**Course Syllabus**

1. **Color**
2. **Camera models, camera calibration**
3. **Advanced image pre-processing**
   - Line detection
   - Corner detection
   - Maximally stable extremal regions
4. **Mathematical Morphology**
   - binary
   - gray-scale
   - skeletonization
   - granulometry
   - morphological segmentation
   - Scale in image processing
5. **Wavelet theory in image processing**
6. **Image Compression**
7. **Texture**
8. **Image Registration**
   - rigid
   - non-rigid
   - RANSAC

# References

- Book: Chapter 5, Image Processing, Analysis, and Machine Vision, Sonka et al, latest edition (you may collect a copy of the relevant chapters from my office)
- Papers:
  - Harris and Stephens, 4th Alvey Vision Conference, 147-151, 1988.
  - Matas et al, Image Vision Geometry, 22:761-767, 2004

# Topics

- Line detection
- Interest points
  - Corner Detection
    - Moravec detector
    - Facet model
    - Harris corner detection
  - Maximally stable extremal regions

# Line detection

- Useful in remote sensing, document processing etc.
- Edges:
  - boundaries between regions with relatively distinct gray-levels
  - the most common type of discontinuity in an image
- Lines:
  - instances of thin lines in an image occur frequently enough
  - it is useful to have a separate mechanism for detecting them.

# Line detection: How?

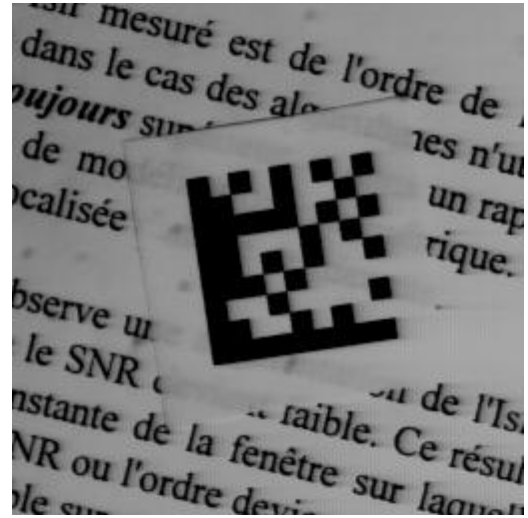
- Possible approaches: Hough transform (more global analysis and may not be considered as a local pre-processing technique)
- Convolve with line detection kernels

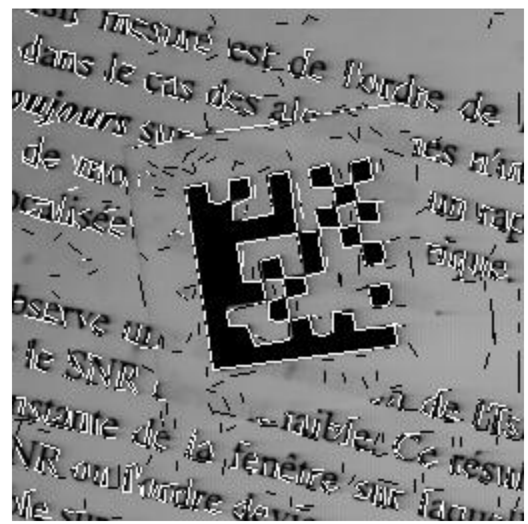
$$L_h = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

$$L_v = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

$$L_o = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

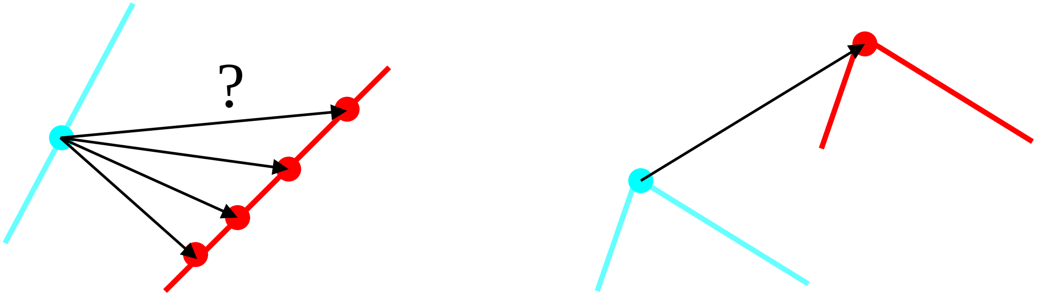
How to detection lines along other directions?



