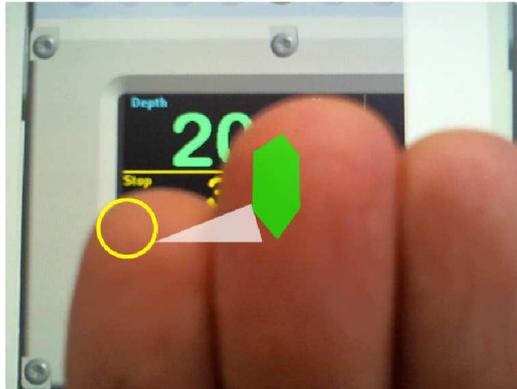


Рис. 3. Видеоизображение с добавленными геометрическими примитивами.

### Уменьшение изображения

Уменьшение изображения в  $2^N$  раз достигается соответствующим уменьшением количества пикселей изображения по каждой координате.

%Установка параметров сжатия в 4 раза (как  $2^2$ )  
`gaussPyramid = vision.Pyramid('PyramidLevel', 2);`

%Считывание изображения jpg файла, перевод в цвета серого (из uint8 в single)

`I = im2single(imread('DC_OS.jpg'));`

`J = step(gaussPyramid, I);` сжатие

%Отображение исходной и сжатой картинок

`figure, imshow(I); title('Original Image');`

`figure, imshow(J); title('Reduced Image');`



Рис. 4. Уменьшение изображения в 4 раза.

### Изменение размера изображения

`pct=imresize(pct,[400 NaN]);` % Изменение размеров матрицы pct изображения с сохранением отношения между сторонами, например, с  $\langle 196 \times 259 \times 3 \text{ uint8} \rangle$  в  $\langle 400 \times 529 \times 3 \text{ uint8} \rangle$

### Выделение фрагмента изображения

`Imf = imcrop(pct_bw, [32, 27, 220, 153]);` % Из файла выделяет фрагмент  $\langle 154 \times 221 \times 3 \text{ uint8} \rangle$

### Преобразование RGB изображения в палитру серого

`pct_bw=rgb2gray(pct);` % Преобразование RGB  $\langle 196 \times 259 \times 3 \text{ uint8} \rangle$  в  $\langle 196 \times 259 \text{ uint8} \rangle$



Рис. 5. Приведение RGB изображения к полутонам серого.

### Фильтрация изображения

2d медианная фильтрация изображений в полутонах серого

`pct_filt=medfilt2(pct_bw,[3 3]); % Медианный фильтр для устранения шума.`

### Размытие изображения

`se=strel('disk',6); % Создание морфологического структурного элемента (STREL)- массив 0 и 1 с расположением единиц в виде круга, эллипса, квадрата`

`pct_dilasi=imdilate(pct_filt,se); % Размытие изображения`



Рис. 6. Примеры размытия левого изображения.

### Утонение сплошных элементов изображения

`pct_eroding=imerode(pct_filt,se); % Eroding the gray image with structural element.`



Рис. 7. Пример утонения сплошных элементов изображения.

## Вычитание константы из изображения или другого изображения

```
pct_edge_enhancement=imsubtract(pct_dilasi,pct_eroding); % вычитание из pct_dilasi pct_eroding
```

Программа подготовки и вычитания изображений:

```
pct = imread('DC_OS.jpg');  
pct_bw=rgb2gray(pct);
```

```
se=strel('disk',1);  
pct_eroding=imerode(pct_bw,se);
```

```
pct_edge_enhancement=imsubtract(pct_bw,pct_eroding);  
figure, imshow(pct_edge_enhancement);
```



Рис. 8. Результат (справа) вычитания среднего изображения из левого.

## Конвертирование матрицы в изображение серых тонов

```
pct_edge_enhancement_double=mat2gray(double(pct_edge_enhancement)); % конвертирование 196 x 258 uint8 [0 .. 255] > 196 x 258 double [0 .. 255] > 196 x 258 double [0 .. 1]
```

## 2d свёртка

```
pct_double_konv=conv2(pct_edge_enhancement_double,[1 1;1 1]); % Свёртка double изображения
```

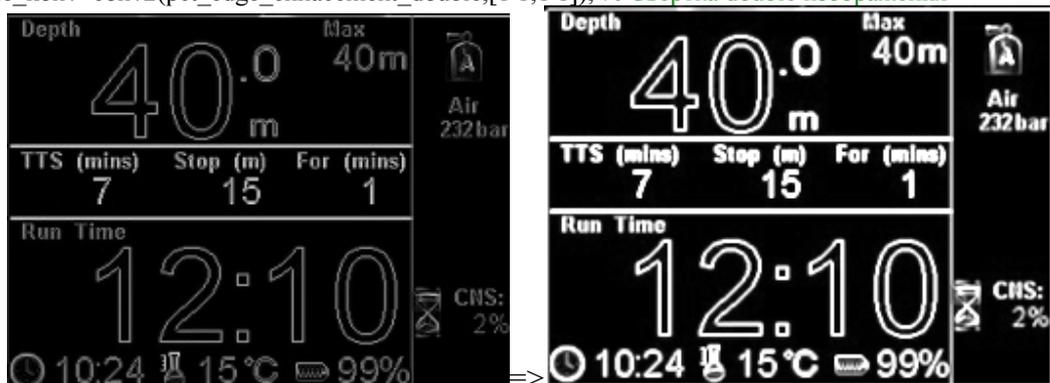


Рис. 9. Действие свертки.

## Масштабирование интенсивности изображения

```
pct_intens=imadjust(pct_double_konv,[0.5 0.7],[0 1],0.1); % Интенсивность [0.5 0.7] или 128 .. 179 масштабируется в [0 1] или 0 .. 255; 0.1 - gamma
```



Рис. 10. Масштабирование интенсивности.

### Преобразование числовой матрицы в логическую

`pct_logic=logical(pct_intens);` % Преобразование `double` или `uint8` в `binary`; все, что `> 0` становится `1`.

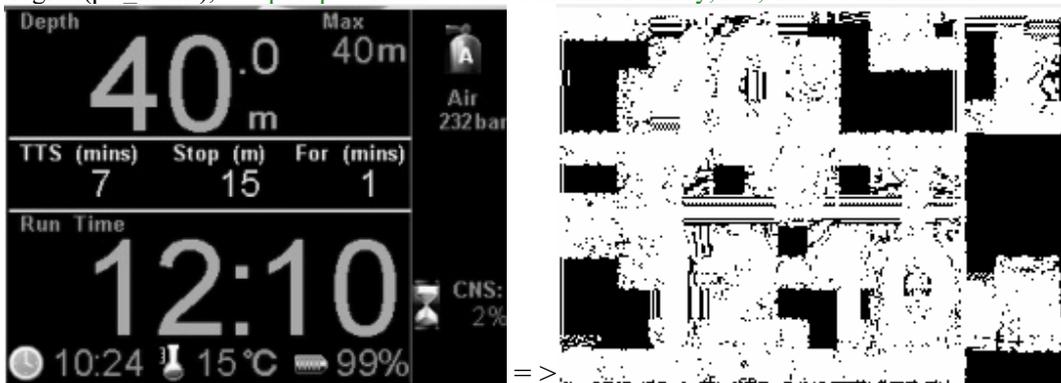


Рис. 11. Преобразование `<196 x 256 double>` матрицы в бинарную `<196 x 256 logical>`

### Заливка оконтуренных элементов

`pct_fill=imfill(pct_line_delete,'holes');`

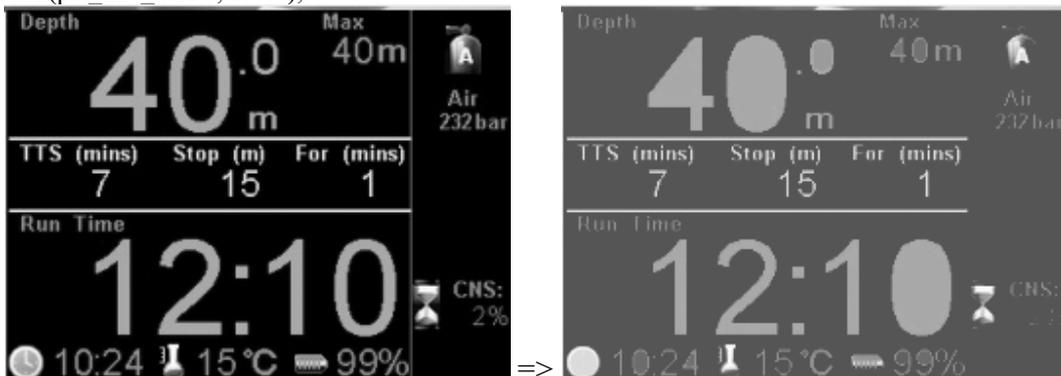


Рис. 12. Заливка элементов с безразрывными контурами `<196 x 256 double>`

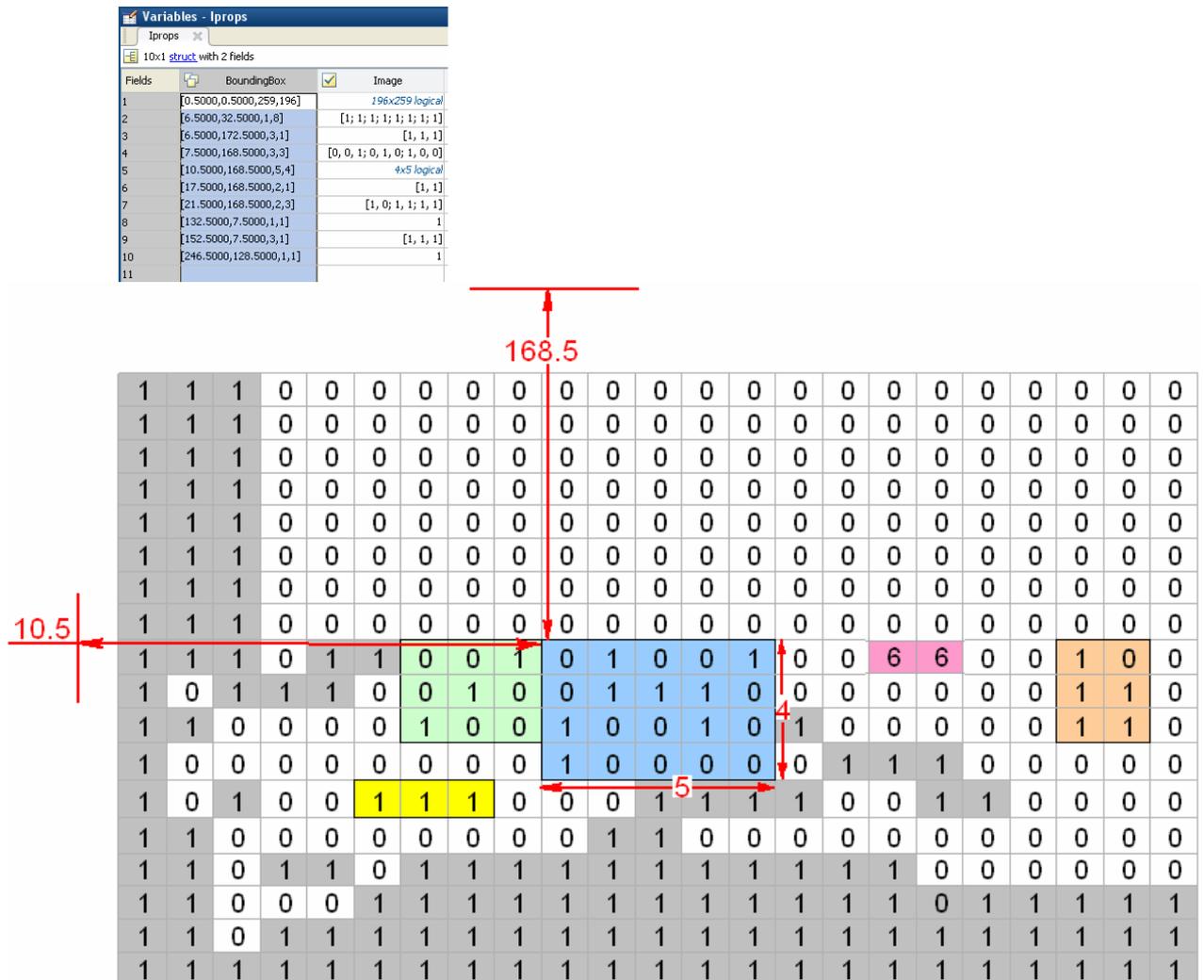
### Удаление сплошных (островковых) фрагментов (шумов) из бинарного изображения

`pct_final=bwareaopen(pct_logic,500);` % удаляет "белые == 1" объекты `> 500` пикселей.



