Lecture 9 Finding More Shapes

COMP6223 Computer Vision (MSc)

How can we go from conic sections to general shapes?

Book
pp
215-246

Department of
Electronics and
Computer Science

UNIVERSITY OF
Southampton
School of Electronics
and Computer Science

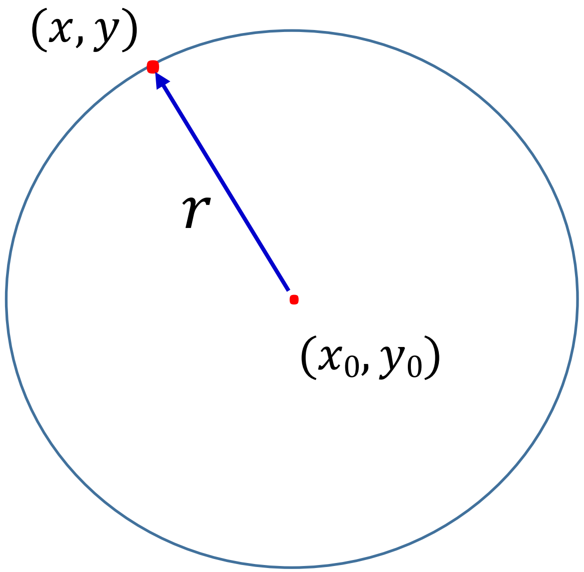
Content

1. What more versions of the Hough transform are possible?
2. What are its limits?
3. Can it be used to detect shapes that are not given by an equation?

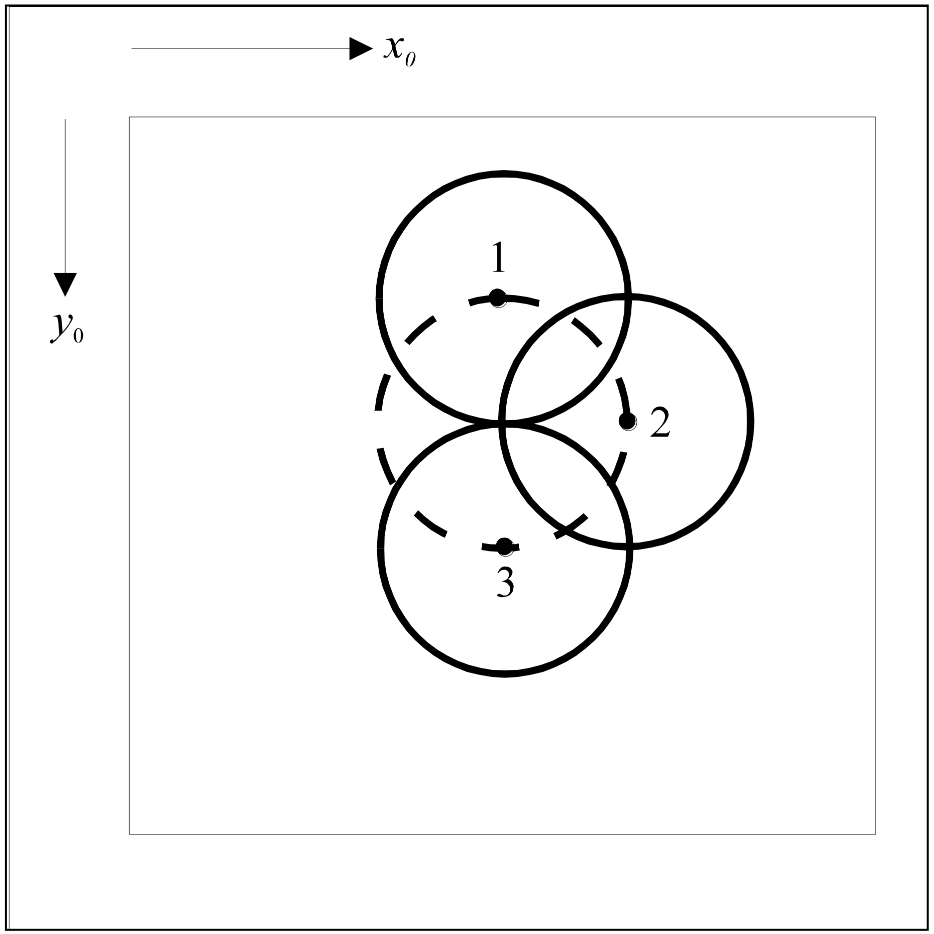
Hough Transform for Circles

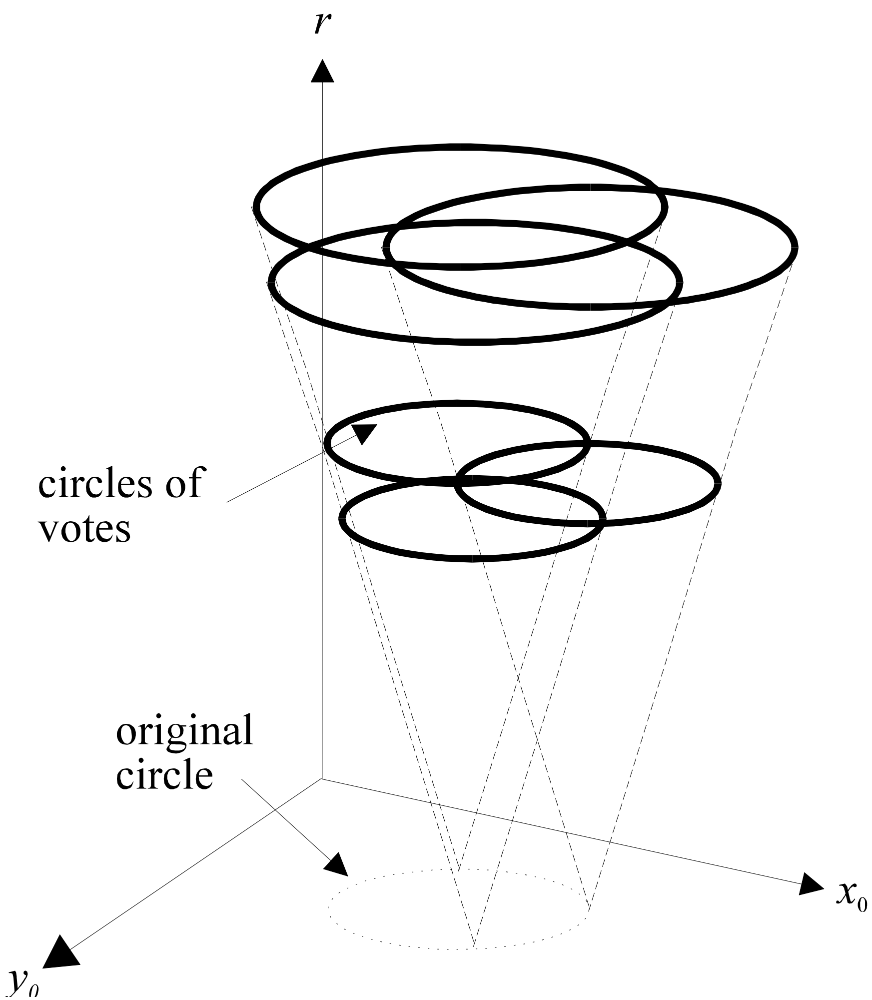
Again, it's duality: $$(x - x_0)^2 + (y - y_0)^2 = r^2$$

