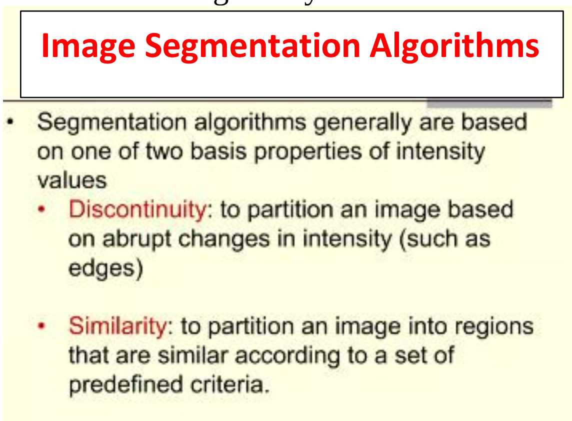
NOTES

What is segmentation –

Image segmentation is a method of dividing a digital image into subgroups called image segments, reducing the complexity of the image and enabling further processing or analysis of each image segment. Technically, segmentation is the assignment of labels to pixels to identify objects, people, or other important elements in the image.
A common uses of image segmentation is in object detection, medical image analysis, computer vision for autonomous vehicles, face recognition and detection, video surveillance, and satellite image analysis.



Detection of isolated points

Detection of lines and edges

Thresholding technique

Region growing

Region splitting and merging

Two principal approaches

• Edge-based segmentation – partition an image based on abrupt changes in intensity (edges)

• Region-based segmentation – partition an image into regions that are similar according to a set of predefined criteria.

Detection of Discontinuities

• Detect the three basic types of gray-level discontinuities – points , lines , edges

• Use the image sharpening techniques –

1. The first order derivatives produce thicker edges
2. The second order derives have a strong response to fine detail, The second order derives have a strong response to fine detail, such as thin lines and isolated points, and noise
3. Laplacian operation

• Can be done by running a mask through the image

Edge Detection

• Edge detection is the approach for segmenting images based on abrupt changes in intensity images based on abrupt changes in intensity.

• What is an edge

– an edge is a set of connected pixels that lie on the boundary an edge is a set of connected pixels that lie on the boundary between two regions.

– Edges are abrupt changes in intensity, discontinuity in image brightness or contrast; usually edges occur on the boundary of two regions.

– an edge is a "local" concept whereas a region boundary, owing to the way, g it is defined, is a more global idea.

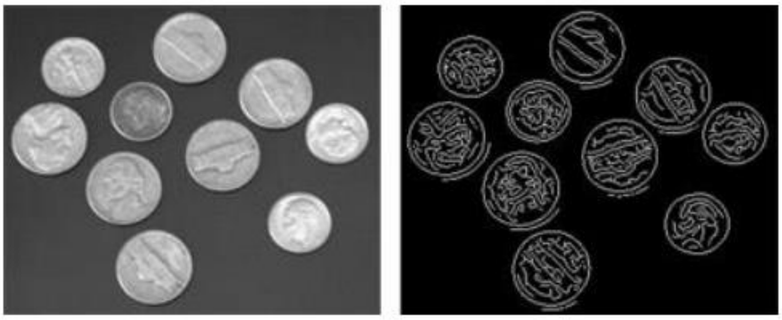
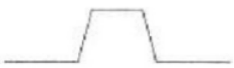
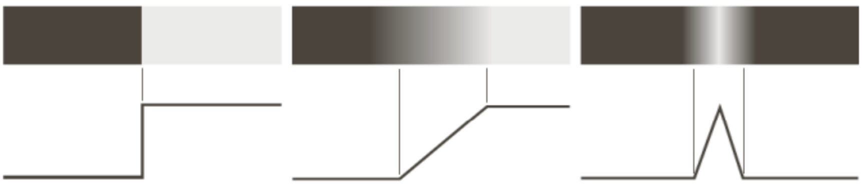


Figure: Original image (left) and edge (right)

• Edge model types – Step edge, ramp edge (thick edge), roof edge and ridge edge.