## Нахождение координат и размеров островковых зон

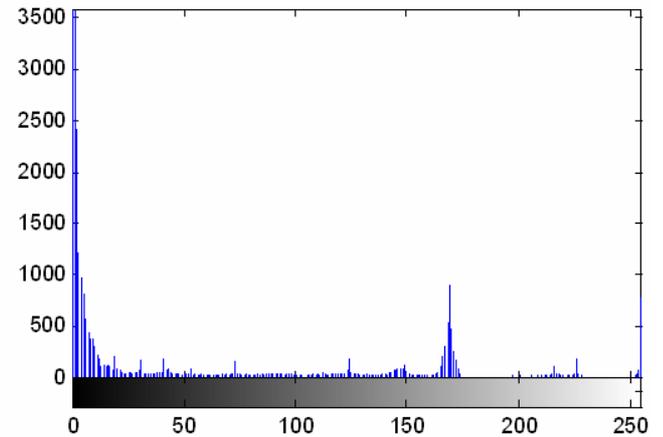
`Iprops=regionprops(labelled,'BoundingBox','Image');`



**Рис. 15.** Фрагмент графического представления структурной переменной `Iprops`, содержащей координаты и размеры прямоугольных зон пронумерованных групп и бинарную матрицу каждой группы. Выделены координаты и размеры пятой группы.

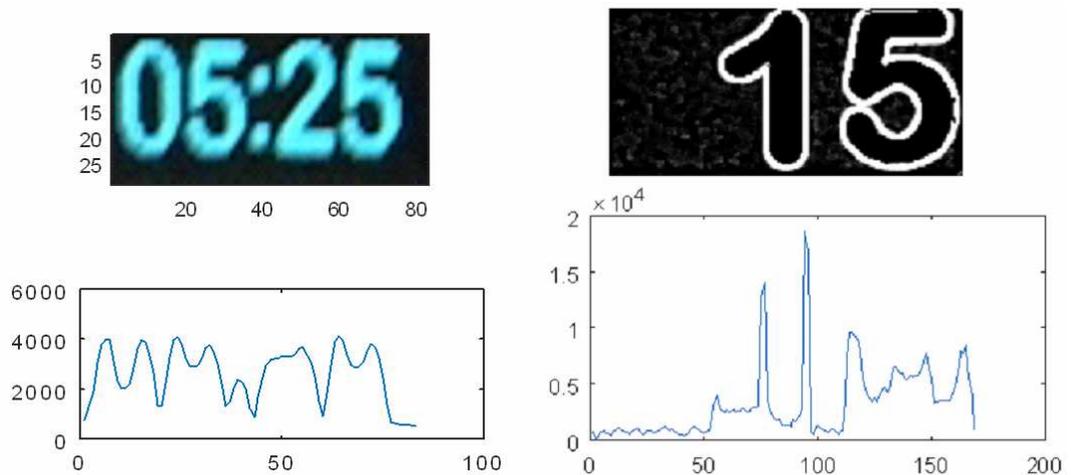
## Гистограмма

Гистограмма `imhist(img_reference)` позволяет получить зависимость количества пикселей для каждой интенсивности.

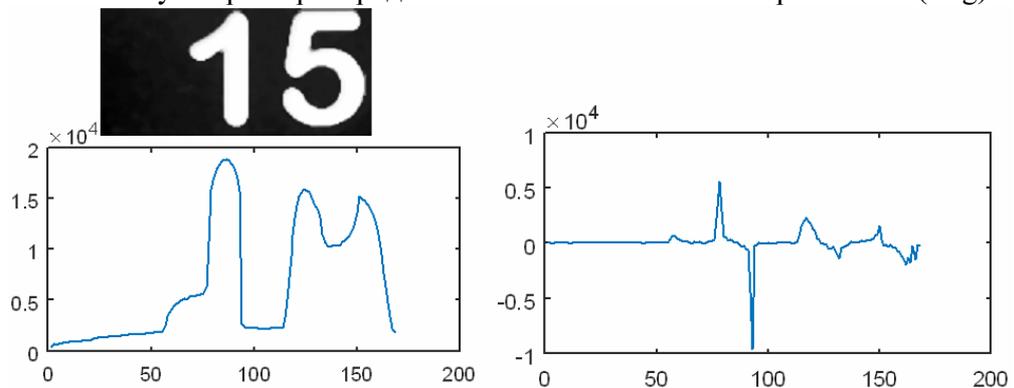


**Рис. 16.** Изображение (слева) и его гистограмма – количество полутонов серого. Гистограмма показывает, что в изображении преобладает черный цвет нулевой интенсивности: ~3500 точек.

**Распределение интенсивностей**



**Рис. 17.** Суммарное распределение интенсивности по строкам  $\text{sum}(\text{Img})$ .



**Рис. 18.** Слева - сумма интенсивности пикселей по строкам  $\text{sum}(\text{Img})$  после медианного фильтра  $\text{Img} = \text{medfilt2}(\text{Img}, [5\ 5])$ . Справа - производная от суммы  $\text{diff}(\text{sum}(\text{Img}))$ .

## Трехмерное построение видеоизображений

Видеокадр можно представить в табличном виде, как это показано на Рис. 19 для символа 9.

	2	2	0	1	1	2	1	5	7	7	7	5	3	5	3	3	3	2	2	2	2	3	1	0	1	0	0	3
3	2	1	0	2	3	3	3	6	23	52	57	54	55	55	54	20	13	3	3	4	1	0	0	0	0	0	0	0
3	0	1	2	3	7	11	50	117	170	198	205	209	202	203	187	146	129	73	24	5	0	1	1	0	0	0	0	1
2	2	3	4	10	54	146	197	207	209	215	215	216	217	215	215	213	211	202	146	56	20	2	2	0	0	0	0	0
4	2	6	7	83	188	213	215	213	214	217	216	218	218	217	218	215	216	211	210	183	102	24	1	0	1	1	0	0
2	4	10	80	198	213	217	218	222	220	220	220	221	220	220	221	221	222	221	220	216	203	121	34	1	1	0	0	0
4	4	77	196	213	216	219	218	222	221	220	216	219	216	218	220	221	222	221	221	217	215	203	102	9	1	0	0	0
5	30	181	213	220	220	221	221	219	216	197	164	148	147	180	206	214	216	221	222	222	220	217	172	47	13	1	0	0
9	107	210	218	219	220	221	221	217	175	118	65	41	40	71	131	198	215	220	222	221	220	217	206	115	18	2	2	0
36	182	213	216	222	222	221	221	189	106	17	10	3	7	9	28	117	198	214	217	222	221	221	219	165	57	6	3	0
67	197	214	216	221	222	221	215	152	54	10	10	1	3	7	6	35	124	211	217	222	222	221	220	206	106	16	7	3
134	215	217	219	221	222	219	189	95	21	5	4	2	2	3	3	10	58	197	216	221	221	222	221	214	151	32	15	3
162	215	218	219	221	221	218	176	49	20	5	4	4	4	3	3	6	23	155	214	220	221	221	221	217	174	63	13	1
167	216	220	221	221	222	217	171	38	14	3	5	2	2	2	2	5	7	133	209	220	219	221	221	220	191	79	17	2
172	216	220	220	221	221	218	169	39	15	3	3	3	3	2	3	6	6	130	215	219	219	222	222	219	216	118	26	4
169	216	219	220	221	221	216	183	48	14	5	5	2	2	2	4	6	7	150	215	222	220	221	220	221	221	146	35	4
145	213	219	219	220	221	217	209	102	17	3	5	2	4	2	5	6	46	198	217	220	220	221	221	221	220	152	36	4
96	206	217	218	220	221	220	215	173	51	7	4	4	4	2	7	14	133	210	216	222	220	222	222	220	220	153	37	4
54	173	215	217	221	221	220	219	208	125	18	7	7	7	7	19	94	201	213	216	222	221	222	222	221	220	151	37	4
24	136	206	215	222	221	222	221	217	204	149	65	32	31	88	169	199	209	218	219	223	222	223	222	222	220	151	41	2
13	68	187	213	221	222	222	221	218	215	210	204	196	192	202	215	211	214	218	218	222	222	223	222	221	221	155	39	4
7	20	121	198	216	216	221	222	222	221	218	216	213	214	216	217	219	219	213	213	218	219	224	223	221	219	155	39	5
4	2	43	133	196	215	221	221	221	221	218	219	216	217	218	218	219	216	185	210	218	220	223	223	221	221	155	38	2
3	2	4	38	126	172	208	214	219	219	221	222	222	222	221	220	203	167	137	202	215	217	222	222	222	217	150	36	1
3	3	4	2	41	97	160	188	206	218	220	220	220	220	205	188	147	87	114	206	217	217	222	221	221	206	116	30	2
3	4	6	6	9	9	63	108	146	162	168	167	165	156	138	102	53	19	119	209	218	218	221	221	219	191	85	15	2
4	4	6	5	3	4	15	17	45	63	70	67	66	66	37	24	14	14	156	213	219	219	221	221	218	181	77	15	2
5	7	27	100	164	167	90	33	9	9	3	3	3	2	2	4	9	30	191	213	220	221	221	221	214	158	45	10	2
5	10	137	206	206	210	204	159	41	7	6	5	4	5	4	4	7	86	208	216	220	220	221	221	204	133	17	12	0
9	68	204	213	217	218	216	206	122	32	6	6	1	4	4	6	21	162	209	214	221	221	221	220	185	86	5	5	1
9	92	197	215	218	218	218	215	202	96	13	10	6	7	7	16	137	202	211	214	221	221	221	209	140	44	5	5	0
4	61	193	218	221	221	222	221	214	197	130	49	20	20	68	154	202	210	219	219	220	220	217	181	87	18	3	3	1
5	24	145	215	221	221	221	222	217	215	205	197	182	185	192	206	210	213	219	218	220	220	200	138	40	16	0	1	1
3	9	107	189	216	216	221	221	221	221	215	214	211	213	215	216	220	220	221	221	218	201	156	77	9	4	1	0	0
2	3	38	137	197	216	221	221	221	221	217	217	216	216	217	218	220	220	221	221	206	158	79	31	6	3	0	0	0
3	3	4	48	137	181	208	210	217	217	221	221	221	220	221	222	219	218	208	194	152	80	12	7	2	0	0	0	3
1	1	6	4	48	114	154	188	203	215	220	220	220	217	220	211	204	194	165	126	68	30	6	6	2	1	1	0	3
2	1	0	4	2	9	52	104	141	164	175	177	176	180	176	156	138	112	75	28	9	4	1	1	0	0	1	1	3
3	1	1	5	5	4	14	19	41	66	84	84	86	86	83	65	36	34	17	16	5	4	0	2	2	0	1	1	2
1	0	2	2	2	3	3	5	3	1	0	3	3	2	3	1	4	2	2	4	1	3	3	2	3	0	2	1	2
1	1	2	2	2	2	3	5	5	2	3	2	4	4	4	5	2	4	4	4	5	3	2	3	4	2	0	1	1

Рис. 19. Табличное распределение интенсивности пикселей символа "9" в Excel.

Графопостроитель MATLAB имеет кнопку "Rotate 3D" (Рис. 20), которая позволяет отображать выводимые данные в 3d формате.



Рис. 20. Кнопка графопостроителя для перевода отображаемых данных в 3d формат.

