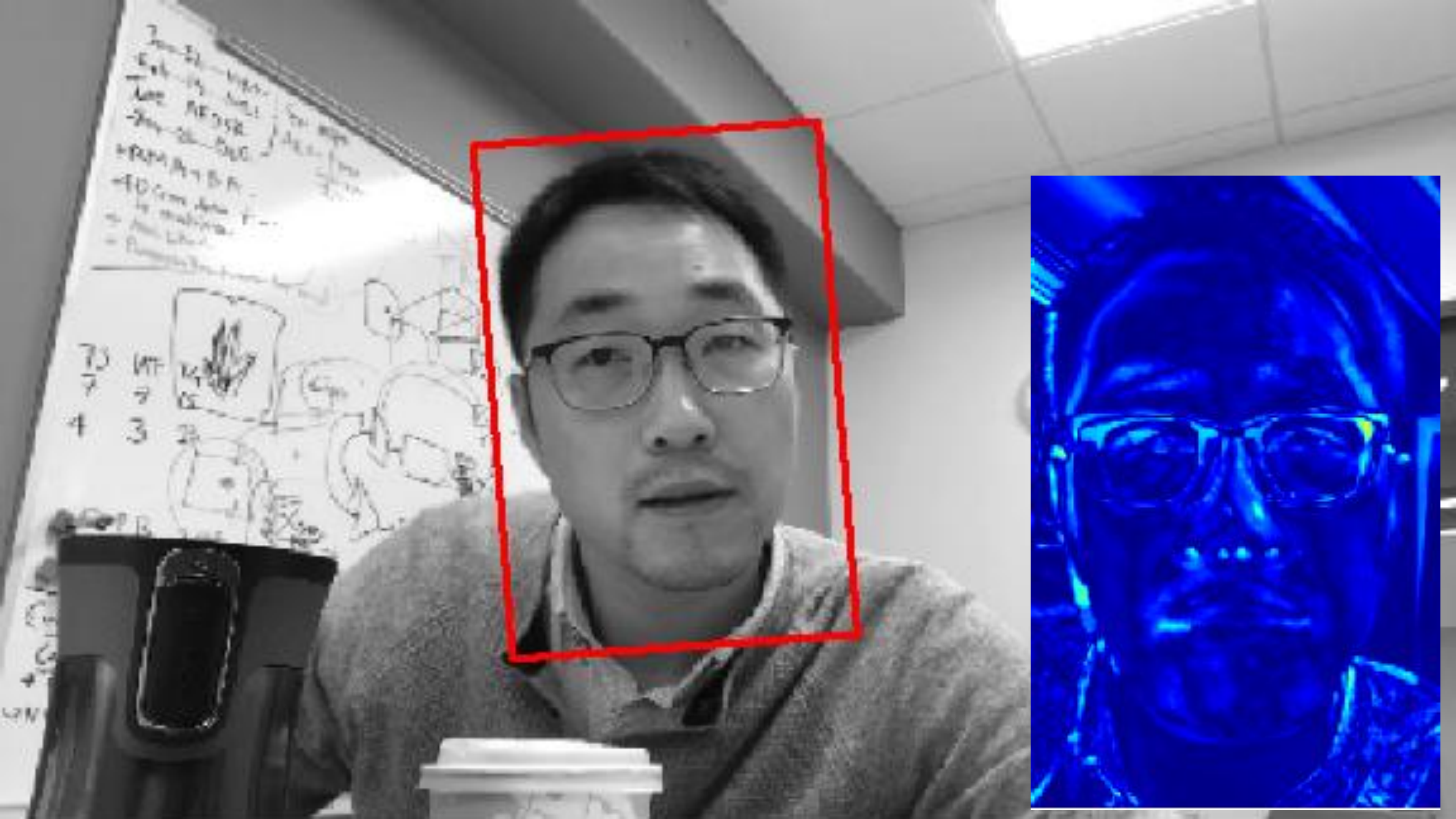
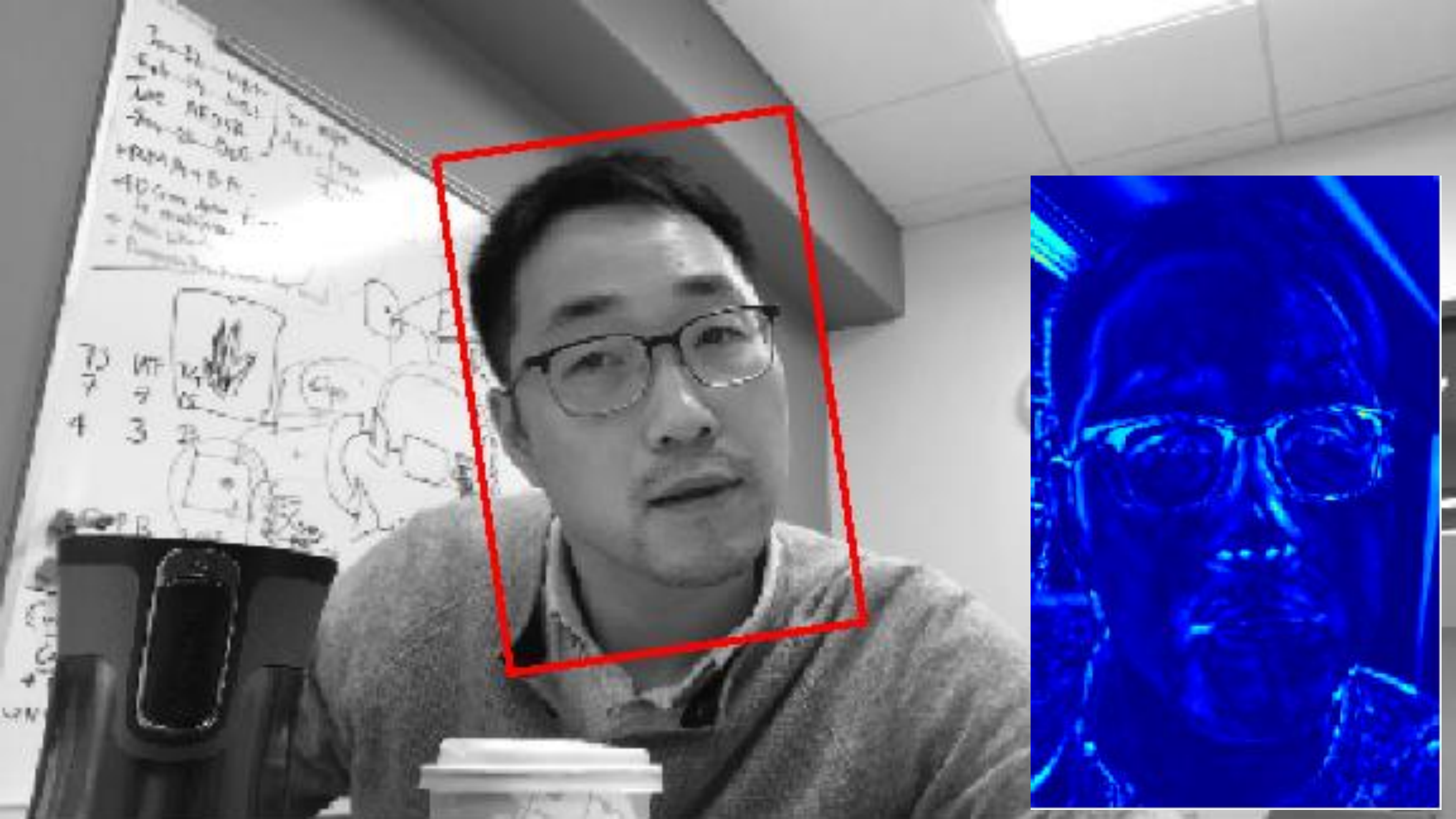
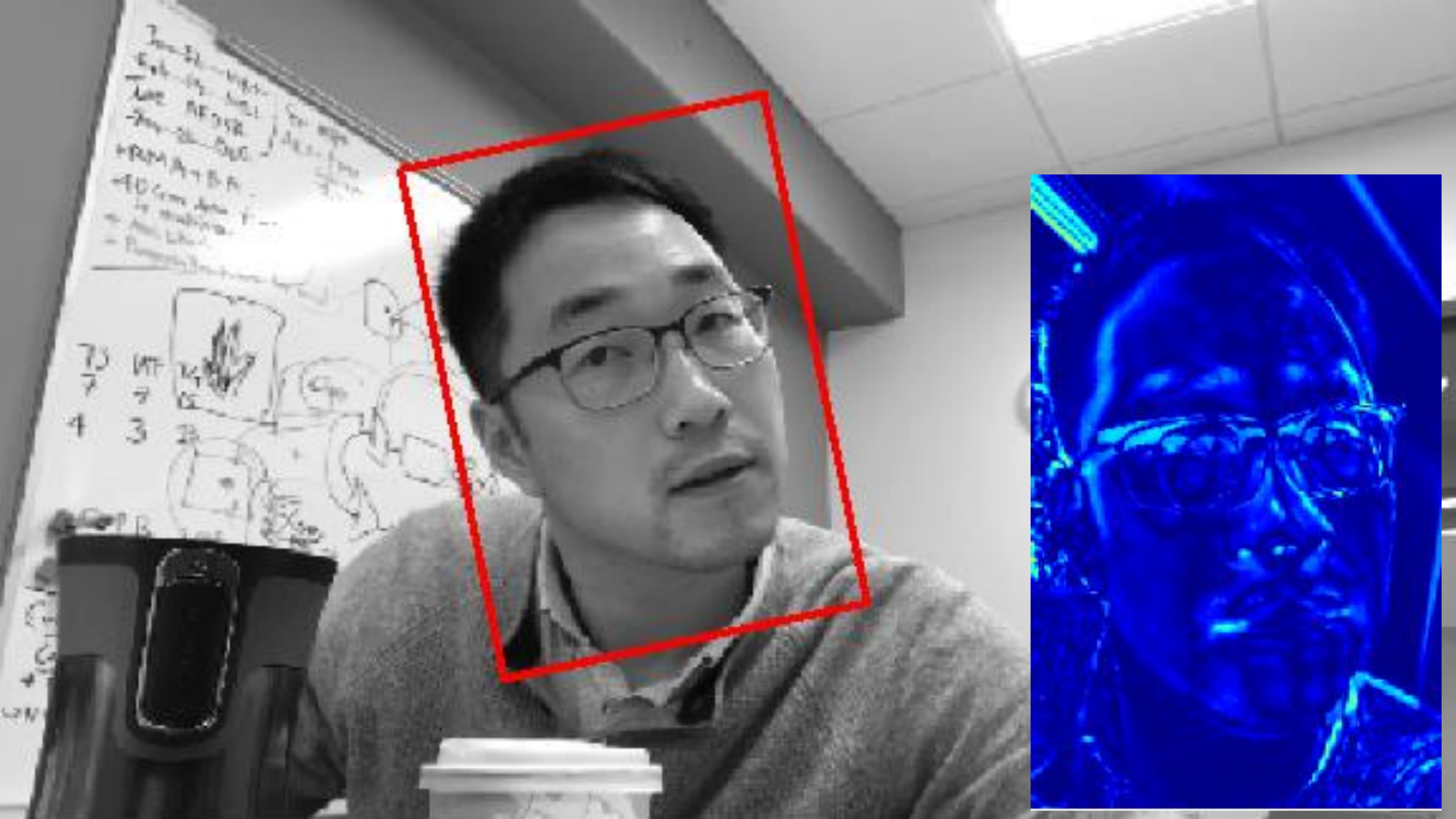
NONPARAMETRIC TRACKING

HYUN SOO PARK








Nationale-Nederlanden
jackpot €5 milj.
ING

NONRIGID TRACKING

Desired algorithm:

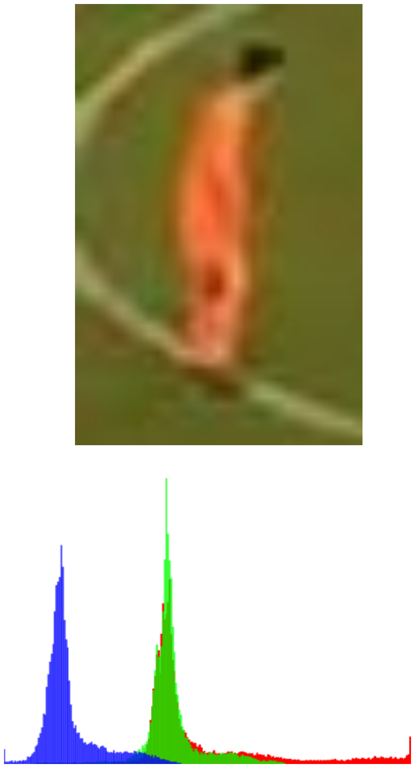
- Invariant to nonrigid transformation

HOG, SIFT, and parametric image alignment do not work.

NONRIGID TRACKING

Desired algorithm:

- Invariant to nonrigid transformation
  - Color histogram



NONRIGID TRACKING

Desired algorithm:

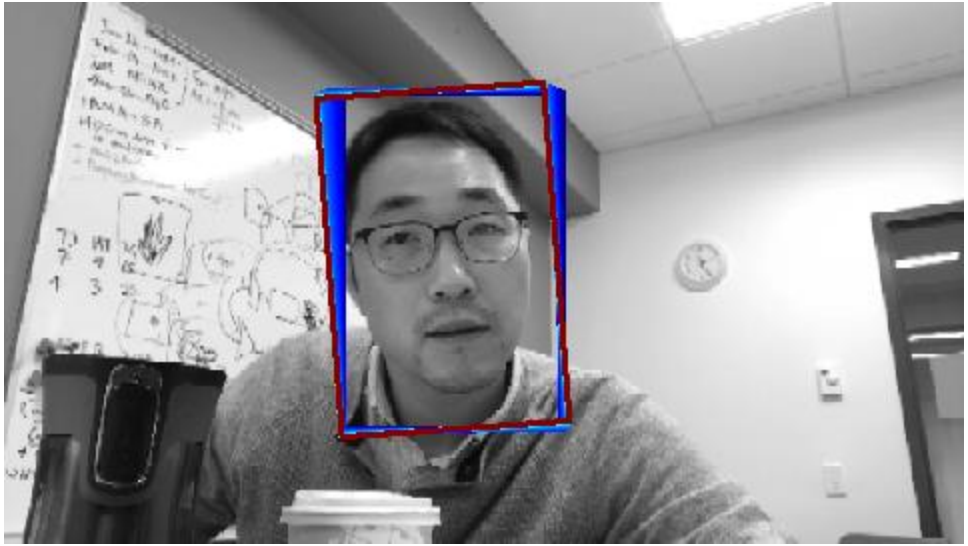
- Invariant to nonrigid transformation
  - Color histogram
- Computationally efficient

Sliding window reqiures too much computation.

NONRIGID TRACKING

Desired algorithm:

- Invariant to nonrigid transformation
  - Color histogram
- Computationally efficient
  - Gradient based tracking



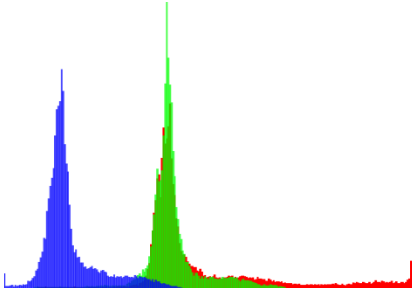
Nonrigid Tracking

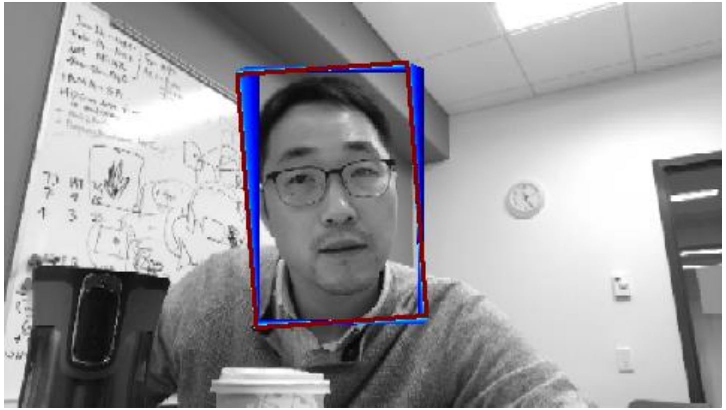
Desired algorithm:

- Invariant to nonrigid transformation
  - Color histogram (not spatial rep.)

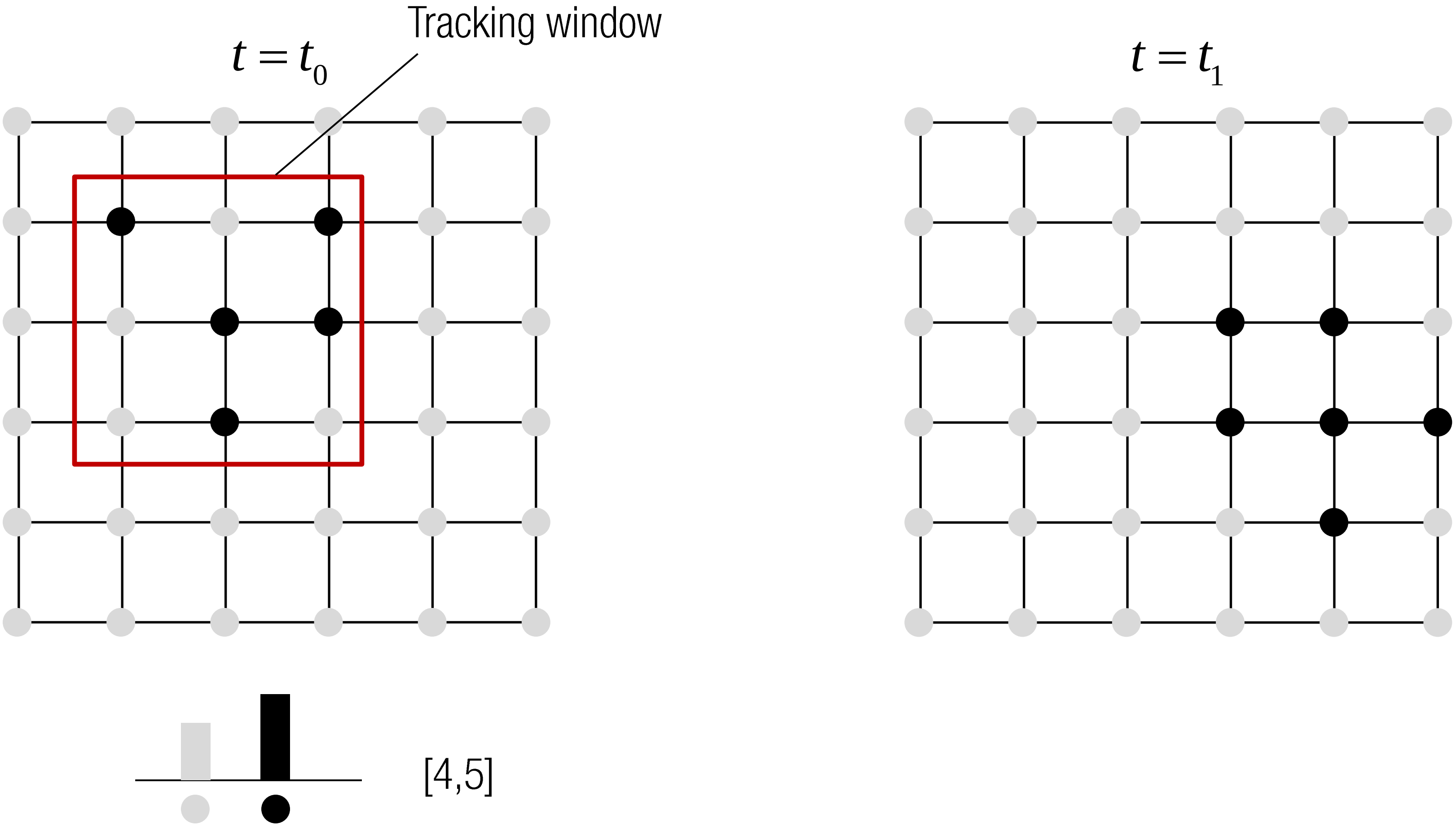
- Computationally efficient
  - Gradient based tracking (relying on spatial rep.)