Ridges

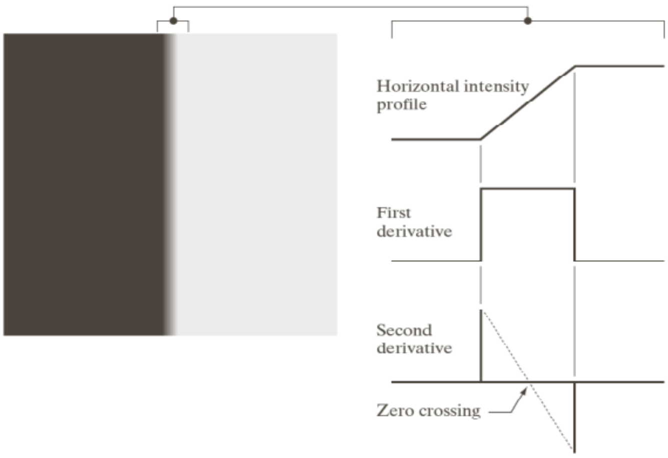




a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

First and Second Derivatives at the edge



a b

FIGURE 10.10 (a) Two regions of constant intensity separated by an ideal vertical ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

1. The magnitude of the first derivative can be used to detect the presence of an edge at a point
2. Second derivative produces two values for every edge in an image. An imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge. **zero-crossing point:** the center of thick edges

Image gradient

- Gradient is a vector

$$\nabla \mathrm{f} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

- The magnitude of the gradient

$$M(x,y) = mag(\nabla \mathrm{f}) = \sqrt{g_x^2 + g_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- The direction of the gradient vector

$$\alpha(x,y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

The direction of an edge at (x, y) is perpendicular to the direction of the gradient vector at that point

Example

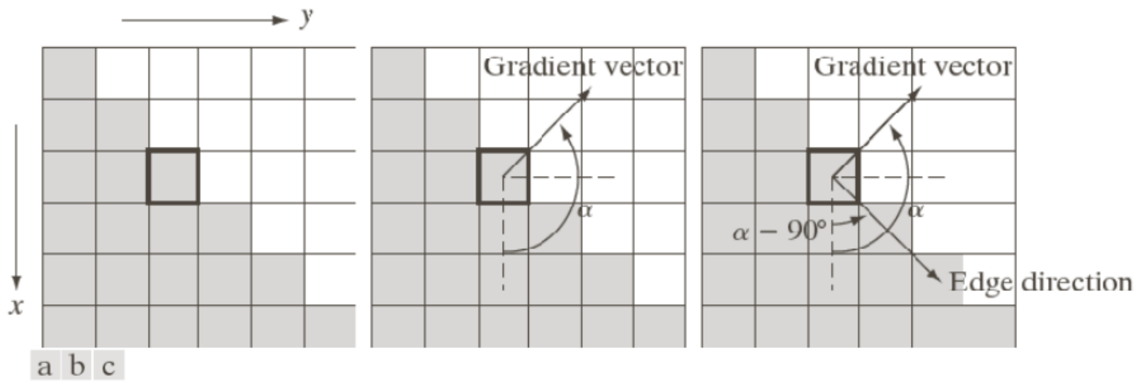


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Edge detection

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.

So, edge detection, as the name suggests, is the automatic detection of object edges in an image. Edge is defined as the locality of connected components, where the image intensity varies rapidly. Thus, it is clear that we need some form of 'derivative' to extract edges. The gradient operator, Laplacian operators etc. are used for this.
The flow of image processing goes as: Image acquisition->pre-processing->enhancement->morphological operations and other processing->segmentation of objects-> description of objects-> object recognition using some sort of intelligence and pattern matching.

The segmentation process, where objects are 'separated' and 'labelled' needs edge detection, as 'detected edges' almost every time help segregate objects (unless you have two apples side by side, without a "visual" boundary. Once edges are detected, object labelling can take place (say, pixel numbers (x_i,y_i,...x_j,y_j) belong to object 1, etc.)
Most machine vision applications require this sort of segmentation, for use in further stages. Also, some things can be completely represented by edges only: like a metal cube being manufactured in an industry). The edge detection will give 4 lines for the cube (from a particular viewpoint) and then angle of intersection, length etc of these lines can be used to distinguish whether the cube has been properly filed.