Points:	x, y	centre:	x_0, y_0	radius:	r
"	x_0, y_0	"	x, y	"	r



Circle Voting and Accumulator Space

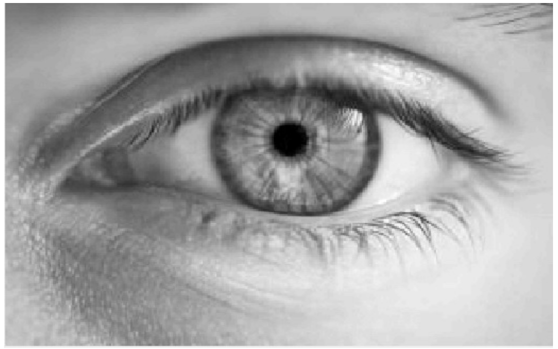




Pseudocode

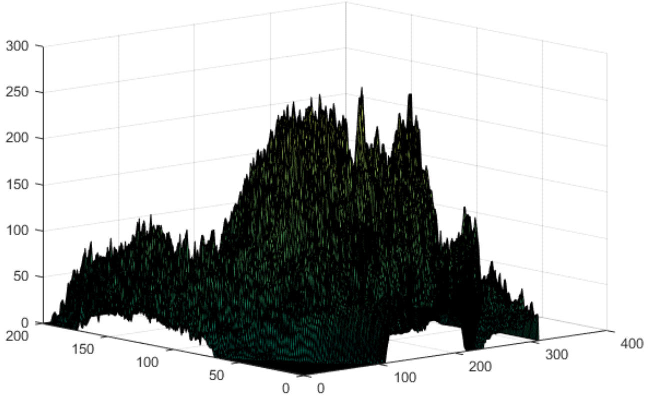
```
accum=0
for all x,y                                     !look at all points
    if edge(y,x)>threshold                      !check significance
        for r = min_r, max_r                    !do values of radius
            for theta = 0, 2*pi                 !go around a circle
                x0=x+r*cos(theta)               !generate x
                y0=y+r*sin(theta)               !generate y
                accum(y0, x0, r ) PLUS 1            !vote in accumulator
y0, x0, r= argmax(accum)                        !peak gives parameters
```

Applying the HT for circles

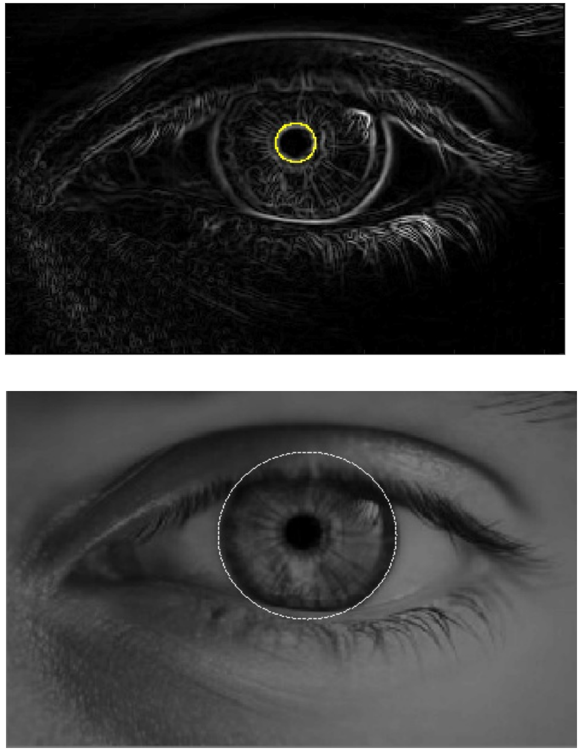


image

(Sobel) edges

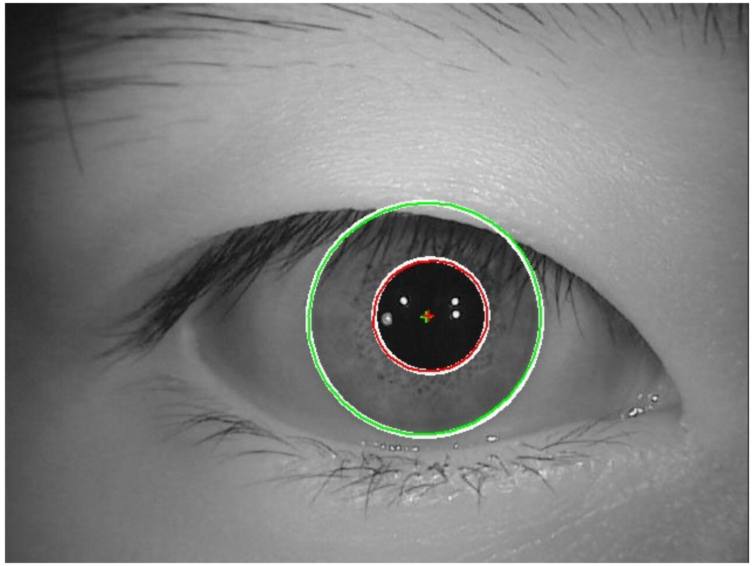


accumulator



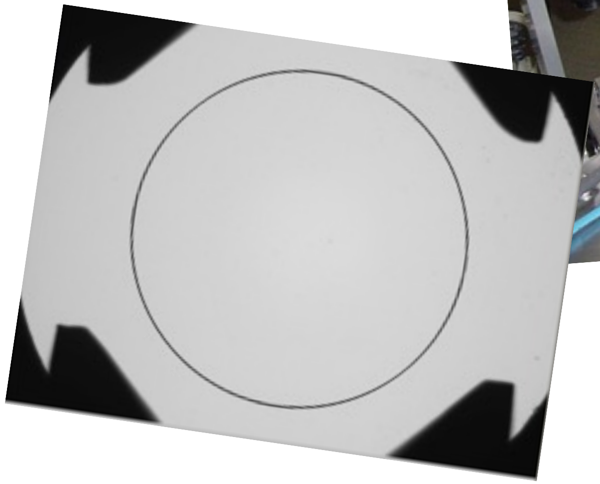
small and large circles

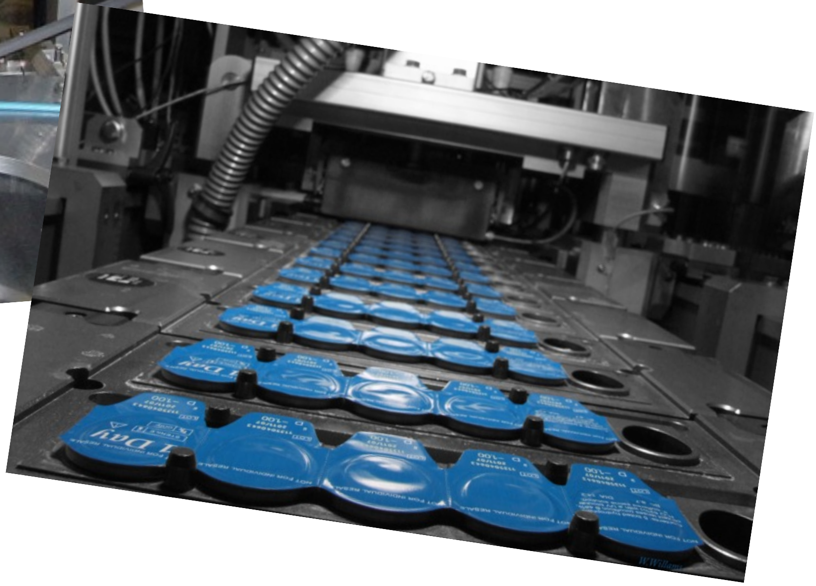
Integrodifferential operator?



https://stackoverflow.com/questions/27058057/comparing-irises-images-with-opencv

Contact lenses





Extensions to conic sections

Ellipse

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$$

described by 4 parameters (i.e., x_0, y_0, a, b).

