Generalized Hough Transform

16-385 Computer Vision

Hough Circles

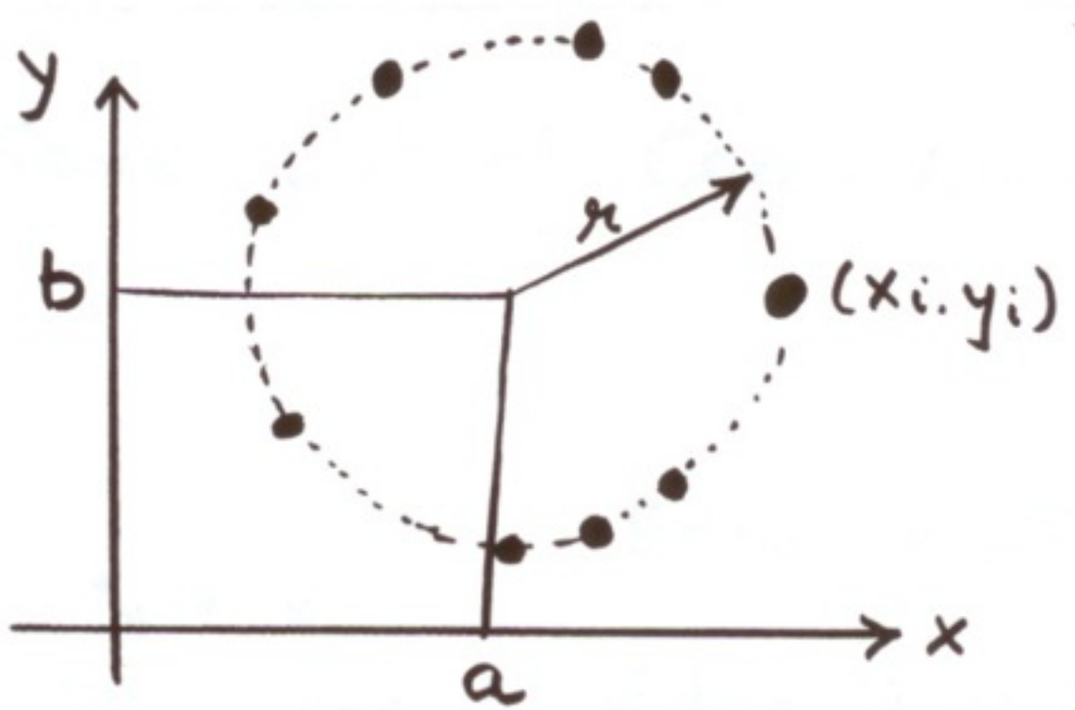
Finding Circles by Hough Transform

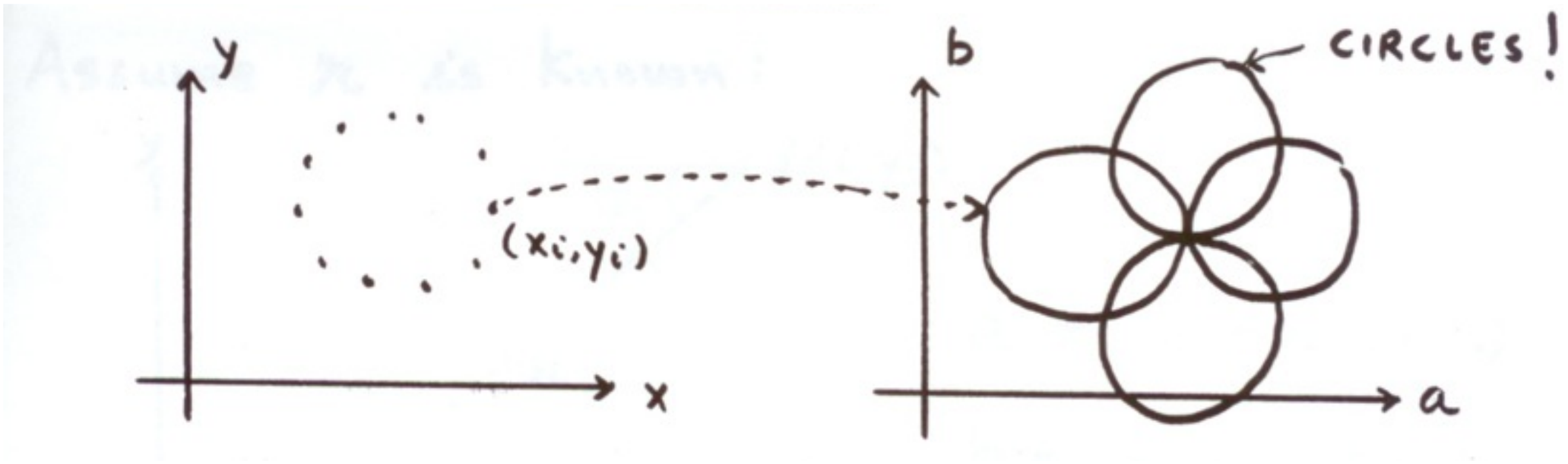
Equation of Circle:

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

If radius is known: (2D Hough Space)

Accumulator Array $A(a,b)$





parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables

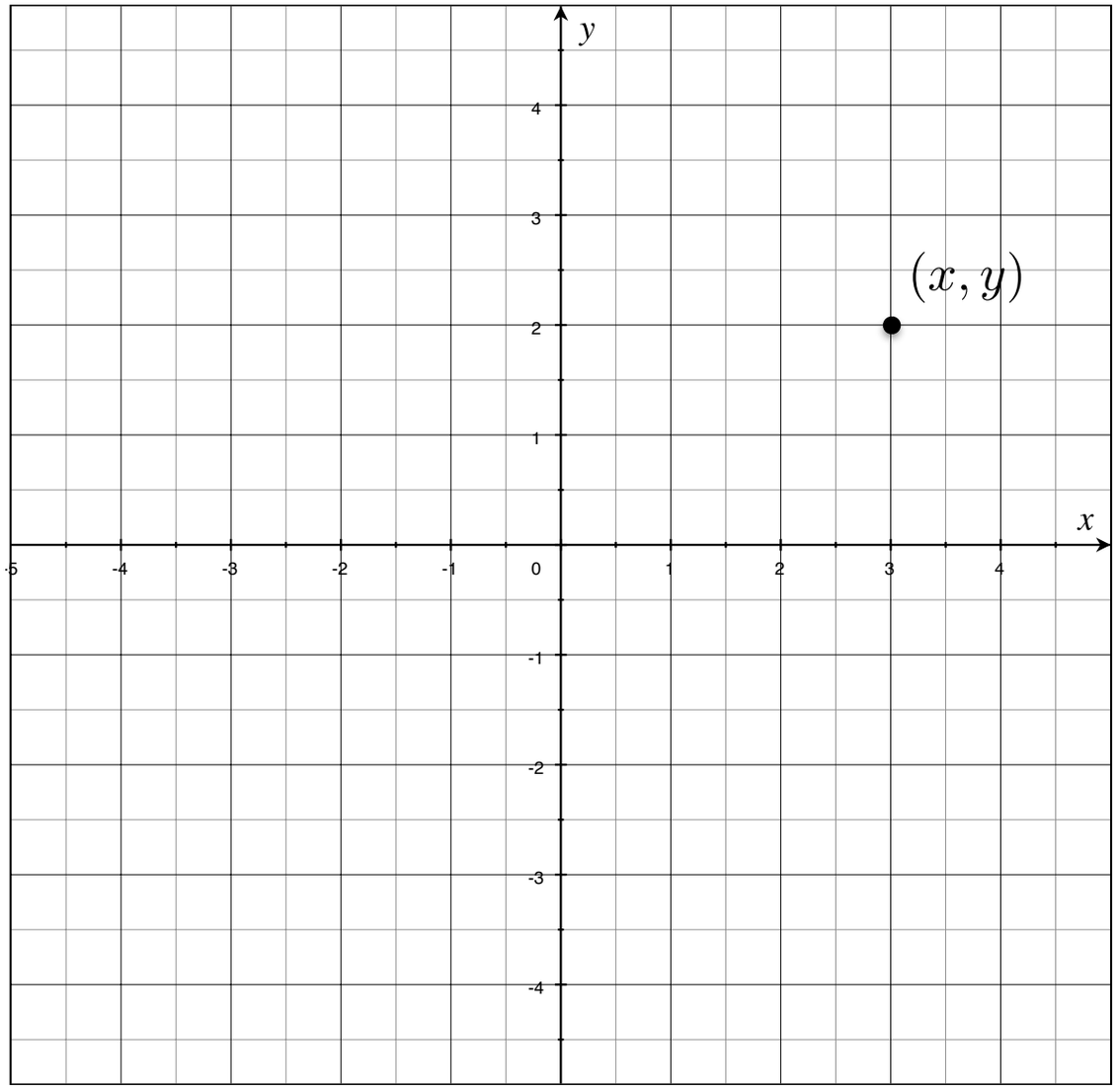
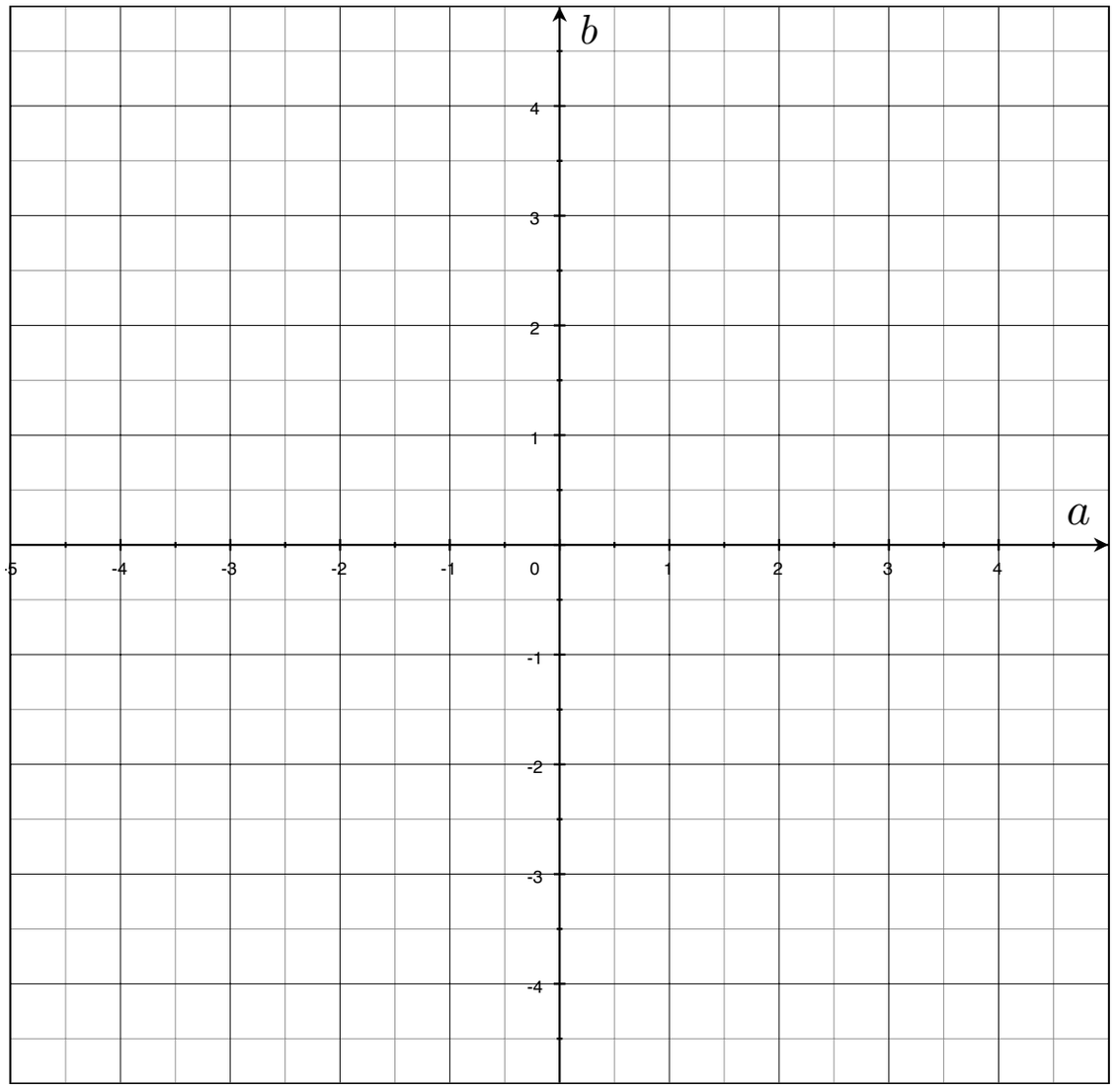