Why we use edge detection?

• Reduce unnecessary information in the image while preserving the structure of the image.

• Extract important features of an image such as corners, lines, and curves.

• Edges provide strong visual clues that can help the recognition process.

Steps in edge detection

1. Image smoothing for noise reduction

2. Detection of edge points. Points on an edge

3. Edge localization

Here are some of the masks for edge detection.

• Prewitt Operator

• Sobel Operator

Prewitt Operator: Prewitt operator is used for edge detection in an image. By using Prewitt operator, we can detect only horizontal and vertical edges.

For vertical Edges

Mask for detection of vertical edges

-1	0	1
-1	0	1
-1	0	1

Above mask will find the edges in vertical direction and it is because the zeros column in the vertical direction. When you will convolve this mask on an image, it will give you the vertical edges in an image.

Horizontal Edges direction

Mask for detection of horizontal edges

-1	-1	-1
0	0	0
1	1	1

Above mask will find edges in horizontal direction and it is because that zeros column is in horizontal direction. When you will convolve this mask onto an image it would prominent horizontal edges in the image.

Sobel Operator: The sobel operator is very similar to Prewitt operator. It is also a derivate mask and is used for edge detection. Like Prewitt operator sobel operator is also used to detect two kinds of edges in an image:

- Vertical direction
- Horizontal direction

How to differ from Prewitt Operator

The main difference is that in Sobel operator the coefficients of masks are not fixed and they can be adjusted according to our requirement unless they do not violate any property of derivative masks.

For vertical Edges

Vertical Mask of Sobel Operator

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

Horizontal Edges direction

Horizontal Mask of Sobel Operator

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

Above mask will find edges in horizontal direction and it is because that zeros column is in horizontal direction. When you will convolve this mask onto an image it would prominent horizontal edges in the image. The only difference between it is that it have 2 and -2 as a center element of first and third row.

Edge Linking and Boundary Detection

• An edge detection algorithm is followed by linking procedures to assemble edge pixels into meaningful edges.

Meaning, set of pixels from edge detecting algorithms, seldom define a boundary completely because of noise, breaks in the boundary etc. Therefore, Edge detecting algorithms are typically followed by linking and other detection procedures, designed to assemble edge pixels into meaningful boundaries.

• Basic approaches

– Local Processing

– Global Processing via the Hough Transform

LOCAL PROCESSING

Analyze the characteristics of pixels in a small neighborhood (3x3, or 5x5) about every point that has undergone edge detection. All points that are similar are linked, forming a boundary of pixels that share some common properties.

2 principal properties for establishing similarity of edge pixels: -

- strength of the response of the gradient operator used to produce the edge pixel
- direction of the gradient.

A point in the predefined neighbourhood of (x,y) is linked to the pixel at (x,y) if both magnitude and direction criteria are satisfied. This process is repeated for every location in the image.

EDGE LINKING AND BOUNDARY DETECTION USING HOUGH TRANSFORM

1. Compute the gradient of an image and threshold it to obtain a binary image.
2. Specify subdivisions in the $\rho\theta$-plane.
3. Examine the counts of the accumulator cells for high pixel concentrations.
4. Examine the relationship (principally for continuity) between pixels in a chosen cell.

Continuity:

1. based on computing the distance between disconnected pixels identified during traversal of the set of pixels corresponding to a given accumulator cell.
2. a gap at any point is significant if the distance between that point and its closet neighbor exceeds a certain threshold.

