

Handling Scale Changes



Scale Space

- What is the size of a feature? ..Make it independent of the pixel size
- **Scale-space representation** of image $c(x, y)$: function $L(x, y, t)$ representing different smoothed versions of c
 - e.g. $L(x, y, t) = G(x, y) * c(x, y)$ where G is Gaussian with $\sigma^2 = t$
- Gaussian smoothing has many nice properties:
 - Non-enhancement of local extrema
 - Causality: if $t_2 > t_1$, $L(x, y, t_2)$ is fully determined by $L(x, y, t_1)$
 - Rotationally symmetric, shift-invariant

$t=64$



$t=16$



$t=4$

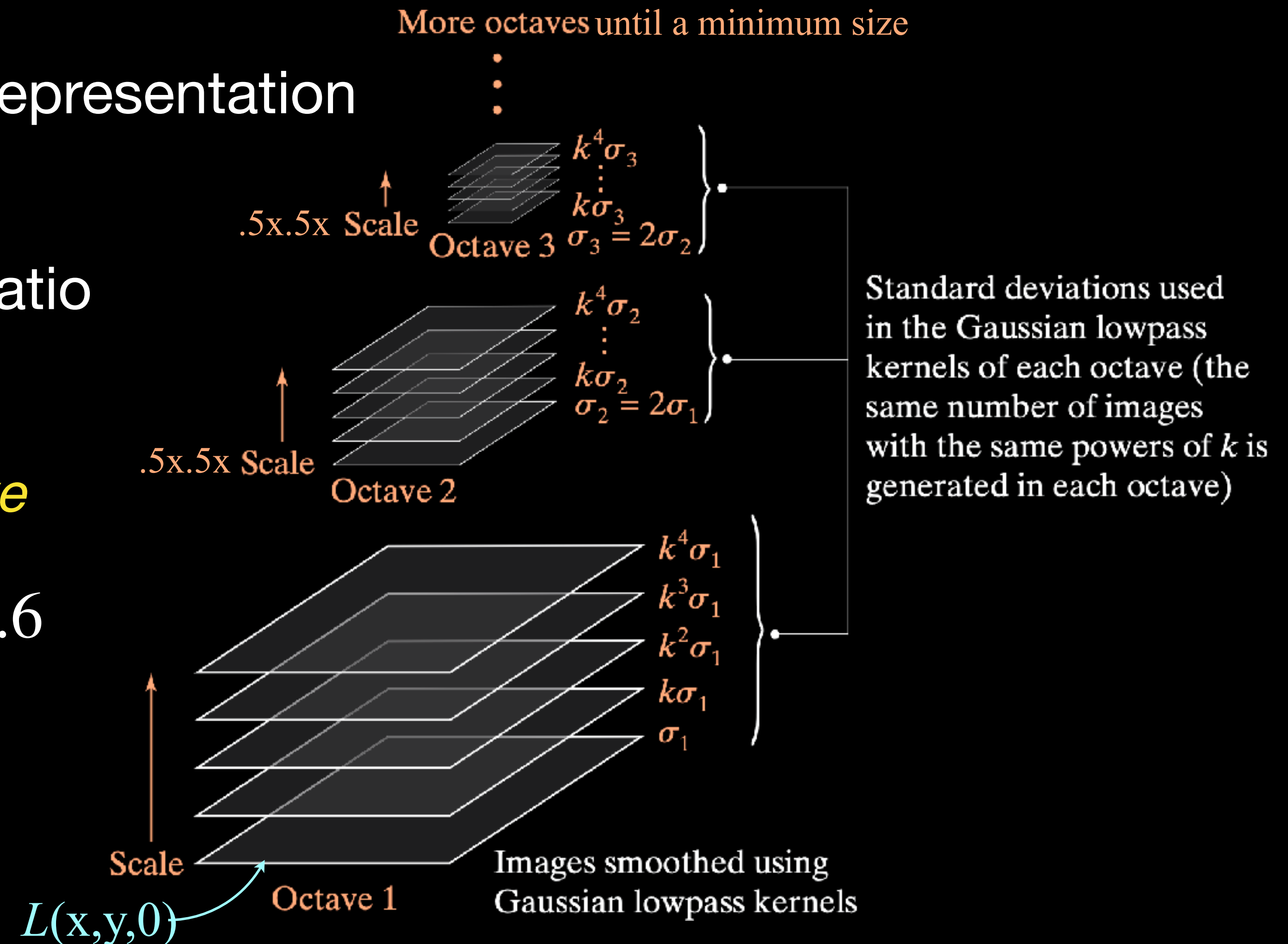


$t=1$



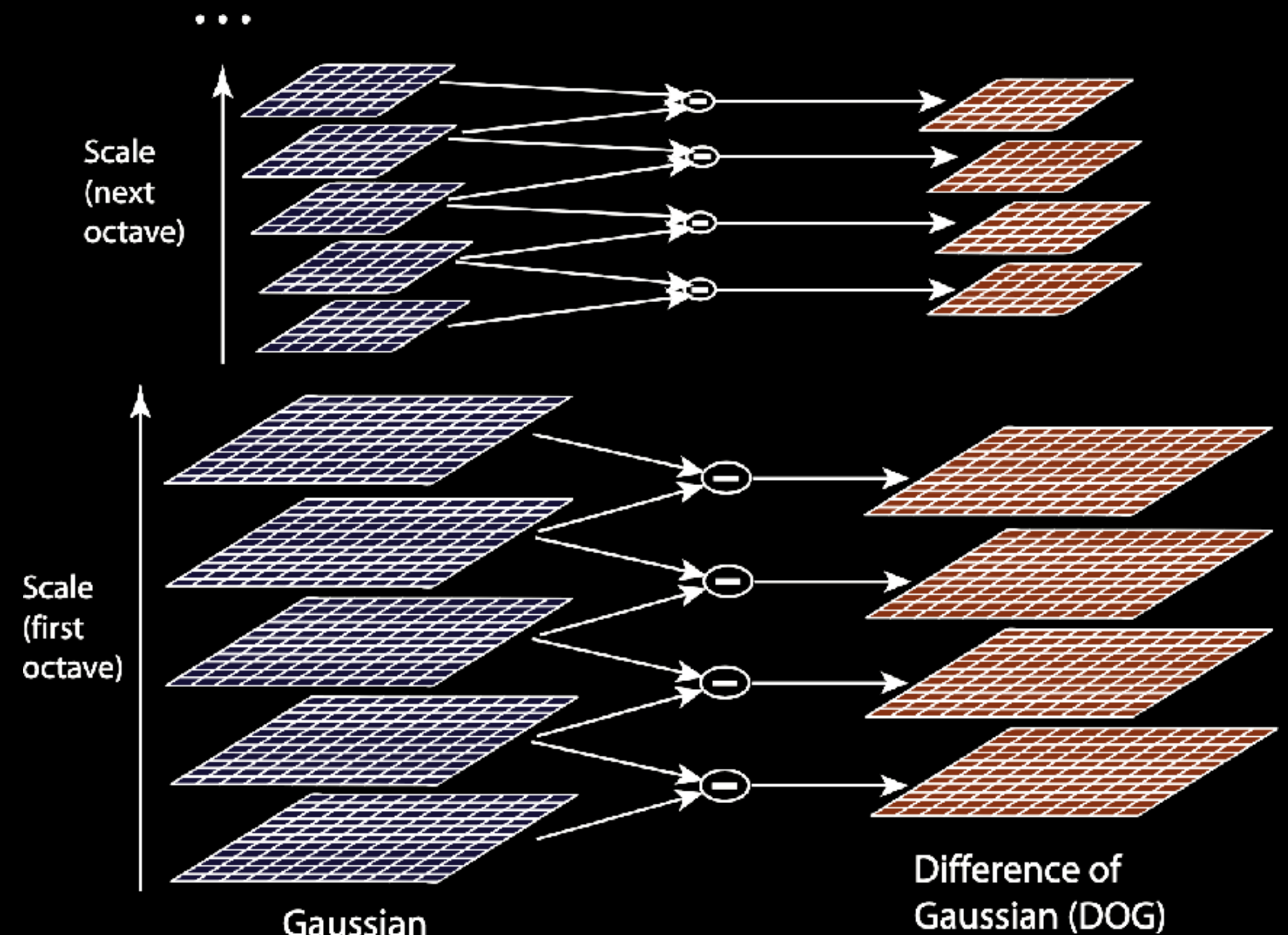
Scale Space Construction

- Gaussian pyramid $\approx$ scale-space representation with scales $\sigma, 2\sigma, 4\sigma, 8\sigma, \dots$
- For better sensitivity, use smaller ratio $\sigma, k\sigma, k^2\sigma, k^3\sigma, \dots$
- $k = 2^{1/s} \rightarrow s + 3$ images per *octave*
- A typical choice: $s = 3$ or $4, \sigma = 1.6$
 - $k \approx 1.19$ or 1.26



Difference between adjacent scales
= difference-of-Gaussian (DoG) image
 $\approx \sigma^2 \cdot \text{Laplacian-of-Gaussian (LoG) image}$