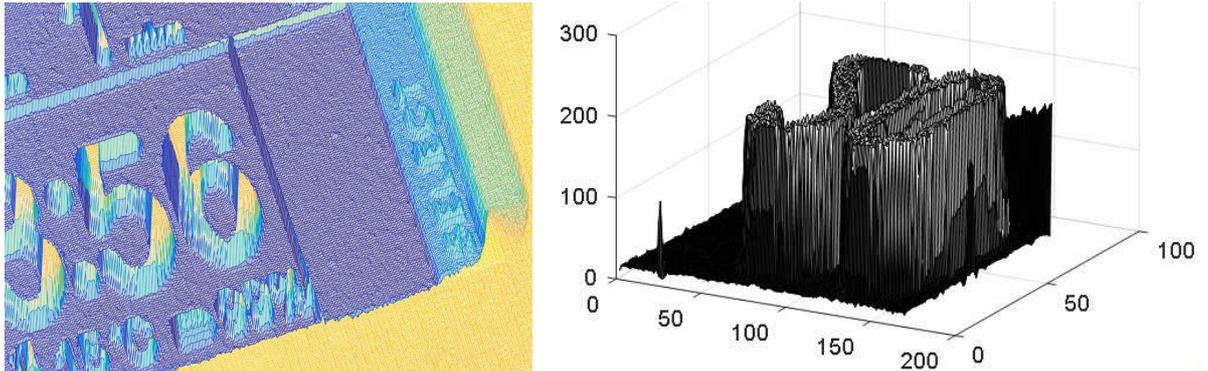


Рис. 21. Примеры 3d изображений. Справа показаны символы "20". По вертикали отображается интенсивность пикселей.

## ВИДЕОКАМЕРА

В проекте использовалась ВЕБ камера высокой четкости HD Logitech C52 [2].

Технические характеристики видеокамеры Logitech C52.

- Видеосвязь в формате HD (1280 x 720 пикселей)
- Видеосъемка в формате HD: до 1280 x 720 пикселей.
- Автофокусировка
- Фотографии: до 8 Мпикс (с программной обработкой)
- Встроенный микрофон с поддержкой технологии Logitech RightSound™

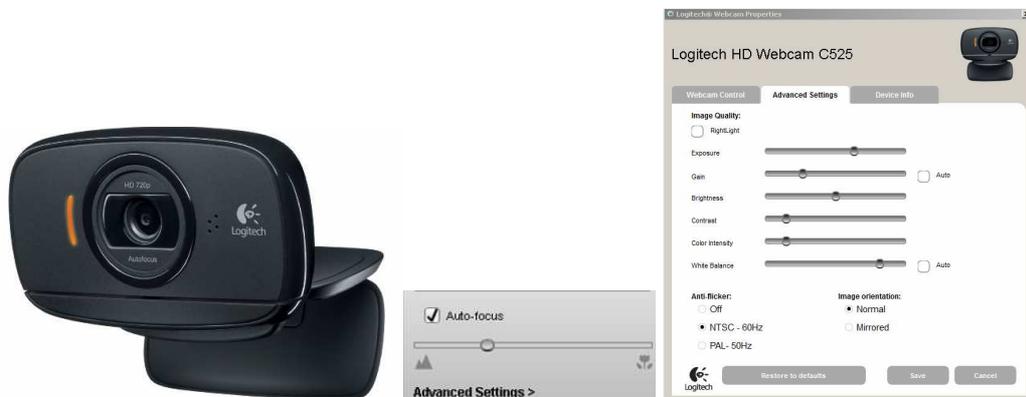


Рис. 22. Внешний вид и параметры настройки видеокамеры Logitech C52.

Основные требования к компьютеру при разрешении камеры 1280 x 720 точек.

- Частота процессора 2.4 ГГц
- Объем ОЗУ 2 ГБ
- Свободное пространство на жестком диске 200 МБ
- USB порт 2.0
- Операционная среда: Windows XP (SP2 or higher), Vista или Windows 7 (32-бит или 64-бит)

## Подключение видеокмеры к MATLAB

1. Проверьте входит ли видеокмера в список устройств MATLAB.

>> webcamlist

2. Если камеры нет в списке, подключите её, используя пакет Support Package Installer (см. Рис. 23)

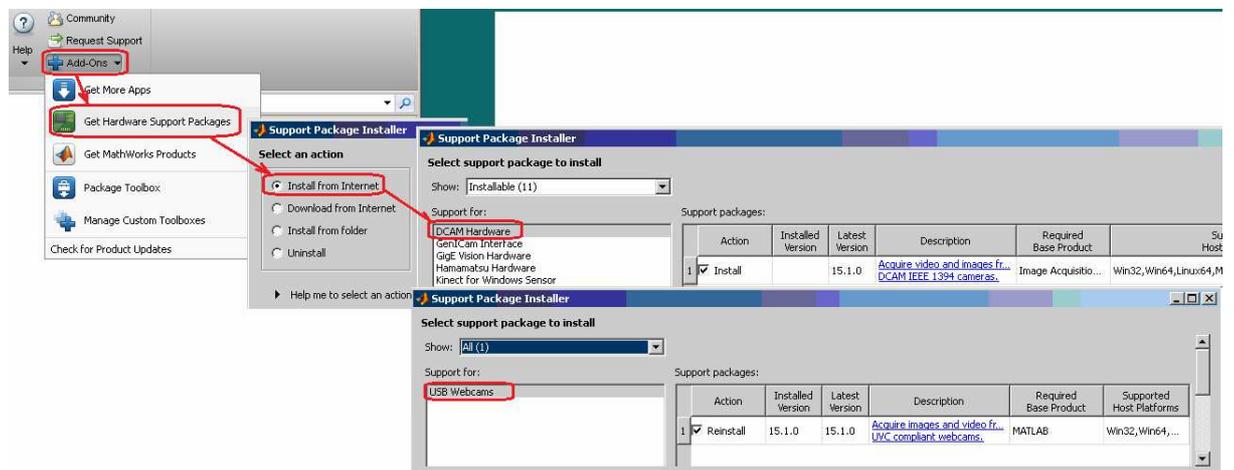


Рис. 23. Подключение видеокмеры к MATLAB.

3. После успешной установки камера попадает в список

>> webcamlist

## Подключение видеокмеры к Simulink

Видеокмеры National Instruments могут подключаться к среде Simulink MATLAB через объекты библиотеки Image Acquisition Toolbox > From Video Device. Для подключения необходимо установить пакет NI-DAQmx используя Support Package Installer (Рис. 23).

Запуск Image Acquisition Tool можно выполнить из командной строки MATLAB:

>>imaqtool

## Порядок ввода видео в MATLAB

Ввод кадров в MATLAB выполняется в следующем порядке.

1. Поиск подключенных к компьютеру видеокмер распознаваемых MATLAB.

>> webcamlist

2. Создание объекта видеокamеры.

>> cam = webcam('Logitech') или

3. Просмотр RGB видео с выводом имени камеры, времени видео, разрешения и частоты кадров.

>> preview(cam)

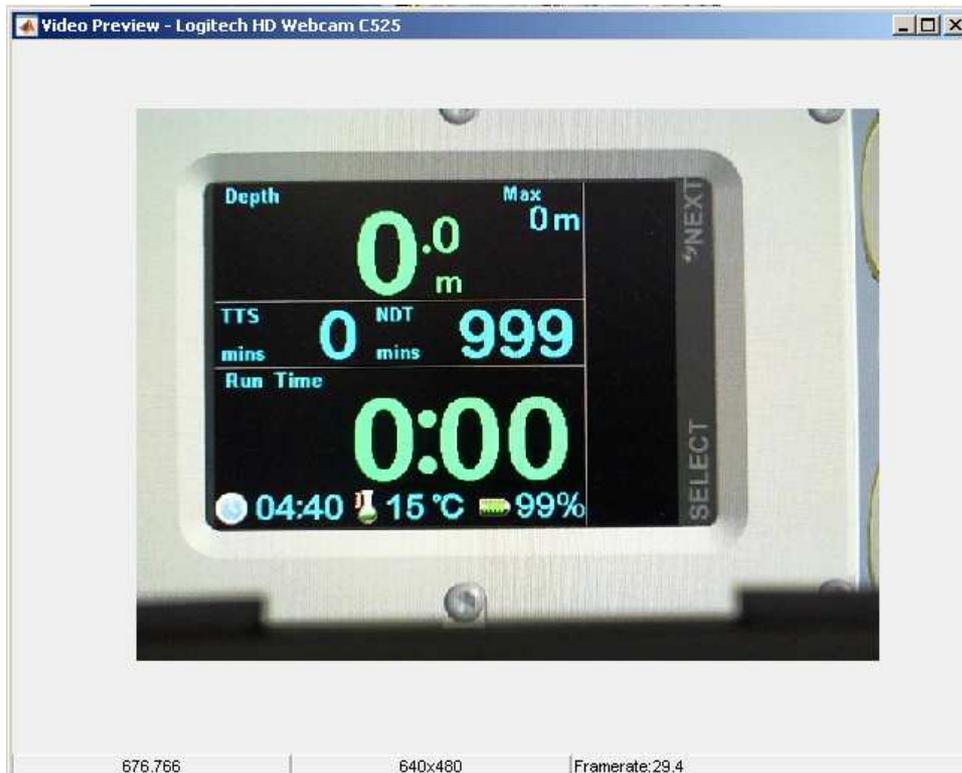


Рис. 24. Вывод видео в MATLAB.

4. Установка требуемых параметров камеры.

cam = webcam (1)

Примеры структурного списка параметров камеры:

<pre>cam = webcam with properties:     Name: 'Logitech HD Webcam C525'     Resolution: '640x480'     AvailableResolutions: {1x22 cell}     Brightness: 128     Contrast: 32     Focus: 60     Gain: 64</pre>	<pre>cam = webcam with properties:     Name: 'Logitech HD Webcam C525'     Resolution: '960x720'     AvailableResolutions: {1x22 cell}     Brightness: 128     WhiteBalanceMode: 'manual'     Pan: 0     Exposure: -5     Tilt: 0     FocusMode: 'manual'     BacklightCompensation: 1</pre>
--	--

<pre> WhiteBalance: 5500 Tilt: 0 Exposure: -4 Pan: 0 Sharpness: 22 Saturation: 32 BacklightCompensation: 1 Zoom: 1 ExposureMode: 'auto' </pre>	<pre> WhiteBalance: 5900 ExposureMode: 'manual'  Sharpness: 30 Saturation: 32 Focus: 90 % 80, 85, 90, 95 adjust Contrast: 32 Zoom: 1 Gain: 64 </pre>
--	--

Получение возможных значений параметров, например, разрешения камеры

```
>> cam.AvailableResolutions
```

```
ans =
'640x480' '160x120' '176x144'
'320x176' '320x240' '432x240'
'352x288' '544x288' '640x360'
'752x416' '800x448' '864x480'
'960x544' '1024x576' '800x600'
'1184x656' '960x720' '1280x720'
'1392x768' '1504x832' '1600x896'
'1280x960'
```

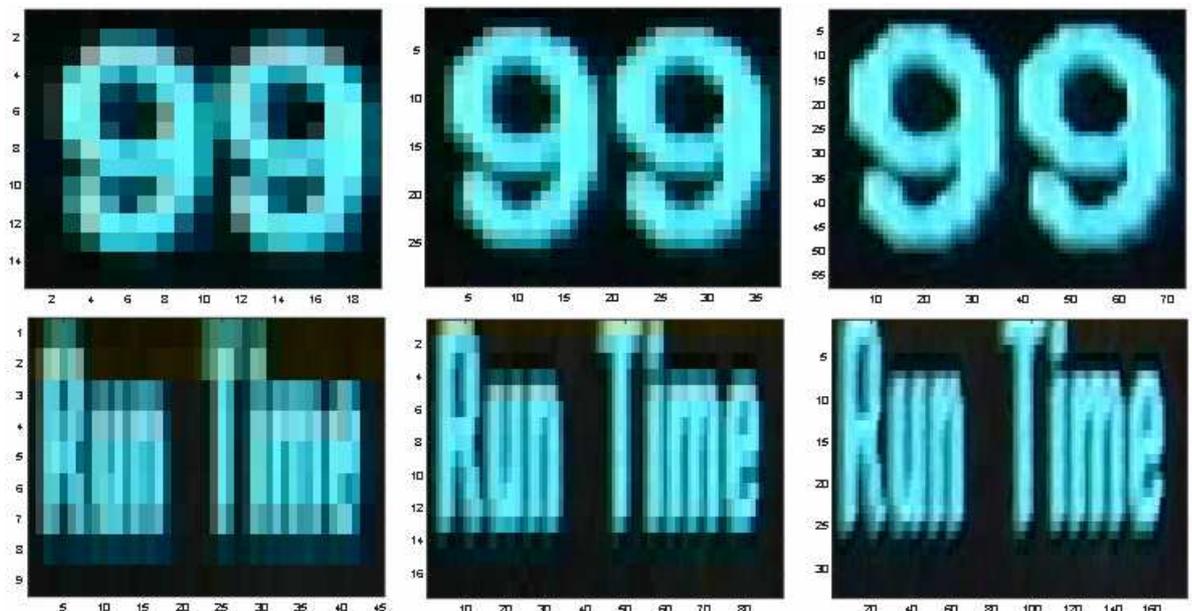


Рис. 25. Изображения при разрешении '320x240' '640x480' и '1280x960' пикселей.