Link criteria:

1) the pixels belonged to one of the set of pixels linked according to the highest count

2) no gaps were longer than 5 pixels

A comprehensive guide to edge detection with Hough transform

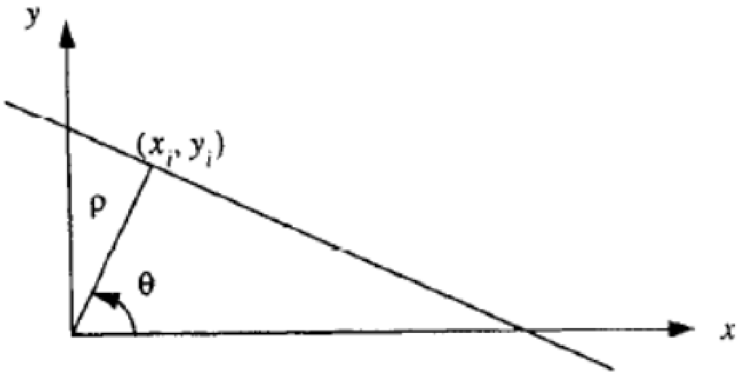
Hough transform is a feature extraction method used in image analysis. The Hough transform is a technique which can be used to isolate features of a particular shape within an image. Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc.

The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

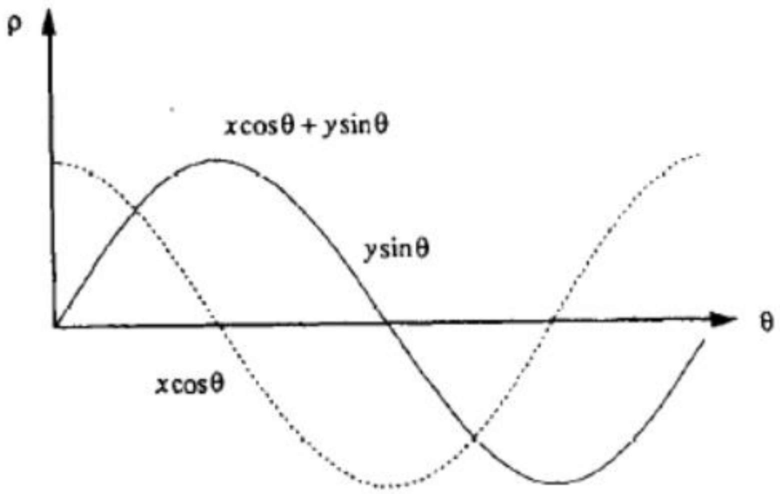
Hough transform in its simplest from can be used to detect straight lines in an image.

It maps a straight line y=mx+c in a Cartesian coordinate system into a single point in the (ρ,θ) plane or ρ=xcos θ+ysin θ

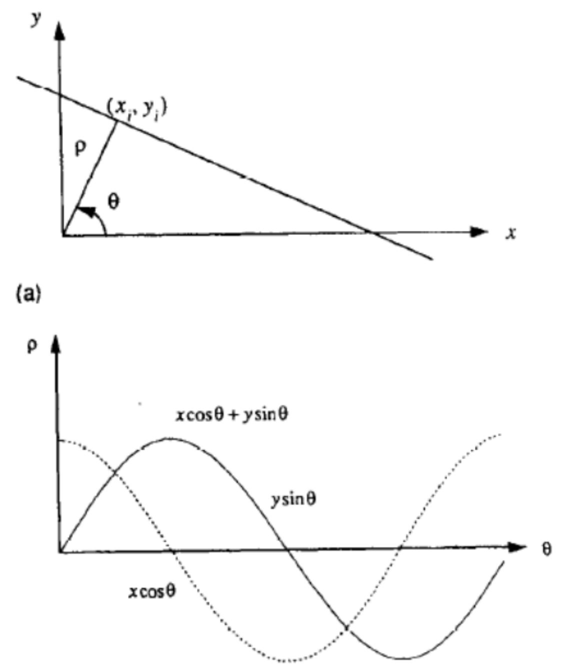
For a point (x,y) in the Cartesian coordinate plane, there will be an infinite number of curves in the (ρ,θ) plane.