Contradictory!

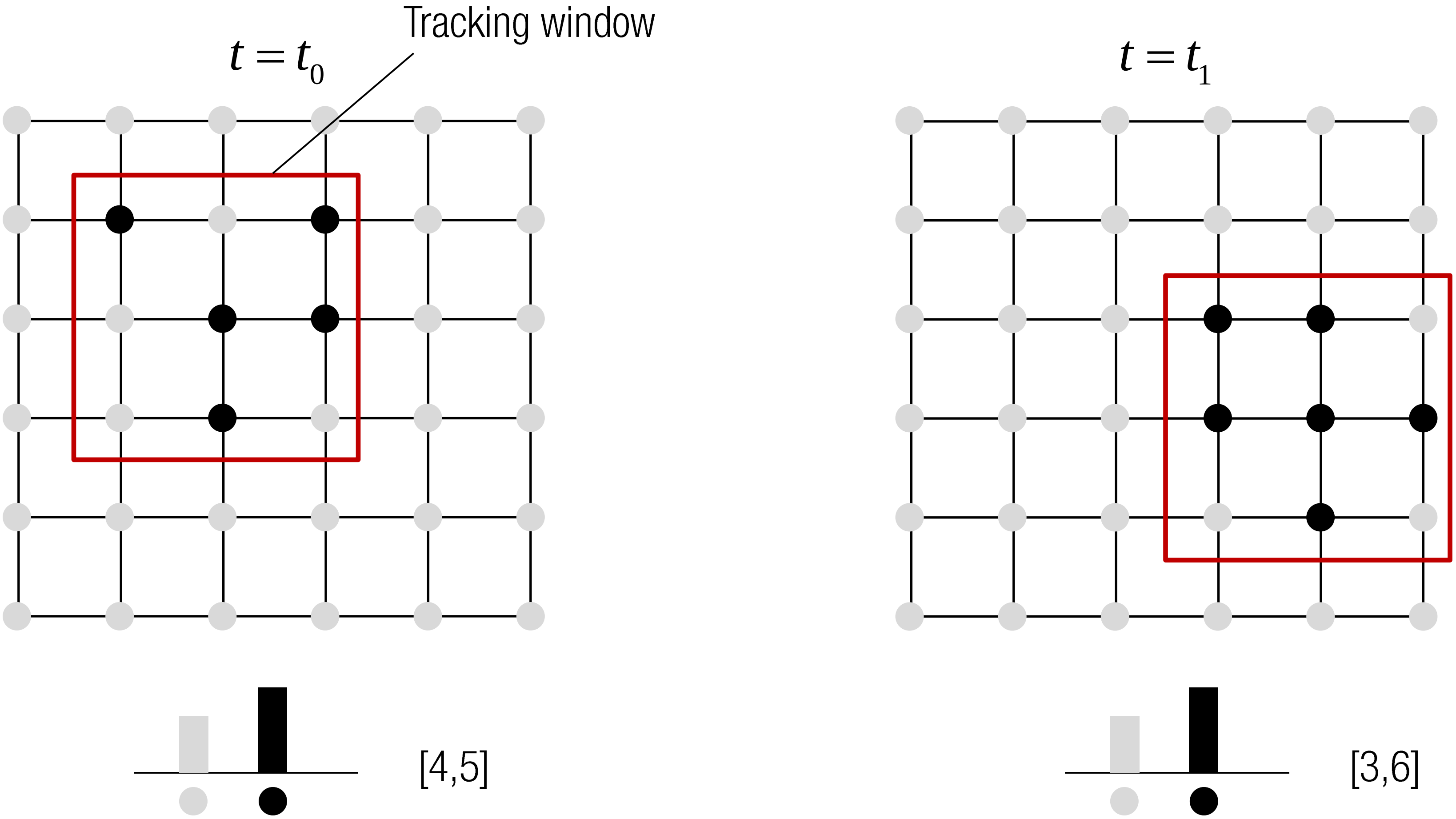




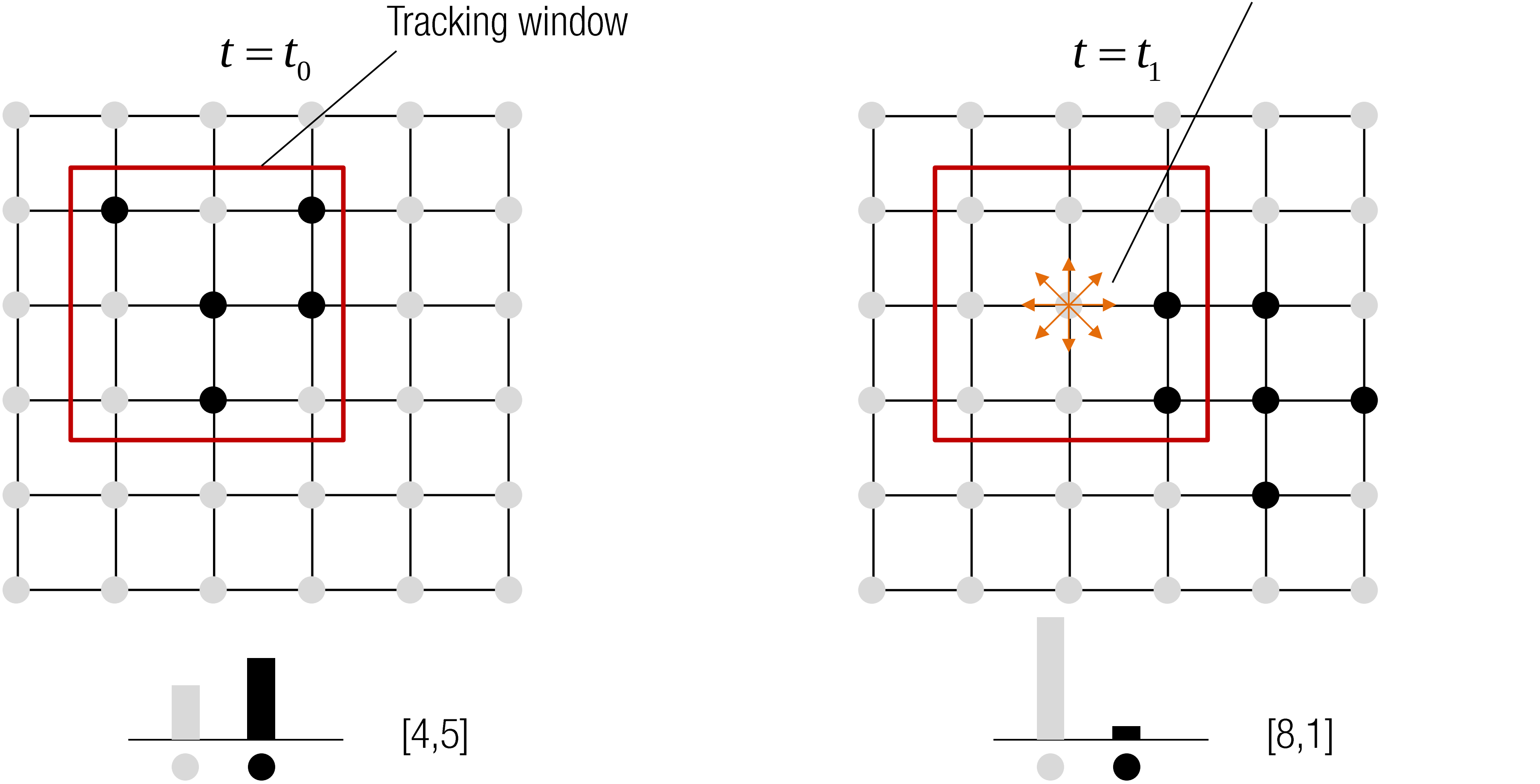
NONRIGID TRACKING FOR BINARY IMAGE



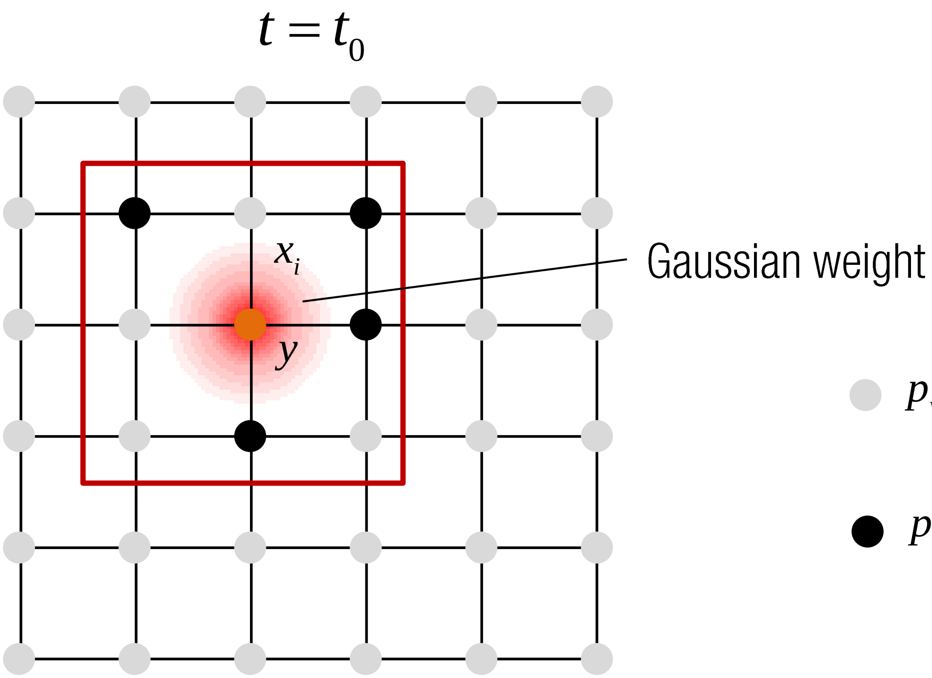
Nonrigid Tracking For Binary Image



Nonrigid Tracking For Binary Image

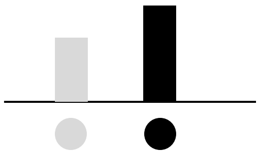


Continuous Representation: Weighted Histogram

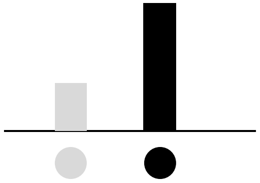


$$p_{white} = C \sum e^{-\frac{(y-x_i)^2}{2\sigma^2}} \delta\left(b(x_i) - 0\right)$$

$$p_{black} = C \sum e^{-\frac{(y-x_i)^2}{2\sigma^2}} \delta\left(b(x_i) - 1\right)$$



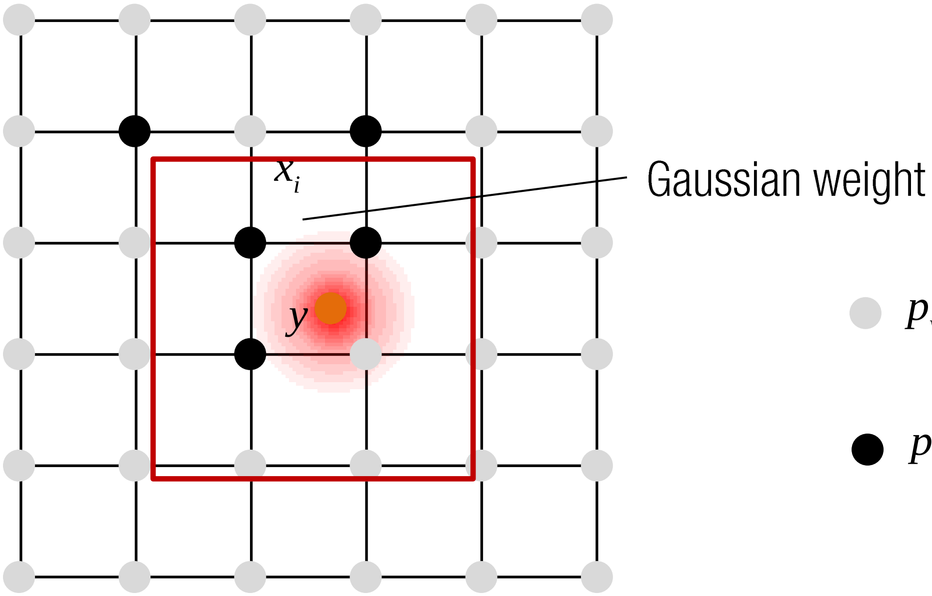
[4,5]



[3.2,5.6]

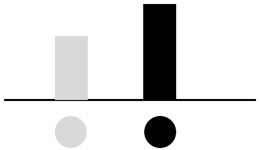
Continuous Representation: Weighted Histogram

$t = t_0$

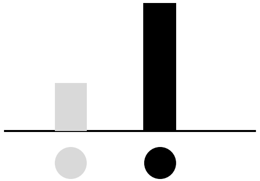


$$p_{white} = C \sum e^{-\frac{(y - x_i)^2}{2\sigma^2}} \delta\left(b(x_i) - 0\right)$$

$$p_{black} = C \sum e^{-\frac{(y - x_i)^2}{2\sigma^2}} \delta\left(b(x_i) - 1\right)$$

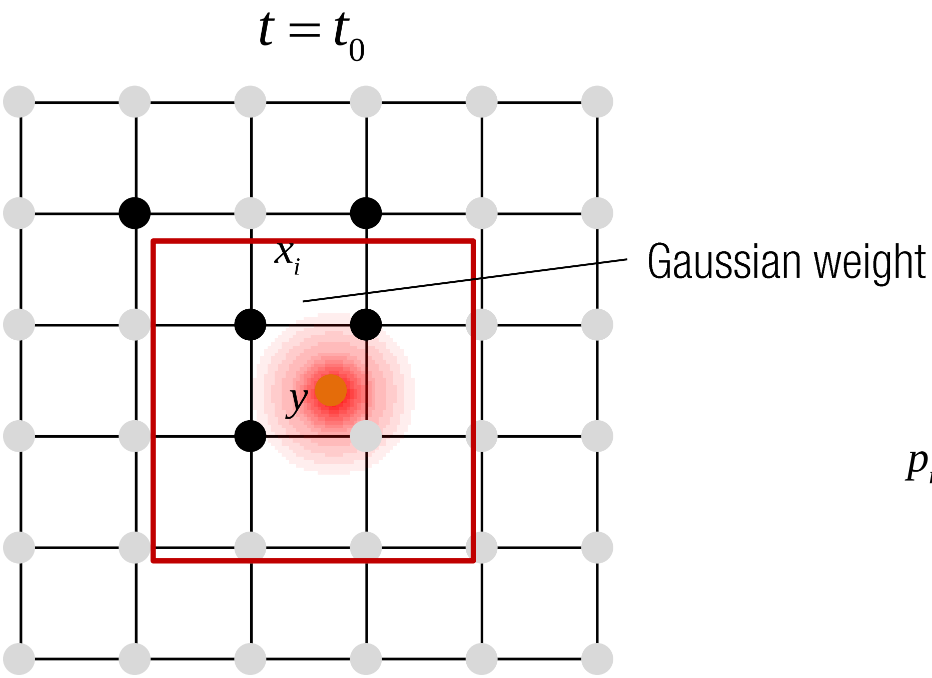


[4,5]

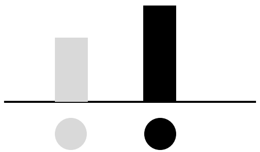


[3.2,5.6]

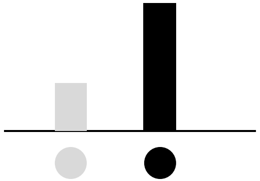
Continuous Representation: Weighted Histogram



$$p_m = C \sum_i e^{-\frac{(y - x_i)^2}{2\sigma^2}} \delta\left(b(x_i) - m\right)$$

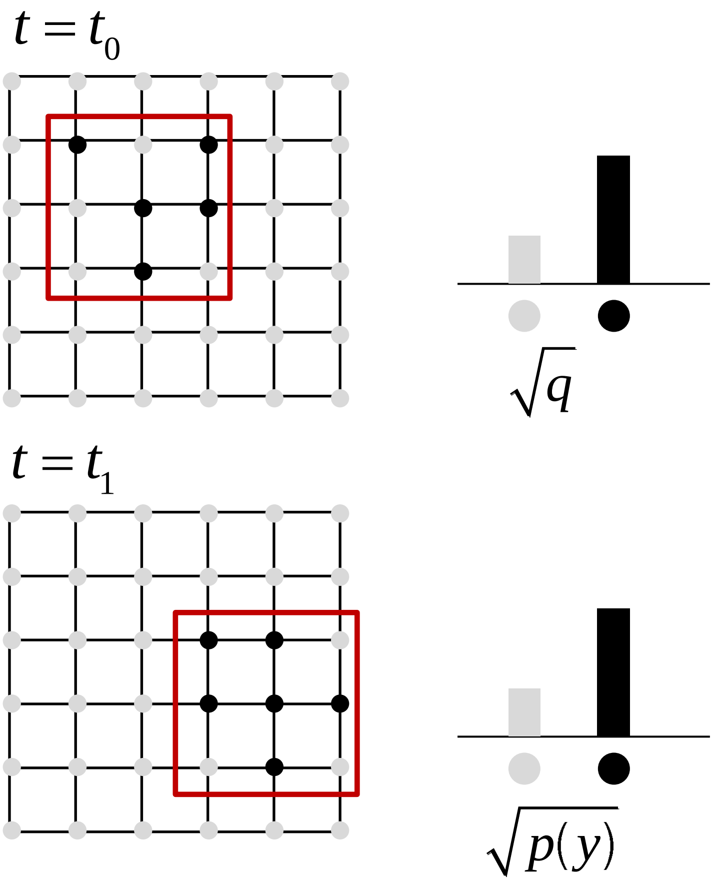


[4,5]



