Practice Report

Class: Image Processing

Teacher: Zhu Weina

Practice: Analyze the motion of the target object in the video

School： School of Information, Yunnan University

Major： Computer Science and Technology.

Name： MD AHAD ALI

student ID： 20183290242

Submission time： 2021/06/21

Score: 100

Abstract: *Moving object detection is an important step in the video processing. We propose an implementation of image subtraction algorithm to detect a moving object. Detection of object motion has been implemented and the velocity is been calculated. The algorithm is processed with Matlab software and we calculated the distance, frame per time, velocity. This can be used in speed measurements In underwater monitoring systems.*

Keyword: object, detection, analysis, velocity, acceleration, position.

Purpose: Analyze the motion of the target object in the video.

Introduction: As a new technology developed on the basis of image and video processing, moving object tracking and recognizing is one of the important research subjects in application fields such as computer vision, robotics and video information processing. It has found extensive applications in video information processing. It has found extensive applications in video compression, target recognition, intelligent monitoring, video retrieval human computer interaction and bio medicine, etc. The procedure of moving object tracking is to decide whether there exist objects moving in video and to position the target basically and recognize it. the accuracy rate of object moving in video and to position the target basically. The accuracy rate of object detection exerts great influence on the tracking and recognition in next steps. Matlab in image and video processing will greatly improve the efficiency of moving object tracking. This paper presents the study on the implementation of Matlab based moving object detecting algorithm, in which frame difference algorithm was chosen as an example for detect moving object in video from the aspects of video acquisition, video type conversion, video gray-level conversion, video filtering, feature extraction and frame segmentation, etc.

Parameter Description: Matlab parameter, generally, is any characteristic that can help in defining or classifying a particular system. That is, a parameter is an element of a system that is useful, or critical, when identifying the system, or when evaluating its performance, status, condition, etc. There are bunch of parameter I used for video processing. Such as:

vision.ForegroundDetector: The ForegroundDetector compares a color or grayscale video frame to a background model to determine whether individual pixels are part of the background or the foreground. It then computes a foreground mask. By using background subtraction, you can detect foreground objects in an image taken from a stationary camera.

vision.VideoFileReader: It is for Read video frames and audio samples from video file. The VideoFileReader object reads video frames, images, and audio samples from a video file. The object can also read image files.

vision.BlobAnalysis: It is for to compute statistics for connected regions in a binary image. We can track a set of points by create the vision.BlobAnalysis object and set its properties and Call the object with arguments, as if it were a function.

Implay: Use the implay function to open the Video Viewer app, which plays MATLAB movies, videos, or image sequences (also called image stacks). Using Video Viewer you

can select the movie or image sequence that you want to play, jump to a specific frame in the sequence, change the frame rate of the display, or perform other viewing activities. You can open multiple Video Viewers to view different movies simultaneously.

Sprint: Format data into string or character vector.

Imabsdiff: Absolute difference of two images.

Regionprops: Measure properties of image regions.

Bwareaopen: Remove small objects from binary image.

Velocity: The velocity of an object is the rate of change of its position with respect to a frame of reference, and is a function of time. Velocity is equivalent to a specification of an object's speed and direction of motion.

Acceleration: Acceleration is a change in the rate of motion, speed or action. Usually, acceleration means the speed is changing, but not always. When an object moves in a circular path at a constant speed, it is still accelerating, because the direction of its velocity is changing.

Description of program interface: Here we can see two primary file such as 'main' and studyOfMovement'. There are also bunch of file I am going to describe:

Main: There are two interface here First of Foreground Detection that I start create foreground detector object by vision.foreground detector and set its properties.

```
d= vision.ForegroundDetector(...
    'NumTrainingFrames', 20,...
    'InitialVariance', 50*50);
```

Then Read in video file.

```
r= vision.VideoFileReader('sample.mp4', ...
    'VideoOutputDataType', 'uint8');
```

Then Create a object for blob analysis and setup video player.

```
blob=vision.BlobAnalysis('MinimumBlobArea', 1100);
player= vision.DeployableVideoPlayer;
```

Now I need to create loop to run through video for foreground detection:

```
while~isDone(r)
    frame=step(r);
    fgMask= step(d, frame);
    [~,~, bbox]= step(blob, fgMask);
    J=insertShape(frame, 'rectangle', bbox);
    step(player, J);

    pause(0.01);
```

```
end
```