Установка требуемого значения параметра

```
>> cam.Resolution = '320x240';
```

5. Выход из режима просмотра видео.

```
>> closePreview(cam)
```

6. Ввод кадра.  
`>> img = snapshot(cam);`
7. Отображение кадра.  
`>> imshow(img)`
8. Удаление объекта камеры.  
`>> clear('cam');`

### Настройка положения камеры относительно объекта

К выводимому в MATLAB видео добавляется рамка с известными координатами (Рис. 26). При настройке положения камеры относительно объекта границы объекта должны совпадать с положением рамки.



Рис. 26. Рамка в видеокадре.

Пример m-кода для отображения видео с базовой рамкой:

```
webcamlist;
cam = webcam(1);
cam.Resolution = '320x240';

target_area = [45, 44, 217, 150];
while(1)
    pct = snapshot(cam);
    % display target
    Imf = insertShape(pct, 'Rectangle', target_area, 'Color', 'green');
    image(Imf);
end
```

## Особенности состояний объекта

В процессе работы устройство меняет формат выводимой на экран информации. Система распознаёт смену формата по наличию/отсутствию полей TTS и Step. Для поля каждого формата построено свое распределение зон определения символов Рис. 27.



Рис. 27. Разные форматы информации выводимой тестируемым устройством.

## РАСПОЗНАВАНИЕ ТЕКСТА

### OCR функция MATLAB

Функция Optical Character Recognition (OCR) MATLAB распознает текст графического файла.

```
% Загрузка изображения
```

```
I = imread('DC_OS.jpg');
```

```
% Распознавание текста
```

```
results = ocr(I);
```

```
% Анализируемые зоны изображения
```

```
wordBBox = results.WordBoundingBoxes
```

```
% Вывод изображения и зон
```

```
figure;
```

```
Iname = insertShape(I, 'Rectangle', wordBBox, 'Color', 'red');
```

```
imshow(Iname);
```



**Рис. 28.** Зоны поиска символов (23 прямоугольные зоны выделены красным), координаты и размеры зон в results.WordBoundingBoxes.

Таблица 1. Результат распознавания текста.

Номер зоны	Существующие символы зоны	Обнаруженные символы results.Words	Процент распознавания
1	Max	Max	100
2	40.0 m	40°	
3	40 m A Air	:	0
4	TTS	US	0
5	(mins)	Lmmx	
6	Stop	Sum	
7	(m)	(m)	100
8	For	An	
9	(mins)	(laws!	
10	7	7	100
11	15	15	100
12	1	1	100
13	Run	Hum	
14	Time	Mme	
15	12: 10 CNS 2	122102??	
16	10:24	01024	
17	15C	!15°o	
18	99%	--99%	
19	Depth TTS (	-	0
20	-	-	
21	-	-	
22	-	-	
23	CNS 2%	-	0

## Пример автоматического нахождения текстовых зон и распознавание текста в MATLAB

Пример автоматического распознавания текста в MATLAB показан в разделе Help MATLAB [3].



Рис. 29. Исходное изображение [3].

## Нахождение фрагмента изображения по корреляционной функции

Вычисление нормированной взаимной корреляции  $\text{norm\_Corr\_f}$  позволяет отслеживать наличие заданных фрагментов и их координаты в кадрах видеоизображения. Это можно использовать, например, при поиске или подсчете объектов или настройке положения камеры.

Примеры реализации алгоритма обнаружения показаны на Рис. 30, Рис. 31, код алгоритма [4].

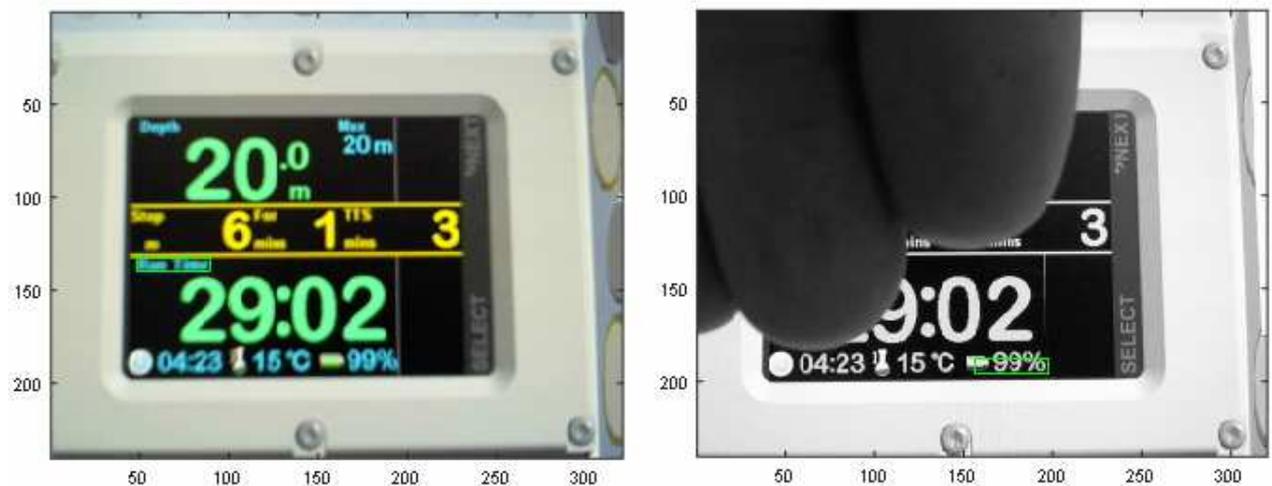
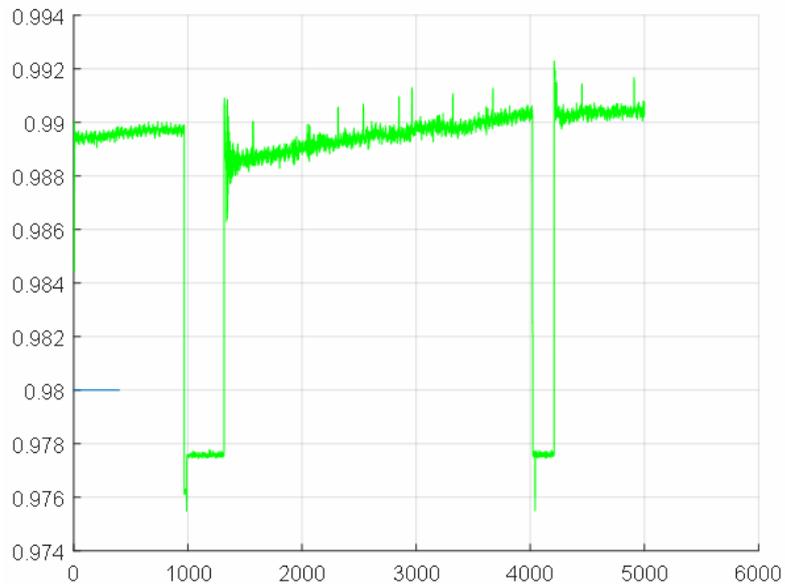


Рис. 30. Зона объекта - текста "Run Time", правильно обнаруженная на левом кадре. При накрытии рукой зоны объекта программа поиска нашла неправильную область объекта (правый кадр).



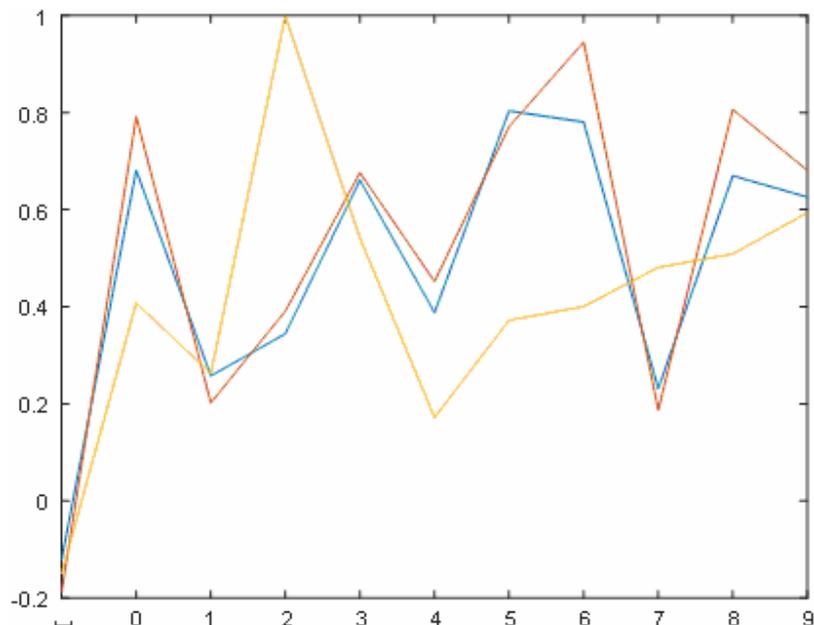
**Рис. 31.** Зависимость корреляционной функции от номера кадра. Значение функции при наличии отыскиваемого объекта  $\sim 0.99$ , а при его отсутствии  $\sim 0.978$ . Малый диапазон изменения функции 0.012 делает алгоритм поиска чувствительным к воздействию помех - нахождению зон без объекта хорошо коррелируемых с эталонным объектом..

#### Улучшение отношения сигнал/шум корреляции увеличением отношения площади символа к площади обнаружения символа

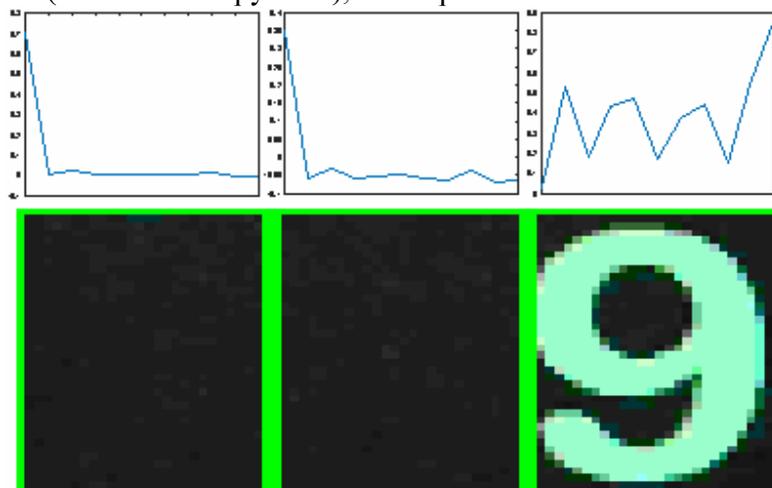
Способность различать символы корреляционным методом зависит от разрешения поля символа, отношения площади символа к площади обнаружения символа и расхождением между определяемым и эталонными символами. Корреляция увеличивается с увеличением разрешения и отношения площадей и уменьшением расхождений между сравниваемыми символами.



**Рис. 32.** Выделение зон расположения символов.



**Рис. 33.** Коэффициент корреляции между цифрами объекта 5 (синяя линия), 6 (коричневая линия) и 2 (желтая линия) в зоне 17x25 пикселей и эталонными символами: пустая зона, символы от 0 до 9. По амплитуде корреляции видно, что двойка хорошо отличается от остальных символов (её легче обнаружить), а пятерка мало отличается от шестерки.



**Рис. 34.** Коэффициенты корреляции между двумя пустыми зонами объекта и зоной с 9 и эталонными символами "пусто"; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9. Графики показывают максимальную корреляцию с "пустым" эталонным символом для первых двух позиций и с эталоном 9 - для последней позиции.

