

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*
TECHNOLOGY

CARLOW

At the heart of South Leinster

Research Manual

What's it like Outside?

By

Dermot Rossiter

Supervised by

Paul Barry

Table of Contents

| | |
|--|----------|
| INTRODUCTION | 2 |
| COMPUTER VISION TECHNOLOGIES..... | 2 |
| OPENCV | 2 |
| <i>Advantages</i> | 2 |
| <i>Disadvantages</i> | 2 |
| SIMPLECV | 3 |
| <i>Advantages</i> | 3 |
| <i>Disadvantages</i> | 3 |
| <i>Technology Choice</i> | 3 |
| IMAGE SEGMENTATION | 3 |
| BLURRING/SMOOTHING | 3 |
| EDGE DETECTION | 4 |
| THRESHOLDING..... | 5 |
| IMAGE ANALYSIS..... | 6 |
| COLOUR DETECTION | 6 |
| SIMILAR RESEARCH | 7 |
| CONCLUSION..... | 7 |
| REFERENCES | 8 |

Introduction

The purpose of this project is to research what computer vision techniques can be used to accurately determine what the weather is like by analysing an image of 'outside'. As the project involves determining if a computer can correctly identify the weather from analysing an image, the best solution would be to research vision techniques related to image segmentation, for cases where unwanted information needs to be extracted to allow for for specific regions to be analysed.

Computer Vision Technologies

For this project, 2 free to use open source computer vision libraries were researched for this project. They are as follows:

OpenCV

Advantages

- It's free
- Open source
- Huge library
- Wide range of Platforms and Devices
- Works with various different programming languages
 - C++
 - Python
 - Java
- Big Community
- Good Documentation

Disadvantages

- Learning Curve involved
- Python has poor documentation compared to C++

SimpleCV

Advantages

- Free
- Open Source
- Uses Python only
- Easy to use

Disadvantages

- Poor Documentation
- Poor support base
- Difficult to find resources

Technology Choice

For this project I decided to use OpenCV over SimpleCV because of its better documentation and much more tutorials available. Also, I decided to use the Python programming language instead of C++ for the simple reason that I find much easier to code with, and the documentation for the python code is easier to understand than the C++ documentation.

Image Segmentation

Blurring/Smoothing

Image blurring, also called image smoothing, works by applying a filter/kernel through the entire image, by calculating and blurring a pixel's neighbouring value based on what type of kernel is used. There are many types of blurring techniques such as averaging, Gaussian blurring, median blurring, etc. But the only main difference in each technique is the type of filter used to blur the image.

The benefits of blurring is to remove noise from an image. Specific regions in an image are much easier to analyse if there is less unnecessary information present in the image.

This technique would be useful for this project by removing unnecessary edges in the case of the need to apply an edge detection algorithm for example, to the image.

