Nonparametric Tracking

Hyun Soo Park


Nationale-Nederlanden
jackpot €5 milj.
ING

Nonrigid Tracking

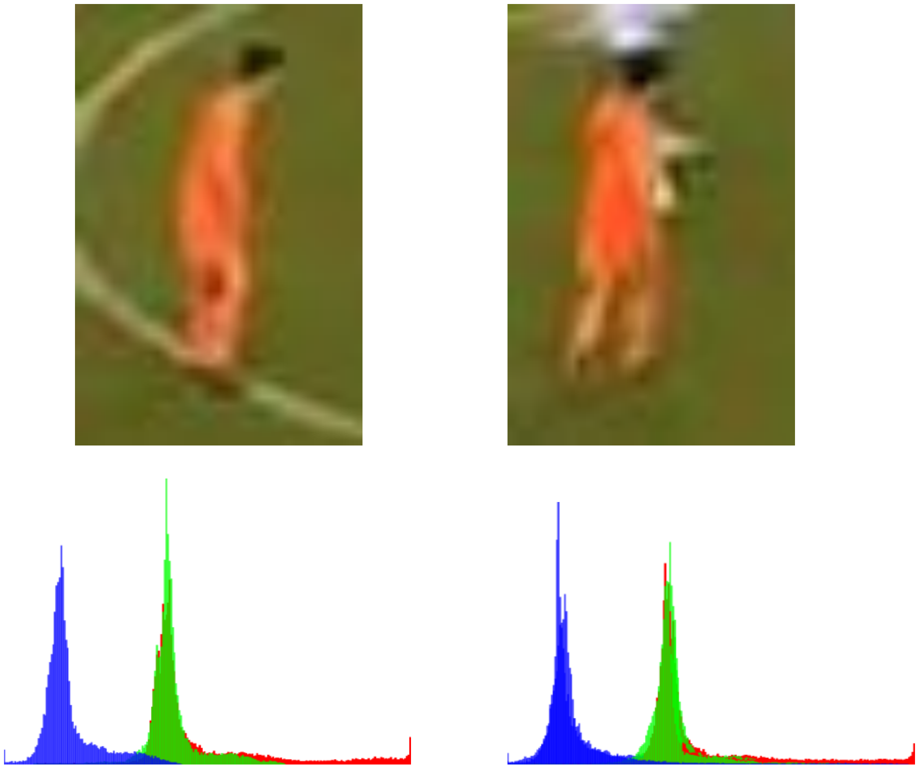
Desired algorithm:
- Invariant to nonrigid transformation


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

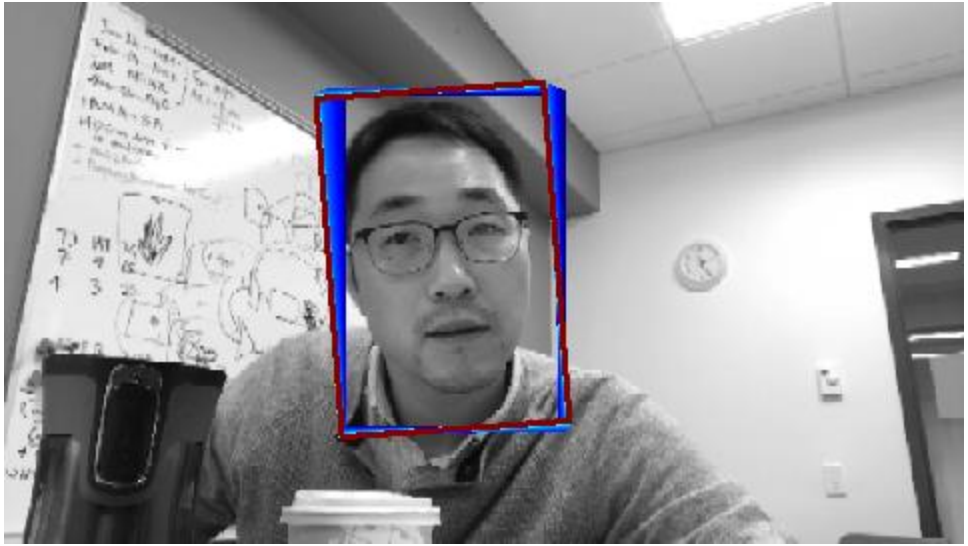
- Computationally efficient

Sliding window reqiures too much computation.

NONRIGID TRACKING

Desired algorithm:

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



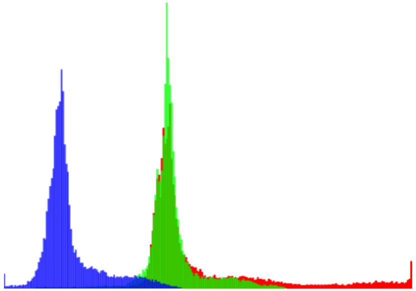
NONRIGID TRACKING

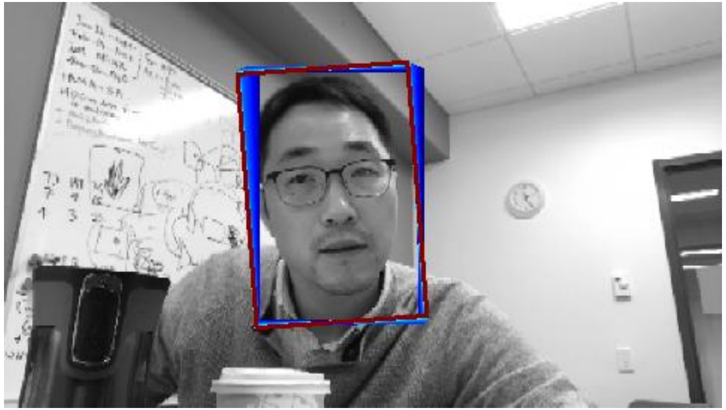
Desired algorithm:

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

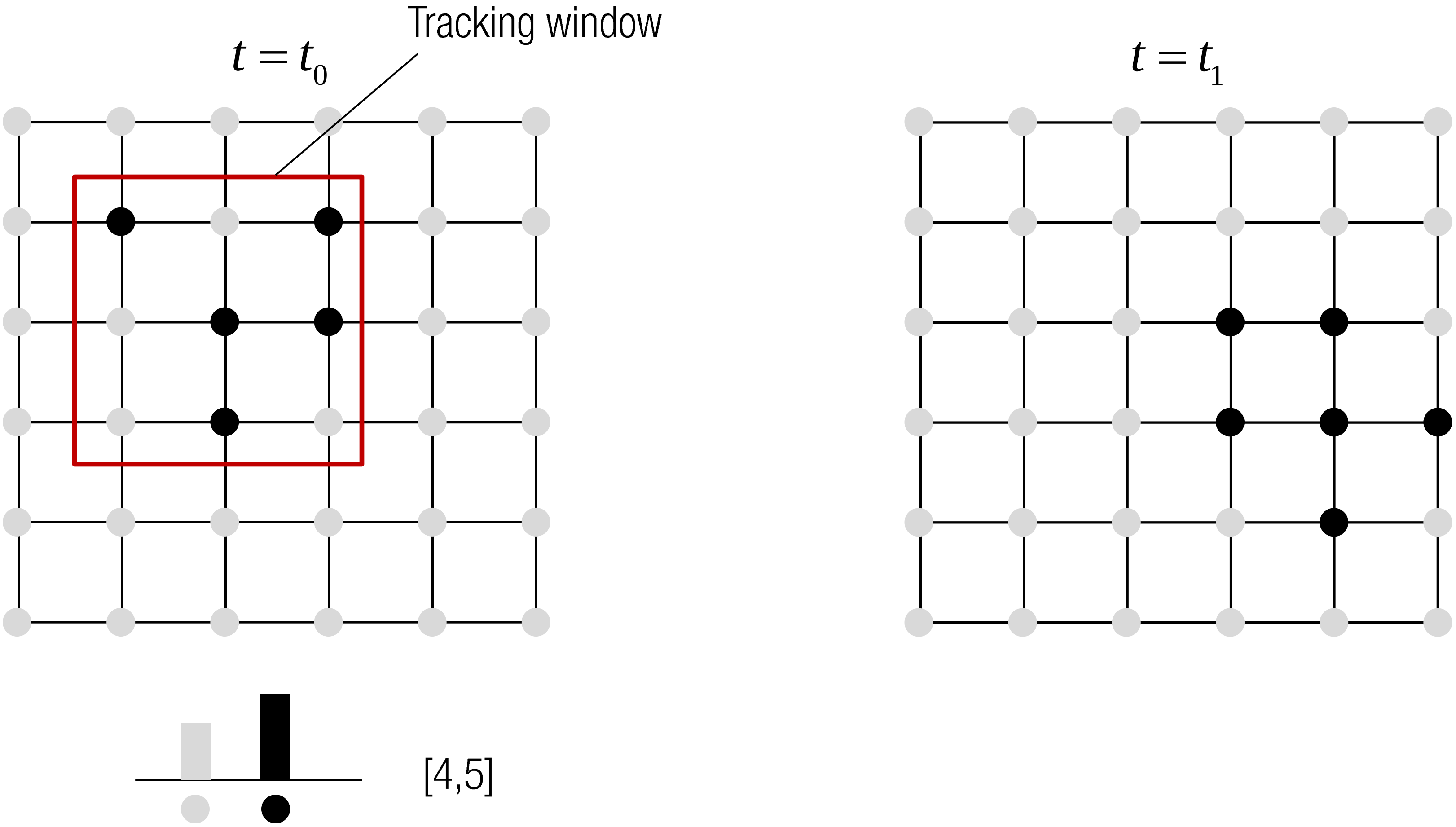
- Computationally efficient
  - Gradient based tracking (relying on spatial rep.)

Contradictory!

