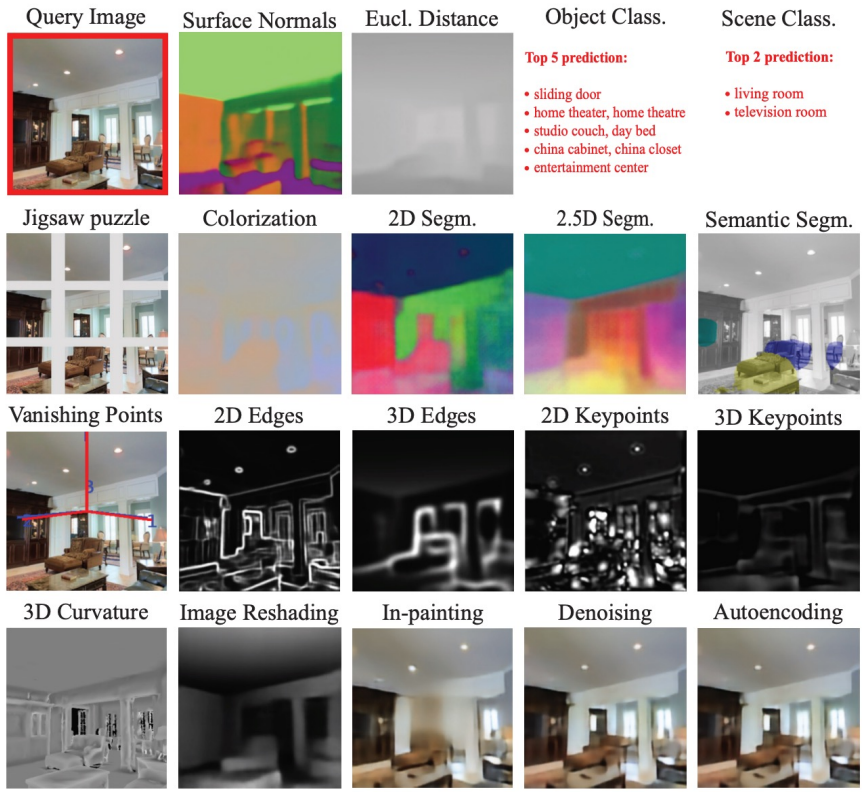


Image Analysis

Fall 2024

Yi-Ting Chen

Image Analysis

Topics:

- Edge detection
- Image Segmentation
- Face Detection/Recognition
- Object Detection/Recognition
- Pose Estimation
- Gesture Recognition
- Handwriting Recognition
- Emotion Recognition
- Scene Analysis
- 3D Scene Reconstruction
- Image Retrieval
- …

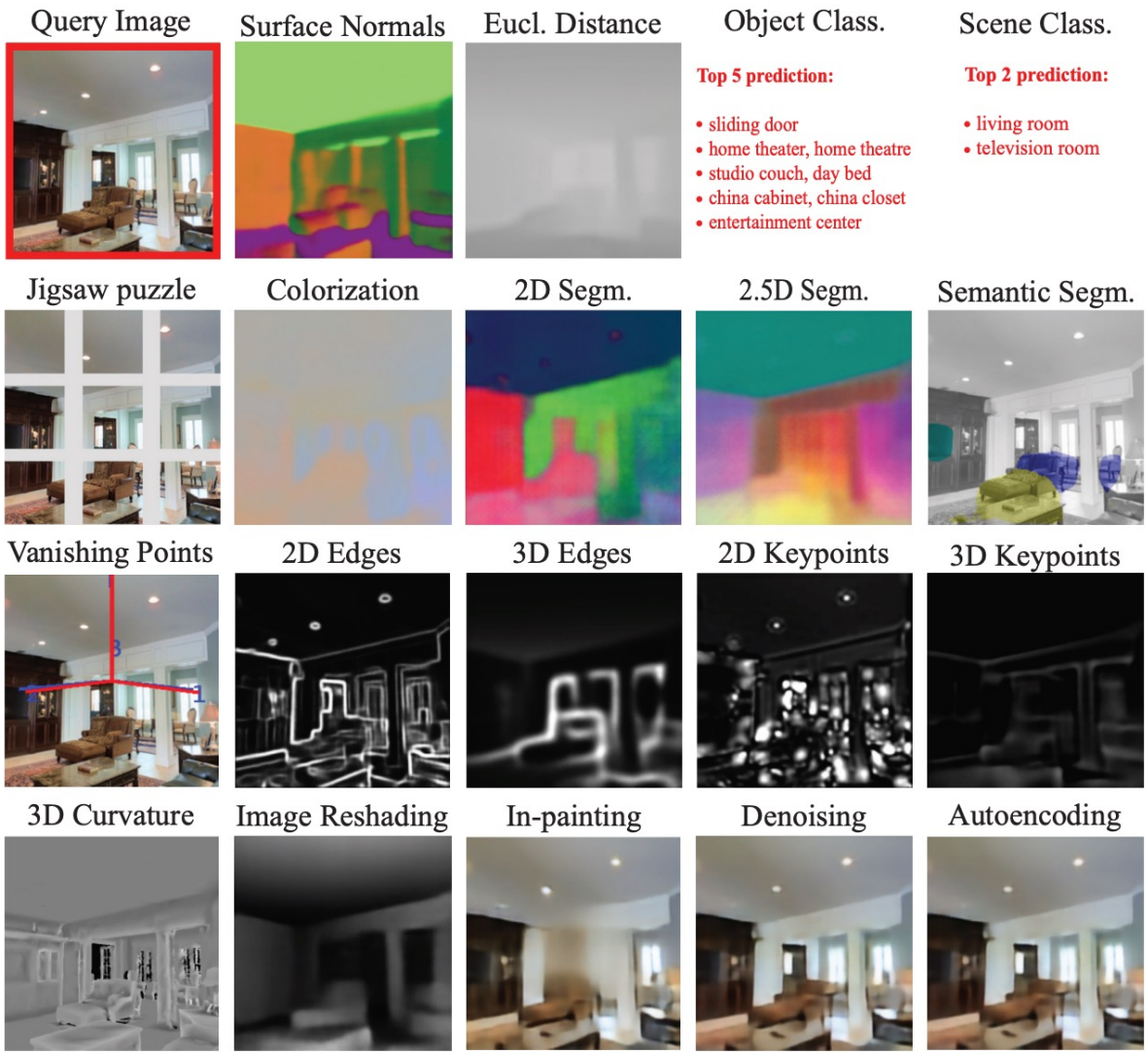
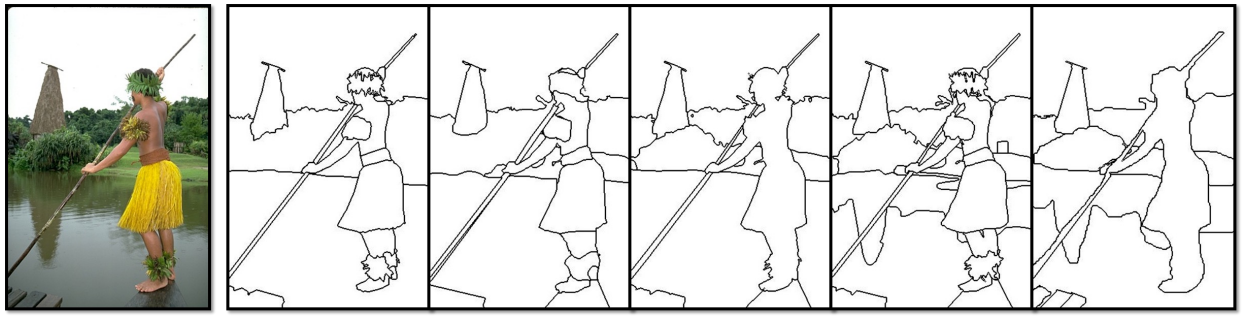

Query Image
Surface Normals
Eucl. Distance
Object Class.
Top 5 prediction:
• sliding door
• home theater, home theatre
• studio couch, day bed
• china cabinet, china closet
• entertainment center
Scene Class.
Top 2 prediction:
• living room
• television room
Jigsaw puzzle
Colorization
2D Segm.
2.5D Segm.
Semantic Segm.
Vanishing Points
2D Edges
3D Edges
2D Keypoints
3D Keypoints
3D Curvature
Image Reshading
In-painting
Denoising
Autoencoding

Image Segmentation

- Subdivide an image into its constituent parts or objects.
- Based on **similarity** and **discontinuity** of intensity/color/texture.

