Hough Transform

600.658 - Seminar on Shape Analysis and Retrieval

Detection of arbitrary shapes

- Partial shape matching can also be viewed as detecting arbitrary shapes
- Hough transform is a method for estimating the parameters of a shape from its boundary points
- The idea can be generalized to estimate "parameters" of arbitrary shapes

Outline

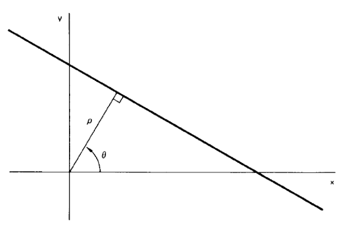
1. Hough Transform for Analytical Shapes
   - Voting in Parameter Space
   - Using Directional Information
   - Error Compensation: Smoothing
2. Generalizing to Non-Analytical Shapes

Outline

1. Hough Transform for Analytical Shapes
   - Voting in Parameter Space
   - Using Directional Information
   - Error Compensation: Smoothing

2. Generalizing to Non-Analytical Shapes

Straight line



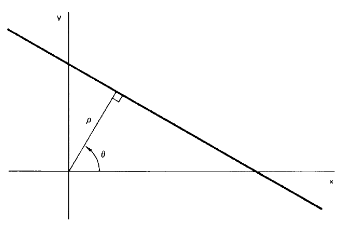
[2]

- Normal parameterization:
  $x \cos\theta + y \sin\theta = \rho$
- Points in picture $\leftrightarrow$ sinusoids in parameter space
- Points in parameter space $\leftrightarrow$ lines in picture
- Sinusoids corresponding to co-linear points intersect at an unique point

Example

Line: $0.6x + 0.4y = 2.4$
Sinusoids intersect at: $\rho = 2.4$, $\theta = 0.9273$

Straight line



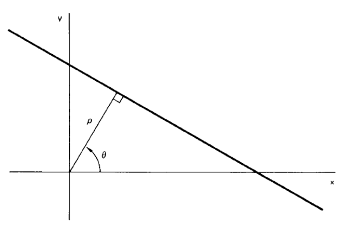
[2]

- Normal parameterization:
  $x \cos\theta + y \sin\theta = \rho$
- Points in picture $\leftrightarrow$ sinusoids in parameter space
- Points in parameter space $\leftrightarrow$ lines in picture
- Sinusoids corresponding to co-linear points intersect at an unique point

Example

Line: $0.6x + 0.4y = 2.4$
Sinusoids intersect at: $\rho = 2.4$, $\theta = 0.9273$

Straight line



[2]

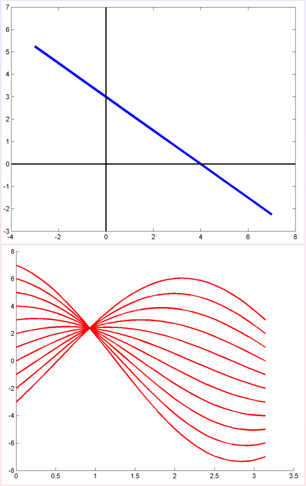
- Normal parameterization: $x \cos\theta + y \sin\theta = \rho$
- Points in picture $\leftrightarrow$ sinusoids in parameter space
- Points in parameter space $\leftrightarrow$ lines in picture
- Sinusoids corresponding to co-linear points intersect at an unique point

Example

Line: $0.6x + 0.4y = 2.4$

Sinusoids intersect at: $\rho = 2.4$, $\theta = 0.9273$

Straight line



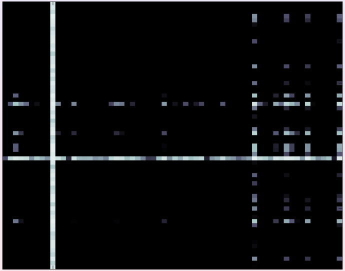
- Normal parameterization: $x \cos\theta + y \sin\theta = \rho$
- Points in picture $\leftrightarrow$ sinusoids in parameter space
- Points in parameter space $\leftrightarrow$ lines in picture
- Sinusoids corresponding to co-linear points intersect at an unique point