Image space

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables

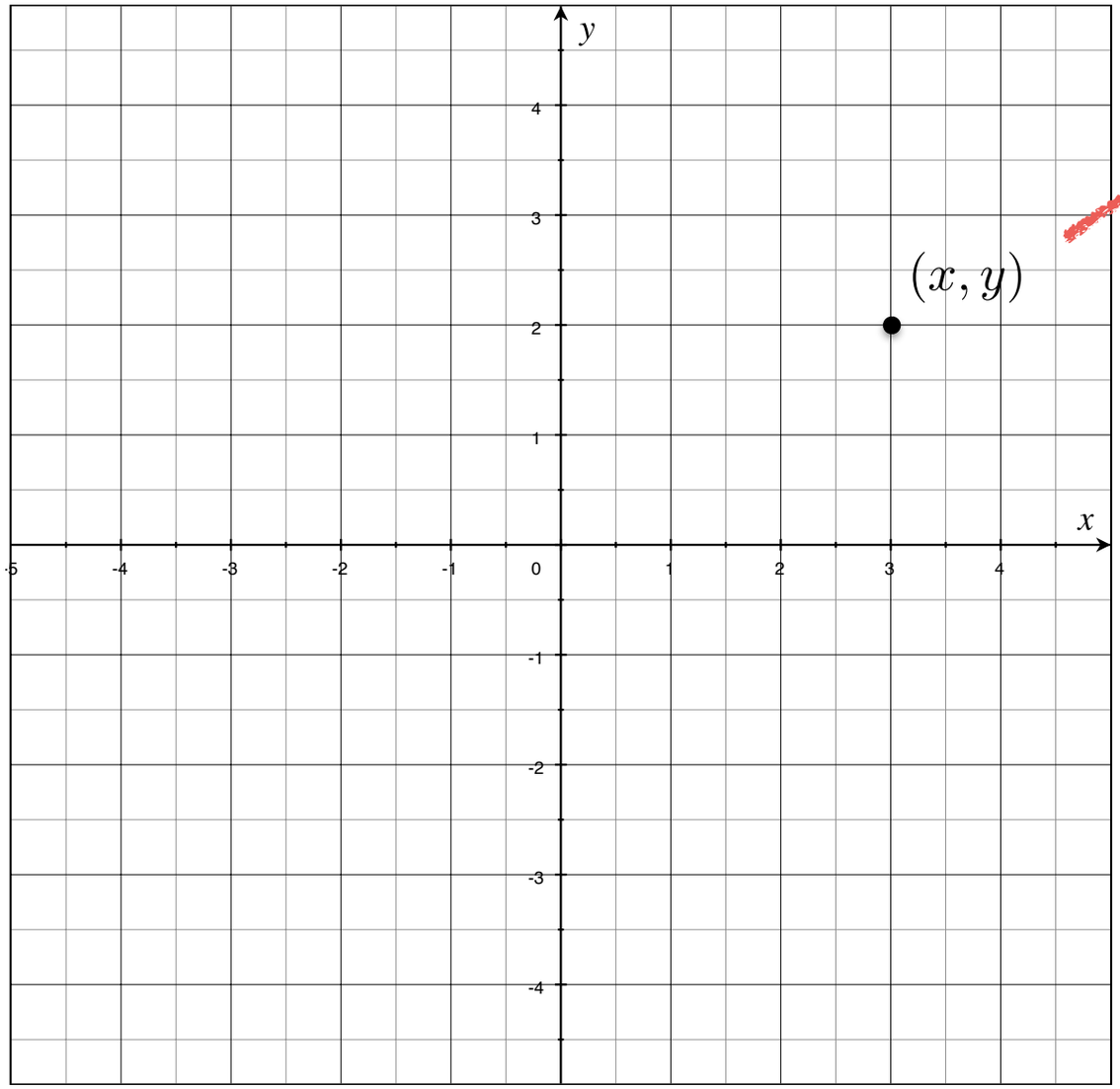


Parameter space

parameters

$$(x-a)^2+(y-b)^2=r^2$$

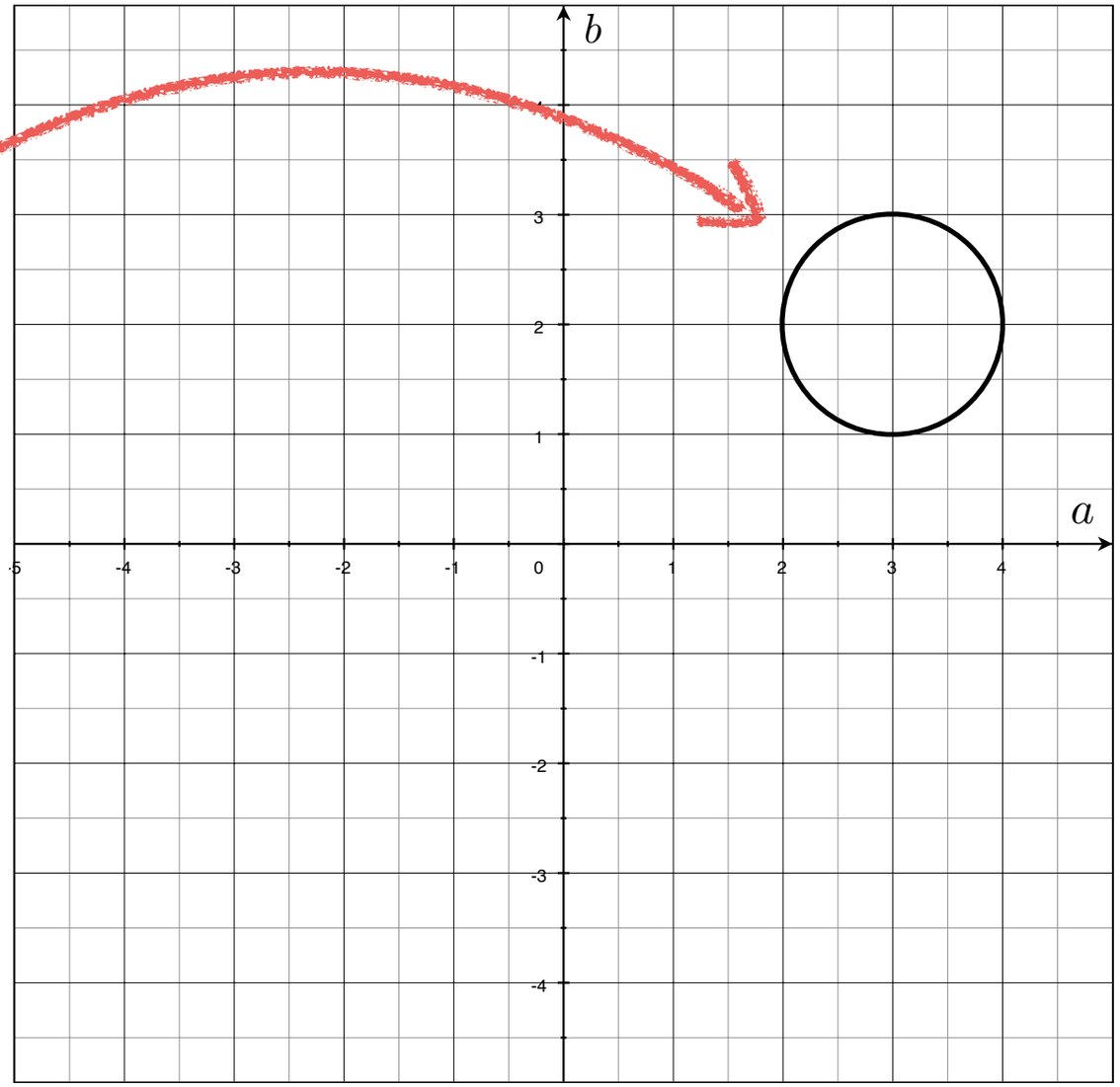
variables



parameters

$$(x-a)^2+(y-b)^2=r^2$$

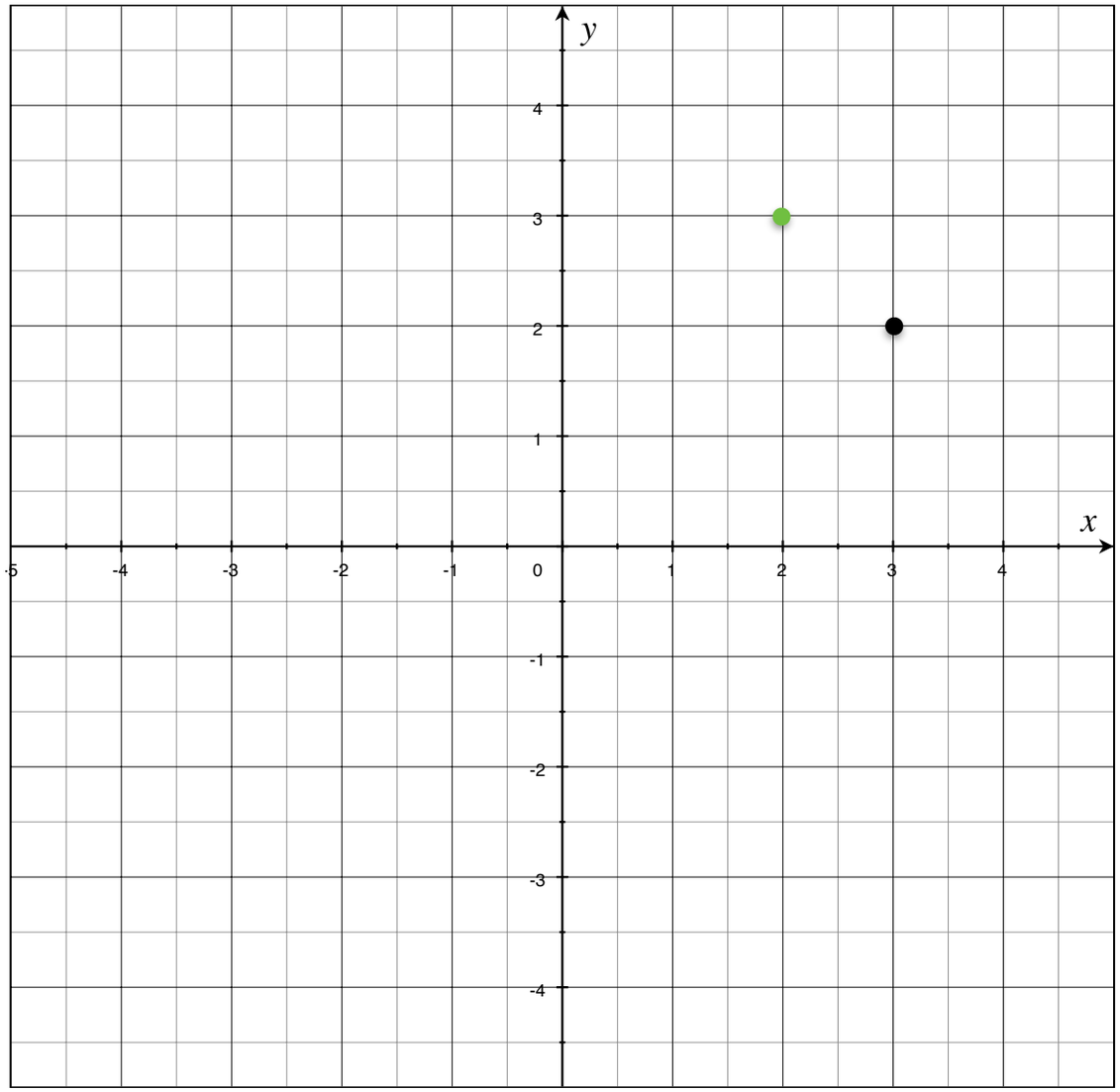
variables



parameters

$$(x-a)^2+(y-b)^2=r^2$$

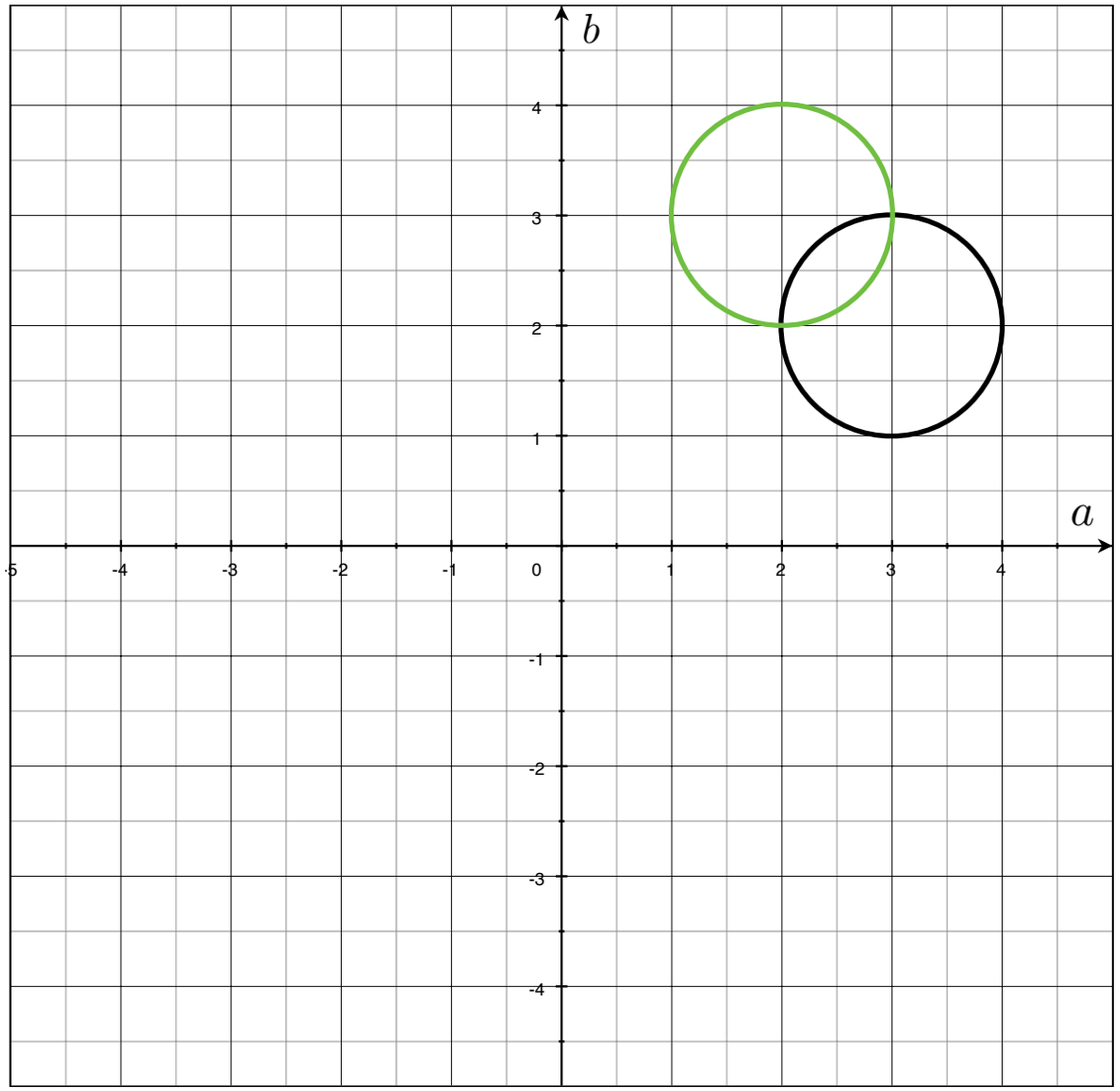
variables



parameters

$$(x-a)^2+(y-b)^2=r^2$$

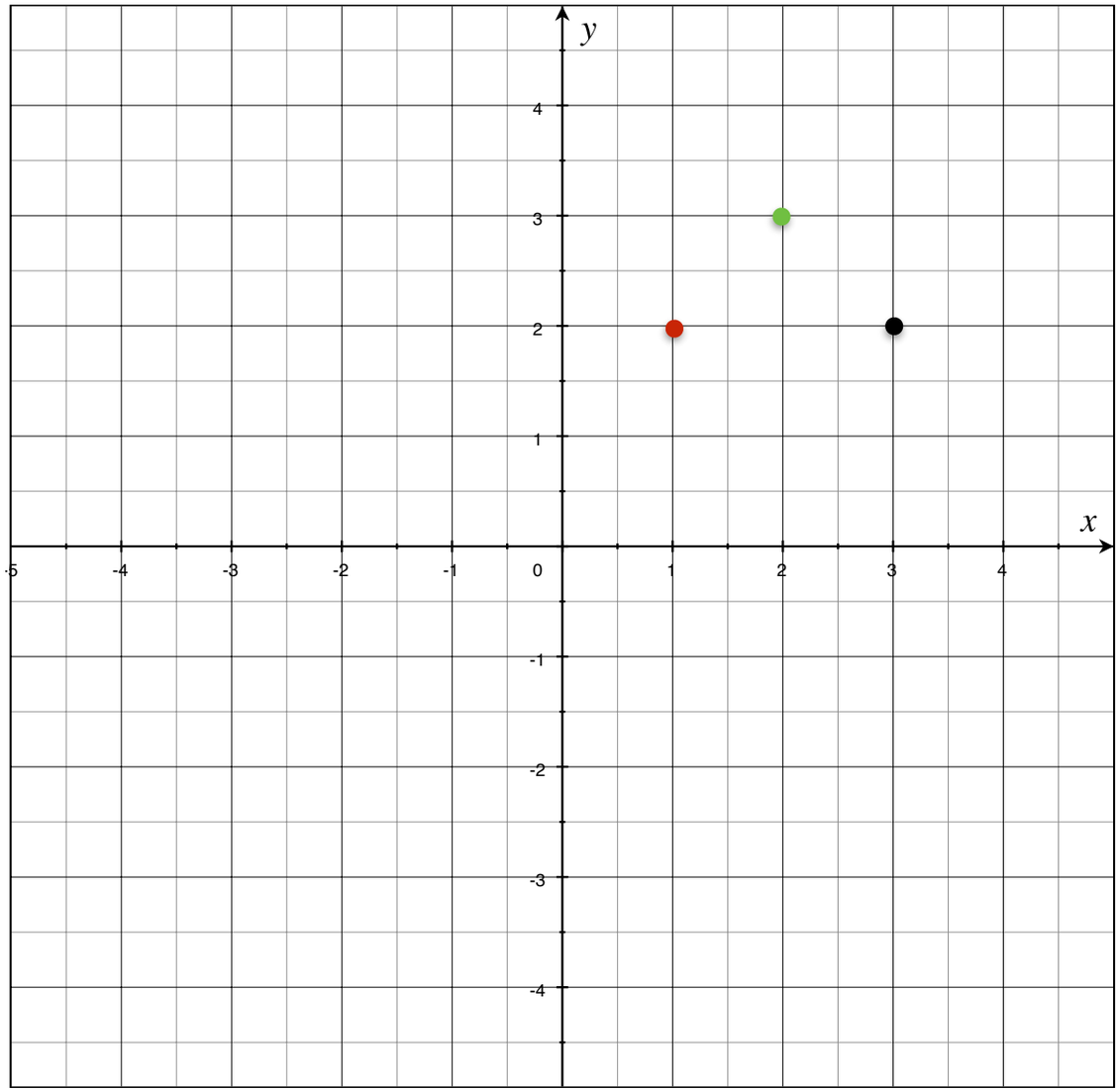
variables



parameters

$$(x-a)^2+(y-b)^2=r^2$$

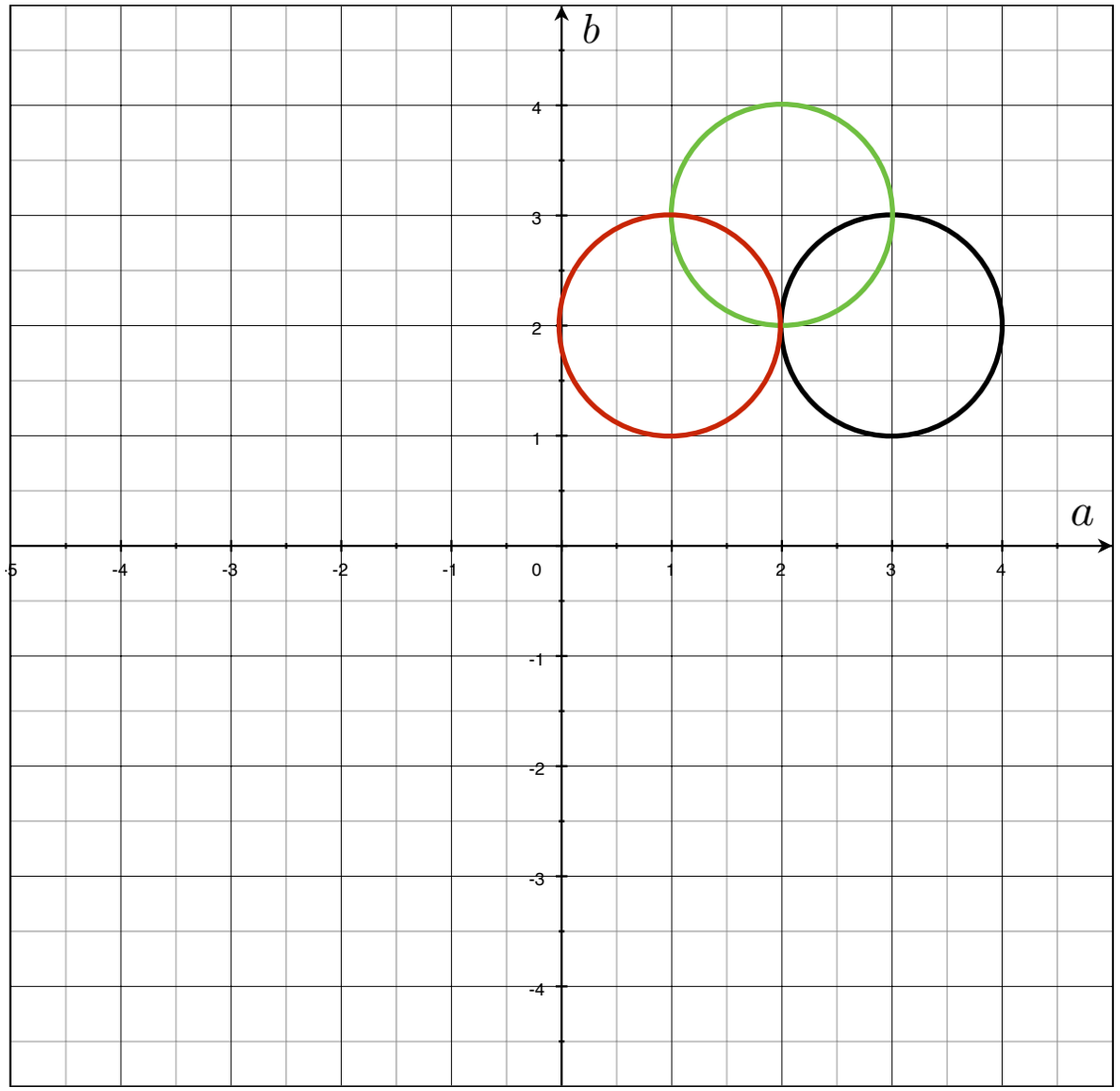
variables



parameters

$$(x-a)^2+(y-b)^2=r^2$$

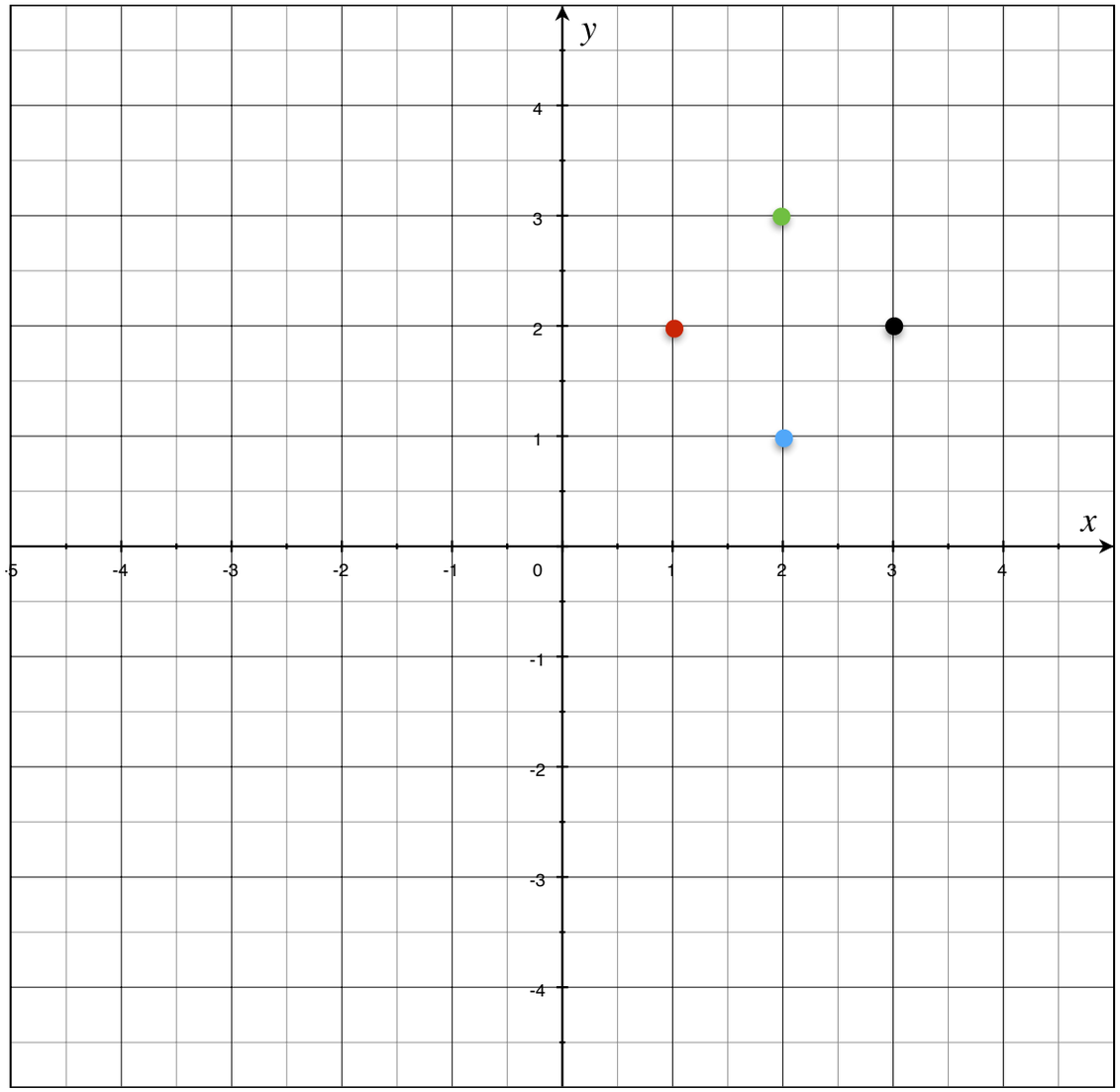