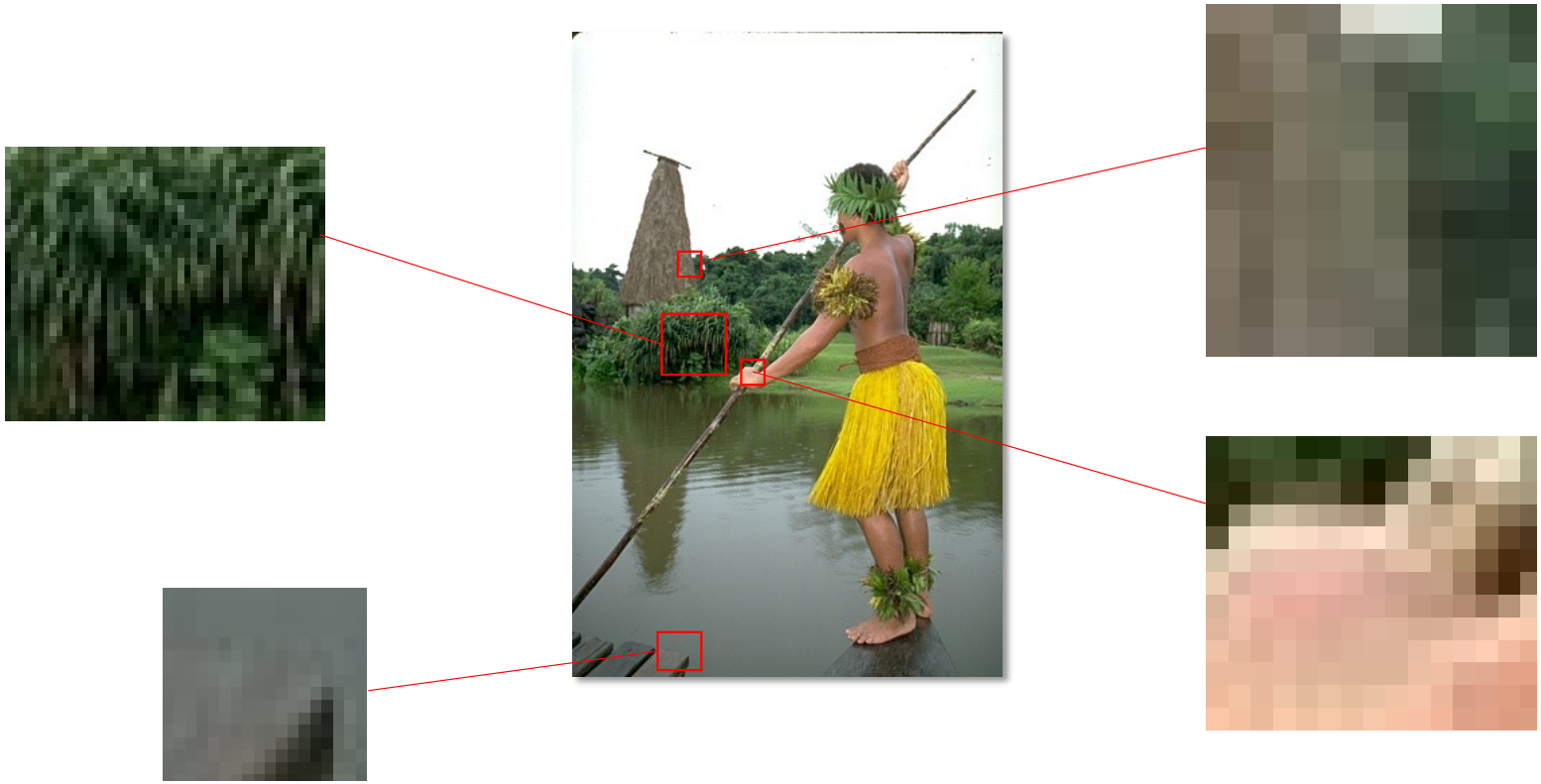
Group pixels with similar features

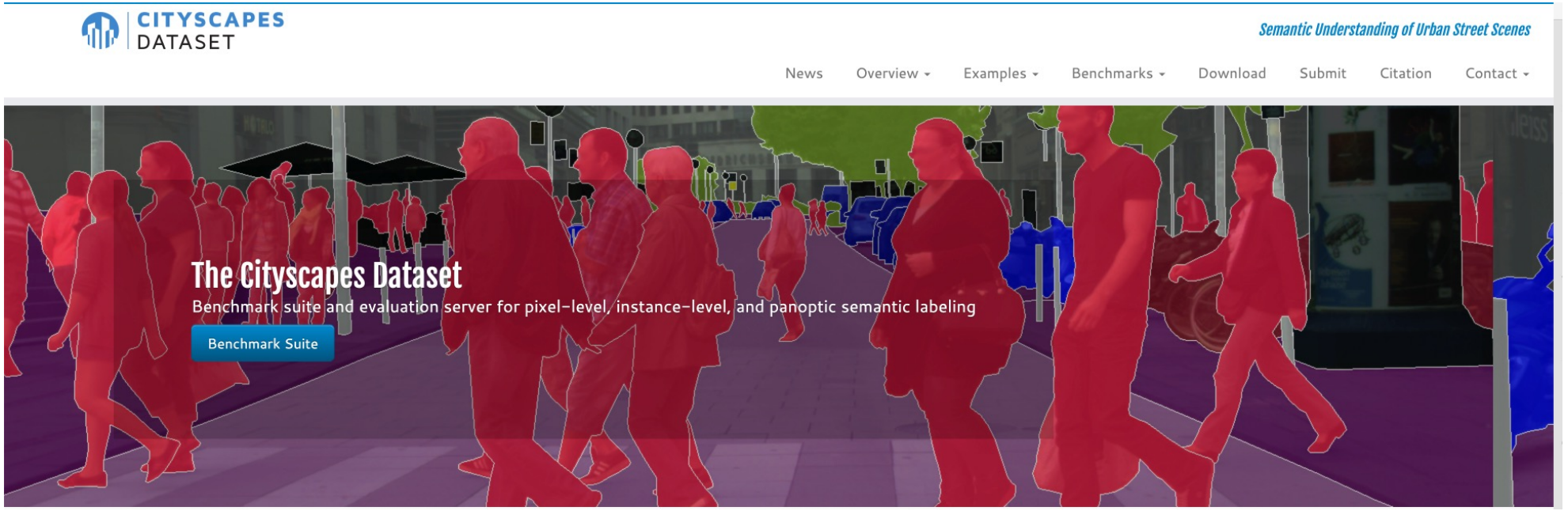
Detect abrupt changes of features



Image

Hand-drawn ground truth



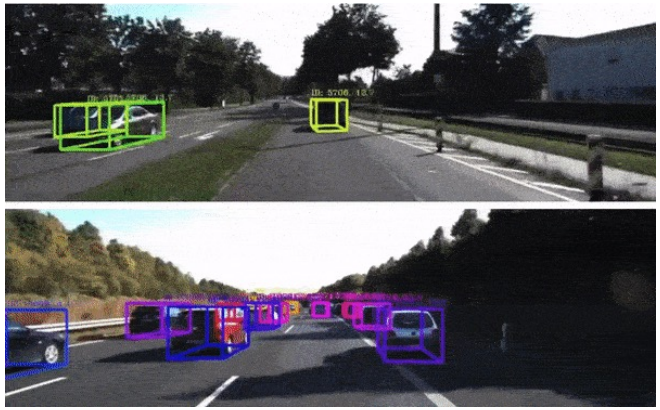


https://www.cityscapes-dataset.com/

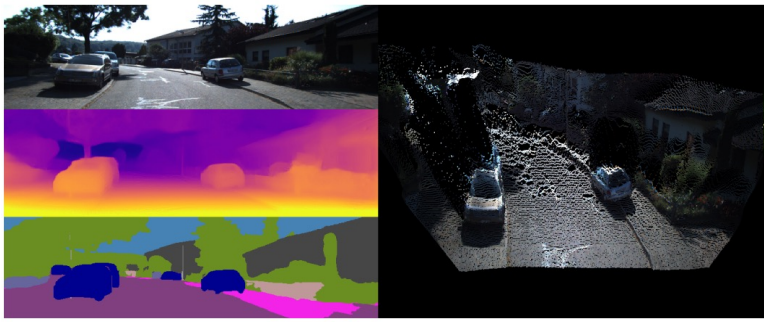
Scene Understanding

Object Detection

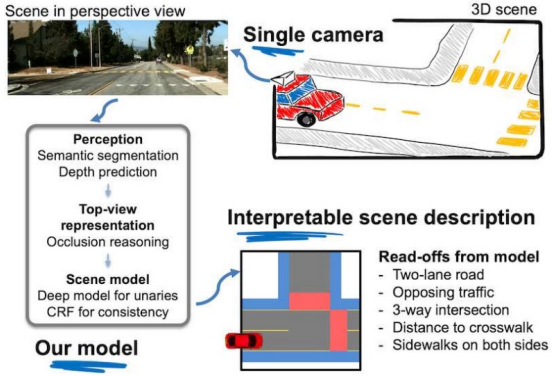
Panoptic Segmentation



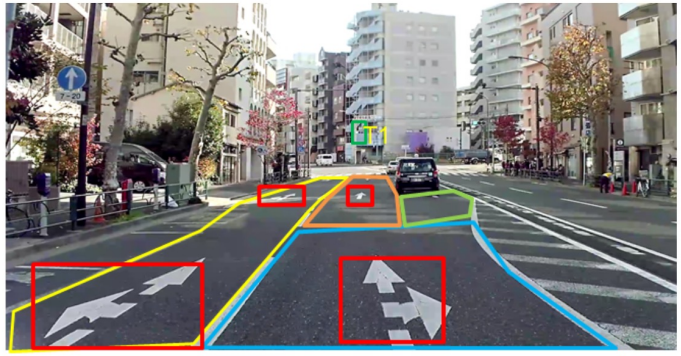
Multiple Object Tracking



Depth Estimation



Road topology

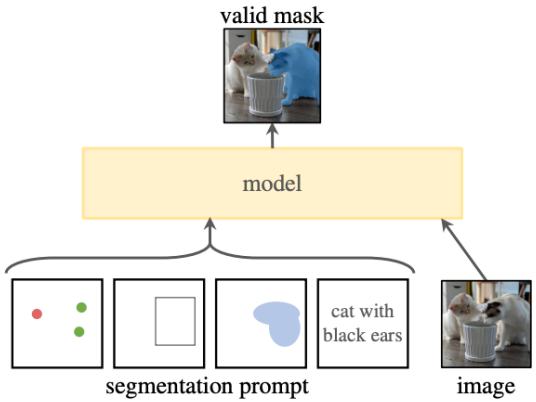


Affordance

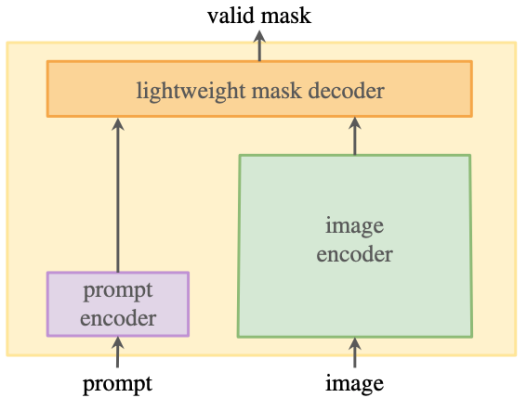
Segment Anything

Alexander Kirillov[1,2,4] Eric Mintun[2] Nikhila Ravi[1,2] Hanzi Mao[2] Chloe Rolland[3] Laura Gustafson[3]
Tete Xiao[3] Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár[4] Ross Girshick[4]
[1]project lead [2]joint first author [3]equal contribution [4]directional lead

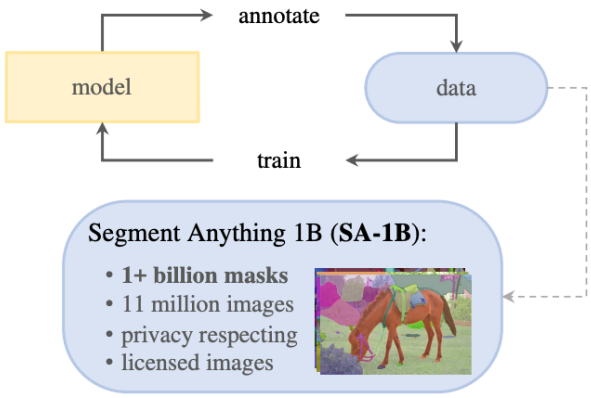
Meta AI Research, FAIR



(a) **Task**: promptable segmentation

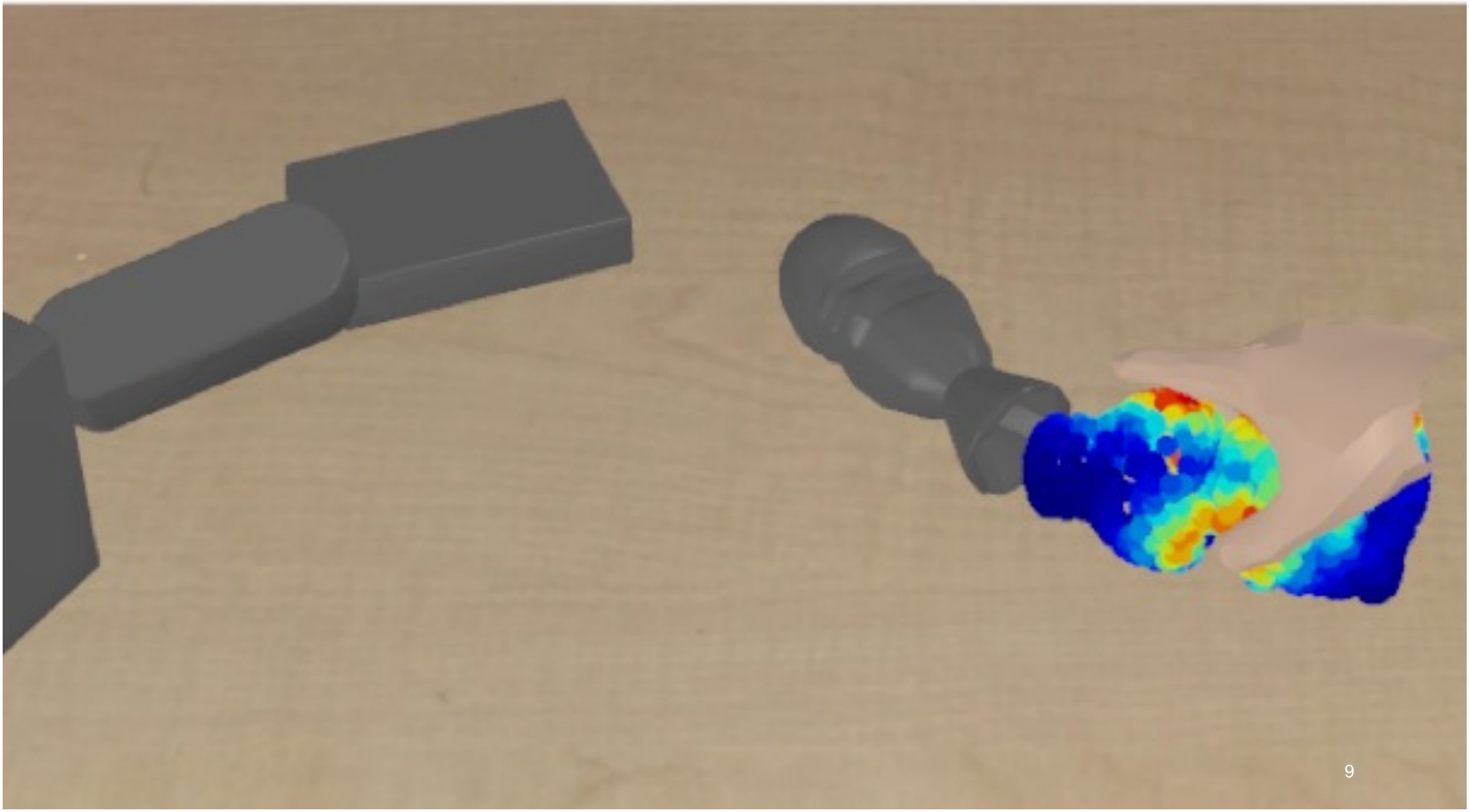


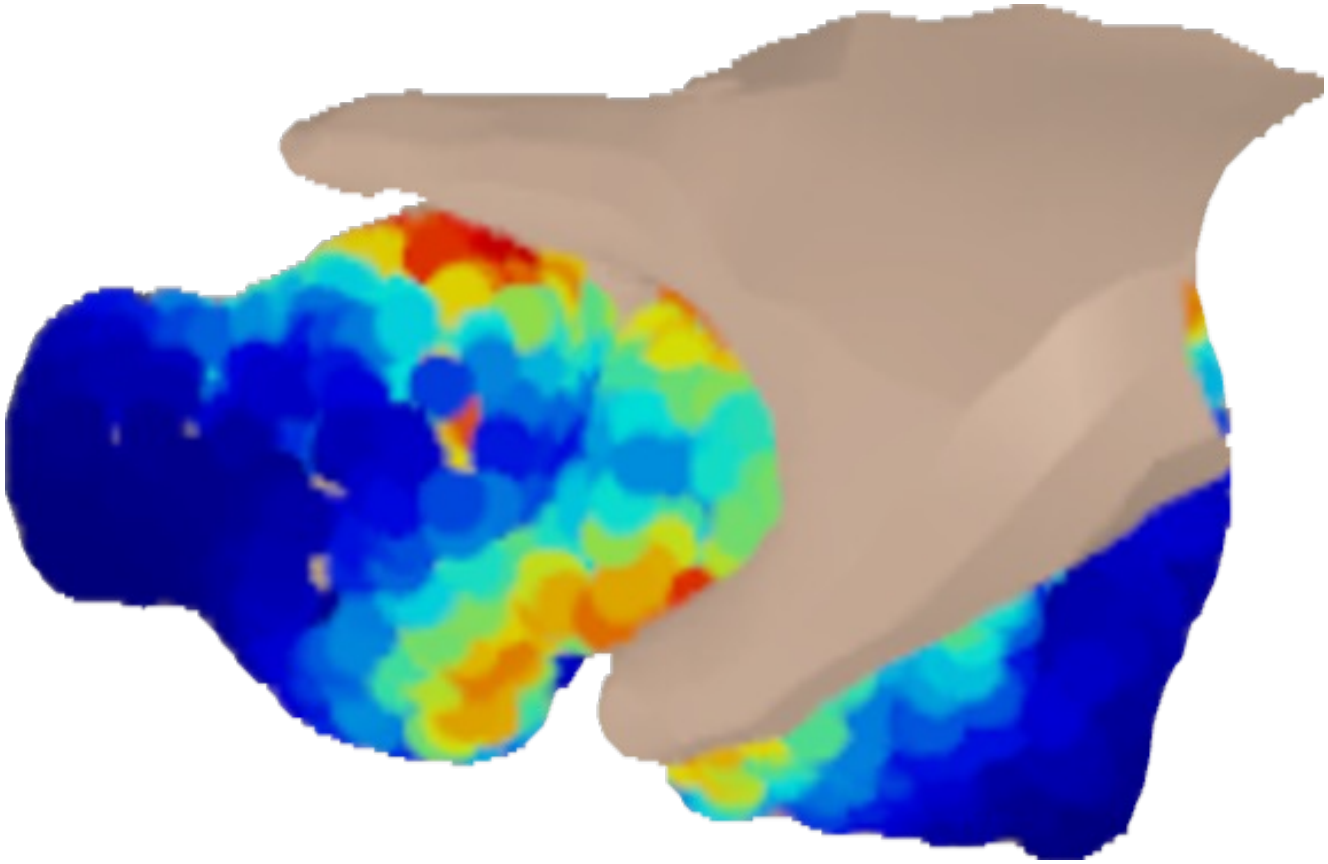
(b) **Model**: Segment Anything Model (**SAM**)



(c) **Data**: data engine (top) & dataset (bottom)

Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation *task*, a segmentation *model* (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a *data* engine for collecting SA-1B, our dataset of over 1 billion masks.





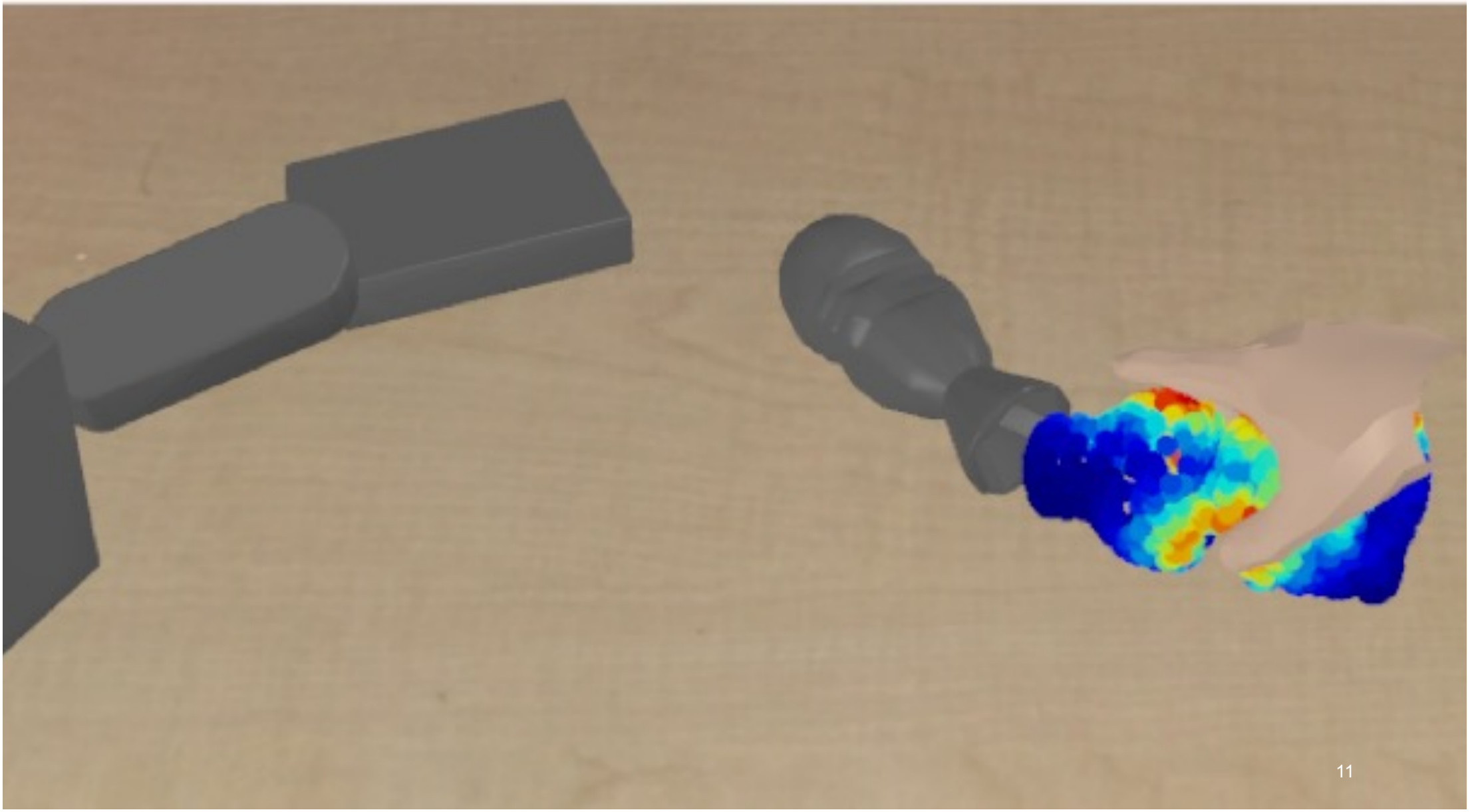
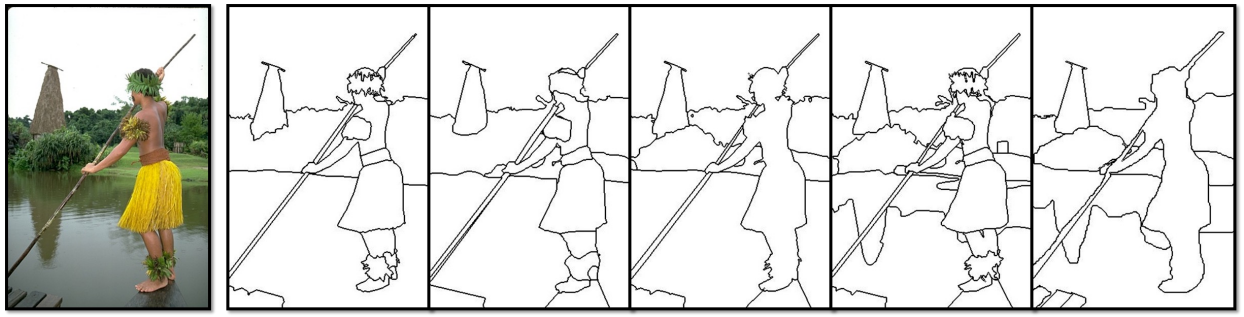


Image Segmentation

- Subdivide an image into its constituent parts or objects.
- Based on **similarity** and **discontinuity** of intensity/color/texture.