[3.2,5.6]

HISTOGRAM MATCH



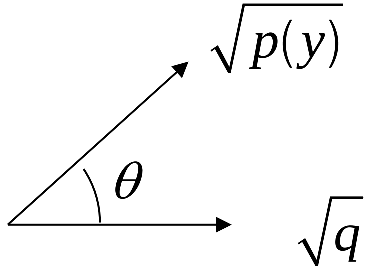
[3.2,5.6]

[3.1,5.9]

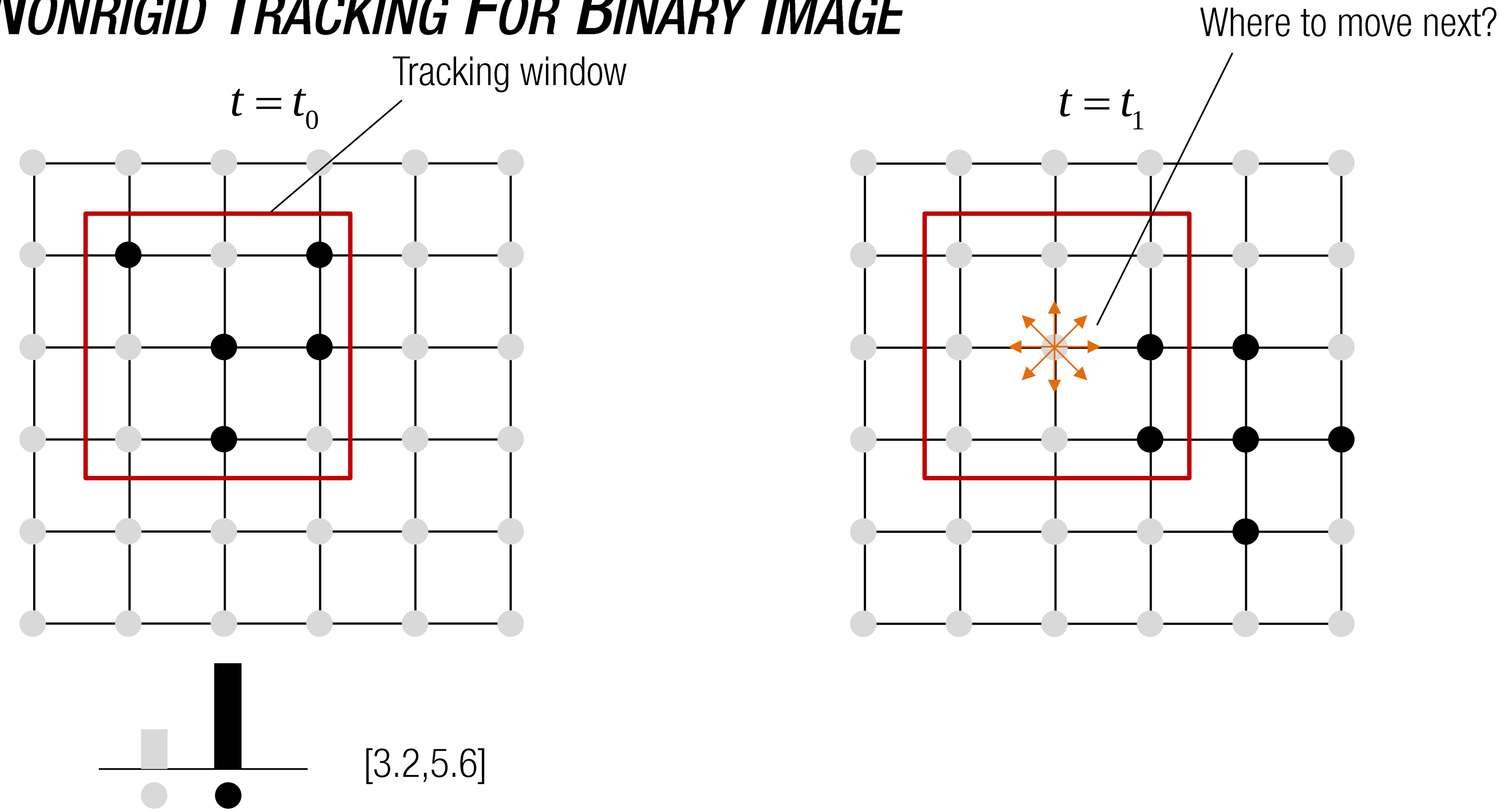
Bhattacharyya coefficient:
A measure of similarity of prob. dist.

$$\rho(y) = \left[\, p(y), q \,\right] = \sum_m \sqrt{p_m(y) q_m}$$

Cosine distance between prob. dist.



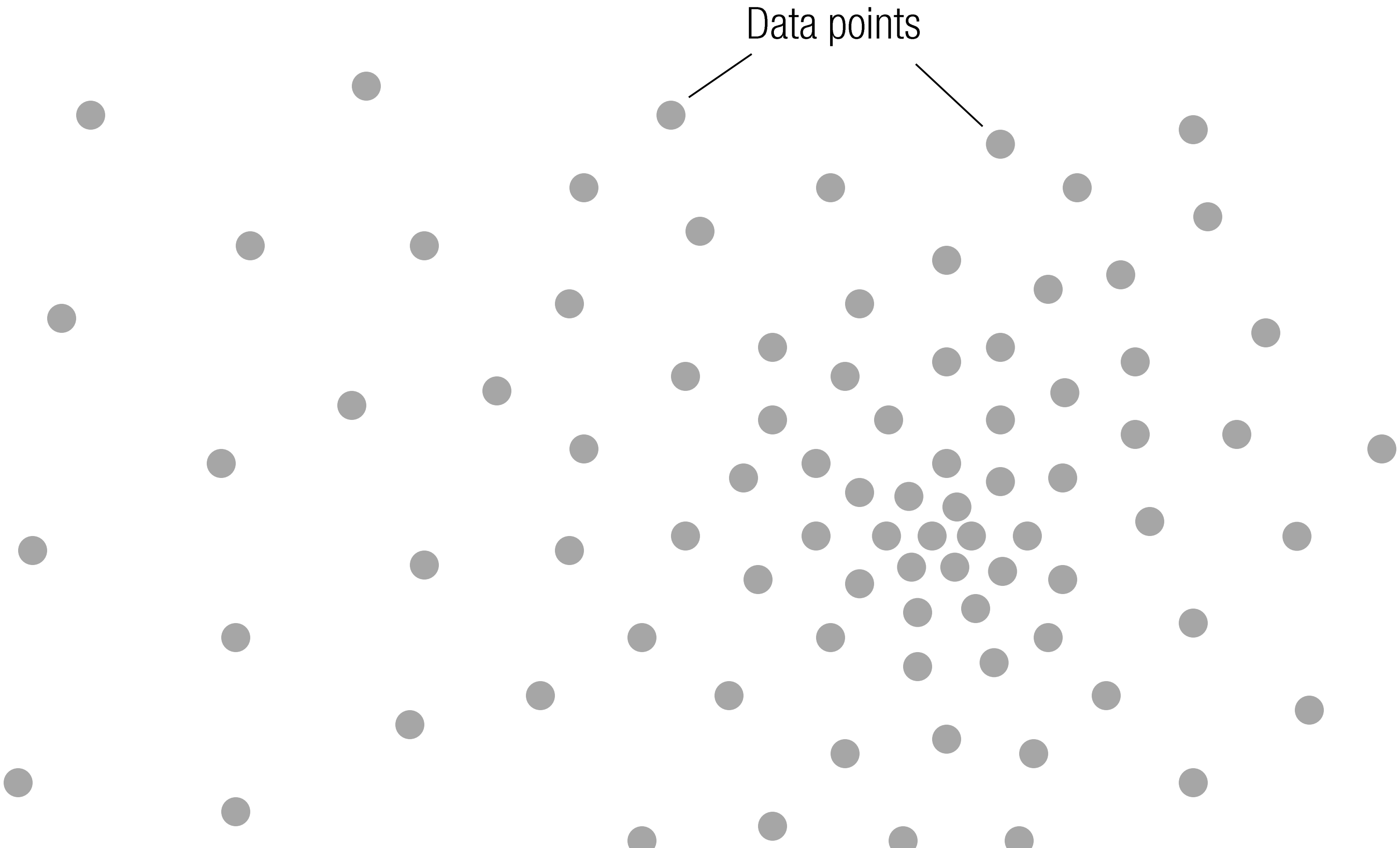
Nonrigid Tracking For Binary Image



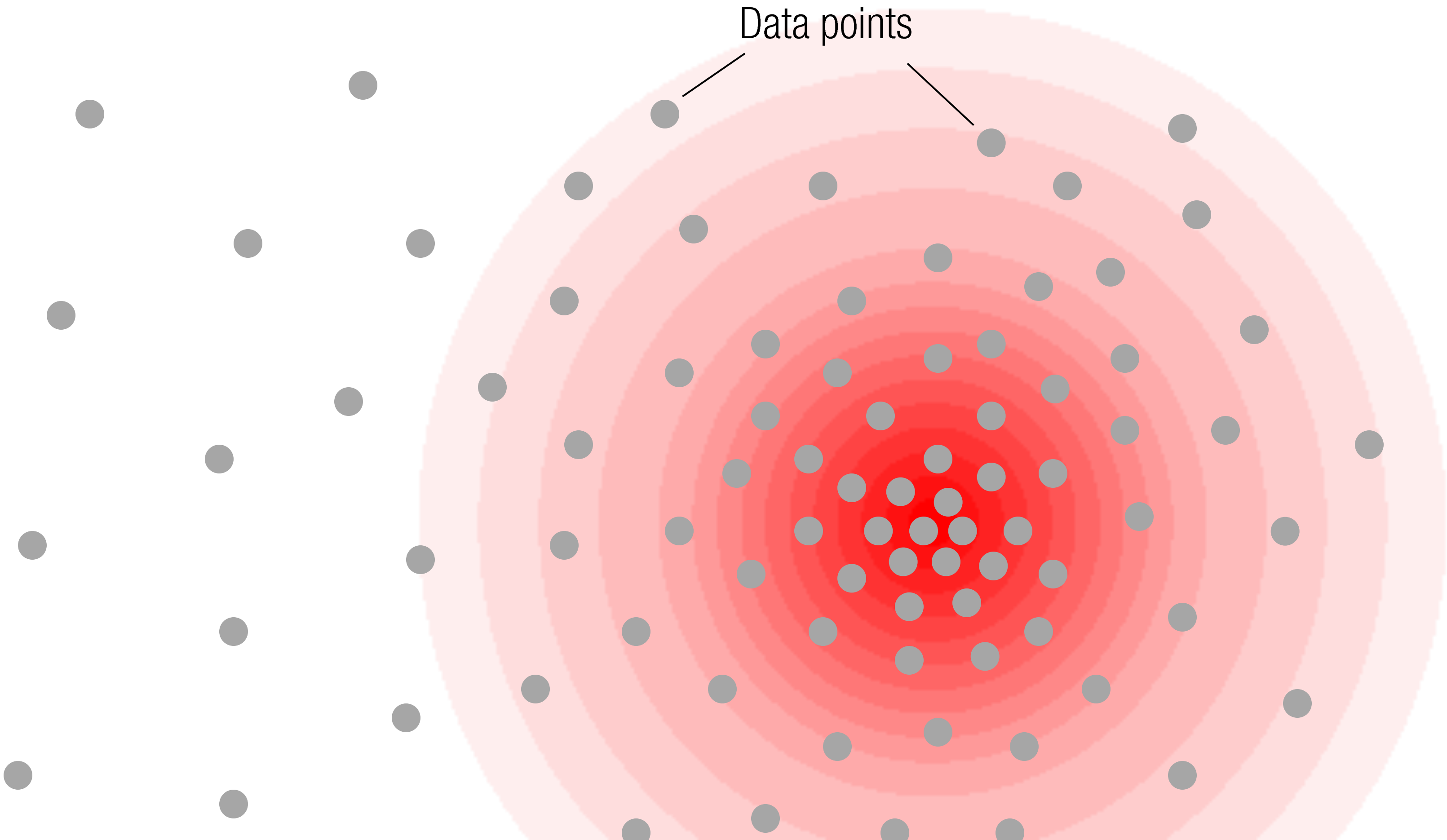
Meanshift Algorithm

Fukunaga and Hostetler, “The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition”, 1975