Example

Line: $0.6x + 0.4y = 2.4$
Sinusoids intersect at: $\rho = 2.4$, $\theta = 0.9273$

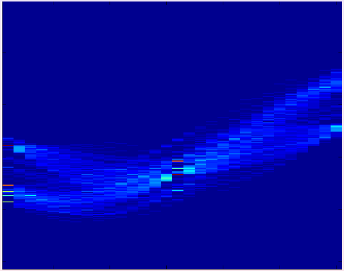
Outline

Quantize parameter space and vote into bins



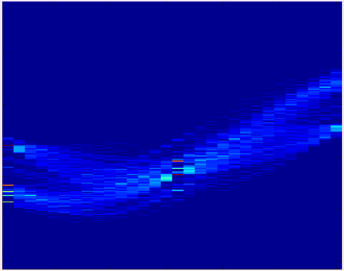
- Let $\rho \in [-R, R]$ and $\theta \in [0, \pi)$
- For each edge point (x_i, y_i), calculate: $\hat{\rho} = x_i \cos\hat{\theta} + y_i \sin\hat{\theta} \quad \forall\hat{\theta} \in [0, \pi)$
- Accumulator: $\mathcal{A}(\hat{\rho}, \hat{\theta}) = \mathcal{A}(\hat{\rho}, \hat{\theta}) + 1$
- Threshold the accumulator values to get parameters for detected lines
- Same general idea applies to other analytical shapes

Quantize parameter space and vote into bins



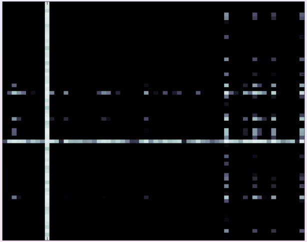
- Let $\rho \in [-R, R]$ and $\theta \in [0, \pi)$
- For each edge point (x_i, y_i), calculate: $\hat{\rho} = x_i \cos\hat{\theta} + y_i \sin\hat{\theta} \qquad \forall\hat{\theta} \in [0, \pi)$
- Accumulator: $\mathcal{A}(\hat{\rho}, \hat{\theta}) = \mathcal{A}(\hat{\rho}, \hat{\theta}) + 1$
- Threshold the accumulator values to get parameters for detected lines
- Same general idea applies to other analytical shapes

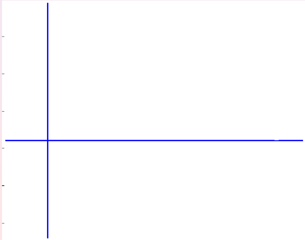
Quantize parameter space and vote into bins



- Let $\rho \in [-R, R]$ and $\theta \in [0, \pi)$
- For each edge point (x_i, y_i), calculate: $\hat{\rho} = x_i \cos\hat{\theta} + y_i \sin\hat{\theta} \qquad \forall\hat{\theta} \in [0, \pi)$
- Accumulator: $\mathcal{A}(\hat{\rho}, \hat{\theta}) = \mathcal{A}(\hat{\rho}, \hat{\theta}) + 1$
- Threshold the accumulator values to get parameters for detected lines
- Same general idea applies to other analytical shapes

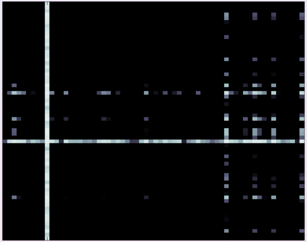
Quantize parameter space and vote into bins

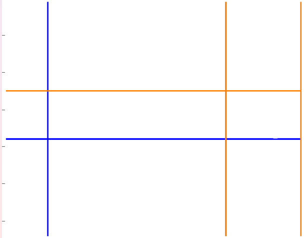




- Let $\rho \in [-R, R]$ and $\theta \in [0, \pi)$
- For each edge point (x_i, y_i), calculate: $\hat{\rho} = x_i \cos\hat{\theta} + y_i \sin\hat{\theta} \quad \forall\hat{\theta} \in [0, \pi)$
- Accumulator: $\mathcal{A}(\hat{\rho}, \hat{\theta}) = \mathcal{A}(\hat{\rho}, \hat{\theta}) + 1$
- Threshold the accumulator values to get parameters for detected lines
  - Threshold at $\mathcal{A}(\hat{\rho}, \hat{\theta}) = 30$
- Same general idea applies to other analytical shapes