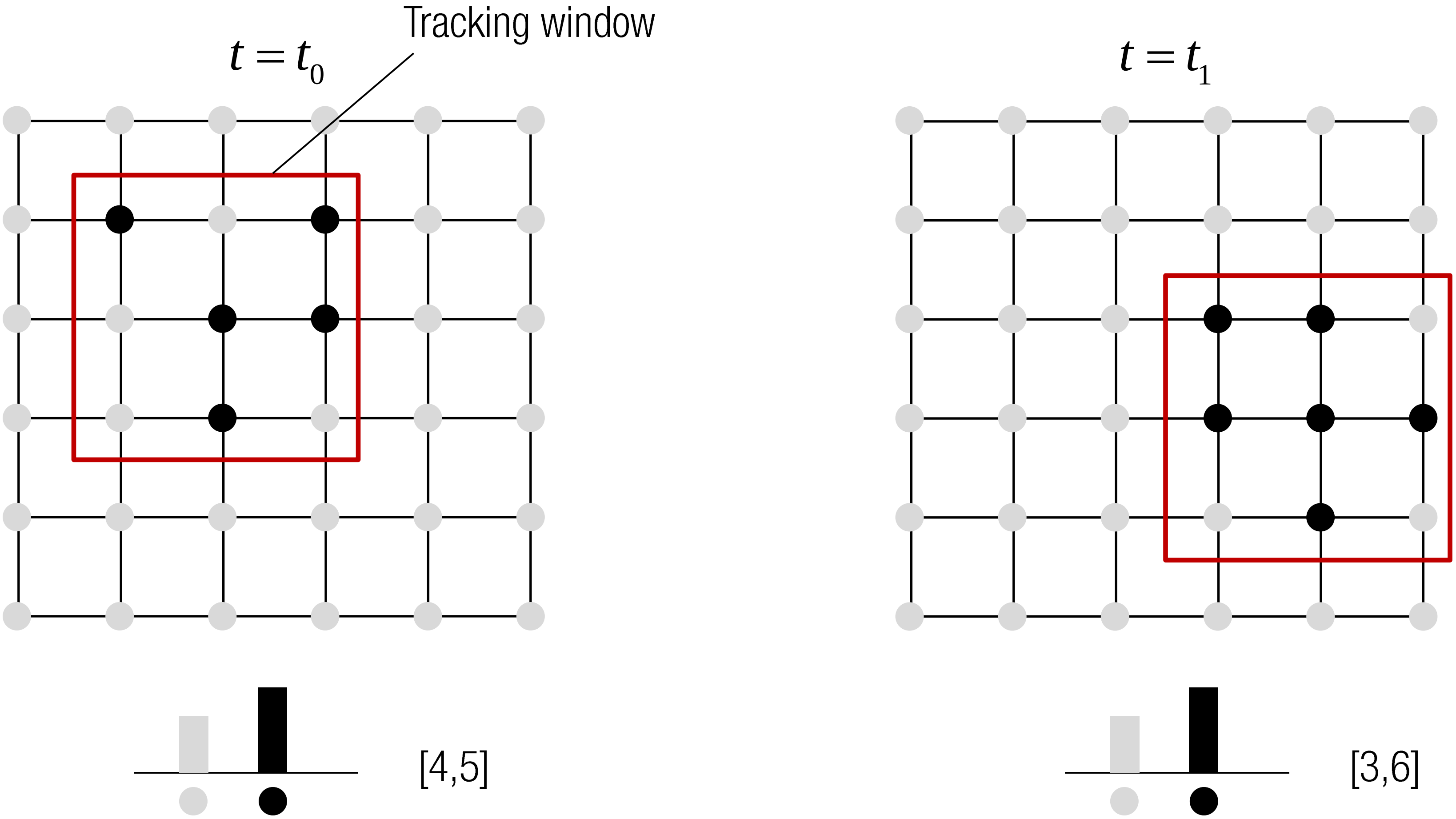




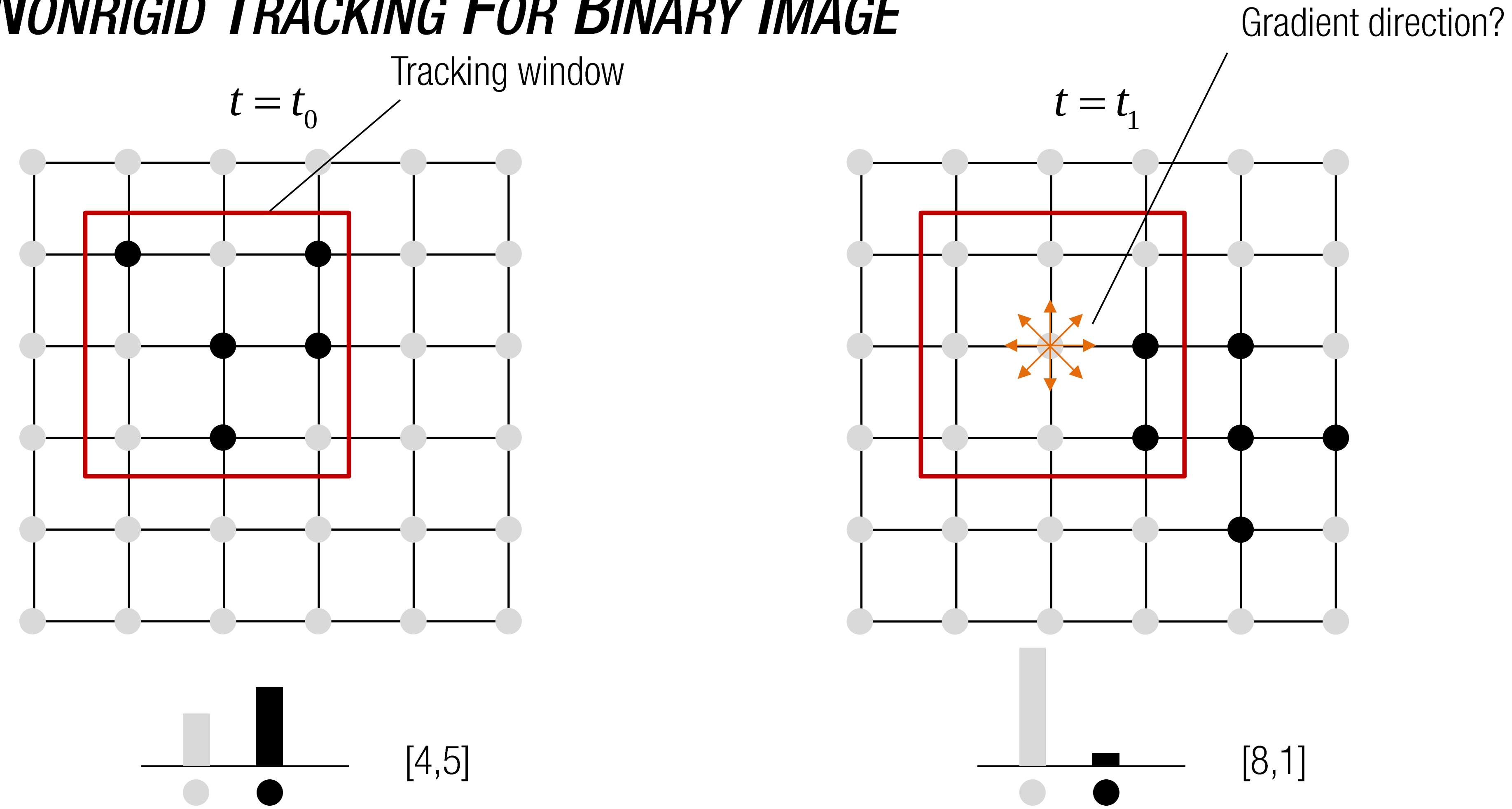
Nonrigid Tracking For Binary Image



Nonrigid Tracking For Binary Image

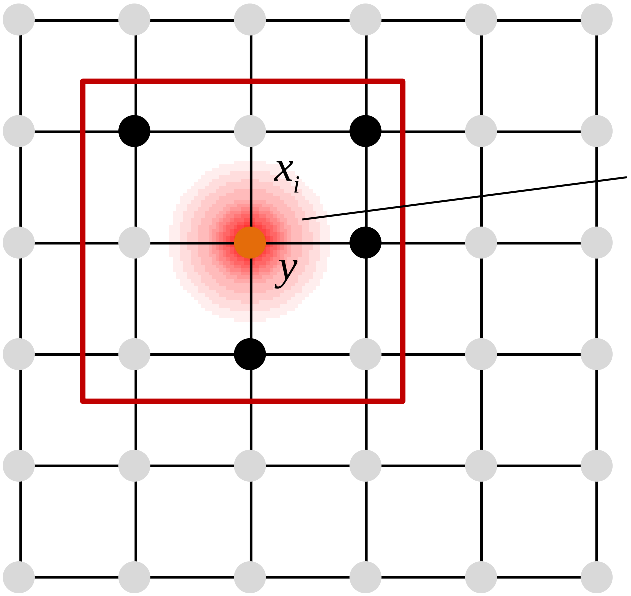


Nonrigid Tracking For Binary Image



Continuous Representation: Weighted Histogram

$t = t_0$

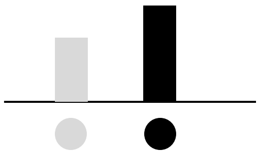


Gaussian weight

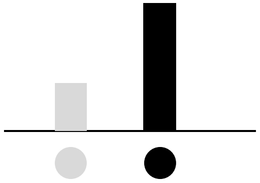
Kronecker Delta

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

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



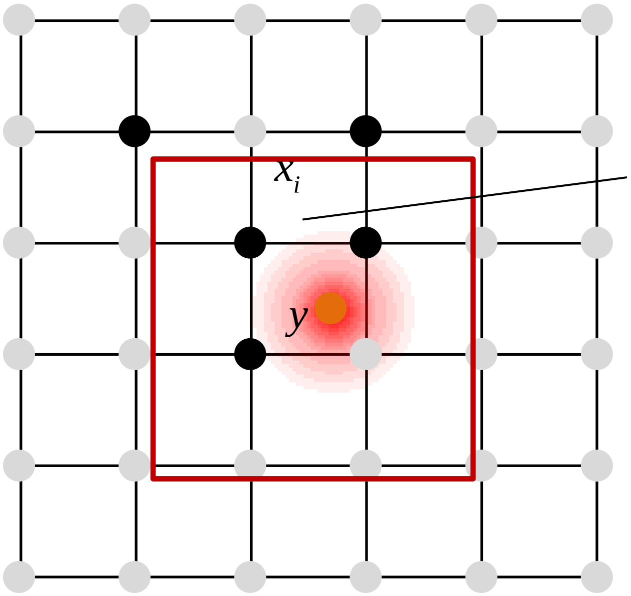
[4,5]



[3.2,5.6]

CONTINUOUS REPRESENTATION: WEIGHTED HISTOGRAM

$t = t_0$

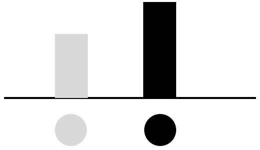


Gaussian weight

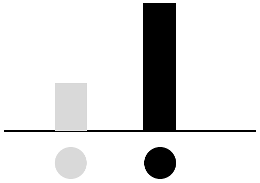
Kronecker Delta

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

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

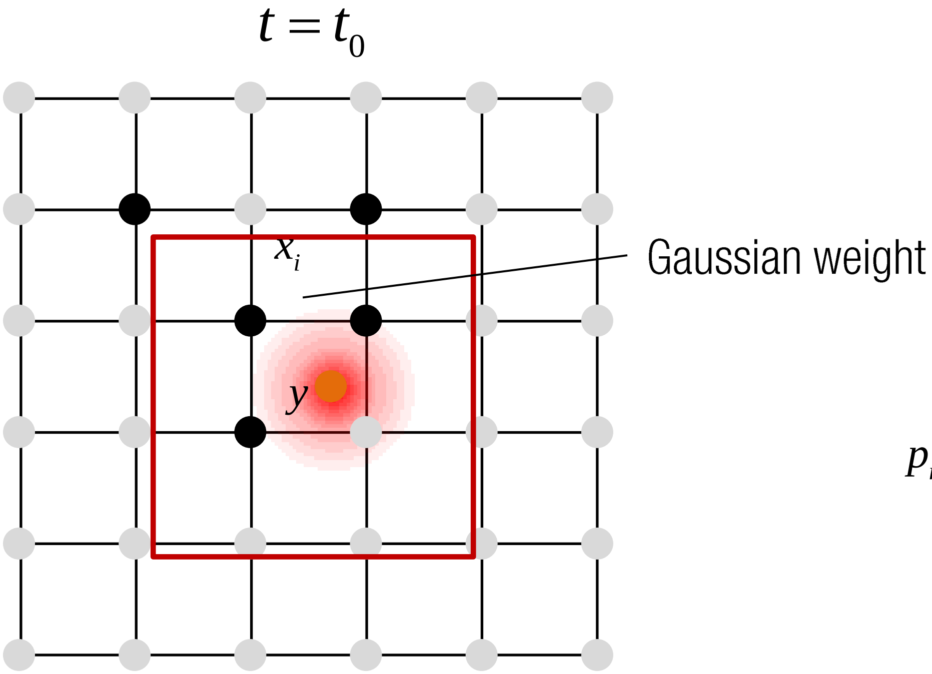


[4,5]

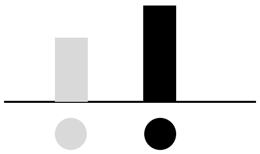


[3.2,5.6]

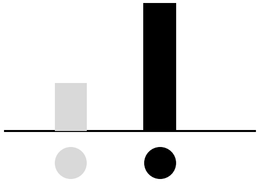
Continuous Representation: Weighted Histogram



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

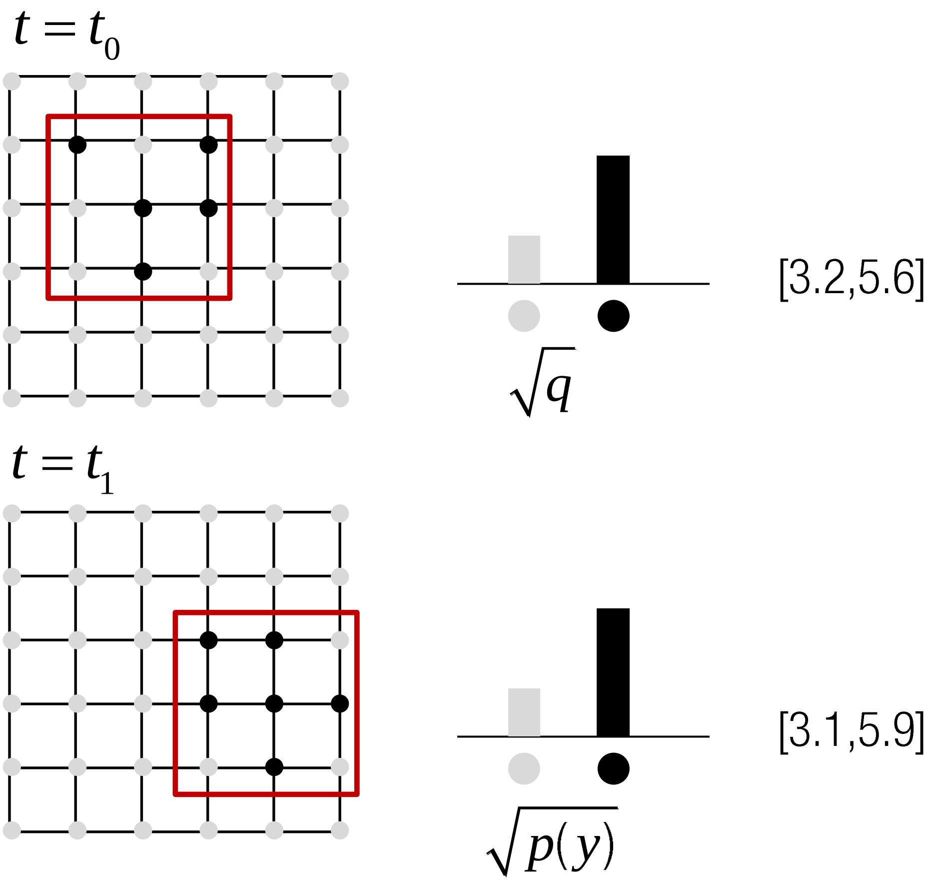


[4,5]



[3.2,5.6]

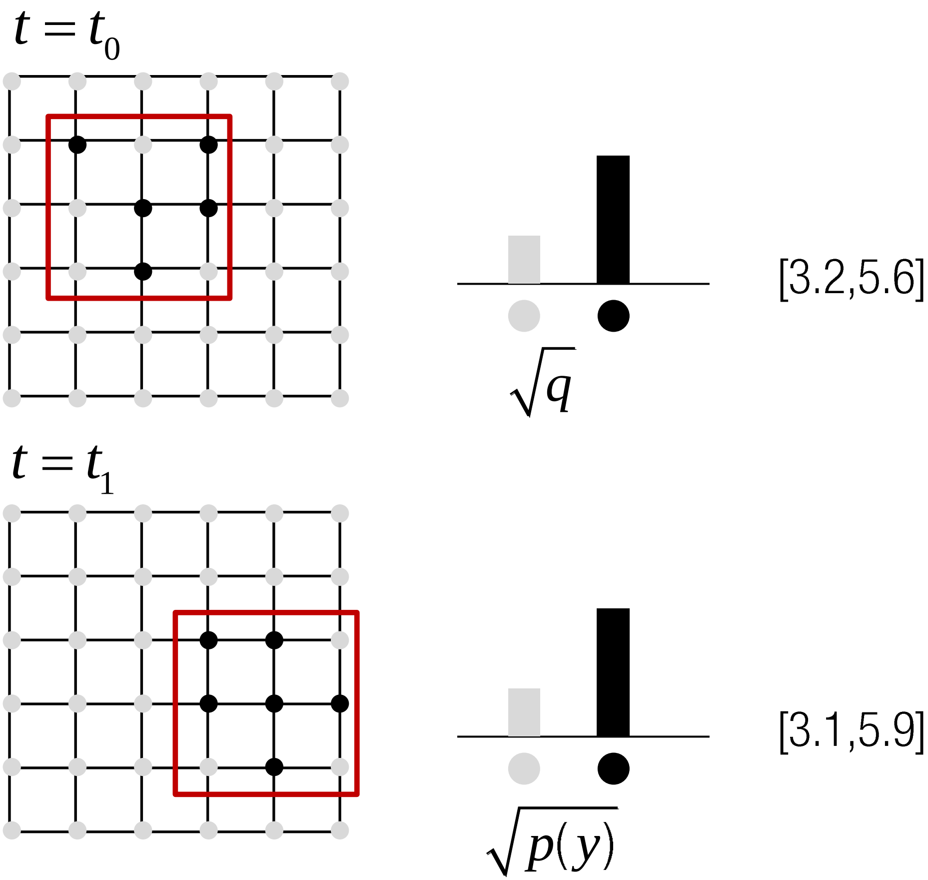
Histogram Match



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

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

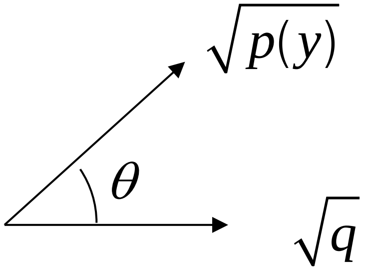
Histogram Match



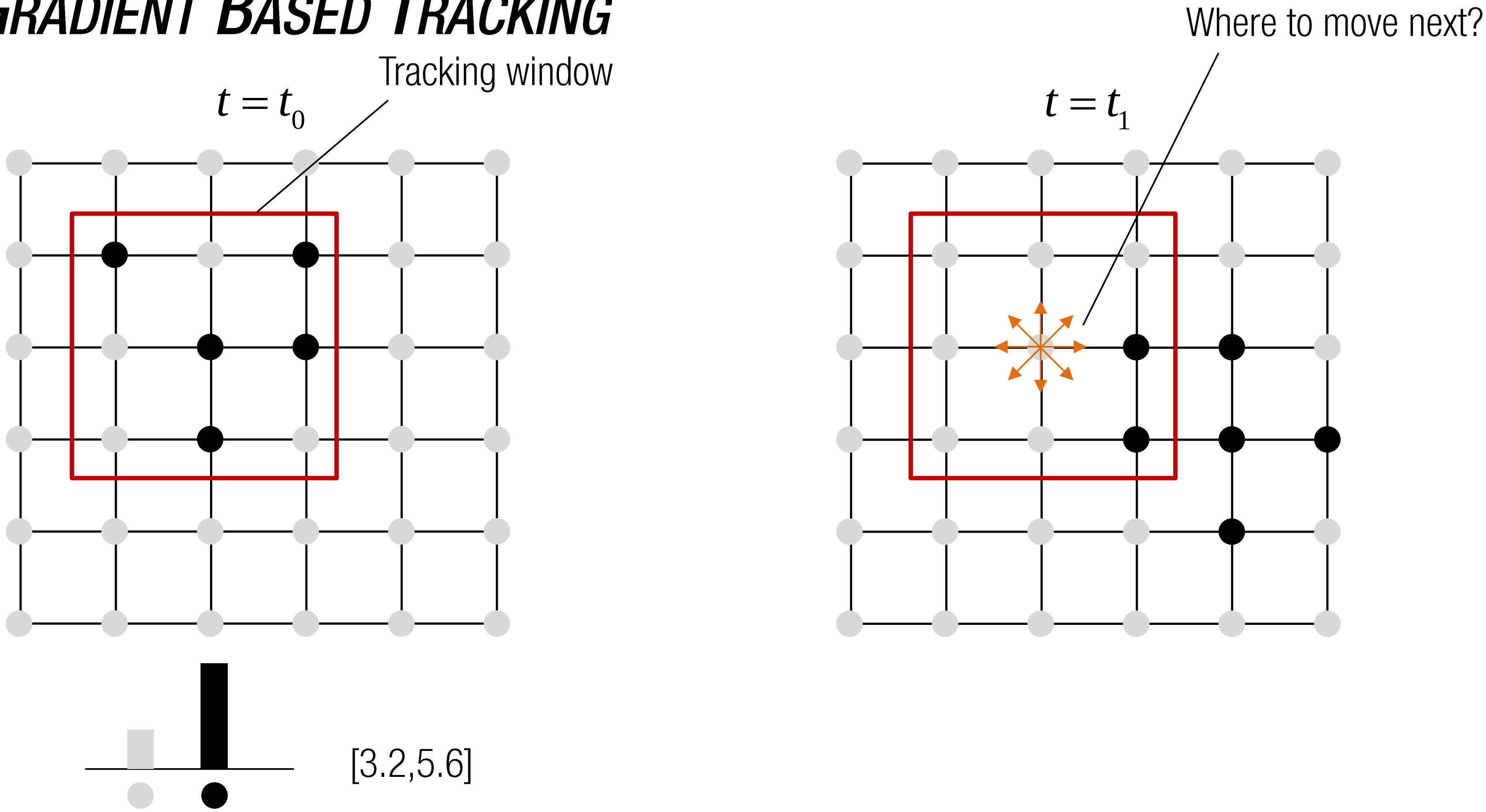
Bhattacharyya coefficient:
A measure of similarity of prob. dist.

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

Cosine distance between prob. dist.