variables



parameters

$$(x-a)^2+(y-b)^2=r^2$$

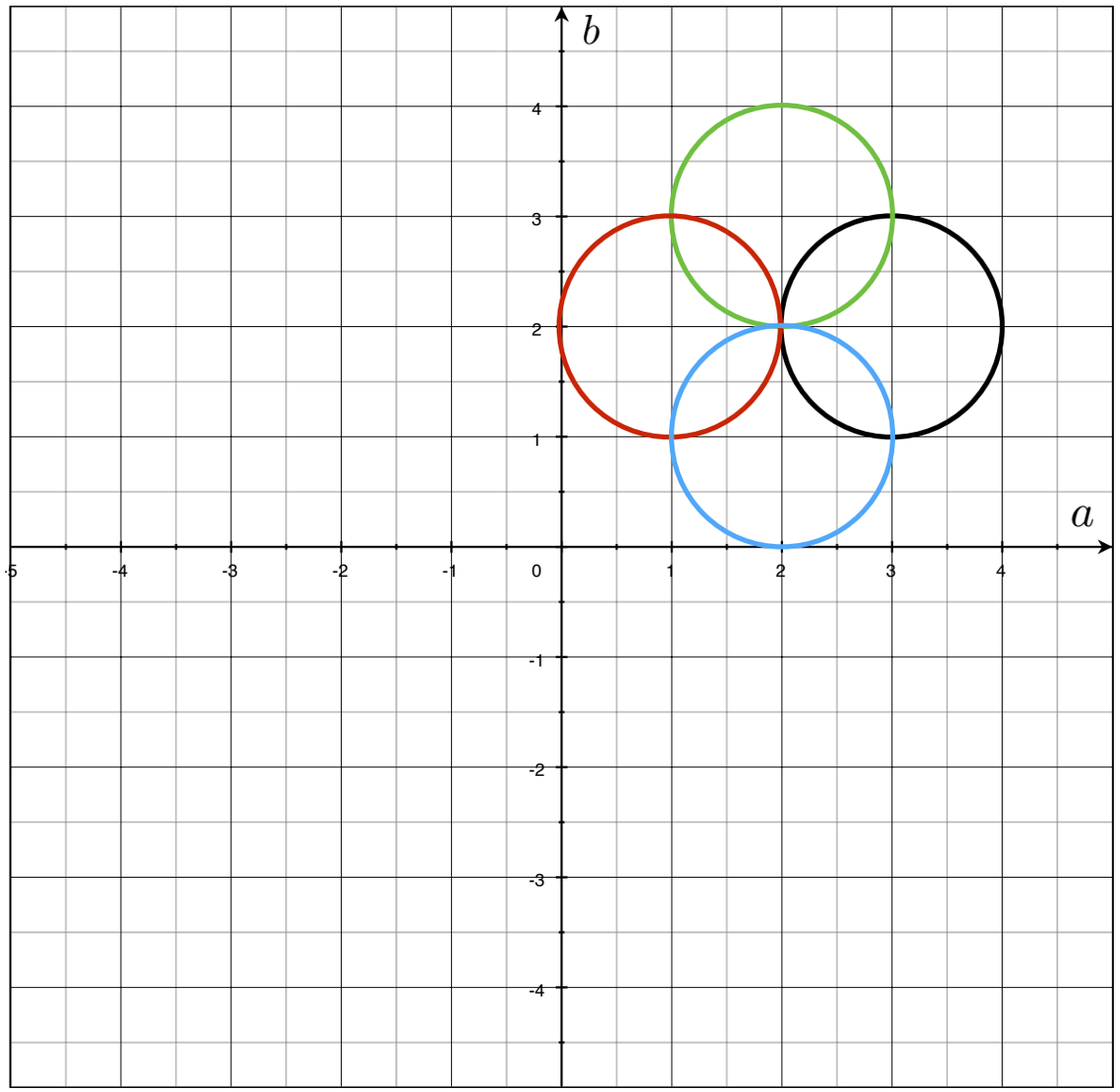
variables



parameters

$$(x-a)^2+(y-b)^2=r^2$$

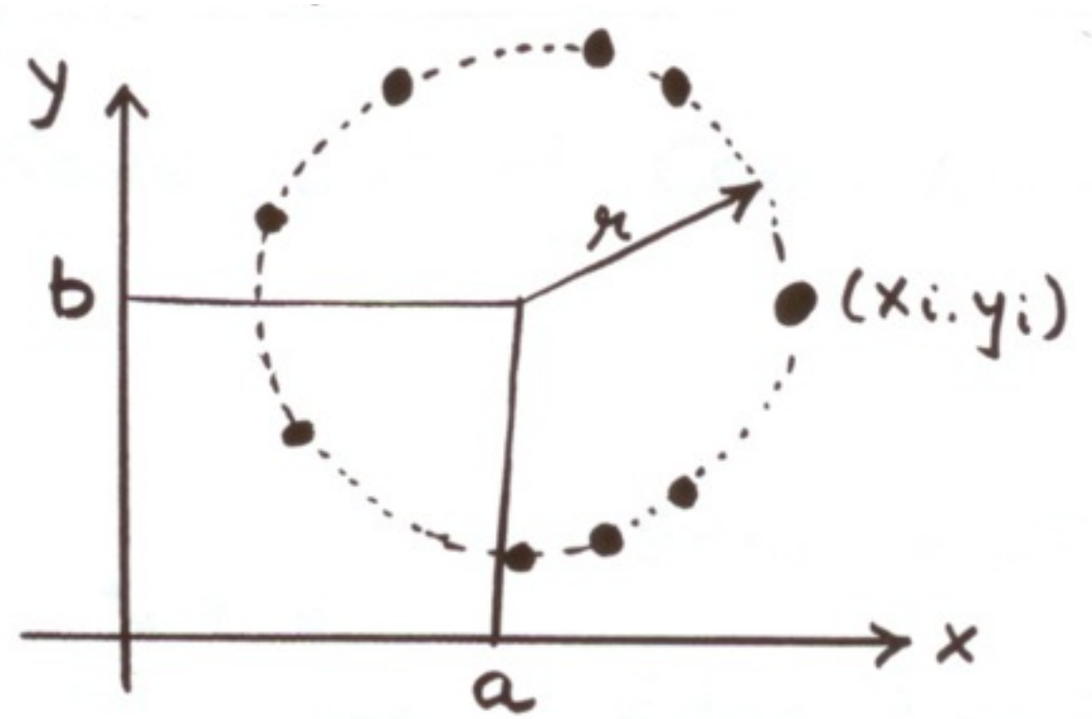
variables



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!

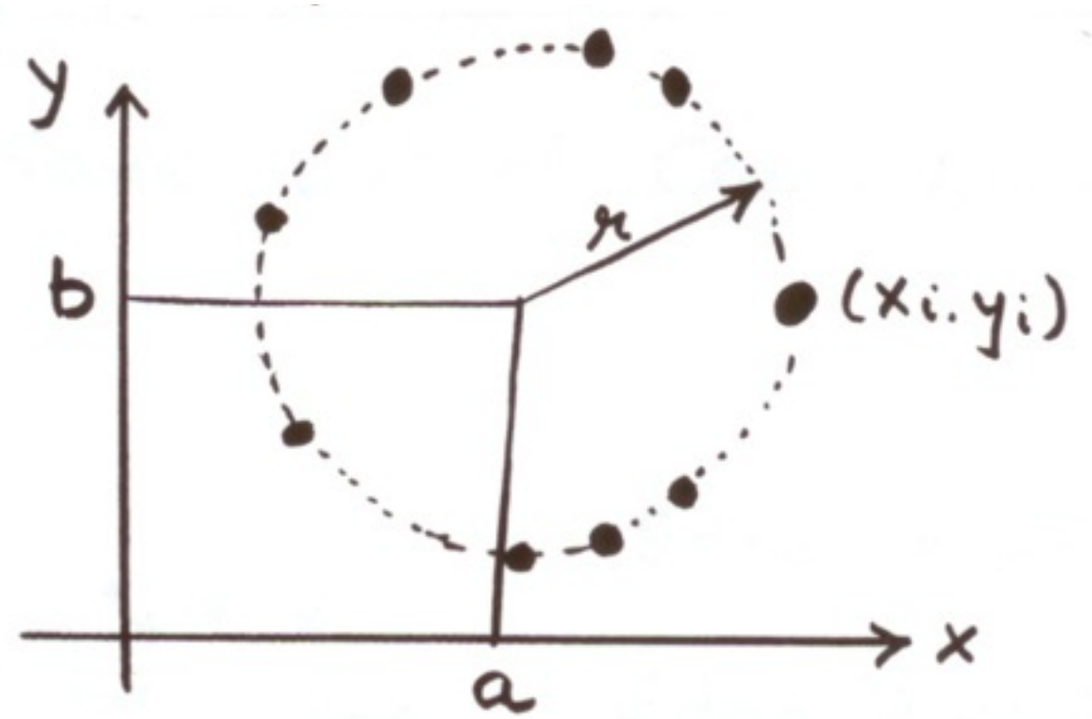
Use Accumulator array $A(a, b, r)$

What is the surface in the hough space?

Finding Circles by Hough Transform

Equation of Circle:

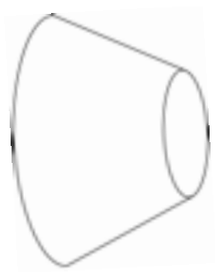
$$(x_i - a)^2 + (y_i - b)^2 = r^2$$



If radius is not known: 3D Hough Space!

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

What is the surface in the hough space?