If each has 100 values, accumulator size:

$$10^2 \times 10^2 \times 10^2 \times 10^2 = 10^8 = 0.1\text{GB}$$

Add rotation, that's **10GB** …. Ouch!

Motivates approaches to save memory and improve speed (since result is optimal)

Speeding it up.....

Use the circular shape as an example.

It‘s a 3D accumulator

Differentiating $(x - x_0)^2 + (y - y_0)^2 = r^2$ $\Rightarrow$ $\frac{dy}{dx} = -\frac{(x - x_0)}{(y - y_0)}$

$$\left(\frac{dy}{dx}\right)^2 (y - y_0)^2 + (y - y_0)^2 = r^2$$

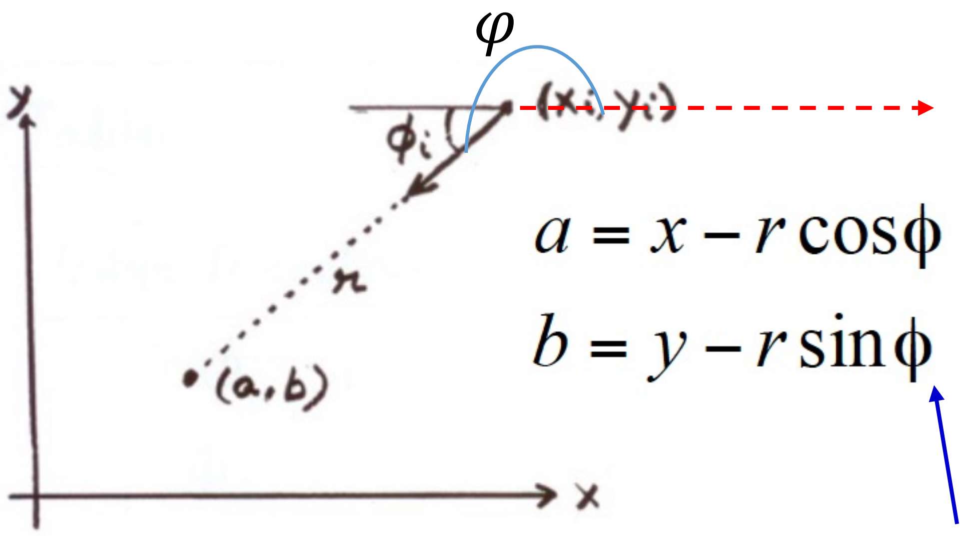
$$y - y_0 = \frac{r}{\sqrt{1 + \left(\frac{dy}{dx}\right)^2}}$$

2D accumulator

This is the edge direction

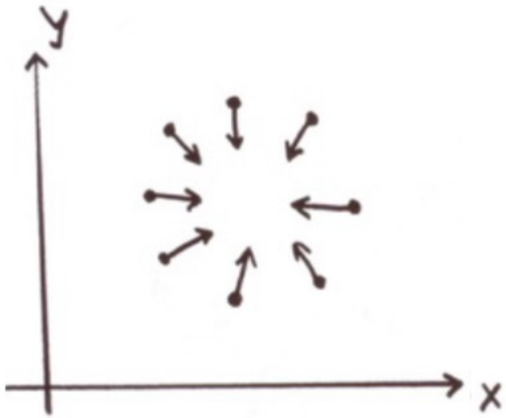
Fireside

$$a = x + r\cos\varphi$$
$$b = y + r\sin\varphi$$

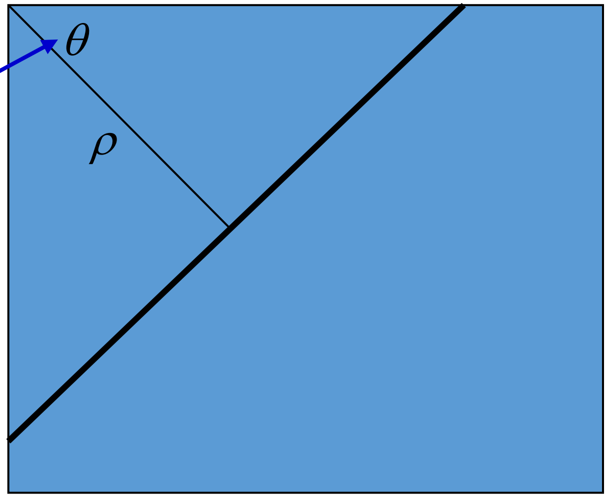


$$a = x - r\cos\phi$$
$$b = y - r\sin\phi$$

Edge direction can be used to compute ϕ, φ and θ to speed up the Hough transform speed.