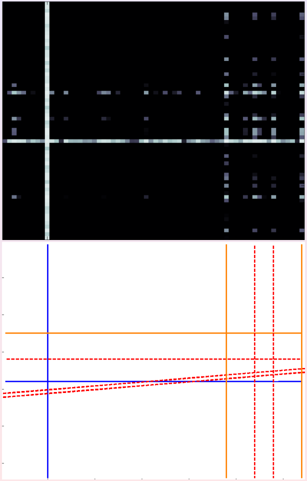
Quantize parameter space and vote into bins





- Let $\rho \in [-R, R]$ and $\theta \in [0, \pi)$
- For each edge point (x_i, y_i), calculate:
  $\hat{\rho} = x_i \cos \hat{\theta} + y_i \sin \hat{\theta} \qquad \forall \hat{\theta} \in [0, \pi)$
- Accumulator: $\mathcal{A}(\hat{\rho}, \hat{\theta}) = \mathcal{A}(\hat{\rho}, \hat{\theta}) + 1$
- Threshold the accumulator values to get parameters for detected lines
  - Threshold at $\mathcal{A}(\hat{\rho}, \hat{\theta}) = 20$
- Same general idea applies to other analytical shapes

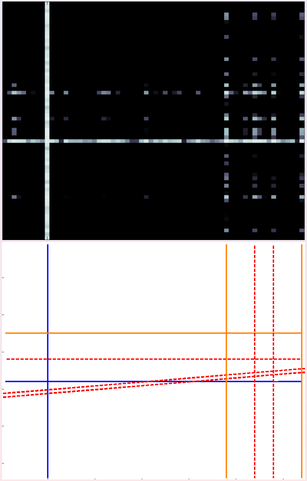
Quantize parameter space and vote into bins



- Let $\rho \in [-R, R]$ and $\theta \in [0, \pi)$
- For each edge point (x_i, y_i), calculate: $\hat{\rho} = x_i \cos \hat{\theta} + y_i \sin \hat{\theta} \qquad \forall \hat{\theta} \in [0, \pi)$
- Accumulator: $\mathcal{A}(\hat{\rho}, \hat{\theta}) = \mathcal{A}(\hat{\rho}, \hat{\theta}) + 1$
- Threshold the accumulator values to get parameters for detected lines
  - Threshold at $\mathcal{A}(\hat{\rho}, \hat{\theta}) = 15$
- Same general idea applies to other analytical shapes

- $\mathcal{O}(nd_\theta)$ computation, instead of $\mathcal{O}(n^2)$
- Can we do better?

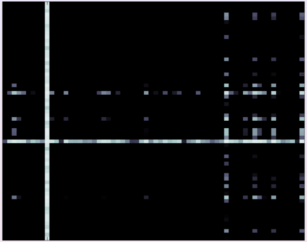
Quantize parameter space and vote into bins

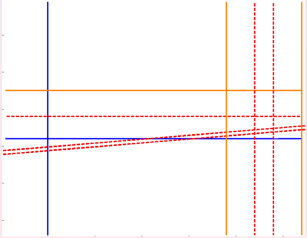


- Let $\rho \in [-R, R]$ and $\theta \in [0, \pi)$
- For each edge point (x_i, y_i), calculate: $\hat{\rho} = x_i \cos\hat{\theta} + y_i \sin\hat{\theta} \quad \forall\hat{\theta} \in [0, \pi)$
- Accumulator: $\mathcal{A}(\hat{\rho}, \hat{\theta}) = \mathcal{A}(\hat{\rho}, \hat{\theta}) + 1$
- Threshold the accumulator values to get parameters for detected lines
  - Threshold at $\mathcal{A}(\hat{\rho}, \hat{\theta}) = 15$
- Same general idea applies to other analytical shapes

- $O(nd_1)$ computation, instead of $O(n^2)$
- Can we do better?