Bright/dark blobs correspond to
local minima/maxima of DoG in x, y, σ space



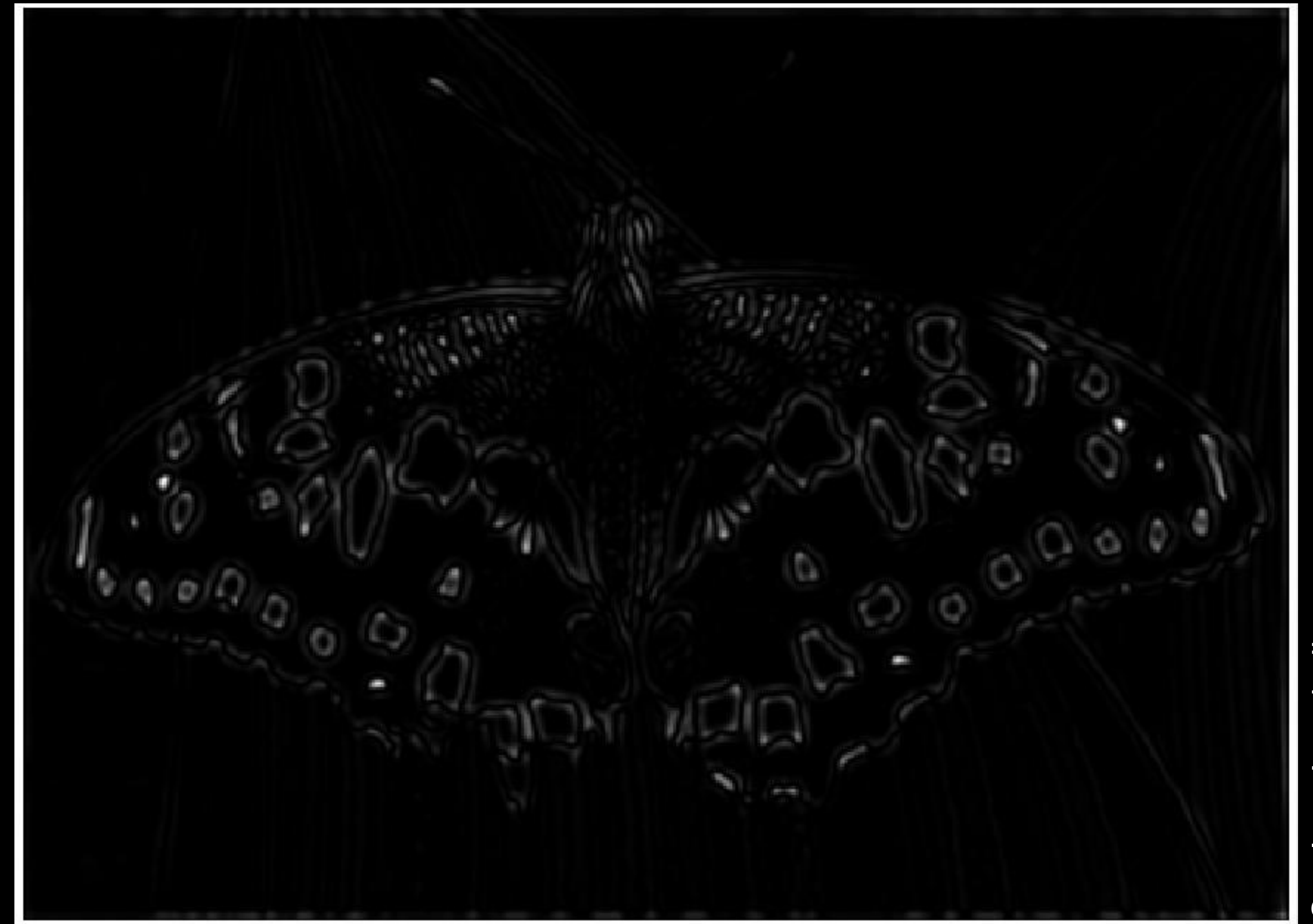
$s+3$ Image layers

Typically use s DoG layers
+ 2 extra for interpolation

Original image



IDoGI²



$\sigma = 2.0$

Original image



IDoGI²