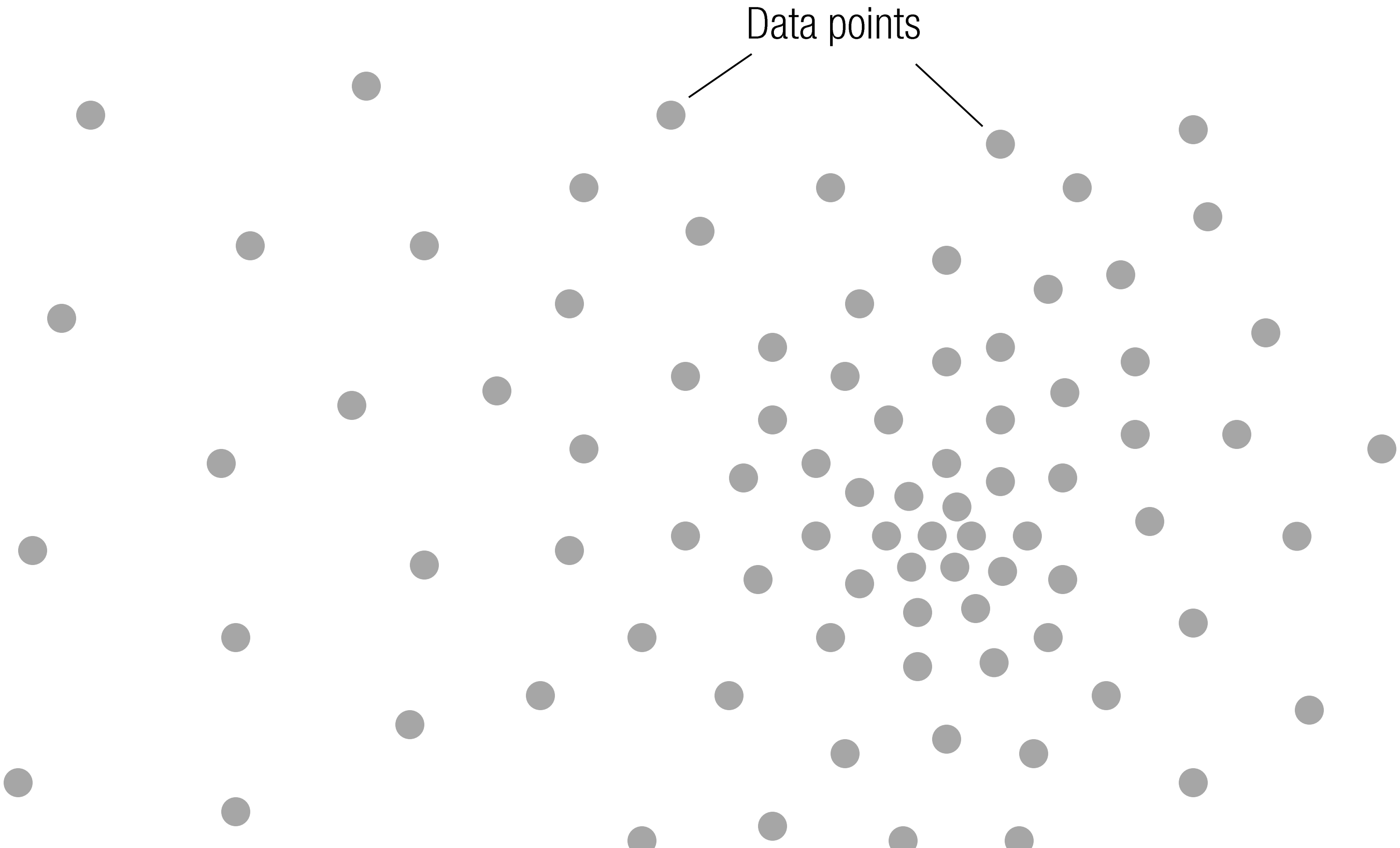
MODE-SEEKING



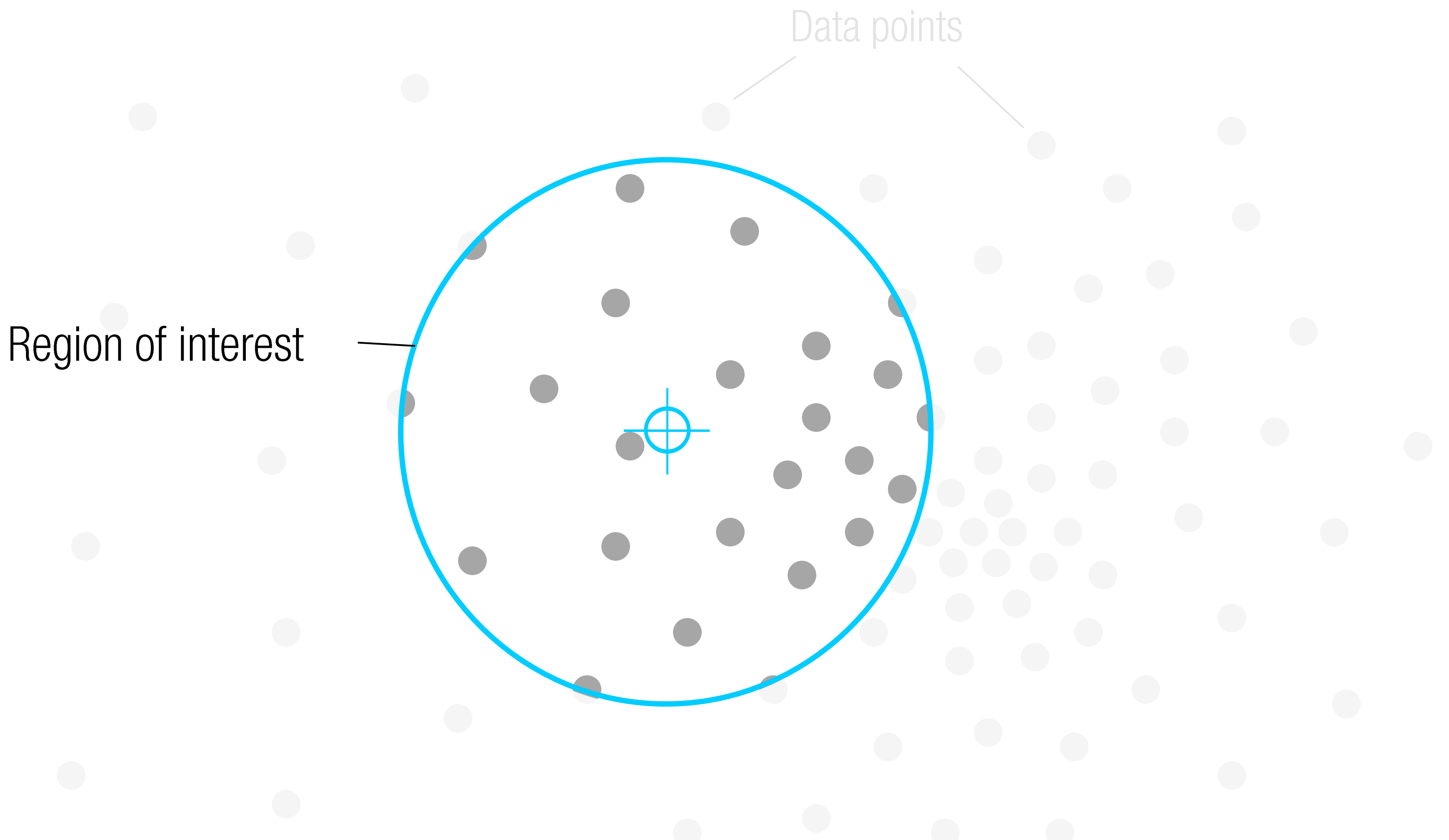
Mode-seeking



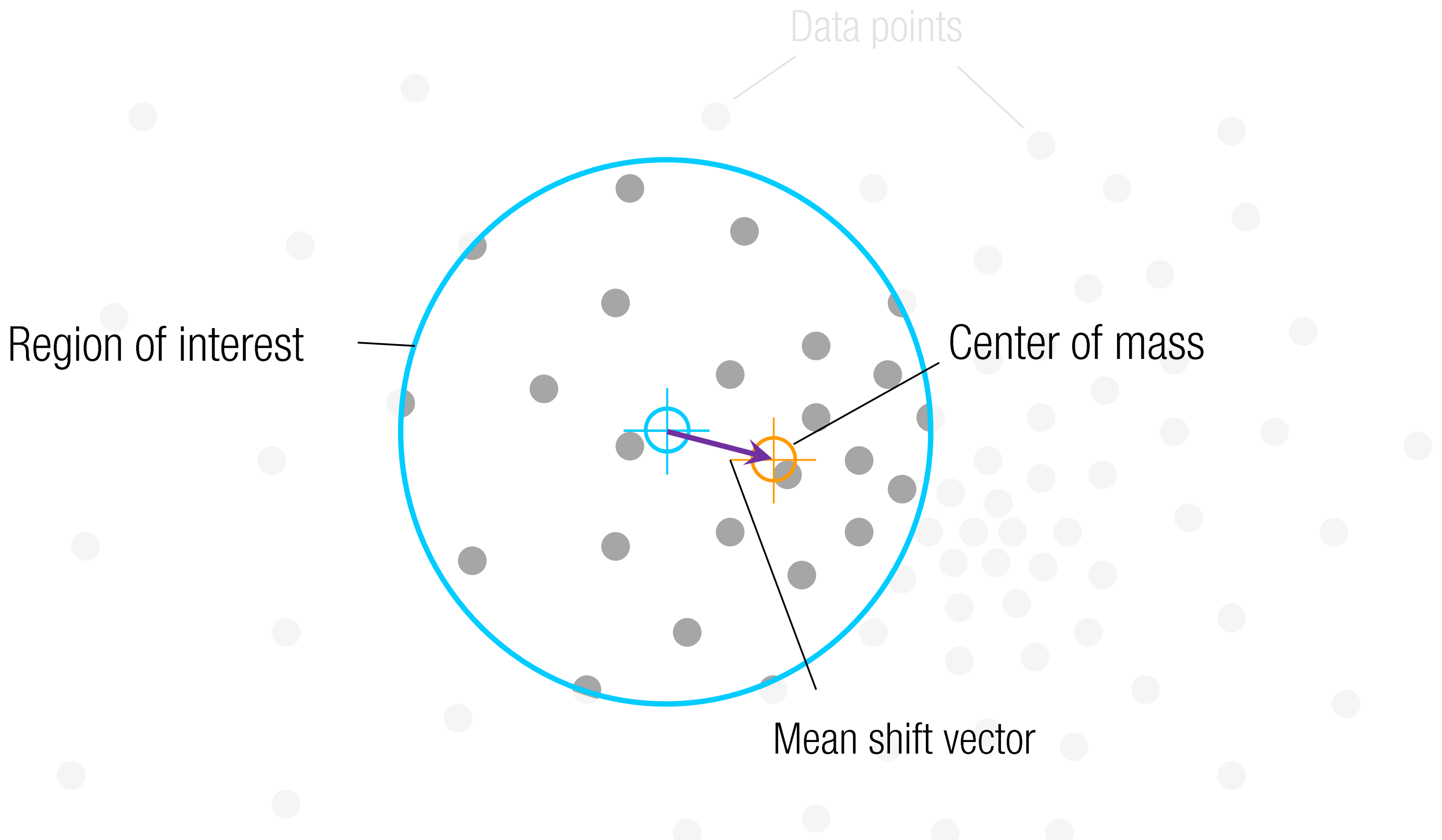
MODE-SEEKING



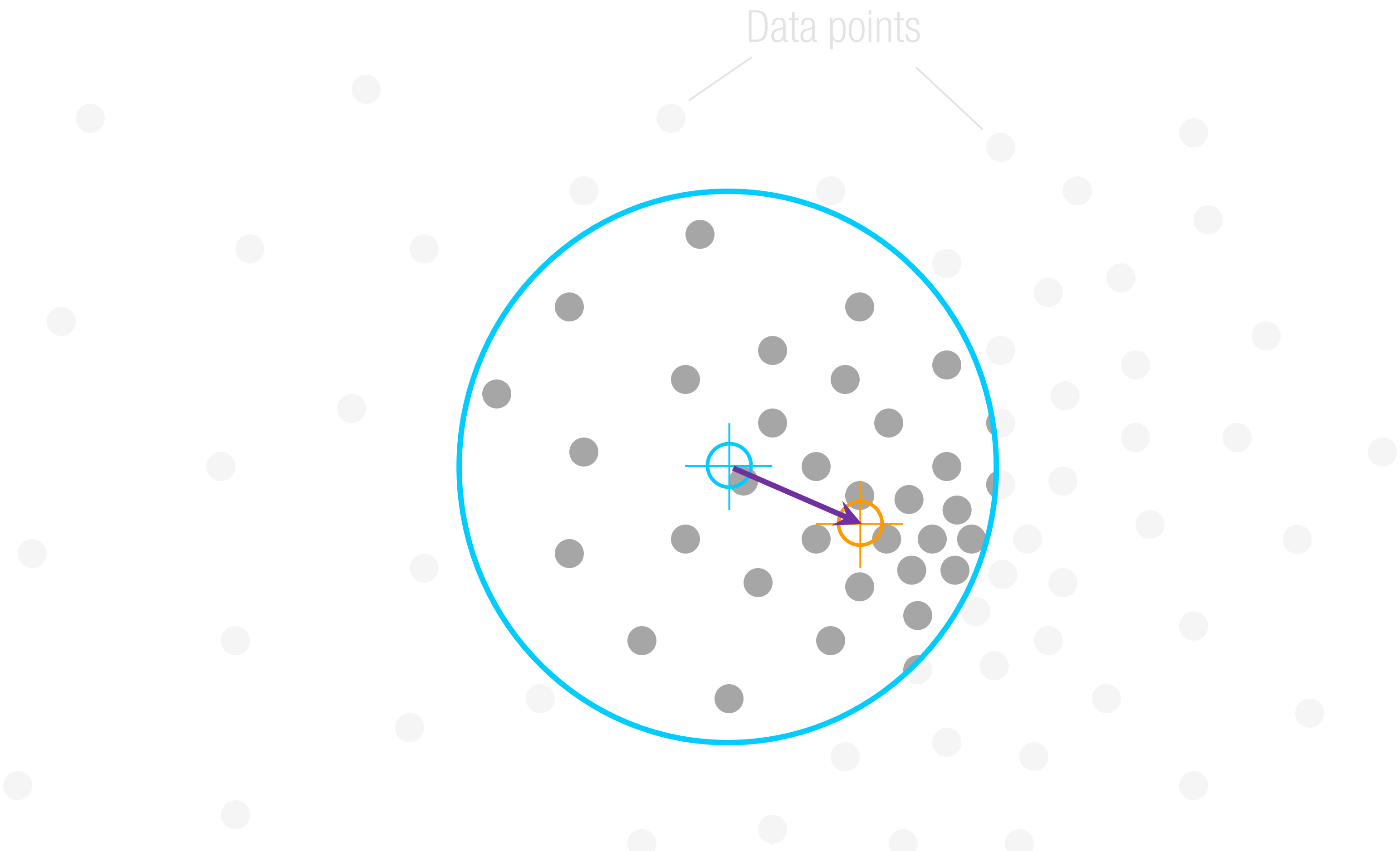
MODE-SEEKING



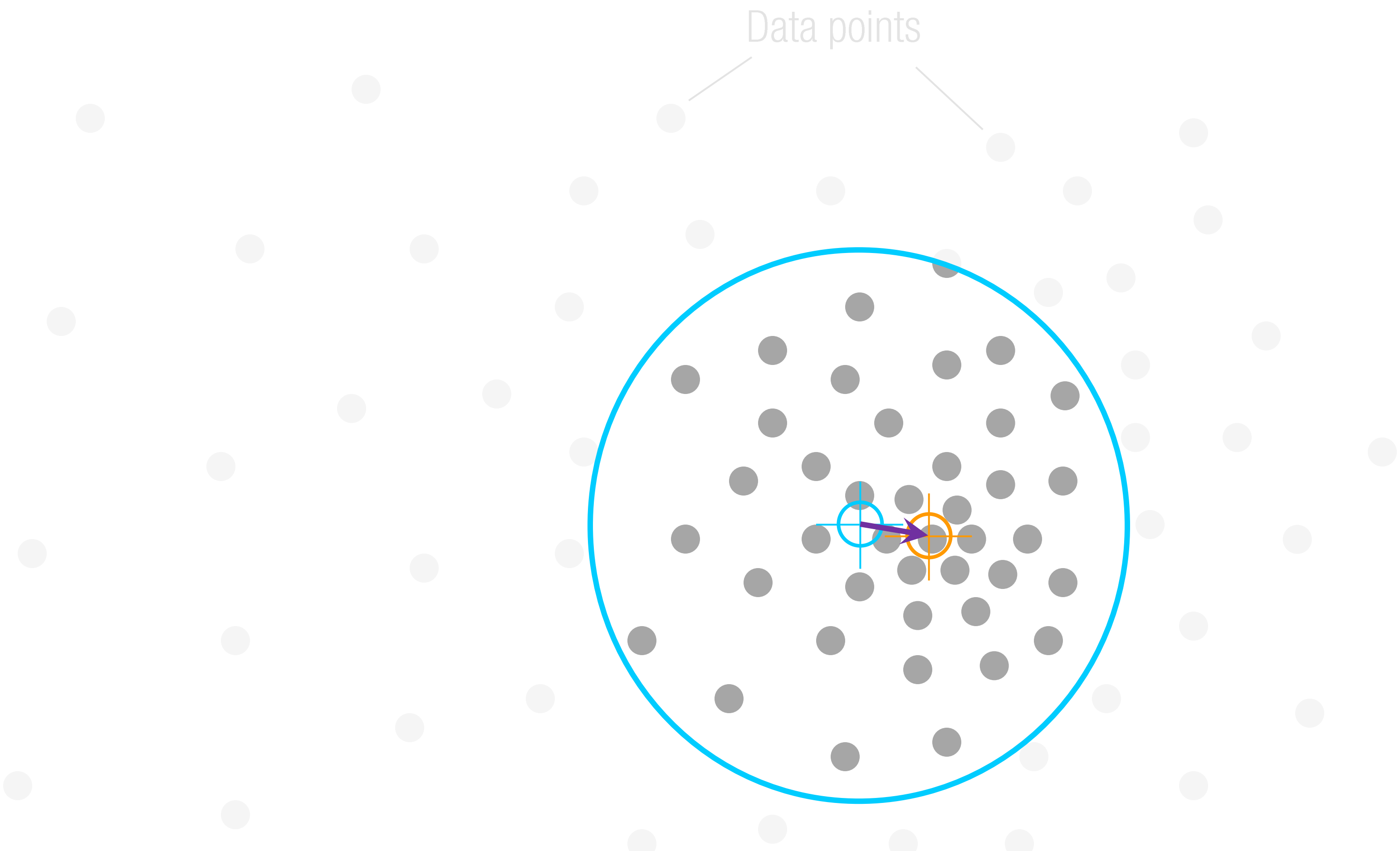
MODE-SEEKING



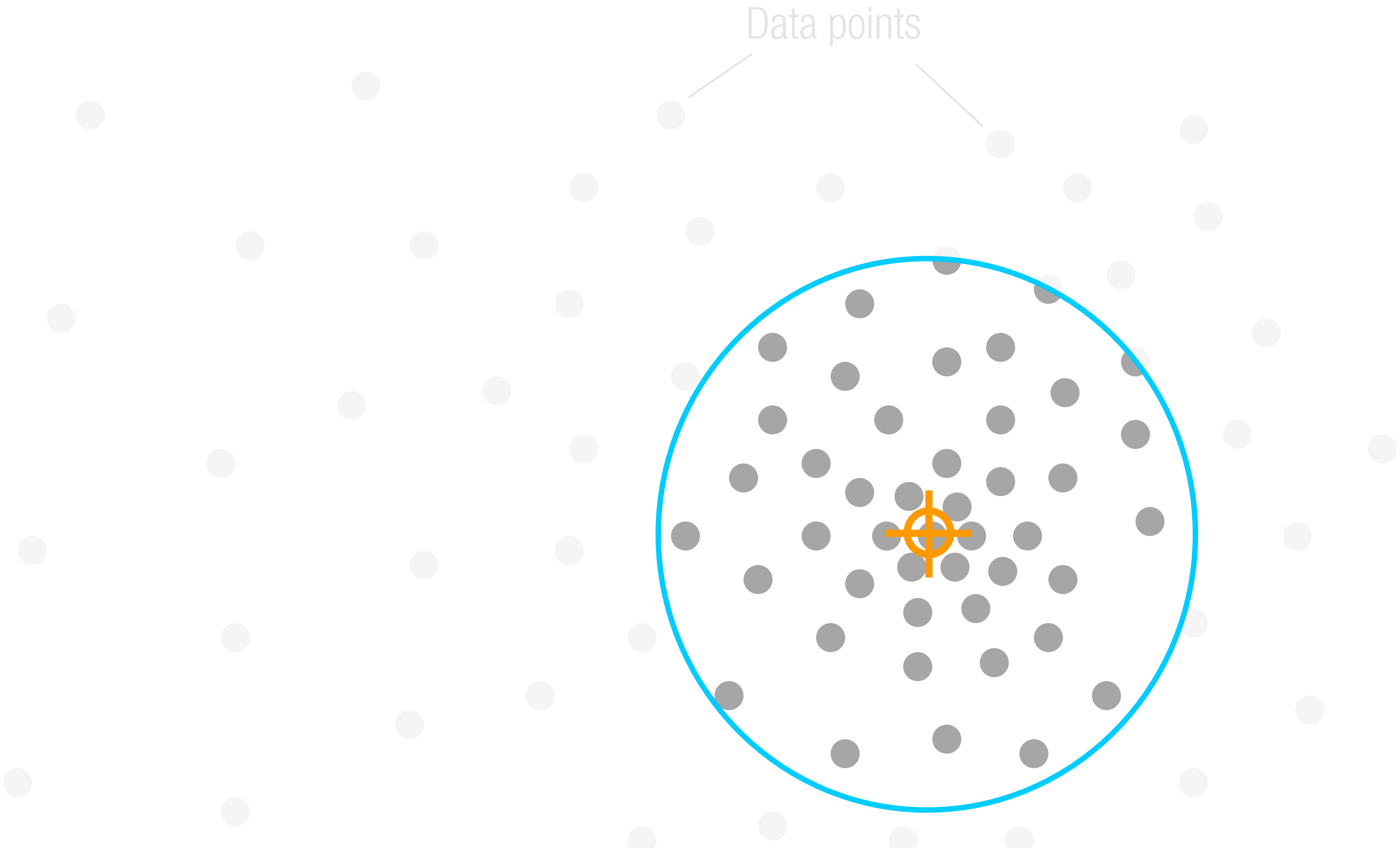
MODE-SEEKING



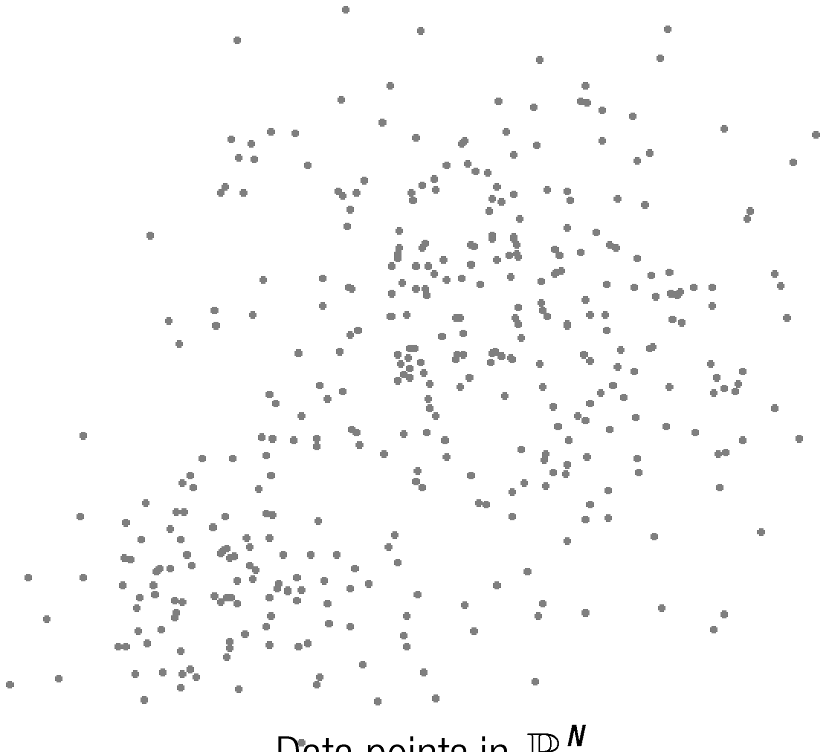
MODE-SEEKING



MODE-SEEKING



Data Density Estimation

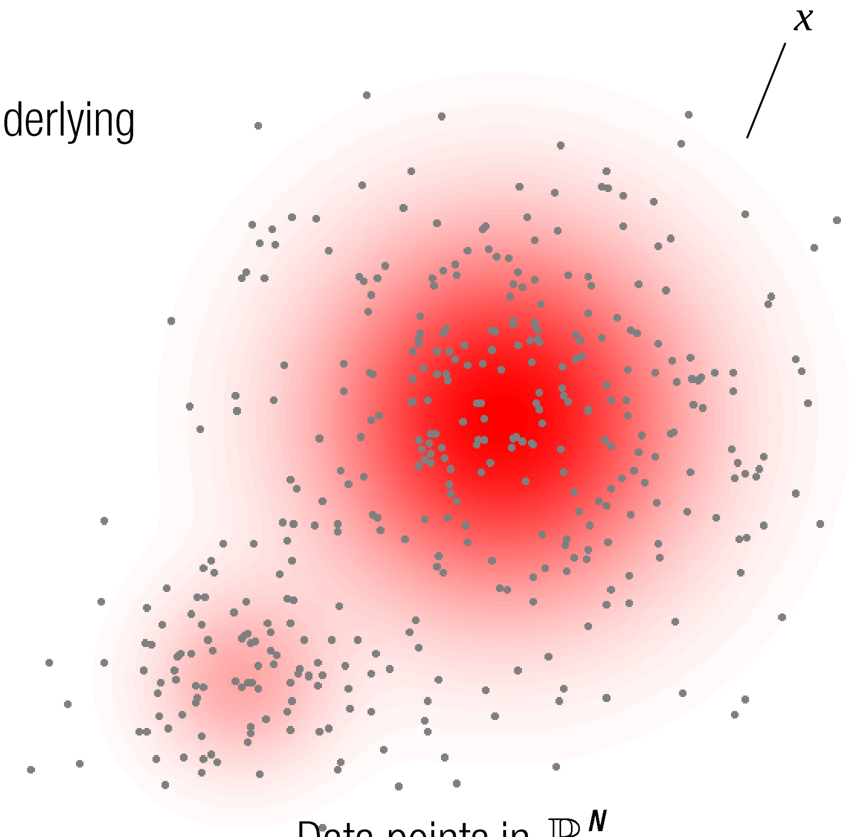


Data points in $\mathbb{R}^N$

Data Density Estimation

Assumption: data are sampled from underlying probability density function (PDF)