And others one Study of movement like Get path, velocity and acceleration that Here this code we can analysis data To use study of movement function which will training the object.

studyOfMovement: This file gets the position, velocity and acceleration of a fish from a videofile recorded with a fixed camera. The detection of this is based in frame subtraction (a common background is subtract from each frame and then treated.). So just detect moving object of different color from background.

detectfish: I used balls algorithm to detect and identifies the moving object.

exportVideo: write a video file from a structure containing all RGB frames in the 'cdata' field.

getAcc: computes the velocity of the object knowing its position in 2 frames.

getBackground: returns a gray background based in frame differences. Substracts frame1 and frame2 and fills the empty space with otherFrame.

getCG: returns the position the gravity center of the object. we use the function REGIONPROPS and the value of 'Centroids'.

getDiameter: returns the value of the diameter of the object in the frame. we use the function REGIONPROPS and the value of 'MajorAxisLength'.

getProf: returns the value of distance from camera to object. we use the relation that exists between two similars triangles, one is the triangle with base the real diameter of object and height the distance; and the other is the triangle with base the distance in pixels of object and height the focus distance which is known in each camera. We need to know in a specific distance the diameter in pixels to calculate the px2m coeficient alpha.

getpx2m: returns the coeficient of meters/pixel. we use divide the diameter in meters of the real object by the pixels of the maximum diameter in the image(we use maximimun because sometimes it doesn't detect que total ball, so we unse the maximum)

getStructure: returns structure and number of frames.

Getvel: Computes the velocity of the object knowing its position in 2 frames.

Graphics: Makes the graphics of CG, vel and acc in 2D and 3D.

substractFrames: subtract frames(giving the absolute value) and remove noise of the input frames.

makeReport: writes a text file with the velocity and acceleration. Each column is: u v w ax ay az.

Process: now we've seen what the program does. Now let's see how does it work. First of all we have to read the video and then we have to use our main function studyOfMovement.

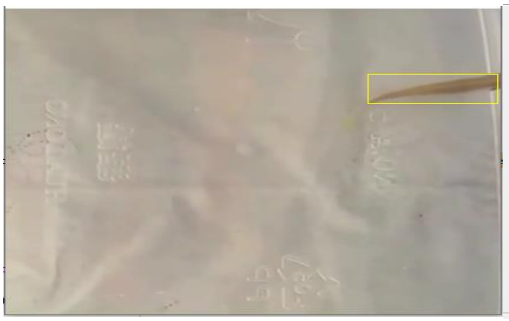
```
[ mov3, CG3, vel3, acc3, FPS3, replay3, detection3, difference3 ] =
studyOfMovement( video3, 1,
floor(video3.NumberOfFrames/2),video3.NumberOfFrames, 20, 10, 26,
500, 25, 1, 0.065, dfocus, 1, 1, 100, 20 );
```

We have to know that this program is based on frame of structure. Now if I go studyOfMovement file. We can see some parameters such as Frame per Second and time frame. First things that we have to do is get a structure called 'mov' that in we have inside RGB and grayscale matrixes. So we will deal grayscale because of our accurate movement information. Since the program is based on frame so we need a common background. There have function that will take frame and this way we can get the fish only that can move. Function takes each frame that subtract the background that will have a region change that is that fish and noise that we want to remove. We will remove it and we will be sure that we can take fish movement only. We have to follow some criteria and we have to take the more changing area and it has to be bigger of parameter that minimum of pixels. So this function will return finally which is logical matrix. Now we can 'getCG' using the function of toolbox image called ratio drops to help get position. Then we get the diameter that lead all the frames. To get velocity we will use CG, px2m (pixel to meter convert), timeframe, nFrame and flag. We will also get acceleration extract velocity, timeframe, nFrame and flag.