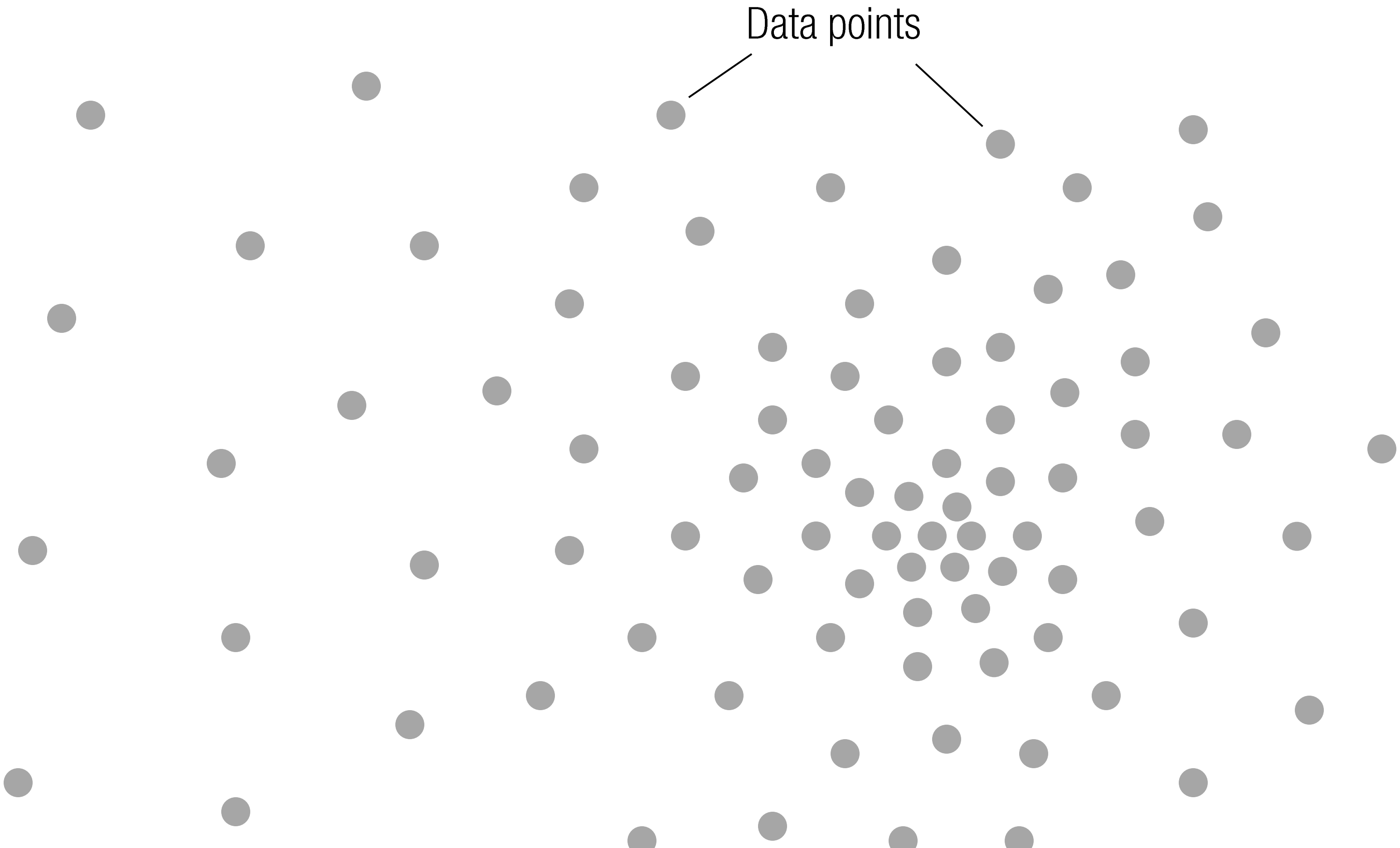
Gradient Based Tracking



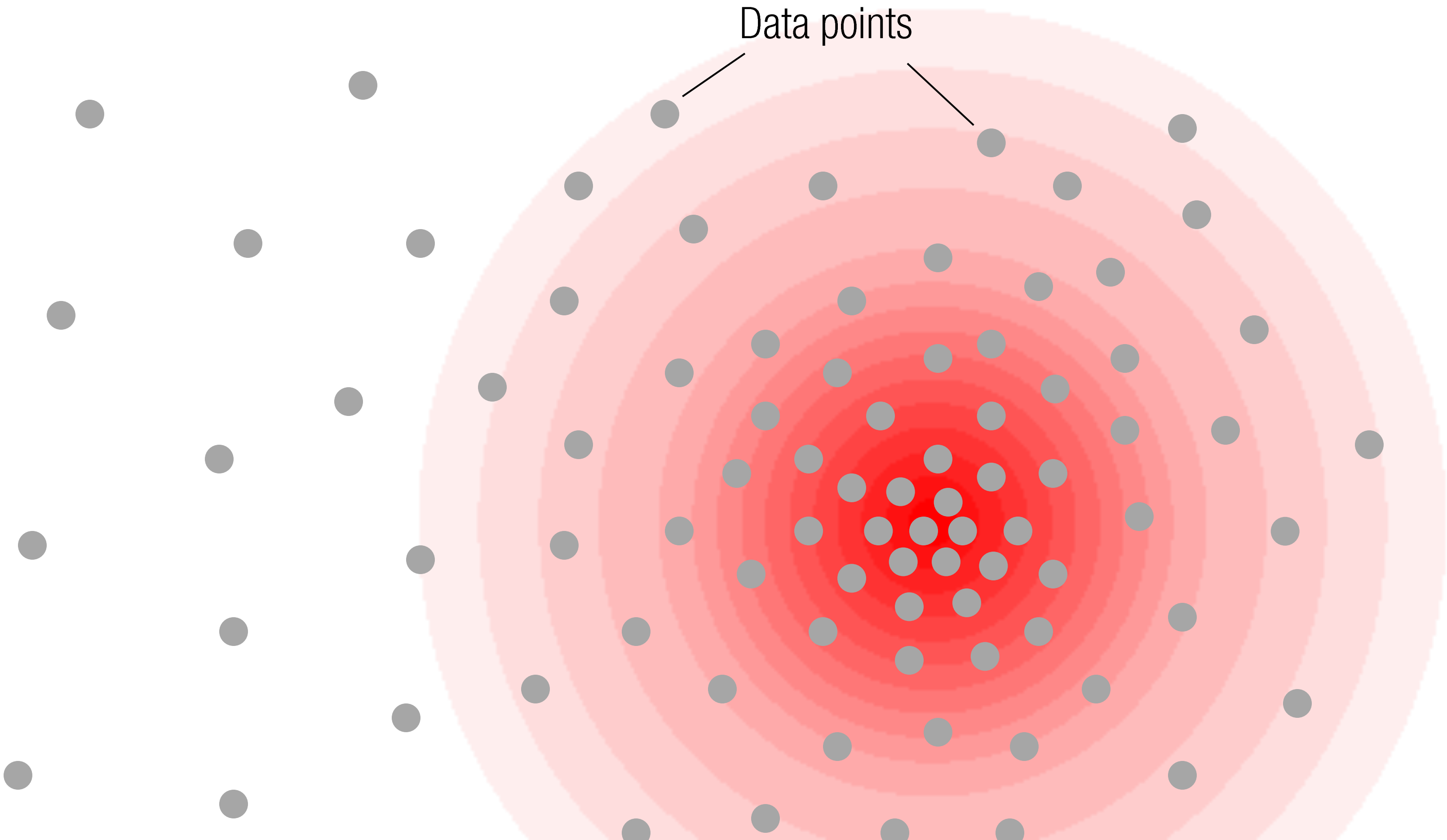
Meanshift Algorithm

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

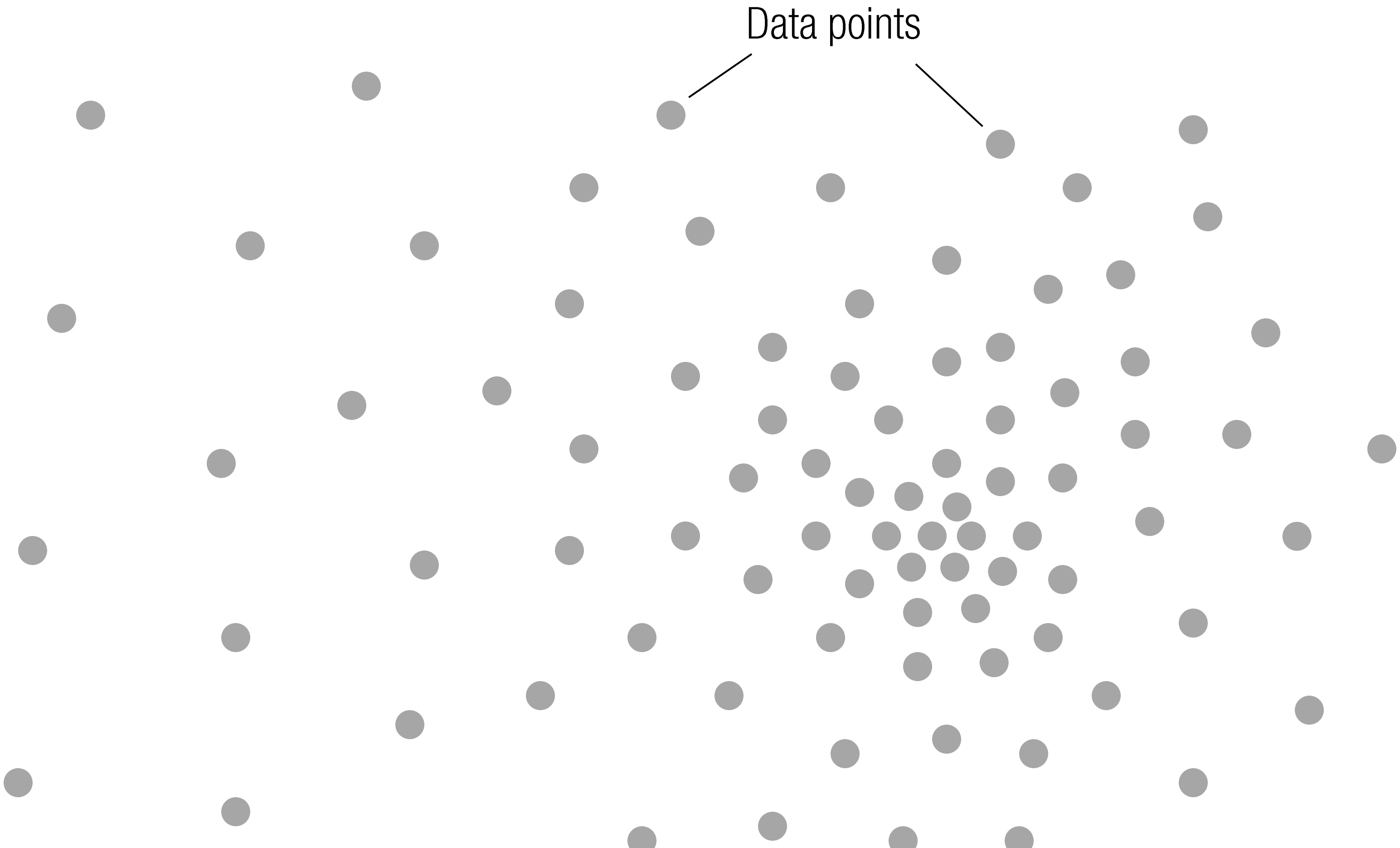
MODE-SEEKING



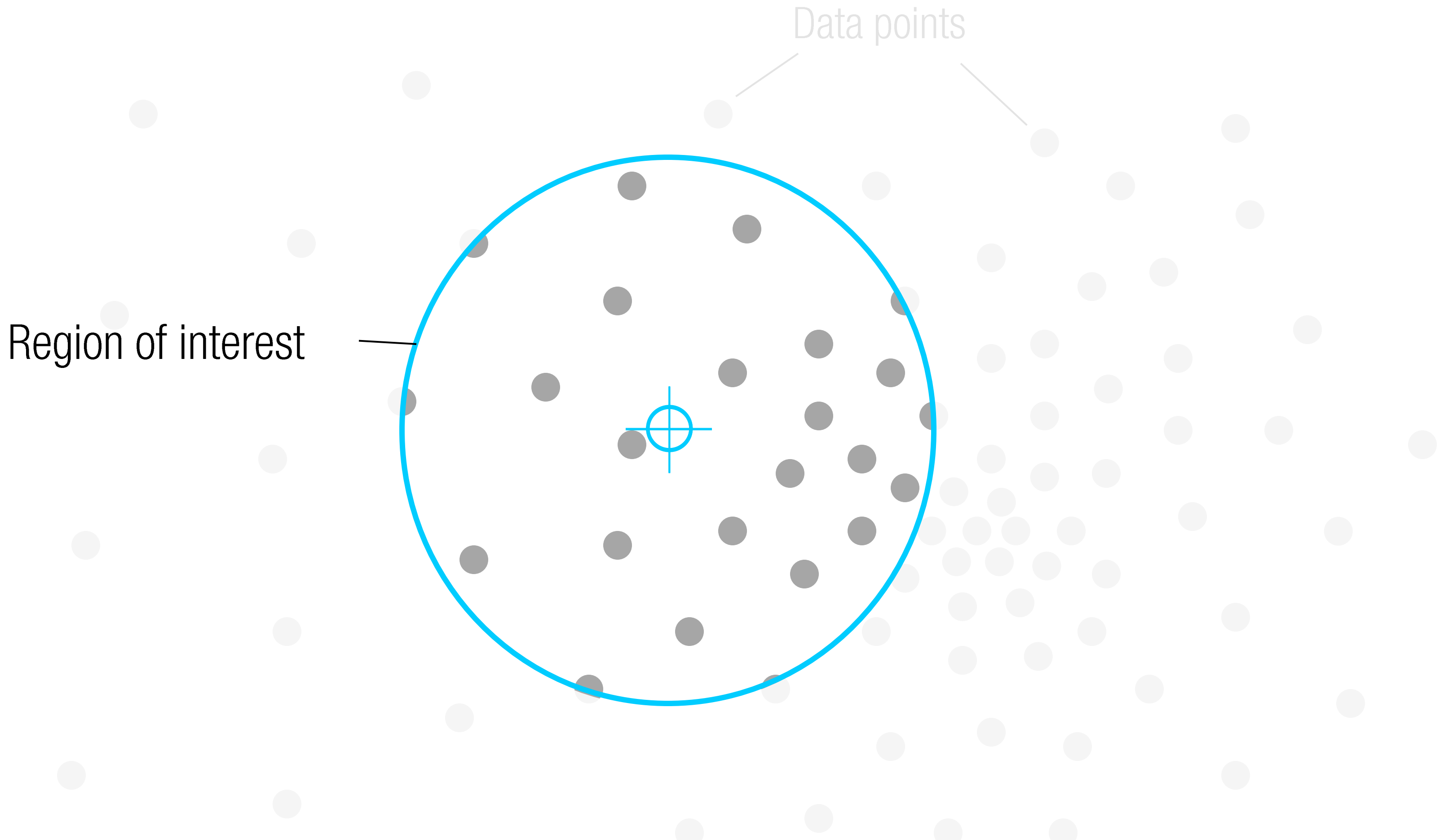
MODE-SEEKING



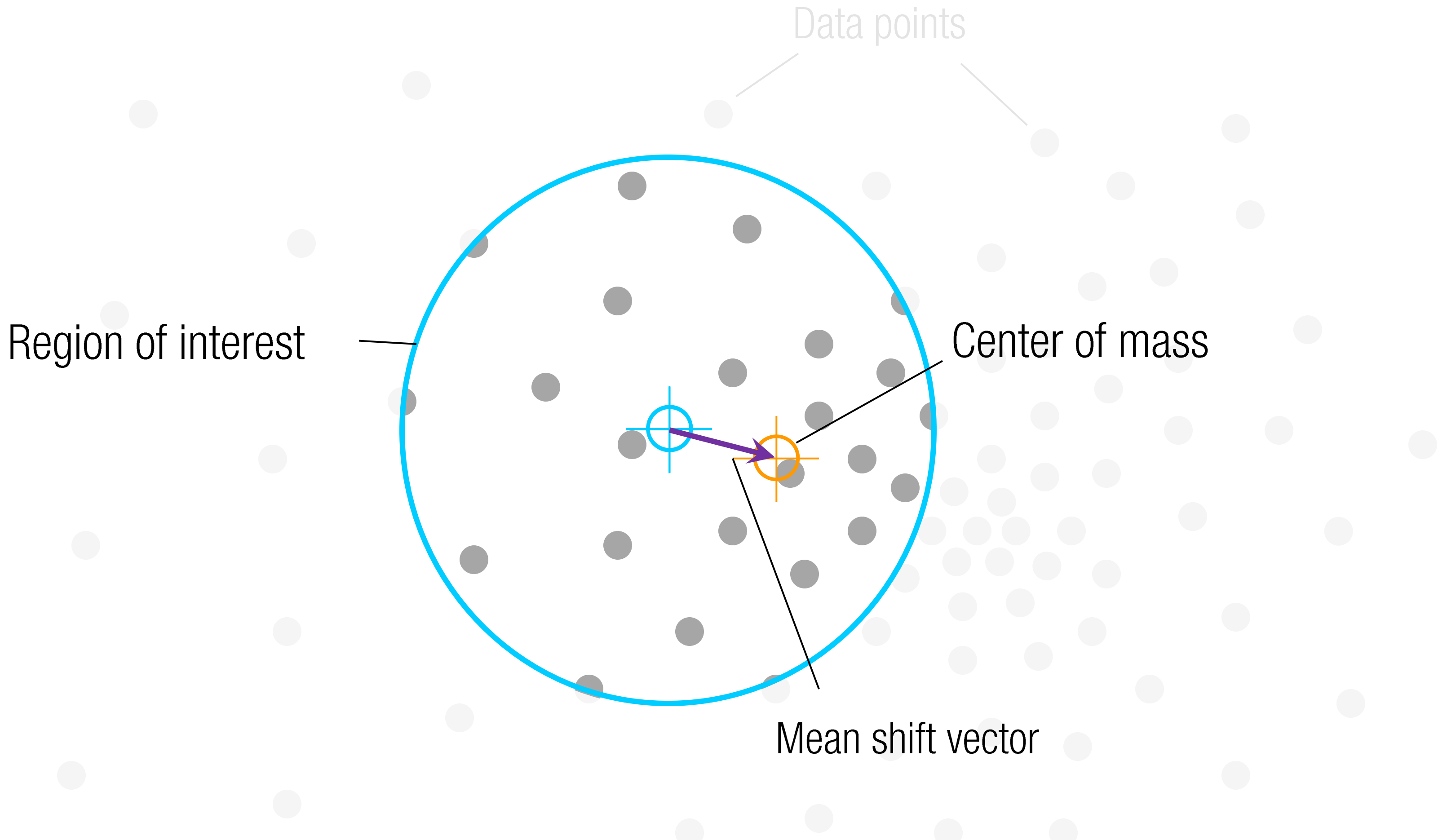
MODE-SEEKING



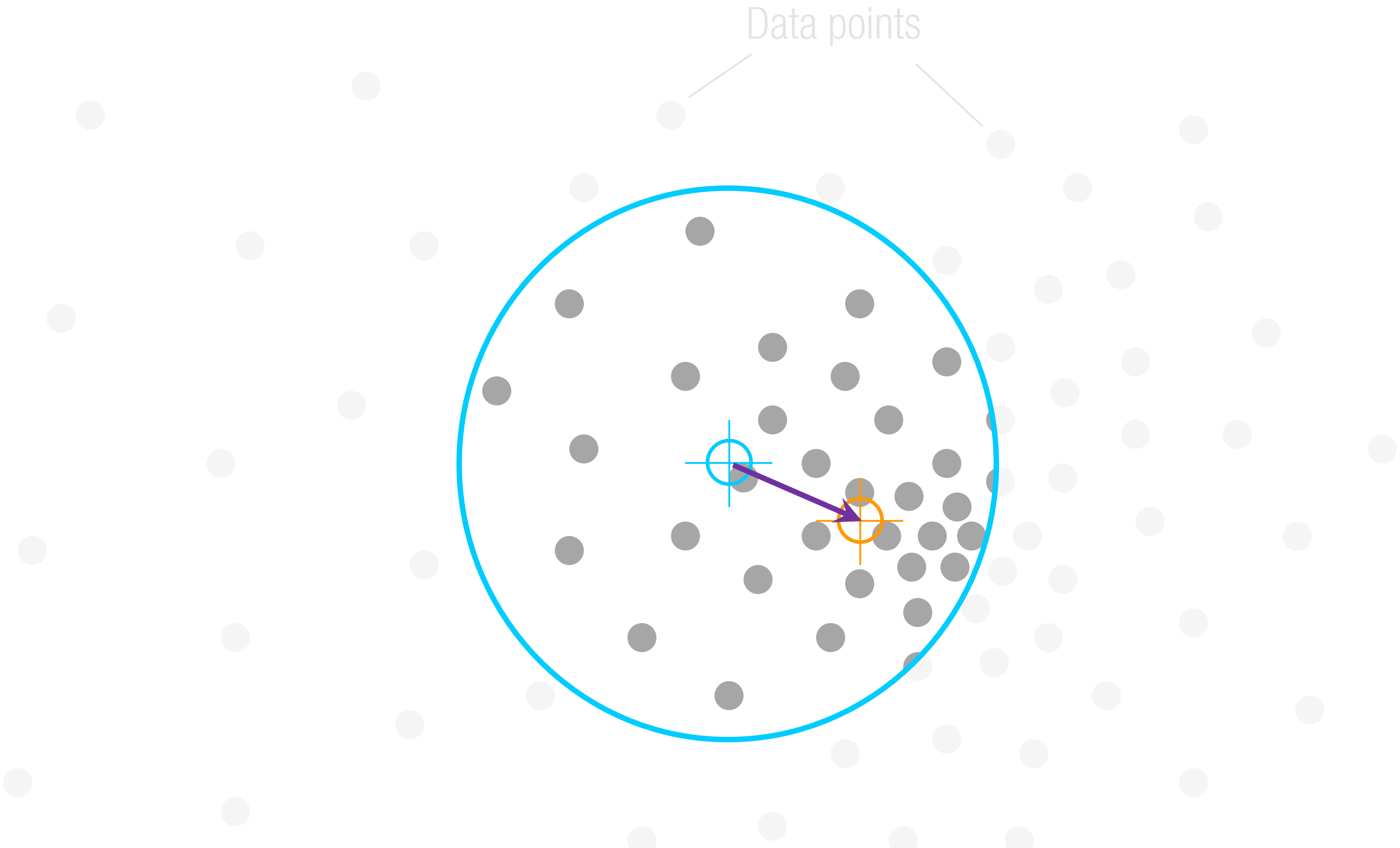
MODE-SEEKING



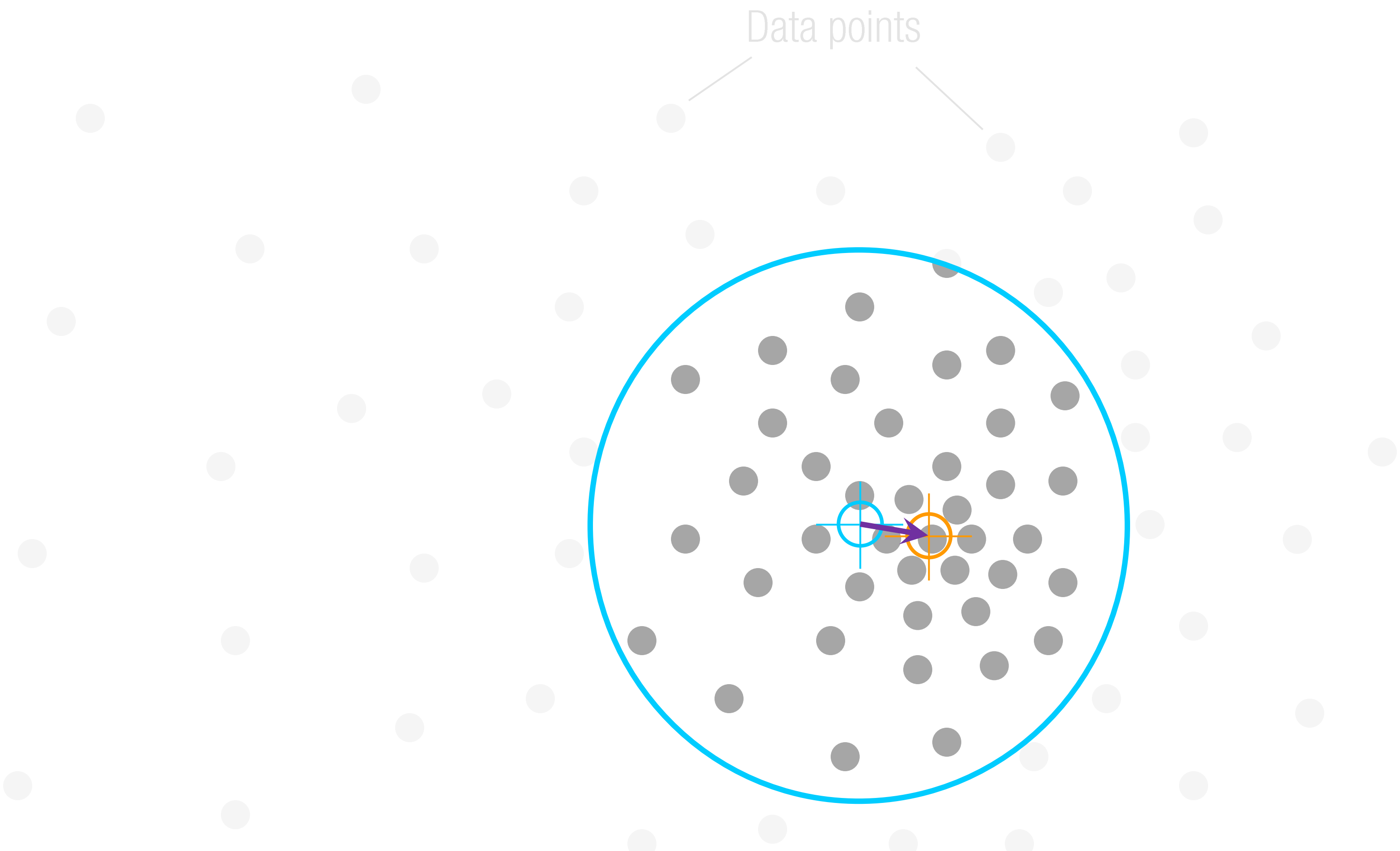
MODE-SEEKING



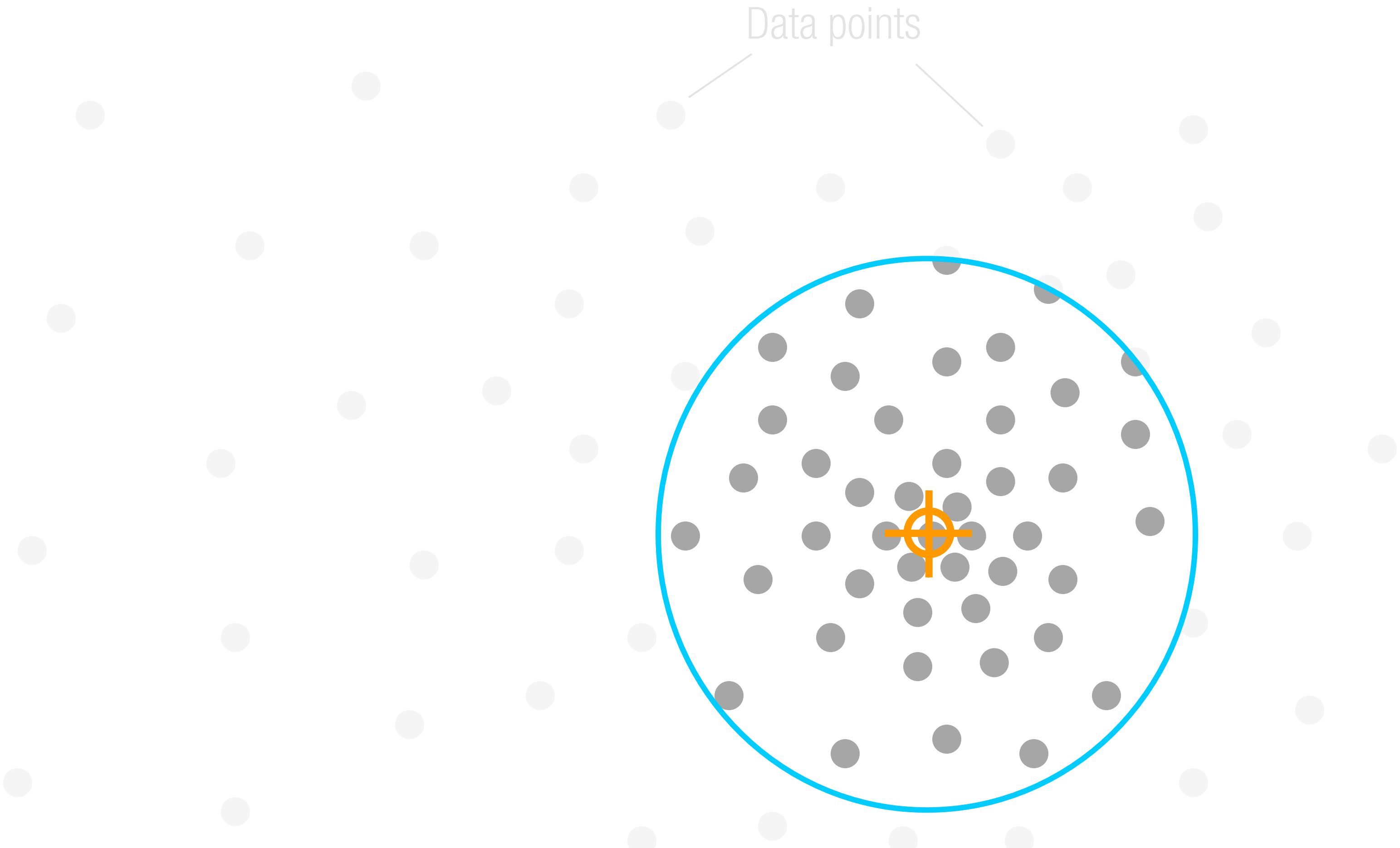
MODE-SEEKING



MODE-SEEKING



MODE-SEEKING



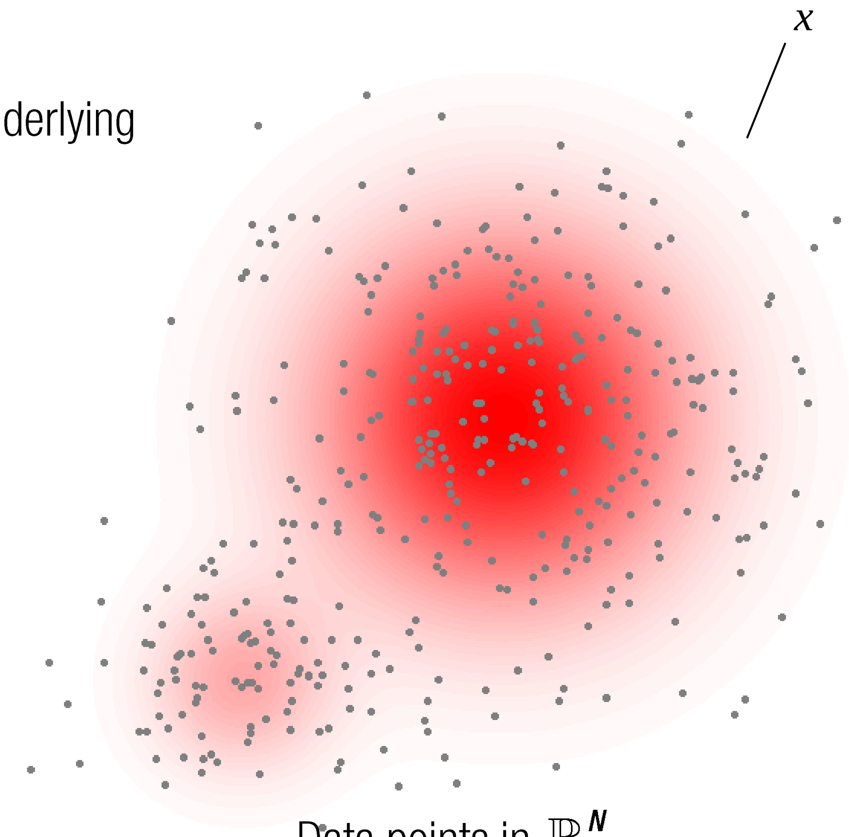
Data Density Estimation

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

Data Density Estimation

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