Group pixels with similar features

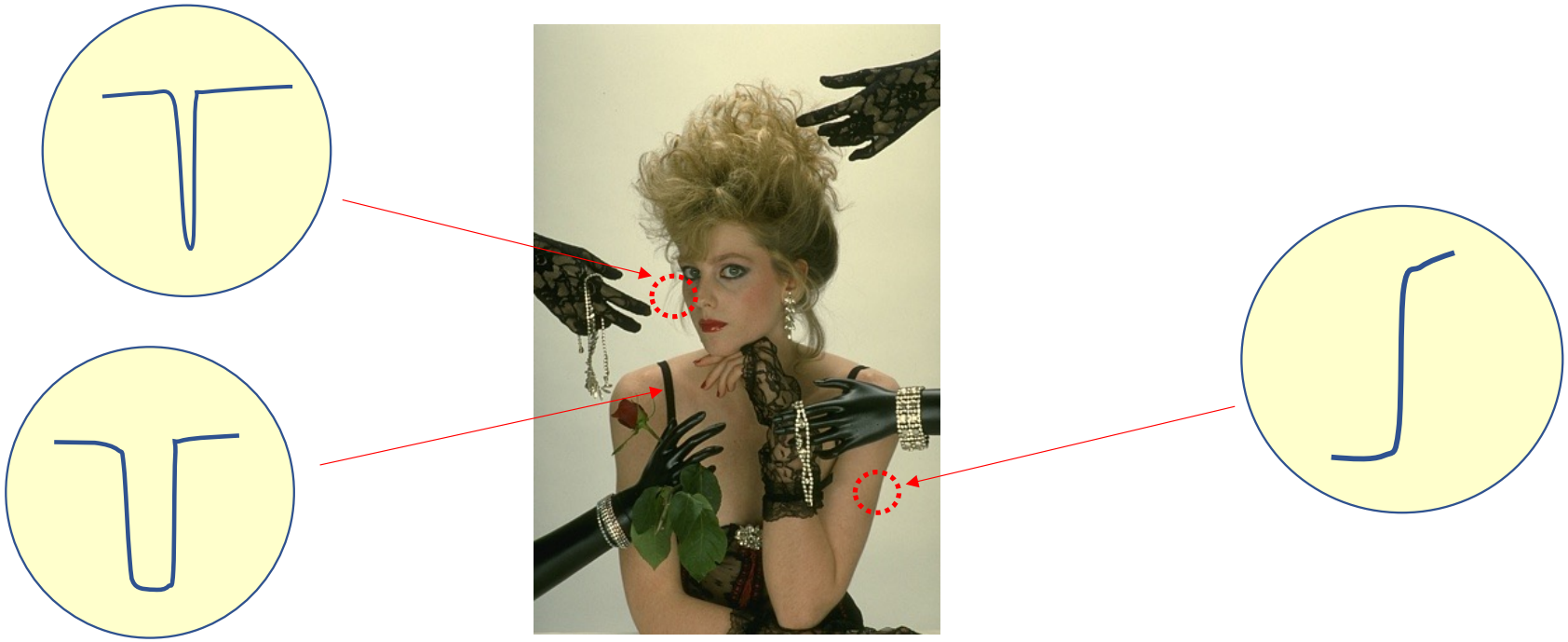
Detect abrupt changes of features



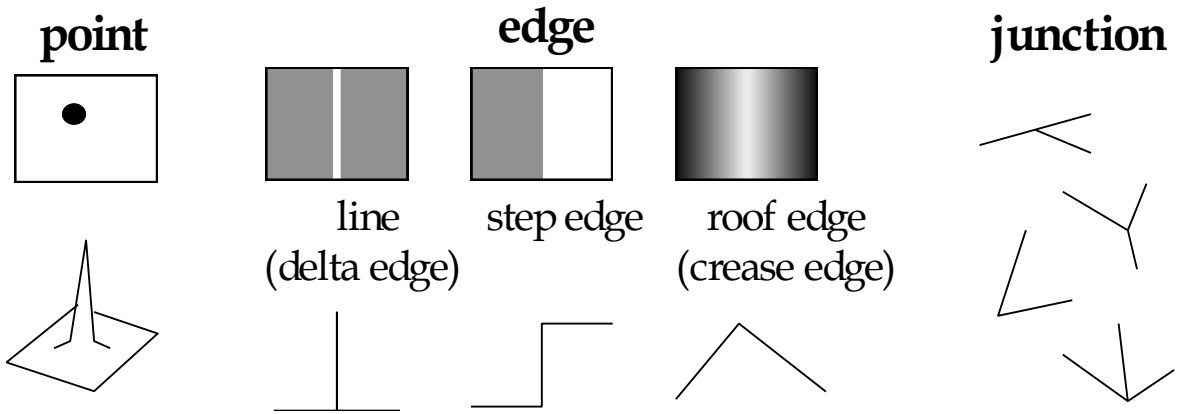
Image

Hand-drawn ground truth

Intensity/Color Discontinuities



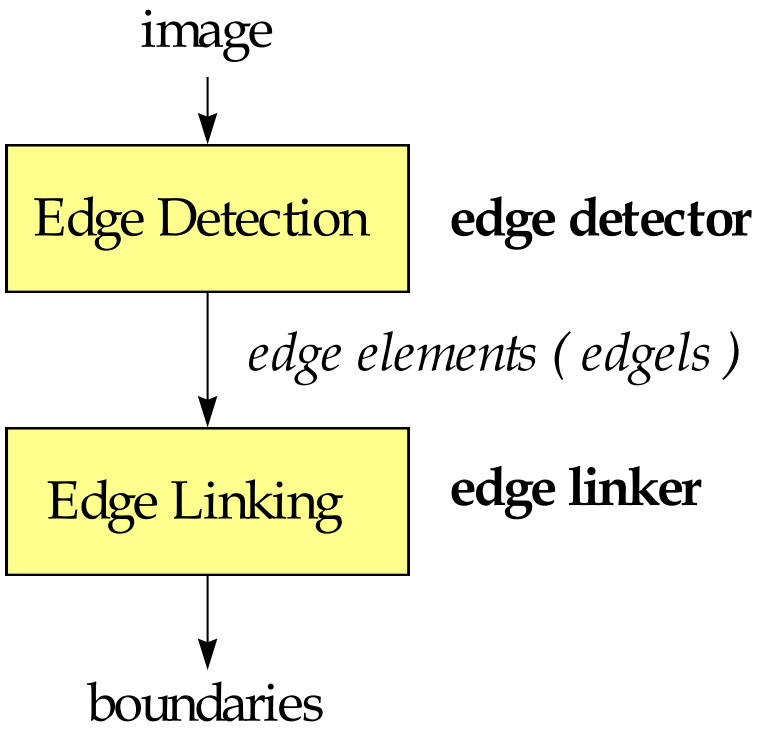
Basic Types of Discontinuities



Most efforts are spent in detecting edges, especially step edges.

Edge-based Segmentation

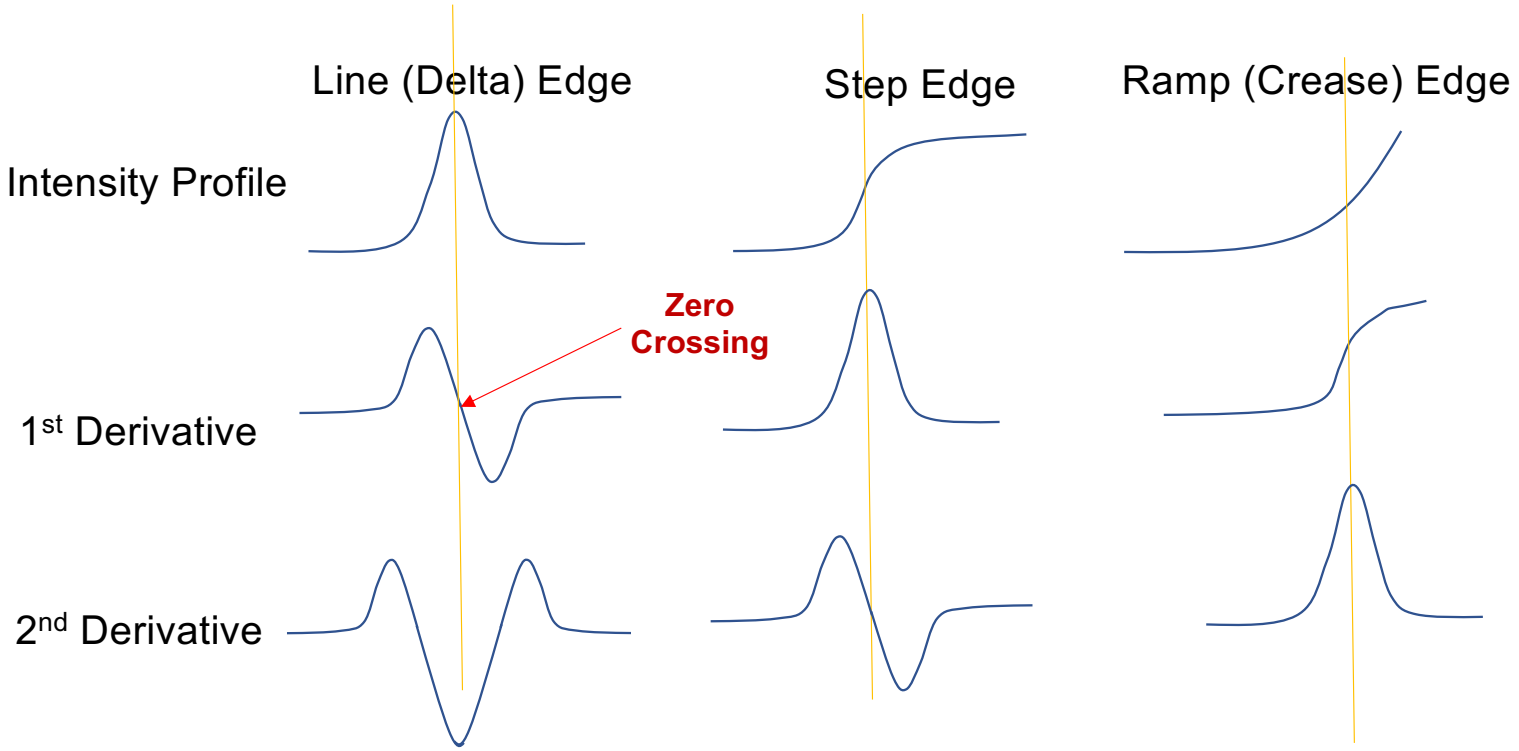
Figure 3.18
Hockenbury/Nolan, *Discovering Psychology*, 7e, © 2016 Worth Publishers

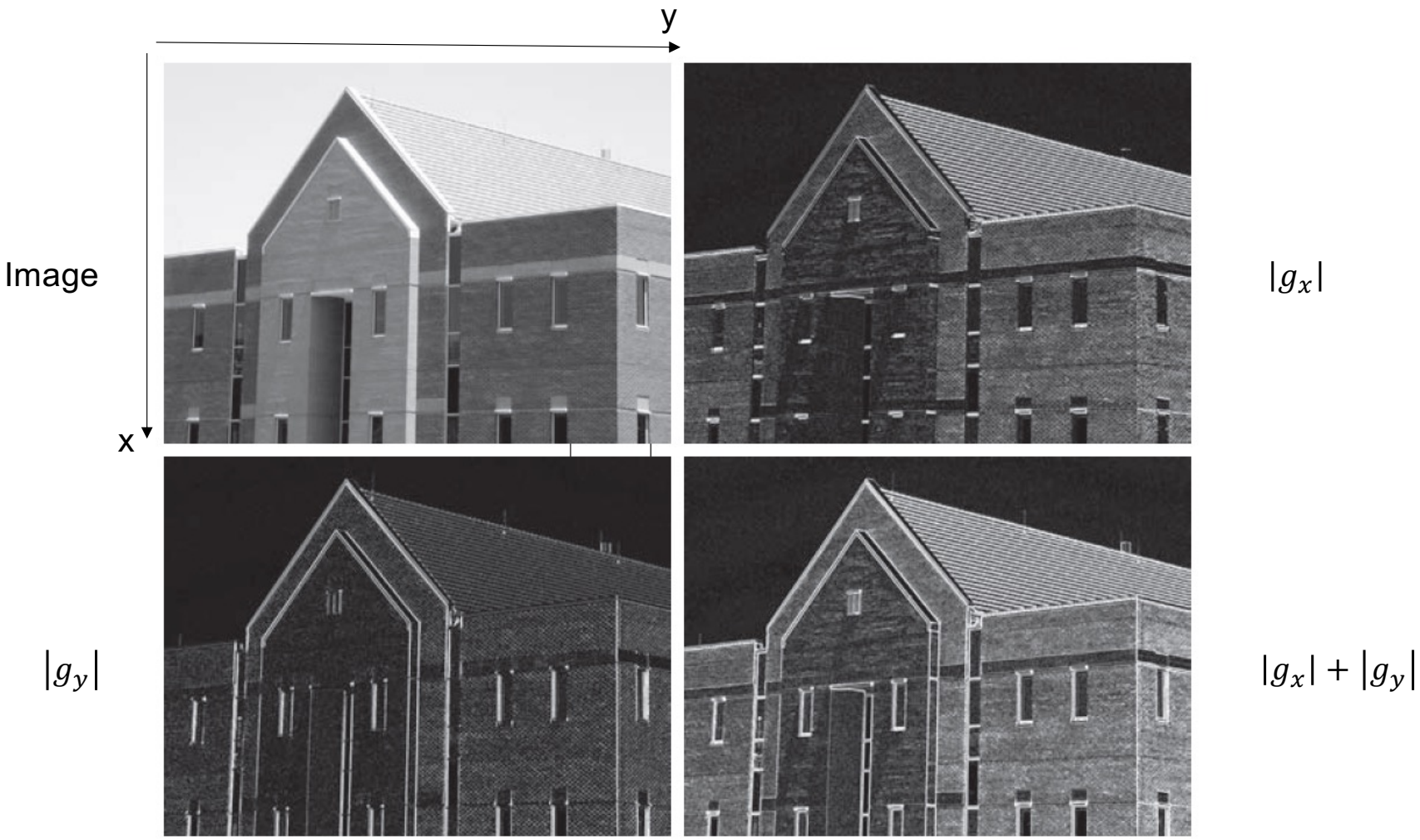


- Template matching
- Statistical methods
- Edge fitting
- Derivative-based methods