Quantize parameter space and vote into bins

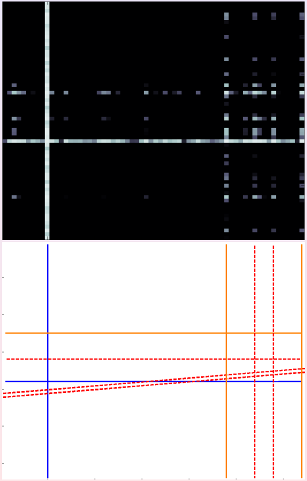




- Let $\rho \in [-R, R]$ and $\theta \in [0, \pi)$
- For each edge point (x_i, y_i), calculate:
  $\hat{\rho} = x_i \cos \hat{\theta} + y_i \sin \hat{\theta} \qquad \forall \hat{\theta} \in [0, \pi)$
- Accumulator: $\mathcal{A}(\hat{\rho}, \hat{\theta}) = \mathcal{A}(\hat{\rho}, \hat{\theta}) + 1$
- Threshold the accumulator values to get parameters for detected lines
  - Threshold at $\mathcal{A}(\hat{\rho}, \hat{\theta}) = 15$
- Same general idea applies to other analytical shapes

- $\mathcal{O}(nd_1)$ computation, instead of $\mathcal{O}(n^2)$
- Can we do better?

Quantize parameter space and vote into bins



- Let $\rho \in [-R, R]$ and $\theta \in [0, \pi)$
- For each edge point (x_i, y_i), calculate: $\hat{\rho} = x_i \cos\hat{\theta} + y_i \sin\hat{\theta} \qquad \forall\hat{\theta} \in [0, \pi)$
- Accumulator: $\mathcal{A}(\hat{\rho}, \hat{\theta}) = \mathcal{A}(\hat{\rho}, \hat{\theta}) + 1$
- Threshold the accumulator values to get parameters for detected lines
  - Threshold at $\mathcal{A}(\hat{\rho}, \hat{\theta}) = 15$
- Same general idea applies to other analytical shapes

- $\mathcal{O}(nd_1)$ computation, instead of $\mathcal{O}(n^2)$
- Can we do better?

Outline

1. Hough Transform for Analytical Shapes
   - Voting in Parameter Space
   - Using Directional Information
   - Error Compensation: Smoothing

2. Generalizing to Non-Analytical Shapes

Use what you have already got!

- More parameters ⇒ More computation and storage
- Gradient information reduces one more free parameter
- For circle: center lies r units along the gradient
- Rotation affects the gradient direction only

- $f(\mathbf{x}, \mathbf{p}) = (x-a)^2 + (y-b)^2 - r^2 = 0 \qquad \mathbf{p} = (a, b, r), \mathbf{x} = (x, y)$
- $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \frac{\pi}{2}\right] \qquad \phi(\mathbf{x})$ is the gradient direction
- Update $\mathcal{A}(\mathbf{p})$ if $f(\mathbf{x}, \mathbf{p}) = 0$ and $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \theta - \frac{\pi}{2}\right]$

Use what you have already got!

- More parameters $\Rightarrow$ More computation and storage
- Gradient information reduces one more free parameter
- For circle: center lies r units along the gradient
- Rotation affects the gradient direction only

- $f(\mathbf{x}, \mathbf{p}) = (x-a)^2 + (y-b)^2 - r^2 = 0 \qquad \mathbf{p} = (a, b, r)$, $\mathbf{x} = (x, y)$
- $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \frac{\pi}{2}\right] \qquad \phi(\mathbf{x})$ is the gradient direction
- Update $\mathcal{A}(\mathbf{p})$ if $f(\mathbf{x}, \mathbf{p}) = 0$ and $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \theta - \frac{\pi}{2}\right]$

Use what you have already got!

- More parameters $\Rightarrow$ More computation and storage
- Gradient information reduces one more free parameter
- For circle: center lies r units along the gradient
- Rotation affects the gradient direction only

- $f(\mathbf{x}, \mathbf{p}) = (x-a)^2 + (y-b)^2 - r^2 = 0 \qquad \mathbf{p} = (a, b, r)$, $\mathbf{x} = (x, y)$
- $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \frac{\pi}{2}\right] \qquad \phi(\mathbf{x})$ is the gradient direction
- Update $\mathcal{A}(\mathbf{p})$ if $f(\mathbf{x}, \mathbf{p}) = 0$ and $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \theta - \frac{\pi}{2}\right]$

Use what you have already got!