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

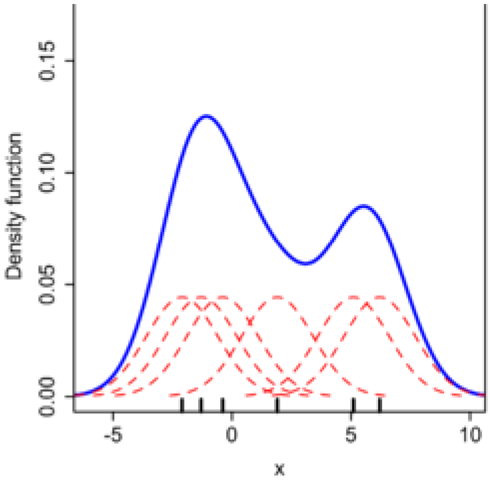


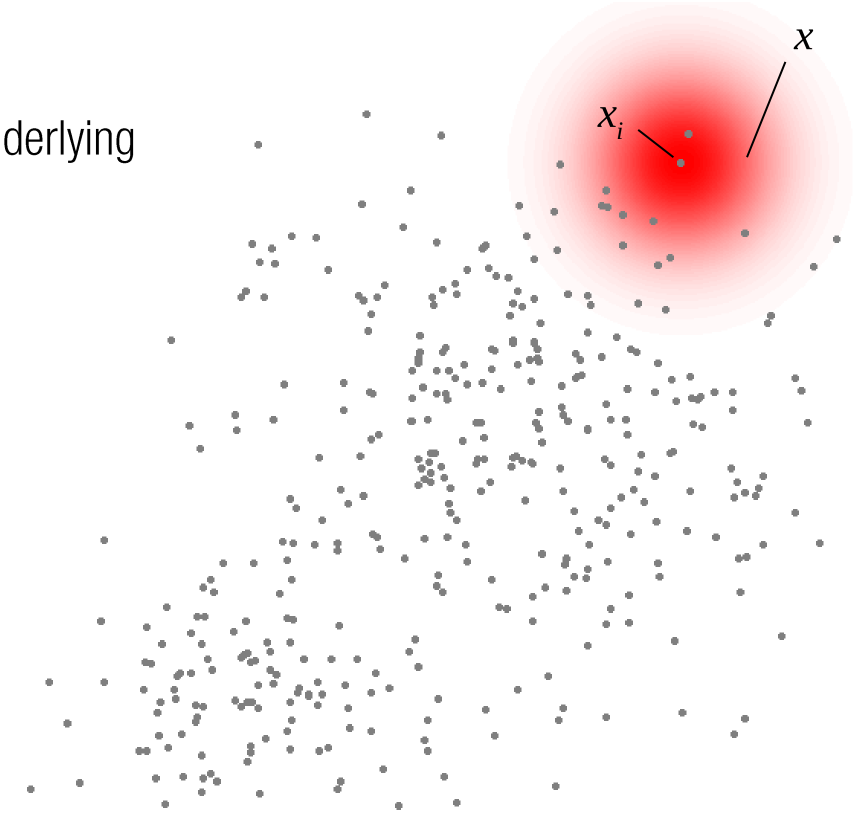
Data points in $\mathbb{R}^N$

DATA DENSITY ESTIMATION

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

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



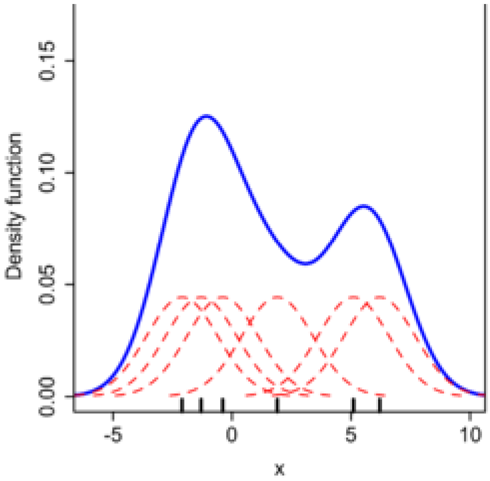


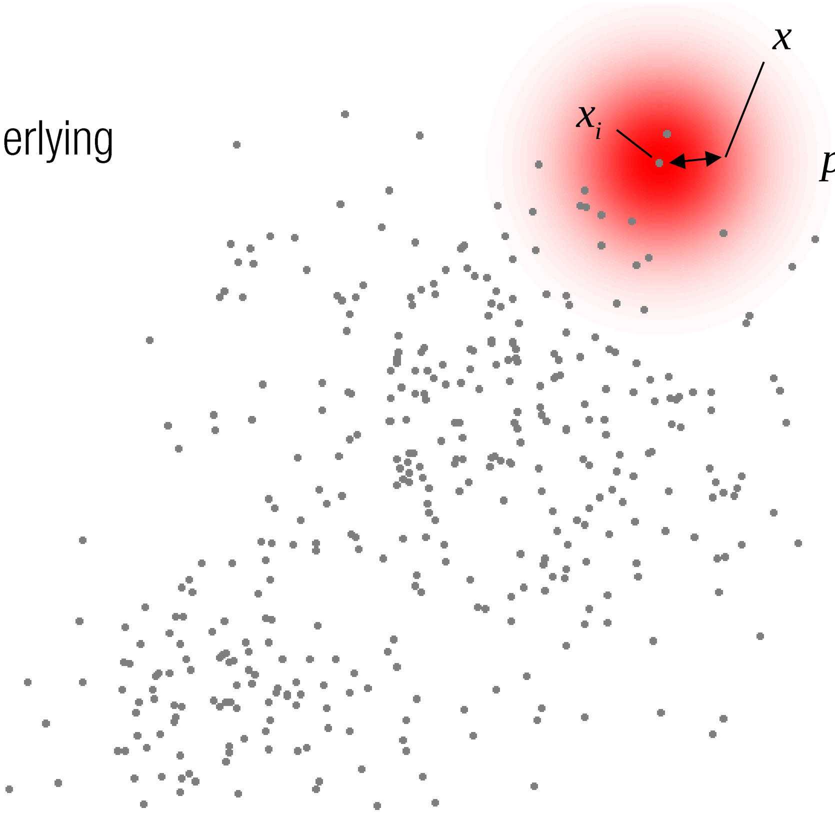
Data points in $\mathbb{R}^N$

DATA DENSITY ESTIMATION

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

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





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

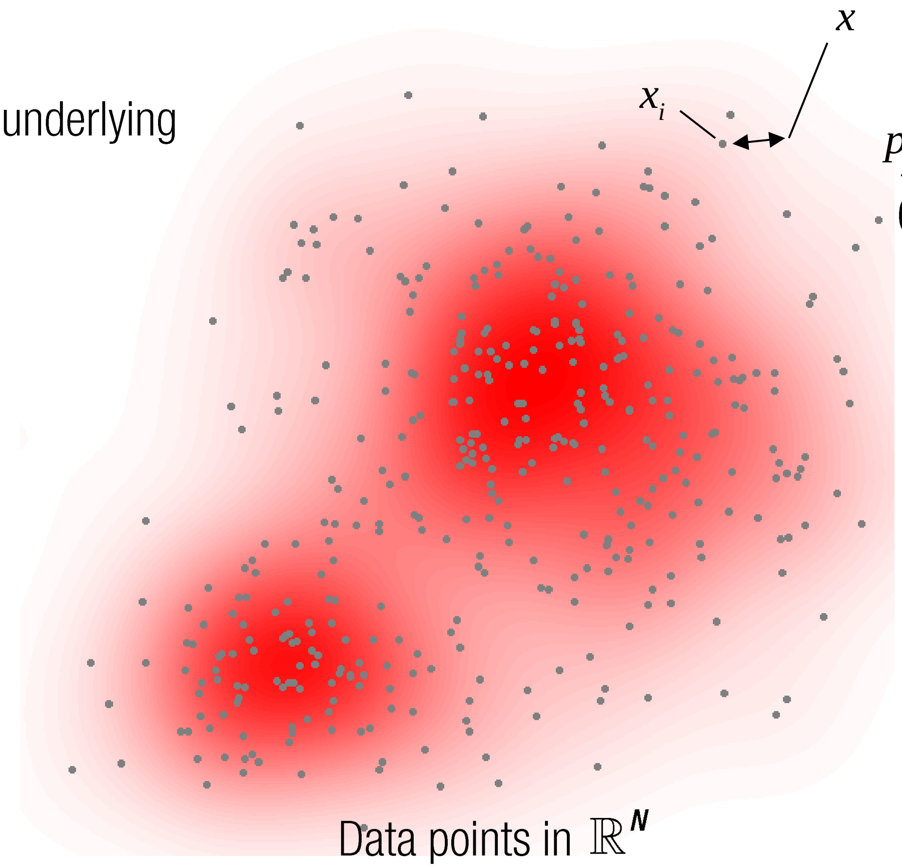
Gaussian kernel

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

Data Density Estimation

