*Object Detection using background subtraction*

Bharadwaz S  
*M.Tech Artificial Intelligence*  
*REVA University*Bengaluru, India  
[bharadwazsripada.ai01@race.reva.edu.in](mailto:bharadwazsripada.ai01@race.reva.edu.in)

Dr. J.B.Simha,  
*REVA academy for Corporate Excellence (REVA),*  
*REVA University*Bengaluru, India  
[jbsimha@gmail.com](mailto:jbsimha@gmail.com)

*Abstract*—Computer vision has come a long way from finding differences between images to self-driving cars. With evolution of neural networks, solving real time computer vision problems has been taken to next level. This paper is aimed to explain the process for object detection using image subtraction. There are various methods of background subtraction, which includes background modelling. In this paper we start by defining background subtraction and go through steps involved in processing of images, followed by exploring post processing techniques to remove superfluous content derived from background subtraction

Keywords— Image subtraction, Computer vision, Neural networks, background modelling

# Introduction

Background subtraction is widely used for tasks like object detection due to its less complexity and good prediction capability. The main idea behind background subtraction is to consider an image as background image which must be a representation of the scene and should not have any objects, and compare this with images from a video stream to identify the presence or absence of an object. Insaf et al. [1] conducted a survey in which background algorithms are classified into Non-recursive and Recursive. This paper aims at identifying objects using background subtraction method by using a static image for background.

The fundamental step in background subtraction is to identify those pixels in current image which are different from the reference image. While there could be different causes for change in the pixel value, it is important to exclude changes due to illumination and changes in shapes of objects. The contents of the paper are split as follows, Section II explains the materials referred for data, Section III explains the Methodologies followed by Results and conclusion in Section IV and V respectively.

# Materials

All the images collected in this dataset are from REVA University. Initially the algorithm was tested with images from [9] for which the results are satisfactory. Then, real time images were taken from the University campus to train and test the algorithm

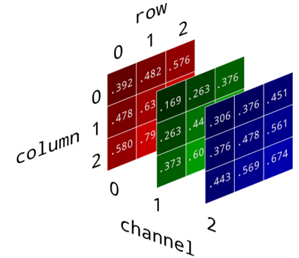
# Methodologies

## RGB to GRAY scale conversion:

RGB to grayscale conversion is an integral part of computer vision problems like object detection due to its reduced bit size. In real time applications gray scale images which are 8 bit wide pixels are taken compared to color images which are 24 bit wide to achieve results faster.[2] Gray image is a monochrome image which consists information related to brightness. Other reasons to use gray images over color images include reduced dimensions, reduced model complexity and applying techniques which only work on gray images. There are different methods to convert an RGB image to a gray scale image. Some of them include, averaging the channels, channel dependent luminance compression, gamma compression, linear approximation of gamma correction.[3] Considering the quality and computational speed we use linear approximation of gamma correction. Each pixel value is iterated through its 3-dimensional color space to arrive at a single pixel value using equation (1)

Y = 0.299R+0.587G+0.114B ----- eq (1)

Where Y denotes the pixel value in gray space and R, G, B denotes the concerned pixel values in 3-dimensional space. To give a better understanding figure (2) shows the pixel arrangement in RGB color space. Figure (3) illustrates the conversion of a color image into grayscale image which we used for our experiment.



Figure(2) – Arrangement of pixels in RGB color space[3]

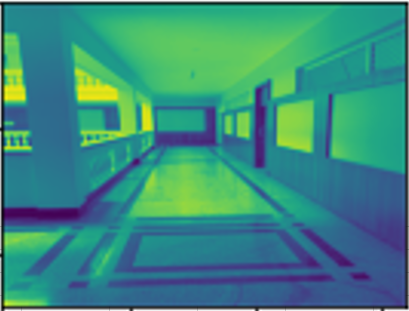
 

Figure (3) – Color image converted into grayscale image

## Gaussian Blurring

In image processing images can be filtered either with a low-pass filter or high-pass filter[4]. Low-pass filters are used for smoothing the image and high-pass filters are used for sharpening the image. The fundamental concept of image blurring is to reduce the noise in the image as it can lead to false detection. Hence, reducing noise improves the quality of the image.

Blurring techniques generally use low-pass filter. The idea behind smoothening is to adjust the pixel value closer to a value, which tends to be an average of all the pixels in the image. Before we go in detail into the blurring technique understanding the concept of ‘kernel’ is key. In image processing, kernels are small matrixes used to alter an image in some way by convoluting between the image and the kernel. Kernels are used for several operations and can only exist in odd number format (3\*3,5\*5, etc.). The convolution operation is the process of adding each element of the image to its neighbor, with the operation weighted by the kernel.[4]

In image processing, Gaussian distribution needs to be approximated by a convolution kernel.[5] Figure (4) is an example of gaussian distribution and a gaussian function for two dimensions is given by equation (2).