### Устранение неопределенности при сравнении символа с пустой зоной

Корреляционная функция может давать неопределенное значение NaN при сравнении символа объекта с пустой эталонной зоной состоящей из нулей. Для устранения такого эффекта интенсивность одного элемента эталона увеличена до 255:

```

function nmb = recognize_1(Img,target_area,Img_ref,index)
nmb = 0;
for n = index(1):index(end)
    Img_target = imcrop(Img,target_area(n,:));
    Img_target = imadjust(Img_target); % Adjust image intensity values
    Img_target(1,1) = 255; % to remove NaN correlation with zero image

    for i = 1:size(Img_ref,1)
        Img_ref(i,1,1) = 255; % to remove NaN correlation with zero image
        Img_reference = reshape(Img_ref(i,:,:),size(Img_ref,2),size(Img_ref,3));
        r(i) = corr2(logical(Img_reference-110),logical(Img_target-110));
    end
    num = find(r == max(r)) - 2;
    if num < 0
        num = 0;
    end
    nmb = nmb*10 + num;
end

```

### Увеличение корреляции использованием типовых шрифтов

В рассматриваемом варианте 320 x 240 пикселей используются шрифты цифр следующих размеров 18; 26; 36 и 60. Подобрать и доработать шрифт эталона к шрифту объекта можно при помощи специальных программ, например, GLCDFontCreator.exe [5].

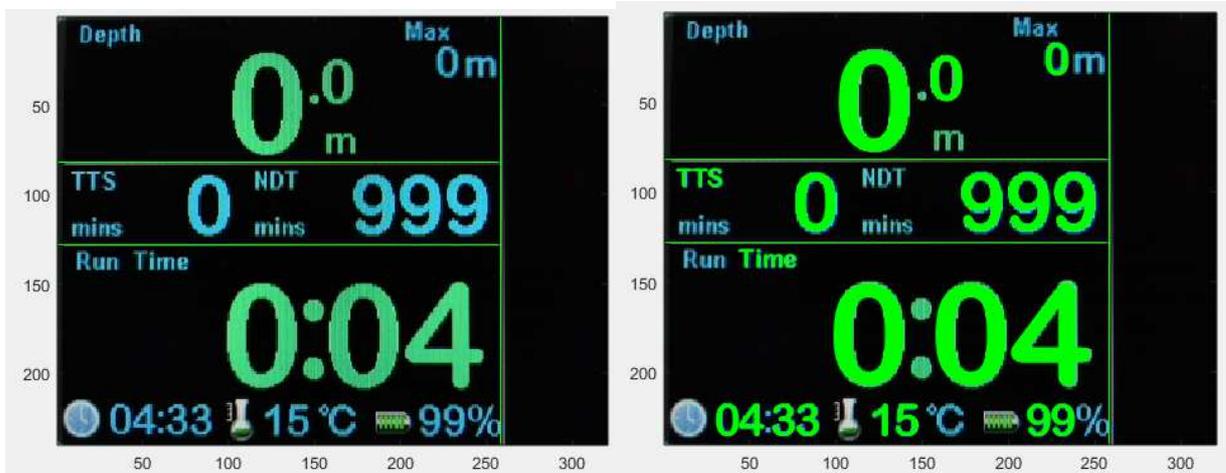


Рис. 35. Оригинальное изображение (слева) и результат наложения 'Arial Rounded MT Bold' шрифтов на оригинальное изображение (справа).

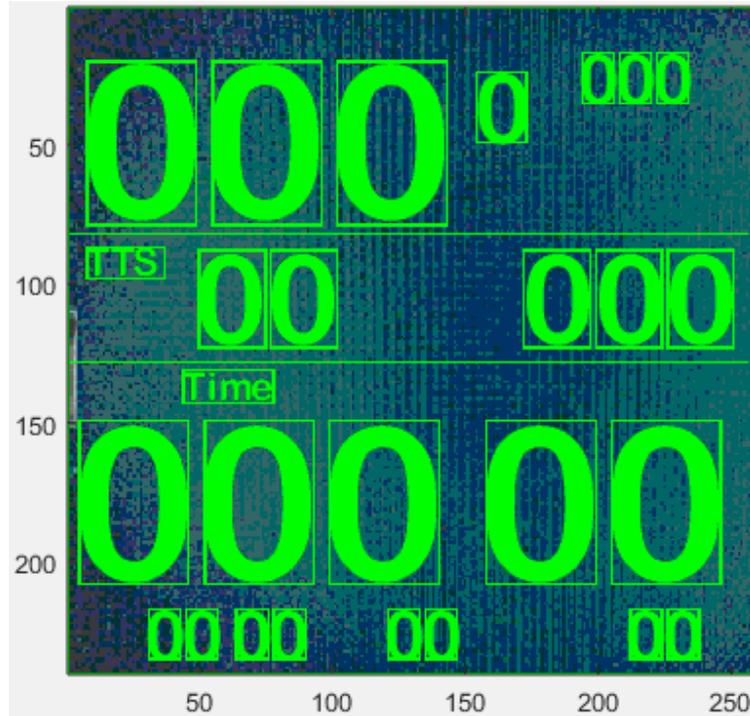


Рис. 36. Вариант сужения зон поиска символов.

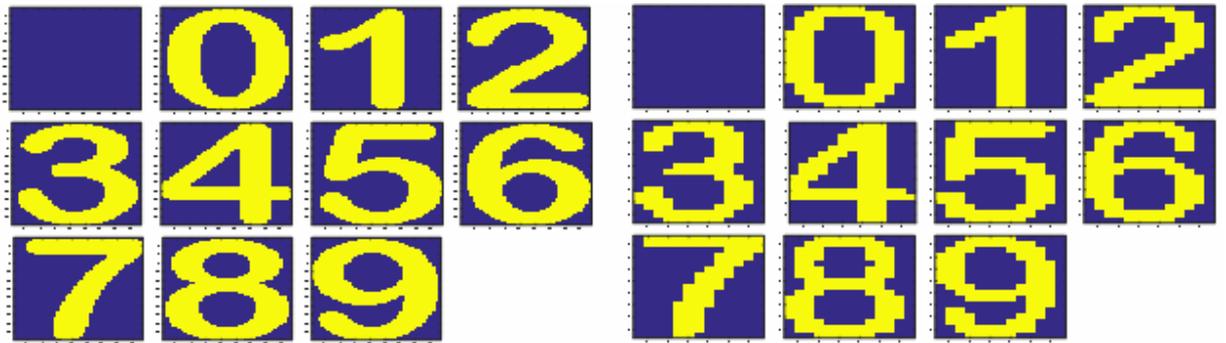


Рис. 37. Варианты эталонных цифр высокого и низкого разрешения. Корреляция (совпадение) между цифрами 5 и 6 выше чем между остальными символами.

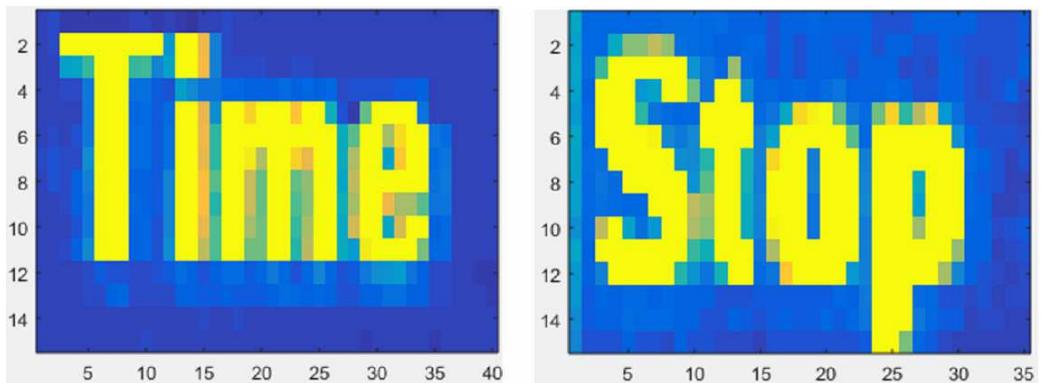


Рис. 38. Пиксельный вид надписей Time и Stop

### Пример интерфейса GUI действующей системы распознавания символов

Система распознавания состояния объекта дополнена Графическим Интерфейсом Пользователя (GUI). Интерфейс пользователя разработан в MATLAB. Интерфейс имеет следующие особенности.

- управляет настройкой и запуском видео
- выводит видео в реальном времени
- показывает качество установки видеокамеры относительно объекта
- подстраивает положение зон распознавания
- показывает результат распознавания символов
- восстанавливает по символам величины параметров
- строит графики изменения параметров объекта
- накапливает и сохраняет результаты
- принимает воздействие пользователя на объект через кнопки NEXT и SELECT

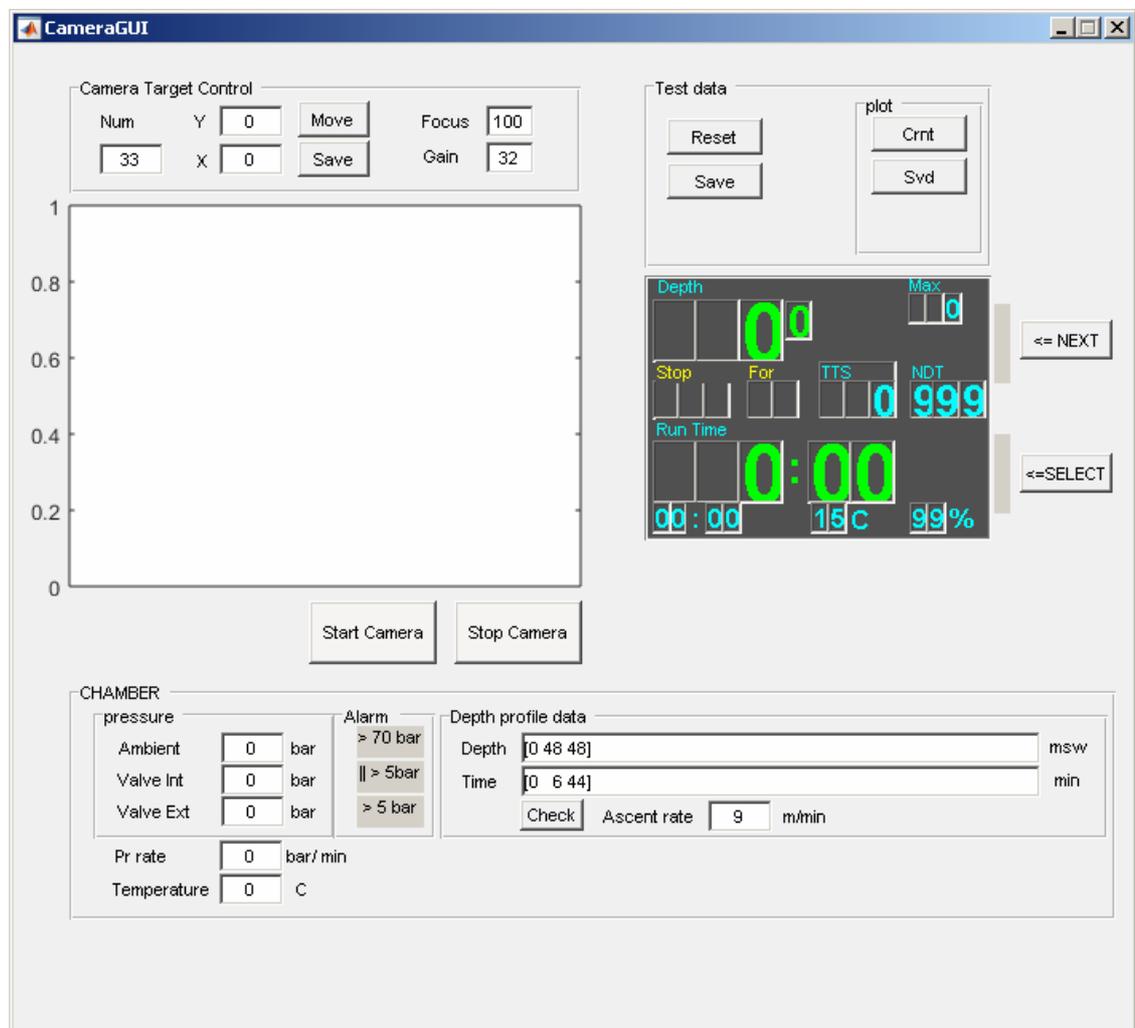


Рис. 39. Начальное состояние Графического Интерфейса Пользователя (GUI) спроектированного в MATLAB.



**Рис. 40.** Фрагмент GUI. Видеоизображение объекта частотой 30 Гц выводится слева. Распознанные в выделенных зеленых зонах символы видеопотока отображаются на шаблоне кадра справа. Точность совмещения камеры и объекта определяется положением рамки самой большой прямоугольной зоны левого кадра.