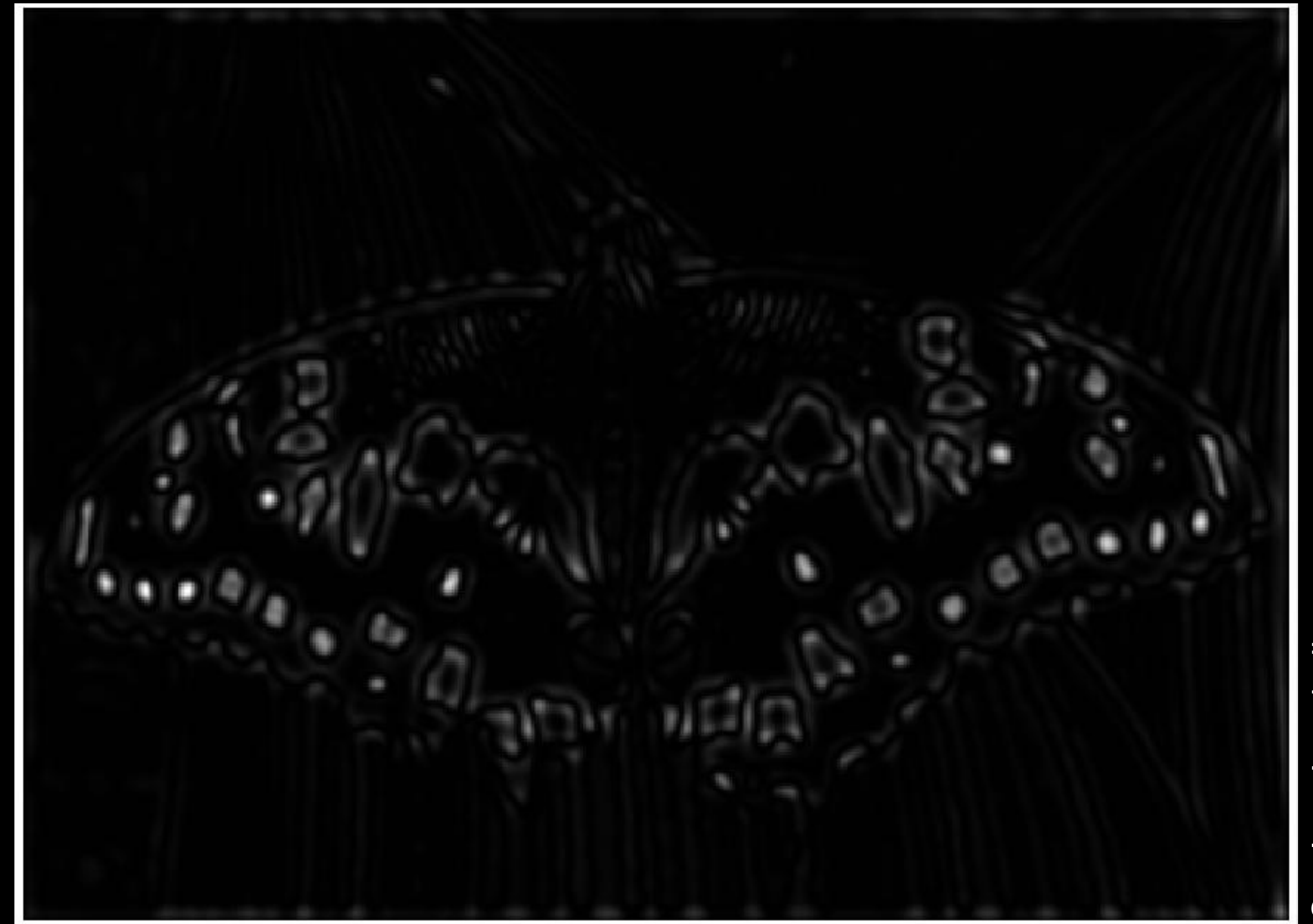


$$\sigma = 2.5$$

Original image



IDoGI²

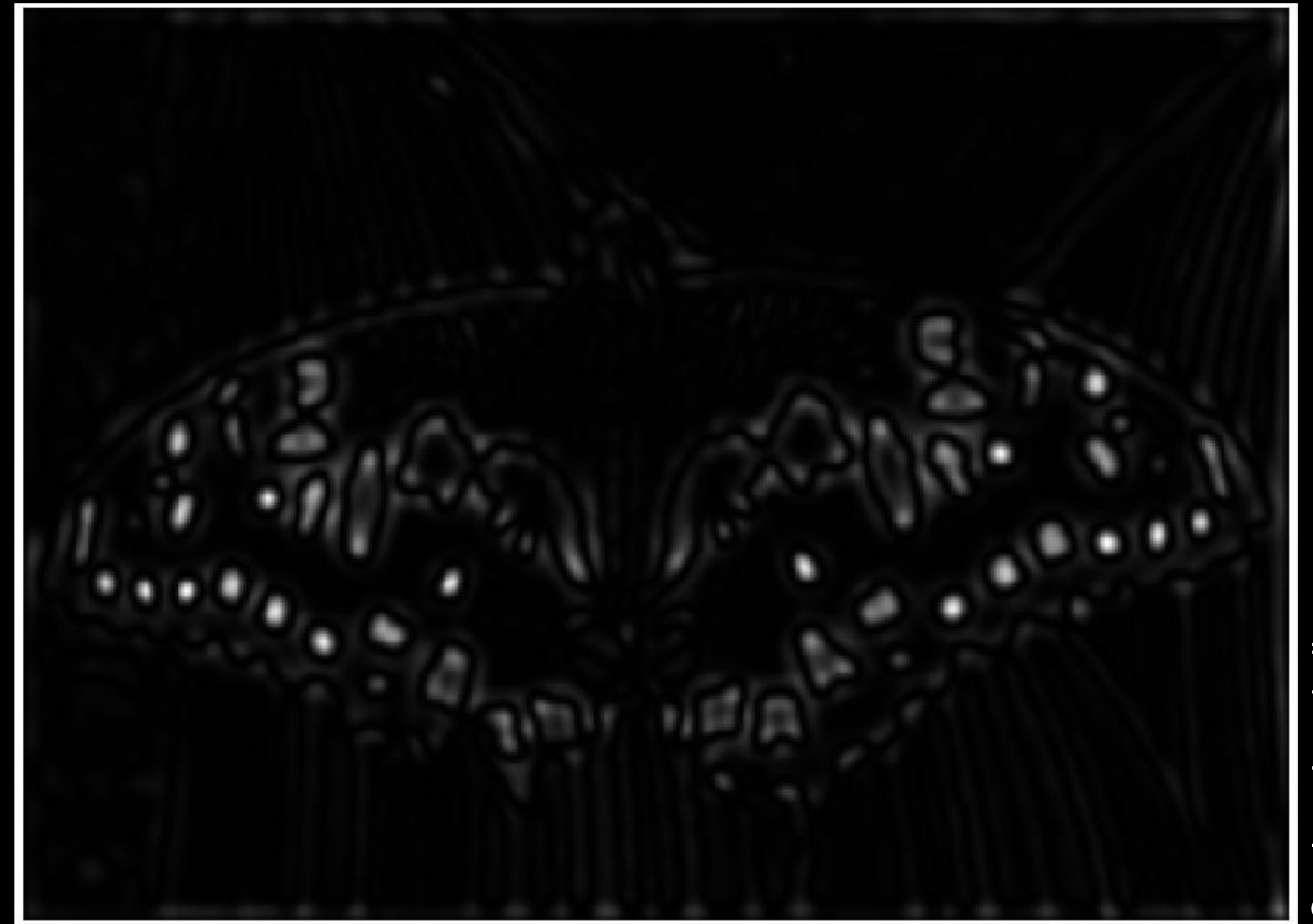


$$\sigma = 3.1$$

Original image



IDoGI²



$$\sigma = 3.9$$

Original image



IDoGI²

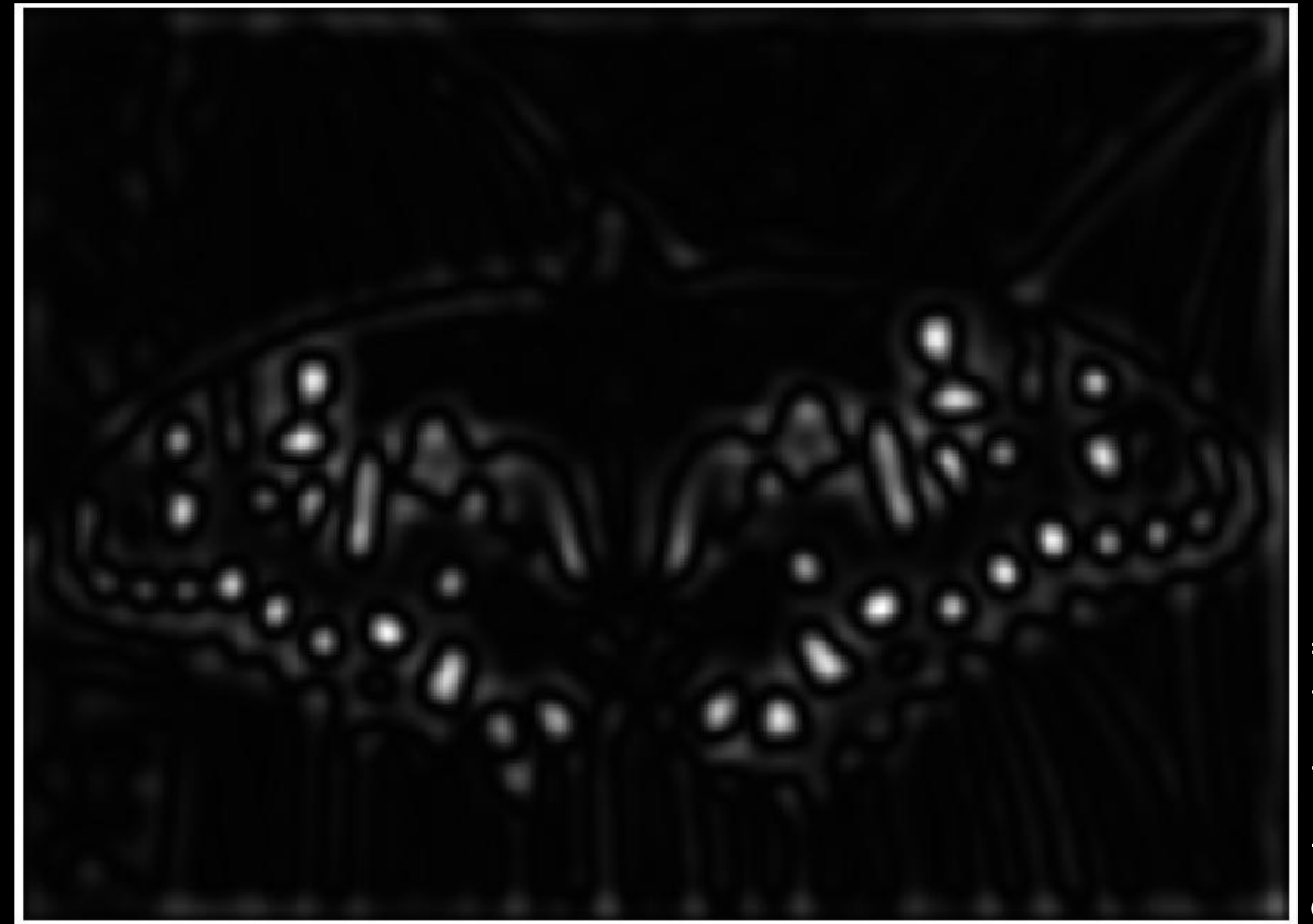


$$\sigma = 4.9$$