# Lines and corner for correspondence

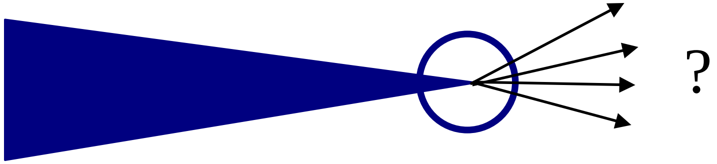
- **Interest points** for solving correspondence problems in time series data.
- Corners are better than lines in solving the above problem due to the **aperture problem**
  - Consider that we want to solve point matching in two images




  - A vertex or corner provides better correspondence

# Corners

- **Challenges**
  - Gradient computation is less reliable near a corner due to ambiguity of edge orientation

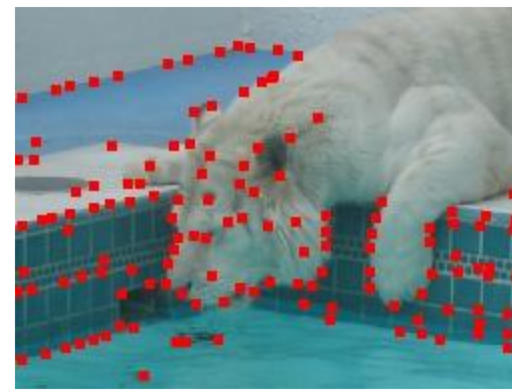



  - Corner detector are usually not very robust.
  - This deficiency is overcome either by manual intervention or large redundancies.
  - The later approach leads to many more corners than needed to estimate transforms between two images.

# Corner detection

- **Moravec detector:** detects corners as the pixels with locally maximal contrast

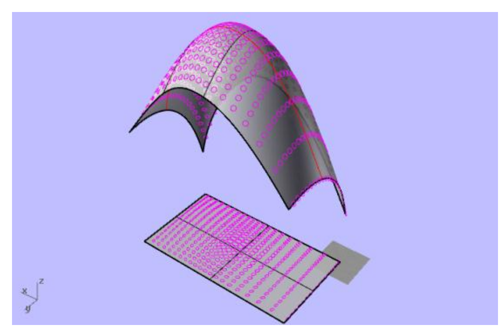
$$MO(i,j) = \frac{1}{8}\sum_{\Delta i = -1}^{1}\sum_{\Delta j = -1}^{1} |f(i+\Delta i, j+\Delta j) - f(i,j)|$$



- **Differential approaches:**
  - Beaudet's approach: Corners are measured as the determinant of the Hessian.
  - Note that the determinant of a Hesian is equivalent to the product of the min & max Gaussian curvatures

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

Corner $measure\ DET(H) = \frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2$



