KU LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN

Problem Solving and Engineering Design part 3

ESAT1A1

Max Beerten
Brent De Bleser
Wouter Devos
Ben Fidlers
Simon Gulix
Tom Kerkhofs

Counting and recognizing non-moving objects by means of image processing

PRELIMINARY REPORT

Co-titular
Tinne Tuytelaars

Coach(es)
Xuanli Chen
José Oramas

ACADEMIC YEAR 2018-2019

Declaration of originality

We hereby declare that this submitted draft is entirely our own, subject to feedback and support given us by the didactic team, and subject to lawful cooperation which was agreed with the same didactic team.
Regarding this draft, we also declare that:

1. *Note has been taken of the text on academic integrity (https://eng.kuleuven.be/studeren/masterproef-en-papers/documenten/20161221-academischeintegriteit-okt2016.pdf).*
2. *No plagiarism has been committed as described on https://eng.kuleuven.be/studeren/masterproef-en-papers/plagiaat.*
3. *All experiments, tests, measurements, …, have been performed as described in this draft, and no data or measurement results have been manipulated.*
4. *All sources employed in this draft – including internet sources – have been correctly referenced.*

Abstract

Contents

List of Tables

List of Figures

1 Introduction

Digital image processing has been a crucial part of the current digitalisation movement. From industrial machinery to customer amusement, the vision of computer-aided systems has become a given for most users. While image alteration and manipulation remain a core part of this image processing, nowadays other image related problems which used to be considered an important part of digital image processing are being solved by artificial intelligence. In this category sits the problem considered in this paper: feature extraction, the counting of objects in an image to be more exact. So why use 'traditional' methods to solve this problem? While being a great solution for many problems, artificial intelligence, as most general solutions certain cases are solved more efficiently by specific solutions. Such is the case with object counting, while deep learning algorithms need a big data set as training material, standard image processing requires only the image itself.
Evidently, regardless which way a method processes images it needs a source of the visual. In this paper the focus lays on live object counting, which is only possible with a camera. The choice of hardware greatly impacts the way which methods can be used. This choice will be covered in !TITEL HARDWARE HERE!.
By far the most important part of this task is the algorithm by which the the objects will be counted. Classically object counting algorithms have a standard group of steps: filtering, converting to an intensity matrix, edge detection, converting to a binary matrix, boundary boxing and the counting itself. These steps don't have a fixed order and can occur multiple times in the method. Most of these steps can also be solved in different ways. There exist a wide range of possible filters, kernels, edge detection methods, etc. which all have there upsides and drawbacks. These choice will be discussed in !TITEL SOFTWARE HERE!.

2 Problem Description

Given is a basket with a standard rectangular shape with fixed dimensions. The objective is to count every object in this basket. If possible, the objects can be outlined by the system.

The system used should be based on color and depth images.

3 Design and implementation

3.1 Hardware

3.2 Software

3.2.1 Analysis colour image

There are a lot of options when it comes to software and there exist many different algorithms for image processing. The diagram on FIG. . . XX. . . shows a couple of different methods. There is no 'right-way' to count objects in an image. Different methods have different advantages and disadvantages. The only things that appears in almost all algorithms are:

- Converting the RGB image to grayscale
- Run filters over the image to remove noise

These things are also visible in the diagram.

Method 1
This method is the most simple and easy to implement. It starts with the grayscale image which has been filtered. Then it runs an thresholding-algorithm with a pre-defined threshold value over the image and the output is a binary image. That's an image that only has 0's and 1's in his matrix. It's also possible to use an algorithm to search for the best threshold value. After that there is a simple edge detection algorithm which makes the edges visible. Advantages: it's an easy and fast algorithm. Disadvantages: with a pre-defined threshold value it just classifies pixels based on colour.

Method 2
Method 2 is the reverse of method 1: it starts with an edge detection algorithm. But a different one than in method 1 and a bit more complicated. This algorithm gives back an greyscale image, not a binary image. After that is a threshold algorithm with a pre-defined threshold value and the edges are converted to a binary image. Because there is a lot of noise with this method, it's recommended to use a noise reduction algorithm. Advantages: it detects all kind of objects, not based on colour or shape. Disadvantages: the boundary between different objects needs to be clear.

Method 3
This method is different. It makes a compromise in functionality: it needs a picture of the background alone before it can detect objects. First it loads an background image, converts it to grayscale and runs some filters over the image. After that, the algorithm loops through the image pixel by pixel, checks if the pixel on the image is within a certain range of the same pixel on the background image. If the pixel is within that range, that pixel gets classified as background. The output is a binary image with only the objects. Advantages: it is better in detecting objects, not based on colour or shape. Disadvantages: There needs to be an image of the empty background, and lightning conditions etc. can't change.

3.2.2 Analysis Depth Sensor

Using the RGB image does have some shortcomings: It is hard to distinguish the object from its shadow, a multicoloured object could be seen as multiple different objects or a lot of reflection could make the object undetectable. These are some of the reason why the usage of the depth sensor is advised.
The data enters the system in a matrix in which every number stands for the distance between the sensor and the object (in milimeters). Firstly ...

4 Implementation

Throughout the project a lot of different approaches were tested and discarded. But in essence they all do the same thing they convert the original image to a binary image. In this binary image the objects are represented by one value and the background by another. Afterwards this binary image is analysed and a simple algorithm suffices to count the objects. In this

fase of the program the same code is applicable. This code consists of a few important parts: the actual counting and the drawing of the boundary boxes.

Counting of the objects
The central objective of this research is counting the amount of objects in a specific screen. The general approach to this problem is converting the image to a binary image were black pixels represent the background and white pixels represent the objects. By counting the groups of pixels it is possible to know how many objects the original image contains. In the image processing toolbox for matlab there exist a few functions that come in really handy for this kind of tasks. One of these functions bwlabel actually counts group of pixels of at least 8 that are connected. The syntax of this function goes as follows: [L, num] = bwlabel(BW); where BW represents the binary (or black and white) image; num represents the number of objects in the BW image and where L represents a matrix were the first group of pixels are numbered 1, the second group 2 etc. that way it's easier to get a count for how many objects there are.

Boundary boxes

Edge detection
Test 123

5 Budget management

6 Course Integration

7 Conclusion

8 List of references

9 Appendix