
IMPROVEMENT OF STOCHASTIC OPTIMIZATION IN DETECTION OF DRIVING DRIVERS MOVEMENT ON AVI VIDEO TYPE USING MATLAB

Solly Aryza¹, Syahril Efendi², Poltak Sihombing³, Sawaluddin⁴, Zulkarnain Lubis⁴

¹Fakultas Sains dan Teknologi Universitas Pembangunan Pancabudi, ^{2,3}Fakultas Ilmu Komputer

Universitas Sumatera Utara, ⁴Fakultas Ilmu Matematika, Universitas Sumatera Utara, ⁴Institute

Tekhnologi Medan,

Sollyaryzalubis@gmail.com

Abstract

Article Info

Received 01 Juni 2021

Revised 10 Juni 2021

Accepted 30 Juni 2021

Vehicle users sometimes do not pay attention to the safety and security of the environment by driving at very high speeds. This is exacerbated by moving that is not by the rules, making it prone to accidents. Detection of the motion of an object in the avi video this time is focused on the operation of detecting the area of moving objects in the avi video. The operation used in processing the detection of an object is an operation on image processing because the video is a collection of images executed in a unit of time. For this reason, this method works by getting information from a video.

Keywords: driver, accident, image, object detection, video.

1. INTRODUCTION

An object that moves may not mean anything to ordinary people. Still, for the field of artificial intelligence, especially the science of pattern recognition (Pattern Recognition), a motion will be something significant [1]. With motion detection technology, it's like we have a surveillance camera at home that can constantly control the activities of objects that move in it. Pattern recognition is a somewhat important aspect that underlies various theories in AI (Artificial Intelligence). The emergence of pattern recognition, object shape, and measurement based on image processing has been utilized in multiple life applications [2]. To detect every passing vehicle, the effort that needs to be done is to install CCTV cameras in strategic places around the road [3].

The use of cameras in this method can be utilized by adding image processing technology. The image processing technology is one of the areas of research that has developed rapidly in recent years. Based on this introduction, digital image processing will be implemented in the vehicle speed detection system in this study [4].

Therefore, the author gets the motivation to try a simple concept to recognize the pattern of a simple idea to acknowledge how the computer can adequately identify a moving object on the video. Here, the author uses the software to try the simulation program to use the 2016 version of Matlab. The primary purpose of this paper is to analyze and try a simple method for detecting moving objects on video so that the computer can retrieve data from the detection results of moving objects. and can be appropriately implemented according to its objectives [5].

The image as the output of a data recording system can be optical in the form of photos, analogue in the form of video signals such as images on a television monitor, or digital, which can be directly stored on a storage medium [6]. From a mathematical perspective, the idea is a continuous function of light intensity in a two-dimensional plane [7]. The light source illuminates the object; then the thing reflects the back part of the light beam; this light reflection is captured by optical devices, such as the human eye, camera, scanner, etc. so that the shadow of the object in the form of an image can be recorded. Image as the output of a data recording system can be optical in the form of photos, analogue in the form of video signals such as images on television monitors, digital, which can be directly stored on a magnetic tape [8].

Images can be grouped into two parts, namely still images and moving images. A still image is a single image that does not move, while a moving image is a series of still pictures displayed in a row (sequential) so that they give the eye the impression of a moving image. Each image in the sequence is called a frame. The pictures on widescreen films or television are essentially made up of hundreds to thousands of frames [9].

There are several classifications of vision, namely Low-Level Vision, Medium Level Vision, and High-Level Vision. Low-Level Vision includes Sensing, which is taking input in the form of images, and Preprocessing, which obtains an image before it is processed. Medium Level Vision consists of the process of Segmentation, Description, Recognition [10].

Segmentation is the process of separating digital images into several regions. The description is the process of describing a snap, while Recognition is an introduction to an embodiment. At a higher level (High-Level Vision), there is an Interpretation process, where Interpretation is an ability to estimate the original shape of the image obtained. This can be done by getting various information needed on the image. So the Interpretation process requires the detection, identification, and measurement of the features in a snap [11].

2. METHOD

The moving image that has been taken as a dataset first goes through the preprocessing stage before entering the system. The length of the video is cut according to the size of the moving image (vehicle), starting to be seen on the camera until the image is not visible. The video results that have been cut will be processed by the system, starting from the pre-processing stage, feature extraction, to speed calculations.