Circle

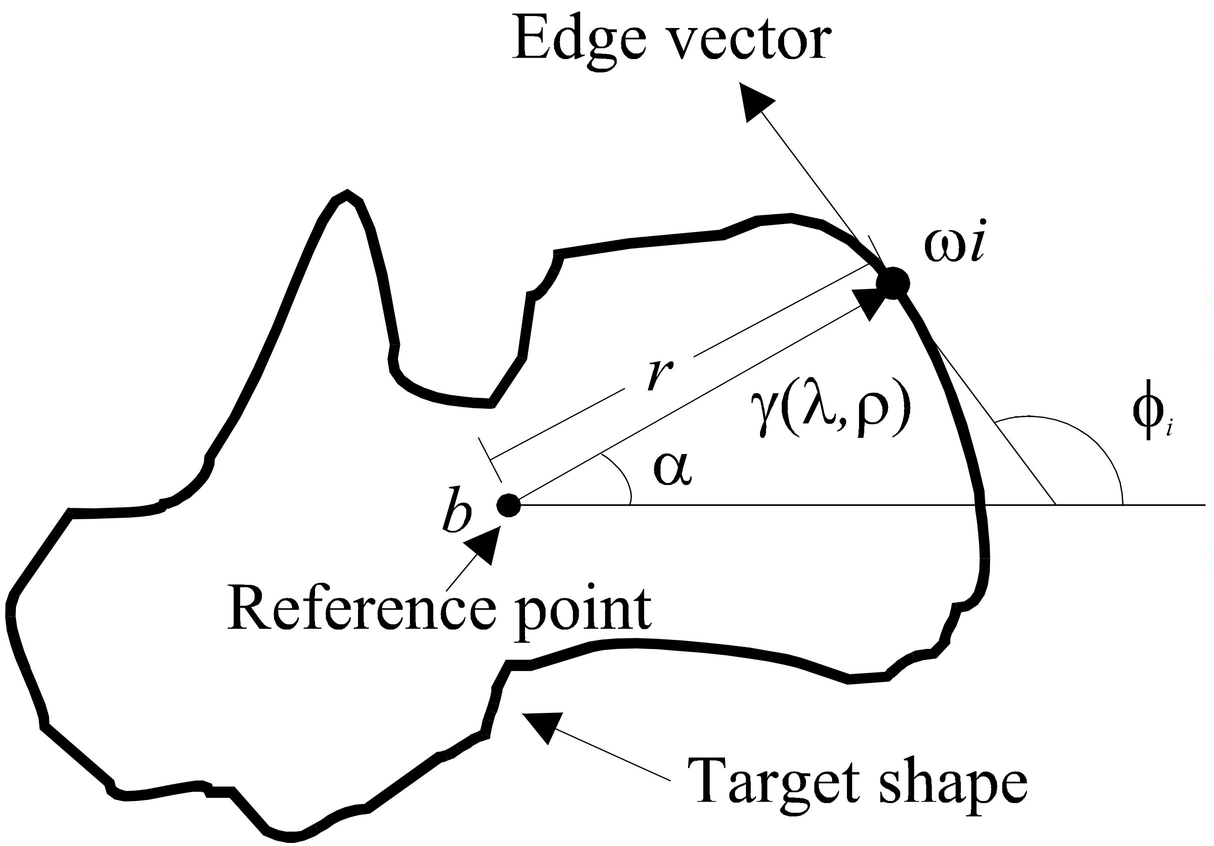


Line $\rho = x\cos\theta + y\sin\theta$

Arbitrary Shapes

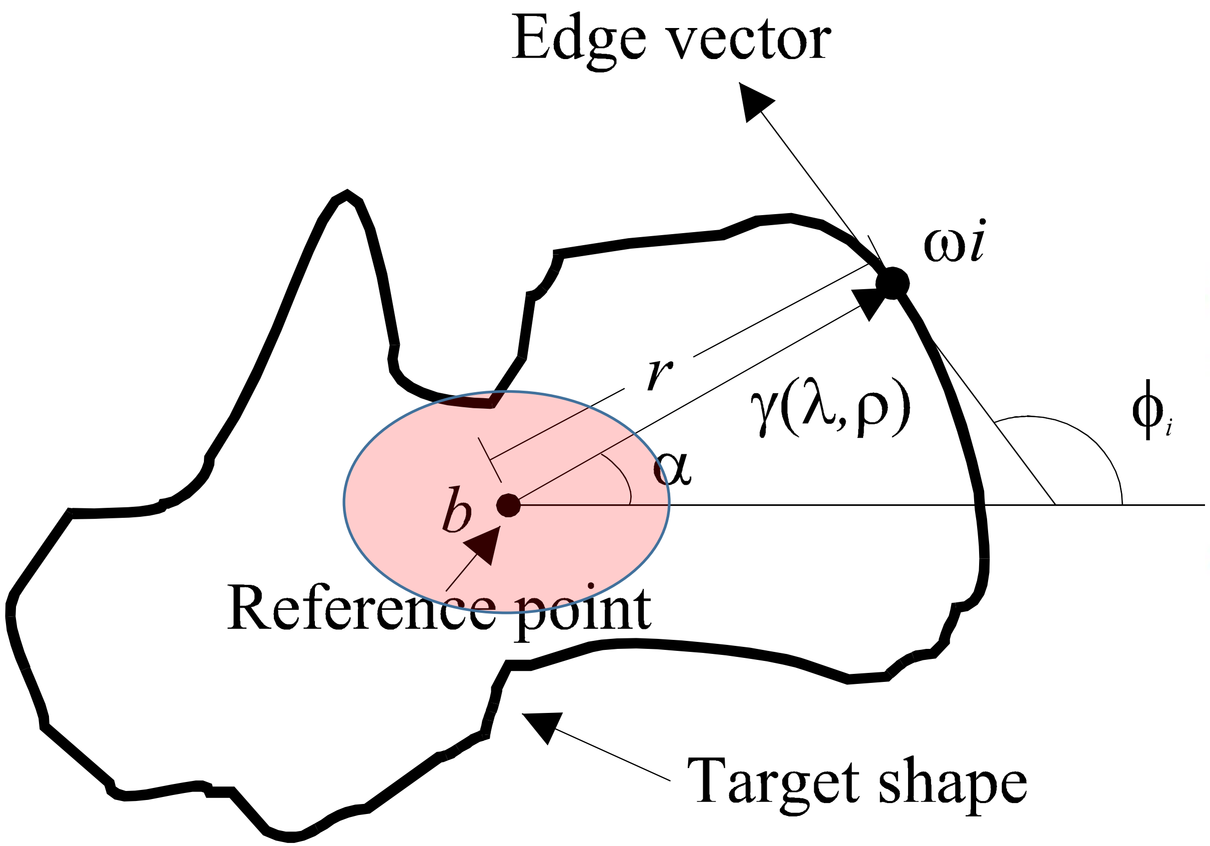
- Use Generalised Hough transform
- Form (discrete) look-up-table (R-table)
- Vote via look-up-table

R-table Construction



$\hat{\varphi}'_i$	$\gamma = (r, \alpha)$
0	$(r_0, \alpha_0), (r_1, \alpha_1), (r_2, \alpha_2)$
$\Delta\phi$	$\vdots$
$2\Delta\phi$	$\vdots$
...	...

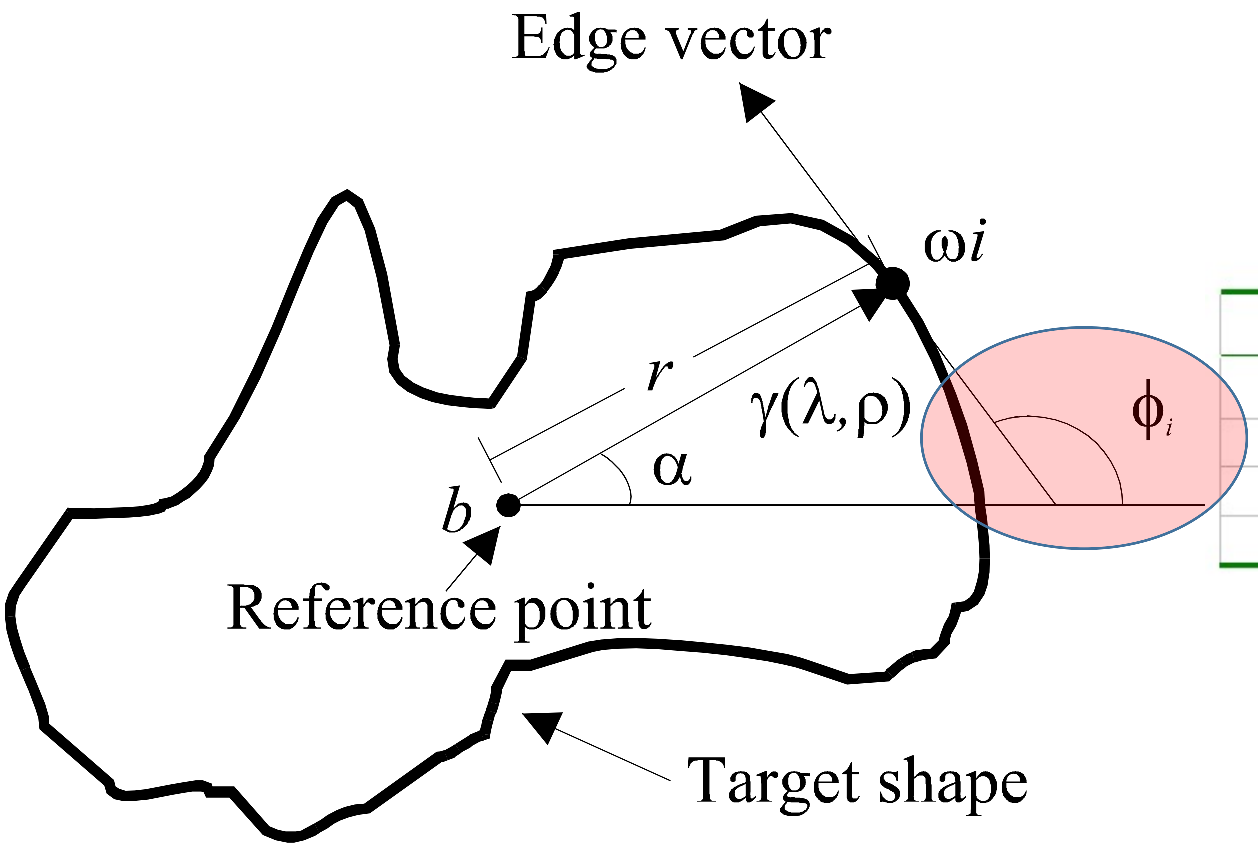
R-table Construction



$\hat{\varphi}'_i$	$\gamma = (r, \alpha)$
0	$(r_0, \alpha_0), (r_1, \alpha_1), (r_2, \alpha_2)$
$\Delta\phi$	$\vdots$
$2\Delta\phi$	$\vdots$
...	...