# *Continued ...*

- Using a bi-cubic facet model

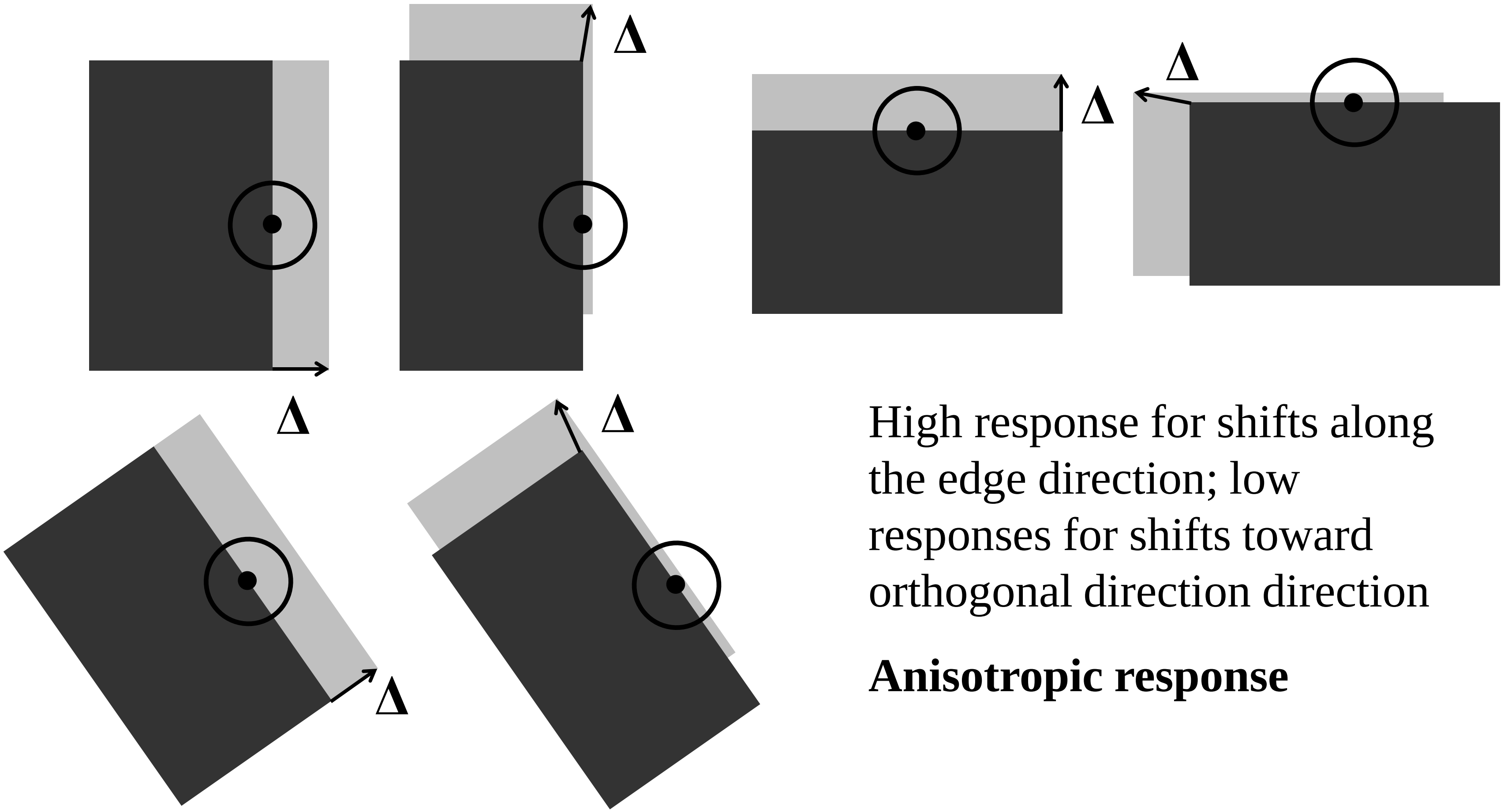
$$f(i,j) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$$

$$KR = \frac{2\left(c_4c_3^2 + c_6c_2^2 - c_5c_2c_3\right)}{c_2^2 + c_3^2}$$

$$ZH = \frac{2\left(c_4c_3^2 + c_6c_2^2 - c_5c_2c_3\right)}{\left(c_2^2 + c_3^2\right)^{\frac{3}{2}}}$$

# Harris corner detector

- Key idea: Measure changes over a neighborhood due to a shift and then analyze its dependency on shift orientation
- Orientation dependency of the response for lines