Cartesian xy co-ordinate system



pΘ parametric plane

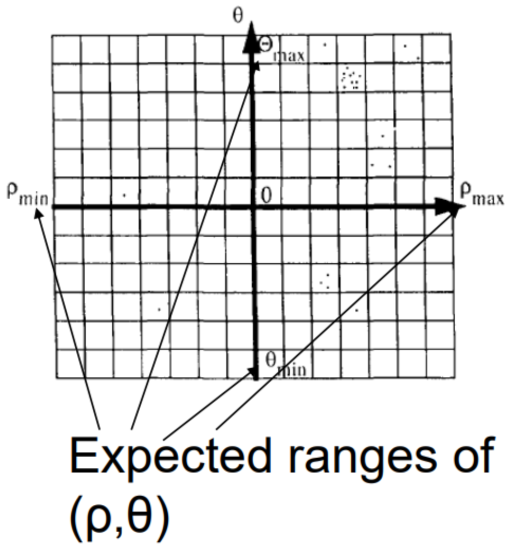


When two points (x_i,y_i) and (x_j,y_j) lie on the same straight line , the curves in the (ρ,θ) plane which correspond, respectively , to the two points (x_i,y_i) and (x_j,y_j)in the Cartecian plane will intersect at a point

This intersection point determines the parameter of the line that joins these two points.

Similar arguments apply for the three collinear points.

This property between Cartesian plane and the parametric plane will be useful in finding the line that fits points in the xy plane. The Hough transform approach is to find the points of intersection in the curves, each of which corresponds to a line in the Cartesian xy plane.



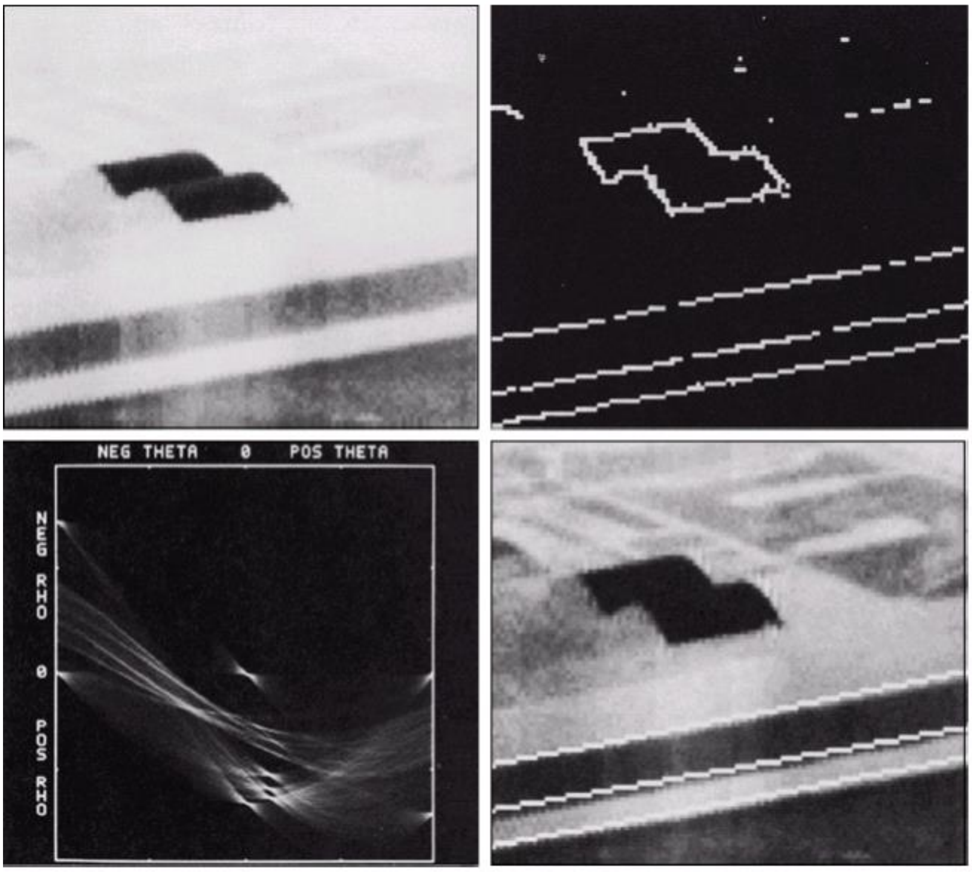
The two dimensional parameter space is descritized into the finite interval (Accumulator cell or 2D Bins)

Each of the cells first initialized to zero

For each point in the cartesian image plane compute the value of ρ_i for various values of θ_i

p_i=x cos θ_i + y sin θ_i

If there are N collinear points lying on a line, we get N sinusoidal curve that intersects at (ρ_i,θ_i) in the parametric space. The peaks in the (ρ,θ) plane accumulator array therefore give strong evidence of the existence of lines in the image.