Need to start somewhere

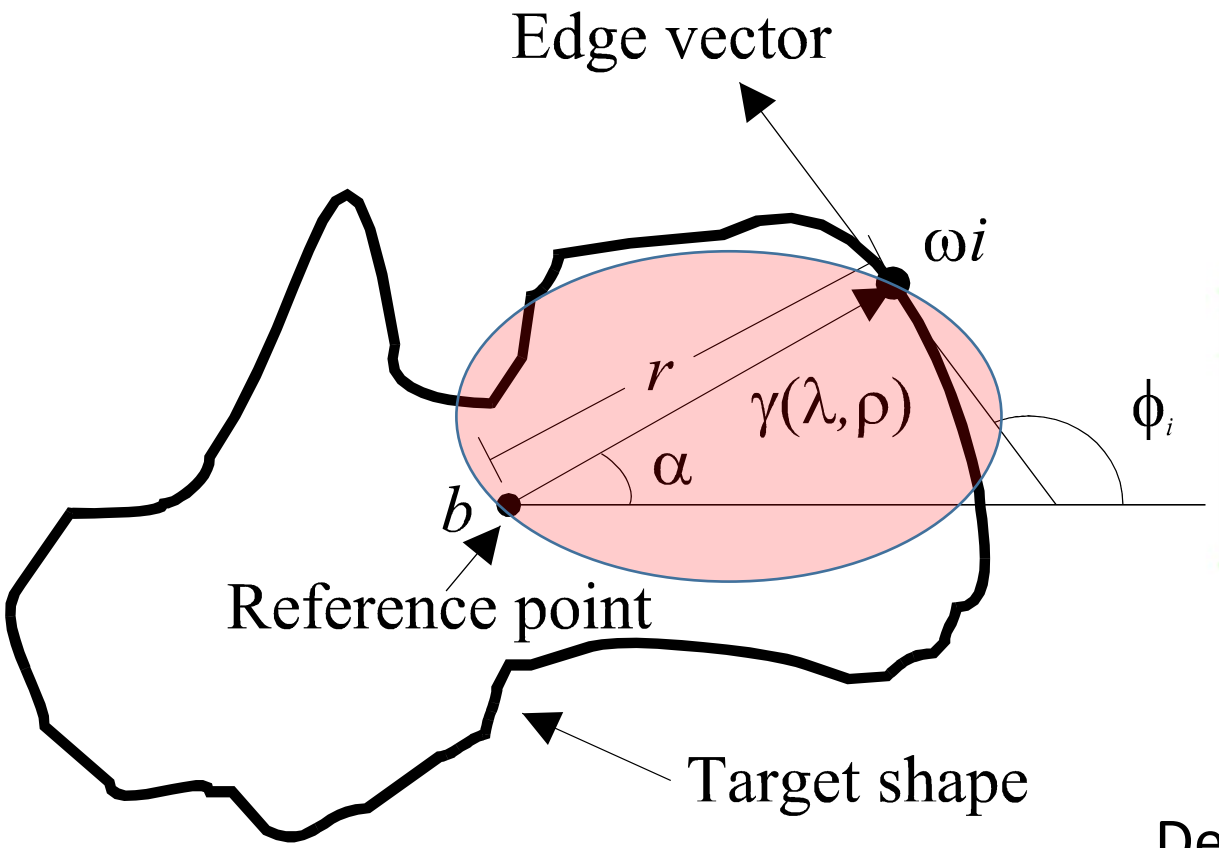
R-table Construction



$\hat{\varphi}'_i$	$\gamma=(r,\alpha)$
0	$(r_0,\alpha_0),(r_1,\alpha_1),(r_2,\alpha_2)$
$\Delta\phi$	$\vdots$
$2\Delta\phi$	$\vdots$
...	...

Measure edge direction

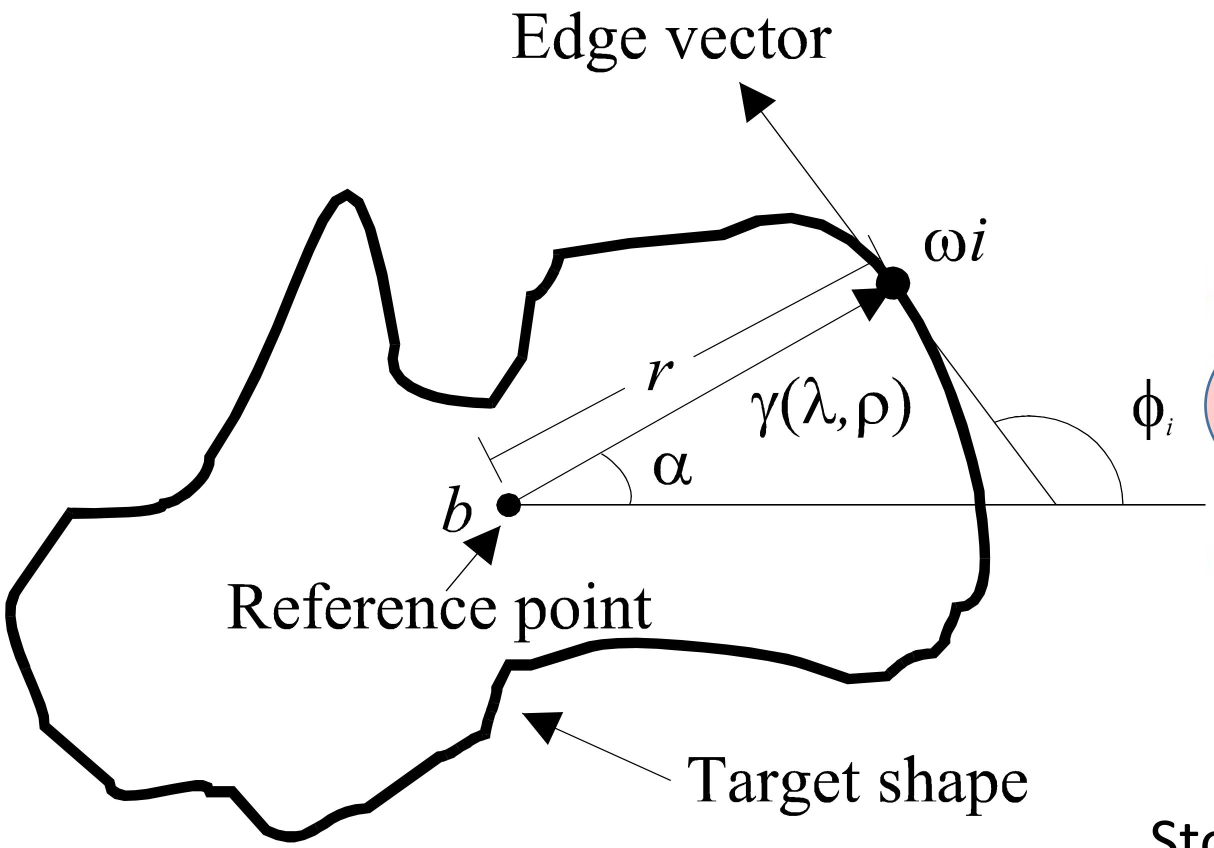
R-table Construction



$\hat{\varphi}'_i$	$\gamma = (r, \alpha)$
0	$(r_0, \alpha_0), (r_1, \alpha_1), (r_2, \alpha_2)$
$\Delta\phi$	⋮
$2\Delta\phi$	⋮
...	...