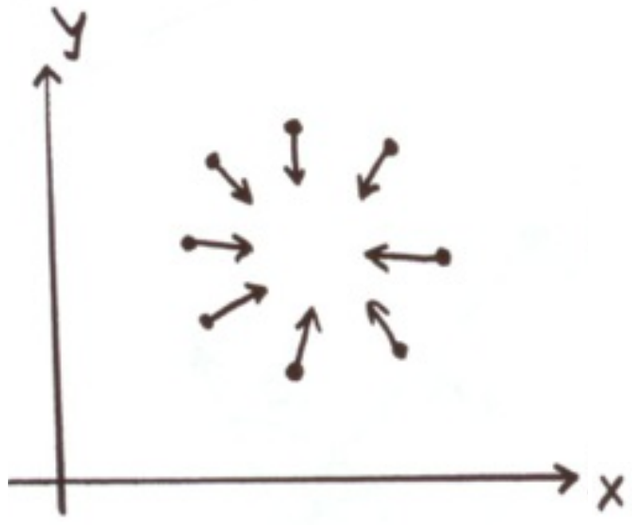


Using Gradient Information

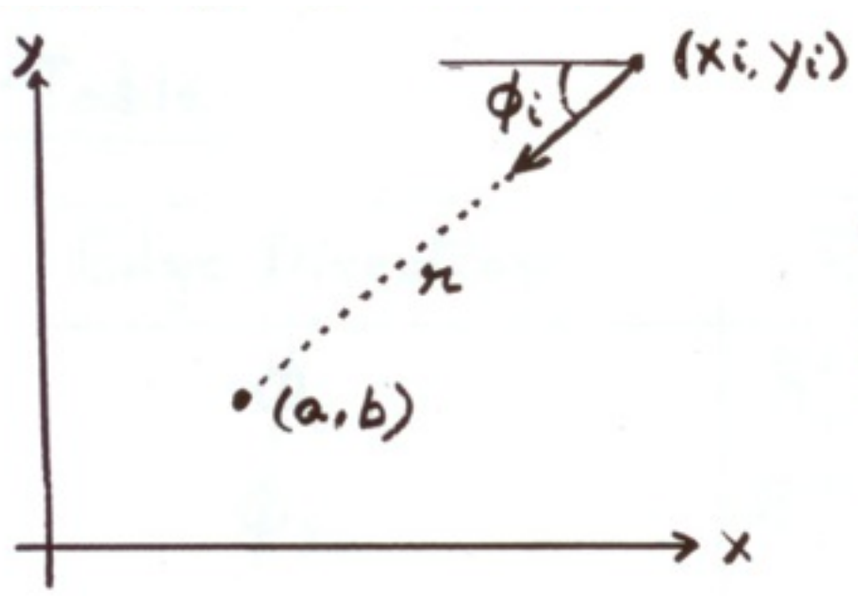
Gradient information can save lot of computation:

Edge Location (x_i, y_i)

Edge Direction ϕ_i



Assume radius is known:



$$a = x - r\cos\phi$$

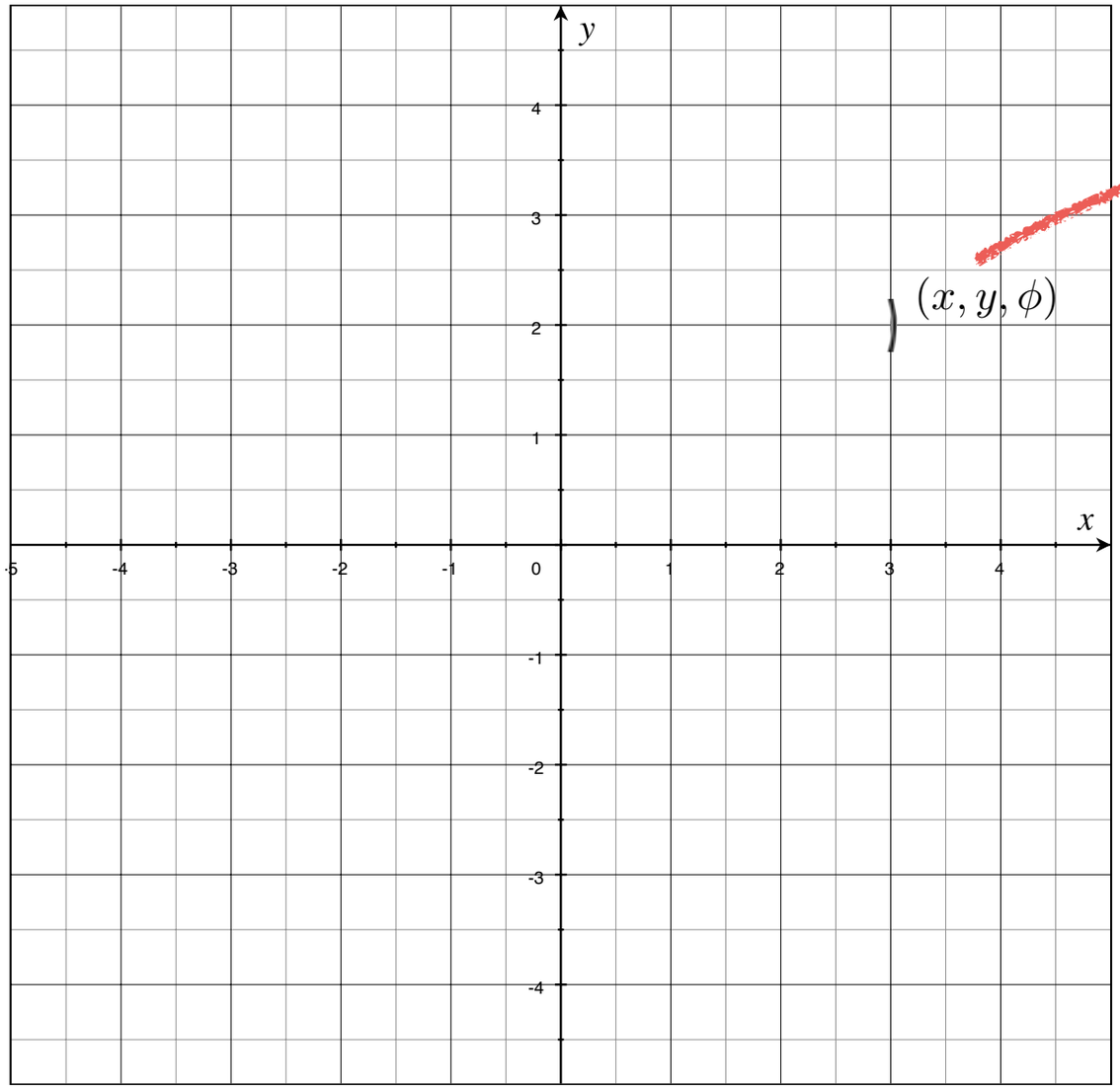
$$b = y - r\sin\phi$$

Need to increment only one point in accumulator!!

parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables

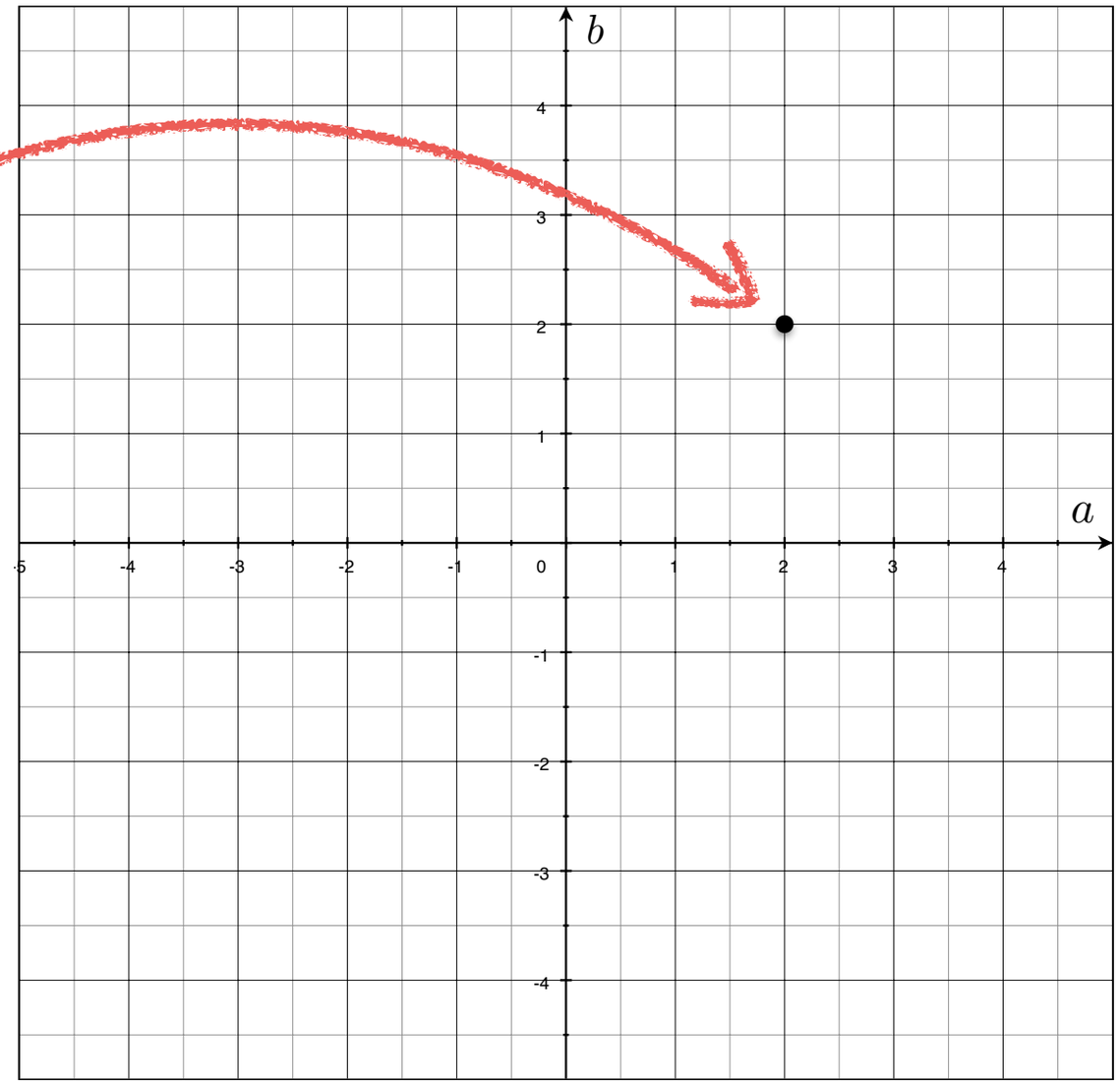


(x, y, ϕ)