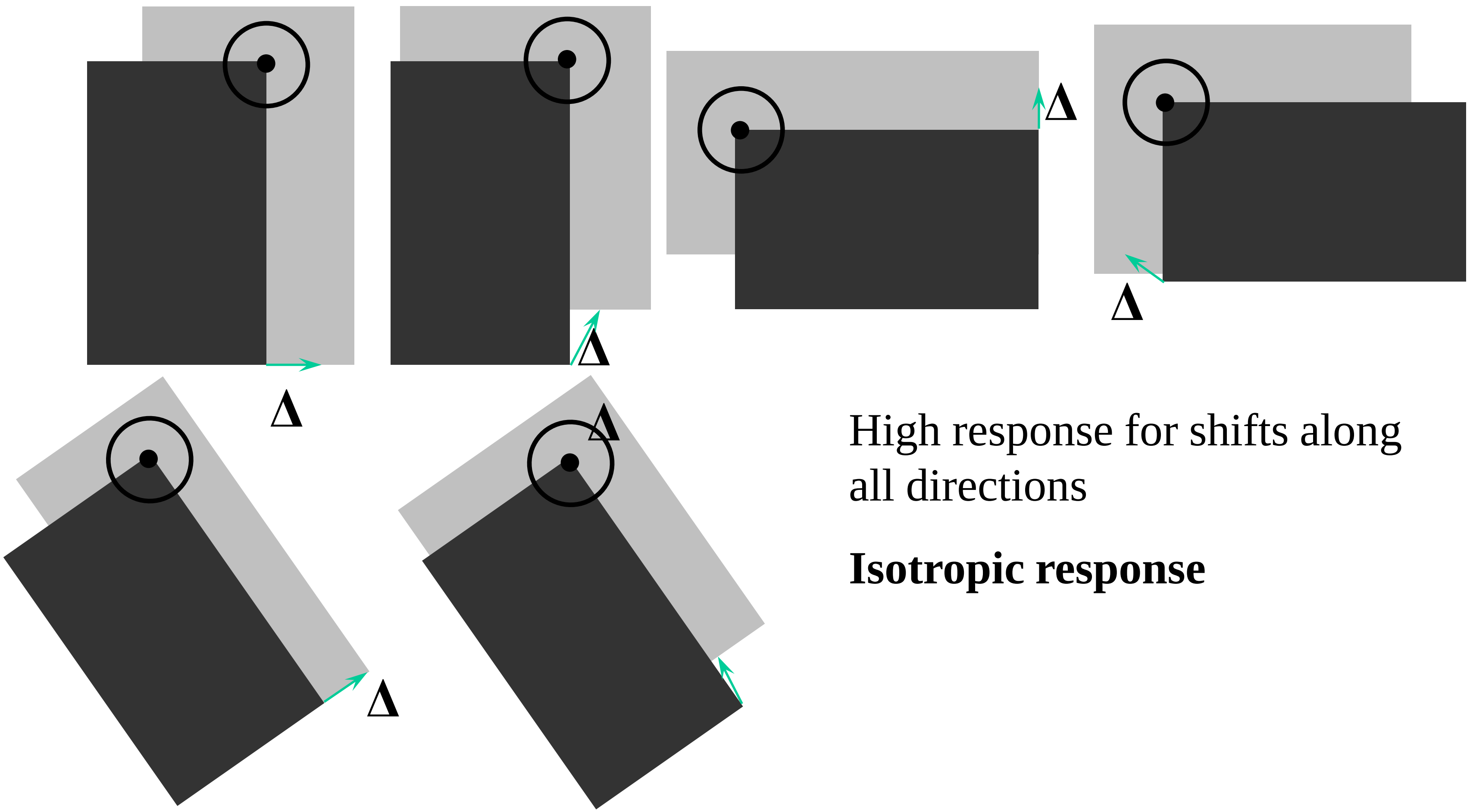



High response for shifts along the edge direction; low responses for shifts toward orthogonal direction direction

**Anisotropic response**

# Key idea: *continued* ...

- Orientation dependence of the shift response for corners




High response for shifts along all directions

**Isotropic response**

# Harris corner: mathematical formulation

- An image patch or neighborhood W is shifted by a shift vector $\mathbf{\Delta} = [\Delta x, \Delta y]^T$
- A corner does not have the aperture problem and therefore should show high shift response for all orientation of $\mathbf{\Delta}$.
- The square intensity difference between the original and the shifted image over the neighborhood W is

$$S_W(\mathbf{\Delta}) = \sum_{(x_i,y_i)\in W} \left(f(x_i, y_i) - f(x_i + \Delta x, y_i + \Delta y)\right)^2$$

- Apply first-order Taylor expansion

$$f(x_i + \Delta x, y_i + \Delta y) \approx f(x_i, y_i) + \begin{bmatrix} \frac{\partial f(x_i, y_i)}{\partial x} & \frac{\partial f(x_i, y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

# *Continued ...*

$$S(x,y,\mathbf{\Delta}) = \sum_{(x_i,y_i)\in W} \left( f(x_i,y_i) - f(x_i,y_i) - \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} & \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2$$

$$= \sum_{(x_i,y_i)\in W} \left( \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} & \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2$$

$$= \sum_{(x_i,y_i)\in W} \left( \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} & \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^{\mathrm{T}} \left( \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} & \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)$$

$$= \sum_{(x_i,y_i)\in W} \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} \\ \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} & \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

$$= \sum_{(x_i,y_i)\in W} \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \left( \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} \\ \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} & \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \right) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

# *Continued ...*

$$= \sum_{(x_i,y_i)\in W} [\mathit{\Delta} x \quad \mathit{\Delta} y] \left( \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} \\ \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} & \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \right) \begin{bmatrix} \mathit{\Delta} x \\ \mathit{\Delta} y \end{bmatrix}$$

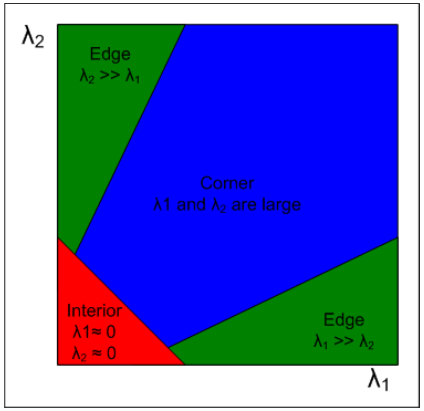
$$= [\mathit{\Delta} x \quad \mathit{\Delta} y] \left( \sum_{(x_i,y_i)\in W} \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} \\ \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f(x_i,y_i)}{\partial x} & \frac{\partial f(x_i,y_i)}{\partial y} \end{bmatrix} \right) \begin{bmatrix} \mathit{\Delta} x \\ \mathit{\Delta} y \end{bmatrix}$$