Determine length and direction to reference point

R-table Construction

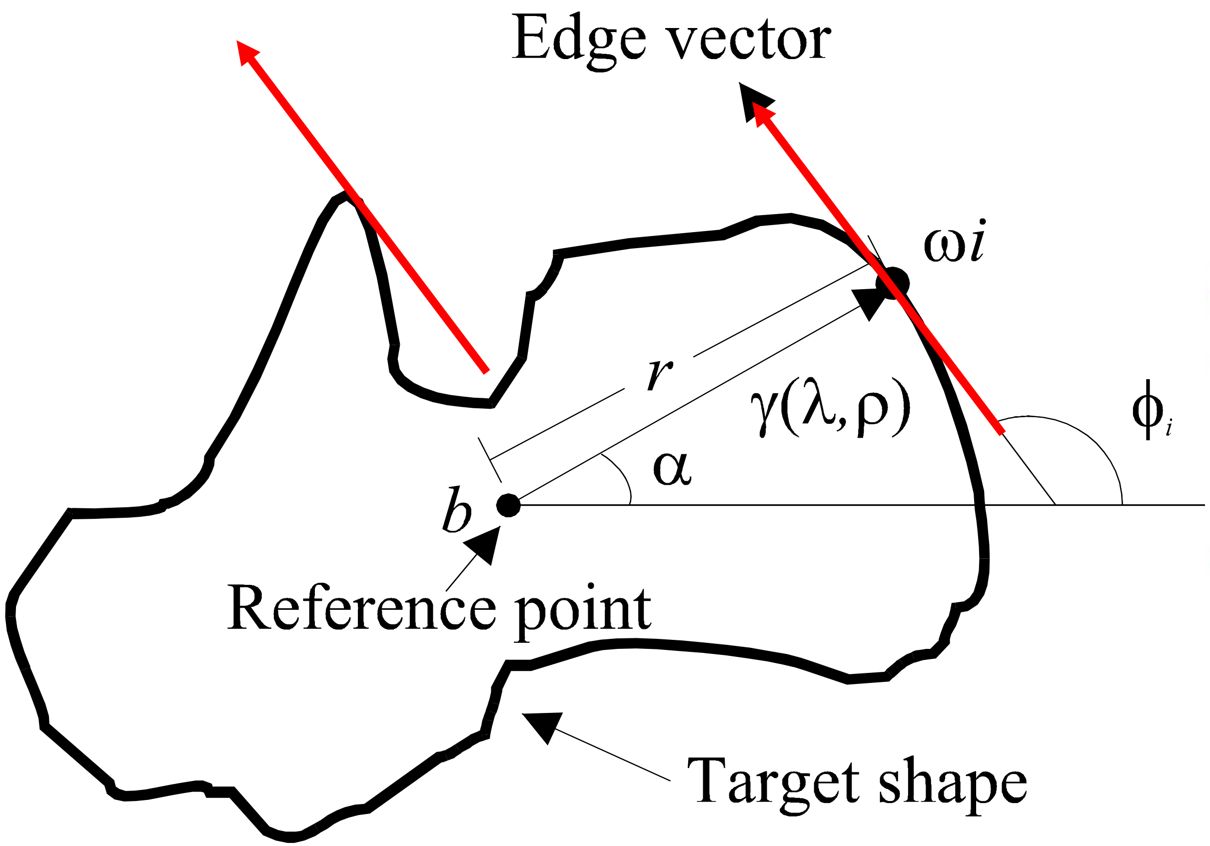


$\hat{\varphi}'_i$	$\gamma = (r, \alpha)$
0	$(r_0, \alpha_0), (r_1, \alpha_1), (r_2, \alpha_2)$
$\Delta\phi$	⋮
$2\Delta\phi$	⋮
…	…

Store length and direction indexed by edge direction

R-table Construction

$$x_c = x_i - r\cos(\alpha)$$
$$y_c = y_i - r\sin(\alpha)$$



$\hat{\varphi}'_i$	$\gamma=(r,\alpha)$
0	$(r_0,\alpha_0),(r_1,\alpha_1),(r_2,\alpha_2)$
$\Delta\phi$	⋮
$2\Delta\phi$	⋮
...	...

Edge direction is not a unique description
Gives noise in accumulator

Procedure for GHT

Preparation

1. Determine centre of template shape
2. Form R-table from template shape

Application

1. Use R-table to vote for points in the real image

```
For edge points > threshold
Get edge direction(x,y)
For all R-table entries with direction(x,y)
   Vote in accumulator (@distance, @direction)
```

2. `Argmax(accumulator)` gives centre co-ordinates of shape

Arbitrary Shapes

- Use Generalised Hough transform
- Form (discrete) look-up-table (R-table)
- Vote via look-up-table

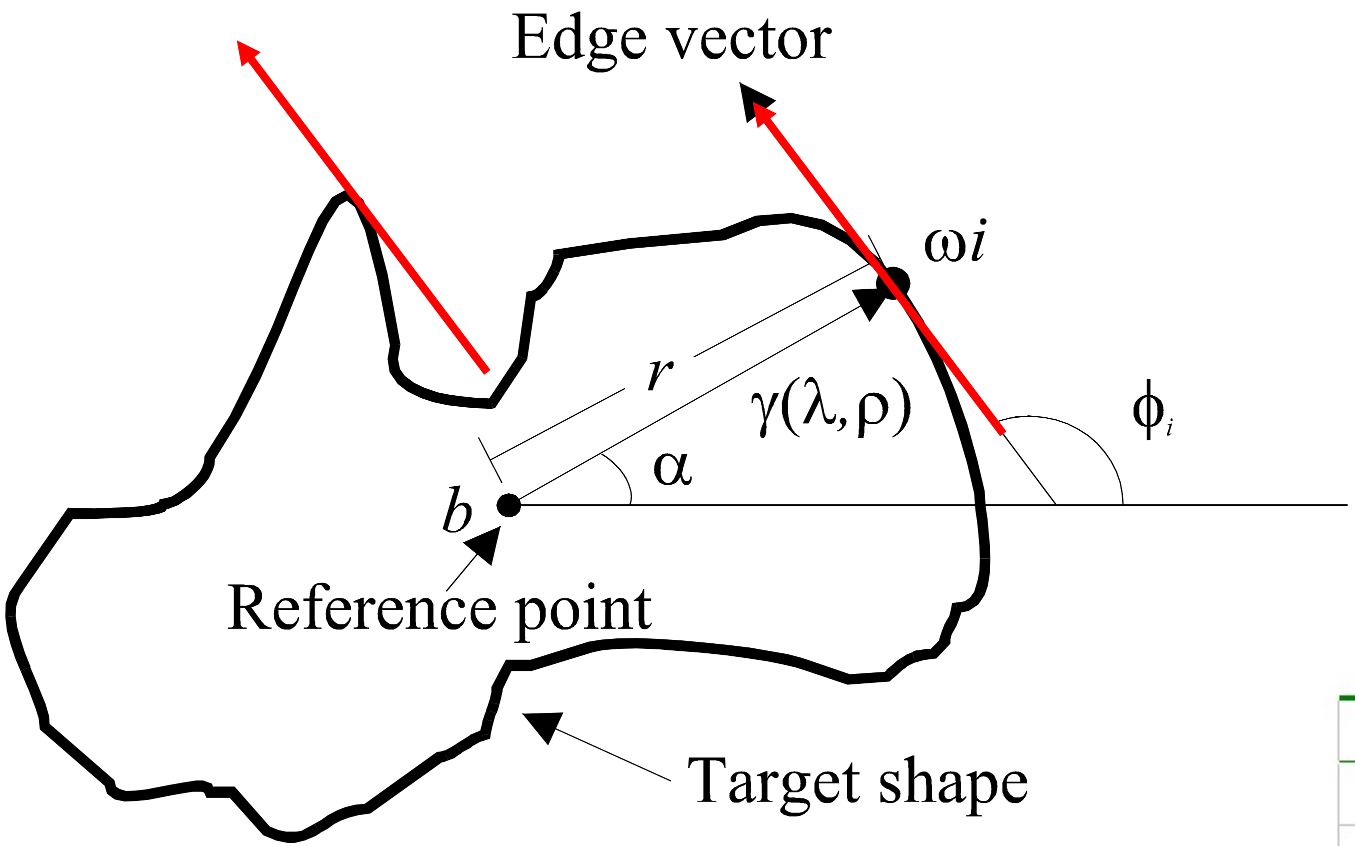
- Scale? scale R-table voting
- Orientation? Rotate R-table voting