- Local approach
- Global approach

Derivative-Based Methods

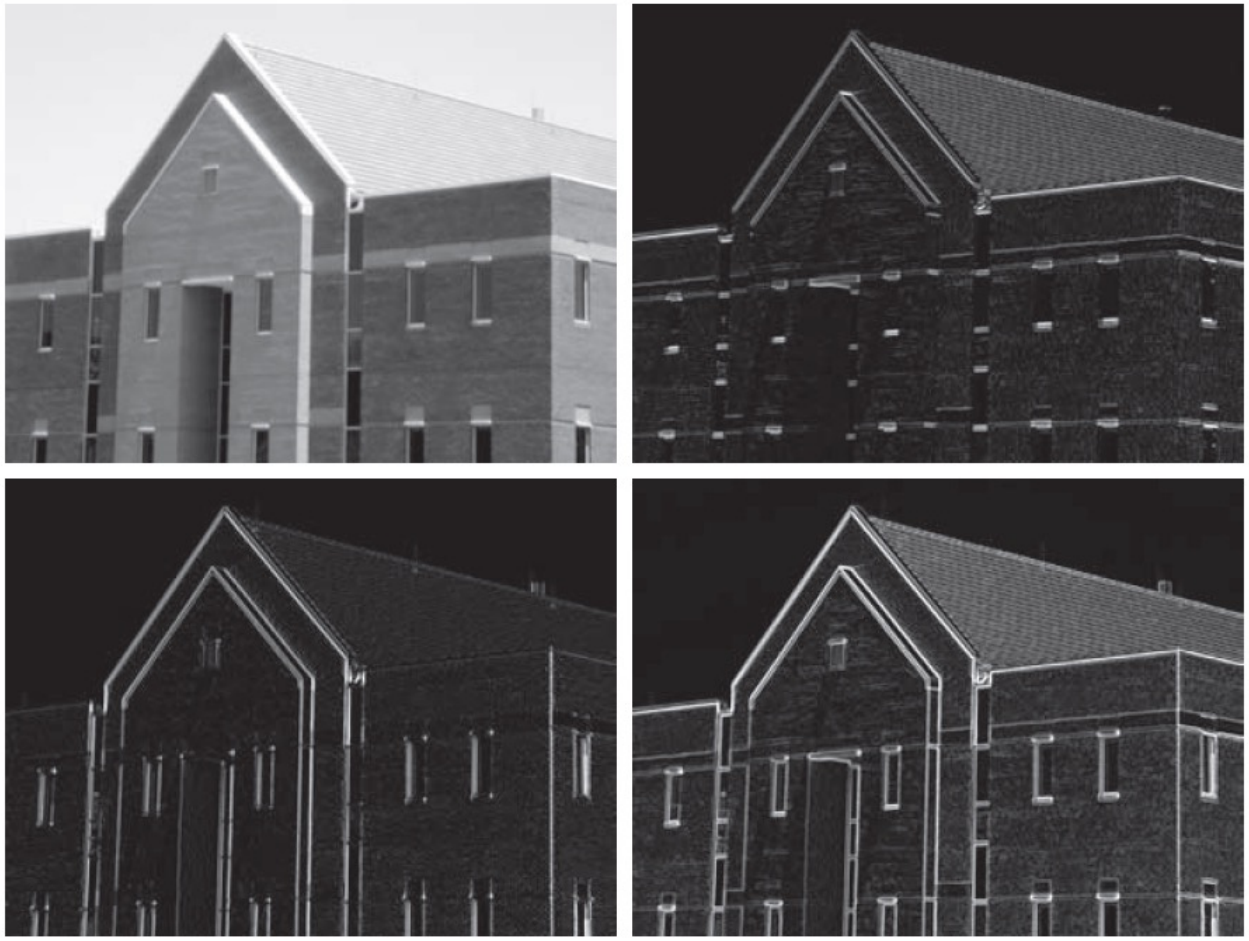



y
x
Image
$|g_x|$
$|g_y|$
$|g_x| + |g_y|$

Smoothed Image

y

x



$|g_x|$

$|g_y|$

$|g_x| + |g_y|$

Detection of strong edges by setting thresholds on the gradient image

Result of original image

Result of smoothed image

- **2nd derivative: (insensitive to shading)**

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

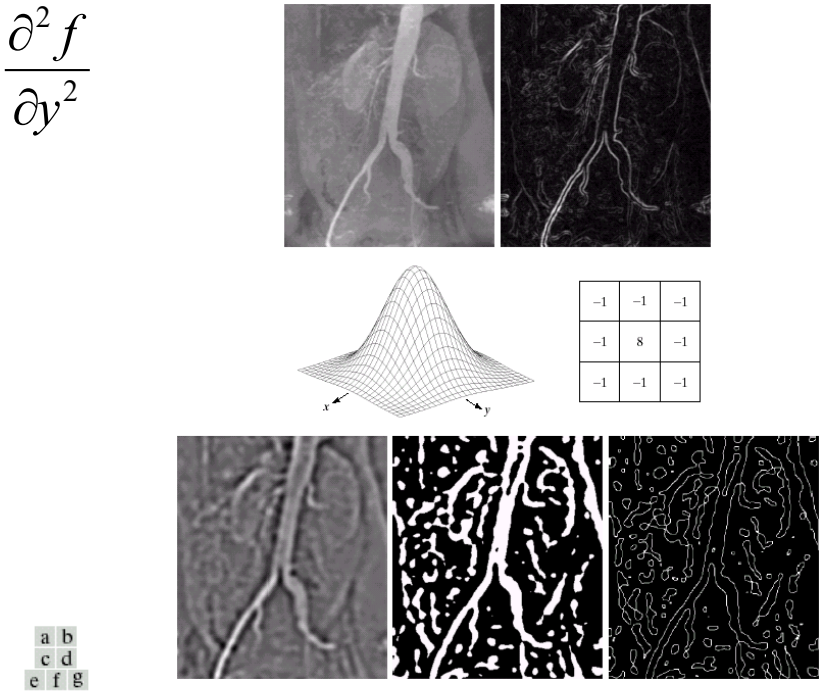
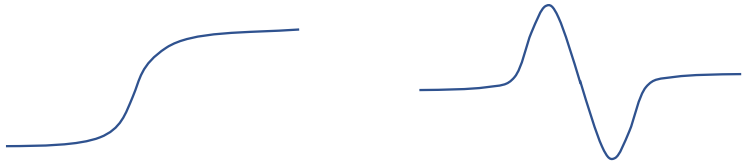


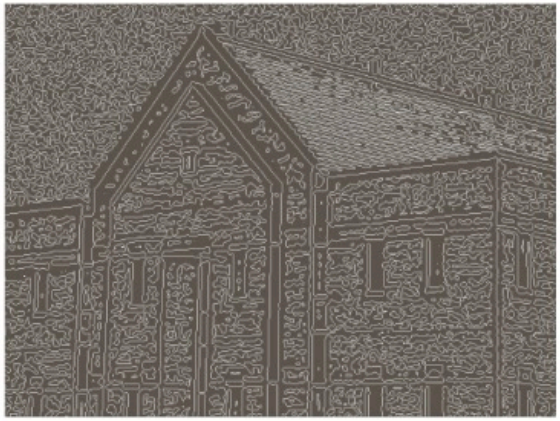
FIGURE 10.15 (a) Original image. (b) Sobel gradient (shown for comparison). (c) Spatial Gaussian smoothing function. (d) Laplacian mask. (e) LoG. (f) Thresholded LoG. (g) Zero crossings. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)



$f(x, y)$

$\nabla^2 f(x, y)$

Zero crossings



Detected Edges

Canny Operator (1983)

John Canny

FOLLOW

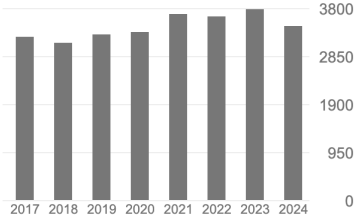
University of California, Berkeley
Verified email at cs.berkeley.edu

HCI Ubicomp ICTD Data Mining Health Technologies

TITLE	CITED BY	YEAR
A computational approach to edge detection J Canny IEEE Transactions on pattern analysis and machine intelligence, 679-698	45915	1986
The complexity of robot motion planning J Canny MIT press	2378	1988
Finding edges and lines in images JF Canny	1575	1983
Planning optimal grasps. C Ferrari, JF Canny ICRA 3 (4), 6	1216	1992
Evaluating protein transfer learning with TAPE R Rao, N Bhattacharya, N Thomas, Y Duan, P Chen, J Canny, P Abbeel, ... Advances in neural information processing systems 32	905	2019
Some algebraic and geometric computations in PSPACE J Canny Proceedings of the twentieth annual ACM symposium on Theory of computing …	841	1988

Cited by VIEW ALL

	All	Since 2019
Citations	78539	21194
h-index	91	45
i10-index	247	143



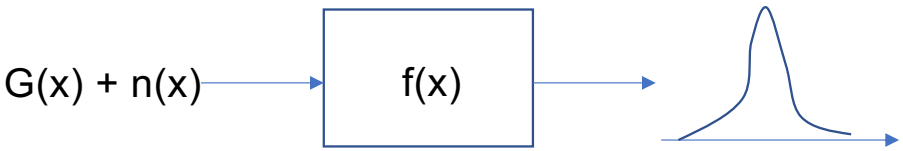
Public access VIEW ALL

0 articles	18 articles
not available	available

Based on funding mandates

Canny, J., A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986

Canny Operator (1983)



In summary, the three performance criteria are as follows:

1) Good detection. There should be a low probability of failing to mark real edge points, and low probability of falsely marking nonedge points. Since both these probabilities are monotonically decreasing functions of the output signal-to-noise ratio, this criterion corresponds to maximizing signal-to-noise ratio.

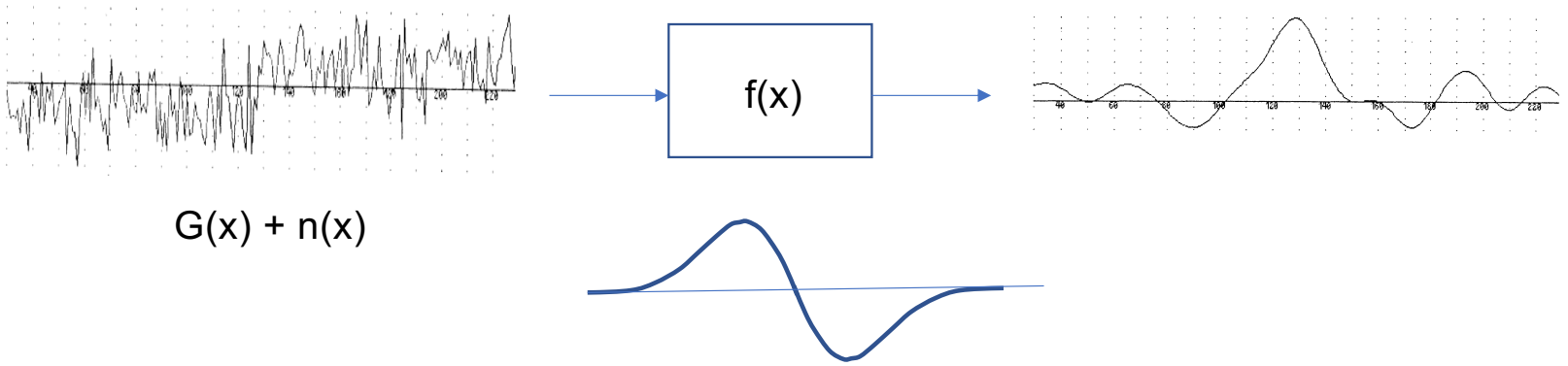
2) Good localization. The points marked as edge points by the operator should be as close as possible to the center of the true edge.

3) Only one response to a single edge. This is implicitly captured in the first criterion since when there are two responses to the same edge, one of them must be considered false. However, the mathematical form of the first criterion did not capture the multiple response requirement and it had to be made explicit.

Canny, J., A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986

By numerical optimization, the resulting filter for detecting ***step*** edges can be approximated effectively by the **first derivative of a Gaussian smoothing filter**.

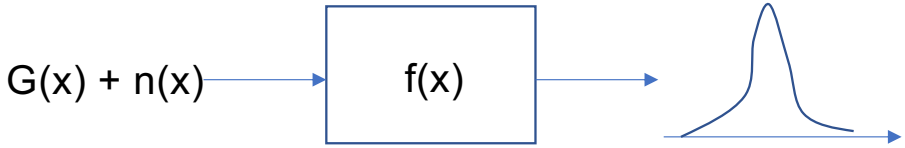
$$\frac{\partial}{\partial x}(f(x,y) * G(x,y)) = f(x,y) * (\frac{\partial G(x,y)}{\partial x})$$



Canny Operator (1983)

1-D case

G(x): signal
f(x): impulse response of the filter



3 performance criteria

- Good detection

$$SNR = \frac{\left|\int_{-W}^{W} G(-x) f(x) dx\right|}{n_0 \sqrt{\int_{-W}^{W} f^2(x) dx}}$$

- Good localization

$$Localization = \frac{\left|\int_{-W}^{W} G'(-x) f'(x) dx\right|}{n_0 \sqrt{\int_{-W}^{W} f'^2(x) dx}}$$

- Only one response to a single edge

$$\text{mean distance between zero} - \text{crossings of } f' = x_{zc}(f) = \pi \sqrt{\frac{\int_{-\infty}^{\infty} f'^2(x) dx}{\int_{-\infty}^{\infty} f''^2(x) dx}}$$

Canny, J., A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986

2-D case

Perform detection along the edge norma

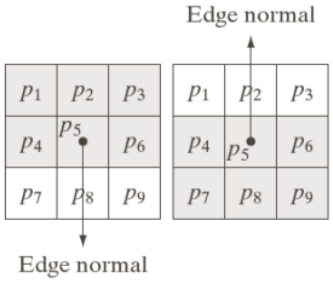
$$f_s(x,y) = G(x,y) * f(x,y)$$

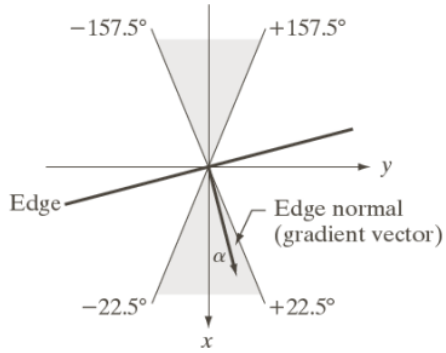
$$g_x = \frac{\partial f_s}{\partial x} = f(x,y) * G_x(x,y)$$

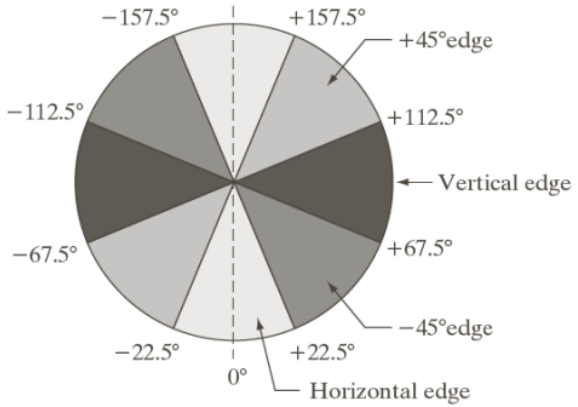
$$g_y = \frac{\partial f_s}{\partial y} = f(x,y) * G_y(x,y)$$

