

Hough Transform

What is the Hough Transform?

- A method to detect the presence of different types of **shapes** in an image – eg. Lines, circles, ellipses, polygons etc.
- Used for detection of important features in images- eg: roads or buildings in satellite imagery or photos, biological cells in microscope images.



<http://ugroad.com/index.php/projects/remote-sensing-to-measure-regional-growthworkshop/>



http://microscopy.zeiss.com/microscopy/en_de/products/light-microscopes/axio-observer-for-biology.html

Beyond edge detection

- The output of edge detectors can be **noisy** and yield outputs with **missing points**, and it is often difficult to group edges to isolate specific shapes such as lines, circles or ellipses.

Hough transform for line detection: big picture

- Perform edge detection on the image using (say) Canny's algorithm.
- Every significant edge point casts one **vote** for every line on which it could potentially lie.
- The lines which get **maximum votes** are the winners and can be marked out prominently.
- The crux is: **how do you represent a line?**



<http://www.keymolen.com/2013/05/hough-transformation-c-implementation.html>

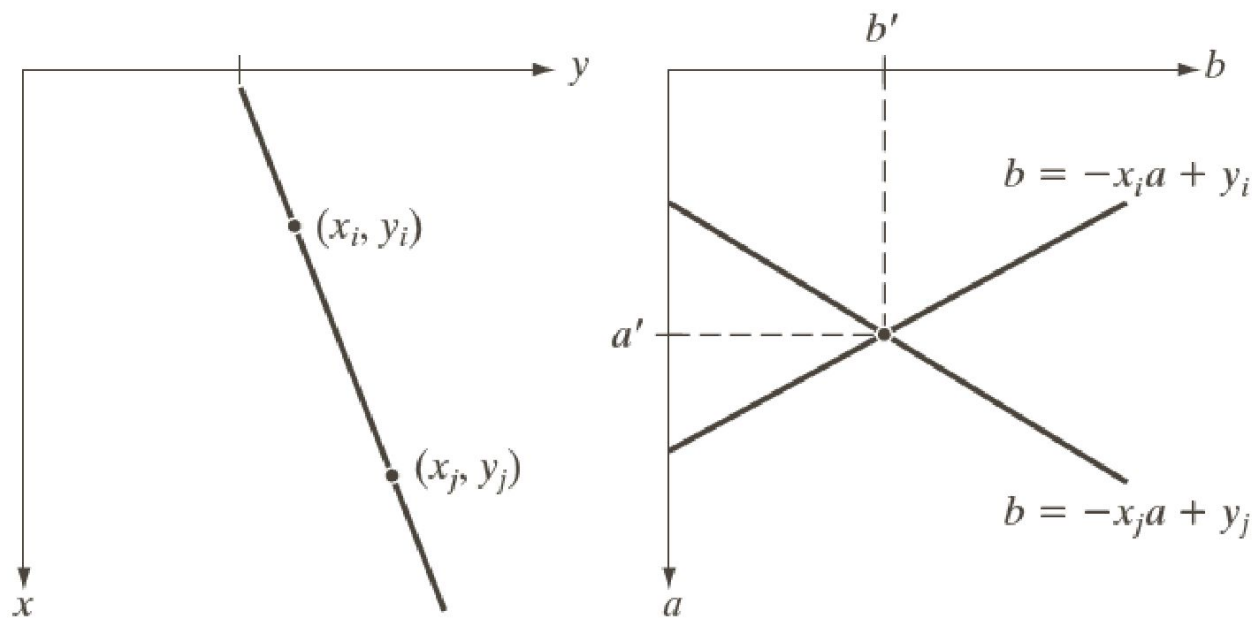


Hough transform for line detection

- Consider a point (x_i, y_i) in (x, y) space.
- A line passing through this point can be represented as $y_i = ax_i + b$ (**slope-intercept form**). The line is parameterized by (a, b) .
- There are infinitely many lines passing through the point (x_i, y_i) , and each of them will satisfy $y_i = ax_i + b$ for some value of (a, b) . Thus a line becomes a point in the (a, b) space.
- The (a, b) space is called the **parameter space**.