Original image



IDoGI²

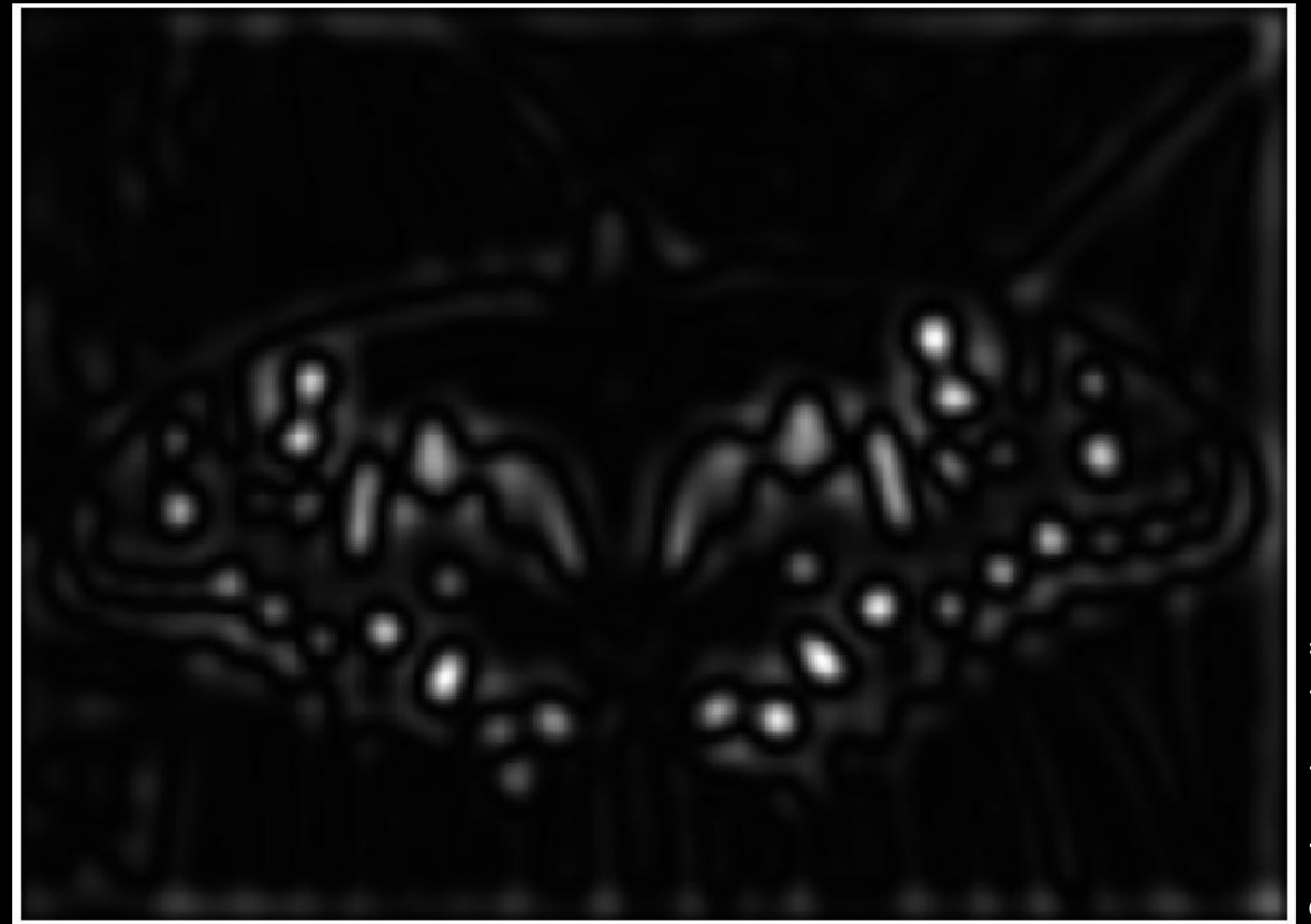


$$\sigma = 6.1$$

Original image



IDoGI²



$$\sigma = 7.7$$

Original image



IDoGI²



$$\sigma = 9.6$$

Original image



IDoGI²



$\sigma = 12.0$

Original image



IDoGI²



$\sigma = 15.0$