parameters

$$(x-a)^2+(y-b)^2=r^2$$

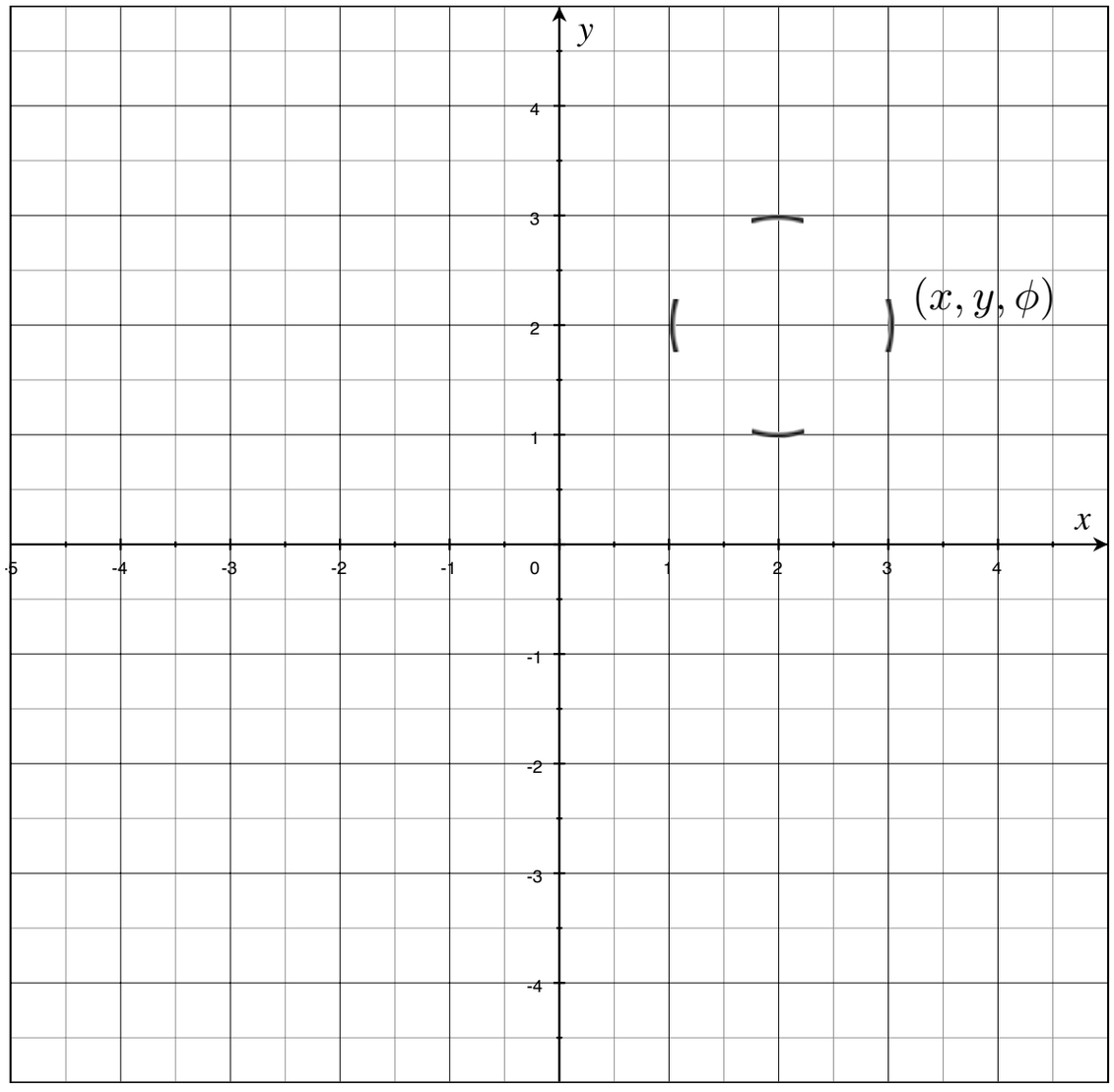
variables



parameters

$$(x-a)^2+(y-b)^2=r^2$$

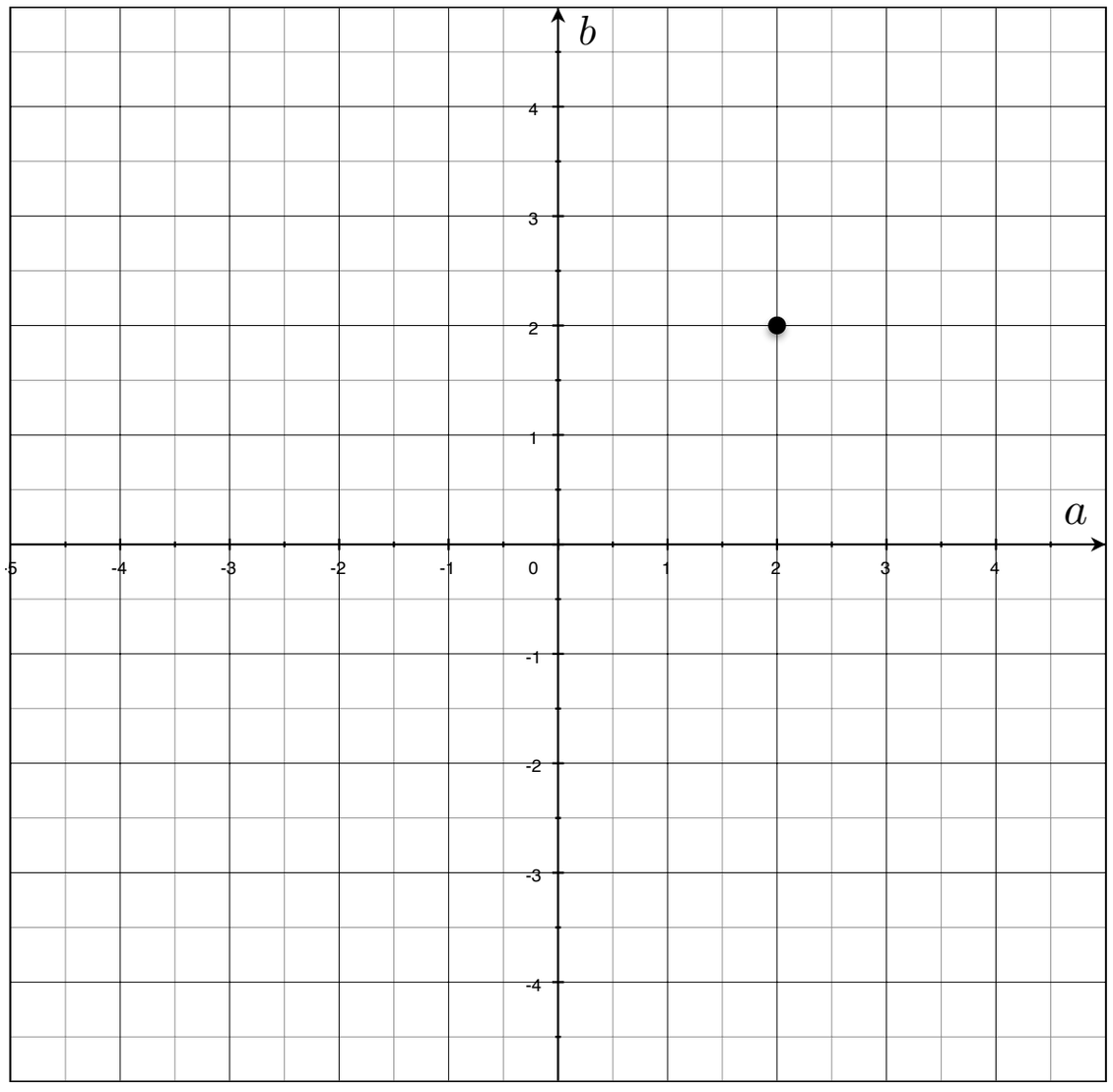
variables

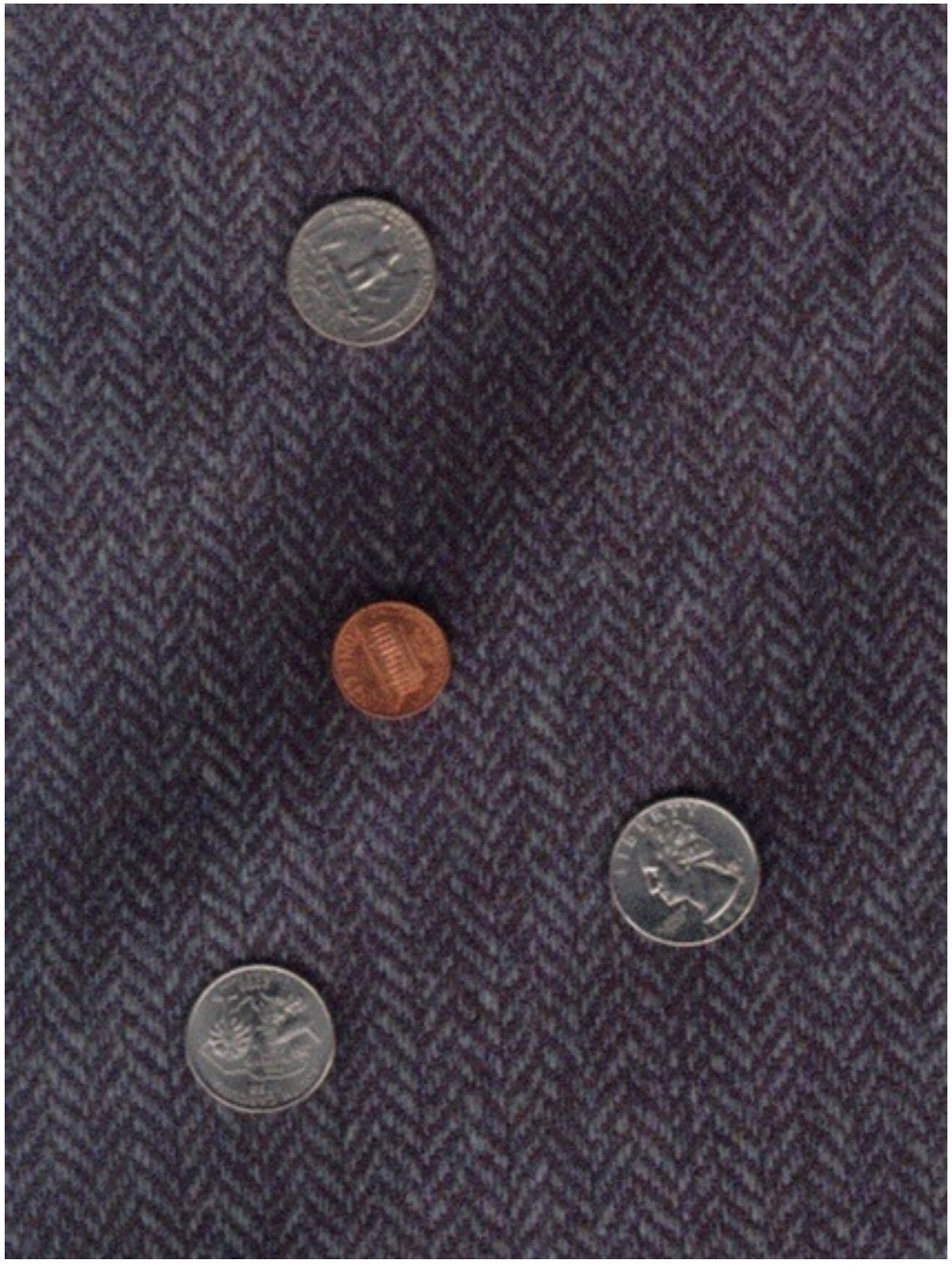


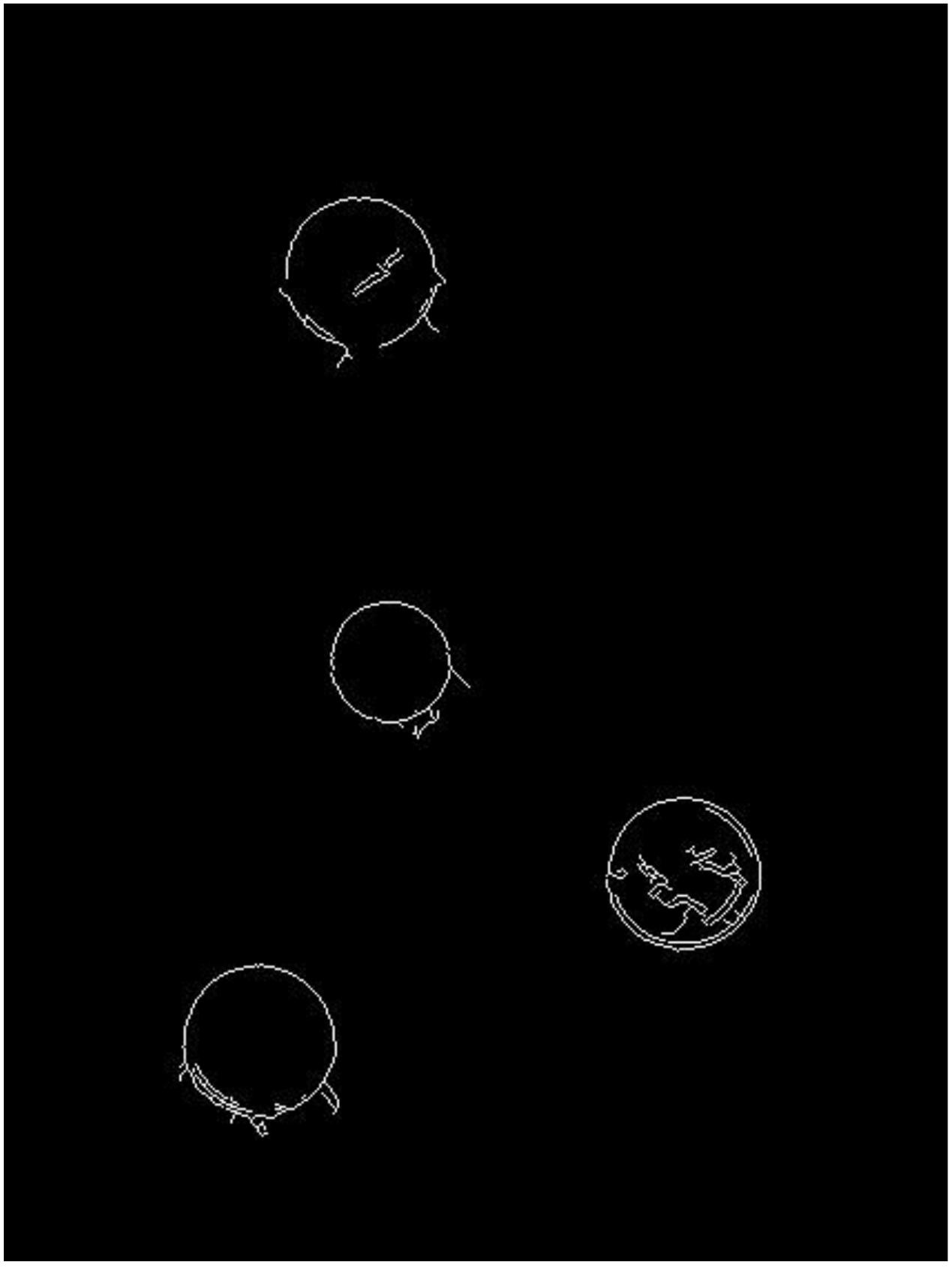
parameters

$$(x-a)^2+(y-b)^2=r^2$$

variables



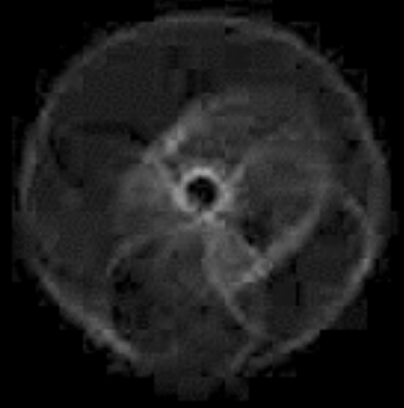




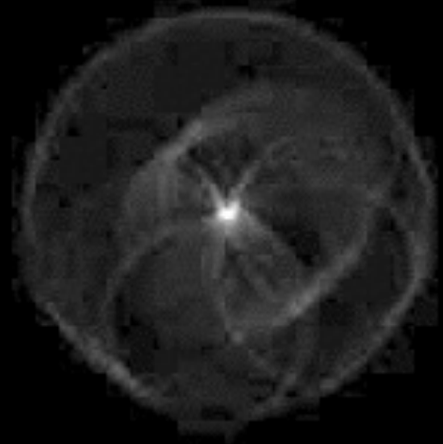
Pennie Hough detector

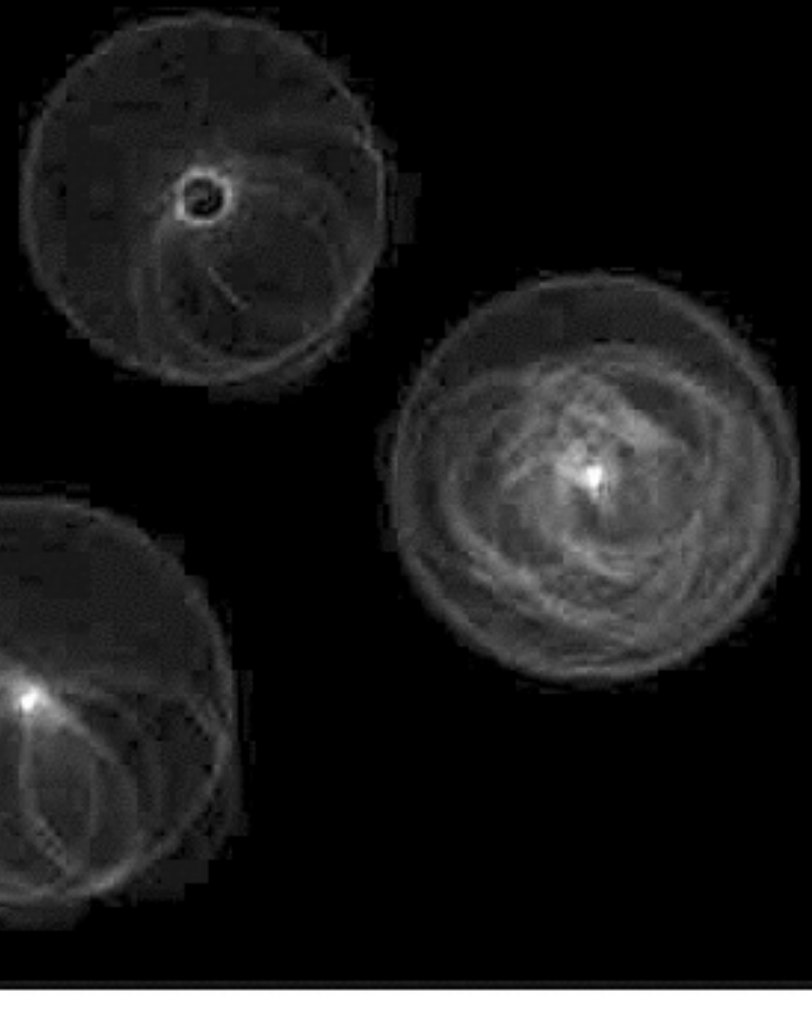
Quarter Hough detector

Pennie Hough detector

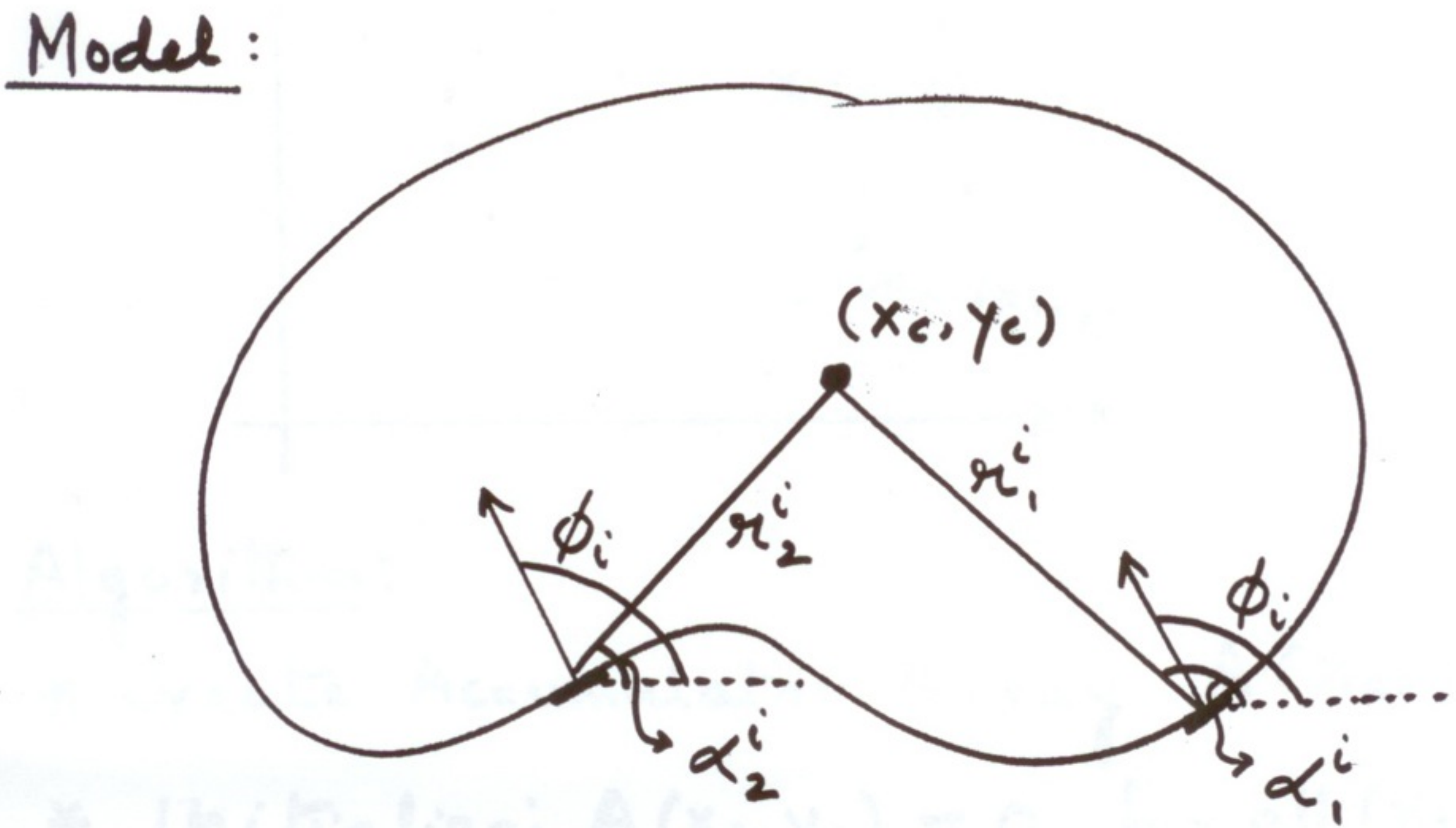


Quarter Hough detector





Generalized Hough Transform



ϕ-Table

Edge Direction	$\overline{r} = (r, \alpha)$
ϕ_1	$\overline{r}^1_1, \overline{r}^1_2, \overline{r}^1_3$
ϕ_2	$\overline{r}^2_1, \overline{r}^2_2$
$\vdots$	$\vdots \quad \vdots$
ϕ_i	$\overline{r}^i_1 \quad \overline{r}^i_2$
$\vdots$	$\vdots \quad \vdots$
ϕ_n	$\overline{r}^n_1, \overline{r}^n_2$

Generalized Hough Transform

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

Create Accumulator Array $A(x_c, y_c)$

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

For each edge point (x_i, y_i, ϕ_i)