- More parameters $\Rightarrow$ More computation and storage
- Gradient information reduces one more free parameter
- For circle: center lies r units along the gradient
- Rotation affects the gradient direction only

- $f(\mathbf{x}, \mathbf{p}) = (x-a)^2 + (y-b)^2 - r^2 = 0 \qquad \mathbf{p} = (a, b, r)$, $\mathbf{x} = (x, y)$
- $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \frac{\pi}{2}\right] \qquad \phi(\mathbf{x})$ is the gradient direction
- Update $\mathcal{A}(\mathbf{p})$ if $f(\mathbf{x}, \mathbf{p}) = 0$ and $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \theta - \frac{\pi}{2}\right]$

Use what you have already got!

- More parameters $\Rightarrow$ More computation and storage
- Gradient information reduces one more free parameter
- For circle: center lies r units along the gradient
- Rotation affects the gradient direction only

- $f(\mathbf{x}, \mathbf{p}) = (x-a)^2 + (y-b)^2 - r^2 = 0 \qquad \mathbf{p} = (a, b, r)$, $\mathbf{x} = (x, y)$
- $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \frac{\pi}{2}\right] \qquad \phi(\mathbf{x})$ is the gradient direction
- Update $\mathcal{A}(\mathbf{p})$ if $f(\mathbf{x}, \mathbf{p}) = 0$ and $\frac{\mathrm{d}f}{\mathrm{d}x}(\mathbf{x}, \mathbf{p}) = 0$
- $\frac{\mathrm{d}y}{\mathrm{d}x} = \tan\left[\phi(\mathbf{x}) - \theta - \frac{\pi}{2}\right]$

Outline

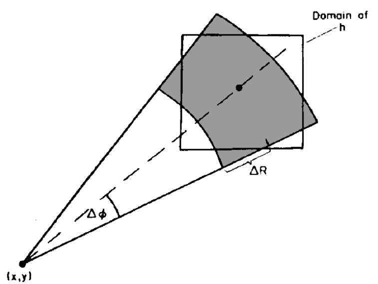
1. Hough Transform for Analytical Shapes
   - Voting in Parameter Space
   - Using Directional Information
   - Error Compensation: Smoothing

2. Generalizing to Non-Analytical Shapes

Compensating for errors

- Errors can cause $\mathcal{A}(\mathbf{p}')$ to be incremented, where $\mathbf{p}'$ is close to the actual parameter $\mathbf{p}$
- Compensate for the uncertainty of measurement in parameter space
- Smooth the accumulator by incrementing counts of nearby cells accroding to some point-spread function h
- Equivalent to convolving $\mathcal{A} * h$

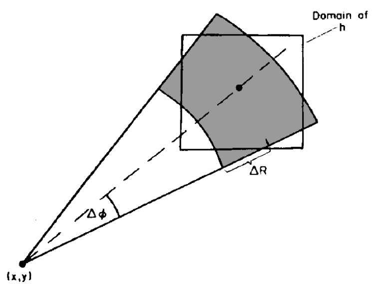
Compensating for errors



[1]

- Errors can cause $\mathcal{A}(\mathbf{p}')$ to be incremented, where $\mathbf{p}'$ is close to the actual parameter $\mathbf{p}$
- Compensate for the uncertainty of measurement in parameter space
- Smooth the accumulator by incrementing counts of nearby cells accroding to some point-spread function h
- Equivalent to convolving $\mathcal{A} * h$

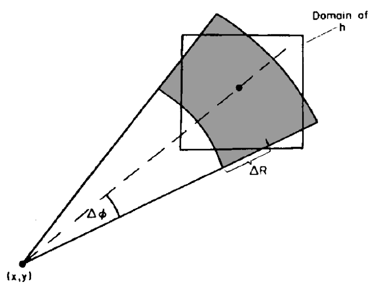
Compensating for errors



[1]