Original image



Detected blobs



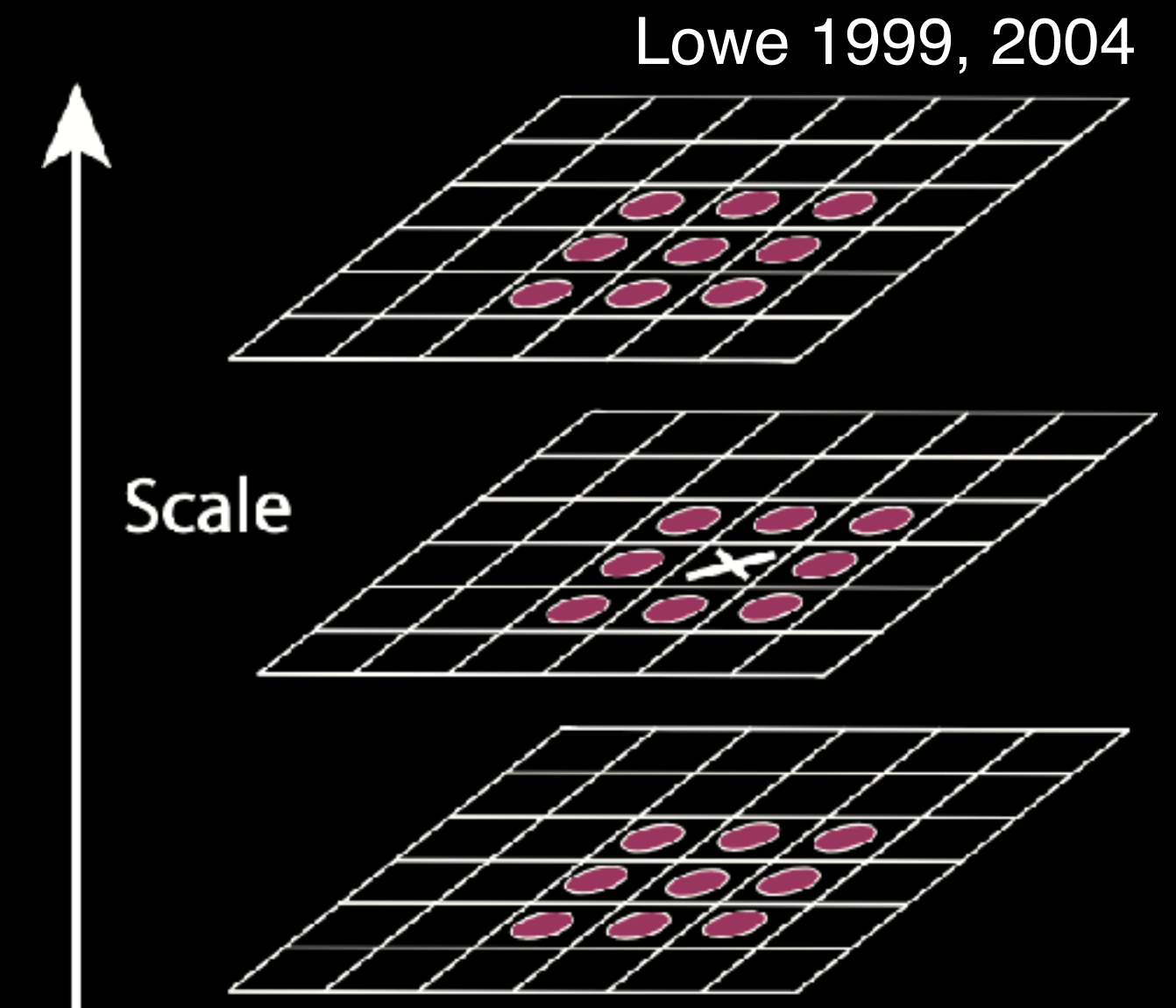
SIFT: Scale-Invariant Feature Transform

- Find local extrema of DoG, $D(x, y, \sigma)$
- Reject low-contrast points, e.g., $|D| < 0.03$ (for $I_{\max} = 1$)
- Reject edges: Form structure tensor