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

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



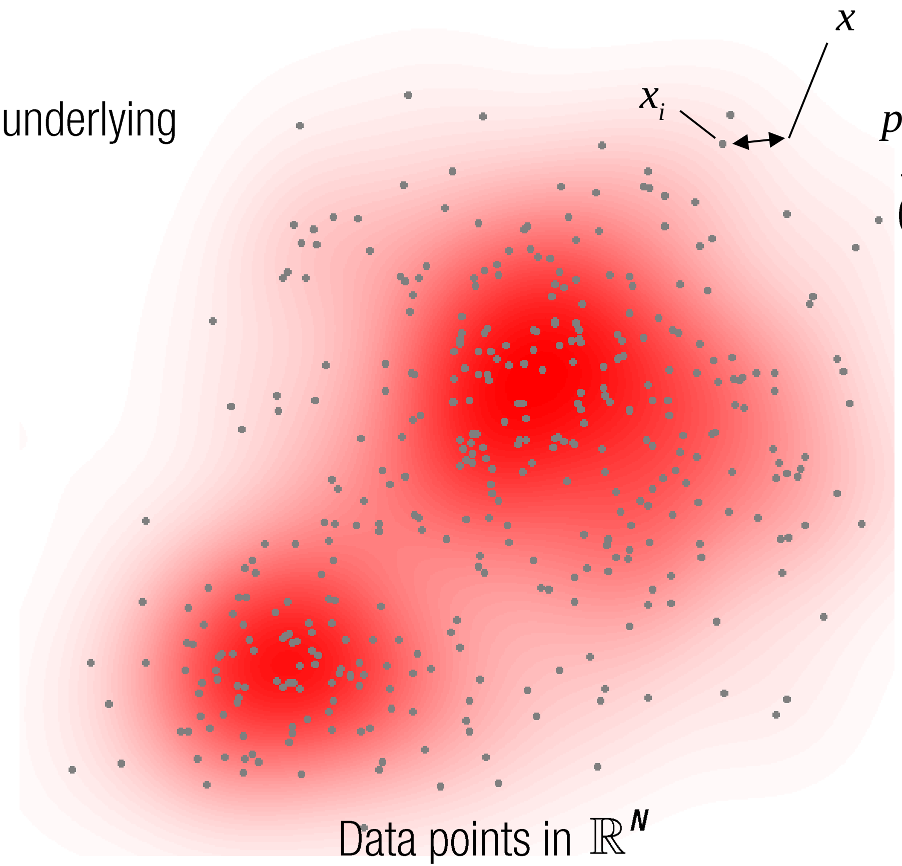
$$p(x \mid x_i) = Ce^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

Gaussian kernel

Data Density Estimation

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

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



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

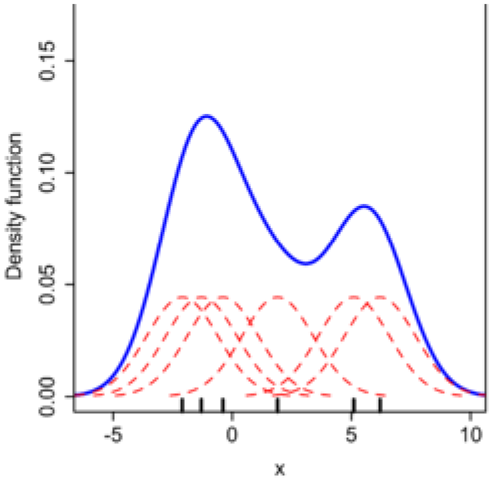
General kernel

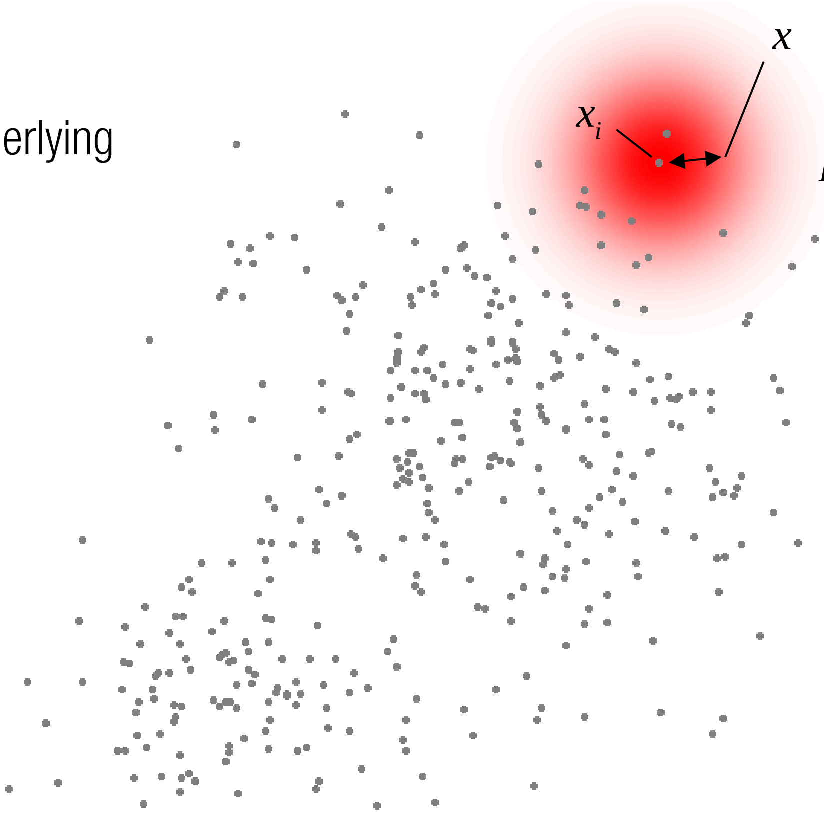
Data points in $\mathbb{R}^N$

Data Density Estimation

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

$$
\begin{aligned}
P(x) &\approx P(x \mid D) \\
&\approx p(x \mid x_1) + \cdots + p(x \mid x_n)
\end{aligned}
$$