Hough transform for line detection

- The set of all lines passing through (x_i, y_i) will constitute a **single line** in **parameter space**.
- The set of all lines passing through some other point (x_j, y_j) will constitute another **line** in **parameter space**.
- If the point (x_j, y_j) also satisfies $y_j = ax_j + b$, then the respective lines for these points in **parameter space** will intersect at the **point** (a, b) in **parameter space**.



a b

FIGURE 10.31
 (a) xy -plane.
 (b) Parameter space.

Hough transform for line detection

- There is a problem with the slope intercept representation for a line – the slope is infinity for vertical lines, and hence a can take values that range from $-\infty$ to $+\infty$.
- It is more convenient to use the normal representation, i.e. the form $x \cos \Theta + y \sin \Theta = \rho$.
- The value of Θ is the angle made by a line perpendicular to the original w.r.t. the X axis, and ranges from -90 to +90 degrees.
- The value of ρ is the perpendicular distance from the origin onto the line, and it ranges from 0 to the *length of the largest diagonal in the image* (why?).

Hough transform for line detection

- The parameter space for the line has changed from (a, b) to (ρ, Θ) . A single point in the XY space corresponds to a **sinusoid** in this parameter space.
- The line joining two points in XY space is represented by a point obtained from the **intersection of their corresponding two sinusoids** in the parameter space.

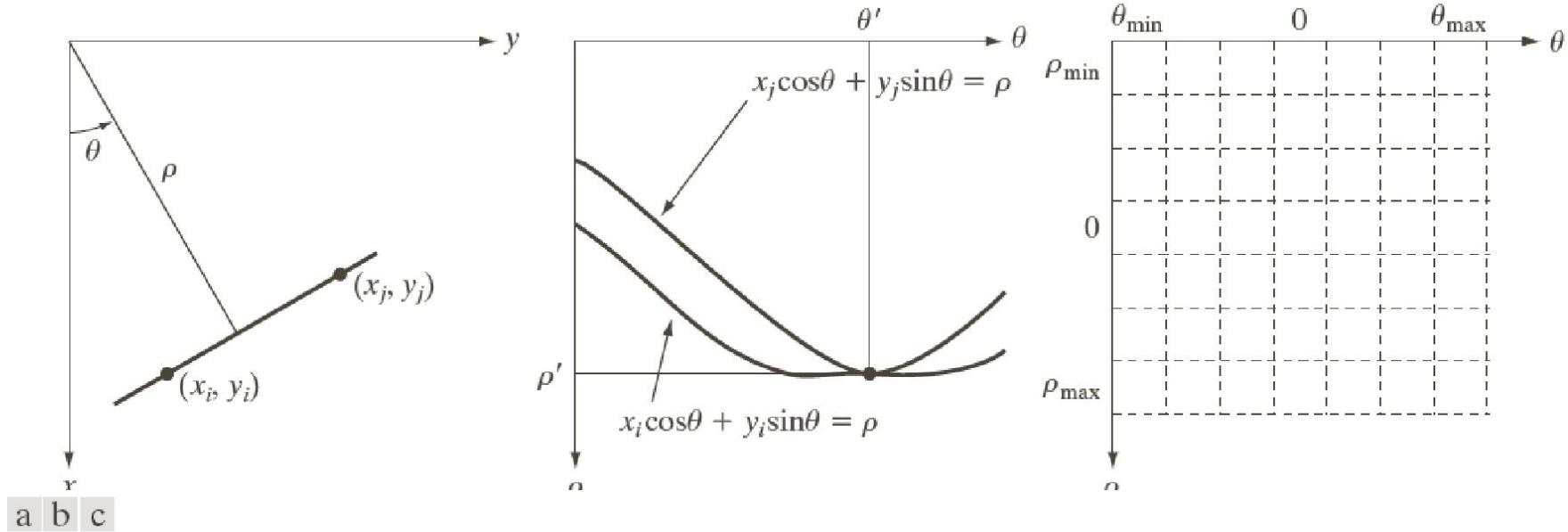
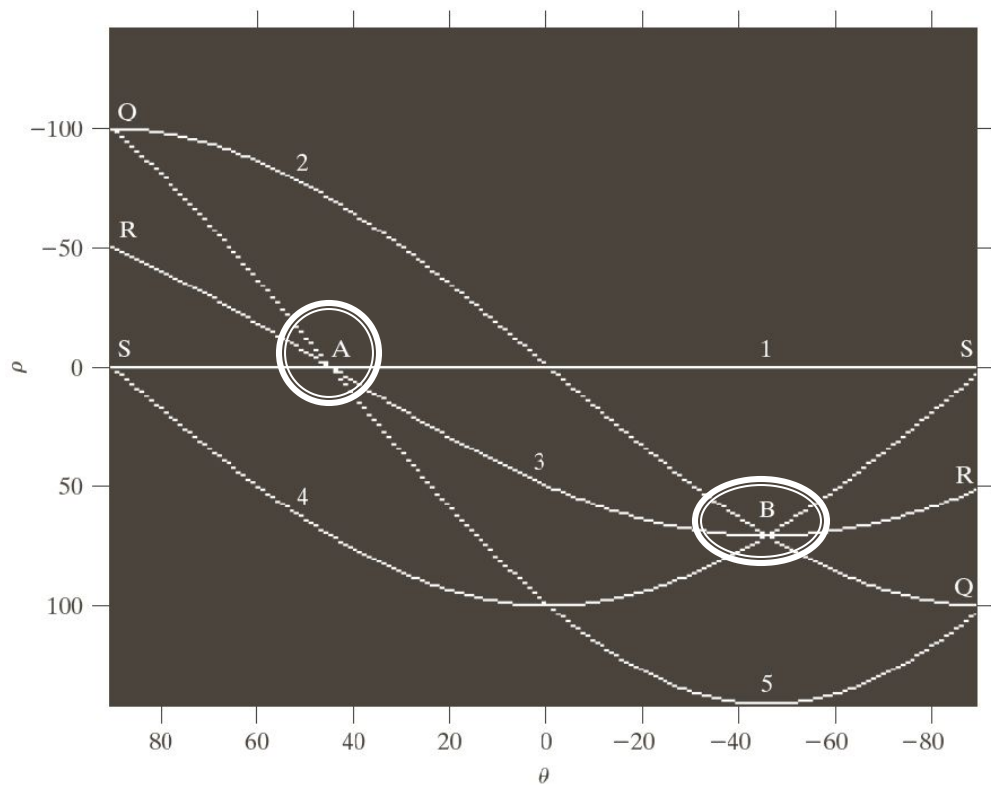


FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy -plane. (b) Sinusoidal curves in the $\rho\theta$ -plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy -plane. (c) Division of the $\rho\theta$ -plane into accumulator cells.



a
b

FIGURE 10.33

(a) Image of size 101×101 pixels, containing five points.

(b) Corresponding parameter space.

(The points in (a) were enlarged to make them easier to see.)

Hough transform for line detection: algorithm

- We will divide the entire (ρ, Θ) space into bins - also called **accumulator cells**, which form a 2D array **A**.
- Perform edge detection on the image using (say) Canny's algorithm.
- For every edge point (x_k, y_k) in the XY space, and for every value of Θ ranging from -90 to $+90$, compute the corresponding value of ρ , using the fact that $\rho = x_k \cos \Theta + y_k \sin \Theta$, and approximate it to the closest allowed cell value in (ρ, Θ) space.
- If some angle Θ_a yields a solution ρ_a then increment the frequency count $A(\Theta_a, \rho_a)$ by 1 (this is called **voting**).
- At the end, the value in $A(\Theta_a, \rho_a)$ indicates the number of edge points that (approximately) satisfied the equation $x \cos \Theta_a + y \sin \Theta_a = \rho_a$.

Hough transform for line detection: algorithm

- Pick those lines (i.e. **cells** in the **parameter space**) which have maximum number of votes.
- Mark out those lines in the original image.
- Note: Any single point votes for many lines, but lines corresponding to actual structures in the image will receive votes from many points. Votes for the other lines (i.e. not corresponding to prominent structures) will get scattered.