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

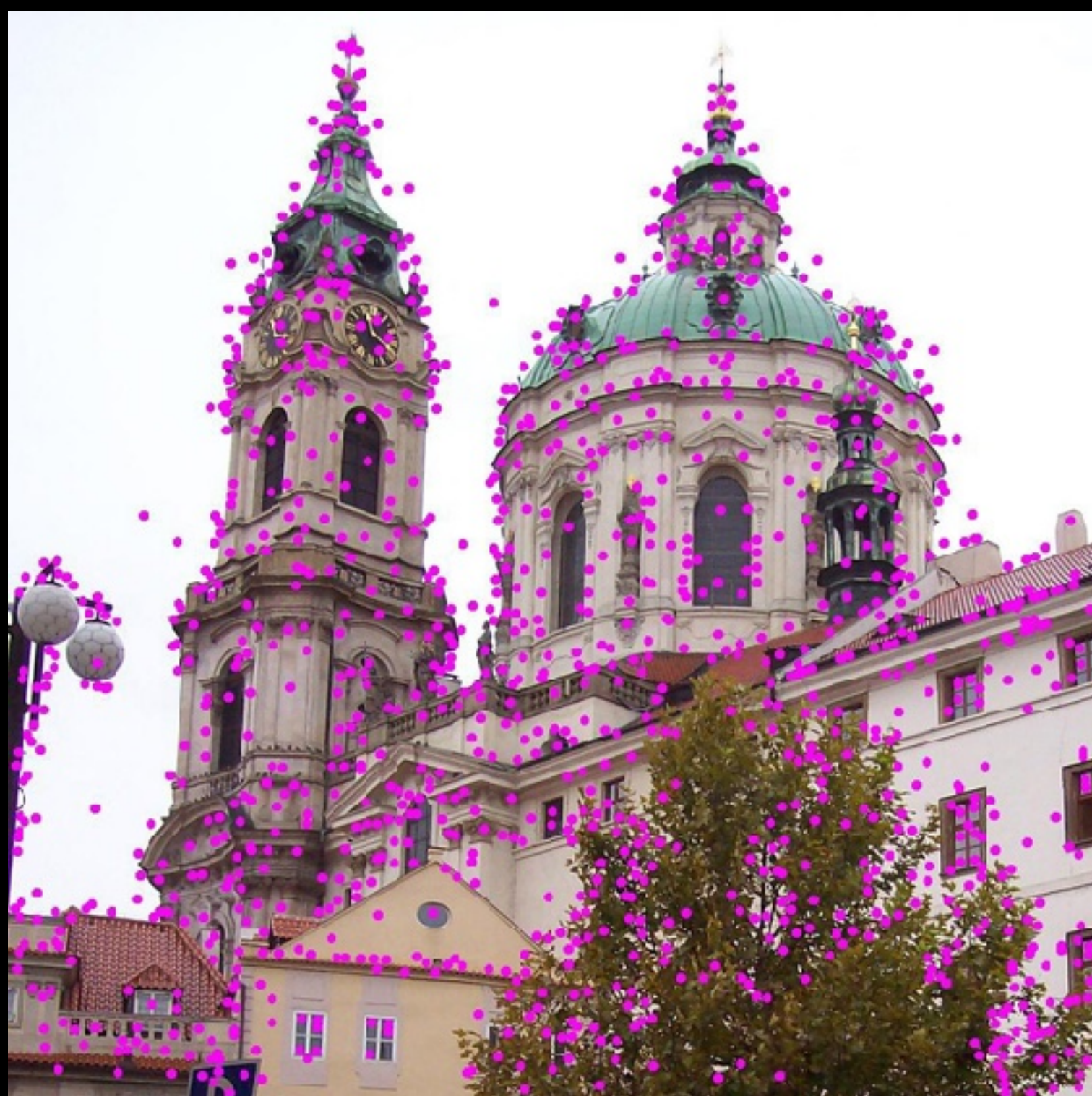
- Like Harris-Stephens, we want minimum Eigenvalue to be large

$$\lambda_{\min} > \frac{1}{r} \lambda_{\max} \iff \frac{\text{tr}(\mathbf{H})^2}{\det(\mathbf{H})} > \frac{(r+1)^2}{r}$$