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

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

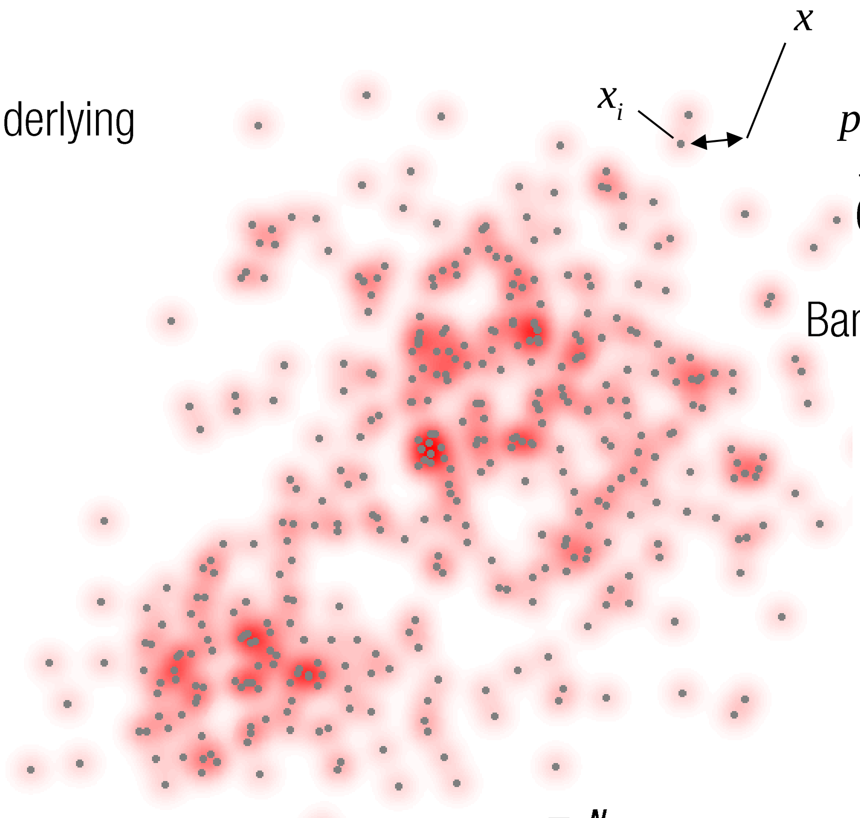
Gaussian kernel

σ: Bandwidth

DATA DENSITY ESTIMATION

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

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



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

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

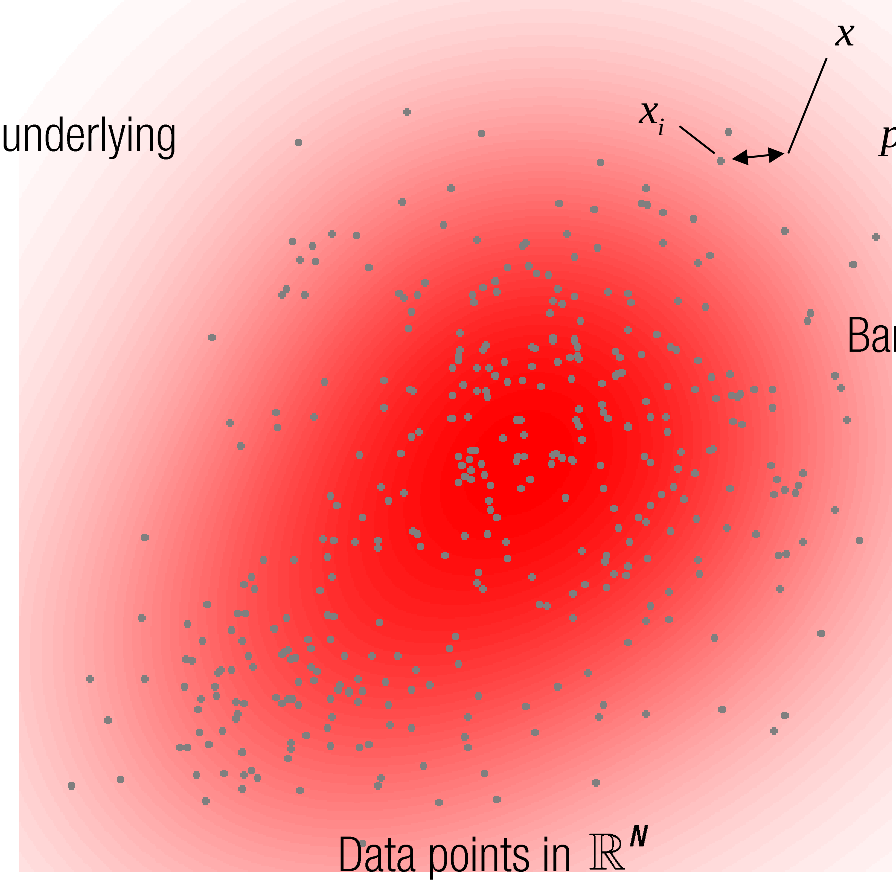
General kernel

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

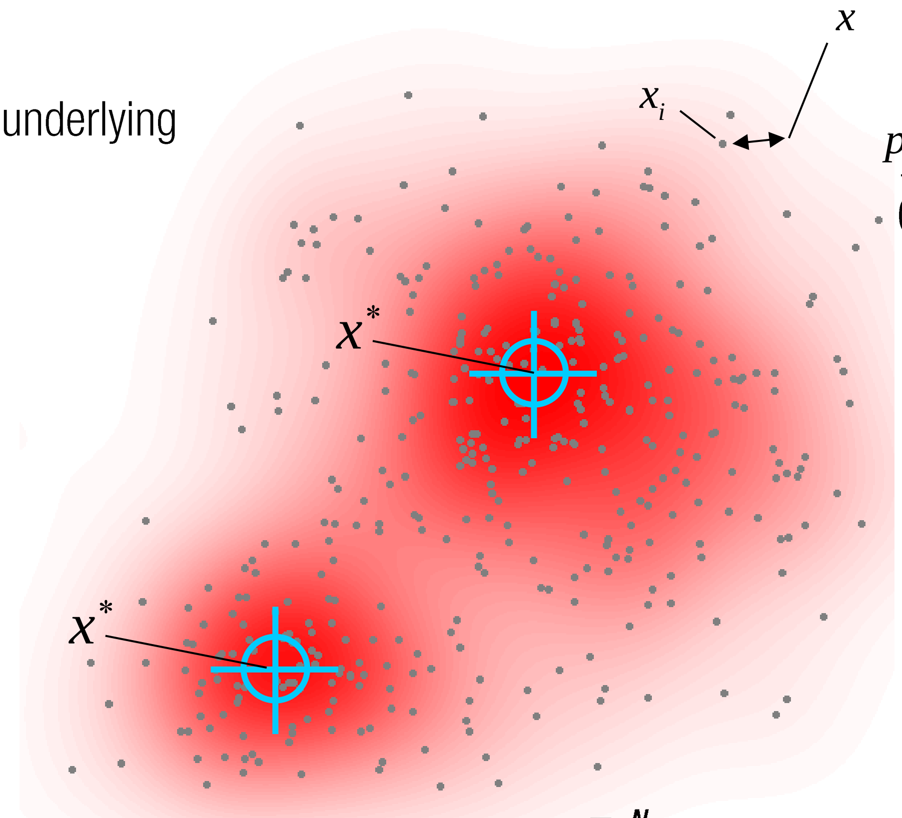
Bandwidth is too big.

MODE-SEEKING ALGORITHM

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

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

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



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

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

Gaussian kernel

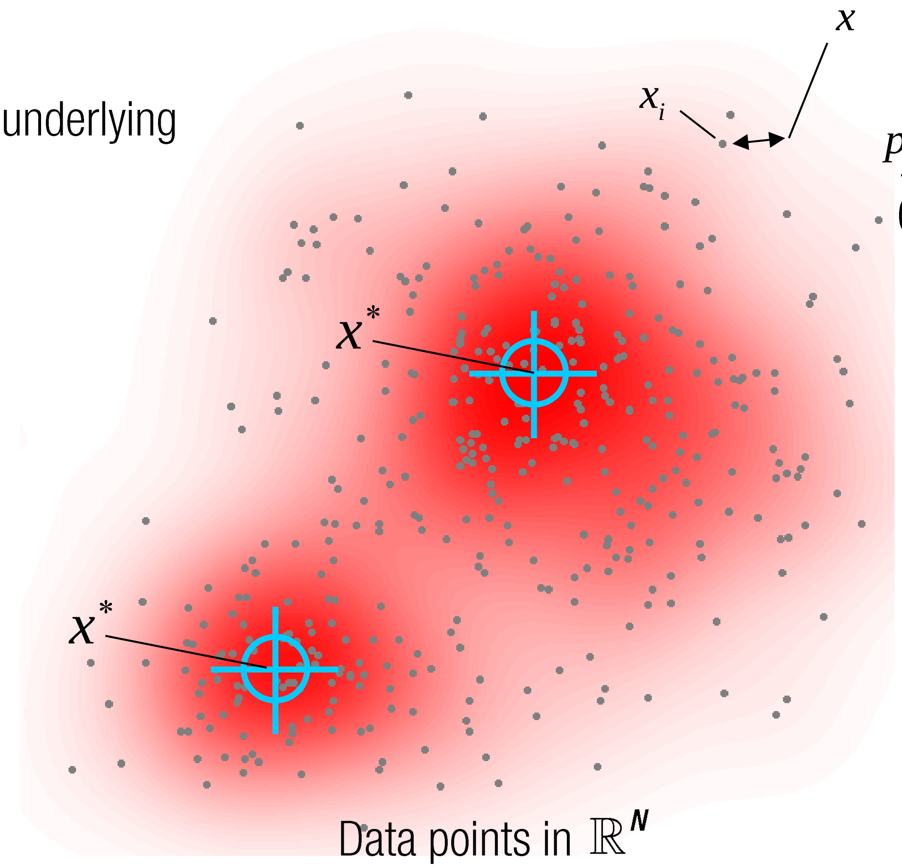
Mode-seeking Algorithm

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

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

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

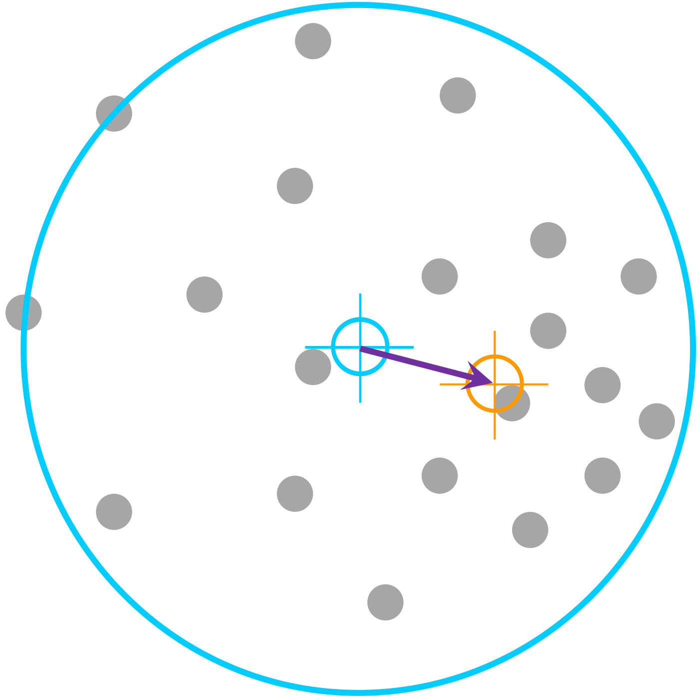
$$\nabla P(x) = 0$$



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

Gaussian kernel

Mode-seeking Algorithm



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

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

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

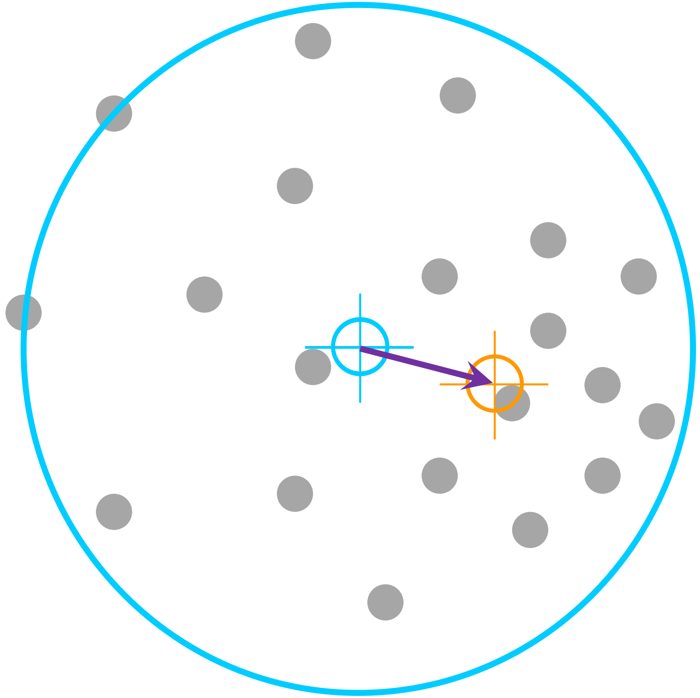
$$= \sum_i \nabla\left(e^{-\frac{(x-x_i)^2}{2\sigma^2}}\right) = \frac{1}{\sigma^2}\sum_i (x_i - x)\left(e^{-\frac{(x-x_i)^2}{2\sigma^2}}\right)$$

$$= \frac{1}{\sigma^2}\sum_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}}\right)\left(\underbrace{\frac{\sum_i x_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}}\right)}{\sum_i \left(e^{-\frac{(x-x_i)^2}{2\sigma^2}}\right)}}_{\text{Weighted mean}} - \underbrace{x}_{\text{Shift}}\right)$$