For each entry $\bar{r}_k^i$ in table, compute:

$$x_c = x_i + r_k^i \cos\alpha_k^i$$

$$y_c = y_i + r_k^i \sin\alpha_k^i$$

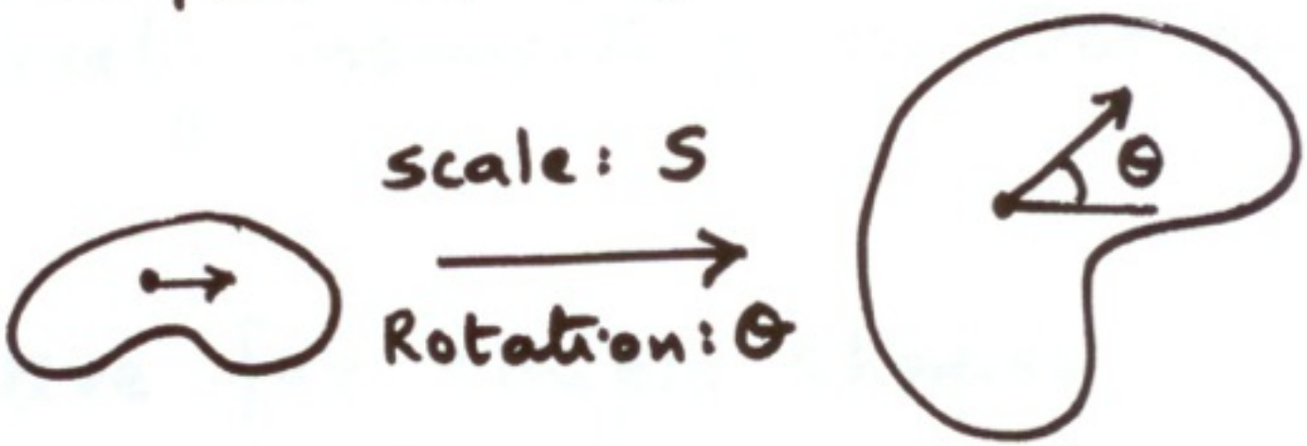
Increment Accumulator: $A(x_c, y_c) = A(x_c, y_c) + 1$

Find Local Maxima in $A(x_c, y_c)$

Scale & Rotation:

Use Accumulator Array:

$$A[x_c, y_c, S, \theta]$$



Use:

$$x_c = x_i + r^i_k S \cos(\alpha^i_k + \theta)$$
$$y_c = y_i + r^i_k S \sin(\alpha^i_k + \theta)$$

$$A(x_c, y_c, S, \theta) = A(x_c, y_c, S, \theta) + 1 .$$

A. Train phase:

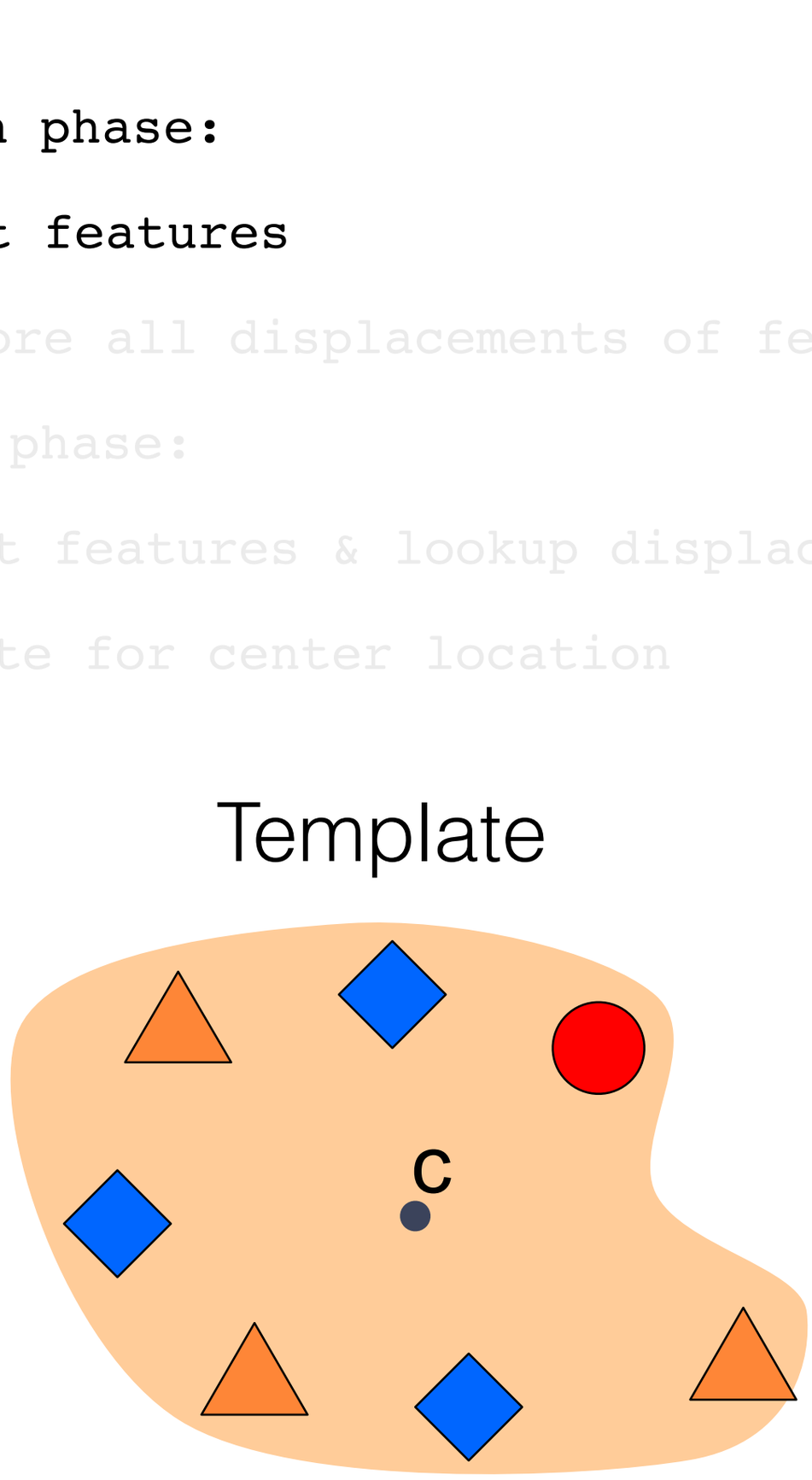
   1. Get features

   2. Store all displacements of feature from center

B. Test phase:

   1. Get features & lookup displacements

   2. Vote for center location



A. Train phase:

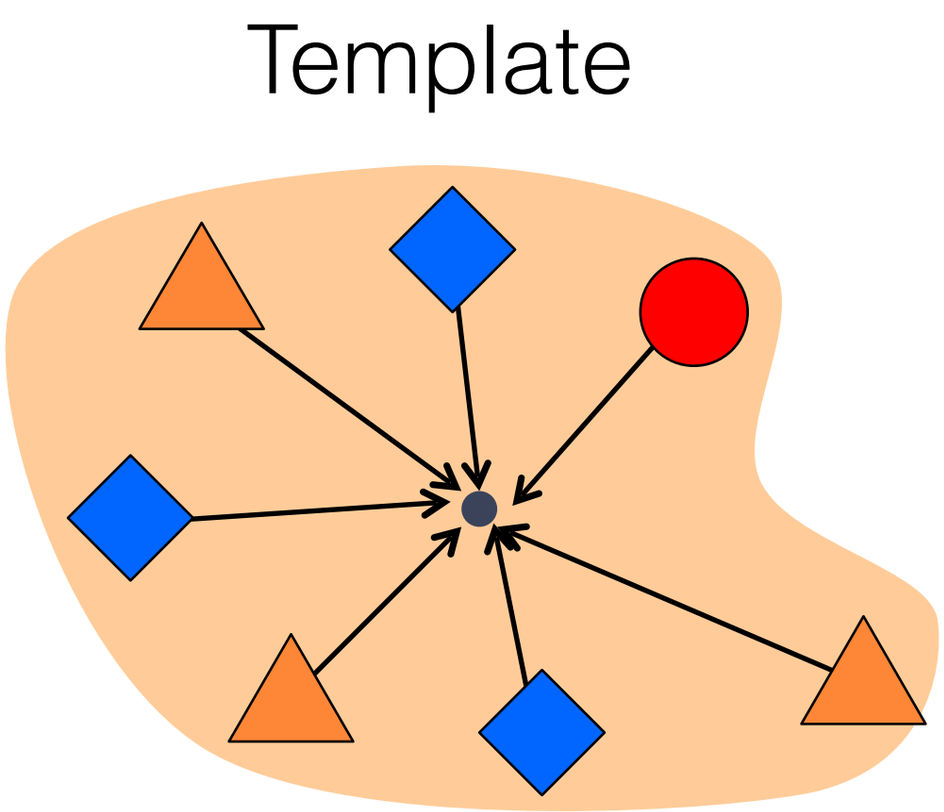
   1. Get features

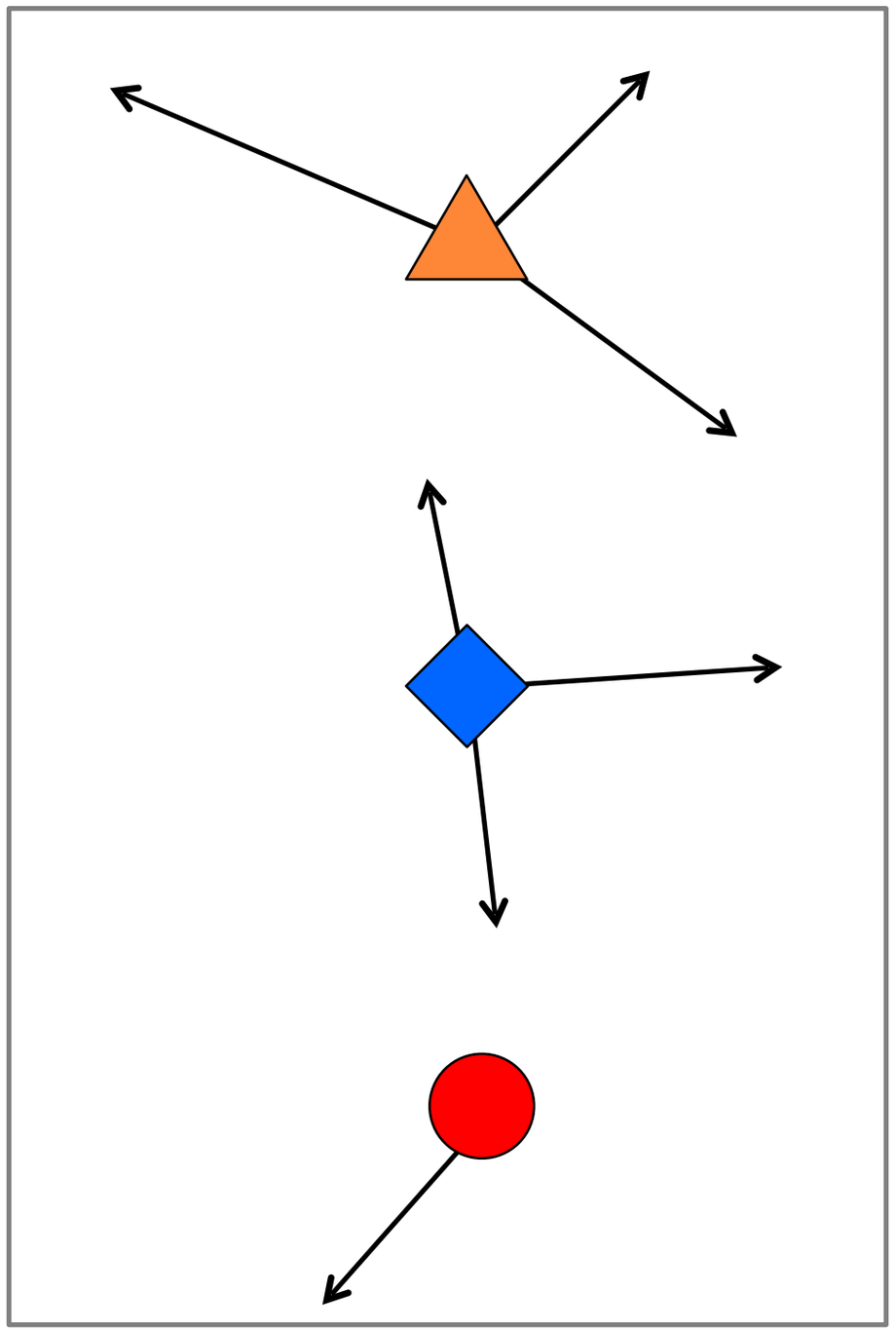
   2. Store all displacements of feature from center