The initial process carried out in analyzing objects in a binary image is object segmentation. The segmentation process aims to group object pixels into regions that represent objects.

There are two approaches used in object segmentation, segmentation based on area boundaries (edges of objects). Edge pixels are traced so that the series of pixels that become the boundary between the object and the background can be known as a whole (border following algorithm). Segmentation into basic shapes (e.g. letter segmentation into vertical and horizontal lines, object segmentation into circles, ellipses and so on).

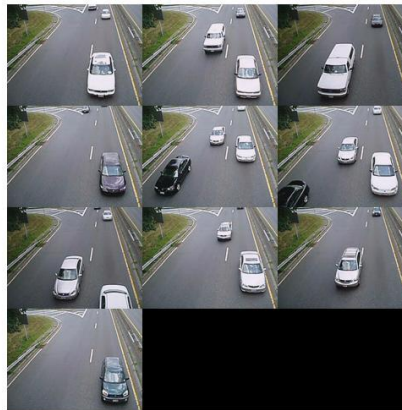
Thresholding is used to adjust the number of degrees of grey in the image. By using thresholding, the degree of grey can be changed as desired. For example, it is hoped to use a grey degree of 16, then divide the value of the grey degree by 16.

3. ANALYZE AND RESULT

The avi video used in this report is an avi file that has the properties of a pixel size of 1471 x 185, with a frame rate of 29.97 frames/second, a data rate of 1285 kbps and a video size of 2.55 MB, with the file name "videotraffic".



Figure 1. Snippet of “videotraffic.avi”



Figures 2. Snippet of the final result of object detection via video

Where the program functions to run all the avi videos

are as follows:

```
clear; close all; clc;
implay('videotraffic.avi');
imshow('traffic.jpg'); truesize;

filename = 'videotraffic.avi';
hVidReader = vision.VideoFileReader(filename,
'VideoOutputDataType', 'single');

opticFlow = opticalFlowHS;

hMean2 = vision.Mean('RunningMean', true);

hblob = vision.BlobAnalysis(...
'CentroidOutputPort', false, 'AreaOutputPort', true,
...
'BoundingBoxOutputPort', true,
'ExtentOutputPort', true, ...
'OutputDataType', 'double', ...
'MinimumBlobArea', 250, 'MaximumBlobArea',
3600, 'MaximumCount', 80);
```

```
sz = get(0,'ScreenSize');
pos = [180 sz(4)-300 200 200];
hVideo1 = vision.VideoPlayer('Name','Original
Video','Position',pos);
pos(1) = pos(1)+220;
hVideo2 = vision.VideoPlayer('Name','Motion
Vector','Position',pos);
pos(1) = pos(1)+220;
hVideo3 = vision.VideoPlayer('Name','Thresholded
Video','Position',pos);
pos(1) = pos(1)+220;
hVideo4 =
vision.VideoPlayer('Name','Results','Position',pos);

% PAUSE/RESUME
a1=false; a2=false;
sz = get(0,'ScreenSize');
figure('MenuBar','none','ToolBar','none','Position',[20
sz(4)-150 100 120])
uicontrol('Style','pushbutton','String','PAUSE',...
'Position',[20 70 80 40], 'Callback','a1=true;');
uicontrol('Style','pushbutton','String','RESUME',...
```

```

    'Position', [20 20 80 40], 'Callback', 'a2=true;');
frameCap = zeros([120 160 3 9], 'single');
capNum = 0;

while ~isDone(hVidReader)
    frame = step(hVidReader);
    grayFrame = rgb2gray(frame);
    flow = estimateFlow(opticFlow, grayFrame);

    y1 = flow.Magnitude;

    vel_th = 4 * step(hMean2, mean(y1(:)));

    fGround = (y1 >= vel_th);
    segmentedObjects = medfilt2(fGround);

    segmentedObjects2 =
imclose(imerode(segmentedObjects, strel('square',2)),
strel('line',5,45));

    [area, bbox, extent] = step(hblob,
segmentedObjects);

    isCar = extent > 0.4;
    bbox(~isCar, :) = [];

    spacing = size(frame,1) - (bbox(:,2)+bbox(:,4)-1);
    crossing = spacing <= 17 & spacing >=1;
    if sum(crossing)
        capNum = capNum+1;
        frameCap(:, :, :, capNum) = frame;