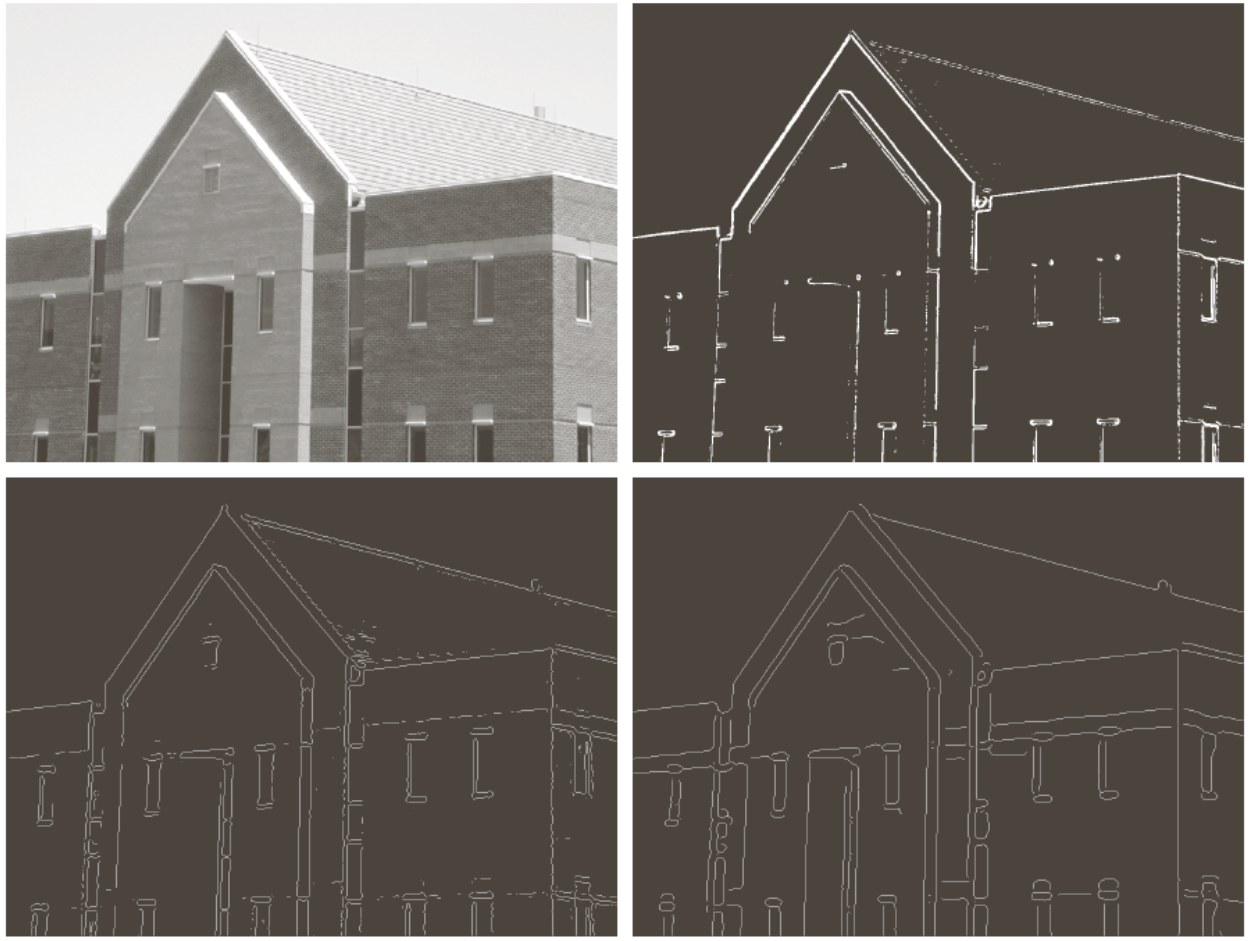
$$M(x,y) = \sqrt{g_x^2 + g_y^2}$$

$$\alpha(x,y) = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$



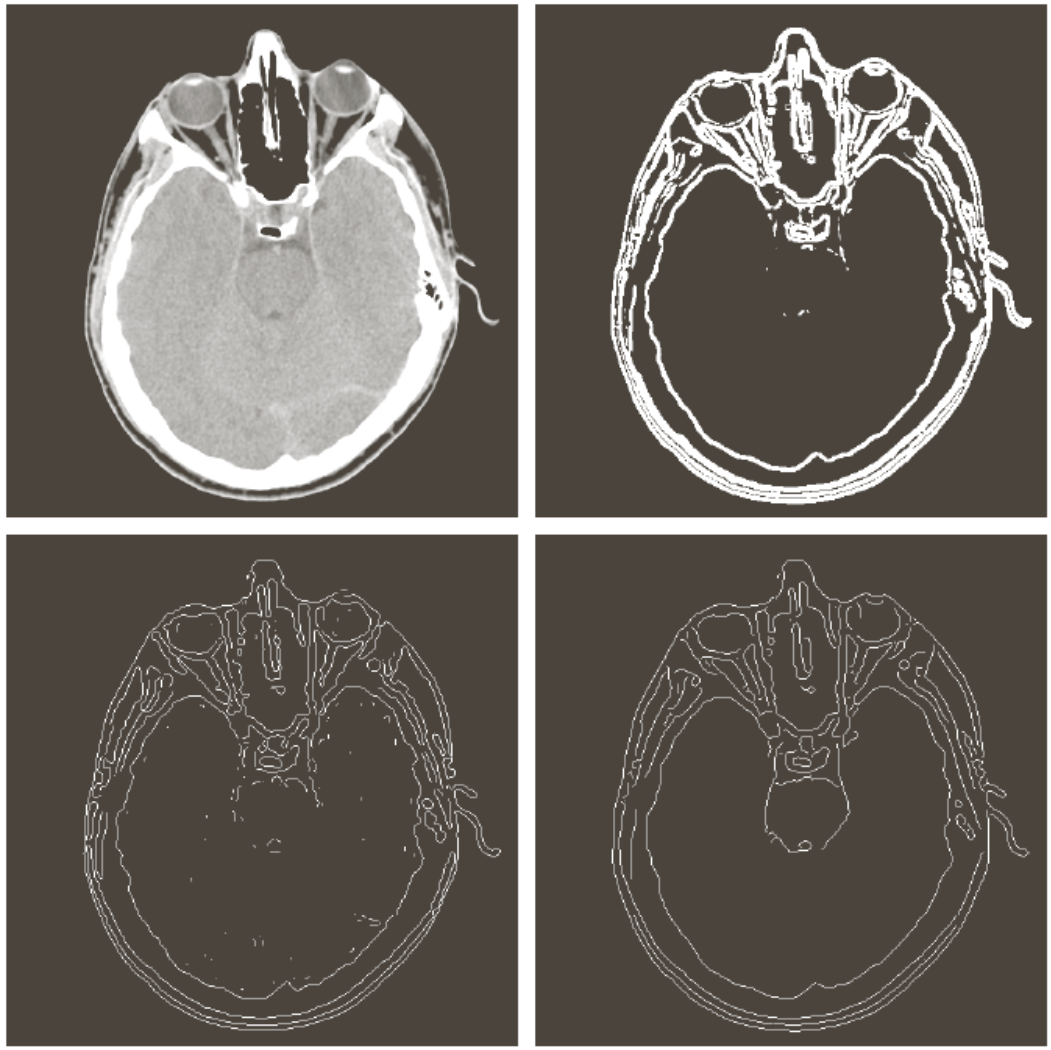






a b
c d

FIGURE 10.25
(a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1].
(b) Thresholded gradient of smoothed image.
(c) Image obtained using the Marr-Hildreth algorithm.
(d) Image obtained using the Canny algorithm.
Note the significant improvement of the Canny image compared to the other two.



a b
c d

FIGURE 10.26
(a) Original head CT image of size 512 × 512 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

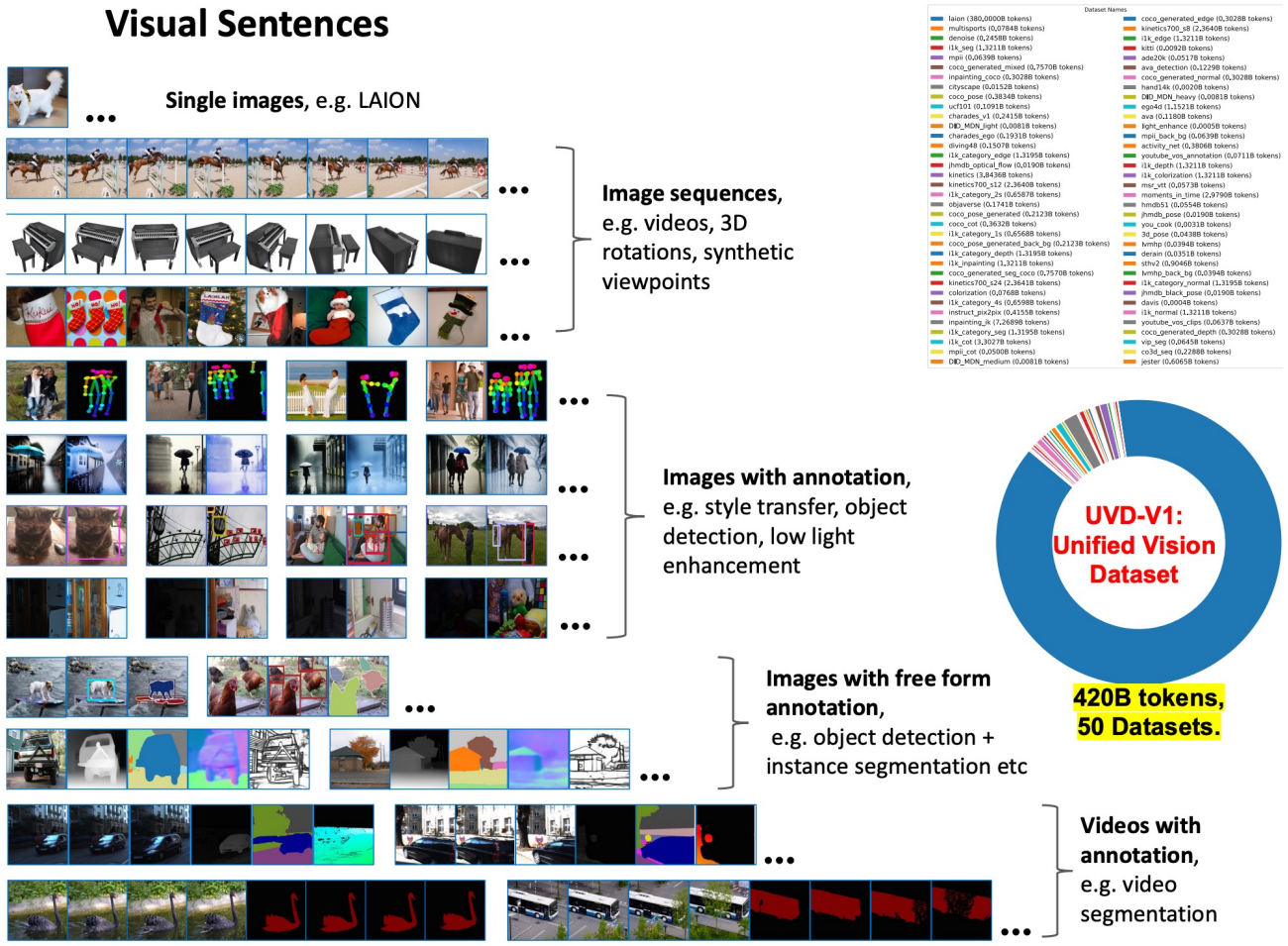
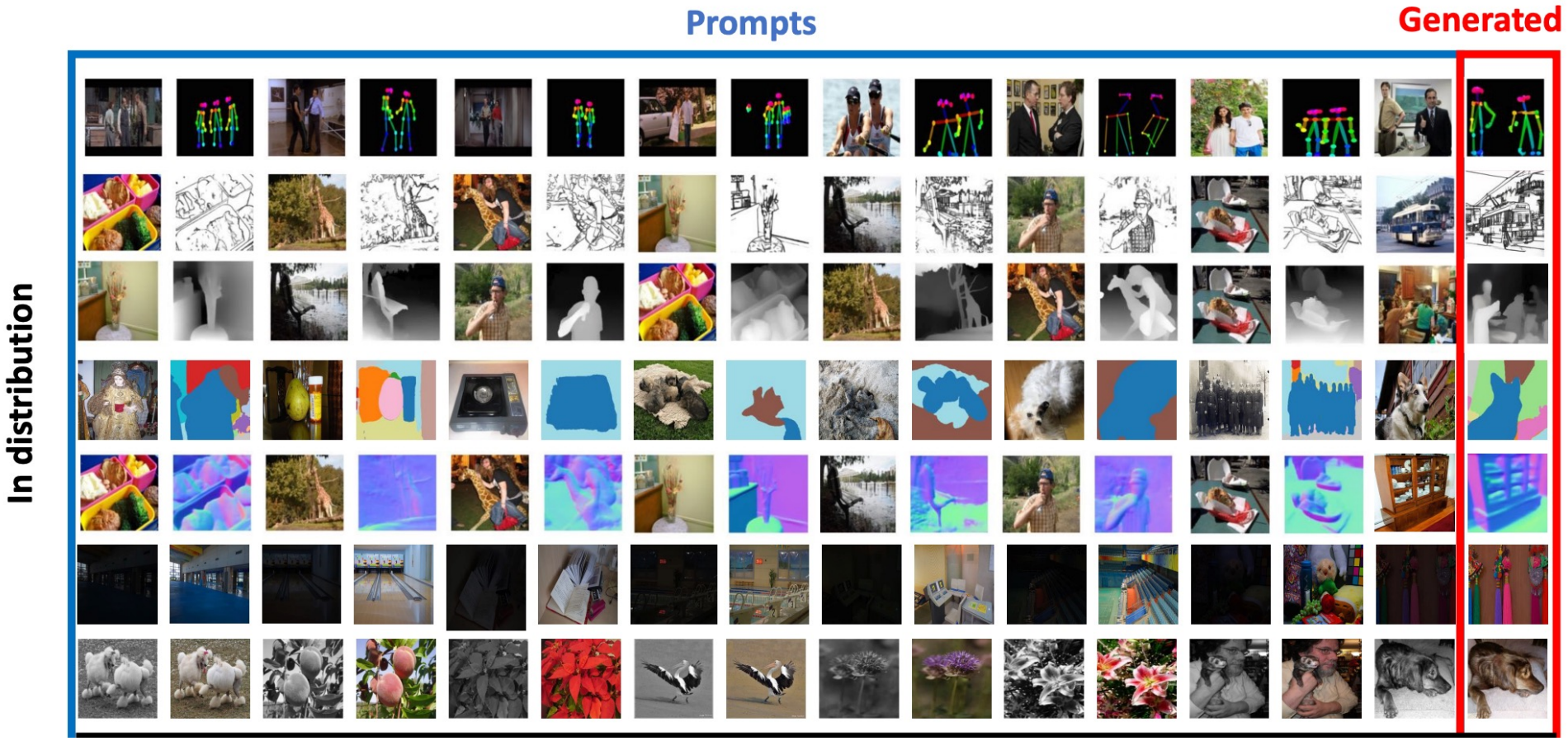


Figure 1. **Visual sentences** allow us to format diverse vision data into the unified structure of image sequences.


Prompts
Generated
In distribution

Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials

Philipp Krähenbühl
Computer Science Department
Stanford University
philkr@cs.stanford.edu

Vladlen Koltun
Computer Science Department
Stanford University
vladlen@cs.stanford.edu

Abstract

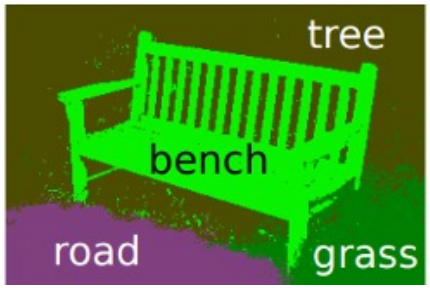
Most state-of-the-art techniques for multi-class image segmentation and labeling use conditional random fields defined over pixels or image regions. While region-level models often feature dense pairwise connectivity, pixel-level models are considerably larger and have only permitted sparse graph structures. In this paper, we consider fully connected CRF models defined on the complete set of pixels in an image. The resulting graphs have billions of edges, making traditional inference algorithms impractical. Our main contribution is a highly efficient approximate inference algorithm for fully connected CRF models in which the pairwise edge potentials are defined by a linear combination of Gaussian kernels. Our experiments demonstrate that dense connectivity at the pixel level substantially improves segmentation and labeling accuracy.