B. Test phase:

   1. Get features & lookup displacements

   2. Vote for center location





A. Train phase:

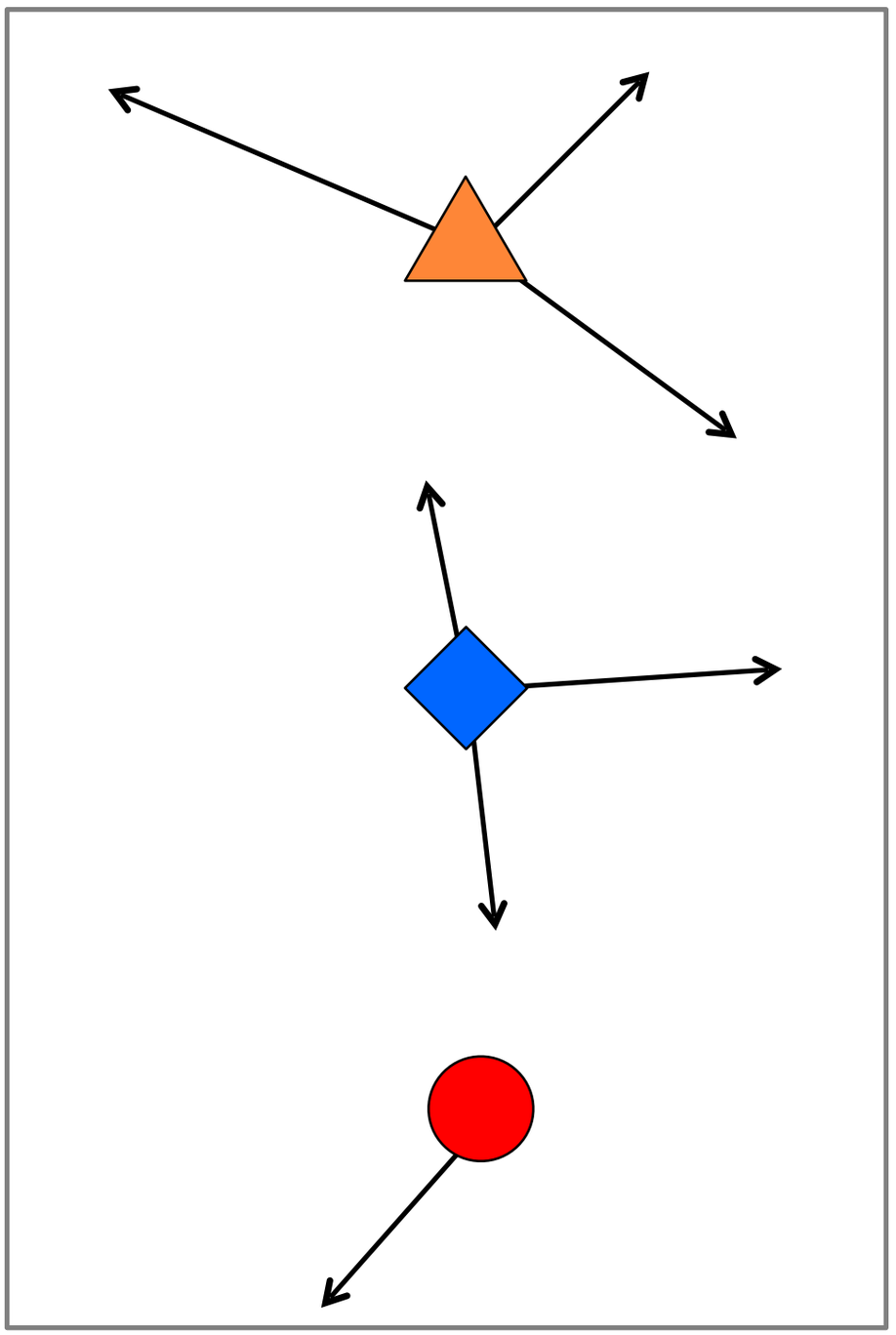
   1. Get features

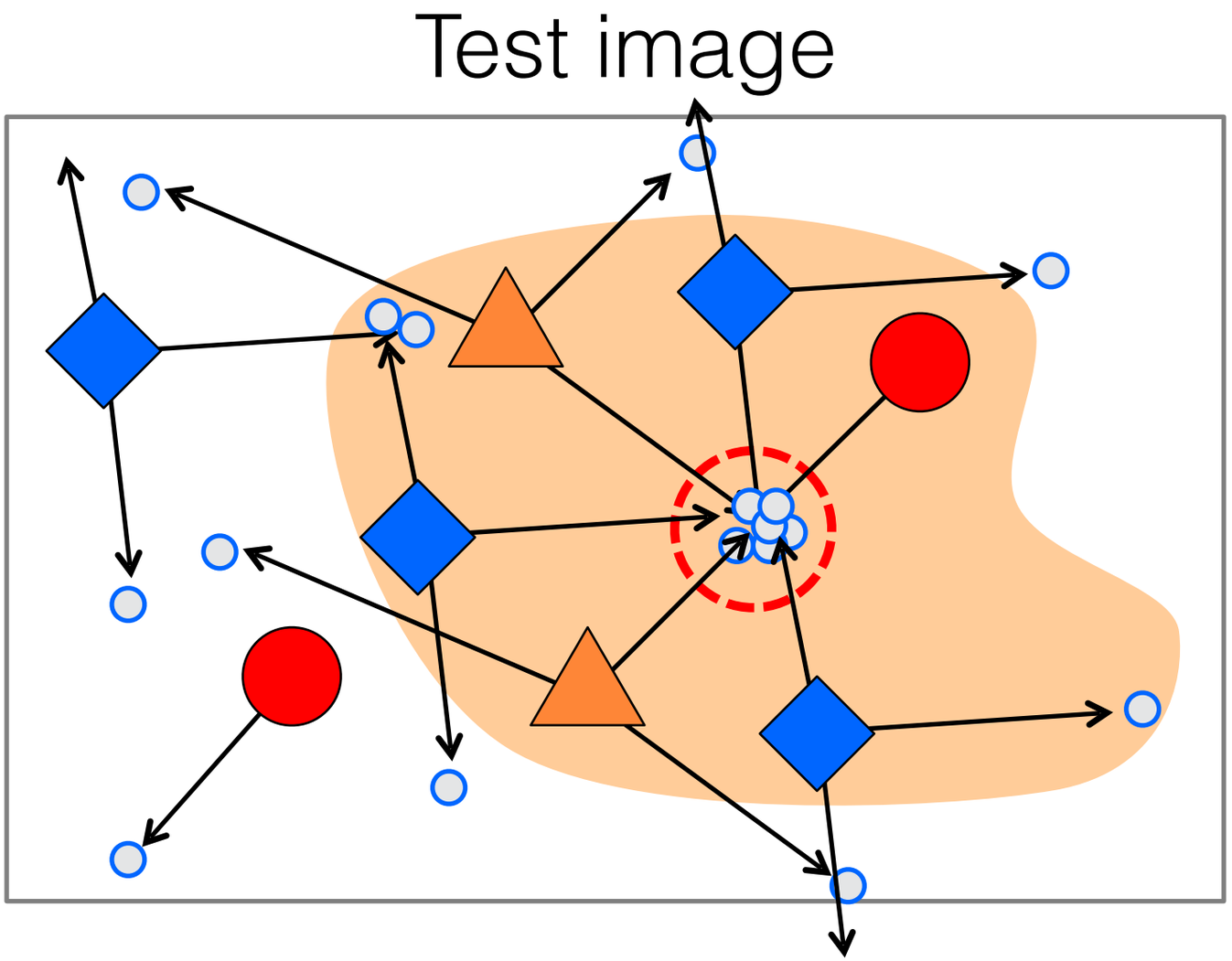
   2. Store all displacements of feature from center

B. Test phase:

   1. Get features & lookup displacements

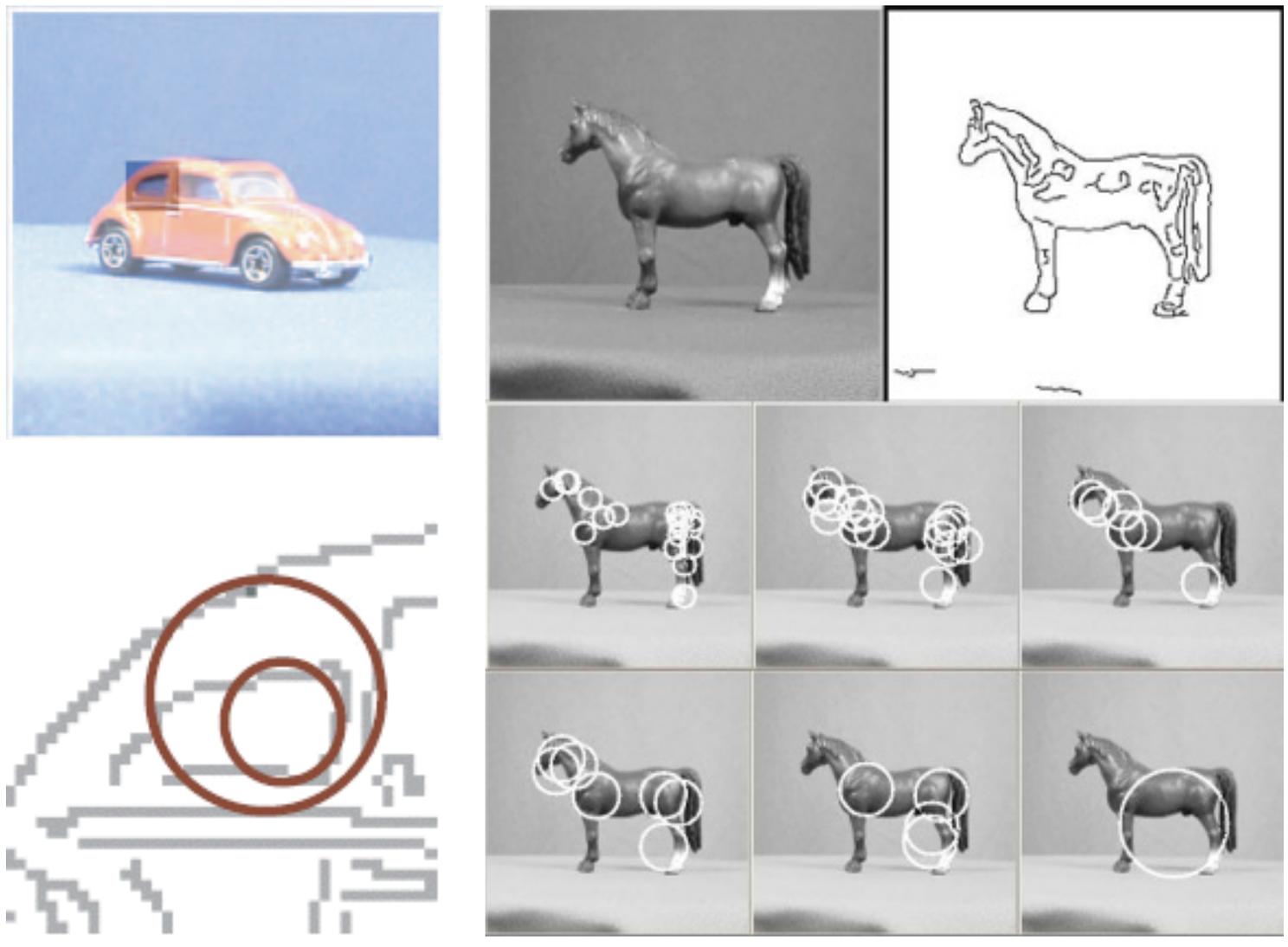
   2. Vote for center location





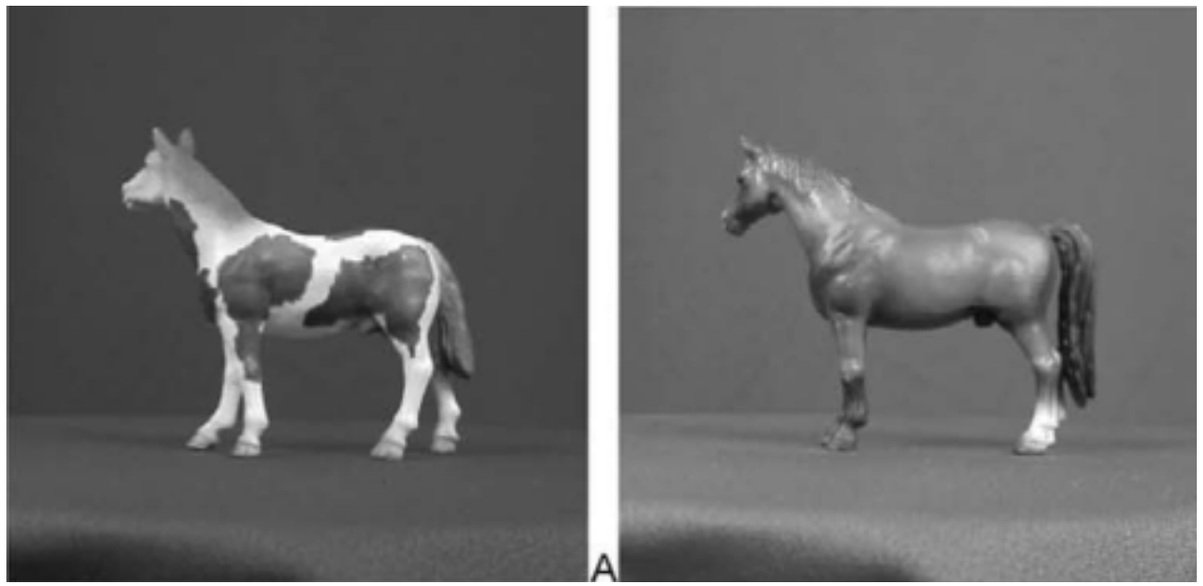
Application of Hough Transforms

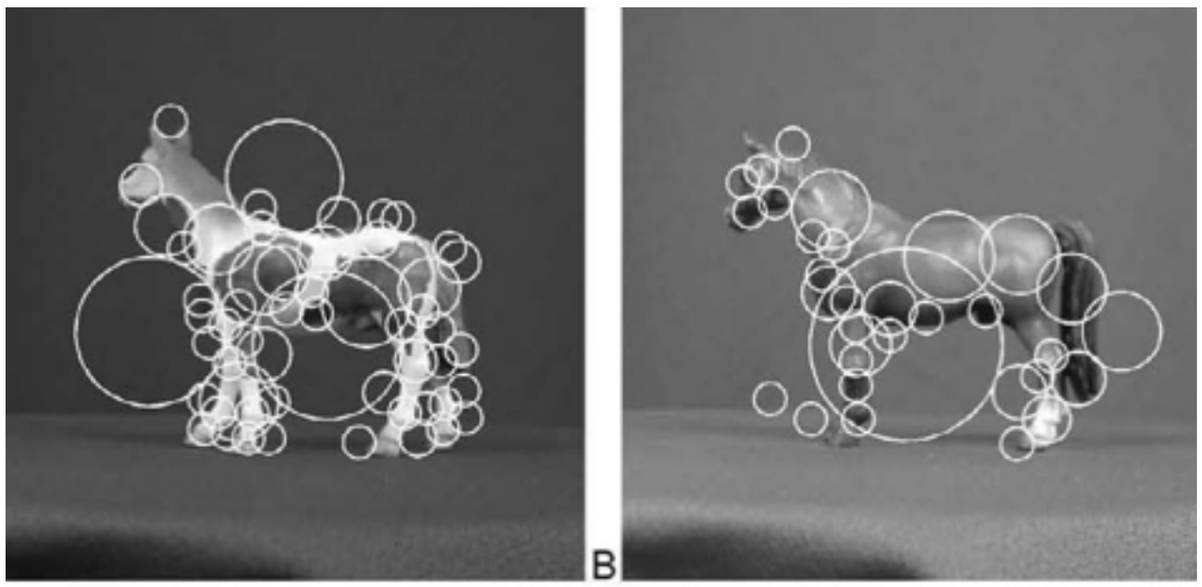
Detecting shape features



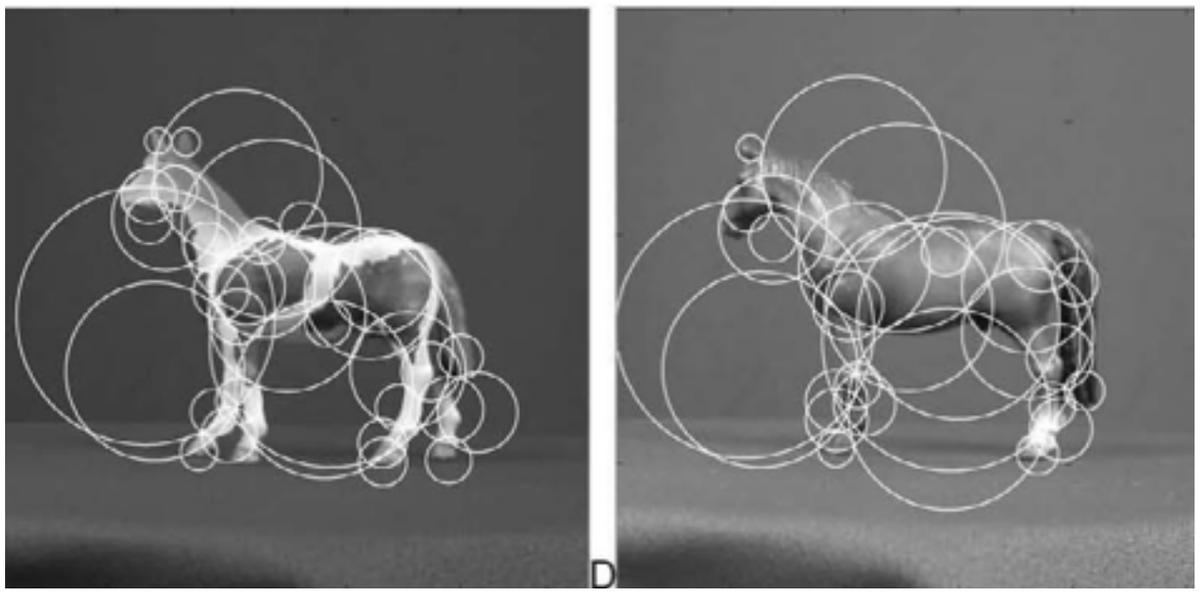
F. Jurie and C. Schmid, Scale-invariant shape features for recognition of object categories, CVPR 2004

