

Image Formation and Analysis (Formation et Analyse d'Images)

James L. Crowley

ENSIMAG 3 - MMIS Option MIRV

First Semester 2010/2011

Lesson 6

6 December 2010

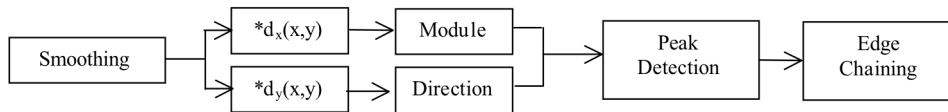
Detection and Description of Image Contrast

Lesson Outline:

1	Describing Contrast with Edges	2
	Edge Detection using integer coefficient filters	2
2	The Hough Transform	3
	The Algorithm	3
	Generalisation of the Hough Transform	4
3	Second Derivatives	5
	Integer Coefficient Second Derivatives	6
4	Beyond Edges - Describing Local Appearance	7
5	Image Description Using Gaussian Derivatives	8
	Gaussian Derivatives Operators	8
	The Sampled Gaussian Functions	8

1 Describing Contrast with Edges

Edge Detection using integer coefficient filters



$$\nabla P(i, j) = \begin{bmatrix} \frac{\Delta p(i, j)}{\Delta i} \\ \frac{\Delta p(i, j)}{\Delta j} \end{bmatrix} = \begin{bmatrix} m_1 * p(i, j) \\ m_2 * p(i, j) \end{bmatrix} = \begin{bmatrix} E_1(i, j) \\ E_2(i, j) \end{bmatrix}$$

Gradient:

$$E(i, j) = \|\bar{E}(i, j)\| = \sqrt{E_1(i, j)^2 + E_2(i, j)^2}$$

Direction of maximum contrast

$$\varphi(i, j) = \text{Tan}^{-1}\left(\frac{E_2(i, j)}{E_1(i, j)}\right)$$

Steps:

- 1) Smoothing - Suppress high frequency noise
- 2) Gradient - Compute first derivatives in row and column
- 3) Detection - Non-maximum suppression with double threshold
- 4) Chaining - Assembly of connected points above threshold.
- 5) Line Segment approximation (multiple algorithms exist).

2 The Hough Transform

The Hough transform is an "optimal" statistical detector for estimating parametric functions from discrete samples. This method was invented for interpreting bubble chamber images in particle physics. It is based on "voting" for possible parameters.

This transform was invented by P.V.C. Hough, Machine Analysis of Bubble Chamber Pictures, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

It was patented in a crude form by IBM in 1962 using $y = mx+c$.

It was made popular by Duda and Hart :

Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11–15 (January, 1972)

The Algorithm

Consider the line equation

$$x \cos(\theta) + y \sin(\theta) + c = 0$$

In the image, for each x,y (free parameters) we need to determine (c, θ)

In the Hough transform, we will create a dual space in which (c, θ) are free parameters.

We will estimate lines as peaks in this dual space. To find peaks we build an accumulator array : $h(c, \theta)$.

Let the c be an integer $c \in [0, D]$ where D is the "diagonal distance of the image.

Let θ be an integer $\theta \in [0, 179]$

Algorithm:

allocate a table $h(c, \theta)$ initially set to 0

For each x, y of the image

for θ from 0 to 179

$$c = -x \cdot \cos(\theta) - y \cdot \sin(\theta)$$

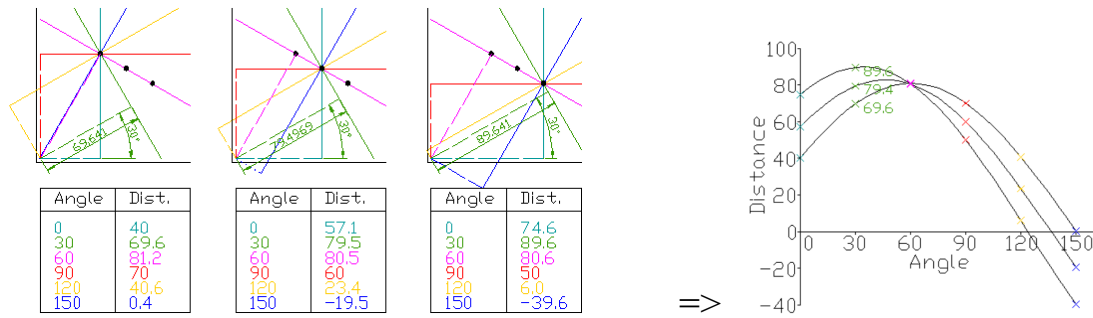
$$h(c, \theta) = h(c, \theta) + E(x, y)$$

End

End

The resulting table accumulates contrast.

Peaks in $h(c, \theta)$ correspond to line segments in the image.



Because we know $\theta(x, y)$, we can limit the evaluation to $\theta(x, y) \pm \Delta\theta$

Generalisation of the Hough Transform

We can represent a circle with the equation:

$$(x - a)^2 + (y - b)^2 = r^2$$

We can use this to create a Hough space $h(a, b, r)$ for limited ranges of r .

The ranges of a and b are the possible positions of circles.