- Inherent problems with discretisation

R-table Construction

$$x_c = x_i - rS\cos(\alpha + \theta)$$
$$y_c = y_i - rS\sin(\alpha + \theta)$$

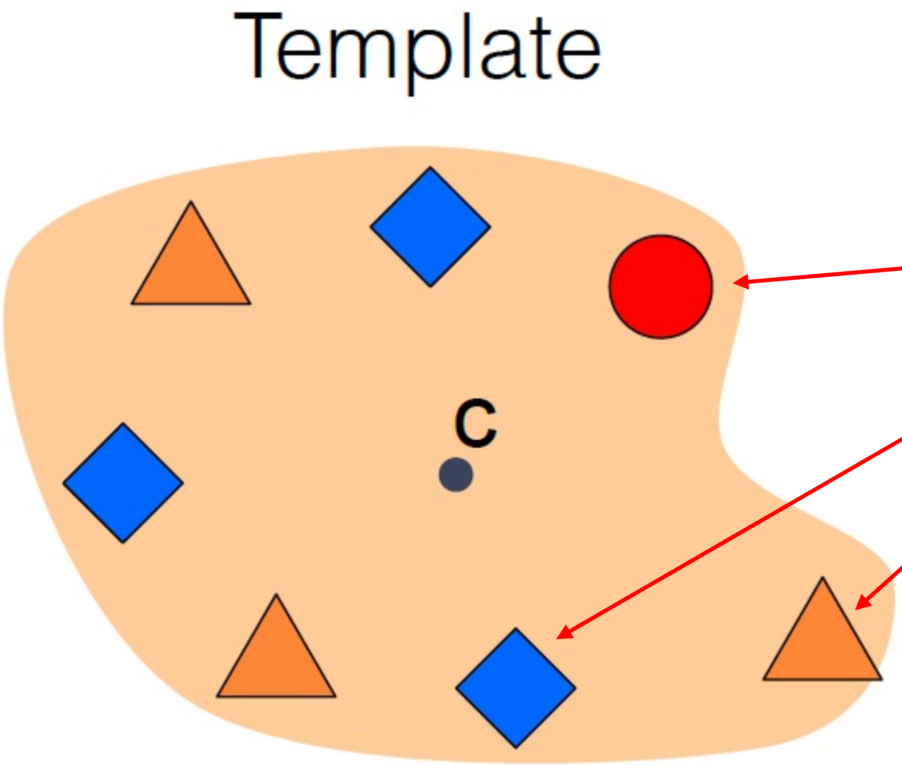
Scale Orientation



Important: for the case of rotation, before checking with the R-table, the edge direction of the edge point needs to minus the rotation θ

$\hat{\varphi}'_i$	$\gamma = (r, \alpha)$
0	$(r_0, \alpha_0), (r_1, \alpha_1), (r_2, \alpha_2)$
$\Delta\phi$	$\vdots$
$2\Delta\phi$	$\vdots$
...	...