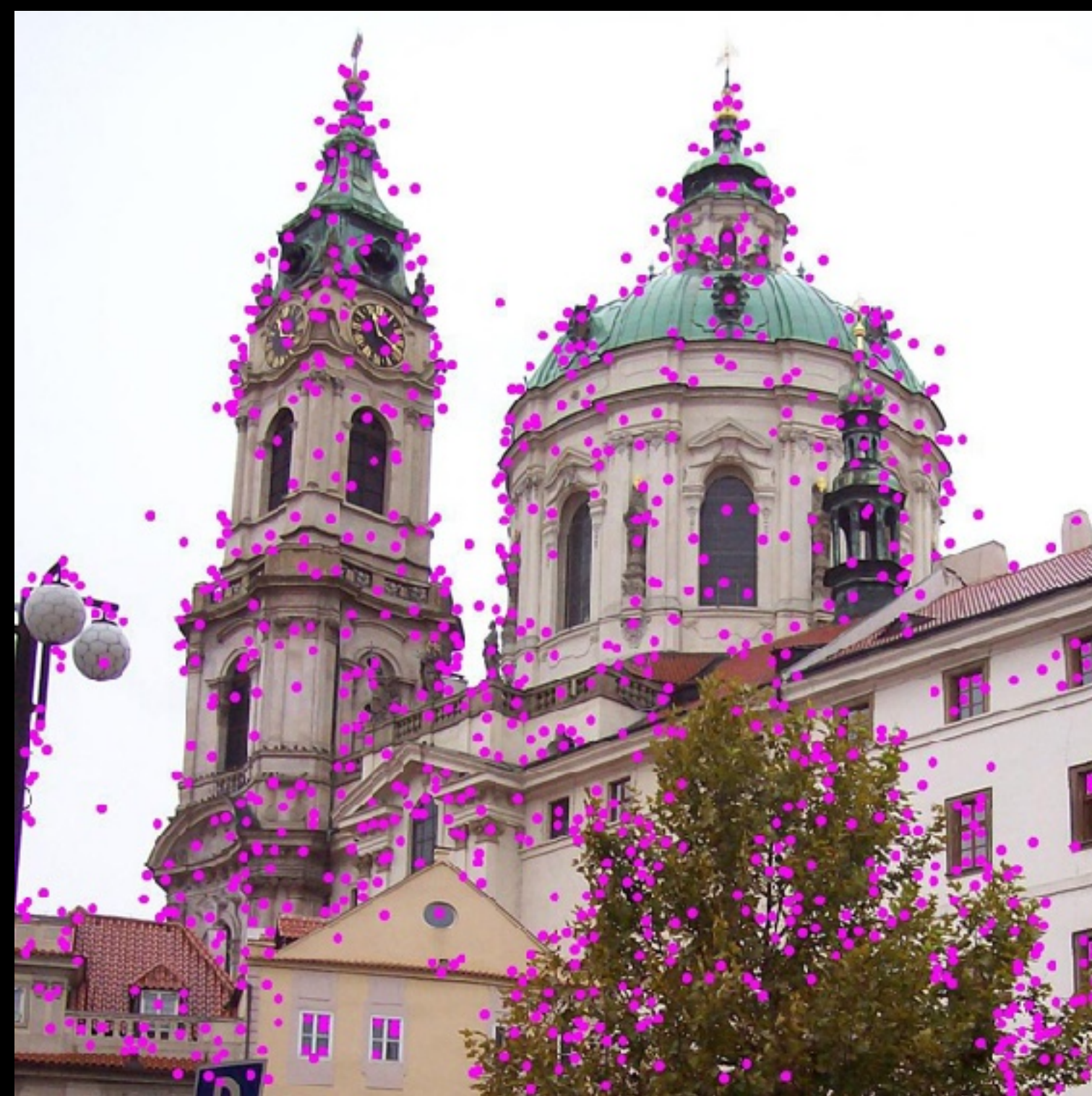




All scale-space extrema



Discard low-contrast points



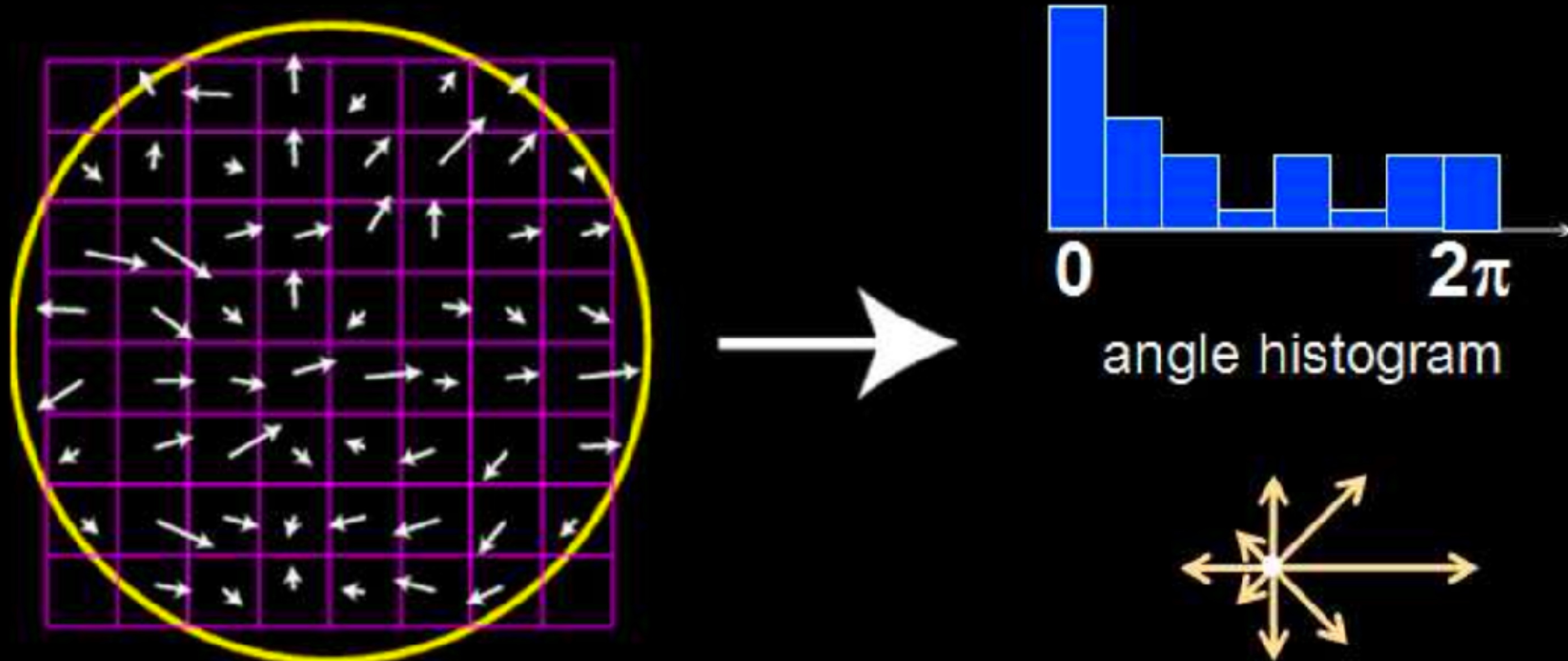
Discard points on edges



Making the Feature Rotationally Invariant

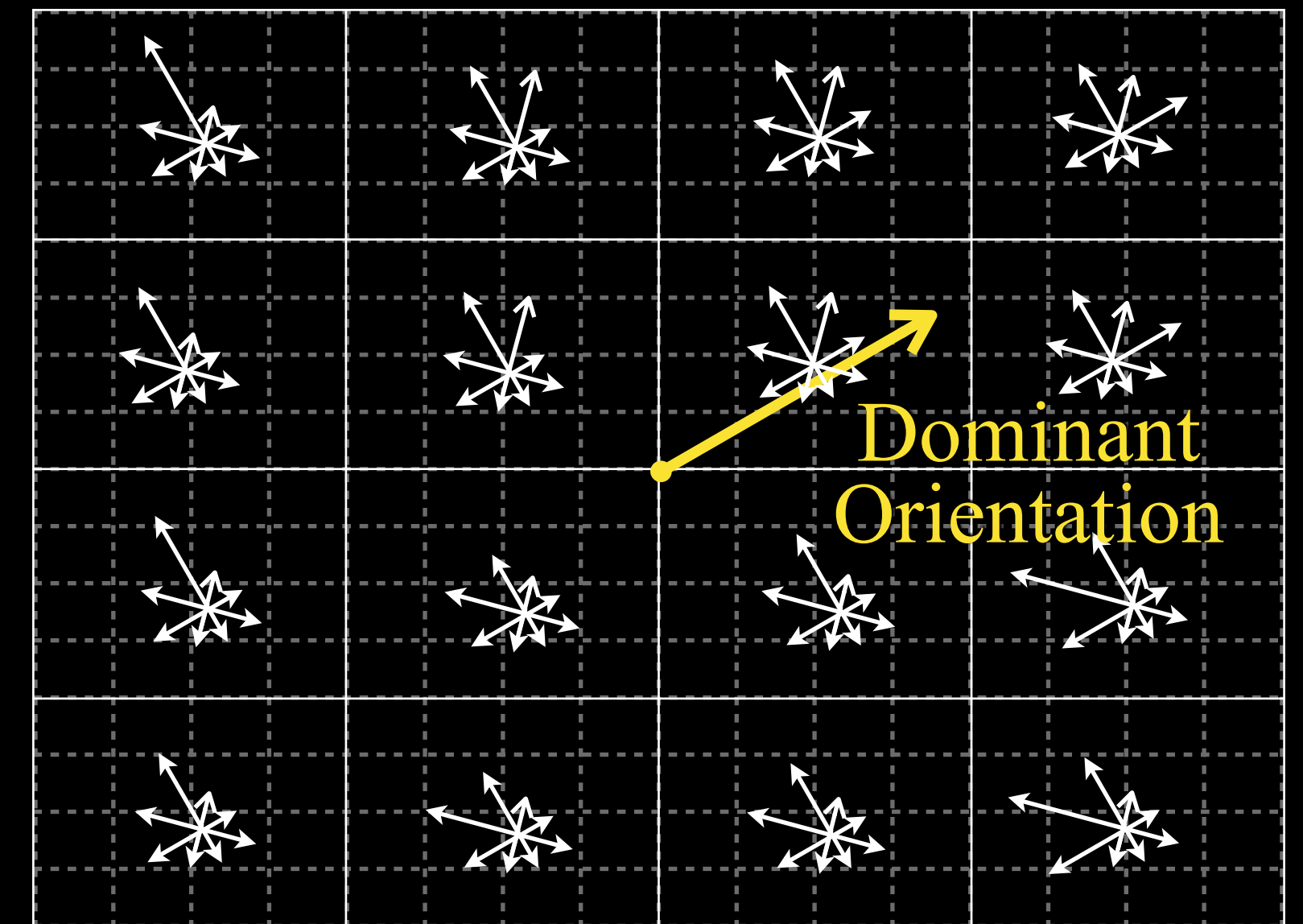
- The descriptor should be relative to the feature's orientation, not the image's general rotation
- Choose the *dominant orientation* of each keypoint
 - But the gradient direction at (x, y) is not robust if its magnitude is small
 - Instead use the most “common” gradient direction in neighbourhood of (x, y)
- Make the feature descriptor relative to this orientation
- To preserve the scale invariance, do all further computations on the smoothed image at the appropriate scale: $L(x, y, \sigma)$, not on the original colors

Choosing Dominant Orientation

- Most common gradient directions in neighbourhood of (x, y) = peaks in local orientation histogram
 - Weighted by gradient magnitude and a Gaussian falloff (e.g., $\sigma = 1.5 \times \text{scale of key point}$)
 - Use fractional increments in the Histogram, as with Hough
- 
- Image gradients
- angle histogram
- 0 2π
- Consider multiple peaks if $>80\%$ height of highest peak separate keypoints

The SIFT Feature Descriptor

- To match keypoints, we need a *feature descriptor* for SIFT features
1. Build histogram of local gradient directions in the frame of the dominant gradient