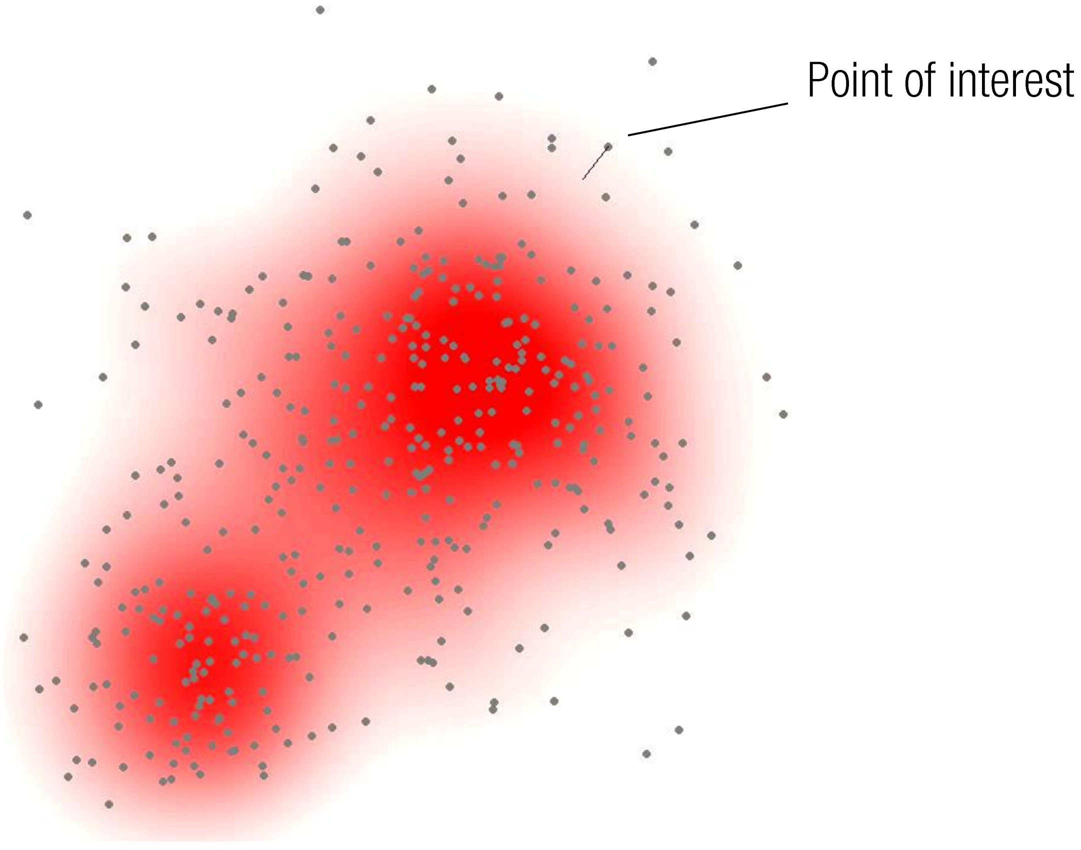
Mode-seeking Algorithm

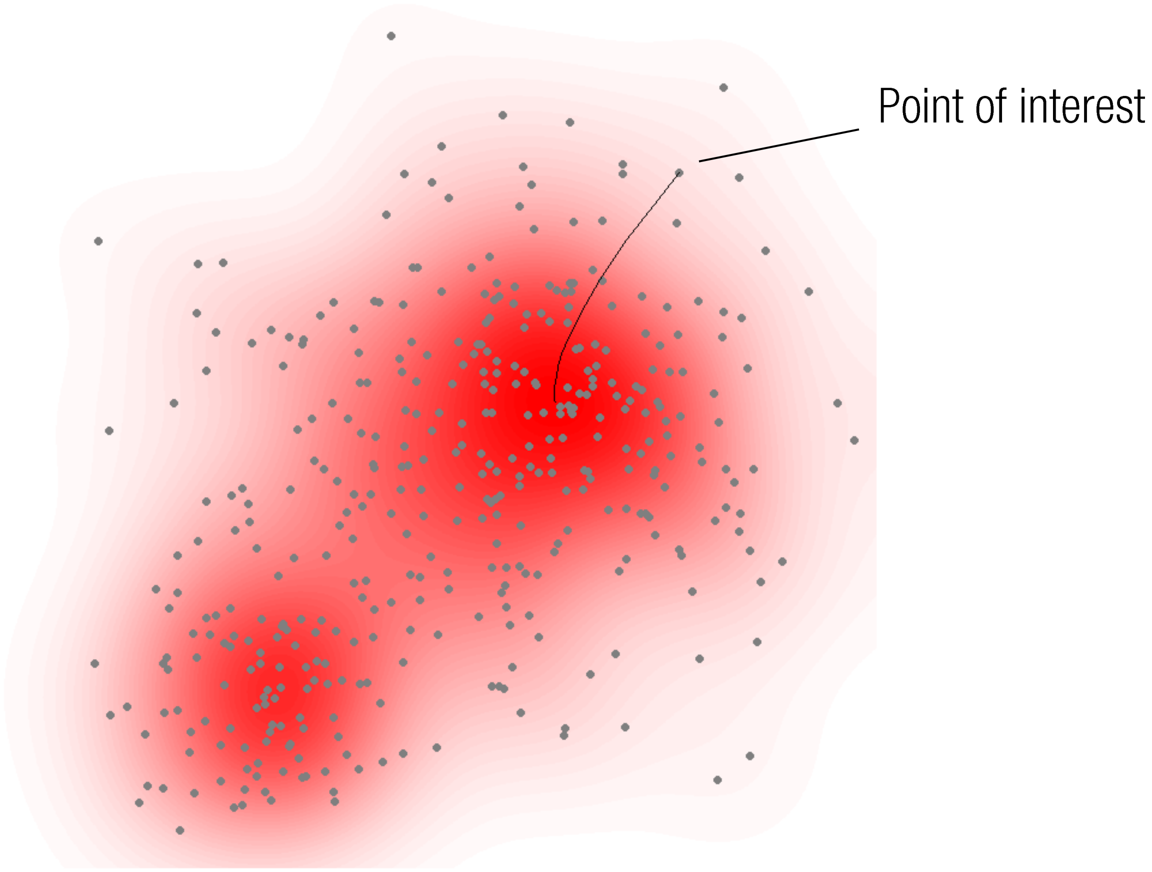
$$x^* = \operatorname*{argmax}_x P(x)$$

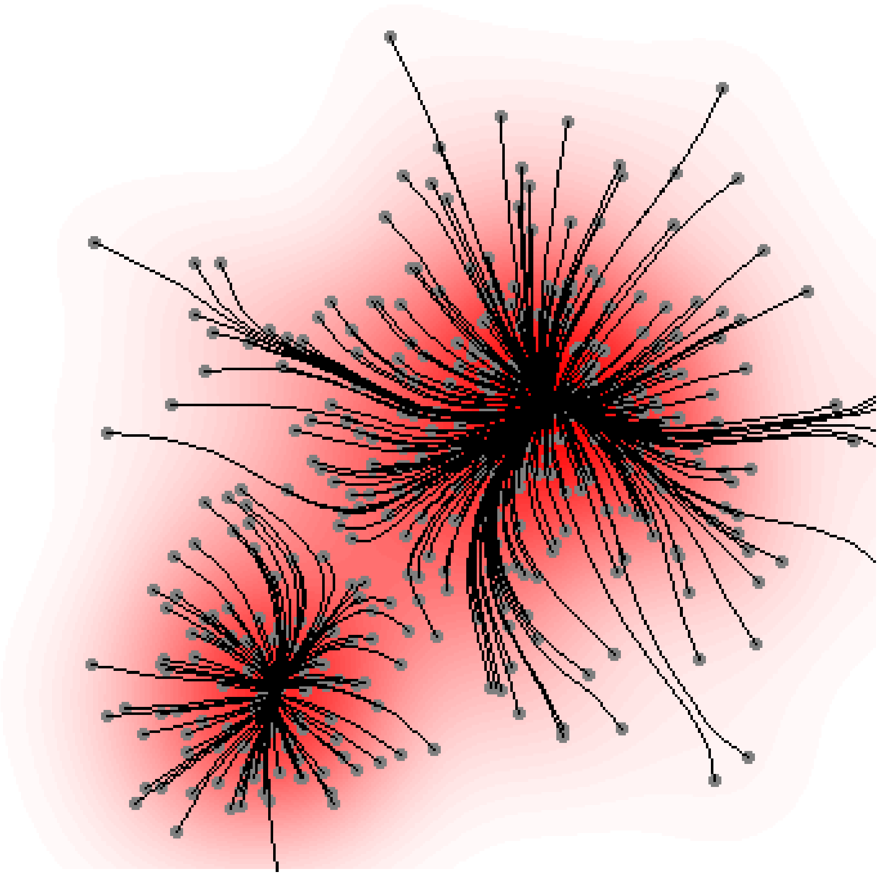
$$\nabla P(x) = 0$$



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


Point of interest


Point of interest

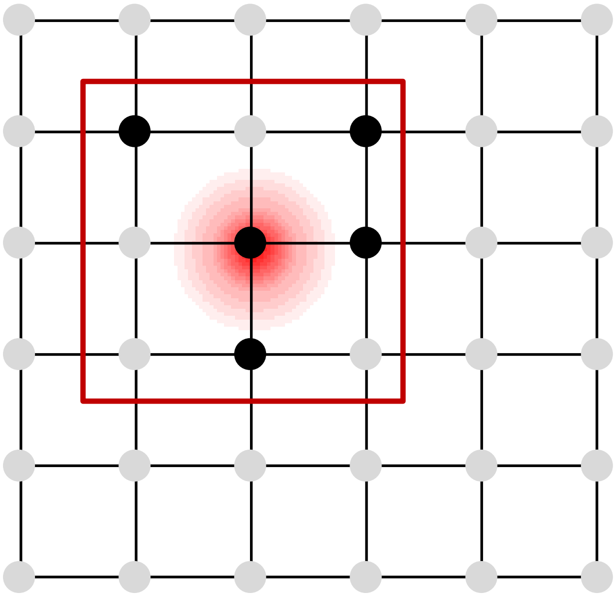


Properties of mean shift

- 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

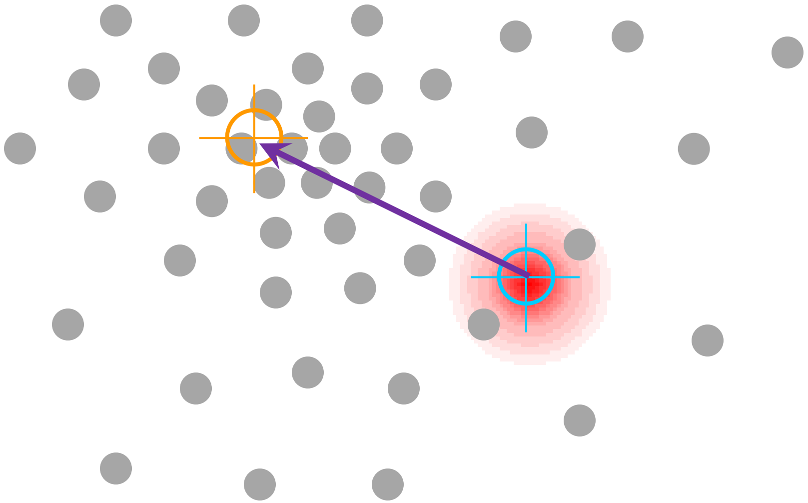
Grid with weight



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

Regular grid with weight

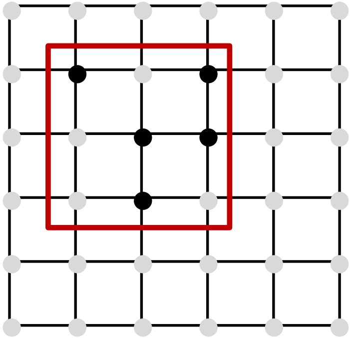
Data sample



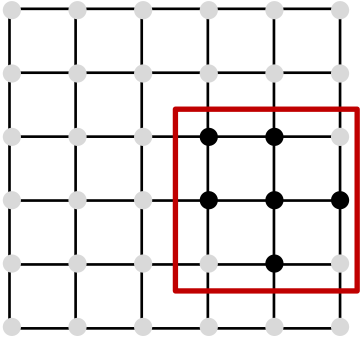
$$p(y) = C\sum_i e^{-\frac{(x-x_i)^2}{2\sigma^2}}$$

Histogram Match

$t = t_0$



$t = t_1$

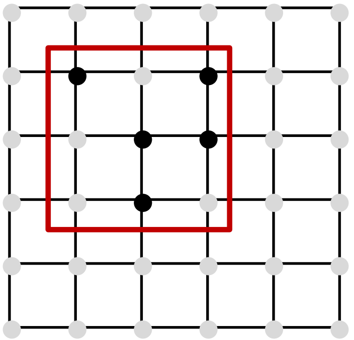


Maximize Bhattacharyya coefficient:

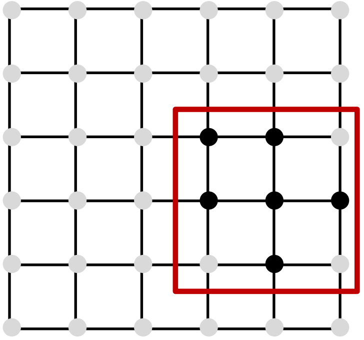
$$
\begin{aligned}
y^* &= \underset{y}{\operatorname{argmax}}\, \rho(y) \\
&= \underset{y}{\operatorname{argmax}} \sum_m \sqrt{p_m(y)\, q_m}
\end{aligned}
$$

Histogram Match

$t = t_0$



$t = t_1$



Maximize Bhattacharyya coefficient:

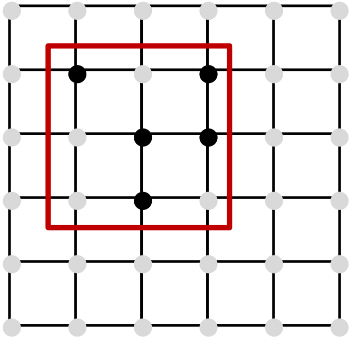
$$y^* = \underset{y}{\operatorname{argmax}}\, \rho(y)$$

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

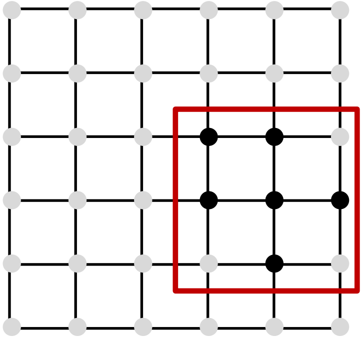
$$\nabla \rho(y) = \frac{1}{2} \sum_m \nabla p_m(y) \sqrt{\frac{q_m}{p_m(y)}}$$

Histogram Match

$t = t_0$



$t = t_1$



Maximize Bhattacharyya coefficient:

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

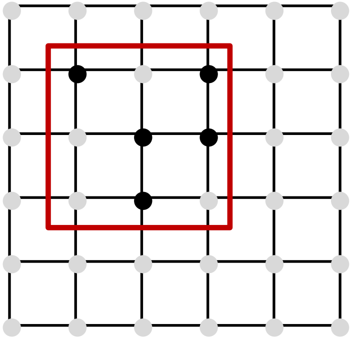
$$= \underset{y}{\operatorname{argmax}} \sum_m \sqrt{p_m(y) q_m}$$

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

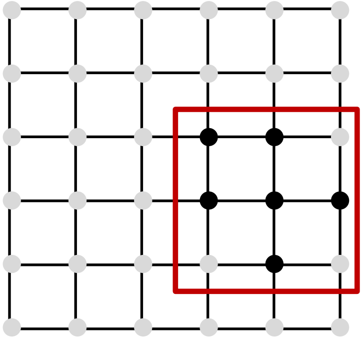
$$= \frac{C}{2} \sum_m \left(\sqrt{\frac{q_m}{p_m(y)}} \sum_i (x_i - y) e^{-\frac{(y - x_i)^2}{2\sigma^2}} \delta\big(b(x_i) - m\big) \right)$$

HISTOGRAM MATCH

$t = t_0$



$t = t_1$



Maximize Bhattacharyya coefficient:

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

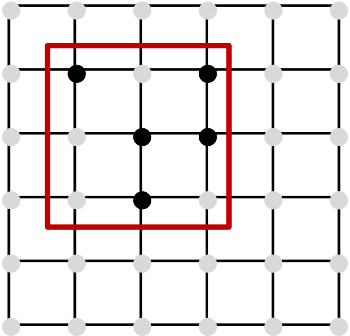
$$= \underset{y}{\operatorname{argmax}} \sum_m \sqrt{p_m(y) q_m}$$

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

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

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

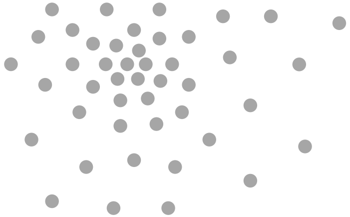
Comparison



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

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

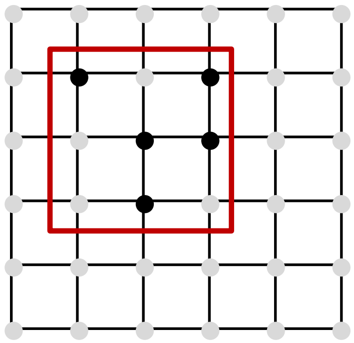
where $w_i = \sum_m \sqrt{\frac{q_m}{p_m(y)}}\,\delta\big(b(x_i) - m\big)$

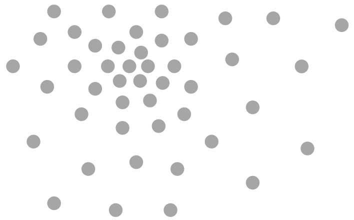


$$y^* = \operatorname*{argmax}_y P(y)$$

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

Comparison





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

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

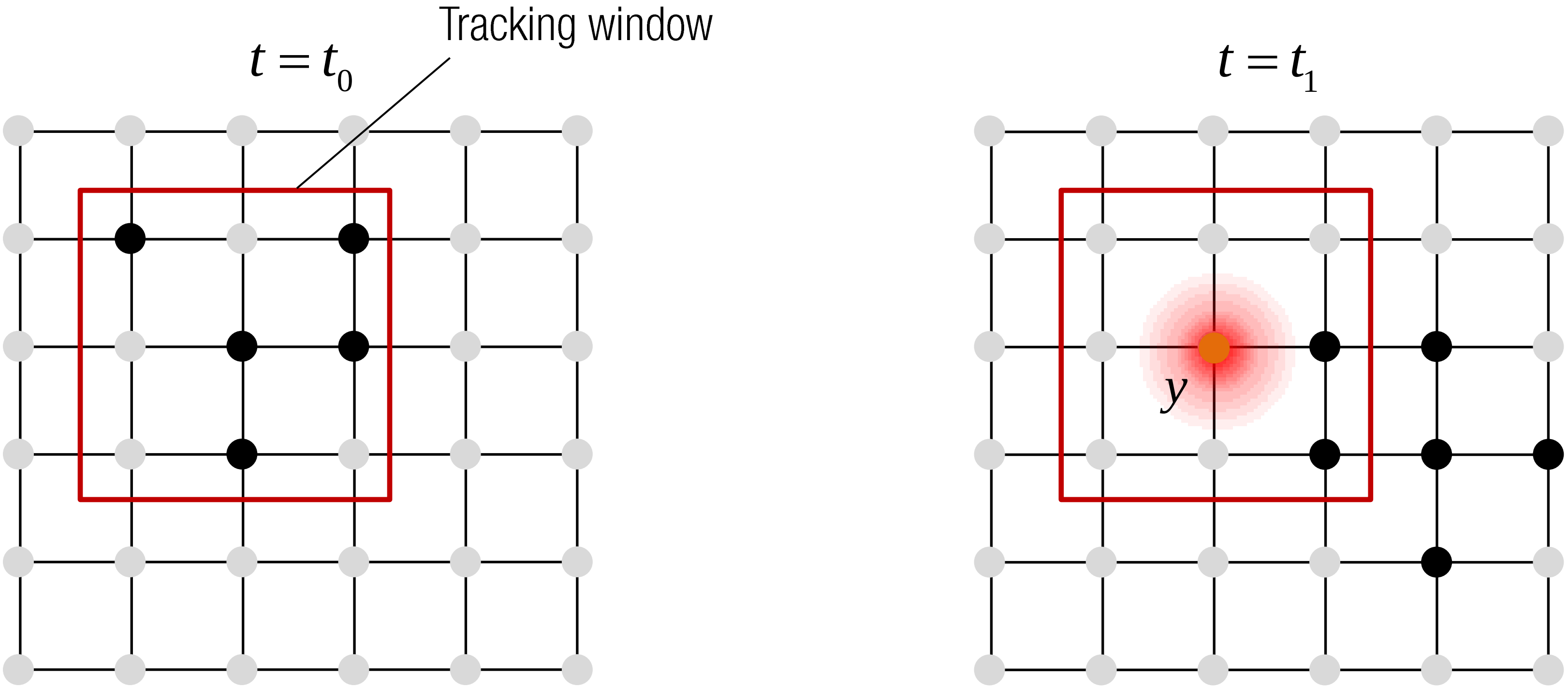
$$\nabla \rho(y) = \frac{C}{2} \sum_i (x_i - y) w_i e^{-\frac{(y-x_i)^2}{2\sigma^2}}$$

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

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

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

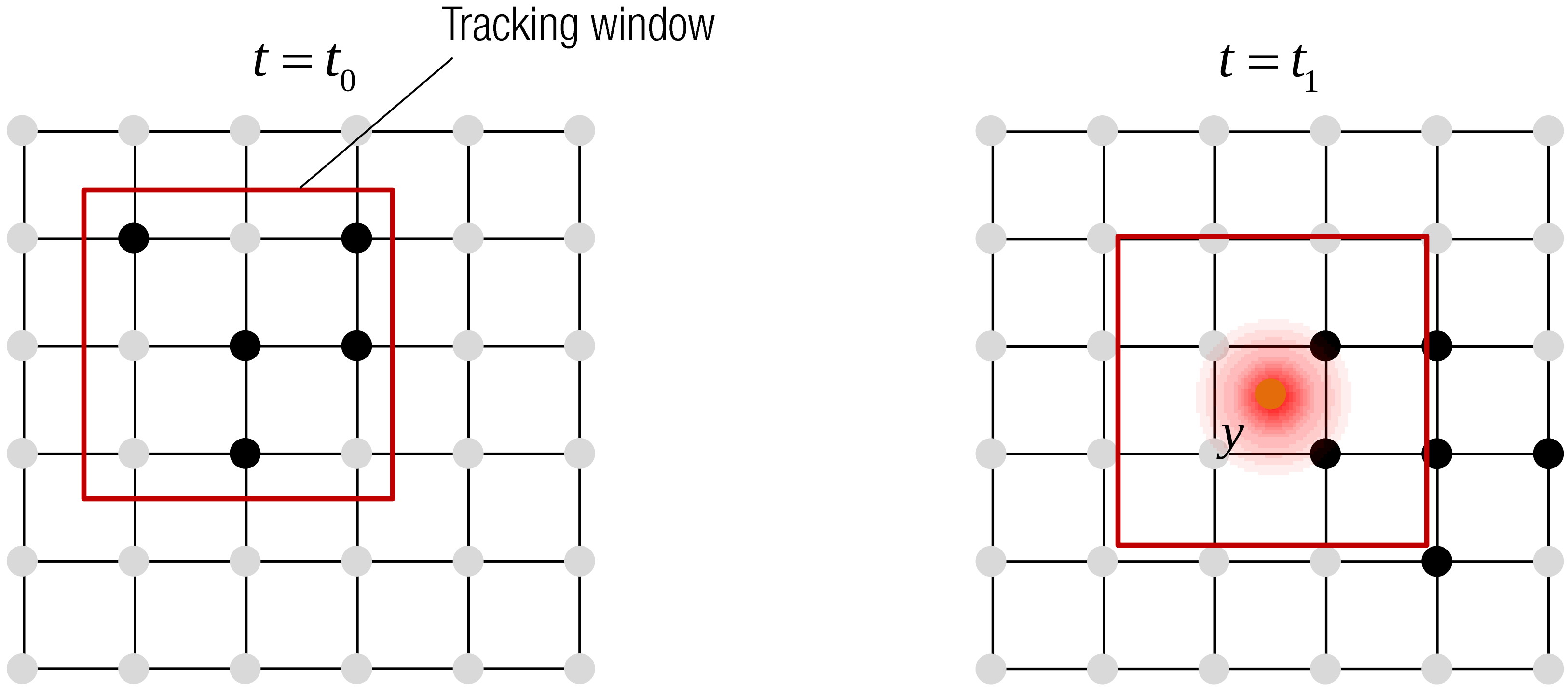
Nonrigid Tracking For Binary Image



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

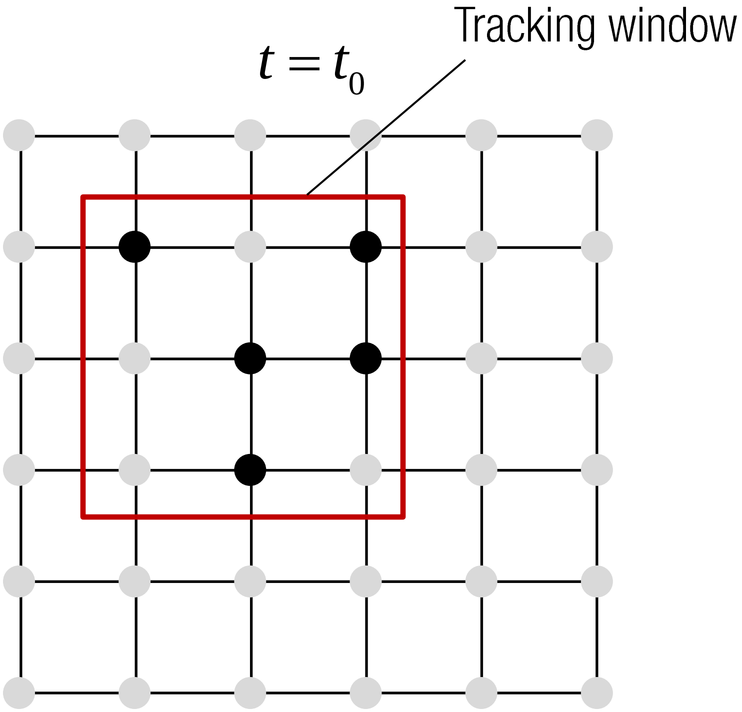
$$p(y) = \begin{bmatrix} p_0(y) & p_1(y) \end{bmatrix}$$

