Generalized Hough Transform

16-385 Computer Vision
Hough Circles
Finding Circles by Hough Transform

Equation of Circle:

\[(x_i - a)^2 + (y_i - b)^2 = r^2\]

If radius is known: (2D Hough Space)

Accumulator Array \(A(a, b)\)
\[(x - a)^2 + (y - b)^2 = r^2\]

Parameters

Variables

Image space

Parameter space
\[(x - a)^2 + (y - b)^2 = r^2\]
\[(x - a)^2 + (y - b)^2 = r^2\]
\((x - a)^2 + (y - b)^2 = r^2\)
\[(x - a)^2 + (y - b)^2 = r^2\]
Finding Circles by Hough Transform

Equation of Circle:

\[(x_i - a)^2 + (y_i - b)^2 = r^2\]

If radius is not known: 3D Hough Space!

Use Accumulator array \(A(a, b, r)\)

What is the surface in the hough space?
Finding Circles by Hough Transform

Equation of Circle:

\[(x_i - a)^2 + (y_i - b)^2 = r^2\]

If radius is not known: 3D Hough Space!

Use Accumulator array \( A(a, b, r) \)

What is the surface in the hough space?
Using Gradient Information

Gradient information can save lot of computation:

Edge Location \((x_i, y_i)\)

Edge Direction \(\phi_i\)

Assume radius is known:

\[
\begin{align*}
    a &= x - r \cos \phi \\
    b &= y - r \sin \phi
\end{align*}
\]

Need to increment only one point in accumulator!!
\[(x - a)^2 + (y - b)^2 = r^2\]
\[(x - a)^2 + (y - b)^2 = r^2\]
Pennie Hough detector

Quarter Hough detector
Pennie Hough detector

Quarter Hough detector
Generalized Hough Transform
<table>
<thead>
<tr>
<th>Edge Direction</th>
<th>$\bar{\tau} = (\tau, \kappa)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_1$</td>
<td>$\bar{\tau}_1, \bar{\tau}_2, \bar{\tau}_3$</td>
</tr>
<tr>
<td>$\phi_2$</td>
<td>$\bar{\tau}_1, \bar{\tau}_2$</td>
</tr>
<tr>
<td>$\phi_i$</td>
<td>$\bar{\tau}_1, \bar{\tau}_2$</td>
</tr>
<tr>
<td>$\phi_n$</td>
<td>$\bar{\tau}_1, \bar{\tau}_2$</td>
</tr>
</tbody>
</table>
Generalized Hough Transform

Find Object Center \((x_c, y_c)\) given edges \((x_i, y_i, \phi_i)\)

Create Accumulator Array \(A(x_c, y_c)\)

Initialize: \(A(x_c, y_c) = 0\) \(\forall (x_c, y_c)\)

For each edge point \((x_i, y_i, \phi_i)\)

For each entry \(r_k^i\) in table, compute:

\[
    x_c = x_i + r_k^i \cos \alpha_k^i
\]

\[
    y_c = y_i + r_k^i \sin \alpha_k^i
\]

Increment Accumulator: \(A(x_c, y_c) = A(x_c, y_c) + 1\)

Find Local Maxima in \(A(x_c, y_c)\)
Scale & Rotation:

Use Accumulator Array:

\[ A[x_c, y_c, s, \theta] \]

\[ \text{scale: } s \]

\[ \text{Rotation: } \theta \]

Use:

\[ x_c = x_i + r_k^i s \cos (\alpha_k^i + \theta) \]

\[ y_c = y_i + r_k^i s \sin (\alpha_k^i + \theta) \]

\[ A(x_c, y_c, s, \theta) = A(x_c, y_c, s, \theta) + 1. \]
A. Train phase:
   1. Get features
   2. Store all displacements of feature from center

B. Test phase:
   1. Get features & lookup displacements
   2. Vote for center location
A. Train phase:
   1. Get features
   2. Store all displacements of feature from center

B. Test phase:
   1. Get features & lookup displacements
   2. Vote for center location
A. Train phase:
1. Get features
2. Store all displacements of feature from center

B. Test phase:
1. Get features & lookup displacements
2. Vote for center location

Test image
Application of Hough Transforms
Detecting shape features

F. Jurie and C. Schmid, Scale-invariant shape features for recognition of object categories, CVPR 2004
Original images

Laplacian circles

Hough-like circles

Which feature detector is more consistent?
Robustness to scale and clutter
Object detection

Index displacements by “visual codeword”

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004
Train phase

1. get features
Train phase

2. store displacements
Test phase
The Hough transform ...

Deals with occlusion well?  
Detects multiple instances?  
Robust to noise?  
Good computational complexity?  
Easy to set parameters?