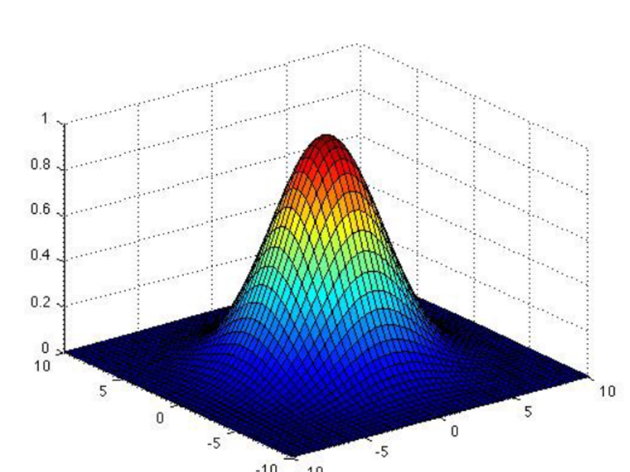
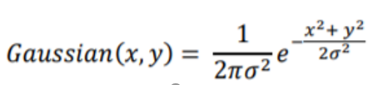
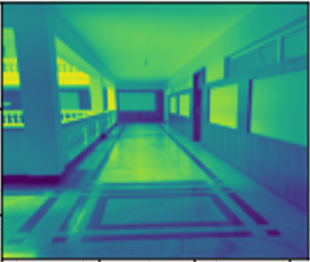
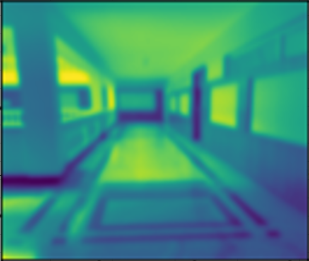


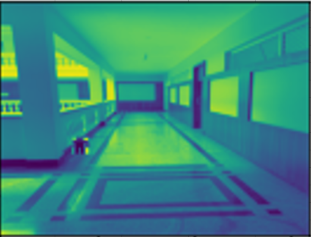
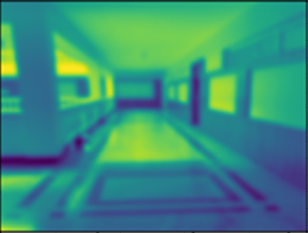
Figure (4) – Gaussian distribution

 ---eq (2)

Here x, y corresponds to a coordinate system starting at the origin of the image (top leftmost pixel) and σ denotes the standard deviation. The recalculated pixel values are basically arrived at considering the weighted average of pixel’s neighborhood. Thus, farther the neighboring pixel from the original pixel, lesser the influence it will have on the recalculated pixel. Figure (5) illustrates the effect of gaussian blurring on the image taken for our experiment.

5.a 5.b

5.c 5.d

Figure (5): Grayscale image before and after applying gaussian blurring. 5.a and 5.b – background image;

5.c and 5.d image with object

## Finding Difference

As both background and current images are preprocessed, difference between both the images can be computed. This can be done by finding structural similarity. The concept of structural similarity was introduced by Wang et al. in [6]. Given an image, the luminance of the surface of the object in the image is described as the product of illumination and reflectance, but the structures of the object are independent of the illumination [6]. Similarity measures of any two equations can be explained using equation (3).

S (x, y) = f (l(x, y), c(x, y), s(x, y)) ---- eq(3)

Where l(x, y) is the function of luminance comparison,

c(x, y) is the function of contrast comparison and

s(x, y) is the function of structure comparison.

The output of subtracting two images consists of values greater than zero only in those pixels, where change is detected. Figure (6) illustrated the change detected between background image and current image.



Figure (6)

## Image Thresholding

There are several methods of image thresholding. An exhaustive survey has been conducted by Chen in [7] and broadly classified image thresholding methods into six categories: Histogram shape-based , clustering-based methods, entropy-based, object attribute-based, spatial methods and local methods. In our experiment we use histogram shape-based method. The histogram and probability mass function (PMF) are given by h(g) and p(g) where g=0...G, G being maximum luminance value in the image. The cumulative probability function for an image is the summation of pixel values and is given by equation (4).

P(g) = ----eq(4)

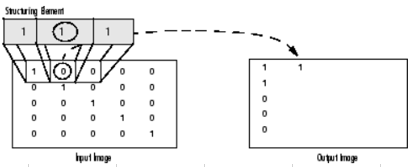
If *T* is the threshold value, the background and foreground mass functions can be expressed as pb(g) , 0≤g≤T and pf(g),T+1≤g≤G. This means all the values less than threshold value are made 0 and values above the threshold are made 1. As a result, only areas where the differences are found are highlighted. The choice of value T is made by trial and error and may vary depending on the application. Figure (7) illustrates the result after thresholding, highlighting only areas where differences are observed.