- Errors can cause $\mathcal{A}(\mathbf{p}')$ to be incremented, where $\mathbf{p}'$ is close to the actual parameter $\mathbf{p}$
- Compensate for the uncertainty of measurement in parameter space
- Smooth the accumulator by incrementing counts of nearby cells accroding to some point-spread function h
- Equivalent to convolving $\mathcal{A} * h$

Compensating for errors



[1]

- Errors can cause $\mathcal{A}(\mathbf{p}')$ to be incremented, where $\mathbf{p}'$ is close to the actual parameter $\mathbf{p}$
- Compensate for the uncertainty of measurement in parameter space
- Smooth the accumulator by incrementing counts of nearby cells accroding to some point-spread function h
- Equivalent to convolving $\mathcal{A} * h$

Outline

The R-table

- Any shape is specified by the set of boundary points $B = \{\mathbf{x}_B\}$
- For a shape, define $\mathbf{p}$ as: $\mathbf{p} = \{\mathbf{x}^0, s, \theta\}$
- For each $\mathbf{x}_B$, compute $\mathbf{r} = \mathbf{x}^0 - \mathbf{x}_B$, and store as function of ϕ
- For each edge pixel $\mathbf{x}$ (with gradient direction $\phi(\mathbf{x})$) in an image, obtain $\mathbf{r}$ from the table and update $A(\mathbf{x} + \mathbf{r})$

i	ϕ_i	R_{ϕ_i}
0	0	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B, \mathbf{x}_B \in B, \phi(\mathbf{x}_B) = 0\}$
1	$\Delta\phi$	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B, \mathbf{x}_B \in B, \phi(\mathbf{x}_B) = \Delta\phi\}$
2	$2\Delta\phi$	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B, \mathbf{x}_B \in B, \phi(\mathbf{x}_B) = 2\Delta\phi\}$
...	...	...

The R-table

- Any shape is specified by the set of boundary points $B = \{\mathbf{x}_B\}$
- For a shape, define $\mathbf{p}$ as: $\mathbf{p} = \{\mathbf{x}^0, s, \theta\}$
- For each $\mathbf{x}_B$, compute $\mathbf{r} = \mathbf{x}^0 - \mathbf{x}_B$, and store as function of ϕ
- For each edge pixel $\mathbf{x}$ (with gradient direction $\phi(\mathbf{x})$) in an image, obtain $\mathbf{r}$ from the table and update $\mathcal{A}(\mathbf{x} + \mathbf{r})$

i	ϕ_i	R_{ϕ_i}
0	0	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B, \mathbf{x}_B \in B, \phi(\mathbf{x}_B) = 0\}$
1	$\Delta\phi$	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B, \mathbf{x}_B \in B, \phi(\mathbf{x}_B) = \Delta\phi\}$
2	$2\Delta\phi$	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B, \mathbf{x}_B \in B, \phi(\mathbf{x}_B) = 2\Delta\phi\}$
...	...	...

The R-table

- Any shape is specified by the set of boundary points $B = \{\mathbf{x}_B\}$
- For a shape, define $\mathbf{p}$ as: $\mathbf{p} = \{\mathbf{x}^0, s, \theta\}$
- For each $\mathbf{x}_B$, compute $\mathbf{r} = \mathbf{x}^0 - \mathbf{x}_B$, and store as function of ϕ
- For each edge pixel $\mathbf{x}$ (with gradient direction $\phi(\mathbf{x})$) in an image, obtain $\mathbf{r}$ from the table and update $\mathcal{A}(\mathbf{x} + \mathbf{r})$

i	ϕ_i	R_{ϕ_i}
0	0	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B,\ \mathbf{x}_B \in B,\ \phi(\mathbf{x}_B) = 0\}$
1	$\Delta\phi$	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B,\ \mathbf{x}_B \in B,\ \phi(\mathbf{x}_B) = \Delta\phi\}$
2	$2\Delta\phi$	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B,\ \mathbf{x}_B \in B,\ \phi(\mathbf{x}_B) = 2\Delta\phi\}$
...	...	...