$$= [\mathit{\Delta} x \quad \mathit{\Delta} y] \begin{bmatrix} \sum_{(x_i,y_i)\in W} \left(\frac{\partial f(x_i,y_i)}{\partial x}\right)^2 & \sum_{(x_i,y_i)\in W} \frac{\partial f(x_i,y_i)}{\partial x} \frac{\partial f(x_i,y_i)}{\partial y} \\ \sum_{(x_i,y_i)\in W} \frac{\partial f(x_i,y_i)}{\partial x} \frac{\partial f(x_i,y_i)}{\partial y} & \sum_{(x_i,y_i)\in W} \left(\frac{\partial f(x_i,y_i)}{\partial y}\right)^2 \end{bmatrix} \begin{bmatrix} \mathit{\Delta} x \\ \mathit{\Delta} y \end{bmatrix}$$

$$= \mathbf{\Delta}^{\mathrm{T}} \mathrm{A}_w(x,y) \mathbf{\Delta}$$

# Harris matrix

- The matrix $\mathbf{A}_W$ is called the Harris matrix and its symmetric and positive semi-definite. Eigen-value decomposition of of $\mathbf{A}_W$ gives eigenvectors and eigenvalues ($\lambda_1$, $\lambda_2$) of the response matrix.
- Three distinct situations:
  - Both $\lambda_1$ and $\lambda_2$ are small $\Rightarrow$ no edge or corner; a flat region
  - $\lambda_i$ is large but $\lambda_{j \neq i}$ is small $\Rightarrow$ existence of an edge; no corner
  - Both $\lambda_1$ and $\lambda_2$ are large $\Rightarrow$ existence of a corner



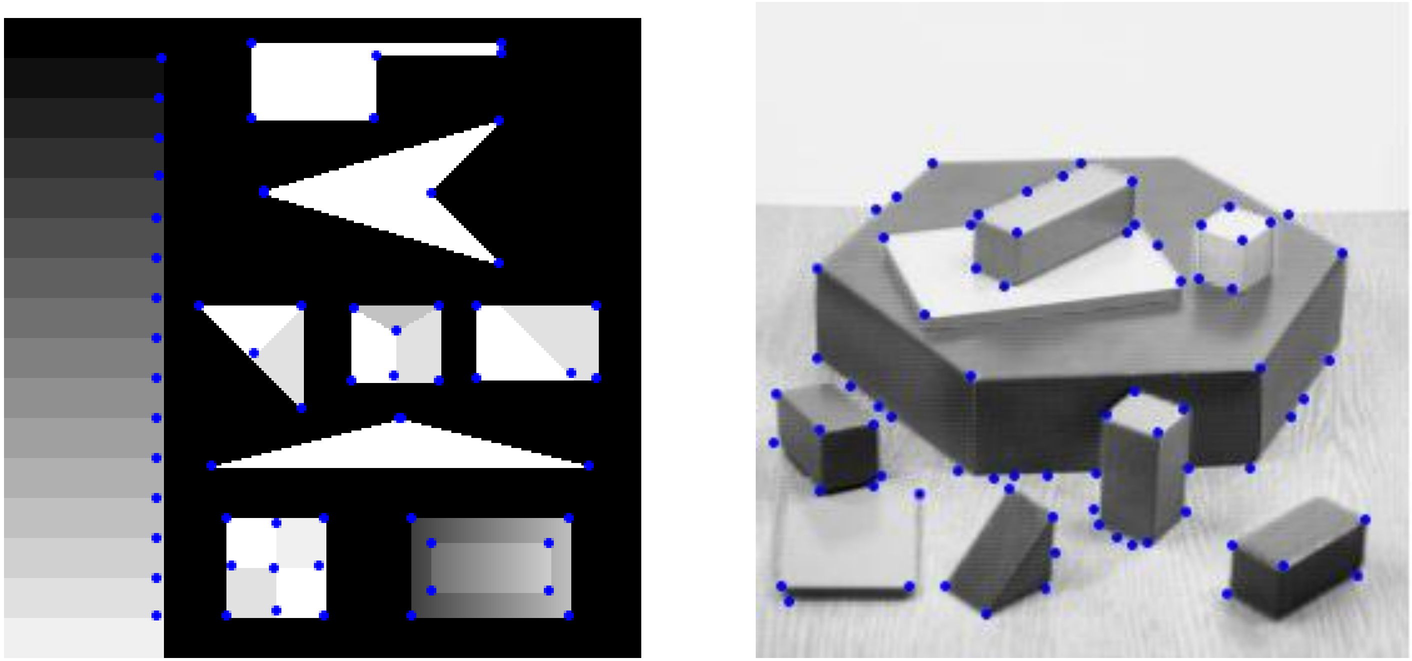

- Avoid eigenvalue decomposition and compute a single response measure
  - Harris response function

