Edge Detection

Leonhard Applis

TH Nürnberg

05.11.2018

Table of Contents

What makes an edge?

Figure: Felix

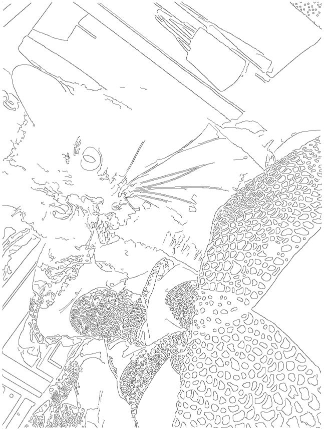


Figure: Felix's Edges

Problem I: Contrast

Figure: High Contrast Felix

Figure: Low Contrast Felix

Problem II: Smoothness

Figure: Smooth Felix

Problem III: Noise

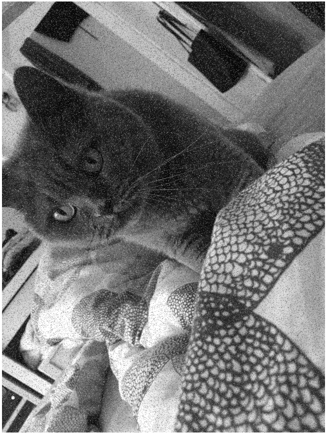


Figure: Salted Felix

Definition

In Image Processing, an edge can be defined as a set of contiguous pixel positions where an abrupt change of intensity, gray- or color-values occur. Edges represent boundaries between objects and background. Sometimes, the edge-pixel-sequence may be broken due to insufficient intensity difference.(Malay K. Pakhira)

Table of Contents

Requirements

1. color values known (for examples only grayscale)
2. picture scale known
3. loaded as pixelmatrix

One dimensional approach

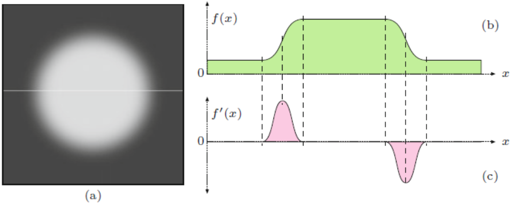


Figure: One dimensional image function and derivation

Only applyable with known, steady functions

Approximating discrete derivation

Problem: the image function is discrete, therefore we need to approximate the derivation

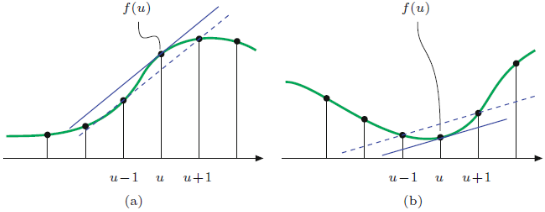


Figure: Approximation of the derivation for discrete imagefunctions

$$\frac{df}{dx}(u) \approx \frac{f(u+1) - f(u-1)}{(u+1) - (u-1)} = \frac{f(u+1) - f(u-1)}{2}$$

Two dimensional approach

If working with full images,we got two dimensions and therefore two partial derivations:

$$I_x = \frac{\partial I}{\partial x}(u,v), I_y = \frac{\partial I}{\partial y}(u,v)$$

the **gradient** at the point *(u,v)* is

$$\nabla I(u,v) = \begin{pmatrix} I_x(u,v) \\ I_y(u,v) \end{pmatrix}$$

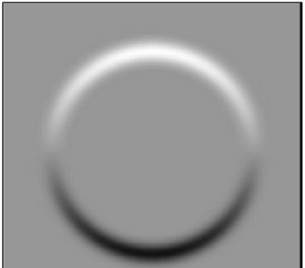
And the **magnitude** is

$$|\nabla I| = \sqrt{I_x^2 + I_y^2}$$

Example

(a) I

(b) I_x



Implementation with filters

Expressing the gradient as a *linear filter* is simple:

$$I_x = \begin{bmatrix} -0.5 & 0 & 0.5 \end{bmatrix} I_y = \begin{bmatrix} -0.5 \\ 0 \\ 0.5 \end{bmatrix}$$

Table of Contents

Table of Contents

Table of Contents