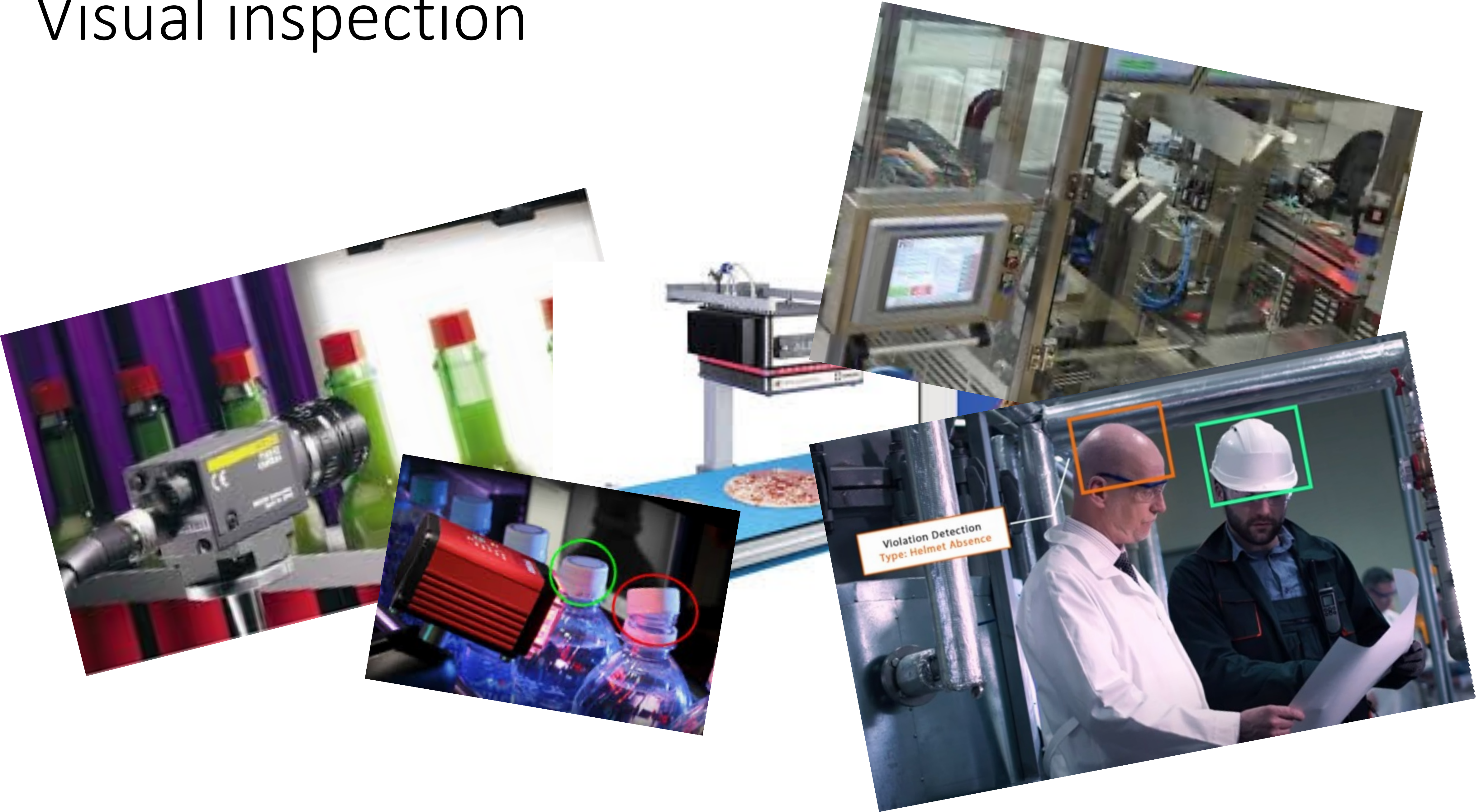
Further example



Firstly, generate an R-table for **objects**

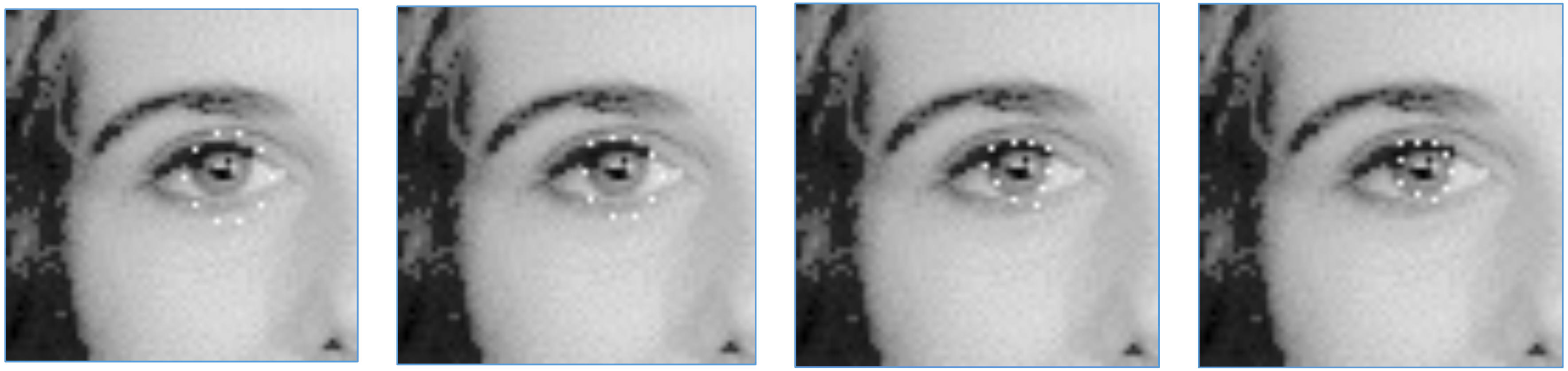
Then, use the R-table to vote for points

Visual inspection

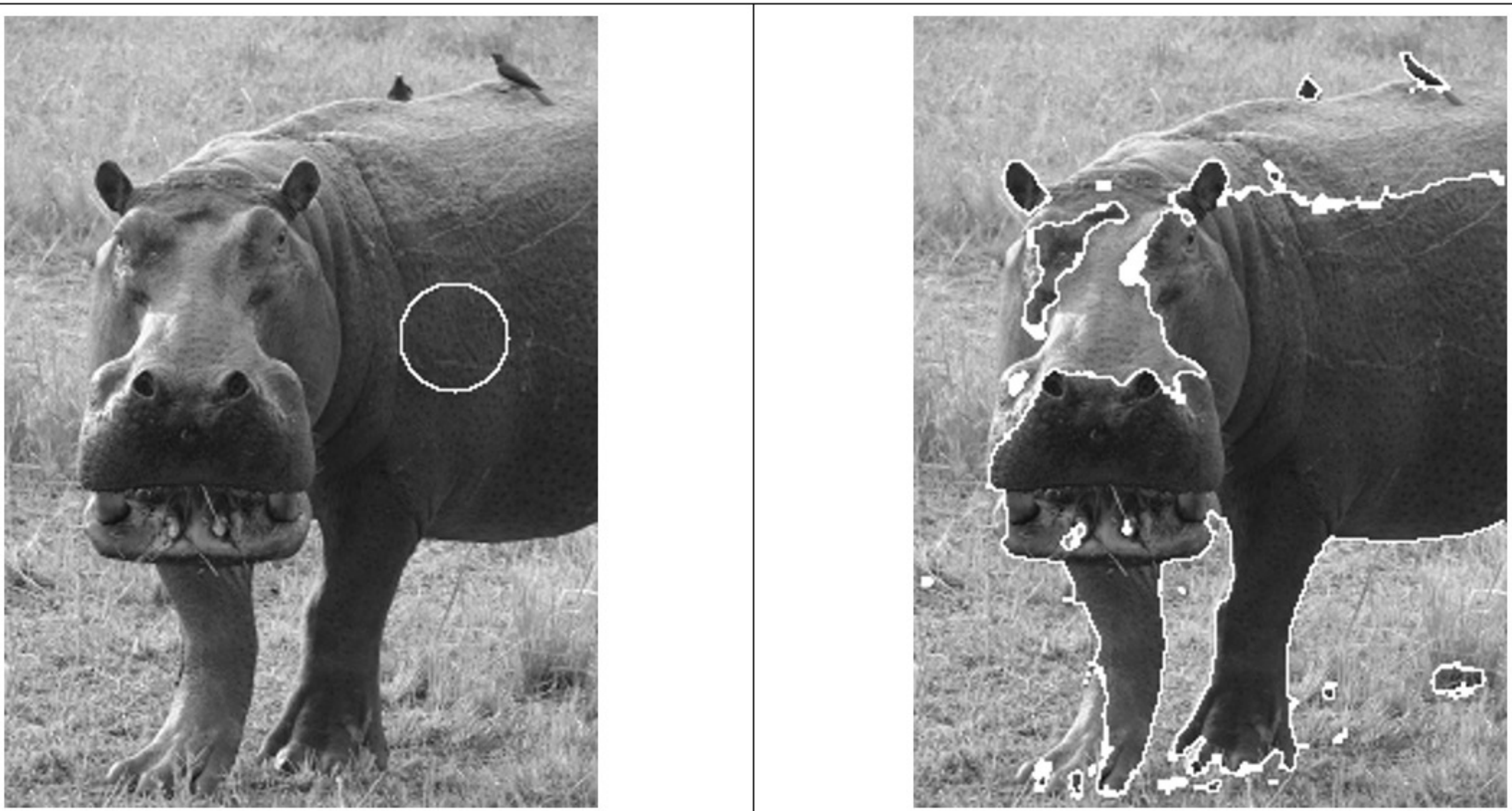


Active Contours

- For **unknown** arbitrary shapes: extract by evolution
- Elastic band analogy
- Balloon analogy
- Discrete vs. continuous



Geometric active contours



(a) initialisation

(b) result

Extraction by a Level-Set Based Approach

Main points so far

1. conic sections become more complex and take more time
2. can use Generalised Hough Transform for complex shapes
3. shape detection IS computer vision. Many more approaches

Let’s see how computer vision can work