    $$R(A) = \det(A) - \kappa * trace^2(A)$$
  - A value of κ between 0.04 and 0.15 has be used in literature.

# Algorithm: Harris corner detection

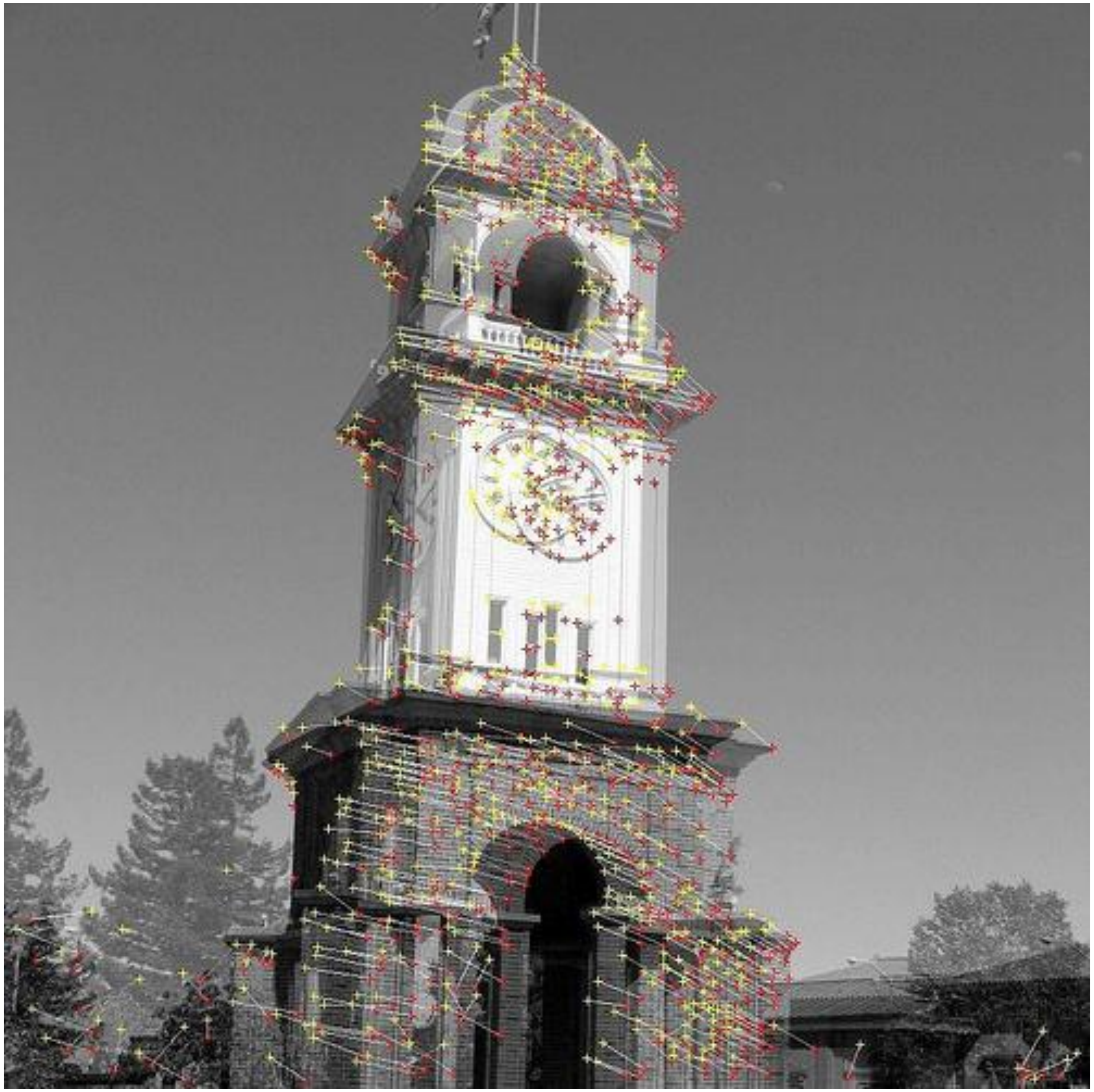
1. Filter the image with a Gaussian
2. Estimate intensity gradient in two coordinate directions
3. For each pixel $c$ and a neighborhood window $W$
   a. Calculate the local Harris matrix $A$
   b. Compute the response function $R(A)$
4. Choose the best candidates for corners by selecting thresholds on the response function $R(A)$
5. Apply non-maximal suppression