$$P(x) \approx P(x \mid D)$$

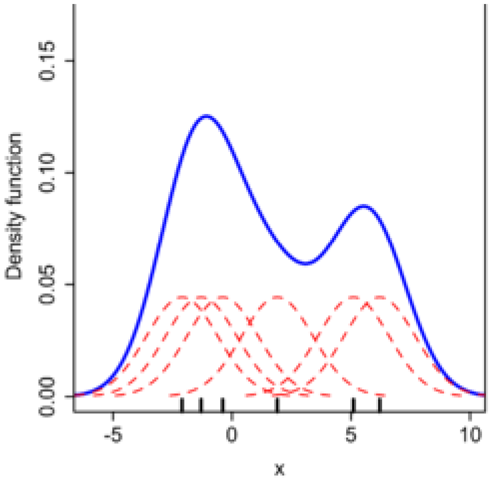


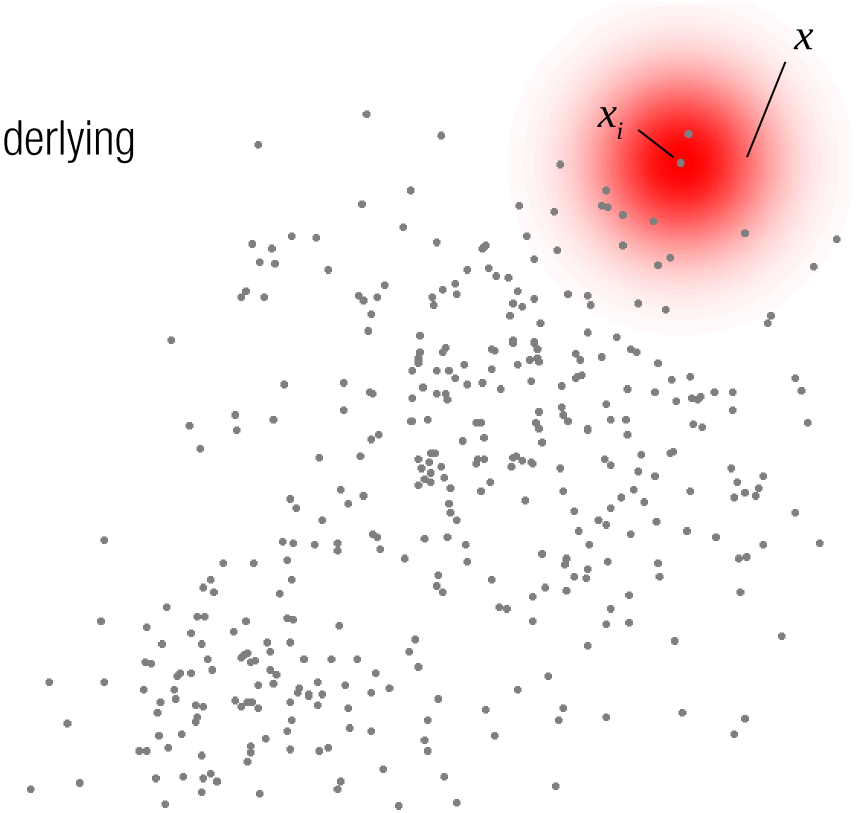
Data points in $\mathbb{R}^N$

DATA DENSITY ESTIMATION

Assumption: data are sampled from underlying probability density function (PDF)

$$P(x) \approx P(x \mid D)$$
$$\approx p(x \mid x_1) + \cdots + p(x \mid x_n)$$



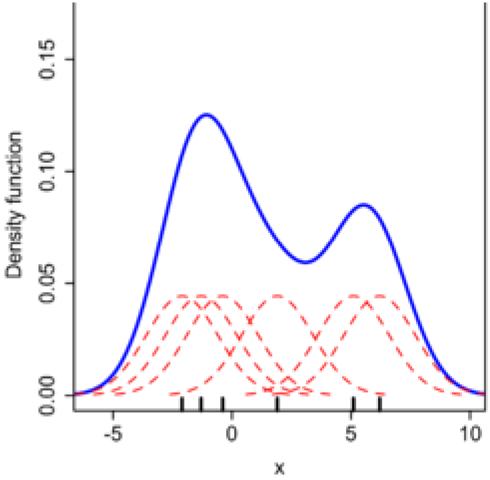


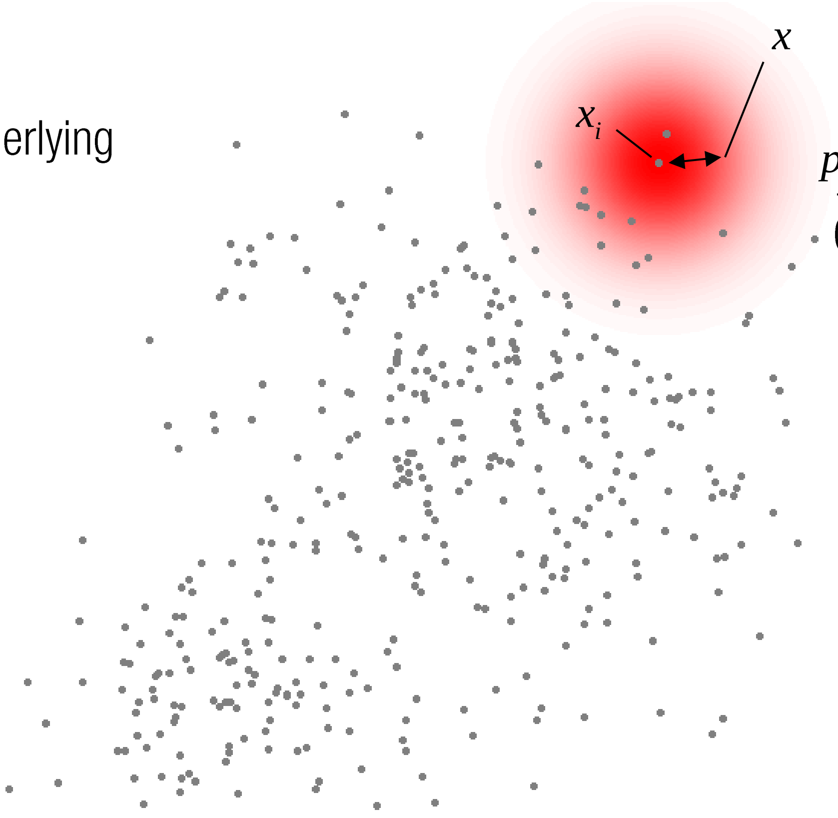
Data points in $\mathbb{R}^N$

DATA DENSITY ESTIMATION

Assumption: data are sampled from underlying probability density function (PDF)

$$P(x) \approx P(x \mid D)$$
$$\approx p(x \mid x_1) + \cdots + p(x \mid x_n)$$





$$p(x \mid x_i) = Ce^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

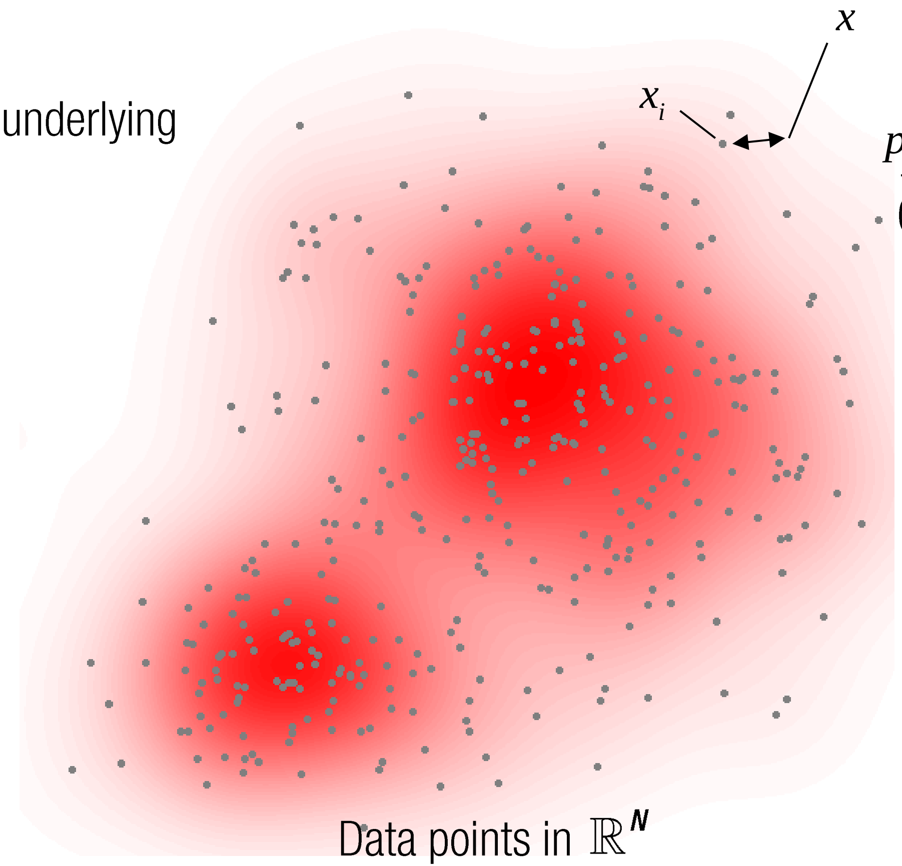
Gaussian kernel

Data points in $\mathbb{R}^N$

Data Density Estimation

Assumption: data are sampled from underlying probability density function (PDF)

$$
\begin{aligned}
P(x) &\approx P(x \mid D) \\
&\approx p(x \mid x_1) + \cdots + p(x \mid x_n) \\
&= \frac{1}{n} \sum_i c_i e^{-\frac{(x-x_i)^2}{2\sigma^2}}
\end{aligned}
$$



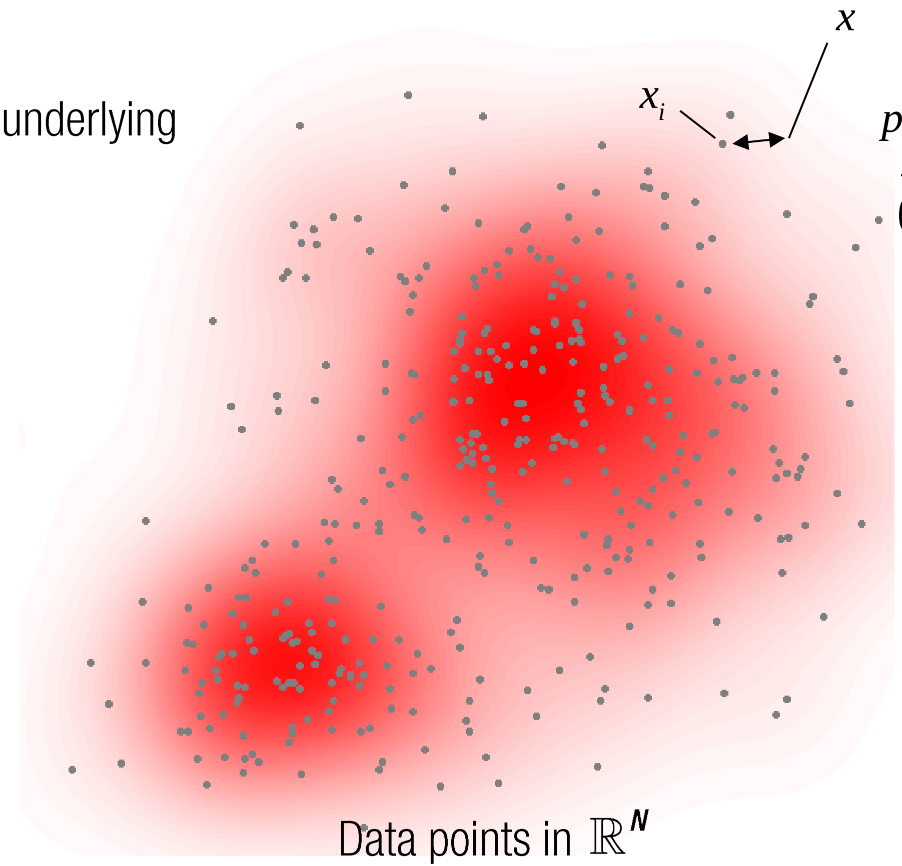
$$p(x \mid x_i) = Ce^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

Gaussian kernel

Data Density Estimation

Assumption: data are sampled from underlying probability density function (PDF)

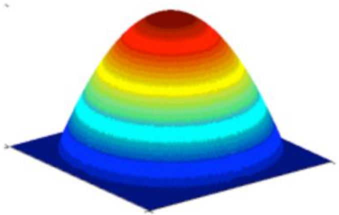
$$
\begin{aligned}
P(x) &\approx P(x \mid D) \\
&\approx p(x \mid x_1) + \cdots + p(x \mid x_n) \\
&= \frac{1}{n} \sum_i K(x - x_i)
\end{aligned}
$$



$$p(x \mid x_i) = K(x - x_i)$$

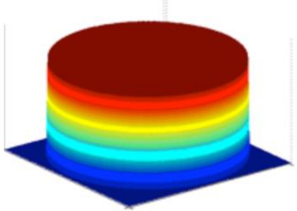
General kernel

Epanechnikov kernel



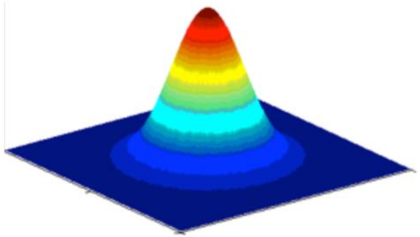
$$K(\boldsymbol{x}, \boldsymbol{x}') = \begin{cases} c(1 - \|\boldsymbol{x} - \boldsymbol{x}'\|^2) & \|\boldsymbol{x} - \boldsymbol{x}'\|^2 \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Uniform kernel



$$K(\boldsymbol{x}, \boldsymbol{x}') = \begin{cases} c & \|\boldsymbol{x} - \boldsymbol{x}'\|^2 \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Normal kernel



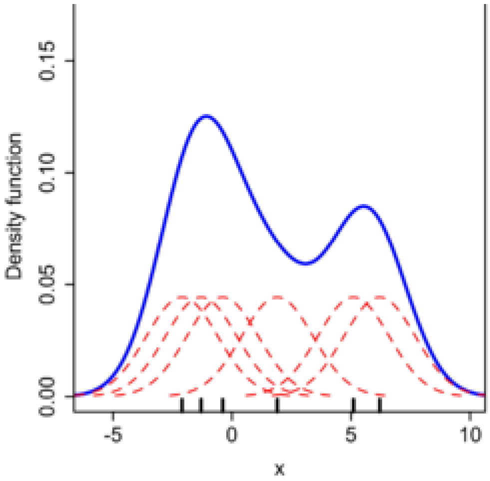
$$K(\boldsymbol{x}, \boldsymbol{x}') = c \exp\left(\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2\right)$$