 - 4×4 grid centered at keypoint, using detected orientation & scale
 - 8 bins for directions
 2. Normalize to unit sum
 - Then clip to [0,0.2] and normalize again (similar to CLAHE)
- Results in 4×4×8 numbers → 128-dimensional feature vector





SIFT Feature Computation

- Find Scale-Space Peaks: Local extrema in DoGs
 - It's a feature at that scale
- Discard unstable, low-contrast, or edge-prone points
 - Assign precise sub-pixel coordinates for peaks for accuracy
- Compute local gradients to establish dominant orientations
 - Allowing rotation invariance
- Feature description: gradient orientation-histograms for patch around keypoint
 - 128-dimensional feature

Matching Keypoints

Distance between descriptors

- Compare with all key points and find the nearest
- Usually leads to many false matches
 - To avoid ambiguous matches, keep only if $\text{dist}(\text{nearest}) < 0.8 \text{ dist}(\text{2nd-nearest})$
 - Or, exploit spatial coherence in matching



Matching Keypoints

Distance between descriptors

- Compare with all key points and find the nearest
- Usually leads to many false matches
 - To avoid ambiguous matches, keep only if $\text{dist}(\text{nearest}) < 0.8 \text{ dist}(\text{2nd-nearest})$
 - Or, exploit spatial coherence in matching
 - If feature A matches feature A', features around A are also likely to match features around A'