### Результат Распознавания Символов Объекта



**Рис. 41.** Состояние GUI при начальном формате вывода данных объекта.



Рис. 42. Состояние GUI при измененном формате вывода данных объекта.

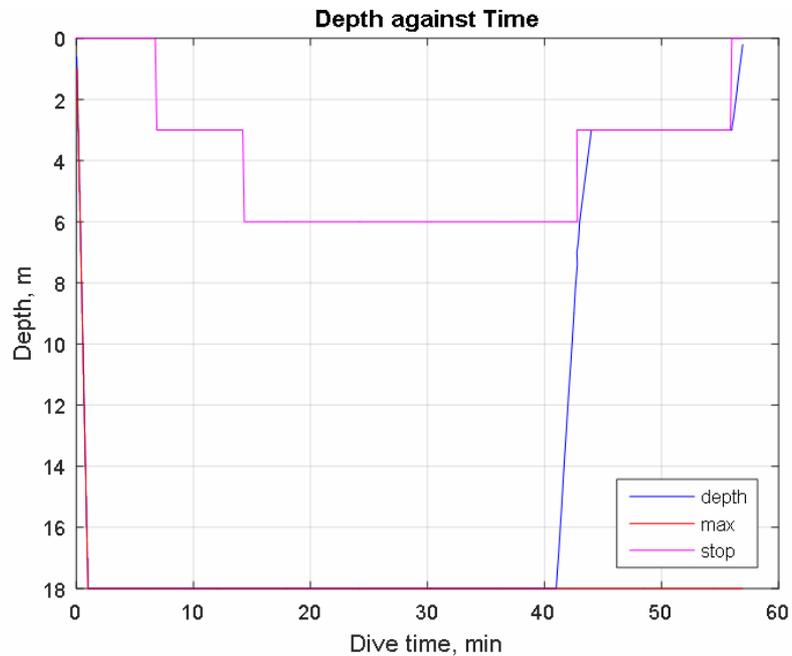


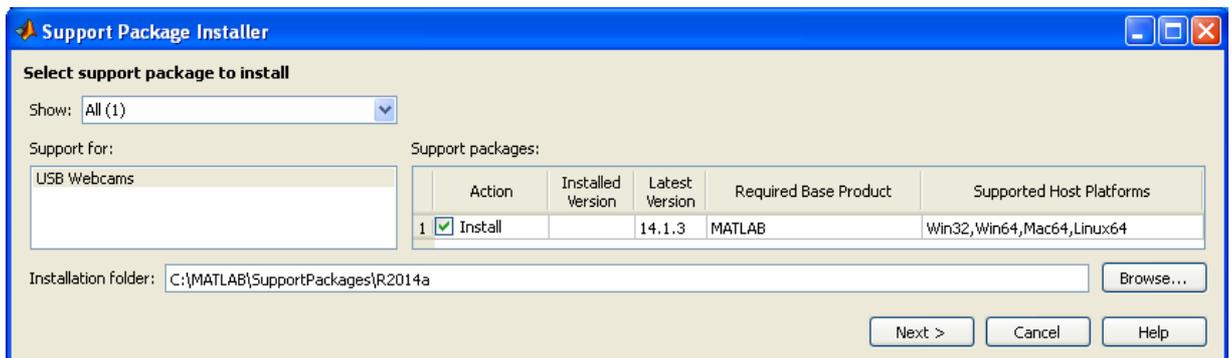
Рис. 43. Графики выводимых на дисплей объекта параметров, считанных бесконтактной системой распознавания символов.

## ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Примечание. Контролируйте выполнение заданий выводом промежуточных изображений.

**Задание 1.** Ввод изображения.

1. Подсоедините видеокамеру к USB порту компьютера.
2. Командой MATLAB `webcamlist` откройте список подключенных вебкамер. Если камера не обнаружена, то подключите её к MATLAB, используя Support Package Installer, как показано в выше.



**Рис. 44.** Интерфейс Support Package Installer.

Примечание: Установка включает регистрацию пользователя на сайте MATLAB. Камера успешно распознаётся в MATLAB под Windows 7 и на команду `webcamlist` дает ответ `>> ans = 'Logitech HD Webcam C525'`

3. Создайте объект и считайте параметры видеокамеры.

```
cam = webcam (1)
ans = webcam with properties:
    Name: 'Logitech HD Webcam C525'
    Resolution: '640x480'
    AvailableResolutions: {1x22 cell}
    Zoom: 1
    WhiteBalanceMode: 'manual'
    Contrast: 32
    BacklightCompensation: 1
    WhiteBalance: 4970
    Saturation: 32
    Tilt: 0
    Exposure: -5
    FocusMode: 'manual'
    Brightness: 128
    Gain: 17
    ExposureMode: 'auto'
    Sharpness: 30
    Focus: 60
    Pan: 0
```

4. Установите минимальное разрешение камеры.

cam.AvailableResolutions

ans =

```
'640x480' '160x120' '176x144'  
'320x176' '320x240' '432x240'  
'352x288' '544x288' '640x360'  
'752x416' '800x448' '864x480'  
'960x544' '1024x576' '800x600'  
'1184x656' '960x720' '1280x720'  
'1392x768' '1504x832' '1600x896'  
'1280x960'
```

cam.Resolution = '320x240';

9. Сделайте снимок и распечатайте его в MATLAB

```
img = snapshot(cam);  
imshow(img)
```

5. Сделайте снимки с другими разрешениями.



Рис. 45. Снимки с разным разрешением.

6. Закройте объект вебкамеры.

```
>> clear('cam');
```

**Задание 2.** Выделение объекта.

1. Установите ПО Logitech для вебкамеры C52

[http://support.logitech.com/ru\\_ru/product/webcam-c170](http://support.logitech.com/ru_ru/product/webcam-c170)

2. Закрепите камеру на расстоянии примерно 40 см от экрана и поместите в зону экрана объект, который можно перемещать вручную.

3. Запустите программу Logitech Webcam Software > Quick Capture.

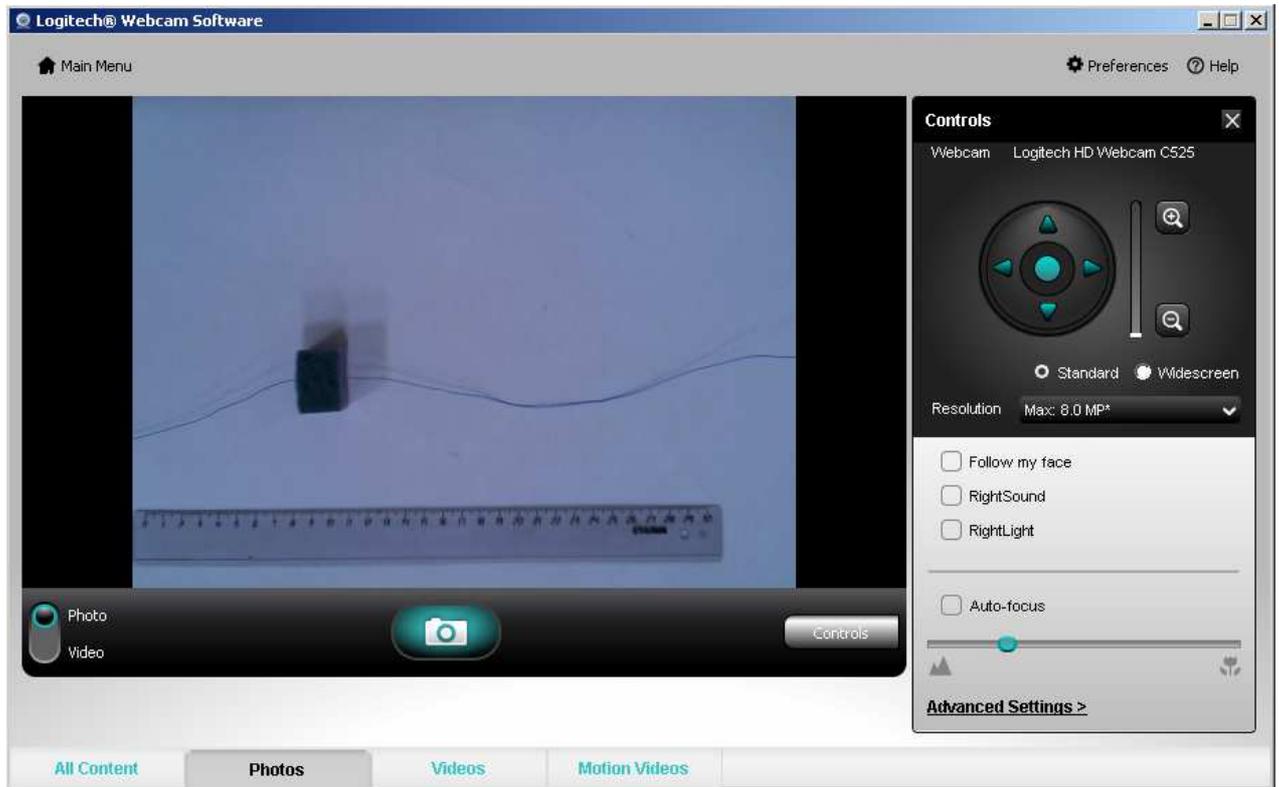


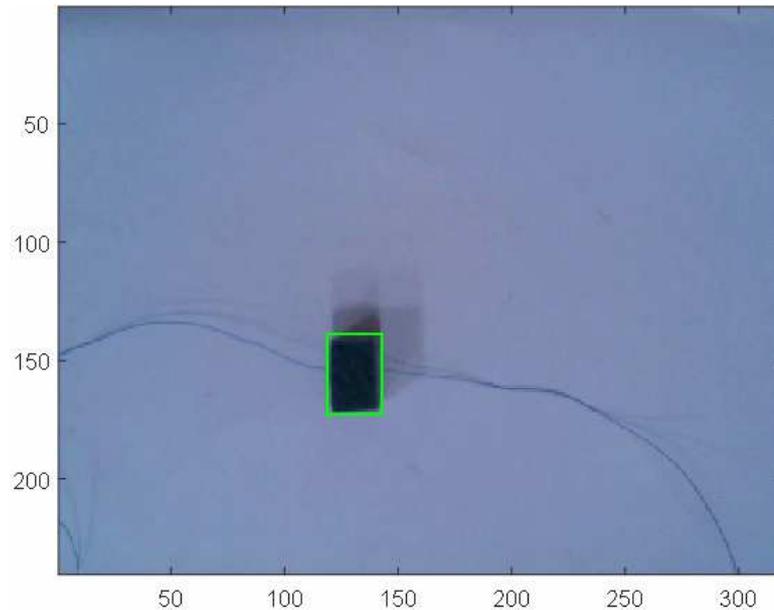
Рис. 46. Интерфейс веб камеры Logitech.

4. Ознакомьтесь с режимами работы камеры (Рис. 46): Advanced Setting, автоматической (Auto-focus) и ручной фокусировкой.
5. Закройте программу Logitech Webcam Software.
6. Оконтурируйте в **MATLAB** объект, находящийся в зоне вебкамеры с разрешением 320x240 пикселей, и сделайте снимок.

```

cam = webcam(1);
cam.Resolution = '320x240';
target_area = [120,140,25,35]; % [X,Y,dX,dY]
pct = snapshot(cam);
Imf = insertShape(pct, 'Rectangle', target_area, 'Color', 'green');
image(Imf);
% clear cam;

```



**Рис. 47.** Оконтуренный объект в растре вебкамеры.

7. Переведите кадр в формат серого цвета. Вырежьте и сохраните оконтуренный фрагмент.

```
% save target
Img=rgb2gray(pct);
target_image = imcrop(Img,target_area); % object
figure, imshow(target_image);
save(' RT_target', 'target_image');
```



**Рис. 48.** Отображение объекта функциями `figure, imshow(target_image);`

8. Определите уровень интенсивности в зоне объекта и экрана.