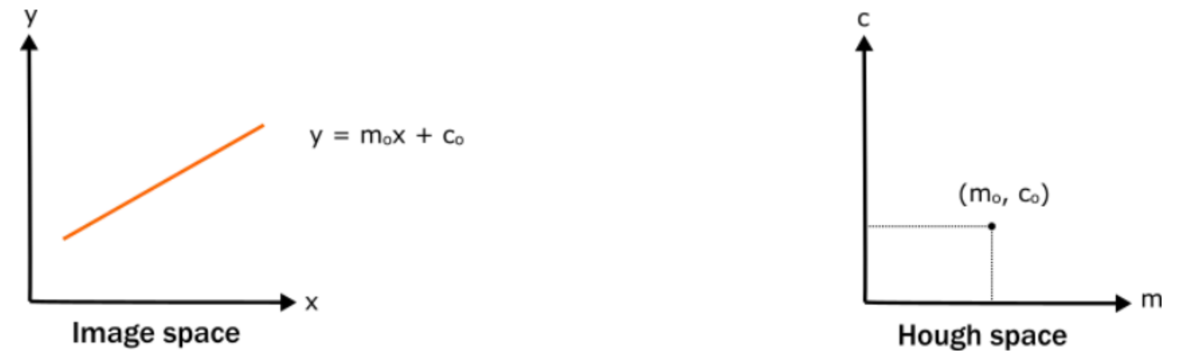


a b
c d

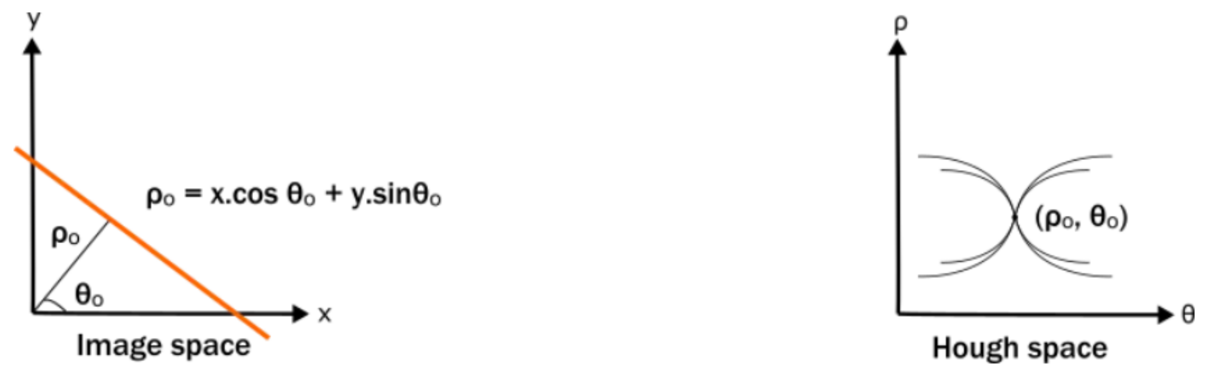
FIGURE 10.21 (a) Infrared image. (b) Thresholded gradient image. (c) Hough transform. (d) Linked pixels. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

Algorithm

A straight line is the simplest boundary we can recognize in an image. Multiple straight lines can form a much complex boundary. We transform the image space into hough space. By doing this we convert a line in image space to a point on hough space.



The equation of the line in the image space is of the form y = mx + c where m is the slope and c is the y-intercept of the line. This line will be transformed to a point of the form (m, c) in the hough space. But in this representation m goes to infinity for vertical lines. So let us use the polar coordinates instead.