```

```

    ROICap{capNum,1} = bbox;
end

    result = insertShape(frame, 'Rectangle', bbox,
'Color','green');

    plotScaleFactor = 20;
    lines = videoptflowlines(flow.Vx + i * flow.Vy,
plotScaleFactor);
    motionVectors = insertShape(frame, 'Line', lines,
'Color','yellow', 'SmoothEdges',false);

    step(hVideo1, frame);
    step(hVideo2, motionVectors);
    step(hVideo3, segmentedObjects);
    step(hVideo4, result);
    pause(0.1);
    drawnow;
    while a1 && ~a2; drawnow; end
    a1=false;a2=false;
end
figure;montage(frameCap);
openvar('ROICap')

release(hVidReader);
release(hMean2);
release(hblob);
release(hVideo1);
release(hVideo2);
release(hVideo3);
release(hVideo4);

```

4. CONCLUSION

Video is a combination of still images that are read sequentially at a time with a certain speed. One method for motion detection in the video is to convert the video into a grayscale image and a binary image in each frame. Motion detection in the video is detecting areas in the image that are read a time sequentially with a certain speed.

References.

- [1] O. Solly and A. Lubis, "Algorithms in Turbo Pascal," p. 2019, 2019.
- [2] M. F. Anggris, M. T. Ananta, and H. M. Az-zahra, "Rancang Bangun Aplikasi Augmented Reality Pengelolaan Rambu-Rambu Lalu Lintas Menggunakan Global Positioning System (GPS) pada Android," vol. 2, no. 8, pp. 2892–2901, 2018.
- [3] S. Aryza *et al.*, "An effect sensitivity harmonics of rotor induction motors based on fuzzy logic," *Int. J. Eng. Technol.*, vol. 7, no. 2.13 Special Issue 13, pp. 418–420, 2018, doi: 10.14419/ijet.v7i2.13.16936.
- [4] S. Halim, E. Girsang, I. N. Ehrich, and A. Napiah, "Effectivity of Gel Ethanolic Extract of

- Senggani Leaves (*Melastoma candidum* D . Don) in Increasing the Number of Fibroblast Cells and Thickness of Collagen Fibers Against Socket Wound after Tooth Extraction on Male White Rats,” *Eff. Gel Ethanolic Extr. Senggani Leaves (Melastoma candidum D. Don) Increasing Number Fibroblast Cells Thick. Collagen Fibers Against Socket Wound after Tooth Extr. Male White Rats*, vol. 60, pp. 159–173, 2019.
- [5] S. A. Lubis *et al.*, “APPLICATION HYBRID ECO CAMPUS VEHICLE BASED ON SOLAR POWER,” vol. 3, no. 2, 2015.
- [6] E. A. Lim and Y. Jayakumar, “A Study of Neuro-fuzzy System in Approximation-based Problems,” *Matematika*, vol. 24, no. 2, pp. 113–130, 2008, [Online]. Available: <http://www.fs.utm.my/matematika/images/stories/matematika/20082422.pdf>.
- [7] M. Waheedabeevi and A. Sukeshkumar, “New online loss- minimization-based control of scalar and vector-controlled induction motor drives,” *2012 IEEE Int. Conf. Power Electron. Drives Energy Syst.*, pp. 1–7, 2012, doi: 10.1109/PEDES.2012.6484347.
- [8] F. Jadot, F. Malrait, J. Moreno-Valenzuela, and R. Sepulchre, “Adaptive regulation of vector-controlled induction motors,” *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 3, pp. 646–657, 2009, doi: 10.1109/TCST.2008.2003434.
- [9] M. I. J. Ibrahim, S. M. Sapuan, E. S. Zainudin, and M. Y. M. Zuhri, “Preparation and characterization of cornhusk/sugar palm fiber reinforced Cornstarch-based hybrid composites,” *J. Mater. Res. Technol.*, vol. 9, no. 1, pp. 200–211, 2020, doi: 10.1016/j.jmrt.2019.10.045.
- [10] K. Hawari, B. Ghazali, J. Ma, R. Xiao, and S. Aryza, “An Innovative Face Detection Based on YCgCr Color Space,” *Phys. Procedia*, vol. 25, pp. 2116–2124, 2012, doi: 10.1016/j.phpro.2012.03.358.
- [11] S. Aryza, A. N. Abdallah, Z. bin Khalidin, Z. Lubis, and M. Jie, “A Fast Induction Motor Speed Estimation based on Hybrid Particle Swarm Optimization (HPSO),” *Phys. Procedia*, vol. 25, pp. 2109–2115, 2012.