Матрица `'target_image'` 36x26 uint8:

```
132 127 118 103 91 86 73 63 54 54 51 51 53 51 53 49 52 55 53 54 59 60 63 74 105 113
132 129 116 101 87 68 58 56 52 49 48 48 48 50 50 50 49 53 50 48 54 61 65 76 107 113
131 117 85 68 59 52 51 47 49 48 48 45 47 46 47 49 49 48 43 40 47 59 68 77 104 112
129 102 48 42 36 37 42 41 40 36 37 40 38 39 40 38 39 38 36 35 40 60 65 78 104 110
126 89 50 40 32 30 30 30 30 30 32 35 32 34 32 30 32 28 29 33 36 49 62 80 105 111
120 81 41 26 25 24 19 23 24 25 26 24 24 32 26 31 30 30 30 31 32 45 59 81 105 111
118 80 36 27 27 19 18 20 24 26 24 23 25 26 25 27 26 31 27 29 32 40 59 81 104 112
120 75 32 30 29 23 19 23 24 23 25 24 26 24 22 25 29 29 31 32 31 44 61 82 106 112
124 80 32 30 31 29 25 26 24 22 26 23 26 26 24 24 26 24 26 32 32 45 61 82 102 109
123 74 37 32 32 32 30 24 25 27 27 25 29 25 25 24 23 23 24 31 37 47 64 81 97 102
```

124 73 39 30 28 30 25 24 24 26 28 26 26 24 24 23 24 25 25 26 32 47 64 84 98 103  
123 71 32 25 23 26 26 23 24 21 24 30 29 27 23 22 22 24 25 23 32 45 65 84 101 107  
124 68 32 25 30 25 25 27 27 22 26 32 29 26 25 23 23 23 25 29 41 64 84 101 109  
123 73 33 28 29 26 26 28 28 24 26 28 23 22 29 26 24 23 24 24 31 39 66 83 102 109  
117 69 39 32 28 29 26 25 28 26 24 28 22 22 26 25 28 24 23 28 30 39 64 78 101 105  
108 72 38 32 30 24 26 24 26 22 23 19 24 27 24 20 22 24 29 33 40 57 73 89 95  
120 71 36 31 26 26 26 28 26 24 22 23 23 24 31 26 24 23 24 31 32 43 65 80 100 106  
113 61 40 25 23 26 31 30 32 23 24 23 24 24 29 28 28 19 22 28 32 39 61 78 102 109  
105 65 37 21 21 26 25 27 32 31 26 23 22 25 26 26 29 23 21 20 26 32 60 78 101 110  
113 68 37 24 24 24 21 28 30 31 26 26 23 27 26 29 30 25 19 21 26 39 58 75 99 110  
113 68 37 32 26 24 21 25 26 29 27 23 24 26 24 26 26 28 25 25 29 36 63 74 102 111  
116 74 38 32 28 24 20 23 24 24 26 27 24 23 23 26 31 31 26 24 24 35 60 76 101 112  
122 85 39 36 28 26 24 24 24 24 24 20 23 24 28 28 27 28 29 28 39 61 76 99 109  
127 91 42 35 32 25 24 28 30 25 23 24 25 28 25 26 25 28 31 31 32 40 64 76 102 109  
129 100 42 32 31 27 24 25 28 25 25 28 29 23 22 25 26 30 31 30 32 39 64 78 102 111  
130 98 39 32 32 32 26 26 26 26 31 31 23 21 25 20 25 28 31 32 36 61 77 104 114  
125 102 41 32 33 34 31 28 30 24 24 24 26 24 26 25 23 24 28 31 32 38 59 81 106 117  
131 98 42 34 33 32 27 28 30 26 25 23 26 23 24 23 20 23 27 29 31 32 60 82 113 123  
133 97 39 34 27 29 28 27 28 27 27 24 25 25 25 26 24 25 30 31 31 36 60 85 126 132  
132 103 46 38 32 30 31 28 28 25 26 28 31 29 28 28 26 26 28 33 32 42 57 94 138 141  
136 116 57 40 32 31 31 32 29 25 28 30 31 32 31 28 24 29 30 35 36 40 62 101 142 143  
138 123 65 34 34 34 32 37 34 32 32 33 32 34 35 37 35 37 34 40 37 45 68 121 142 141  
139 129 85 42 33 32 38 40 37 34 40 45 44 48 56 58 62 66 71 79 78 83 104 138 142 141  
140 137 115 97 91 96 99 102 106 105 107 111 113 118 120 126 124 126 129 130 133 131 134 142 144 141  
140 140 138 135 133 133 137 137 138 139 138 138 139 138 141 139 138 138 138 139 139 140 142 141 141  
141 141 140 139 138 138 138 139 141 140 138 139 140 139 139 138 138 139 138 138 139 139 141 140 141 141

Фрагмент кадра 'Img' (верхний левый угол) 36x26 uint8:

105 110 107 106 106 108 107 109 111 111 111 110 109 110 111 110 108 108  
111 109 113 111 110 112 109 111 112 113 114 110 113 112 115 113 110 113  
112 106 113 113 114 113 114 114 114 114 115 114 113 112 113 113 114 114 111 113  
113 113 114 113 114 115 114 115 116 115 113 113 113 116 114 115 114 114  
114 114 114 114 114 114 116 116 117 115 114 114 114 117 117 117 115 115  
115 114 114 115 115 115 116 116 118 117 117 118 117 117 117 113 116 118  
118 121 115 116 113 115 116 116 119 119 119 116 116 117 118 116 118 118  
120 120 118 118 117 117 117 120 120 119 118 120 118 120 121 122 118  
118 117 119 122 122 122 121 121 122 120 122 120 119 120 121 122 121 122  
122 122 122 123 122 123 123 122 123 122 122 123 120 122 123 123 122 123  
122 122 123 123 124 123 125 123 125 125 125 123 122 122 124 124 125 125  
125 127 126 124 124 125 123 125 129 129 129 127 124 123 125 123 128 128  
126 129 128 127 127 129 126 125 129 130 127 126 127 125 129 125 130 129  
130 126 127 129 130 129 127 127 129 129 129 127 125 125 126 126 125 129  
128 128 129 129 130 130 130 130 130 129 130 130 130 130 129 130 129 130  
130 128 130 131 131 130 130 130 130 131 131 132 131 130 130 131 131 130

## 9. Переведите изображения в двоичный формат

```
Img_eroding = 100; % уровень компаратора  
Img_edge_enhancement=imsubtract(Img,Img_eroding);  
target_image_edge_enhancement=imsubtract(target_image,Img_eroding);
```

```
Img_lgc=logical(Img_edge_enhancement);  
target_image_lgc=logical(target_image_edge_enhancement);
```





### Задание 3. Отслеживание координат объекта.

1. Используя результаты предыдущих заданий напишите программу отслеживания положения объекта соответствующим смещением контурной рамки объекта в видеокадре.

```
clear all
```

```
load ('target_image_art');  
% figure, imshow(target_image);  
dX = size(target_image,2);  
dY = size(target_image,1);
```

```
Img_eroding = 100; % average between screen and object intensity
```

```
cam = webcam(1);  
cam.Resolution = '320x240';
```

```
for i = 1:1000
```

```
    pct = snapshot(cam);  
    Img=rgb2gray(pct); % rgb in gray
```

```
    Img_edge_enhancement=imsubtract(Img,Img_eroding); % negative result of subtract is replaced by zero  
    Img_lgc=logical(Img_edge_enhancement); % '0' if pixel = 0; '1' if pixel > 0  
    %figure, imshow(Img_lgc);
```

```
    Img_inv_lgc = not(Img_lgc); % inversion  
    Img_inv_flt_lgc = bwareaopen(Img_inv_lgc,500); % remove 'island' being less than 500 pixels
```

```
    Iprops=regionprops(Img_inv_flt_lgc,'BoundingBox','Image'); % find remained 'islands'
```

```
    % format: Iprops(i).BoundingBox(:) = [X,Y,dX,dY]  
    obj_num = find(arrayfun(@(x)max(x.BoundingBox(3)),Iprops)< 50 & ... % max(dx)  
        arrayfun(@(x)min(x.BoundingBox(3)),Iprops)> 10 & ... % min(dx)  
        arrayfun(@(x)max(x.BoundingBox(4)),Iprops)< 60 & ... % max(dy)  
        arrayfun(@(x)min(x.BoundingBox(4)),Iprops)> 20); % min(dy)
```

```
    % display target
```

```
    if ~isempty(obj_num)
```

```
        X = Iprops(obj_num(1)).BoundingBox(1);
```

```
        Y = Iprops(obj_num(1)).BoundingBox(2)+11;
```

```
        if X+dX < 320 && Y+dY < 240
```

```
            % object inside the screen
```

```
            target_area = [X,Y,dX-1,dY-1];
```

```
            Imf = insertShape(pct, 'Rectangle', target_area, 'Color', 'green');
```

```
            image(Imf);
```

```
        else
```

```
            % object in screen boarder
```

```
            image(pct);
```

```
        end
```

```
    else
```

```
        % no object found
```

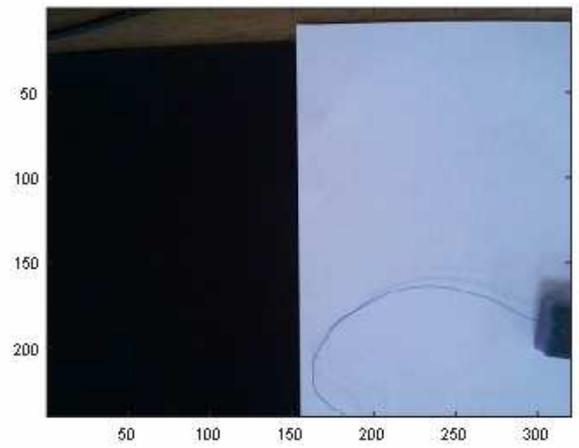
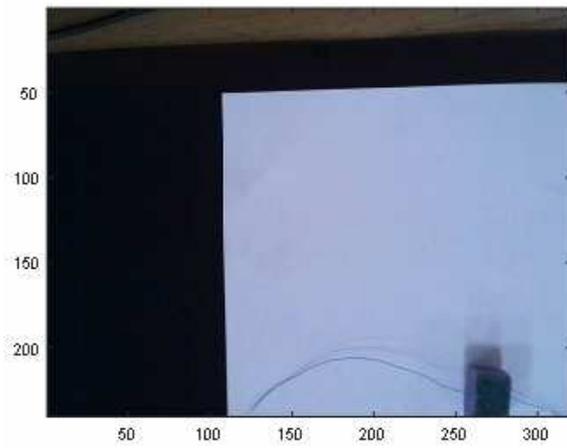
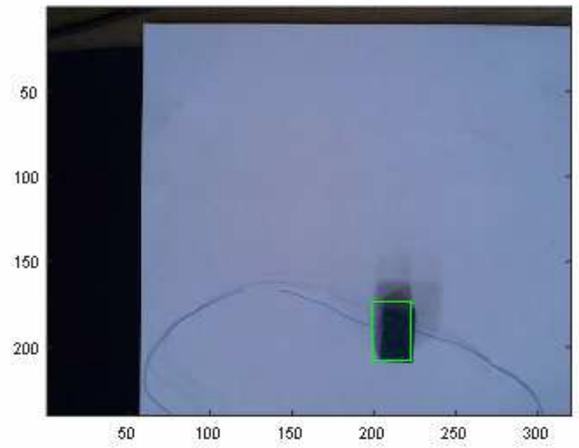
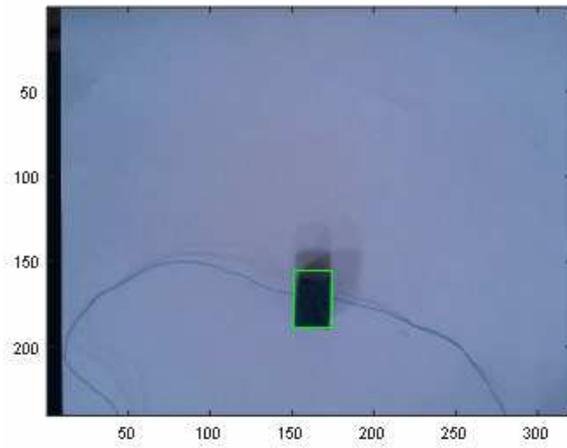
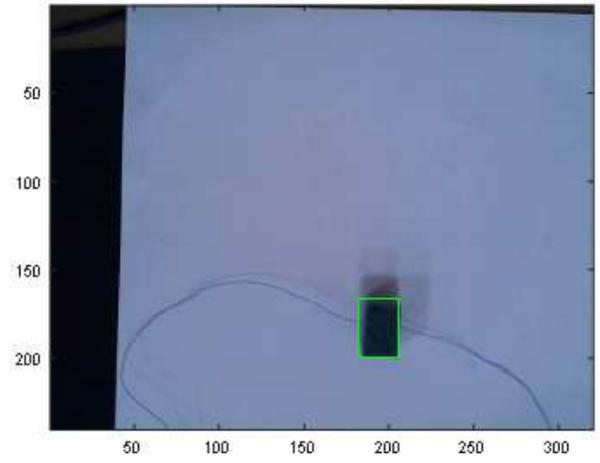
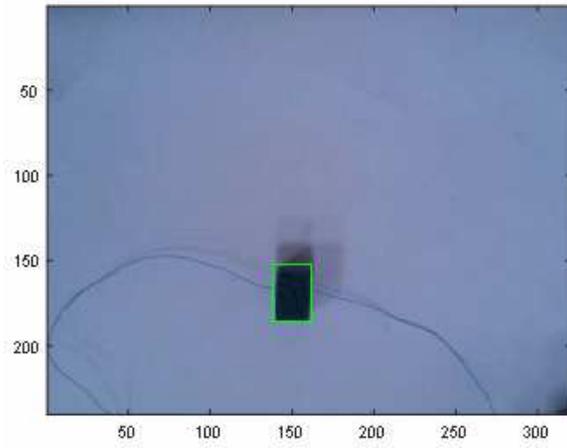
```
        image(pct);
```