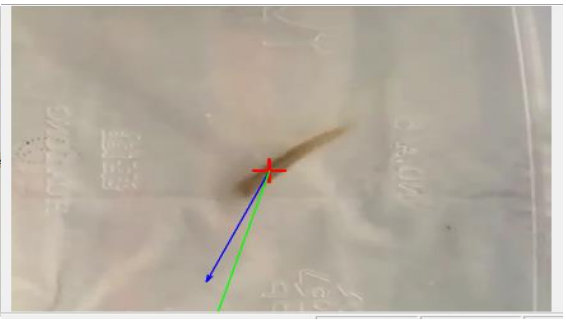
Experiment result and data analysis interface:

As we can see in our program We can detect Movement Object. We also can Detect Velocity, Position and Acceleration. First of all detect the object (A). I am using image thresholding that threshold is converted everything to white and black, based on the threshold value. So first of all I need to extract video as images. The object in video which extracted images are now training with grayscale mode (C). Then I used image mask in (D) that black is background and white is object. After trained the object we can get final video for analysis data (B). Here we can see in (B) the red one is Position, Blue one is velocity and green one is Acceleration.

After analysis data I included in graph. Graph (E) is for velocity. Graph (F) for position. Graph (G) for Acceleration and I include all together in Graph (H).



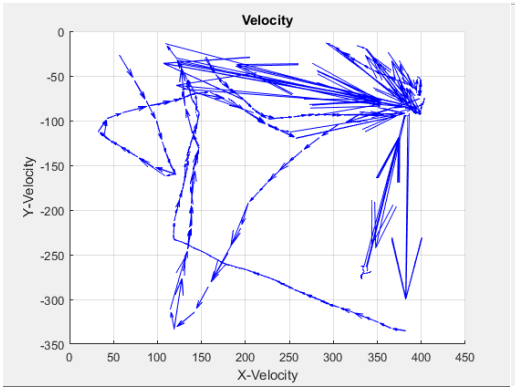
(A)



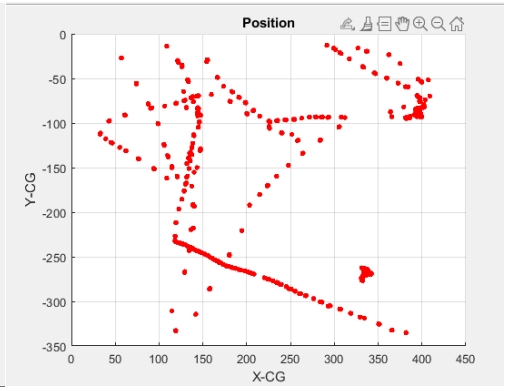
(b)

(C)

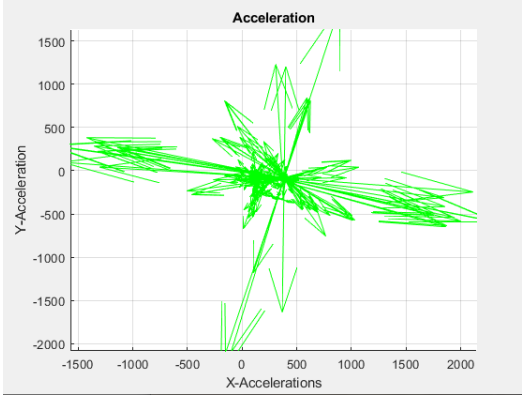
(D)


Velocity
Y-Velocity
X-Velocity
0
-50
-100
-150
-200
-250
-300
-350
0
50
100
150
200
250
300
350
400
450

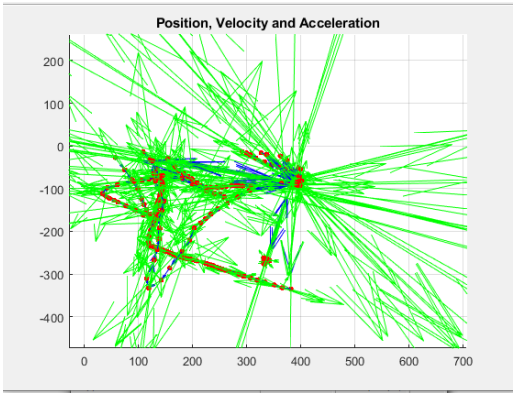
(E)


Position
Y-CG
X-CG
0
-50
-100
-150
-200
-250
-300
-350
0
50
100
150
200
250
300
350
400
450

(F)


Acceleration
Y-Acceleration
X-Accelerations
1500
1000
500
0
-500
-1000
-1500
-2000
-1500
-1000
-500
0
500
1000
1500
2000

(G)


Position, Velocity and Acceleration
200
100
0
-100
-200
-300
-400
0
100
200
300
400
500
600
700

(h)