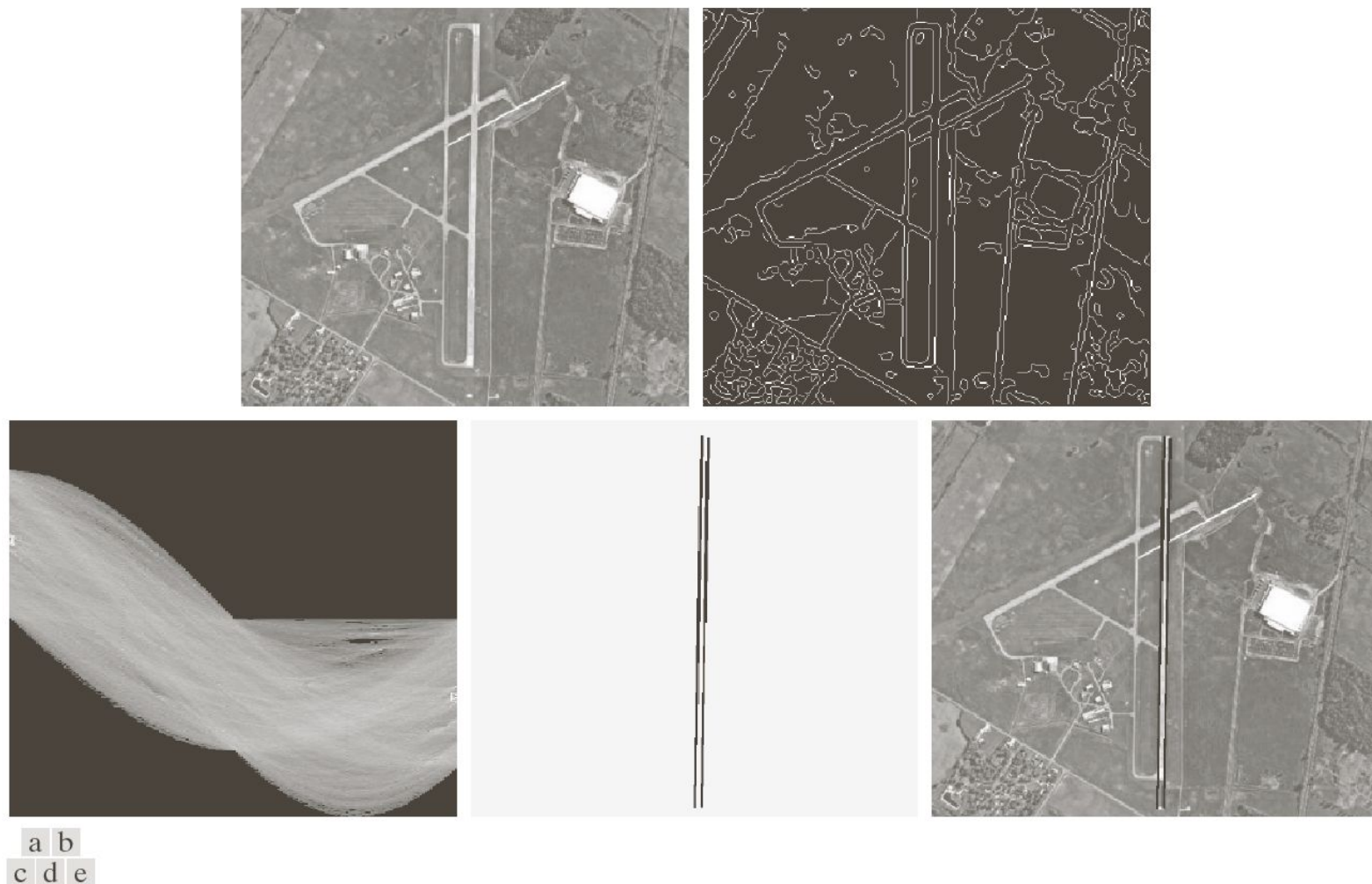
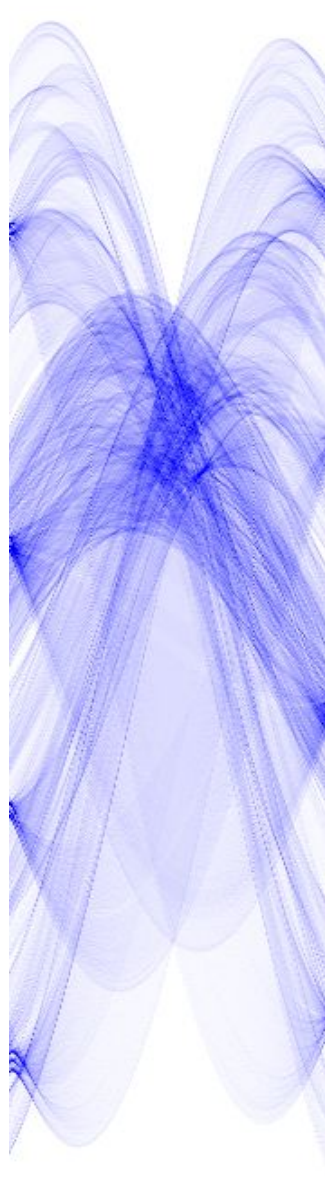