Original images



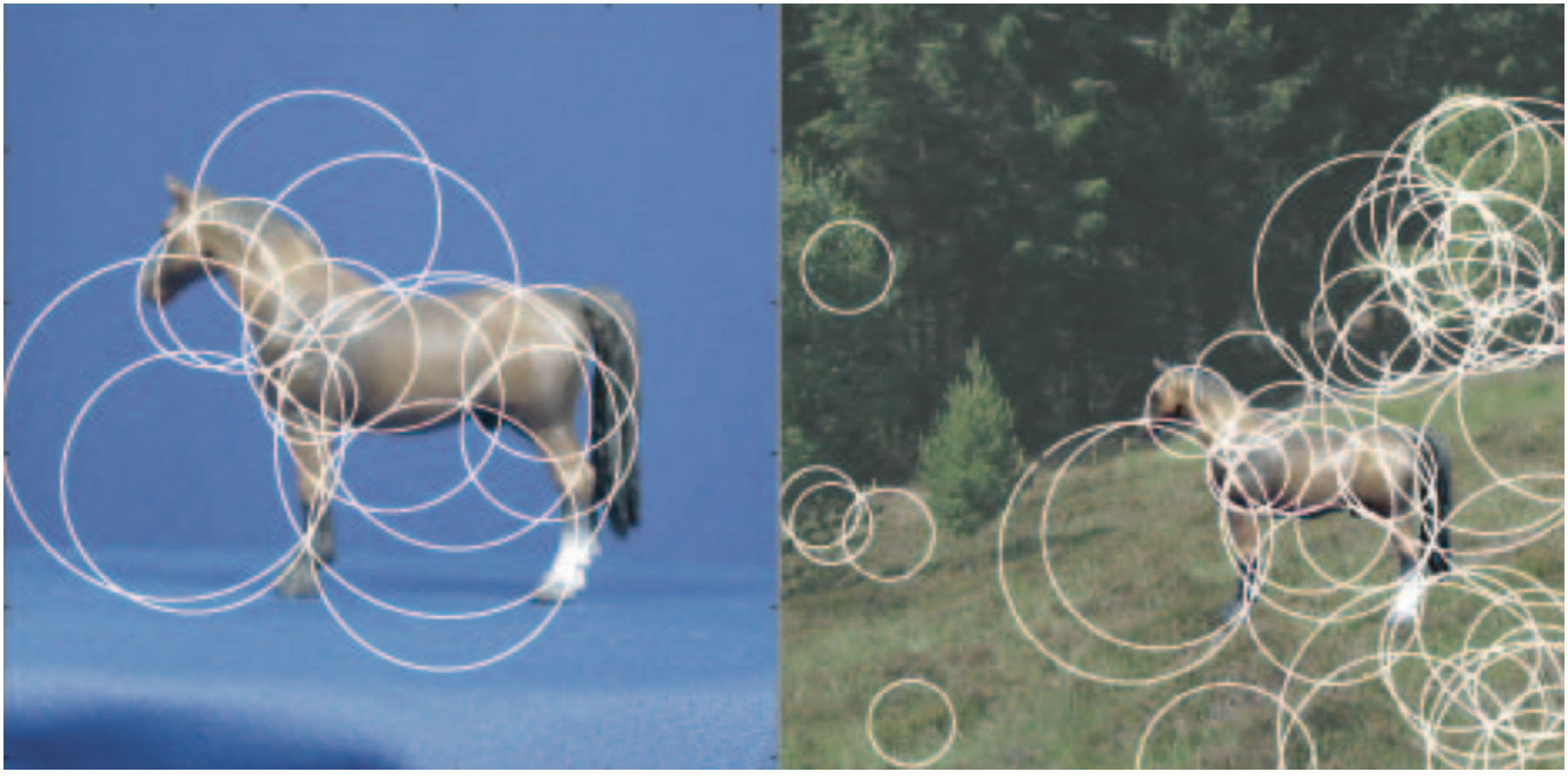


Laplacian circles



Hough-like circles

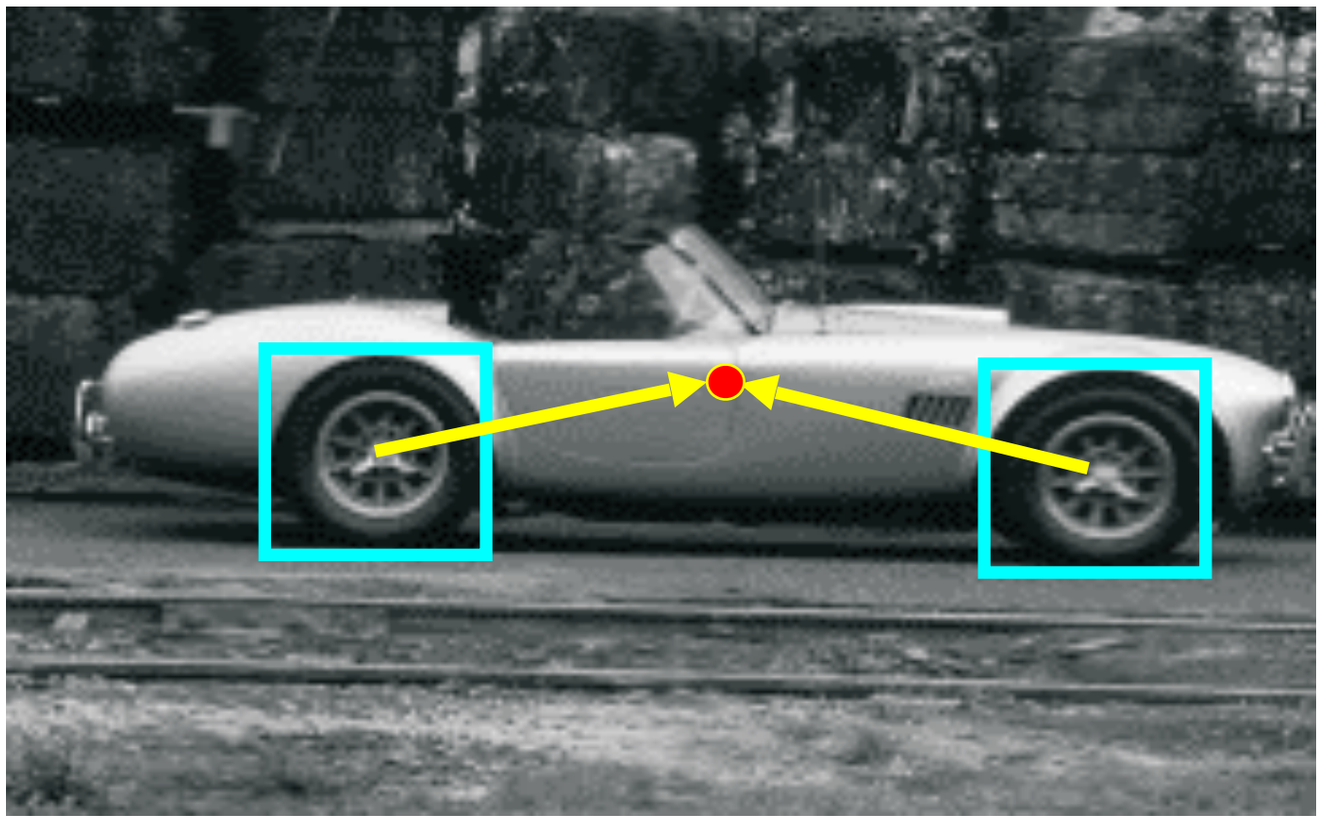
Which feature detector is more consistent?



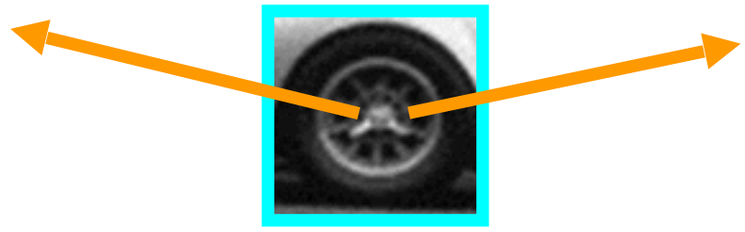
Robustness to scale and clutter

Object detection

Index displacements by “visual codeword”



training image

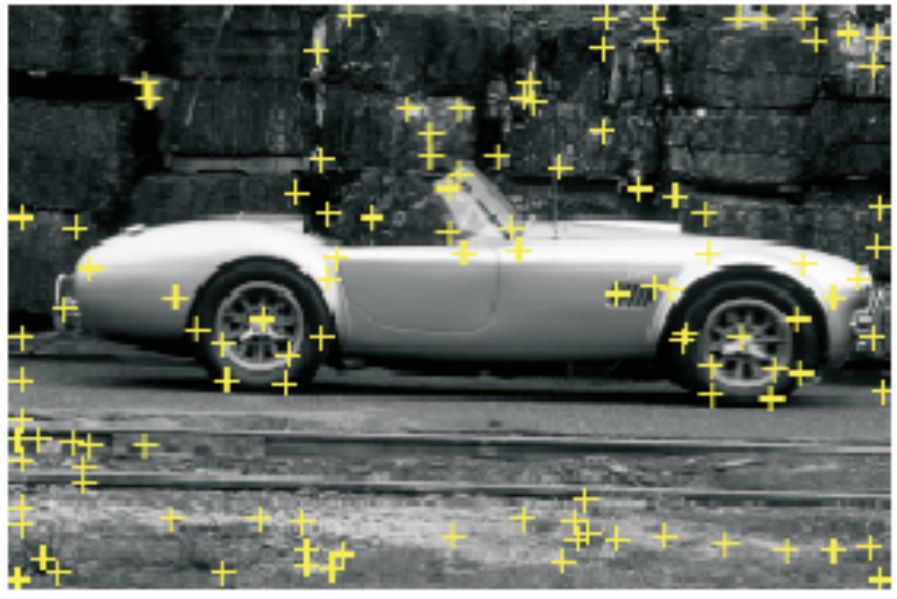


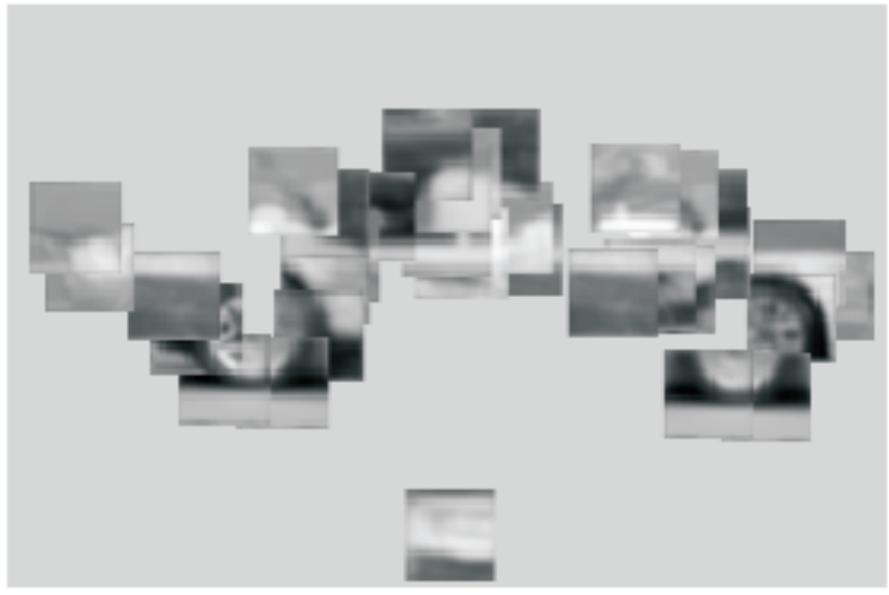
visual codeword with
displacement vectors

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

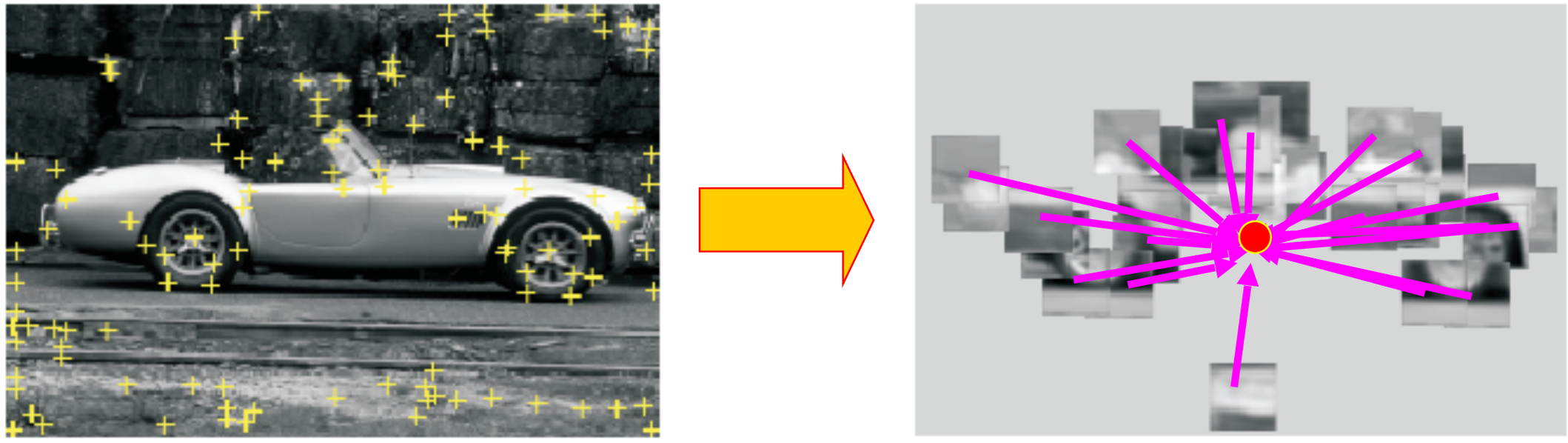
1. get features



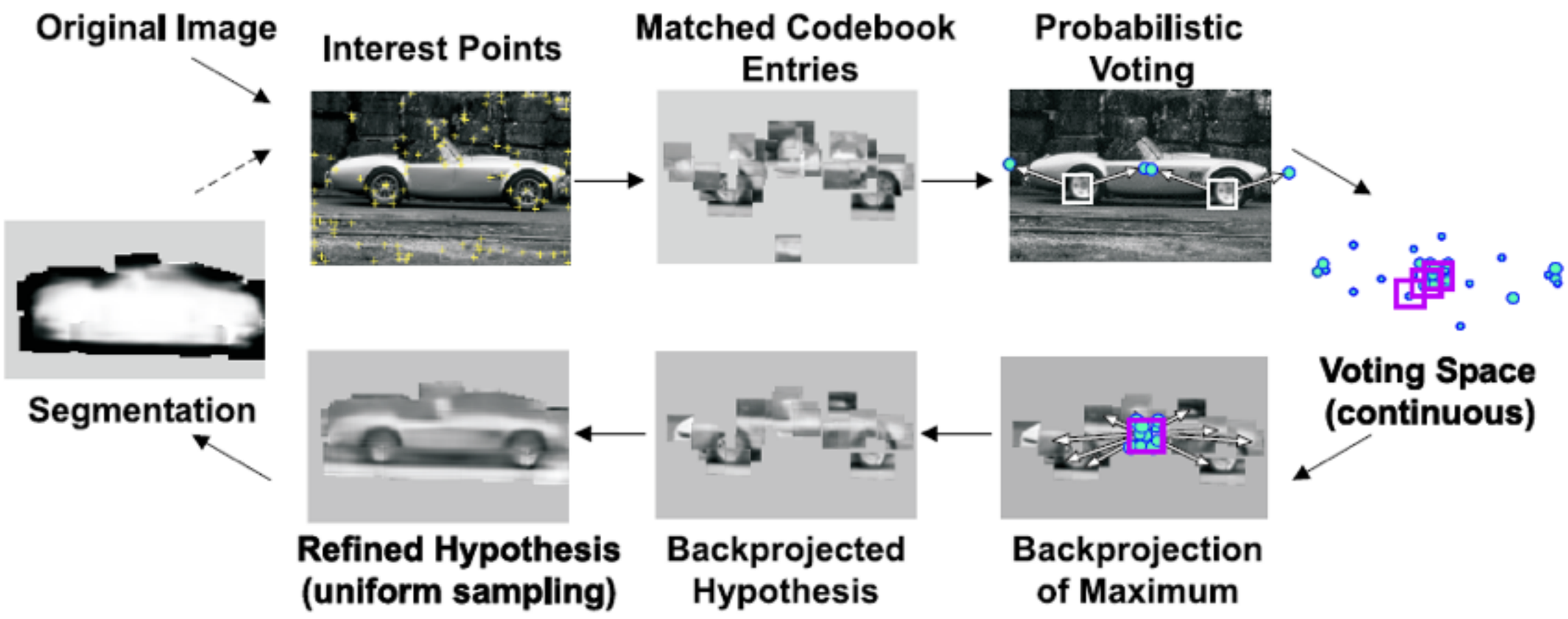


Train phase

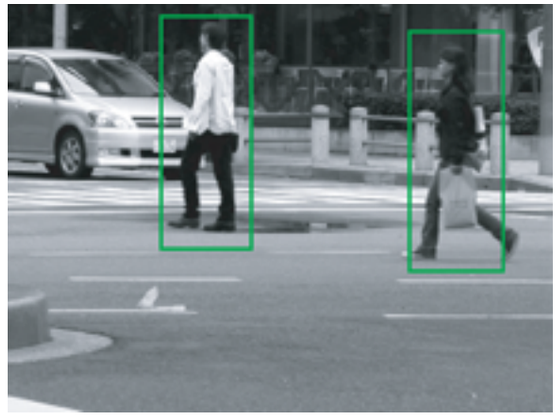
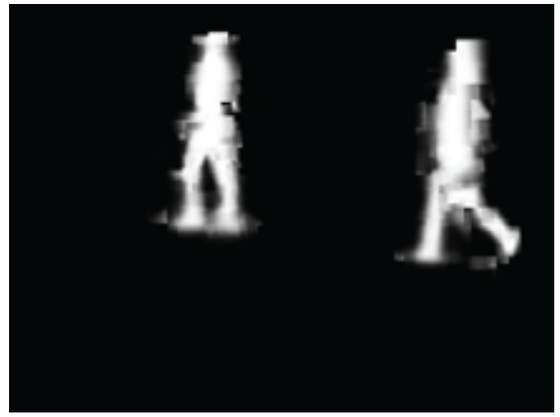
2. store displacements



Test phase

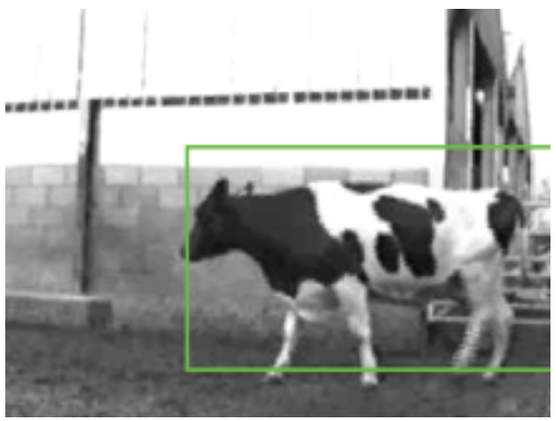

Original Image
Interest Points
Matched Codebook Entries
Probabilistic Voting
Segmentation
Refined Hypothesis (uniform sampling)
Backprojected Hypothesis
Backprojection of Maximum
Voting Space (continuous)



www.solomotobmw.it






The Hough transform …

Deals with occlusion well? 

Detects multiple instances? 

Robust to noise? 

Good computational complexity? 

Easy to set parameters?