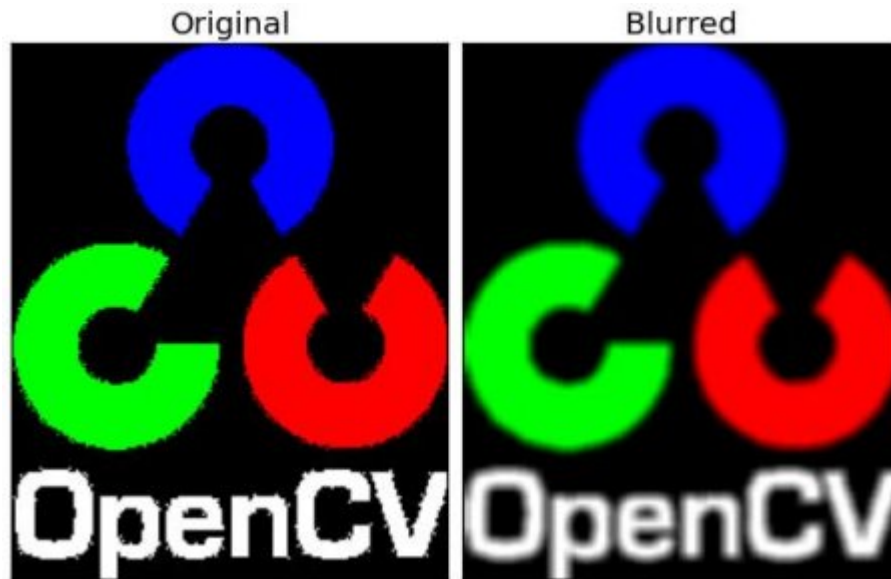


Figure.1: An example of blurring an image. [1]

Edge Detection

Edge detection works by finding points in an image where the image brightness changes sharply. It is commonly used in conjunction with blurred grayscale images. There are many different types of edge detection also, such as Canny edge detection, Sobel edge detection, etc.

Edge detection is widely used in computer vision, in particular in the area of feature extraction. It is commonly used as the first step in extracting information from an image.

This technique would be useful for this project in cases where the horizon or other structural information of importance to extracting the sky region from an image is necessary.



Figure.2: Edge detection example. [2]

The image in figure 2 illustrates of how edge detection could be used for extracting the sky region from an image

Thresholding

Thresholding converts a grayscale image to a binary image. This method works by replacing every pixel value below a threshold point (a value between 0 and 255) to be a black pixel, and any pixel with a value above the threshold point, a white pixel. The threshold point either manually defined, or it is automatically calculated depending on what threshold technique is being used. There are many types of thresholding techniques, such as simple thresholding, adaptive thresholding, Otsu's method of thresholding, etc.

Thresholding is the simplest approach for segmenting an image. It is commonly used for isolating important regions of interest from an image, which makes it easier to analyse a specific piece of information.

Thresholding works better in images where is a significant difference in contrast within the image. As the images below indicate, where the darker tree line is easily identifiable from the original image. This option could work in such cases for extracting the sky region from an image



Example of thresholding. [3]

Image Analysis

Colour Detection

Colour detection works by stepping through each pixel in an image to find its colour value. To find an object based on colour, a range of colour values will need to be defined to get the upper and lower boundaries for the colour. Regions of colour will be identified if a pixel's value falls between the defined range.

It is a useful technique to extract regions of interest based on colour.

This approach could allow for the colour blue to be detected from the sky region of an image.

The image below illustrates the colour blue being identified from the image.



[4]

Similar Research

Some similar research has been done which will be of interest for the implementation of how to use computer vision techniques for this project. These studies are primarily related to detecting the sky region from an image.

Research carried out by Sarhad University of Science & Information Technology [5], proposed a way to detect the sky from an image, based on pixel colour classification using RGB values. They believe that their approach produced “accurate, fast and simple” results.

Conclusion

For the purpose of this project, edge detection will be tested as an option for finding the horizon to try and segment the sky from the image.

References

[1] OpenCV Python Tutorials, Smoothing Images, (2013), [online], Available at: <http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html> [Accessed 10 Oct 2015].

[2] Touqeer Ahmad, Horizon Line Detection using SVM, Dynamic Programming and Edge Detection, (Dec 2013), [online], Available at: <<https://www.youtube.com/watch?v=GkPeNUjig5E>> [Accessed 10 Oct 2015].

[3] Wikipedia, Thresholding (image processing), (Jan 2016), [online], Available at: <[https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))> [Accessed 11 Oct 2015].

[4] Adrian Rosebrock, OpenCV and Python Color Detection, PyImageSearch (Aug 2014), [online], Available at: <<http://www.pyimagesearch.com/2014/08/04/opencv-python-color-detection/>> [Accessed 10 Oct 2015].

[5] Sarhad University of Science & Information Technology (SUIT), An Efficient Approach for Sky Detection, PyImageSearch (July 2013), [online], Available at: <<http://ijcsi.org/papers/IJCSI-10-4-1-222-226.pdf>> [Accessed 12 Oct 2015].