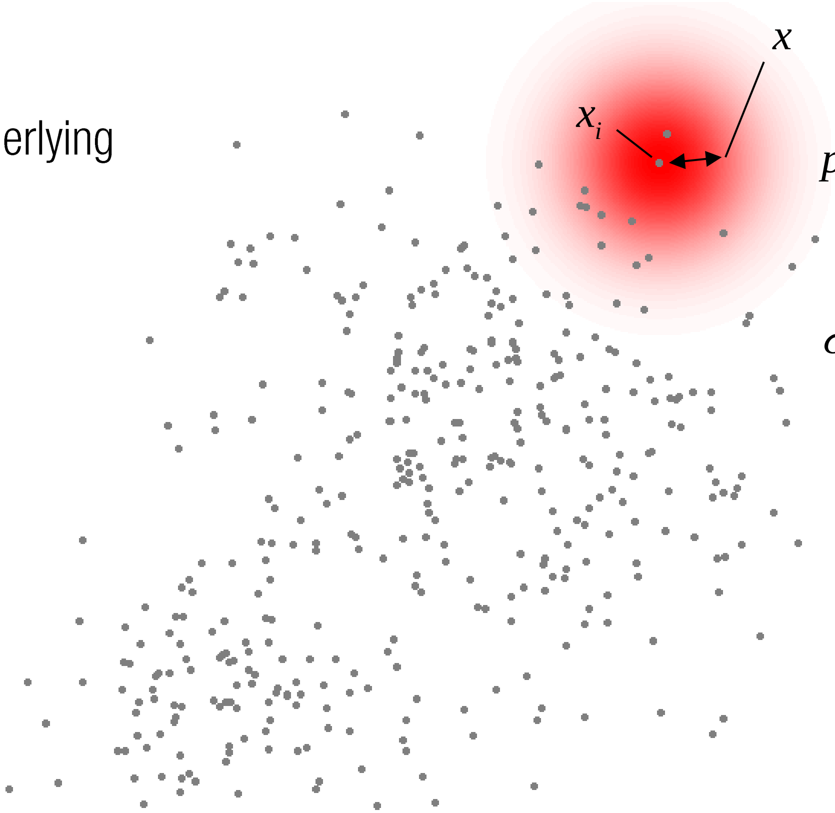
Radially symmetric kernels

DATA DENSITY ESTIMATION

Assumption: data are sampled from underlying probability density function (PDF)

$$P(x) \approx P(x \mid D)$$
$$\approx p(x \mid x_1) + \cdots + p(x \mid x_n)$$





Data points in $\mathbb{R}^N$

$$p(x \mid x_i) = Ce^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

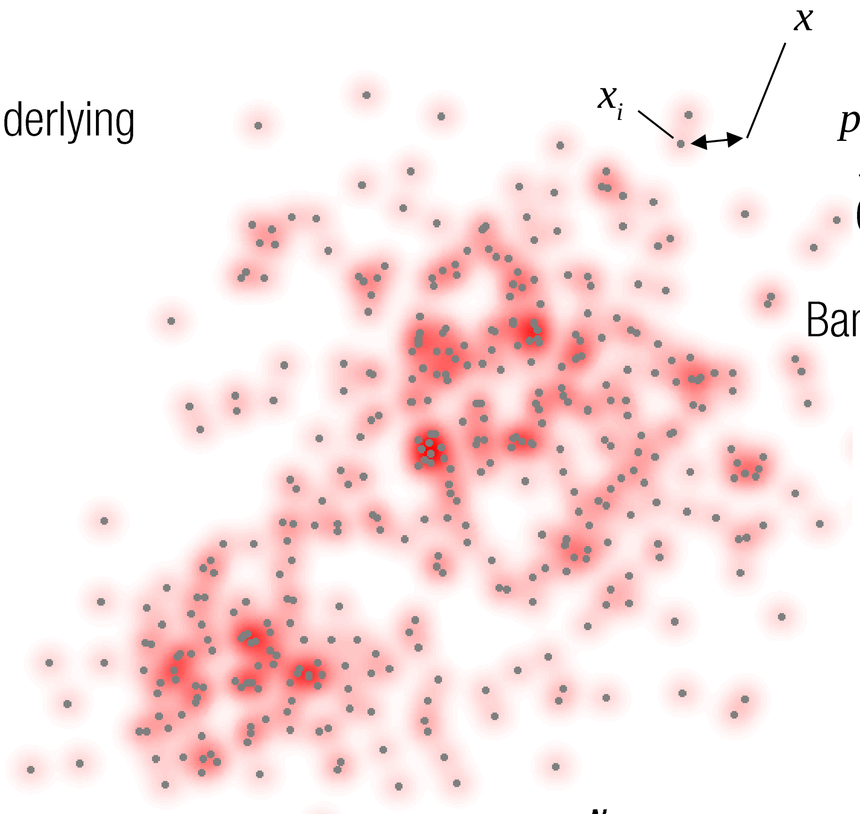
Gaussian kernel

σ: Bandwidth

DATA DENSITY ESTIMATION

Assumption: data are sampled from underlying probability density function (PDF)

$$P(x) \approx P(x \mid D)$$
$$\approx p(x \mid x_1) + \cdots + p(x \mid x_n)$$
$$= \frac{1}{n}\sum_i K(x - x_i)$$



Data points in $\mathbb{R}^N$

$$p(x \mid x_i) = K(x - x_i)$$

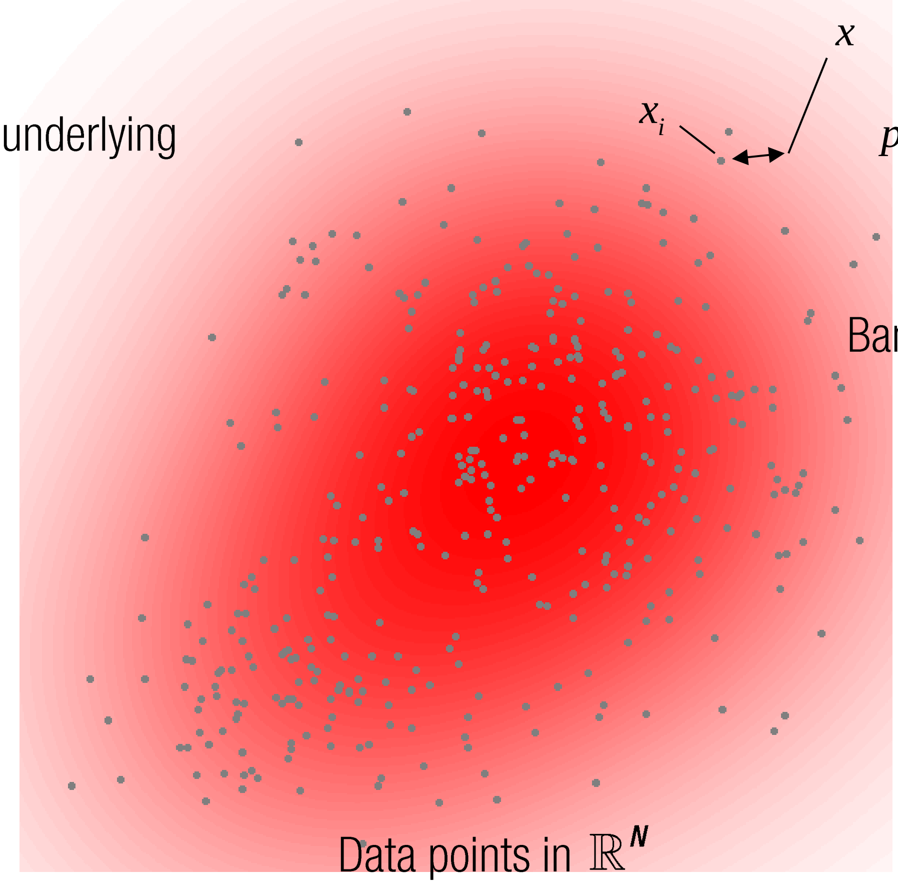
General kernel

Bandwidth is too small.

DATA DENSITY ESTIMATION

Assumption: data are sampled from underlying probability density function (PDF)

$$P(x) \approx P(x \mid D)$$
$$\approx p(x \mid x_1) + \cdots + p(x \mid x_n)$$
$$= \frac{1}{n} \sum_i K(x - x_i)$$



$$p(x \mid x_i) = K(x - x_i)$$

General kernel

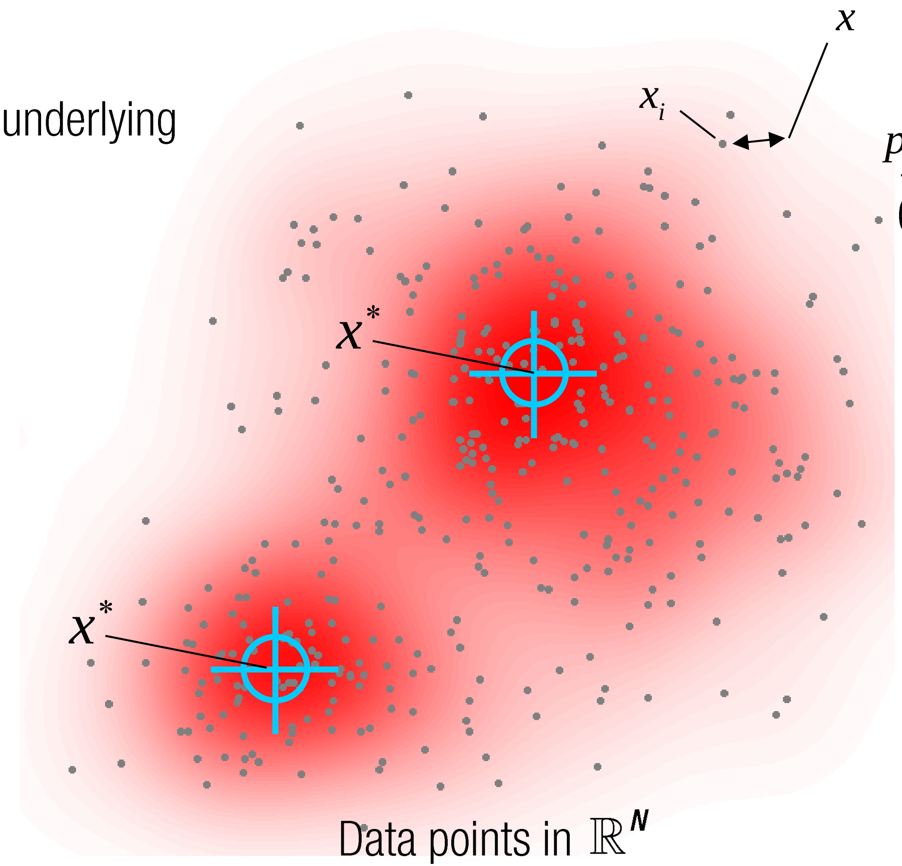
Bandwidth is too big.

Mode-seeking Algorithm

Assumption: data are sampled from underlying probability density function (PDF)

$$
\begin{aligned}
P(x) &\approx P(x \mid D) \\
&\approx p(x \mid x_1) + \cdots + p(x \mid x_n) \\
&= \frac{1}{n} C \sum_i e^{-\frac{(x-x_i)^2}{2\sigma^2}}
\end{aligned}
$$

$$x^* = \underset{x}{\operatorname{argmax}}\, P(x)$$



$$p(x \mid x_i) = C e^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

Gaussian kernel

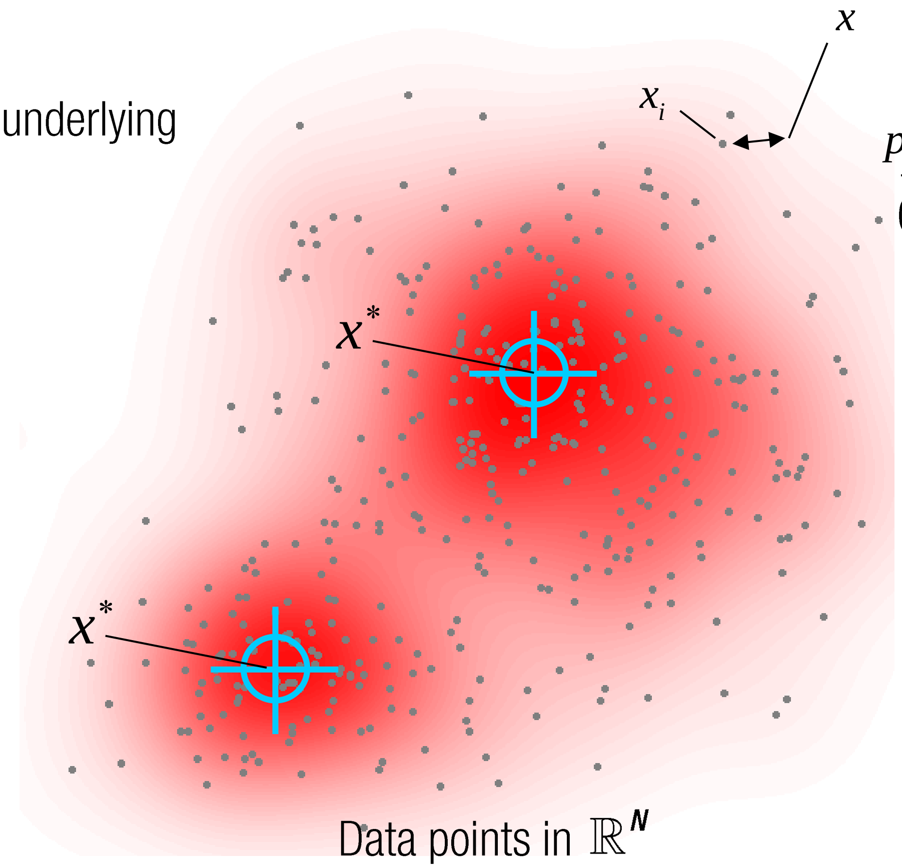
MODE-SEEKING ALGORITHM

Assumption: data are sampled from underlying probability density function (PDF)

$$P(x) \approx P(x \mid D)$$
$$\approx p(x \mid x_1) + \cdots + p(x \mid x_n)$$
$$= \frac{1}{n} C \sum_i e^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

$$x^* = \underset{x}{\operatorname{argmax}}\, P(x)$$

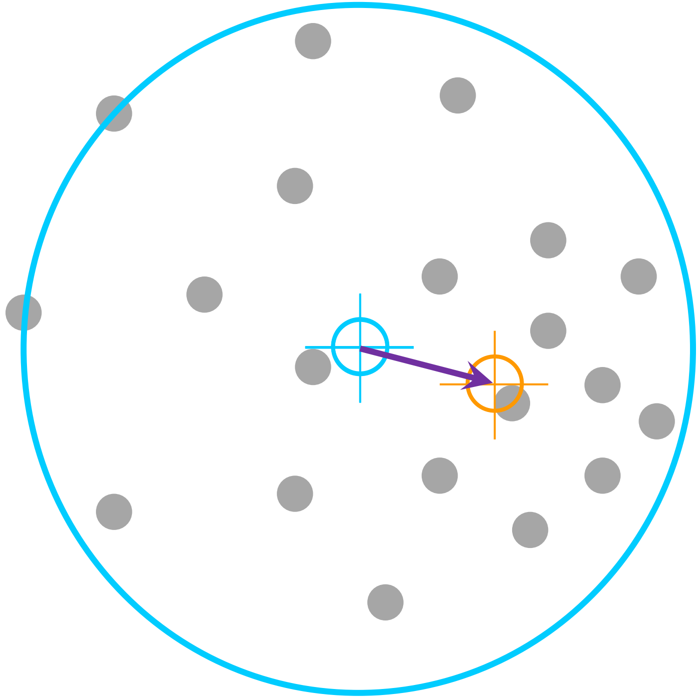
$$\nabla P(x) = 0$$



$$p(x \mid x_i) = Ce^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

Gaussian kernel

MODE-SEEKING ALGORITHM



$$x^* = \underset{x}{\operatorname{argmax}}\, P(x)$$

$$\nabla P(x) = 0$$

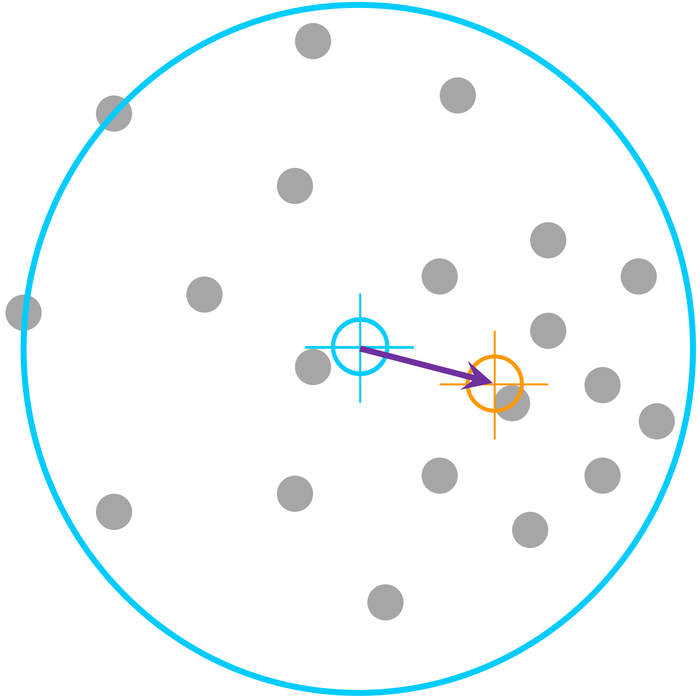
$$\nabla P(x) = \sum_i \nabla K(x - x_i)$$

$$= \sum_i \nabla \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right) = \frac{1}{\sigma^2} \sum_i (x_i - x) \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right)$$

$$= \frac{1}{\sigma^2} \sum_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right) \left(\frac{\sum_i x_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right)}{\sum_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right)} - x \right)$$

