CS-465 Computer Vision

Nazar Khan

PUCIT

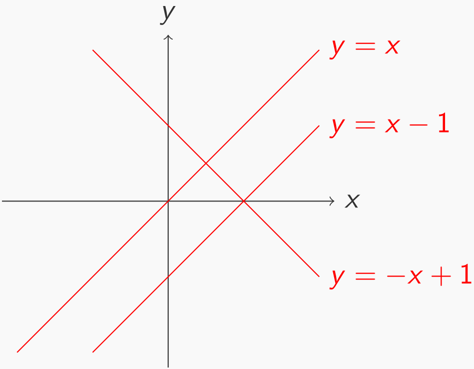
7. Hough Transform

The Hough Transform

- A powerful method for detecting curves from boundary information.
- Exploits the duality between points on a curve and parameters of the curve.
- Can detect analytic as well as non-analytic curves.

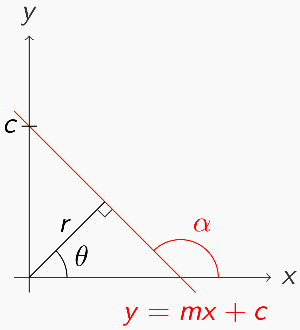
Analytic representation of a line

- In the analytic representation of a line $y = mx + c$, every choice of parameters (m, c) represents a different line.
- This is known as the *slope-intercept* parameter space.
- Weakness: vertical lines have $m = \infty$.



Polar representation of a line

- Solution: Polar representation (r, θ) where
  - r = perpendicular distance of line from origin
  - θ = angle of vector orthogonal to the line
- Every (r, θ) pair represents a 2D line.



$$
\begin{aligned}
y &= mx + c \\
m &= \tan(\alpha) = \tan(\theta + \frac{\pi}{2}) \\
&= \frac{\sin(\theta + \frac{\pi}{2})}{\cos(\theta + \frac{\pi}{2})} = \frac{\cos(\theta)}{-\sin(\theta)} \\
c &= \frac{r}{\sin(\theta)} \\
y &= -\frac{\cos(\theta)}{\sin(\theta)}x + \frac{r}{\sin(\theta)} \\
r &= x\cos(\theta) + y\sin(\theta)
\end{aligned}
$$

Hough Transform for Line Detection

- An algorithm for finding lines given some edge points.
- Given any point (x, y), the line passing through it with angle θ must have perpendicular $r = x\cos(\theta) + y\sin(\theta)$.
- Given any edge pixel (x, y), potentially 360 lines could pass through it assuming angular resolution of 1°.
- So pixel (x, y) should *vote for* all those lines.
- By repeating this process for all edge pixels, actual lines will get a high number of votes.

Hough Transform for Line Detection

Pseudocode

initialize 2D (vote) accumulator array A to all zeros.
for every edge point (x, y)
    for $\theta = 0$ to π
        compute $r = x\cos(\theta) + y\sin(\theta)$
        increment $A(r, \theta)$ by $1 \longleftarrow$ vote of point (x, y) for line (r, θ)
valid lines are where $A >$ threshold

Hough Transform for Line Detection
Detailed Pseudocode

Detailed Pseudocode

1. range_θ = 360 degrees
2. binsize_θ = 1 degree (for example)
3. size_θ = ceil(range_θ / binsize_θ)
4. range_r = 2 * maximum possible r value in image + 1
5. binsize_r = 1 pixel (usually)
6. size_r = ceil(range_r / binsize_r)
7. initialise 2D **accumulator array** *A* of size (size_r, size_θ) to all zeros.
8. for every edge point (*x*,*y*)
   a) for $\theta = -\pi$ to π
      i. compute r=*x*.cos(θ)+*y*.sin(θ)
      ii. *r_ind* ← array index corresponding to r
      iii. *θ_ind* ← array index corresponding to θ
      iv. increment *A*(*r_ind*, *θ_ind*) by 1 ← vote of point (*x*,*y*) for line (*r*,ϕ)
9. valid lines are local maxima of *A* and where *A* > threshold

Improvements

1. After edge detection, we already know the gradient direction at (x, y). So there is no need to iterate over all possible θ. Use the correct θ from the gradient direction.
2. Smooth the accumulator array A to account for uncertainties in the gradient direction.

Hough Transform for Circle Detection

- Analytic representation of circle of radius r centered at (a, b) is $(x-a)^2+(y-b)^2-r^2=0$.
- Hough space has 3 parameters (a, b, r).

Pseudocode

For every boundary point (x, y)

For every (a, b) in image plane

Compute $r(a,b)=\sqrt{(x-a)^2+(y-b)^2}$

Increment $A(a, b, r)$ by 1

A>threshold represents valid circles.

Hough Transform for Circle Detection

- If we know the gradient vector $\nabla I(x, y)$ at point (x, y), then we also know that the center (a, b) can only lie along this line.
- Hough space still has 3 parameters (a, b, r) but we search for r over a 1D space instead of a 2D plane.

Pseudocode

For every boundary point (x, y)

  For every (a, b) along gradient vector $\nabla I(x, y)$

    Compute $r(a, b) = \sqrt{(x-a)^2 + (y-b)^2}$

    Increment $A(a, b, r)$ by 1

A>threshold represents valid circles.

Concluding Points

- Hough space becomes very large $(\text{param}_1 \times \text{param}_2 \times \cdots \times \text{param}_N)$ when number of parameters N is increased.
- Using orientation information $\nabla I(x, y)$ in addition to positional information (x, y) leads to a smaller search space.