FIGURE 10.34 (a) A 502×564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes. (e) Lines superimposed on the original image.



<http://www.keymolen.com/2013/05/hough-transformation-c-implementation.html>



<http://www.knoerig.de/Bildsignalverarbeitung.html>

Speed-up heuristic

- For every edge point, we know the edge direction Θ' (perpendicular to the gradient at that point).
- We can therefore constrain that an edge point is allowed to vote for only those accumulator cells corresponding to (say) $\Theta' - K \leq \Theta \leq \Theta' + K$ where K is around 20 degrees.

Hough transform for circles

- A circle is represented by the equation $(x-a)^2 + (y-b)^2 = r^2$.
- For circle detection, there are 3 parameters (a, b, r) , so the parameter space is 3-dimensional.
- For every edge point in an image, for every reasonable value of r and a , compute the possible values of b . Increment the appropriate accumulator cell (a, b, r) just as before.



http://www.mitov.com/wiki/index.php?title=Class_VLHoughCircles.TVLHoughCircles

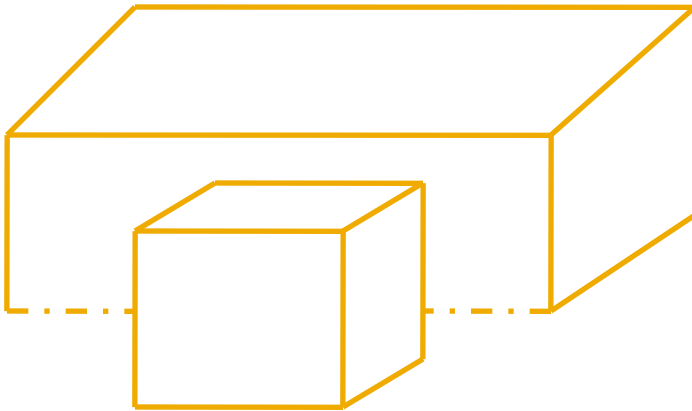
Hough transform for ellipses

- An axis parallel ellipse is given by $\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$
- The parameter space is 4-D.
- In addition, if the major axis is oriented at some angle Θ w.r.t. X axis, the parameter space becomes 5D.

Hough transform: pros



- It can handle cases where pixels lying on the same line have gaps in between them due to noise or occlusions.



Hough transform: cons



- It considers a line to be of infinite extent – you will have to be careful while marking out the final line segments in the image.
- If the resolution of the parameter grid is too coarse, potentially different curves could fall into the same bin.
- If the resolution is too fine, the votes of different points lying close to the same curve will get bifurcated and you may miss some significant structures. In the latter case, the problem is worsened by noise.
- Computational complexity is **exponential** in the number of parameters (why?).