Figure(7)

## Morphological Operations - Dilation

The two most basic operations in mathematical morphology are dilation and erosion [8]. Both operations take the image and a set of coordinate points known as structuring element or kernel as input. Dilation adds pixels to the boundaries of an object in each image and the number of pixels added is dependent on the size of the structuring element. The value of output pixel is the maximum values of all pixels in the neighbourhood. Figure (8) and Figure (9) illustrates the morphological dilation of a binary and grayscale image respectively.



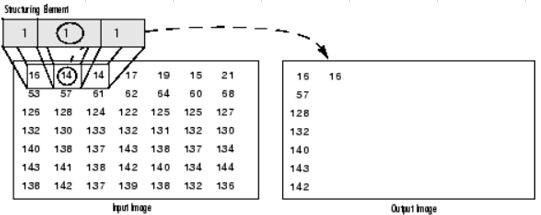


Figure (8) & (9): Morphological Dilation of a binary and grayscale image

Figure (9) illustrates how the output pixel is calculated by taking the highest values from the neighbourhood in the input image. It is also evident the neighbourhood is defined by structuring element or kernel. Figure (10) illustrates the output of before and after dilation.

Figure (10): Comparison before and after dilation

# RESULTS

This section visualizes the results by applying image subtraction as explained in above sessions.

Fig. 11 (a) Background Image 11 (b) Image with Object

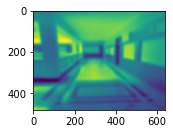
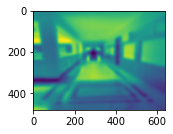
 

Fig. 11 (c) & 11 (d) – Gray scale converted, blurred images

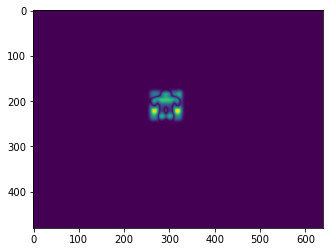
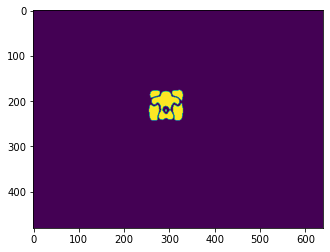
 

Fig. 11 (e) Resultant image of absolute difference

Fig. 11 (g) Resultant image after thresholding

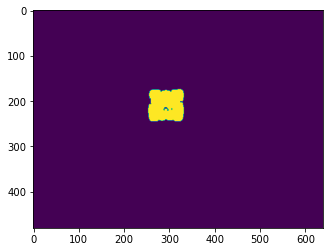
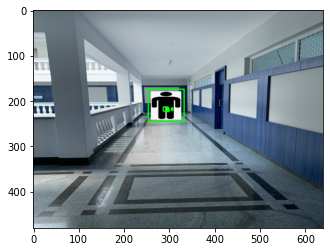
 

Fig. 11 (g) Dilated image Fig. 11 (h) Object detected

# Conclusion

With the above explained approach we can detect the objects in a known environment with less processing and accurate results. However, the main limitation is if the scene changes then the background image needs to be manually updated. To address this, there are sophisticated background modelling techniques which we aim to explore and implement in our future work.

# References

[1] I. Setitra and S. Larabi, “Background subtraction algorithms with post-processing: A review,” *Proc. - Int. Conf. Pattern Recognit.*, pp. 2436–2441, 2014, doi: 10.1109/ICPR.2014.421.

[2] K. Kumar, R. K. Mishra, and D. Nandan, “Efficient Hardware of RGB to Gray Conversion Realized on FPGA and ASIC,” *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 2008–2015, 2020, doi: 10.1016/j.procs.2020.04.215.

[3] iRobot. Brandon Rohrer, “How to Convert an RGB Image to Grayscale.” https://www.kdnuggets.com/2019/12/convert-rgb-image-grayscale.html (accessed Dec. 14, 2020).

[4] E. Pulfer, “Different Approaches to Blurring Digital Images and Their Effect on Facial Detection,” 2019.

[5] E. S. Gedraite and M. Hadad, “Investigation on the effect of a Gaussian Blur in image filtering and segmentation,” *Proc. Elmar - Int. Symp. Electron. Mar.*, no. August, pp. 393–396, 2011.

[6] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.

[7] S. Chen, “Chaotic spread spectrum watermarking for remote sensing images,” *J. Electron. Imaging*, vol. 13, no. 1, p. 220, 2004, doi: 10.1117/1.1631316.

[8] “Morphology - Dilation.” https://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm (accessed Dec. 15, 2020).

[9] <http://www.cvpapers.com/datasets.html>