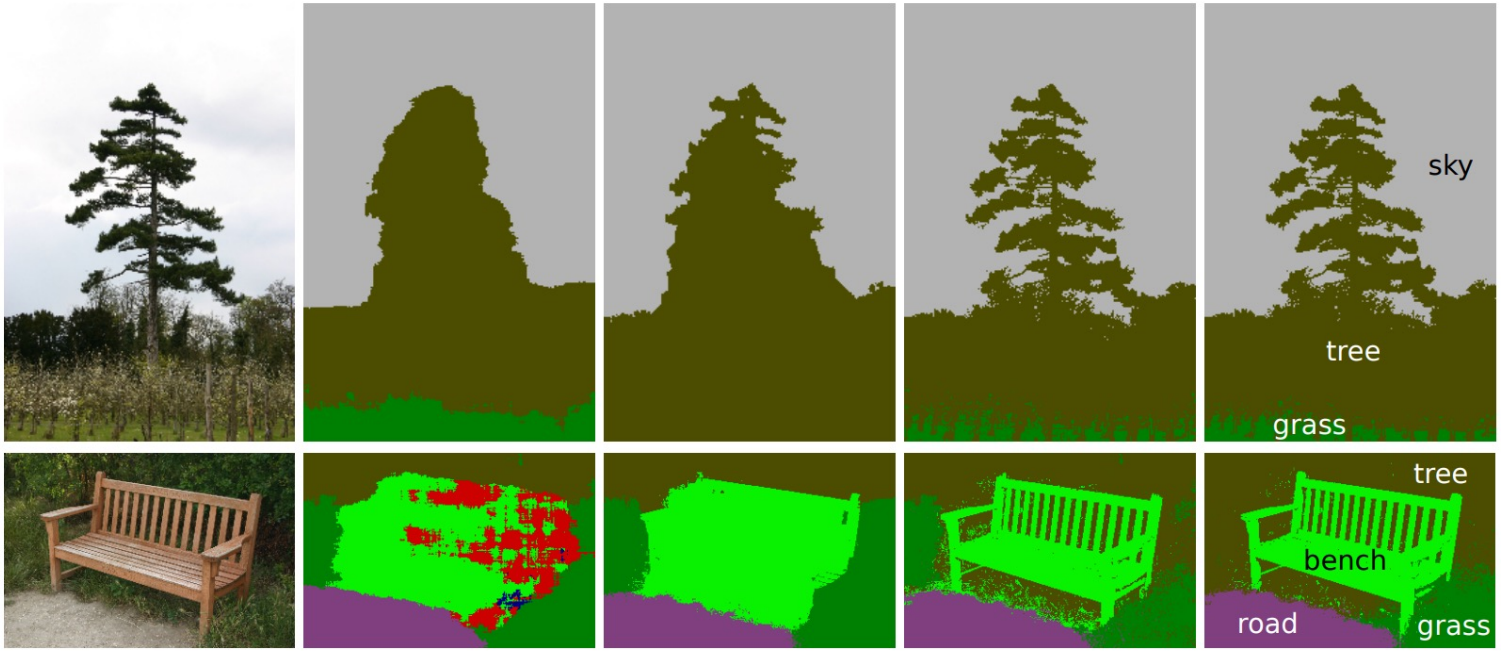
Task-specific Prediction

I

X

Image segmentation





(a) Image (b) Unary classifiers (c) Robust P^n CRF (d) Fully connected CRF, MCMC inference, 36 hrs (e) Fully connected CRF, our approach, 0.2 seconds

Figure 1: Pixel-level classification with a fully connected CRF. (a) Input image from the MSRC-21 dataset. (b) The response of unary classifiers used by our models. (c) Classification produced by the Robust P^n CRF [9]. (d) Classification produced by MCMC inference [17] in a fully connected pixel-level CRF model; the algorithm was run for 36 hours and only partially converged for the bottom image. (e) Classification produced by our inference algorithm in the fully connected model in 0.2 seconds.

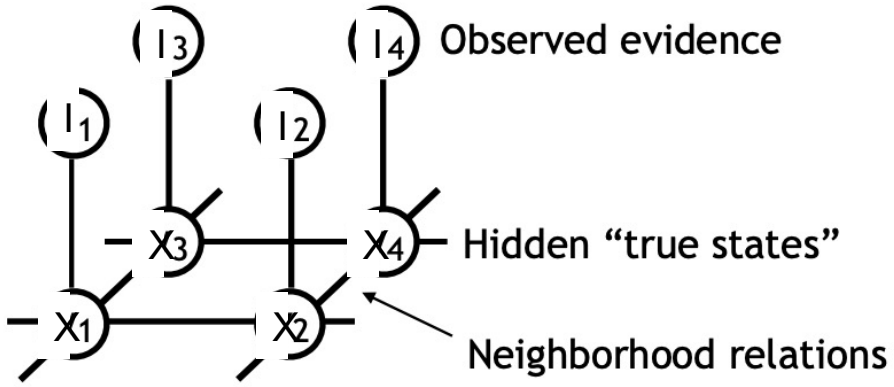
J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. ICML, 2001.

Conditional Random Field

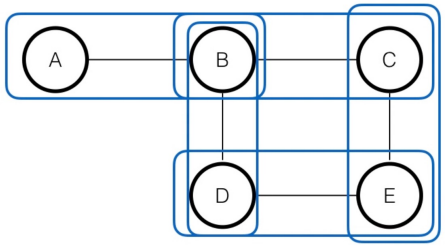
A conditional random field (I,X) is characterized by Gibbs Distribubtion

$$P(\mathbf{X}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp(-\sum_{c \in \mathcal{C}_\mathcal{G}} \phi_c(\mathbf{X}_c|\mathbf{I}))$$

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph on $\mathbf{X}$ and each clique c in a set of cliques $\mathcal{C}_\mathcal{G}$ in $\mathcal{G}$ induces a potential ϕ_c



Undirected Graphical Models



$$P(A, B, C, D, E) \propto \phi(A, B)\phi(B, C)\phi(B, D)\phi(C, E)\phi(D, E)$$

$$P(X) = \frac{1}{Z} \prod_{c \in \text{cliques}(G)} \phi_C(x_c)$$ potential functions

https://youtu.be/iBQkZdPHICs?si=8Ysr0palWK5N_Zzy

Gibbs Energy of Labels

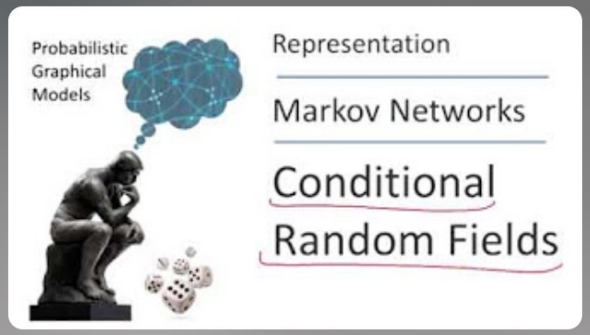
$$P(\mathbf{X}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp(-\sum_{c \in \mathcal{C}_\mathcal{G}} \phi_c(\mathbf{X}_c|\mathbf{I}))$$

$$E(\mathbf{x}|\mathbf{I}) = \sum_{c \in \mathcal{C}_\mathcal{G}} \phi_c(\mathbf{x}_c|\mathbf{I})$$

The maximum a posteriori (MAP) labeling of the random field is

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{L}^N} P(\mathbf{x}|\mathbf{I})$$

https://graphics.stanford.edu/projects/densecrf/densecrf.pdf



probabilistic graphical models

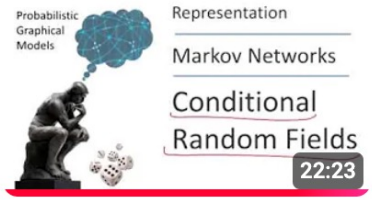
by Danai Triantafyllidou

Playlist • 10 videos • 9,253 views

Play all

https://youtube.com/playlist?list=PLQI7D2xuMMNq5lj52YpCjGvgOrjvX4h5G&si=YX9jI8WHV2ssbzmr

1



Conditional Random Fields - Stanford University (By Daphne Koller)

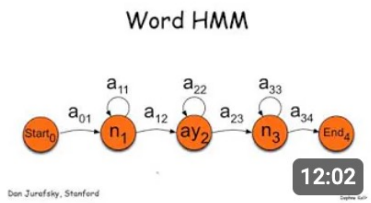
Machine Learning TV • 108K views • 7 years ago

2

Pairwise Markov Networks - Stanford University

Machine Learning TV • 10K views • 7 years ago

3



Template Models: Hidden Markov Models - Stanford University

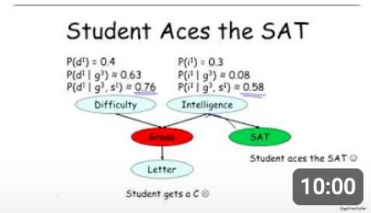
Machine Learning TV • 79K views • 7 years ago

4

Semantics & Factorization - Stanford University

Machine Learning TV • 8.1K views • 7 years ago

5



Reasoning Patterns - Stanford University

Machine Learning TV • 4.8K views • 7 years ago

In the fully connected pairwise CRF model, $\mathcal{G}$ is the complete graph on $\mathbf{X}$ and $\mathcal{C}_{\mathcal{G}}$ is the set of all unary and pairwise cliques. The corresponding Gibbs energy is

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j), \tag{1}$$

where i and j range from 1 to N. The unary potential $\psi_u(x_i)$ is computed independently for each pixel by a classifier that produces a distribution over the label assignment x_i given image features. The unary potential used in our implementation incorporates shape, texture, location, and color descriptors and is described in Section 5. Since the output of the unary classifier for each pixel is produced independently from the outputs of the classifiers for other pixels, the MAP labeling produced by the unary classifiers alone is generally noisy and inconsistent, as shown in Figure 1(b).

https://graphics.stanford.edu/projects/densecrf/densecrf.pdf

In the fully connected pairwise CRF model, $\mathcal{G}$ is the complete graph on $\mathbf{X}$ and $\mathcal{C}_{\mathcal{G}}$ is the set of all unary and pairwise cliques. The corresponding Gibbs energy is

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j), \tag{1}$$

where i and j range from 1 to N. The unary potential $\psi_u(x_i)$ is computed independently for each pixel by a classifier that produces a distribution over the label assignment x_i given image features. The unary potential used in our implementation incorporates shape, texture, location, and color descriptors and is described in Section 5. Since the output of the unary classifier for each pixel is produced independently from the outputs of the classifiers for other pixels, the MAP labeling produced by the unary classifiers alone is generally noisy and inconsistent, as shown in Figure 1(b).

The unary potentials used in our implementation are derived from TextonBoost [19, 13]. We use the 17-dimensional filter bank suggested by Shotton et al. [19], and follow Ladický et al. [13] by adding color, histogram of oriented gradients (HOG), and pixel location features. Our evaluation

https://graphics.stanford.edu/projects/densecrf/densecrf.pdf

(a) Image (b) Unary classifiers

wise CRF model, $\mathcal{G}$ is the complete graph on $\mathbf{X}$ and $\mathcal{C}_{\mathcal{G}}$ is the set of all
The corresponding Gibbs energy is

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j), \quad (1)$$

to N. The unary potential $\psi_u(x_i)$ is computed independently for each
oduces a distribution over the label assignment x_i given image features.
n our implementation incorporates shape, texture, location, and color
d in Section 5. Since the output of the unary classifier for each pixel
from the outputs of the classifiers for other pixels, the MAP labeling
sifiers alone is generally noisy and inconsistent, as shown in Figure 1(b).

in our implementation are derived from TextonBoost [19, 13]. We use
ank suggested by Shotton et al. [19], and follow Ladický et al. [13] by
oriented gradients (HOG), and pixel location features. Our evaluation

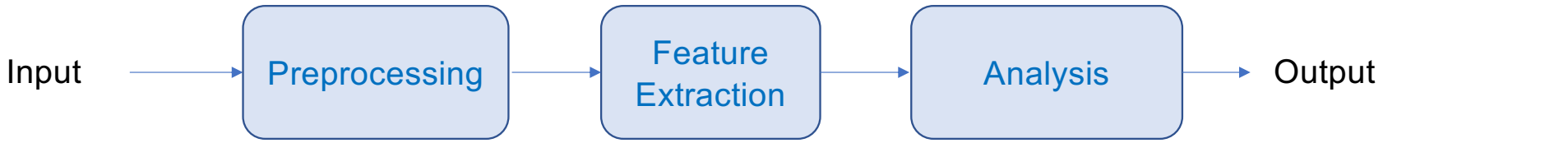
The pairwise potentials in our model have the form

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \underbrace{\sum_{m=1}^{K} w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)},$$

For multi-class image segmentation and labeling we use contrast-sensitive two-kernel potentials, defined in terms of the color vectors I_i and I_j and positions p_i and p_j:

$$k(\mathbf{f}_i, \mathbf{f}_j) = w^{(1)} \underbrace{\exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right)}_{\text{appearance kernel}} + w^{(2)} \underbrace{\exp\left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}\right)}_{\text{smoothness kernel}}. \tag{3}$$

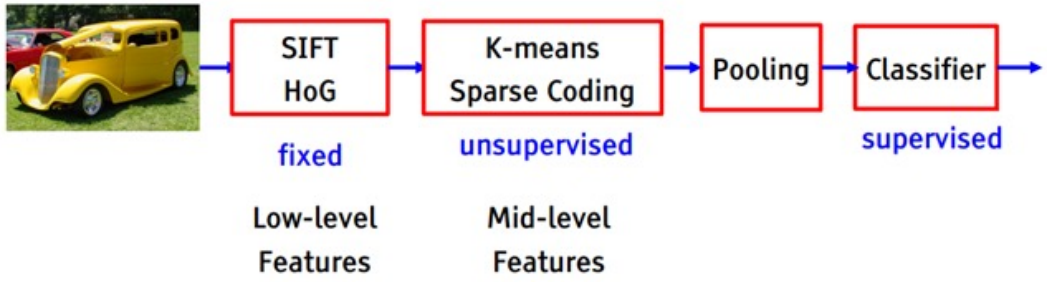
The *appearance kernel* is inspired by the observation that nearby pixels with similar color are likely to be in the same class. The degrees of nearness and similarity are controlled by parameters θ_α and θ_β. The *smoothness kernel* removes small isolated regions [19]. The parameters are learned from data, as described in Section 4.