Nonrigid Tracking For Binary Image

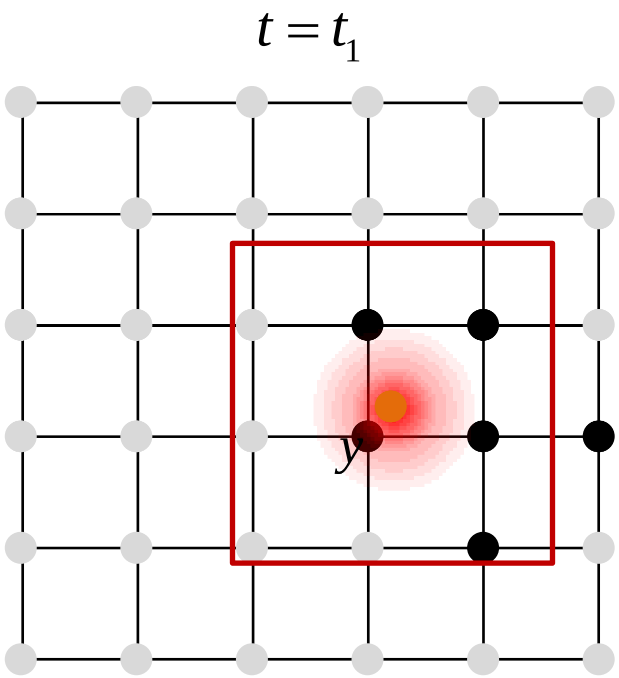


$$q = \begin{bmatrix} q_0 & q_1 \end{bmatrix} \qquad p(y) = \begin{bmatrix} p_0(y) & p_1(y) \end{bmatrix}$$

NONRIGID TRACKING FOR BINARY IMAGE

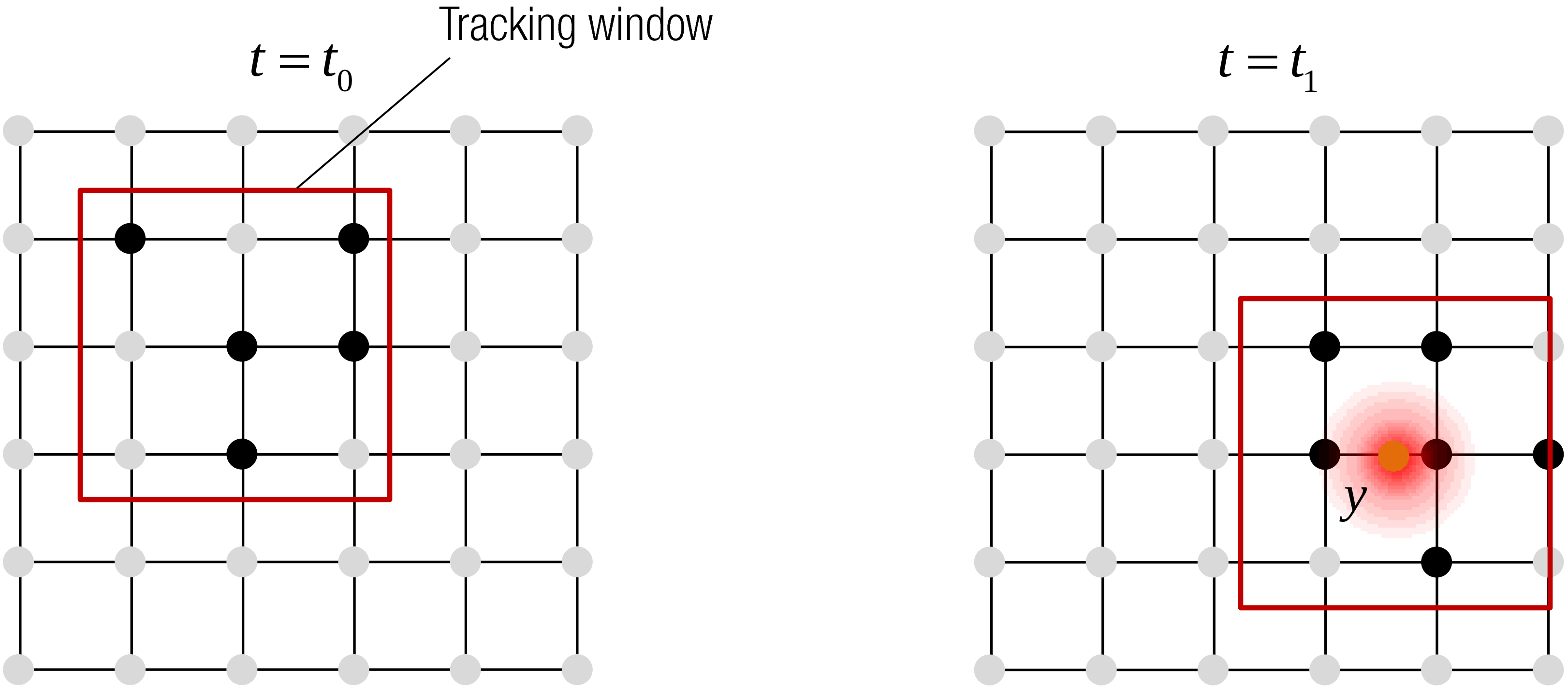


$$q = \begin{bmatrix} q_0 & q_1 \end{bmatrix}$$



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

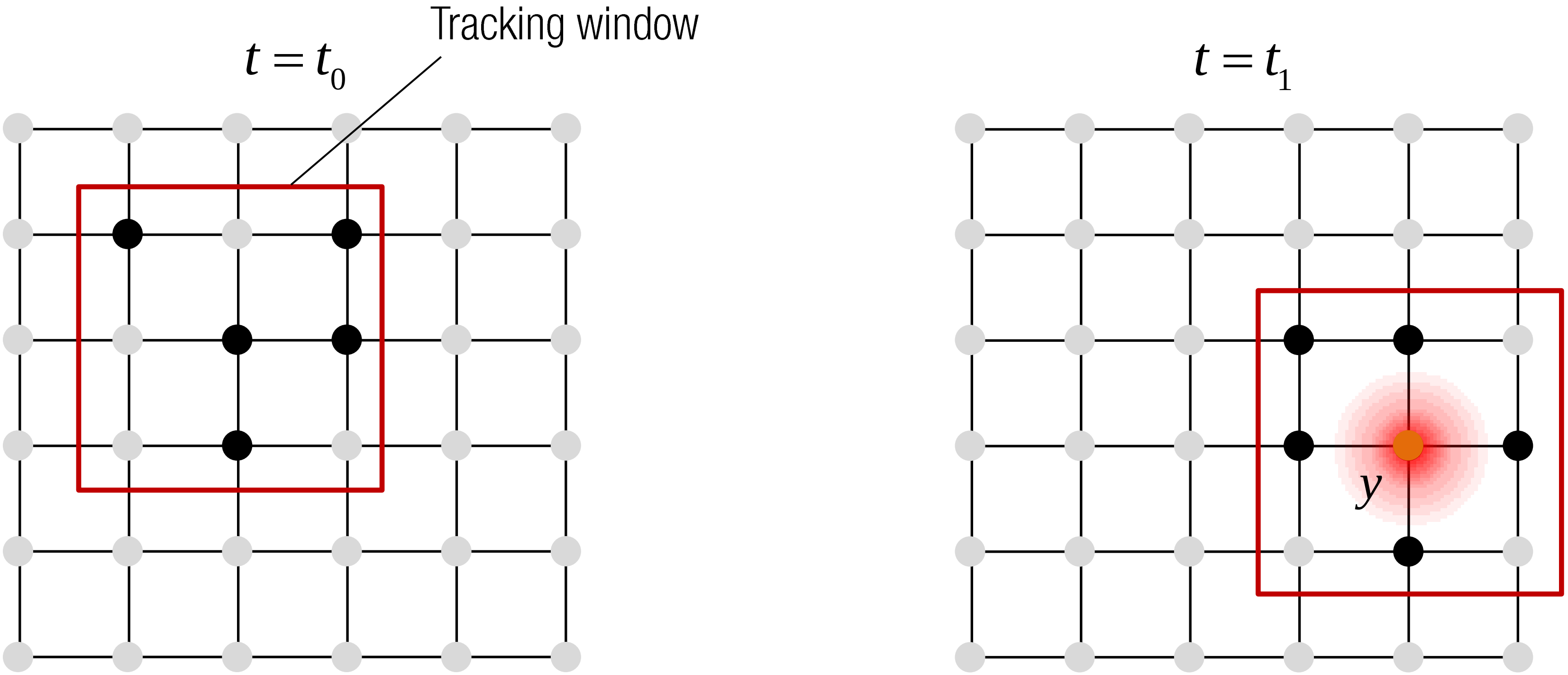
Nonrigid Tracking For Binary Image



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

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

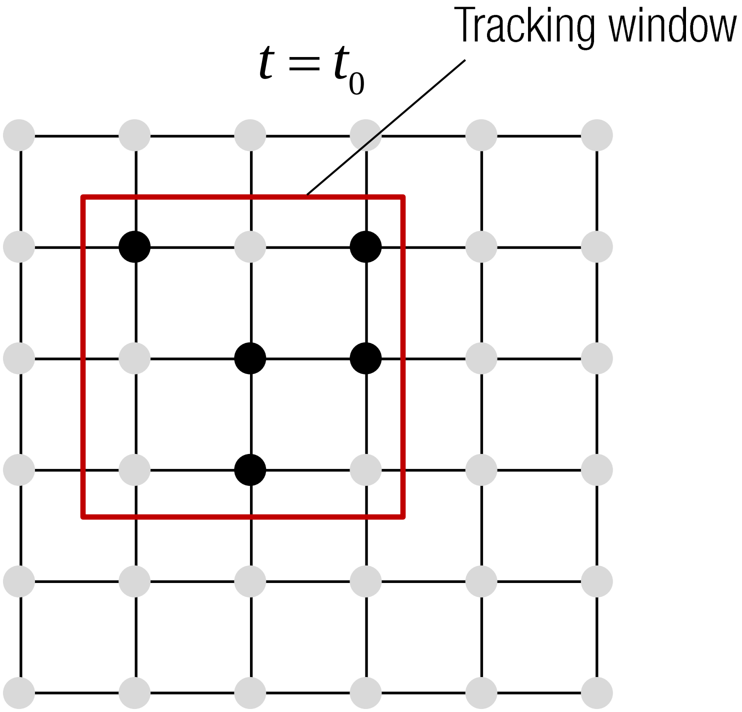
Nonrigid Tracking For Binary Image



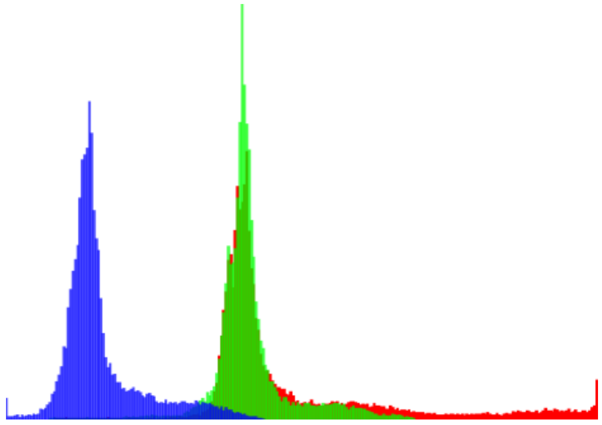
$$q = \begin{bmatrix} q_0 & q_1 \end{bmatrix}$$

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

Nonrigid Tracking For Color Image

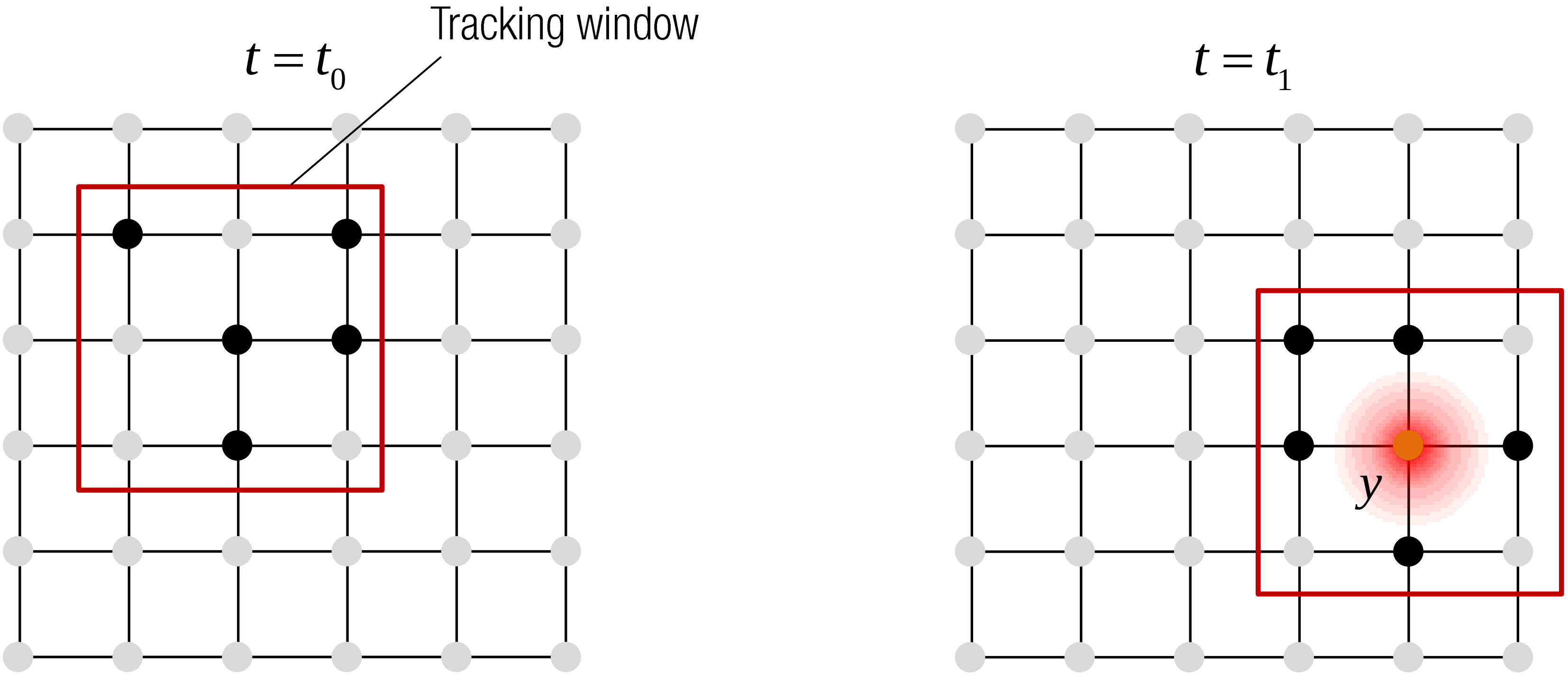


$$q = \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}$$



Discretized color histogram

Nonrigid Tracking For Color Image



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

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


87
59
18
75