The R-table

- Any shape is specified by the set of boundary points $B = \{\mathbf{x}_B\}$
- For a shape, define **p** as: $\mathbf{p} = \{\mathbf{x}^0, s, \theta\}$
- For each $\mathbf{x}_B$, compute $\mathbf{r} = \mathbf{x}^0 - \mathbf{x}_B$, and store as function of ϕ
- For each edge pixel $\mathbf{x}$ (with gradient direction $\phi(\mathbf{x})$) in an image, obtain $\mathbf{r}$ from the table and update $\mathcal{A}(\mathbf{x} + \mathbf{r})$

i	ϕ_i	R_{ϕ_i}
0	0	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B,\ \mathbf{x}_B \in B,\ \phi(\mathbf{x}_B) = 0\}$
1	$\Delta\phi$	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B,\ \mathbf{x}_B \in B,\ \phi(\mathbf{x}_B) = \Delta\phi\}$
2	$2\Delta\phi$	$\{\mathbf{r} \mid \mathbf{x}^0 - \mathbf{r} = \mathbf{x}_B,\ \mathbf{x}_B \in B,\ \phi(\mathbf{x}_B) = 2\Delta\phi\}$
...	...	...

Transformations of the R-table

- Let $R(\phi)$ be the R-table for a shape S
- Scale: $T_s[R(\phi)] = sR(\phi)$
  - Scale all the vectors $\mathbf{r}$ by s
- Rotation: $T_\theta[R(\phi)] = \mathrm{Rot}\{R[(\phi - \theta)\mathrm{mod} 2\pi], \theta\}$
  - Change the indices ϕ to $(\phi - \theta)\mathrm{mod} 2\pi$, find corresponding vectors $\mathbf{r}$ and rotate them by θ

Hough transform for composite shapes

- Let a shape S have two subparts S_1 and S_2 with respective reference points $\mathbf{x}^0$, $\mathbf{x}_1^0$, and $\mathbf{x}_2^0$
- Compute $\mathbf{r}_1 = \mathbf{x}^0 - \mathbf{x}_1^0$ and $\mathbf{r}_2 = \mathbf{x}^0 - \mathbf{x}_2^0$
- R-table for composite shape:
  $R_s(\phi) = [R_{s_1}(\phi) + \mathbf{r}_1]\dot{\cup}[R_{s_2}(\phi) + \mathbf{r}_2]$

Hough transform for composite shapes

- Let a shape S have two subparts S_1 and S_2 with respective reference points $\mathbf{x}^0$, $\mathbf{x}_1^0$, and $\mathbf{x}_2^0$
- Compute $\mathbf{r}_1 = \mathbf{x}^0 - \mathbf{x}_1^0$ and $\mathbf{r}_2 = \mathbf{x}^0 - \mathbf{x}_2^0$
- R-table for composite shape:
  $R_s(\phi) = [R_{s_1}(\phi) + \mathbf{r}_1] \dot{\cup} [R_{s_2}(\phi) + \mathbf{r}_2]$

Hough transform for composite shapes

- Let a shape S have two subparts S_1 and S_2 with respective reference points $\mathbf{x}^0$, $\mathbf{x}_1^0$, and $\mathbf{x}_2^0$
- Compute $\mathbf{r}_1 = \mathbf{x}^0 - \mathbf{x}_1^0$ and $\mathbf{r}_2 = \mathbf{x}^0 - \mathbf{x}_2^0$
- R-table for composite shape:
  $R_s(\phi) = [R_{s_1}(\phi) + \mathbf{r}_1]\dot{\cup}[R_{s_2}(\phi) + \mathbf{r}_2]$

Incrementation Strategies

- Increment the accumulator by a value depending on the gradient: $\mathcal{A}(\mathbf{p}) = \mathcal{A}(\mathbf{p}) + g(\nabla \mathbf{x})$
- Increment by larger values if neighbouring points are incrementing the same reference point
- Try to find progressively longer connected segments using dynamic programming
- Weigh different parts of a composite object differently

References

D. H. Ballard
Generalizing the Hough Transform to Detect Arbitrary Shapes
Pattern Recognition, **13**(2):111-122, 1981.

R. O. Duda and P. E. Hart
Use of the Hough Transformation to Detect Lines and Curves in Pictures
Communications of Association for Computing Machinery, **15**(1):11-15, 1972.