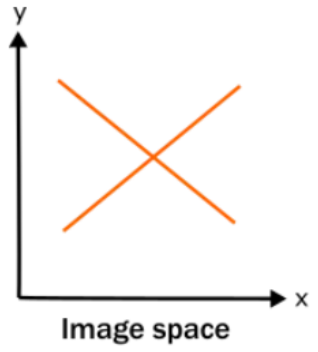


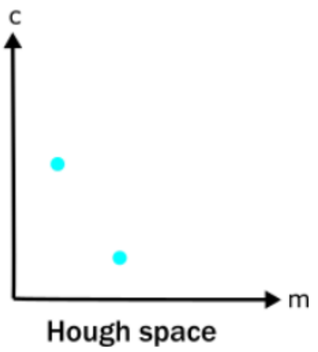
The line is represented by the length of that segment ρ, and the angle θ it makes with the x-axis. This line will be transformed to a point of the form (ρ,θ) in the hough space.

The Hough transform constructs a histogram array representing the parameter space (i.e., an M x N matrix, for M different values of the radius ρ and N different values of angle θ). For each parameter combination, ρ and θ we then find the number of non-zero pixels in the input image that would fall close to the corresponding line, and increment the array at position (ρ,θ) appropriately.

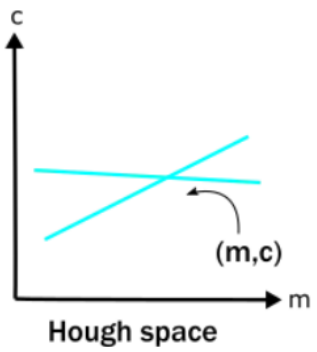
Intuition for line detection

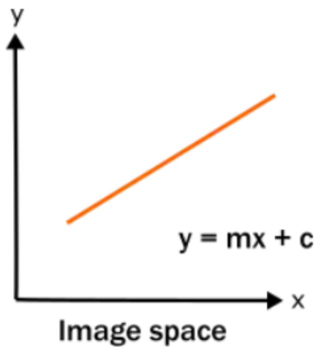
The intersection of multiple lines in image space represent corresponding multiple points in hough space.





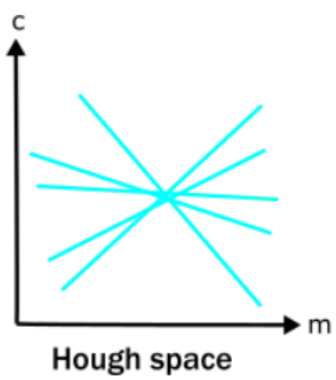
Similarly, the reverse i.e., lines intersecting at a point (m, c) in hough space can be transformed to a line y = mx + c in image space.

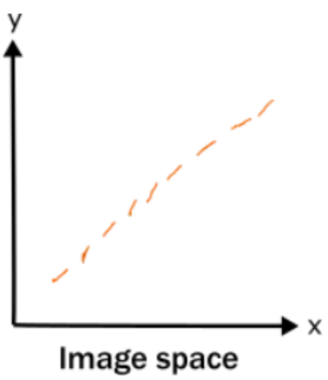




If we have a line made up of many segments or points close to the same line equation in the image space, that turns into many intersecting lines in hough space.

So, consider a line in the image space which is an edge detected and has small discontinuities. To find the continuous line in an image we can transform this discontinuous line in image space to hough space and look for intersection points in hough space. This intersection point in hough space will represent the continuous line in image space.





Compare line and edge detection

Edge detection is the process of finding outlines in an image, whatever they look like.

Line detection finds line segments (sometimes by extension, other geometric figures such as circular arcs).

An Edge in an image is a significant local change in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity.
Line is Non-local change (Line detection usually performed on the output of an edge detector)

Edge detection and line detection are two distinct techniques used in pattern recognition for binary images. Edge detection is the process of locating and defining the boundaries between different objects in an image. On the other hand, line detection is focused on finding straight lines within an image.

Typically, edge detection serves as a foundation for line detection, as edges are used to identify lines in an image. However, it is possible to detect lines without first detecting edges, though it can be more challenging and may lead to less accurate results.

Edge detection is usually a pre-step for line detection, but it's not always a necessary step.