Algorithm:

```

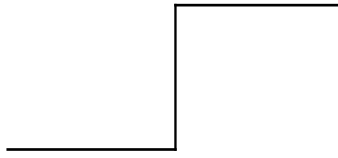
allocate a table  $h(a, b, r)$  initially set to 0.
For each  $x, y$  of the image
  for  $r$  from  $r_{\min}$  to  $r_{\max}$ 
    for  $a$  from 0 to  $a_{\max}$ 
       $b = -y - \sqrt{r^2 - (x - a)^2}$ 
       $h(a, b, r) = h(a, b, r) + E(x, y)$ 
    End
  End
End
End

```

3 Second Derivatives

An alternative to the gradient is to detect edges as zero crossings in the second derivative.

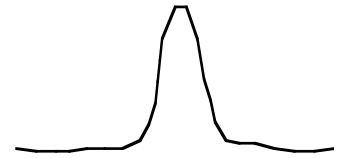
Contrast :



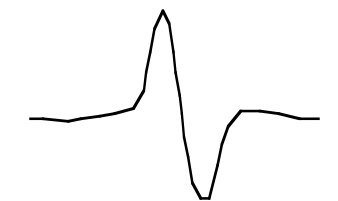
Smoothing :



1st derivative



Second derivative



The zero crossing of the second derivative gives a closed contour corresponding to edges. Unfortunately, many noise contours are commonly observed.

Integer Coefficient Second Derivatives

The second derivative is a form of Laplacian operator:

$$\text{Laplacien: } \nabla^2 p(i,j) = \frac{\Delta_i^2 P(i,j)}{\Delta_i^2} + \frac{\Delta_j^2 P(i,j)}{\Delta_j^2}$$

$$\Delta_i^2 = [1 \ -1] * [1 \ -1] = [-1 \ 2 \ -1]$$

$$\Delta_i^2 = \boxed{-1 \ 2 \ -1} \qquad \Delta_j^2 = \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$$

$$\text{Laplacian: } \nabla^2 p(i,j) = \Delta_i^2 * p(i,j) + \Delta_j^2 * p(i,j)$$

There are several possible discrete forms:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix} + \boxed{-1 \ 2 \ -1}$$

$$L(u,v) = 4 - 2\cos(u) - 2\cos(v)$$

The best is :

$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix} + 2 \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$

$$\text{Gradient: } \|\nabla p(i,j)\| = \sqrt{\left(\frac{\partial P(i,j)}{\partial i}\right)^2 + \left(\frac{\partial P(i,j)}{\partial j}\right)^2}$$

$$\text{Laplacien: } \nabla^2 p(i,j) = \frac{\partial^2 P(i,j)}{\partial i^2} + \frac{\partial^2 P(i,j)}{\partial j^2}$$

In theory

- 1) Zero crossings give closed contours
- 2) Zero crossings can be easily interpolated for high precision.

In practice

Zero crossings detect many small unstable contours.

4 Beyond Edges - Describing Local Appearance

Appearance is the visual manifestation of a phenomena.

We seek a set of K local basis functions, $d_k(x,y)$ to describe "appearance" around a point in an image. Each basis function, $d_k(x,y)$ gives an image "feature", x_k , describing appearance at the pixel (x_o, y_o) .

$$x_k = \sum_{x=-R}^R \sum_{y=-R}^R p(x_o + x, y_o + y) d_k(x, y)$$

Projection of the image neighborhood $P(x_o, y_o)$ onto this set of functions gives a

"feature" vector for appearance, $\vec{X} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_K \end{pmatrix}$

Ideally these functions should be orthogonal

$$\sum_{x=-R}^R \sum_{y=-R}^R d_m(x_o + x, y_o + y) d_n(x, y) = 0 \quad \text{if } n \neq m$$

to minimize the number of features. This is not strictly necessary.

Such a basis can be obtained by a Taylor series representation of the image function $P(x)$ around a point, a :

$$f(x) = f(a) + \frac{1}{1!} f_x(x-a) + \frac{1}{2!} f_{xx}(x-a)^2 + \frac{1}{3!} f_{xxx}(x-a)^3 + \dots$$

The basis set for a Taylor series is the series of local derivatives.

This is called the "local jet". The local jet is a basis for describing appearance.

Because the image is a sampled signal, we need a kernel filter to compute the local Jet. The Gaussian function provides a Kernel with many useful properties.

5 Image Description Using Gaussian Derivatives

Gaussian Derivatives Operators

The Gaussian Function is $G(x, \sigma) = e^{-\frac{x^2}{2\sigma^2}}$

The Gaussian function is invariant to affine transformations.

$$T_a\{G(x, \sigma)\} = G(T_a\{x\}, T_a\{\sigma\})$$

Recall from lesson 2 we saw that $x_r = x_c \frac{F}{z_c}$

The apparent size of an object is inversely proportional to its distance

A change in size (or scale) is a special case of an affine transform:

$$T_s\{G(x, \sigma)\} = G(T_s\{x\}, T_s\{\sigma\}) = G(sx, s\sigma)$$

This is just one of the many interesting properties of the Gaussian function.

The Sampled Gaussian Functions

The Gaussian function is: $G(x, \sigma) = e^{-\frac{x^2}{2\sigma^2}}$

Fourier Transform: $F\{e^{-\frac{x^2}{2\sigma^2}}\} = G(\omega, \sigma) = \sqrt{2\pi} \sigma e^{-\frac{1}{2}\sigma^2\omega^2}$

2D Gaussian Kernel: $G(x, y, \sigma) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} * e^{-\frac{y^2}{2\sigma^2}}$

Fourier Transform: $F\{e^{-\frac{x^2+y^2}{2\sigma^2}}\} = 2\pi\sigma^2 e^{-\frac{1}{2}\sigma^2(u^2+v^2)}$