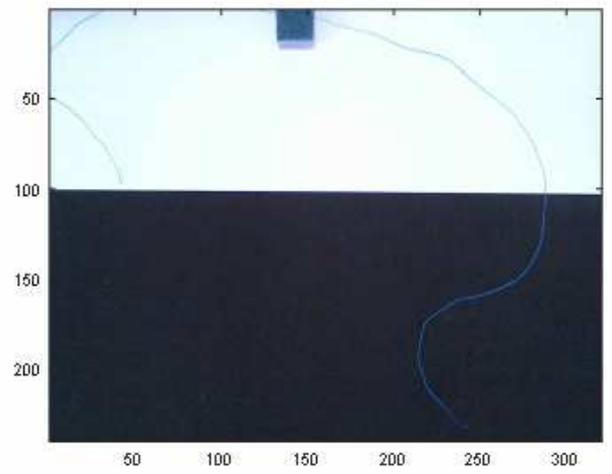
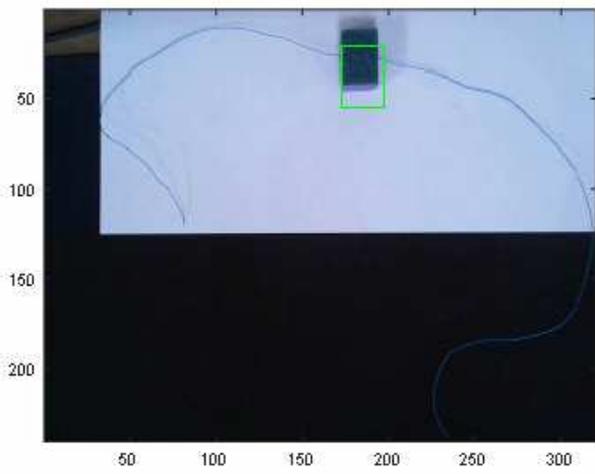
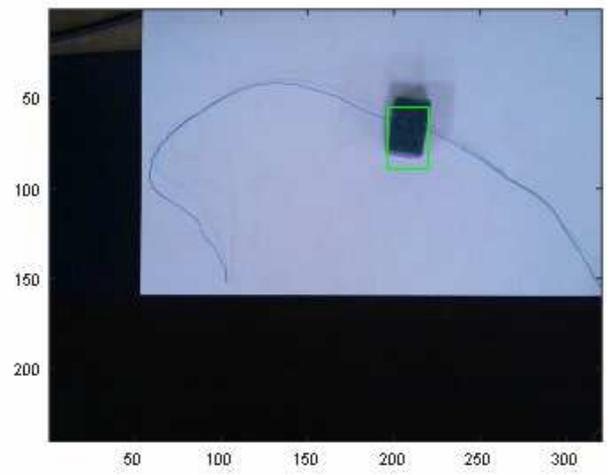
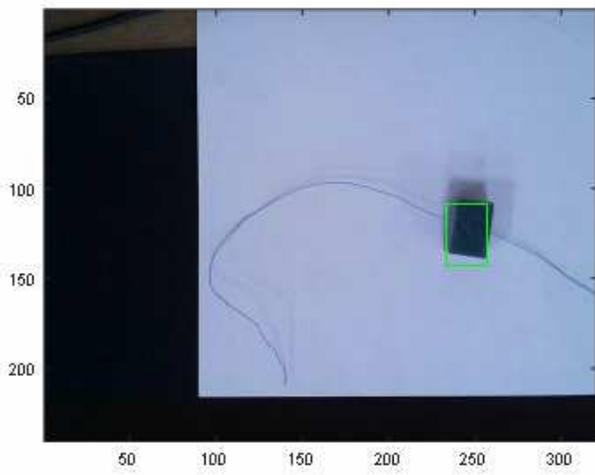
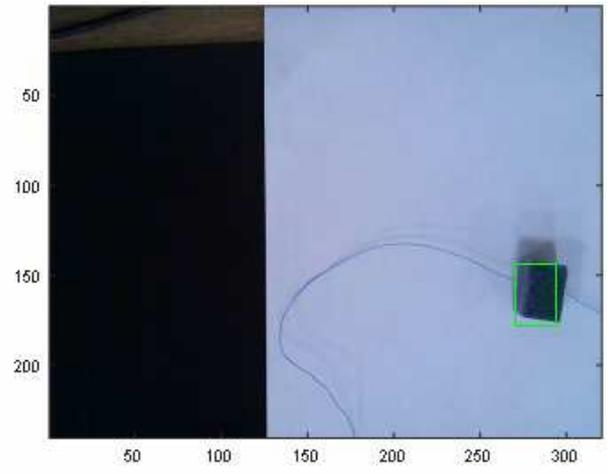
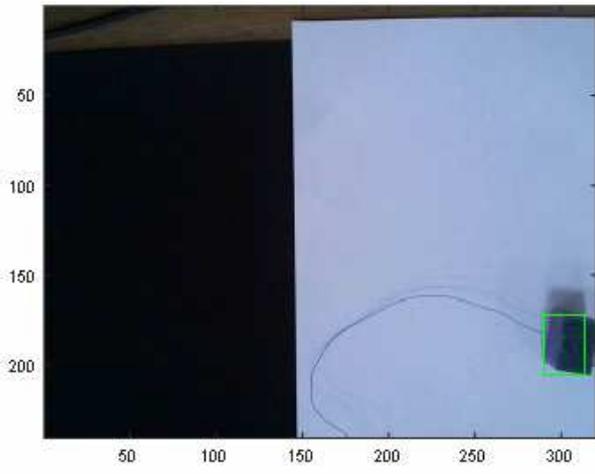
```
    end
```

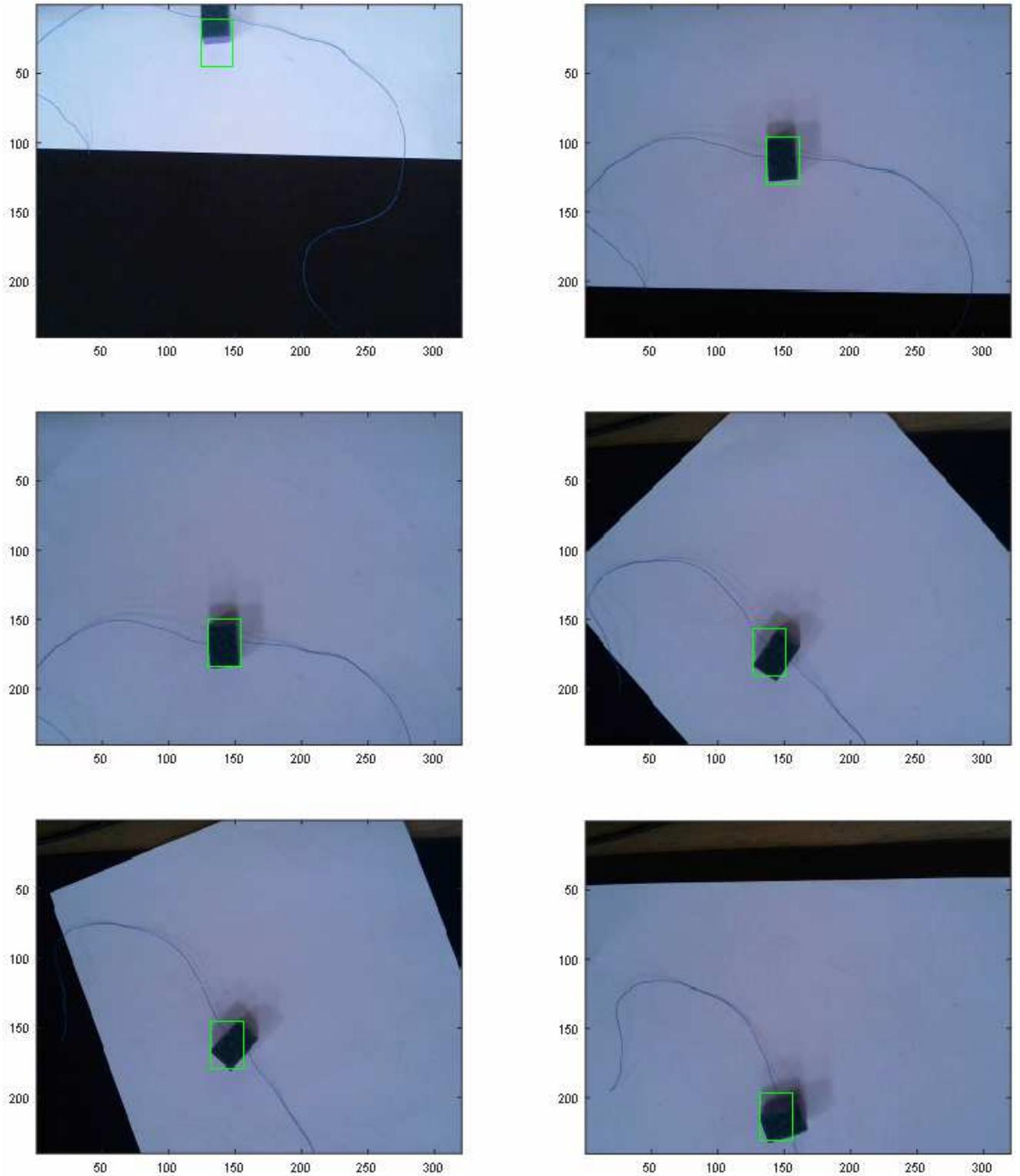
```
end
```

```
clear cam;
```

13. Проверьте работу программы.







**Рис. 51.** Результаты отслеживания координат объекта. Система обладает высокой динамикой. Система распознавания нечувствительна к появлению объектов других форм и размеров.

**Задание 4.** Повышение точности определения координат объекта.

1. Используя корреляционные функции (см. Раздел Общие сведения) разработайте программу поиска объекта в зонах координаты которых находит программа предыдущего задания.
2. Проверьте работу программы.
3. Сравните насколько изменились точность и быстродействие программы.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Назовите области применения систем распознавания.
2. Что является шумом для систем распознавания?
3. Назовите источники ошибок распознавания объектов.
4. Как влияет изменение режимов съёмки на результат распознавания?

### **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. BIO-350 Dive Computer  
<https://www.opensafety.eu/datasheets/Bio350%20Short%20Datasheet.pdf>
2. Портативная веб-камера HD Webcam C525 <http://www.logitech.com/ru-ru/product/hd-webcam-c525>.
3. Help MATLAB > Computer Vision System Toolbox > Examples > Automatically Detect and Recognize Text in Natural Images.
4. Help MATLAB > Object Detection and Recognition > Examples and How To > Pattern Matching.
5. Dr. Bob Davidov. Построение интерфейса пользователя локальной системы управления на базе контроллера Arduino UNO. [http://portalnp.ru/wp-content/uploads/2015/01/15.03\\_Arduino-UNO-Local-Control-User-Interface\\_Ed2a.pdf](http://portalnp.ru/wp-content/uploads/2015/01/15.03_Arduino-UNO-Local-Control-User-Interface_Ed2a.pdf)
6. Dr. Bob Davidov. Компьютерные технологии управления в технических системах <http://portalnp.ru/author/bobdavidov>.