INF 3300, INF4300
The Hough transform

An introduction

Lars Aurdal,
Norsk Regnesentral,
August 21st 2006

Plan

1. What does the basic edge detection operators actually detect.
2. Hough transform.

Basic edge detection

1. We have already looked at many different edge detection operators.
2. Ex.: Remember Sobel:

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Basic edge detection

1. What does a Sobel filter produce?
2. Approximation to the image gradient:

$$\nabla f(x)$$

3. …which is a vector quantity given by:

$$\nabla \mathbf{f}(x,y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Basic edge detection

1. The gradient is a measure of how the function f(x,y) changes as a function of changes in the arguments x and y.
2. The gradient vector points in the direction of maximum change.
3. The length of this vector indicates the size of the gradient:

$$\nabla f = |\nabla \mathbf{f}| = \sqrt{G_x^2 + G_y^2}$$

Basic edge detection

1. The direction of this vector is also an important quantity.
2. If α(x,y) is the direction of f in the point (x,y) then:

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

3. Remember that α(x,y) will be the angle with respect to the x-axis
4. Remember also that the direction of an edge will be perpendicular to the gradient in any given point

How do we interpret the edge maps?

1. Most natural images will produce a very complicated edge map under the Sobel filter.
2. Remember this is an approximation to a derivation and noise is enhanced. Only rarely will the gradient magnitude be zero.
3. Calculating an approximation to the gradient vector in an image will generally not tell you were the **salient edges are**.

Hough transform

1. If the image contains edges of known shapes we might want to look for groups of edge pixels having this specific shape.
2. One method for searching for such known shapes is the Hough transform.

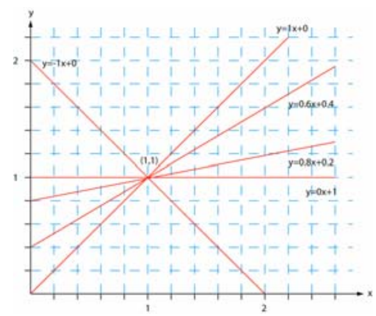
Hough transform – basic idea

1. The Hough transform is based on a very simple observation: A line through the point (x,y) can be written as follows:

$$y = ax + b$$

2. There are infinitely many lines that pass through the point (x,y).
3. Common to them all is that they satisfy the above equation for some set of parameters (a,b).

Hough transform – basic idea



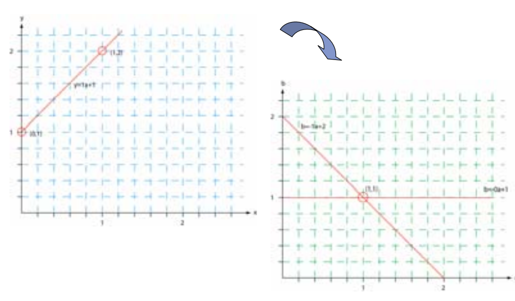
Hough transform – basic idea

1. This equation can obviously be rewritten as follows:

$$b = -xa + y$$

2. We now consider x and y as parameters and a and b as variables.
3. This is a line in (a,b) space parameterized by x and y.
4. Another point (x,y) will give rise to another line in (a,b) space.

Hough transform – basic idea



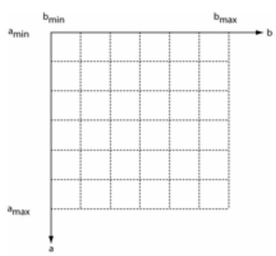
Hough transform – basic idea

1. Two points (x,y) and(z,k) define a line in the (x,y) plane.
2. These two points give rise to two different lines in (a,b) space.
3. In (a,b) space these lines will intersect in a point (a',b') where a' is the rise and b' the intersect of the line defined by (x,y) and (z,k) in (x,y) space.
4. The fact is that all points on the line defined by (x,y) and (z,k) in (x,y) space will parameterize lines that intersect in (a',b') in (a,b) space.

Hough transform – algorithm

1. Quantize the parameter space (a,b), that is, divide it into cells.
2. This quantized space is often referred to as the accumulator cells.
3. In the figure in the next slide a_{min} is the minimal value of a etc.
4. Count the number of times a line intersects a given cell.
5. Cells receiving a minimum number of "votes" are assumed to correspond to lines in (x,y) space.

Hough transform - algorithm



Hough accumulator cells

Hough transform - algorithm

1. Matlab example.

Hough transform – polar representation of lines

1. In practical life we do not use the equation

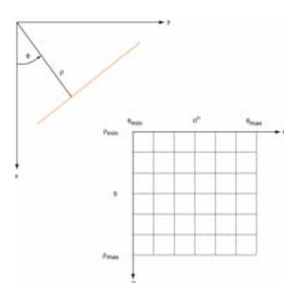
$$y = ax + b$$

in order to represent lines (why?)

