Edge Detection

ABSTRACT:

**Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision, and machine vision . In matlab the most powerful edge-detection method that edge provides is the Canny method. The Canny method differs from the other edge-detection methods in that it uses two different thresholds (to detect strong and weak edges), and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be fooled by noise, and more likely to detect true weak edges.**

Introduction

Edges are places in the image with strong intensity contrast. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when we want to divide the image into areas corresponding to different objects. Representing an image by its edges has the further advantage that the amount of data is reduced significantly while retaining most of the image formation .This can be used for many applications like ball tracking for some pick and place arms using computer vision, and its easy spotting different shaped objects in edged images rather than a normal image .This edge detection can be done simply by filtering(convolving) an image with suitable highpass filter.

***KEYWORDS***

Edges, Edge detection, Laplacian Filter

Edges consist of mainly high frequencies, hence edge can be detected, in theory, by applying a high pass frequency filter in the Fourier domain or by convolving the image with an appropriate kernel in the spatial domain. In practice, edge detection is performed in the spatial domain , as it is computationally less expensive and often yields better results.

It can be seen that the position of the edge can be estimated with the maximum of the first derivative or with the zero-crossing of the second derivative.

For a discrete one dimensional function f(i),the first derivative can be approximated by

df(i)/d(i) = f(i+1) – f(i)

Calculating the formula is equivalent to convolving the function with a kernel of [ -1 , 1 ].Similarly, the second derivative of can be estimated by convolving f(i) with a kernel of [ 1 , -2 , 1 ].In spatial domain , differentiation process sharpens the given image , thus enhances the edges or linear features.

The most commonly used method of differentiation is ‘Gradient’.The important properties of gradient are :

1. The vector G[f(x,y)] points towards the direction of the maximum rate of increase in the function f(x,y).
2. The magnitude of G[f(x,y)] equals to maximum rate of increase of f(x,y) per unit distance in the direction of G, and can be denoted by G[f(x,y)].

These properties have been used for the design of various filter ,non-directional and directional and zero crossing filters.

Classification of Edge detection techniques

Non-directional Filters : Non-directional Filtering operation consists of implementation of laplacian filter and high boost filter. These filters are known as non-directional as they do not enhance any specific feature having a particular orientation.

Directional Filters : Directional Filtering is performed by a simple directional filter and gradient filter. These filters work on the basis of a linear gradient . This techniques however is sensitive to noise , because the process of differentiation amplifies the noise in an image , resulting in false edges.

Zero-Crossing Filters : Zero-Crossing Filters maybe a directional or non- directional using second order derivative of a gaussian function .

LAPLACIAN FILTERS

Laplacian filters are the non- directional filters as they enhance the linear features having any orientation in the image. The exception applies to linear features oriented parallel to the direction of filter movement as these features are not enhanced . Laplacian of any function f is given by ∇2 f

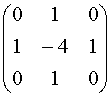
http://academic.mu.edu/phys/matthysd/web226/images/Image172.gif

For an image

 \Delta f(x,y) \approx \frac{f(x-h,y) + f(x+h,y) + f(x,y-h) + f(x,y+h) - 4f(x,y)}{h^2}, \,

Here h=1

There are different ways to find an approximate discrete convolution kernal that approximates the effect of the Laplacian. A possible kernel is



This is called a negative Laplacian because the central peak is negative. It is just as appropriate to reverse the signs of the elements, using -1s and a +4, to get a positive Laplacian. It doesn't matter.

The above equation can be represented as

a1 a2 a3 0 -1 0 a1 a2 a3

a4 x a5 -1 4 -1 a4 Y a5

a6 a7 a8 0 -1 0 a6 a7 a8

Input image Filter or mask Output image

The central pixel (Y) of the output image

= 4x-(a2+a4+a5+a7)

This is conceptually very much similar convolution filtering operation .only the operation window is different .the sum of all elements within the filtering window is zero. hence ,it is a kind of high pass filtering. Basically the Laplacian filtering operation computes the differences between digital counts of the central pixel and the average of the DN value of four adjacent pixels in the horizontal and vertical location. The above equation can be explained as

Y=(X-a4)+(X-a5)+(X-a2)+(X-a7)

So the output image is nothing but the sum of the partial differences in the horizontal and vertical pixels within the operator