Weighted mean Shift

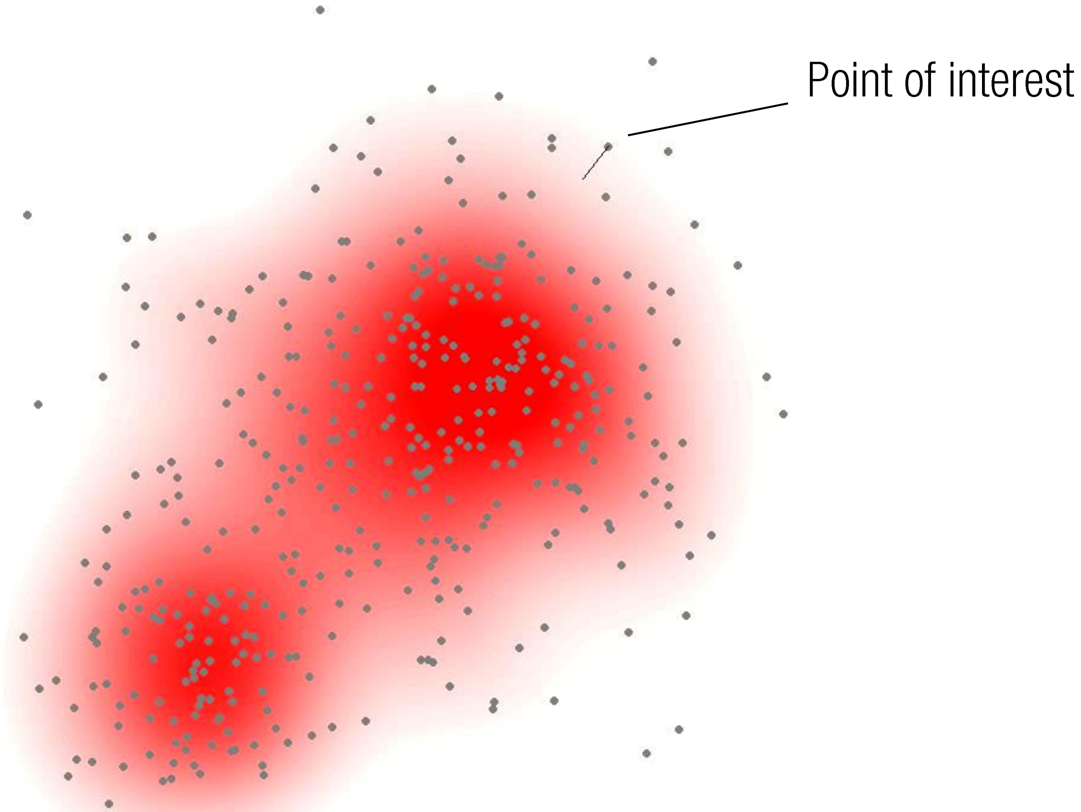
Mode-seeking Algorithm

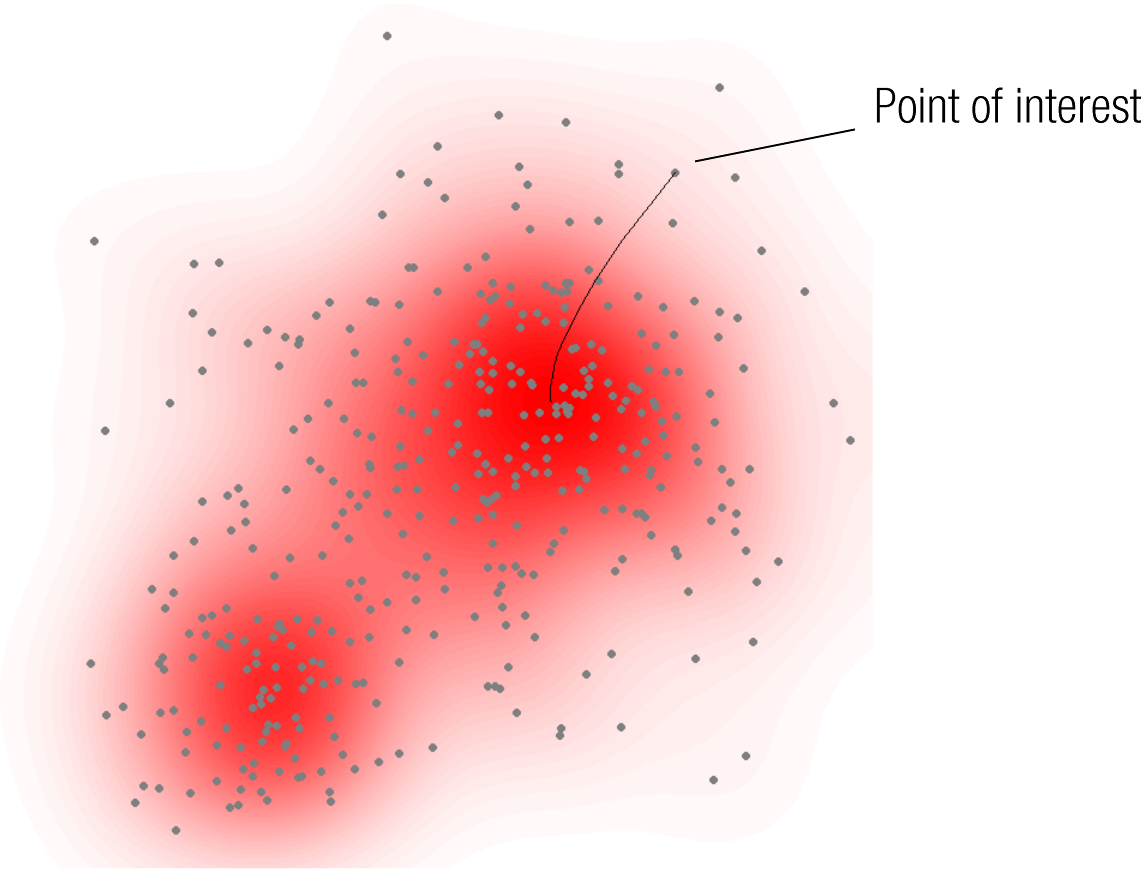


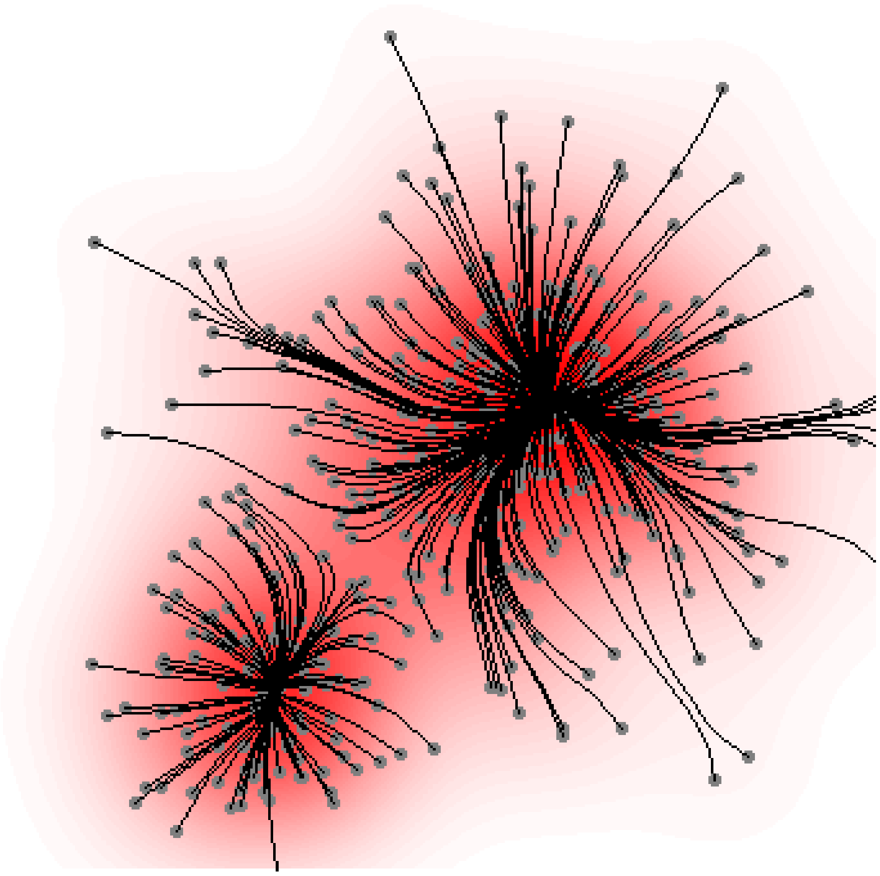
$$x^* = \operatorname*{argmax}_x P(x)$$

$$\nabla P(x) = 0$$

$$x_{new} = \frac{\sum_i x_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right)}{\sum_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right)}$$


Point of interest


Point of interest

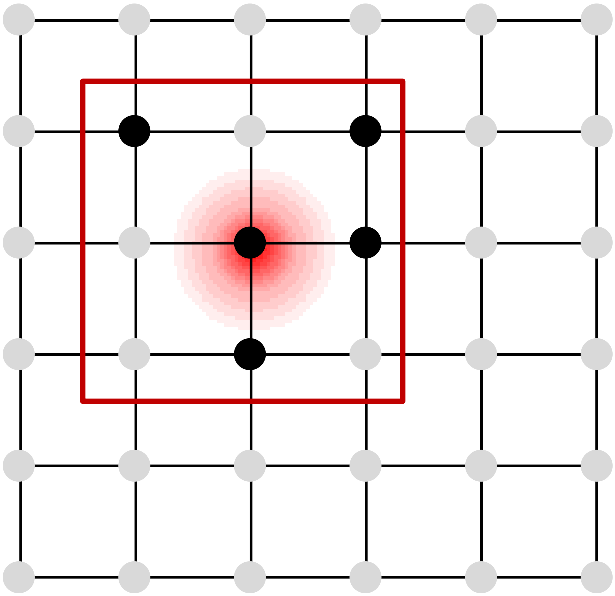


Properties of mean shift

- Determines the step length of mean shift (adaptive gradient ascent).
- Guarantees convergence if the gradient of the kernel is a convex function.
- Requires no parameter except for the bandwidth.
- Detect multiple modes without knowing the number of modes

Comparison

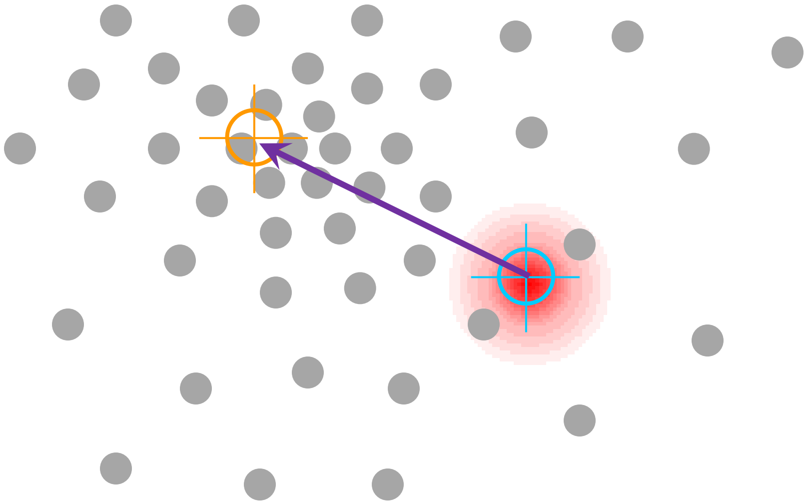
Grid with weight



$$p_m(y) = C\sum_i e^{-\frac{(y-x_i)^2}{2\sigma^2}}\,\delta\left(b(x_i)-m\right)$$

Regular grid with weight

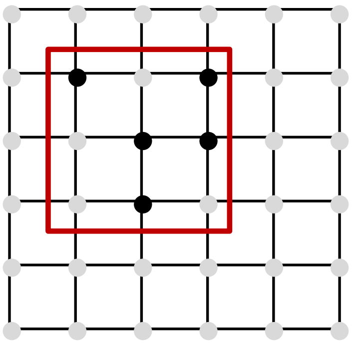
Data sample



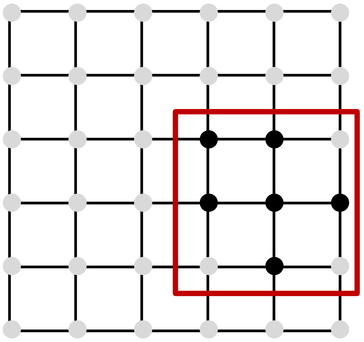
$$p(y) = C\sum_i e^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

HISTOGRAM MATCH

$t = t_0$



$t = t_1$



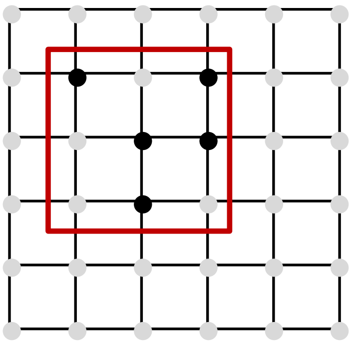
Maximize Bhattacharyya coefficient:

$$y^* = \arg\max_y \rho(y)$$

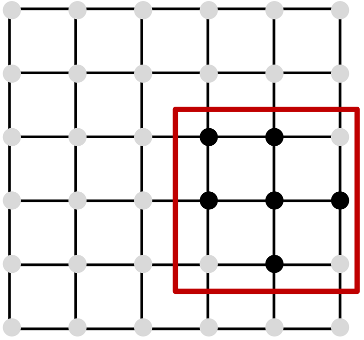
$$= \arg\max_y \sum_m \sqrt{p_m(y) q_m}$$

Histogram Match

$t = t_0$



$t = t_1$



Maximize Bhattacharyya coefficient:

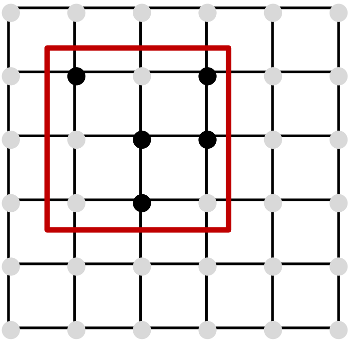
$$y^* = \underset{y}{\operatorname{argmax}}\, \rho(y)$$

$$= \underset{y}{\operatorname{argmax}} \sum_m \sqrt{p_m(y) q_m}$$

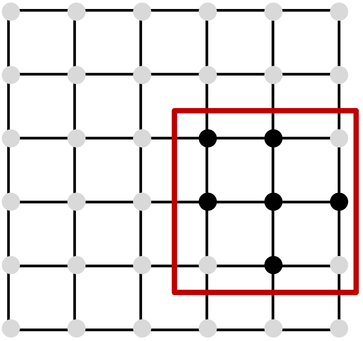
$$\nabla \rho(y) = \frac{1}{2} \sum_m \nabla p_m(y) \sqrt{\frac{q_m}{p_m(y)}}$$

Histogram Match

$t = t_0$



$t = t_1$



Maximize Bhattacharyya coefficient:

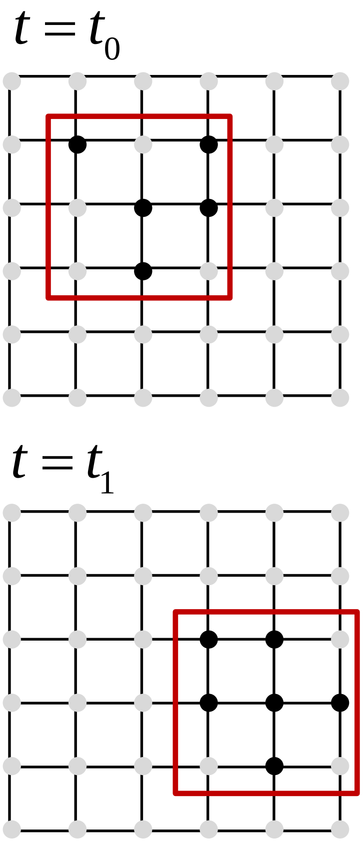
$$y^* = \underset{y}{\operatorname{argmax}}\, \rho(y)$$

$$= \underset{y}{\operatorname{argmax}} \sum_m \sqrt{p_m(y) q_m}$$

$$\nabla \rho(y) = \frac{1}{2} \sum_m \nabla p_m(y) \sqrt{\frac{q_m}{p_m(y)}}$$

$$= \frac{C}{2} \sum_m \left(\sqrt{\frac{q_m}{p_m(y)}} \sum_i (x_i - y) e^{-\frac{(y-x_i)^2}{2\sigma^2}} \delta\left(b(x_i) - m\right) \right)$$

Histogram Match



Maximize Bhattacharyya coefficient:

$$y^* = \arg\max_y \rho(y)$$

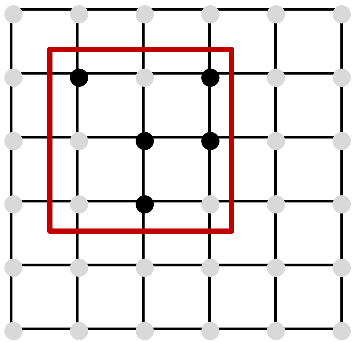
$$= \arg\max_y \sum_m \sqrt{p_m(y) q_m}$$

$$\nabla \rho(y) = \frac{1}{2} \sum_m \nabla p_m(y) \sqrt{\frac{q_m}{p_m(y)}}$$

$$= \frac{C}{2} \sum_m \left(\sqrt{\frac{q_m}{p_m(y)}} \sum_i (x_i - y) e^{-\frac{(y-x_i)^2}{2\sigma^2}} \delta\left(b(x_i) - m\right) \right)$$

$$= \frac{C}{2} \sum_i (x_i - y) w_i e^{-\frac{(y-x_i)^2}{2\sigma^2}} \quad \text{where } w_i = \sum_m \sqrt{\frac{q_m}{p_m(y)}} \delta\left(b(x_i) - m\right)$$

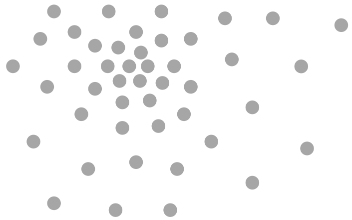
Comparison



$$y^* = \underset{y}{\operatorname{argmax}}\, \rho(y)$$

$$\nabla\rho(y) = \frac{C}{2}\sum_i (x_i - y) w_i e^{-\frac{(y-x_i)^2}{2\sigma^2}}$$

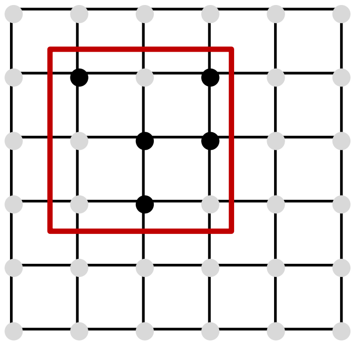
where $w_i = \sum_m \sqrt{\frac{q_m}{p_m(y)}}\,\delta\big(b(x_i) - m\big)$

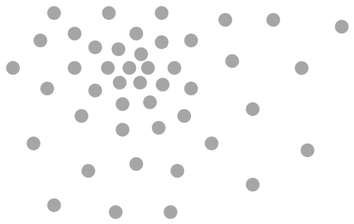


$$y^* = \underset{y}{\operatorname{argmax}}\, P(y)$$

$$\nabla P(y) = \frac{1}{\sigma^2}\sum_i (x_i - y)\left(e^{-\frac{(y-x_i)^2}{2\sigma^2}} \right)$$

COMPARISON





$$y^* = \underset{y}{\operatorname{argmax}}\, \rho(y)$$

$$\nabla \rho(y) = \frac{C}{2} \sum_i (x_i - y) w_i e^{-\frac{(y-x_i)^2}{2\sigma^2}}$$

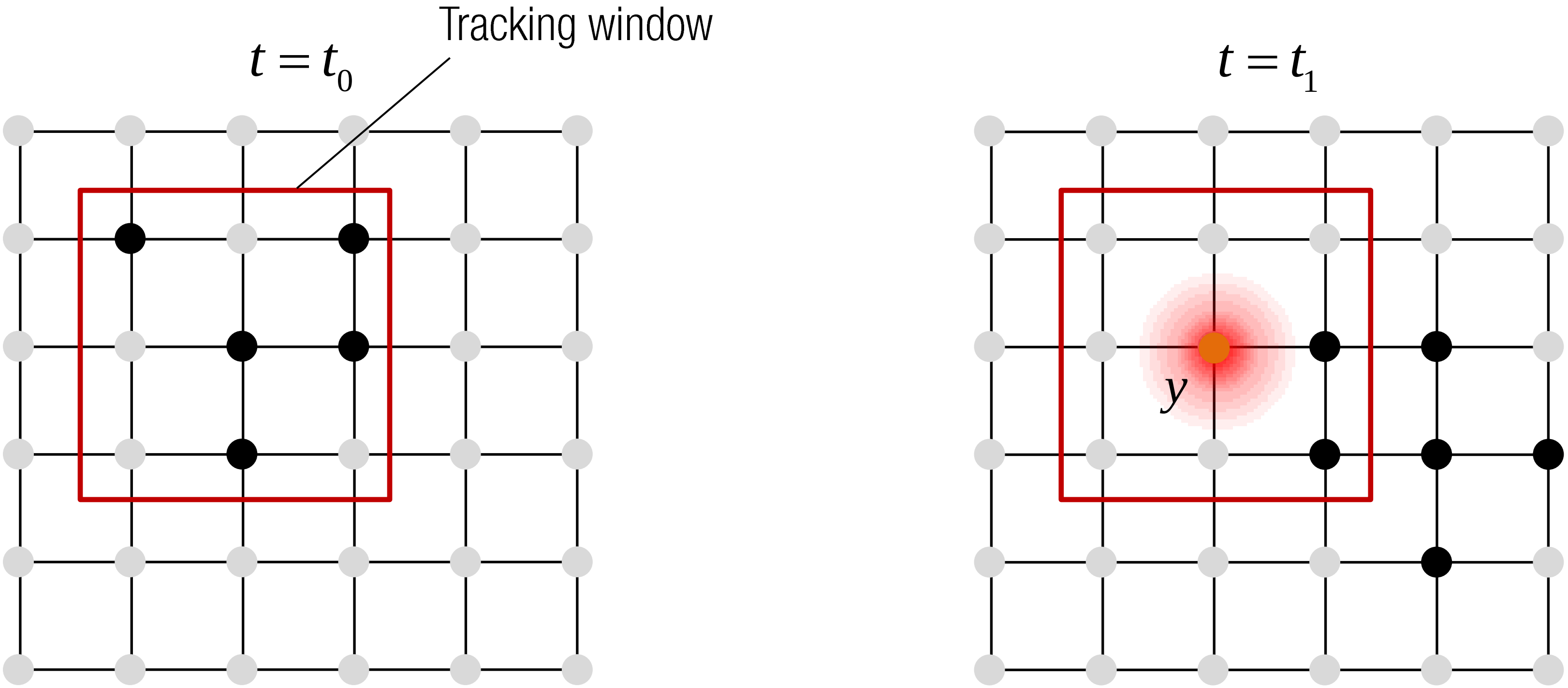
$$y_{new} = \frac{\sum_i x_i w_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right)}{\sum_i w_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right)}$$

$$y^* = \underset{y}{\operatorname{argmax}}\, P(y)$$

$$\nabla P(y) = \frac{1}{\sigma^2} \sum_i (x_i - y) \left(e^{-\frac{(y-x_i)^2}{2\sigma^2}} \right)$$