# Examples



# Examples

# Examples



# Maximally Stable Extremal Regions (MSER)

- Key idea: Consider a movie generated by thresholding a gray level image (intensity values [0,1,…,255]) at all possible thresholds starting from 0 and ending at 255. building

# Maximally Stable Extremal Regions (MSER)

- Key idea: Consider a movie generated by thresholding a gray level image (intensity values [0,1,…,255]) at all possible thresholds starting from 0 and ending at 255.
  - Initially it's an empty image and then some dots (local minima) appear and starts growing
  - New dots appear and starts growing and so on
  - From time to time two disconnected regions get merged
  - Finally, all regions get merged into a single component.
  - BUT, the important observation here is that – starting from a tiny seed area (one or a few pixels), a region continues growing till it fills the object containing the initial seed area and then remains (almost) unchanged for quite sometime in the threshold movie until it get merged with the bigger (generally, parent) object to which it belongs
  - MSER intends to capture these stable regions

# Maximally Stable Extremal Regions (MSER)

- Key idea: Consider a movie generated by thresholding a gray level image (intensity values [0,1,…,255]) at all possible thresholds starting from 0 and ending at 255.
  - BUT, the important observation here is that
  - Starting from a tiny seed area (one or a few pixels), a region continues growing till it fills the object containing the initial seed area
  - Then remains (almost) unchanged for quite sometime in the threshold movie until it get merged with the bigger (generally, parent) object to which it belongs

  - **MSER intends to capture these stable regions**

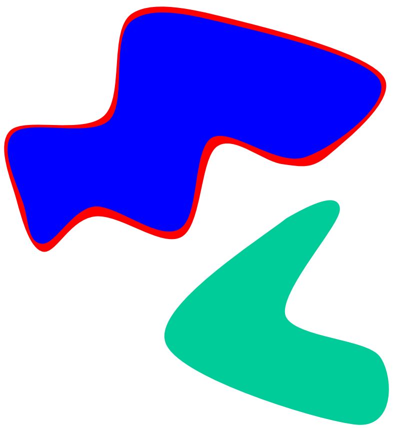
# Maximally Stable Extremal Regions (MSER)

- Key idea: Consider a movie generated by thresholding a gray level image (intensity values [0,1,…,255]) at all possible thresholds starting from 0 and ending at 255. building

# MSER: mathematical formulation

Image $I: D \subset \mathbb{Z}^2 \rightarrow \mathcal{F}$

1) Assumption: $\mathcal{F}$ is fully ordered
   i.e., $\exists$ a reflexive, anti-symmetric and transitive relation on $\mathcal{F}$
   Trivially satisfied in a scalar image
2) There exists an adjacency relation $A \subset D \times D$, e.g., 26-adjacency
   Path: a sequence of points $p_1, p_2, \ldots, p_n$ where $p_i A p_{i+1}$ for all $i$
   Connected region: a maximal subset $R$ of $D$ where every two points $p, q \in R$ are connected by a path entirely contained by $R$
   Boundary
   $\partial R = \{p | p \in D - R \text{ and } \exists q \text{ s.t.} \, pAq\}$



# MSER: mathematical formulation

3) Extremal region: $Q \subset D$ is a connected region for some threshold, i.e., $p \in Q$ and $p \in \partial Q$ implies that $I(p) < I(q)$
4) Maximally stable extremal region (MSER)
An extremal region that is most stable on the threshold video

HOW TO FORMULATE:
Consider a nested sequence of extremal regions $Q_0, Q_1, \dots, Q_{\max}$, subscript → threshold
Relative speed at threshold $i$ on the nested chain

$$speed_i = \frac{|Q_{i+\Delta} - Q_{i+\Delta}|}{2\Delta * |Q_i|}$$

$\Delta$ is a parameter to the method

Finally, $Q_{i^*}$ is a MSER is $speed_{i^*}$ produces a local minima on the nested chain $Q_0, Q_1, \dots, Q_{\max}$ along the threshold variable