2. Rather, we use the polar representation of lines:

$$x\cos\theta + y\sin\theta = \rho$$

Hough transform – polar representation of lines



Polar representation of lines

Hough transform - algorithm using polar representation of lines

1. Input image f is an M×N binary array, edge pixels are marked as ones.
2. Let θ_d and ρ_d be vectors containing the discretized intervals of the parameter space $\rho=[0,\mathrm{sqrt}(M^2+N^2)]$ and $\theta=[0,2\pi]$.
3. The discretization of θ_d and ρ_d must happen with values $\delta\theta$ and $\delta\rho$ giving acceptable precision and sizes of the paramter space.
4. Let the length of the θ_δ and ρ_d vectors be Θ and R respectively.

Hough transform algorithm using polar representation of lines

1. Now let H be the $[\Theta,R]$ accumulator matrix initialized to all zeroes.
2. For all pixels f(x,y)=1 and k=1…Θ let:
   1. $\rho = x \sin(\theta_d(k)) + y \cos(\theta_d(k))$
   2. Find the index j so that $\rho_d(j)$ is closest to ρ.
   3. Increment H(k,j) by one.
3. Find all cells (k_t, j_t) with a value above a user defined threshold t.
4. The output is the set of lines described by $(\rho_d(k_t), \theta_d(k_\tau))$.

Norsk Regnesentral www.nr.no

Hough transform - advantages

1. Advantages:
   a. Conceptually simple.
   b. Easy implementation.
   c. Handles missing and occluded data very gracefully.
   d. Can be adapted to many types of forms, not just lines.

Norsk Regnesentral www.nr.no

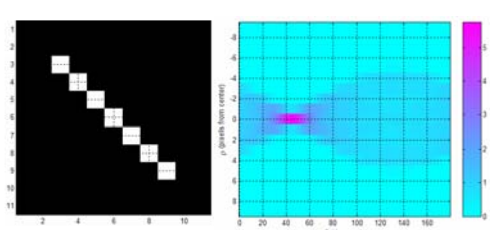
Hough transform - disadvantages

1. Disadvantages:
   a. Computationally complex for objects with many parameters.
   b. Looks for only one single type of object.
   c. Can be “fooled” by “apparent lines”.

Norsk Regnesentral www.nr.no

Hough transform – example 1

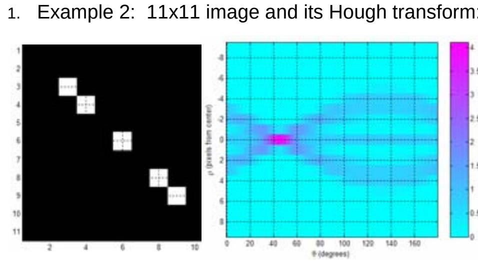
1. Example 1: 11x11 image and its Hough transform:



Norsk Regnesentral www.nr.no

Hough transform – example 2

1. Example 2: 11x11 image and its Hough transform:



Norsk Regnesentral www.nr.no

Hough transform – example 3

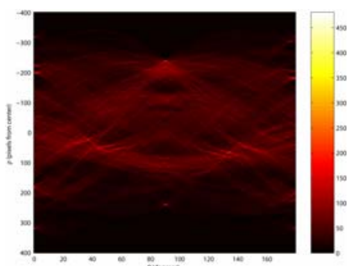
1. Example 3: Natural scene and result of Sobel edge detection:

Hough transform – example 3

1. Example 3: Natural scene and result of Sobel edge detection followed by thresholding:

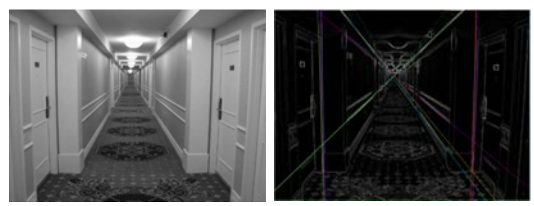
Hough transform – example 3

1. Example 3: Accumulator matrix:



Hough transform – example 3

1. Example 3: Original image and 20 most prominent lines:



Hough transform – exercise 1

1. Next exercise:
   a. Test Hough transform for equal size circles.

Hough transform – exercise 2

1. Next exercise: The randomized Hough transform.
   a. Simple idea (line case): From the edge image, pick two points.
   b. Find the ρ and θ corresponding to this set of points.
   c. Increment the indicated (ρ,θ) cell.
   d. Once a cell reaches a certain (low) count, assume that an edge is present in the image.
   e. Verify this.
   f. If truly present, erase this line from the image
   g. Continue until no more points or until the number of iterations between two detections is to high.
   h. Orders of magnitude faster than the ordinary transform.