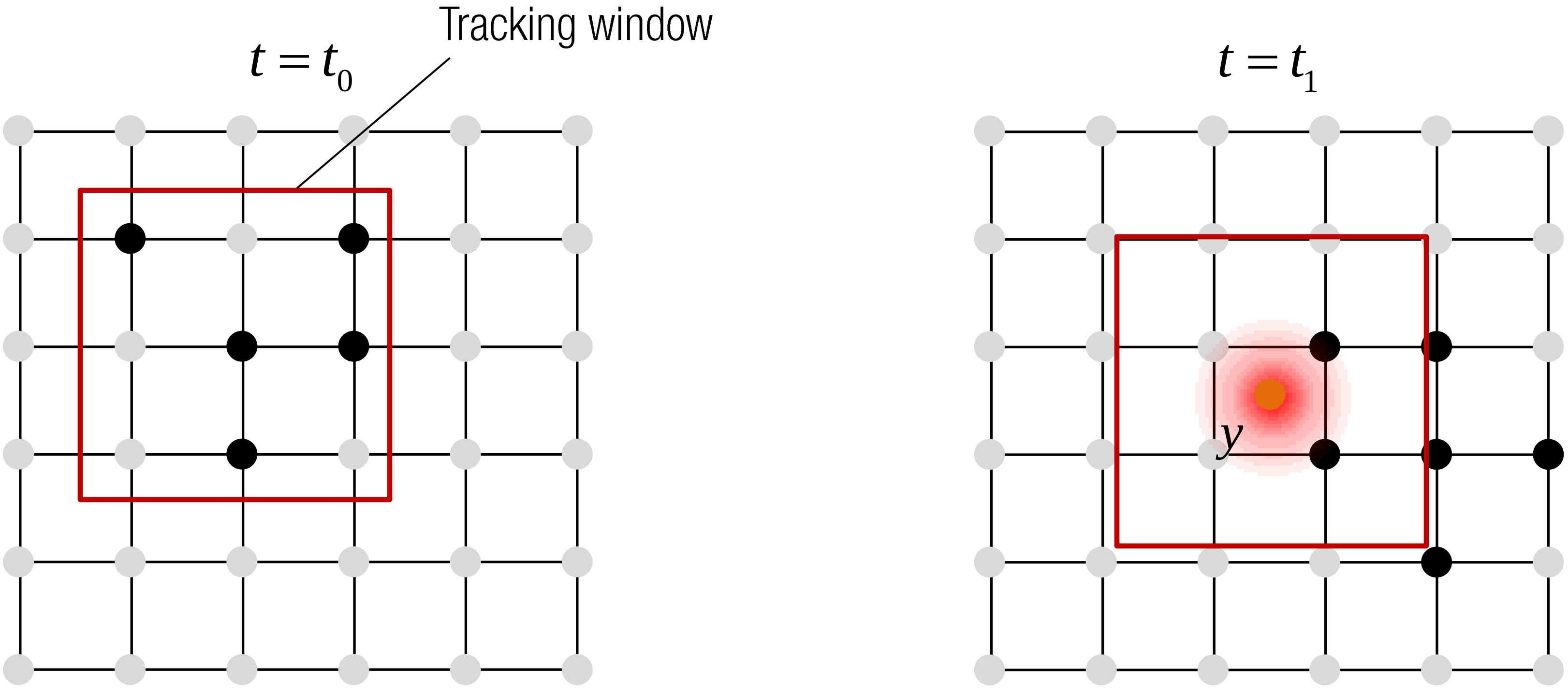
$$y_{new} = \frac{\sum_i x_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right)}{\sum_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}} \right)}$$

NONRIGID TRACKING FOR BINARY IMAGE



$$q = \begin{bmatrix} q_0 & q_1 \end{bmatrix} \qquad p(y) = \begin{bmatrix} p_0(y) & p_1(y) \end{bmatrix}$$

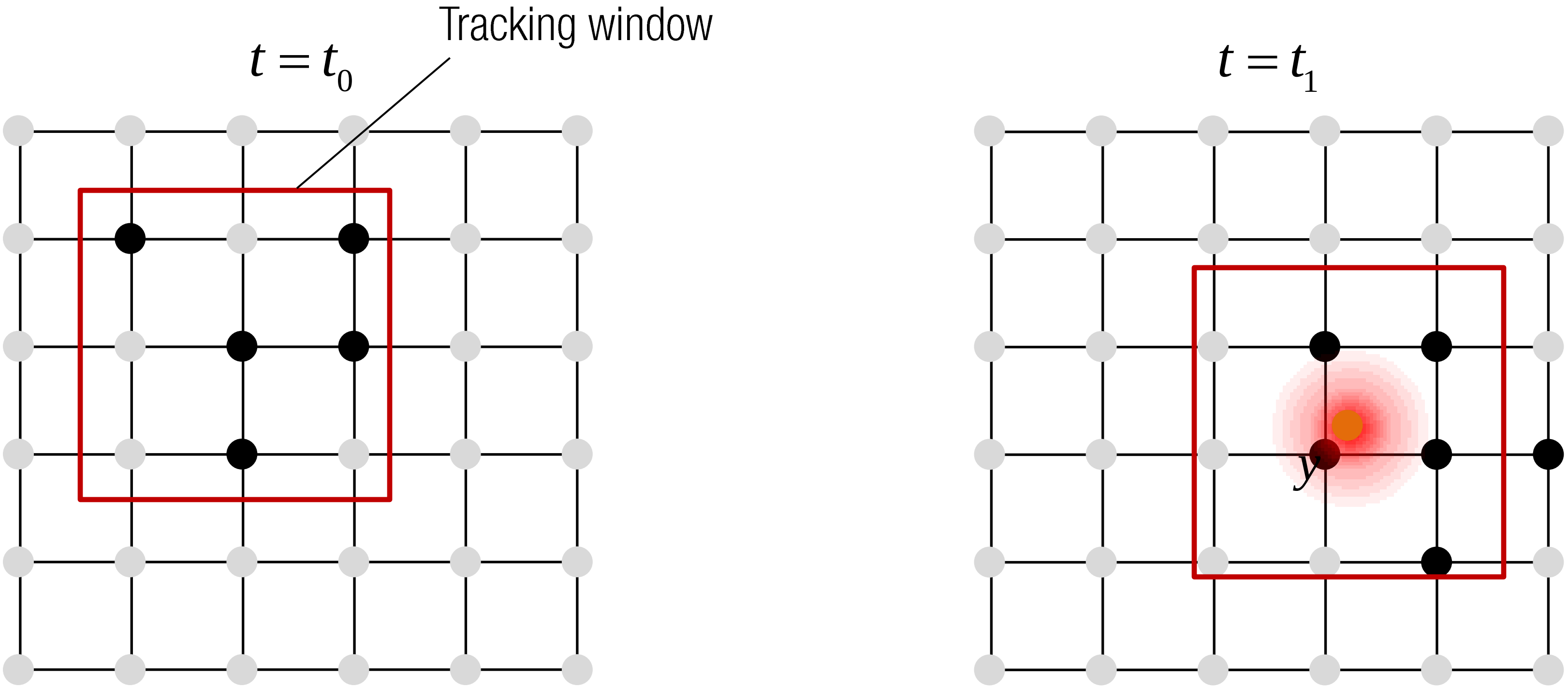
Nonrigid Tracking For Binary Image



$q = \begin{bmatrix} q_0 & q_1 \end{bmatrix}$

$p(y) = \begin{bmatrix} p_0(y) & p_1(y) \end{bmatrix}$

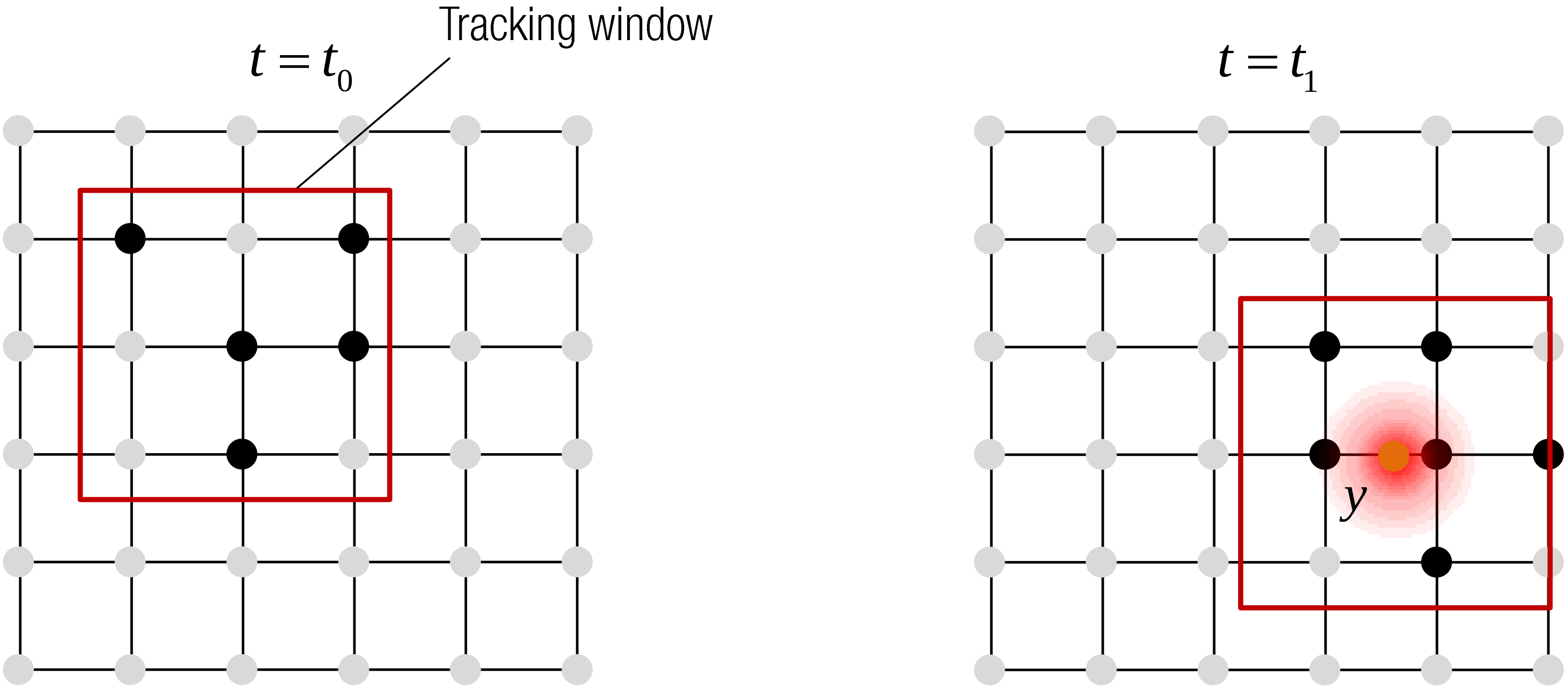
Nonrigid Tracking For Binary Image



$$q = \begin{bmatrix} q_0 & q_1 \end{bmatrix}$$

$$p(y) = \begin{bmatrix} p_0(y) & p_1(y) \end{bmatrix}$$

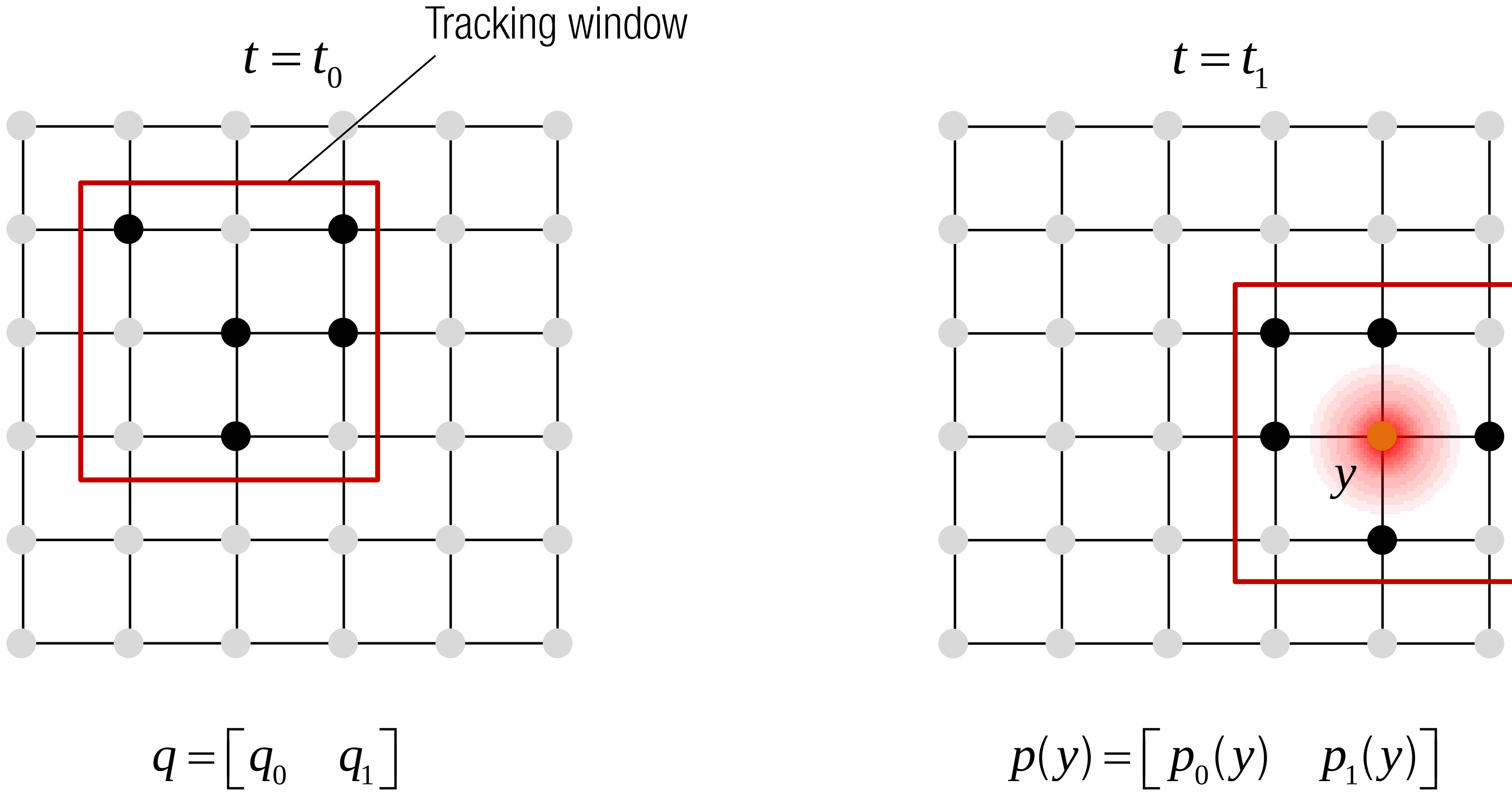
Nonrigid Tracking For Binary Image



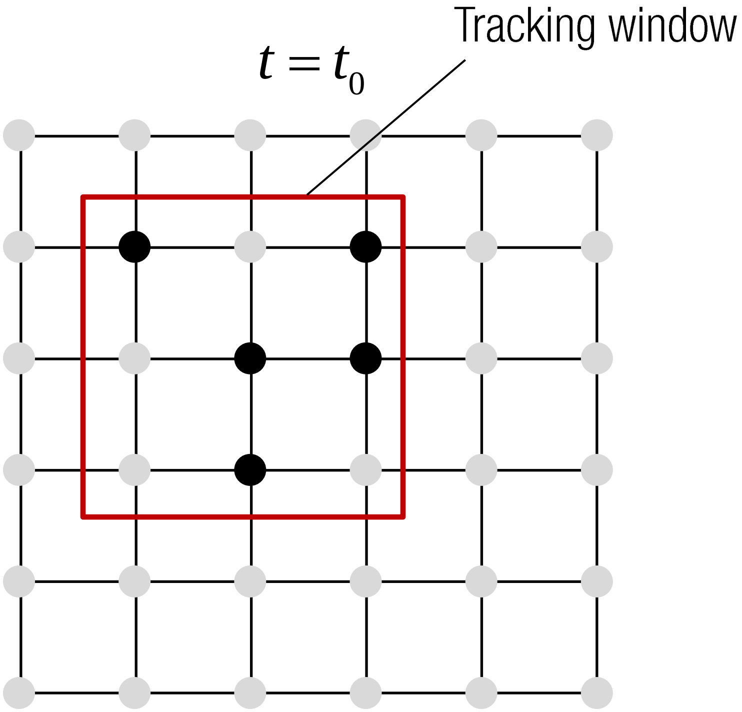
$$q = \begin{bmatrix} q_0 & q_1 \end{bmatrix}$$

$$p(y) = \begin{bmatrix} p_0(y) & p_1(y) \end{bmatrix}$$

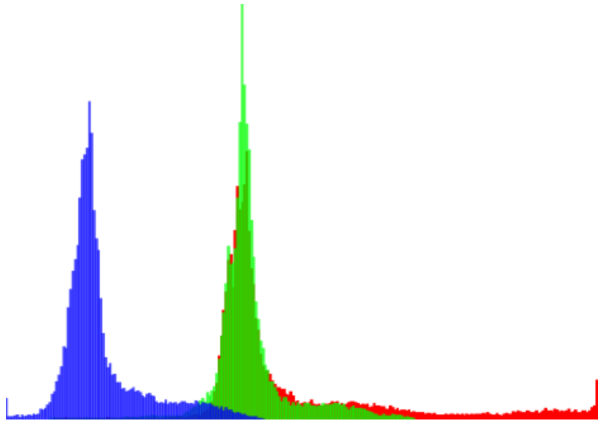
Nonrigid Tracking For Binary Image



Nonrigid Tracking For Color Image



$$q = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}$$



Discretized color histogram

Nonrigid Tracking For Color Image