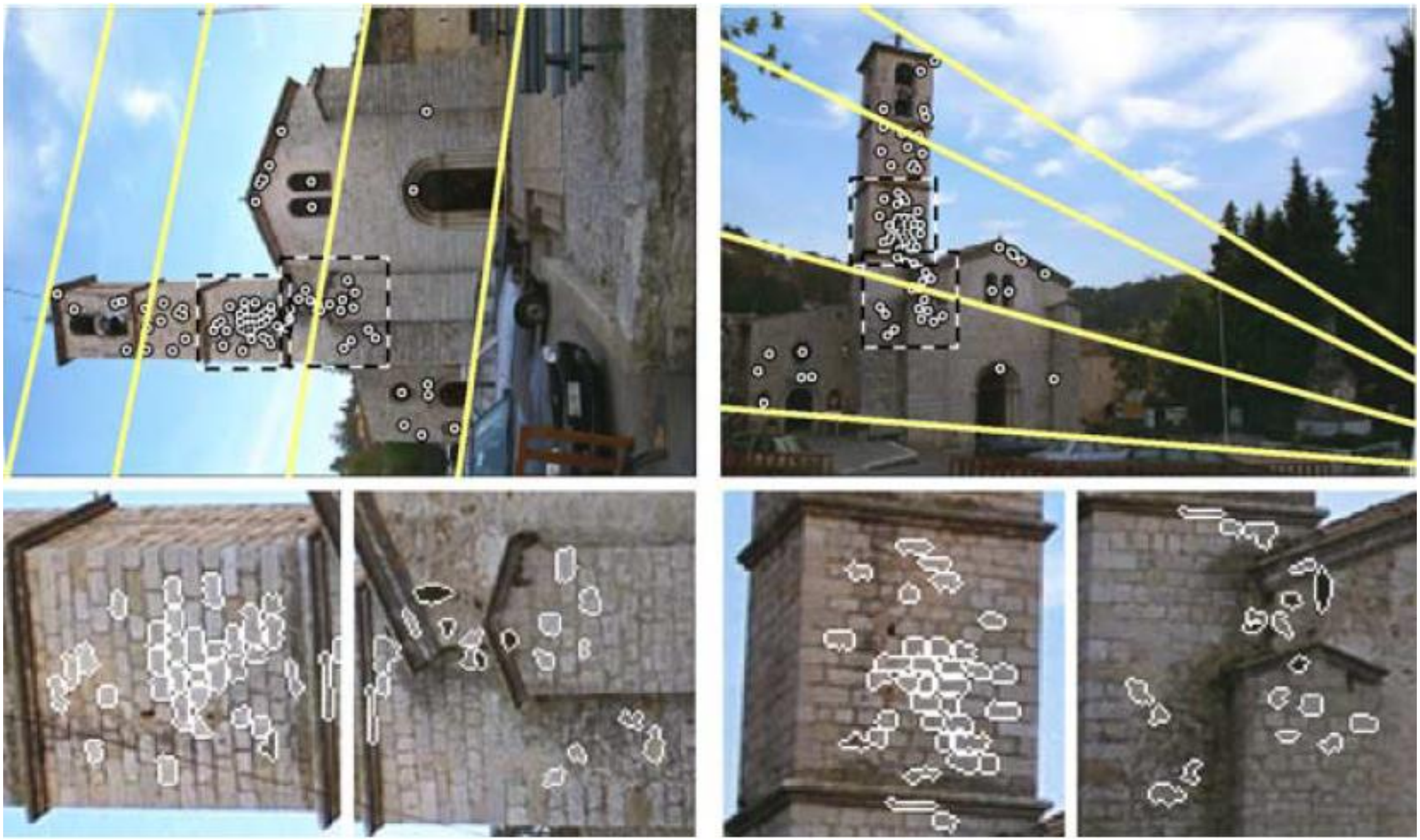
# MSER: properties

- Invariance under monotonic transforms $M : I \to I$ of image intensities
- Invariance under homeomorphic transformations (adjacency preserving) $T: C \to C$ of the image space
- Stability: extremal regions that remain virtually unchanged over a threshold range are selected
- Multi-scale detection: Extremal regions of all scales are detected simultaneously
- The set of all MSERs are enumerated in O(n*log log n) time, where n is the number of image pixels

# Algorithm: MSER enumeration

1. Input: Image *I* and the Δ parameter
2. Output: List of nested extremal regions
3. For all pixels shorted by intensity
    1. Place a pixel in the image as its tern come
    2. Update the connected component structure
    3. Update the area for the effected connected components
4. For all connected components
    1. Detect regions with local minima w.r.t. rate of change of connected component area with threshold; define each such region as a MSER

# Examples



# Examples