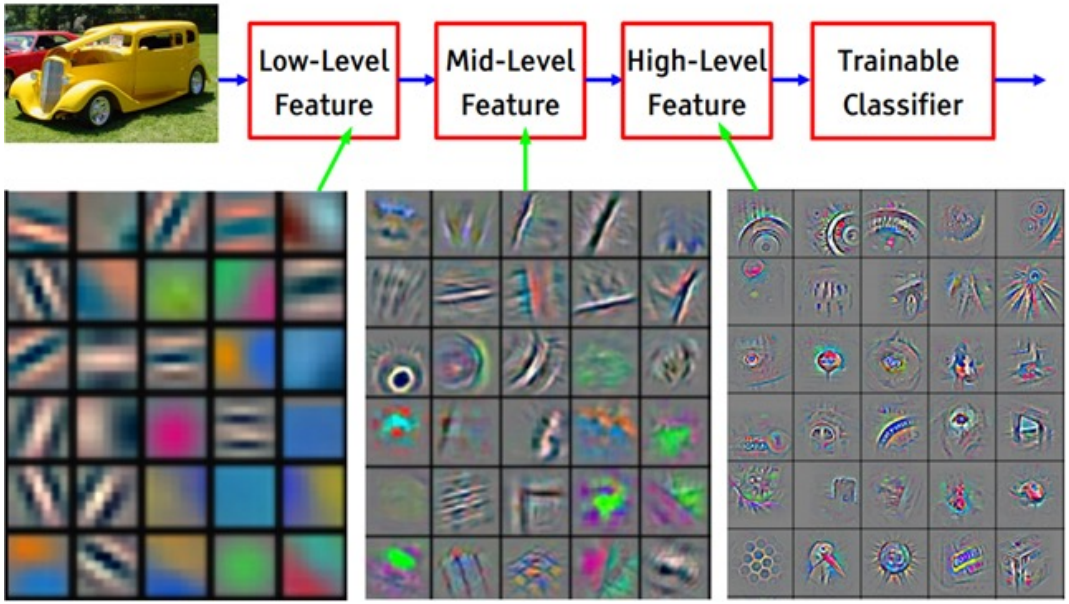
- Noise reduction
- Enhancement
- Restoration
- Segmentation

- Classification/Detection
- Regression

Object recognition 2006-2012



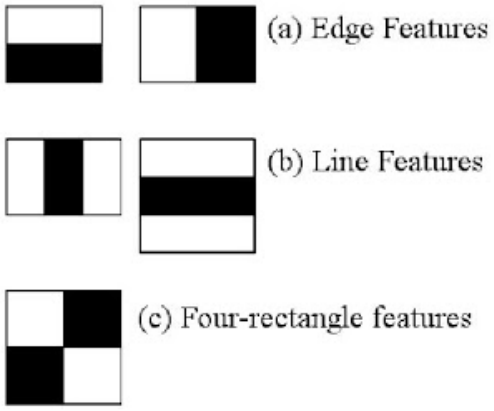
State of the art object recognition using CNNs



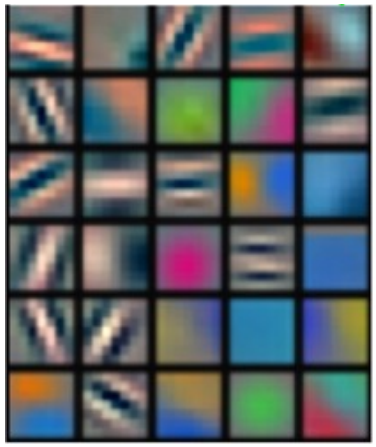
https://distill.pub/2017/feature-visualization/

Feature Engineering vs. Feature Learning

- Could we design an algorithm that can learn features from data automatically?



Manual designed

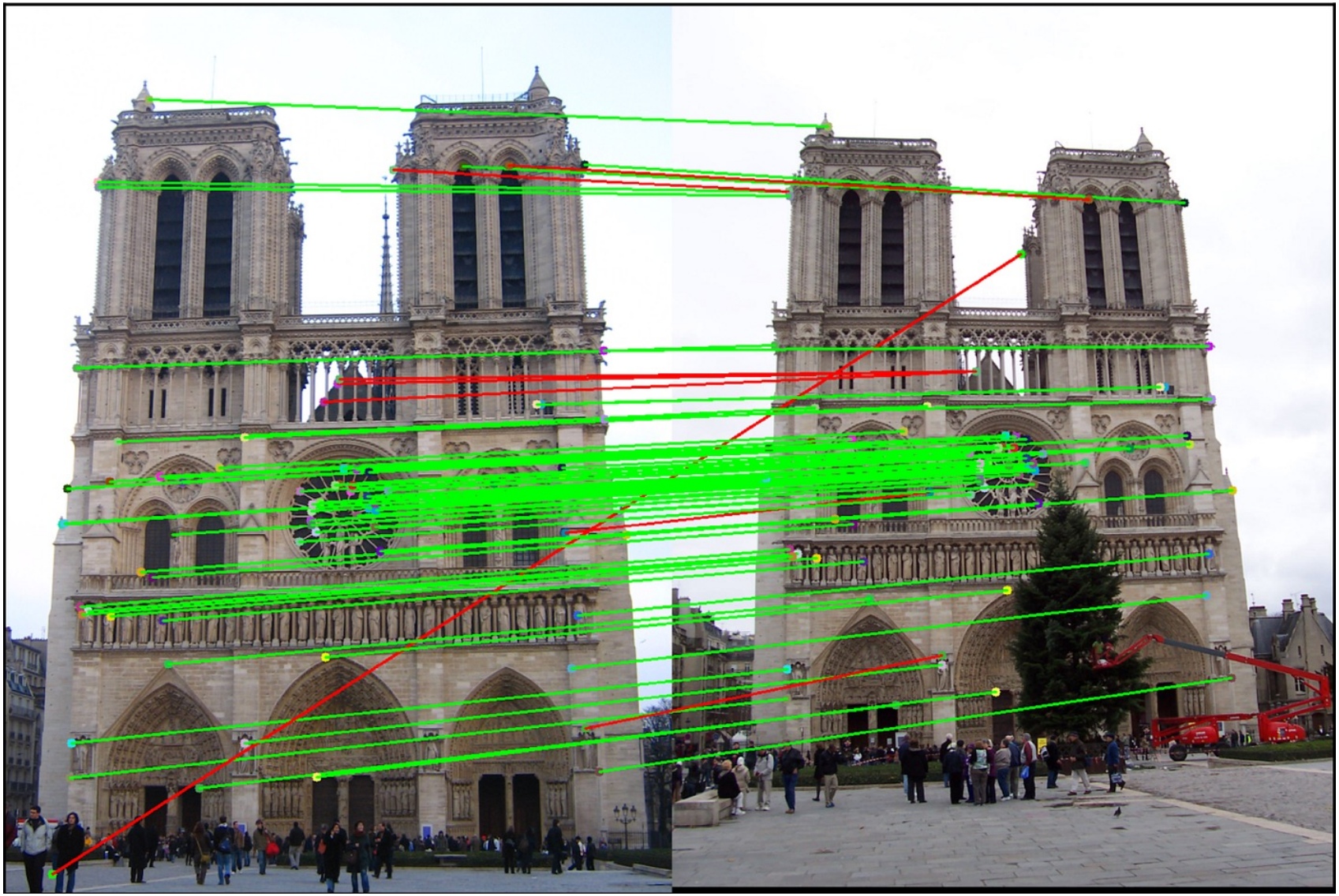


Learned (first few layers)

- We do not know what we will learn in advance
- We let the data, optimization, and task tell us!

Scale-Invariant Feature Transform (SIFT)

- David Lowe, 1999
- References:
  - Lowe, David G. (1999). "Object recognition from local scale-invariant features," *Proceedings of the International Conference on Computer Vision*, Vol. 2, pp. 1150–1157.
  - Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, Vol. 60, Issue 2, pp. 91-110, 2004.
- Invariant to image scaling, translation, and rotation, and partially invariant to illumination changes and viewpoint changes.
- Transform an image into a large collection of local feature vectors.



https://medium.com/@loboateresa/understanding-structure-from-motion-algorithms-fc034875fd0c

COLMAP

Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

Dense models of several landmarks produced by COLMAP's MVS pipeline.

https://colmap.github.io/

File
Processing
Reconstruction
Render
Extras
Help
Model 1
Log
Save
Clear
Time 00:00:00:00
0 Images - 13717405 Points

Scale-Invariant Feature Transform (SIFT)

Scale Space

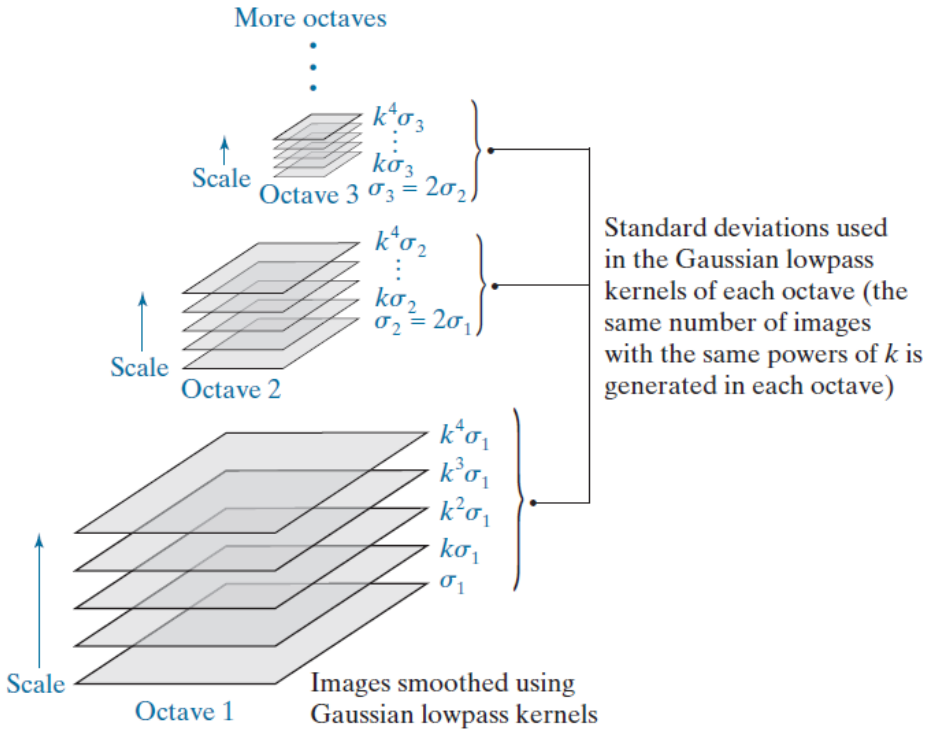
- Search for stable features across all possible scales.
- Scale space represents an image as a one-parameter family of smoothed images.

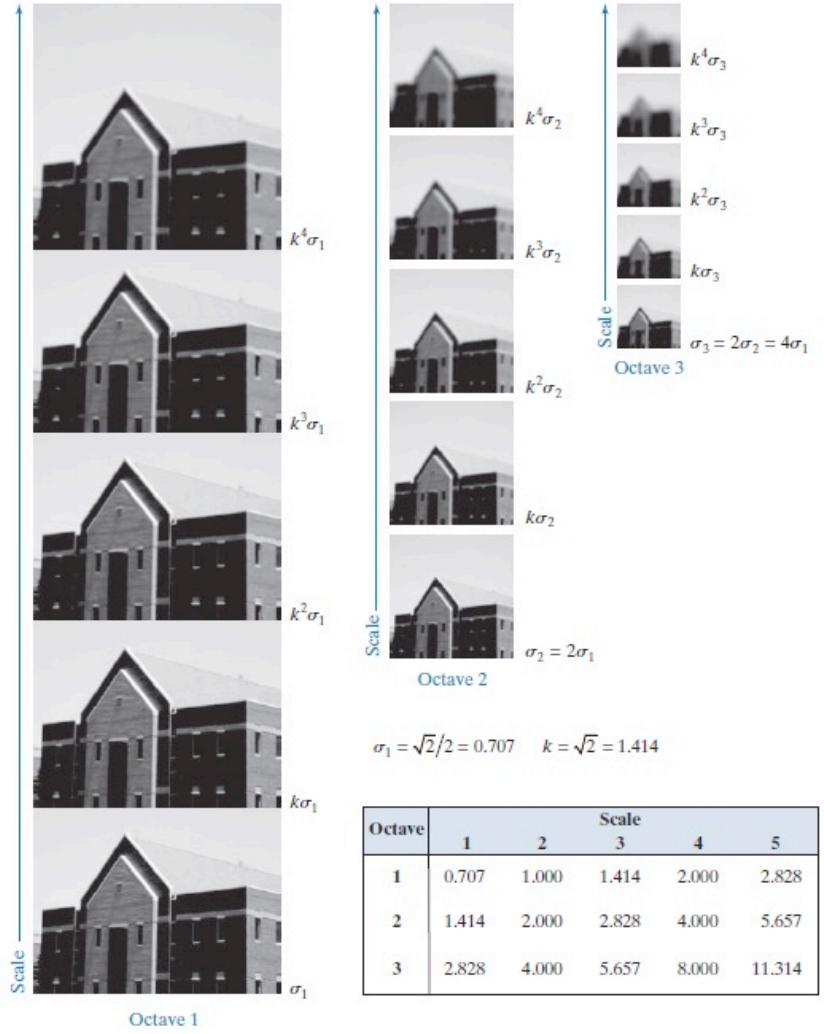
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

where $$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

- The input image is successively convolved with Gaussian kernels having standard deviations $\sigma, k\sigma, k^2\sigma, k^3\sigma, \ldots$
- In SIFT, k is chosen to have the property $k^s\sigma = 2\sigma$; that is, $k = 2^{1/s}$.

$s = 2$





$\sigma_1 = \sqrt{2}/2 = 0.707 \quad k = \sqrt{2} = 1.414$

Octave	Scale				
	1	2	3	4	5
1	0.707	1.000	1.414	2.000	2.828
2	1.414	2.000	2.828	4.000	5.657
3	2.828	4.000	5.657	8.000	11.314

- The first image of a new octave can be obtained directly by down-sampling the third image of the previous octave (named **octave image**) by 2.

Scale-Invariant Feature Transform (SIFT)

Detecting Local Extrema

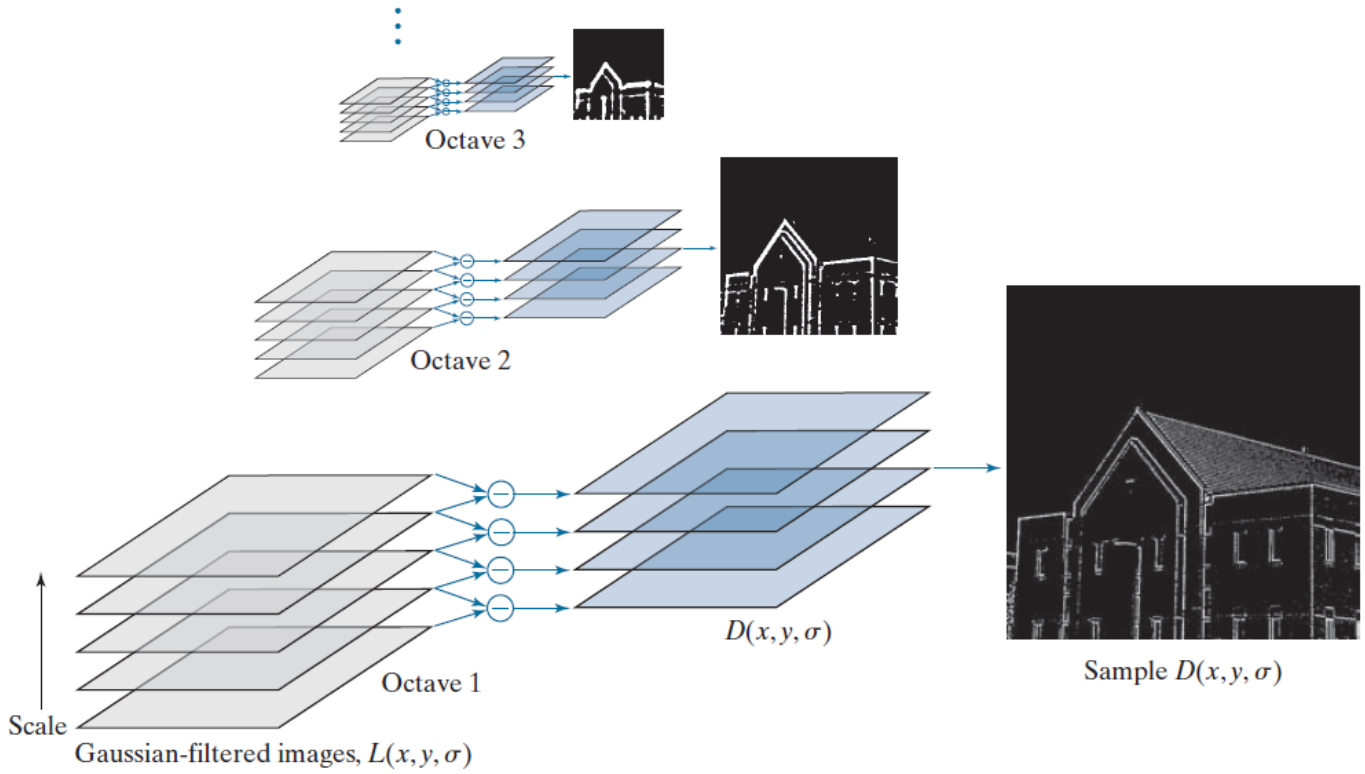
- **Finding the initial keypoints**

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

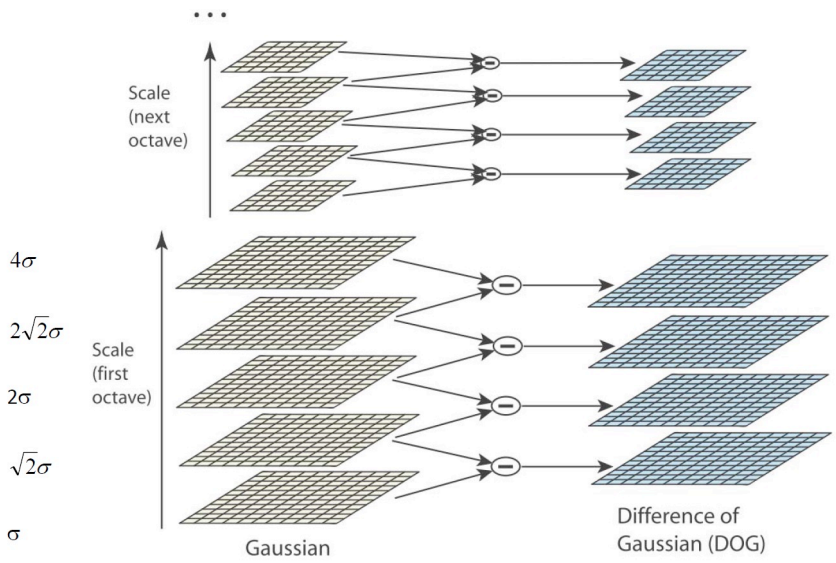
Remarks:

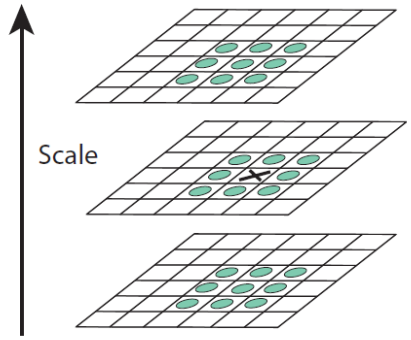
1. The difference of Gaussians (DoG) is an approximation to the Laplacian of a Gaussian (LoG).
2. SIFT looks for key locations at maxima or minima in $D(x, y, \sigma)$.

Scale-Invariant Feature Transform (SIFT)



Scale-Invariant Feature Transform (SIFT)





Local extrema detection

An extreme is selected only if it is larger than all of its 26 neighbors or smaller than all of them.

Scale-Invariant Feature Transform (SIFT)

- **Improving the Accuracy of Keypoint Locations**

  Use Taylor expansion of the scale-space function, $D(x,y,\sigma)$ to find a more accurate location of the keypoint.

$$D(\mathbf{x}) = D + (\frac{\partial D}{\partial \mathbf{x}})^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial D}{\partial \mathbf{x}}\right)\mathbf{x} = D + (\nabla D)^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{H}\mathbf{x}$$

where $\nabla D = \frac{\partial D}{\partial \mathbf{x}} = \begin{bmatrix} \partial D/\partial x \\ \partial D/\partial y \\ \partial D/\partial \sigma \end{bmatrix}$ $\mathbf{H} = \begin{bmatrix} \partial^2 D/\partial x^2 & \partial^2 D/\partial x \partial y & \partial^2 D/\partial x \partial \sigma \\ \partial^2 D/\partial y \partial x & \partial^2 D/\partial y^2 & \partial^2 D/\partial y \partial \sigma \\ \partial^2 D/\partial \sigma \partial x & \partial^2 D/\partial \sigma \partial y & \partial^2 D/\partial \sigma^2 \end{bmatrix}$

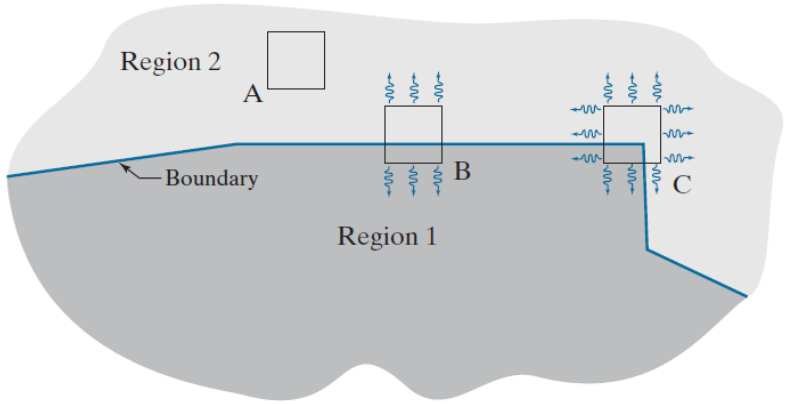
$$\Longrightarrow \hat{\mathbf{x}} = -\mathbf{H}^{-1}(\nabla D) \quad \text{and} \quad D(\hat{\mathbf{x}}) = D + \frac{1}{2}(\nabla D)^T \hat{\mathbf{x}}$$

Scale-Invariant Feature Transform (SIFT)

- **Eliminating edge responses**
  - ✓ Edge points are poorly localized along the edge direction.
  - ✓ Check the principal curvatures at each keypoint based on the Hessian matrix.
  - ✓ Edge points have a large principal curvature across the edge, but a small principal curvature in the perpendicular direction.

$$\mathbf{H} = \begin{bmatrix} \partial^2 D/\partial x^2 & \partial^2 D/\partial x \partial y \\ \partial^2 D/\partial y \partial x & \partial^2 D/\partial y^2 \end{bmatrix}$$

Harris-Stephens Corner Detector



$$C(x,y) = \sum_s \sum_t w(s,t)[f(s+x.t+y) - f(s,t)]^2$$

Weighted sum of squared differences between an image patch and its shifted version

Harris-Stephens Corner Detector

$$f(s+x,t+y) \approx f(s,t) + xf_x(s,t) + yf_y(s,t)$$

$$f_x(s,t) = \frac{\partial f}{\partial x} \qquad f_y(s,t) = \frac{\partial f}{\partial y}$$

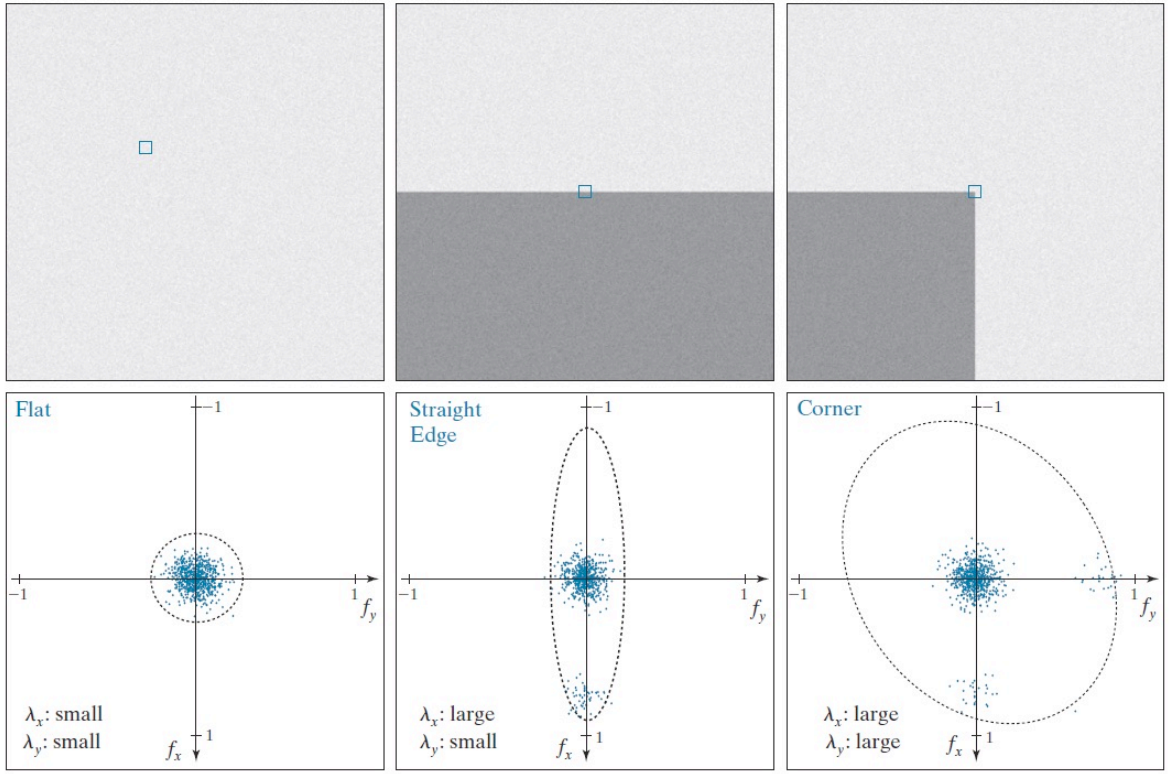
$$\Rightarrow \quad C(x,y) = \sum_s \sum_t w(s,t)\left[xf_x(s,t) + yf_y(s,t)\right]^2 = \begin{bmatrix} x & y \end{bmatrix} \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix}$$

where $\mathbf{M} = \sum_s \sum_t w(s,t)\mathbf{A}$ and $A = \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}$

Harris matrix

Generally, we define $w(s,t) = \begin{cases} 1 & \text{inside the patch} \\ 0 & \text{outside the patch} \end{cases}$ or $w(s,t) = e^{-\frac{(s^2+t^2)}{2\sigma^2}}$.

Principle Component Analysis to Harris Matrix



Scale-Invariant Feature Transform (SIFT)

- **Eliminating edge responses**
  - ✓ Edge points are poorly localized along the edge direction.
  - ✓ Check the principal curvatures at each keypoint based on the Hessian matrix.
  - ✓ Edge points have a large principal curvature across the edge, but a small principal curvature in the perpendicular direction.

$$\mathbf{H} = \begin{bmatrix} \partial^2 D/\partial x^2 & \partial^2 D/\partial x \partial y \\ \partial^2 D/\partial y \partial x & \partial^2 D/\partial y^2 \end{bmatrix}$$

The eigenvalues of **H** are proportional to the curvatures of D.

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta$$
$$Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

α: the larger eigenvalue
β: the smaller eigenvalue

Scale-Invariant Feature Transform (SIFT)

Let $r = \frac{\alpha}{\beta}$

$$\frac{[Tr(\mathbf{H})]^2}{Det(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$

- The minimum of $\frac{(r+1)^2}{r}$ occurs when the eigenvalues are equal, and it increases with r.

$\Rightarrow$ Choose an *r* and remove those keypoints at which $\frac{[Tr(\mathbf{H})]^2}{Det(\mathbf{H})} > \frac{(r+1)^2}{r}$

$$D(\hat{\mathbf{x}}) \geq 0.03$$
$$r = 10$$

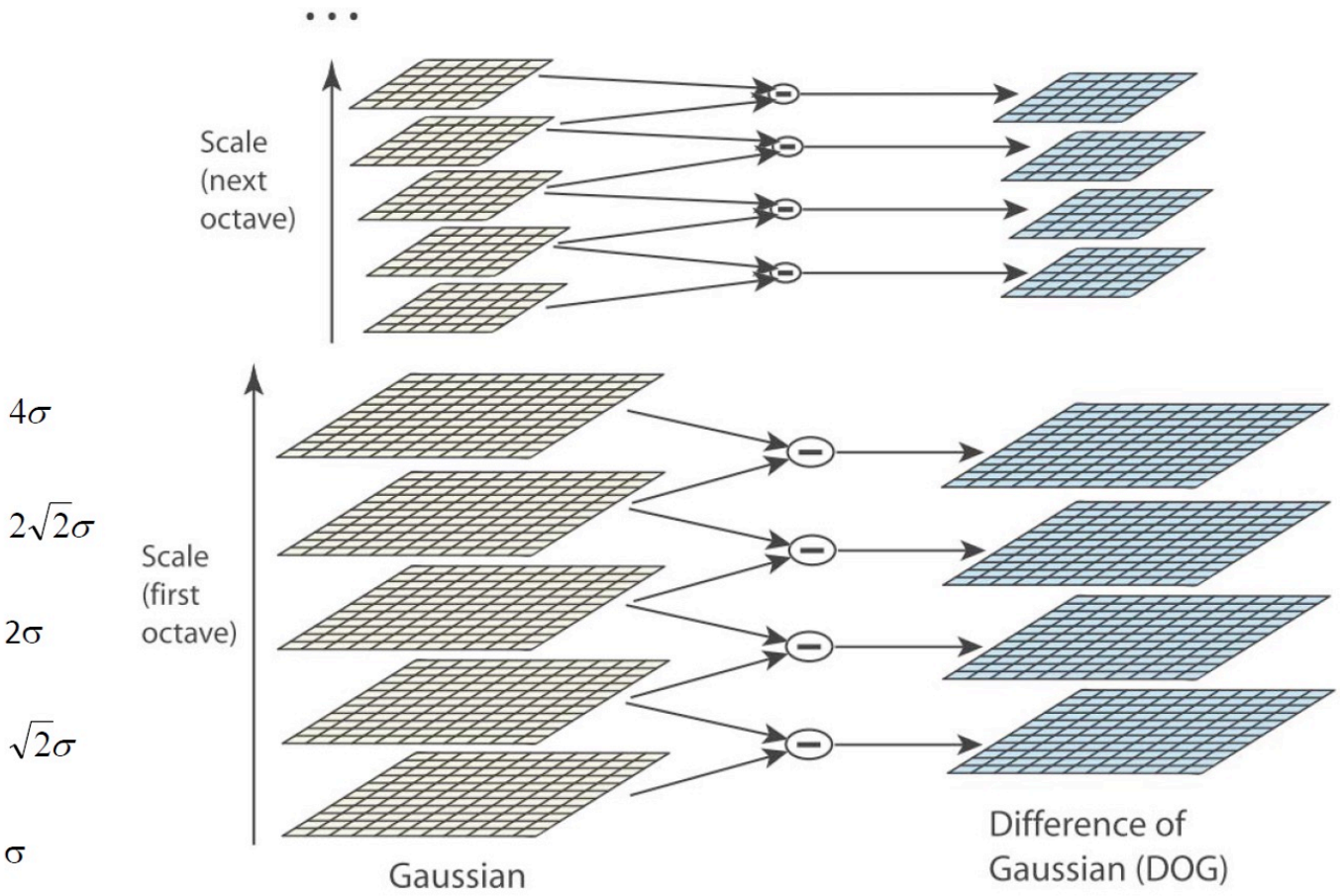
Scale-Invariant Feature Transform (SIFT)

- **Keypoint orientation**
  - Select the Gaussian smoothed image, L, of the closest scale.
    For each image sample, $L(x, y)$, at this scale, we compute the gradient magnitude, $M(x, y)$, and orientation angle, $\theta(x, y)$.

$$M(x,y) = [\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2]^{\frac{1}{2}}$$

$$\theta(x,y) = tan^{-1}[\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}]$$

  - An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint.
    - ✓ 36 bins.
    - ✓ Each sample is weighted by its gradient magnitude and by a Gaussian-weighted circular window with a σ that is 1.5 times that of the closest scale.

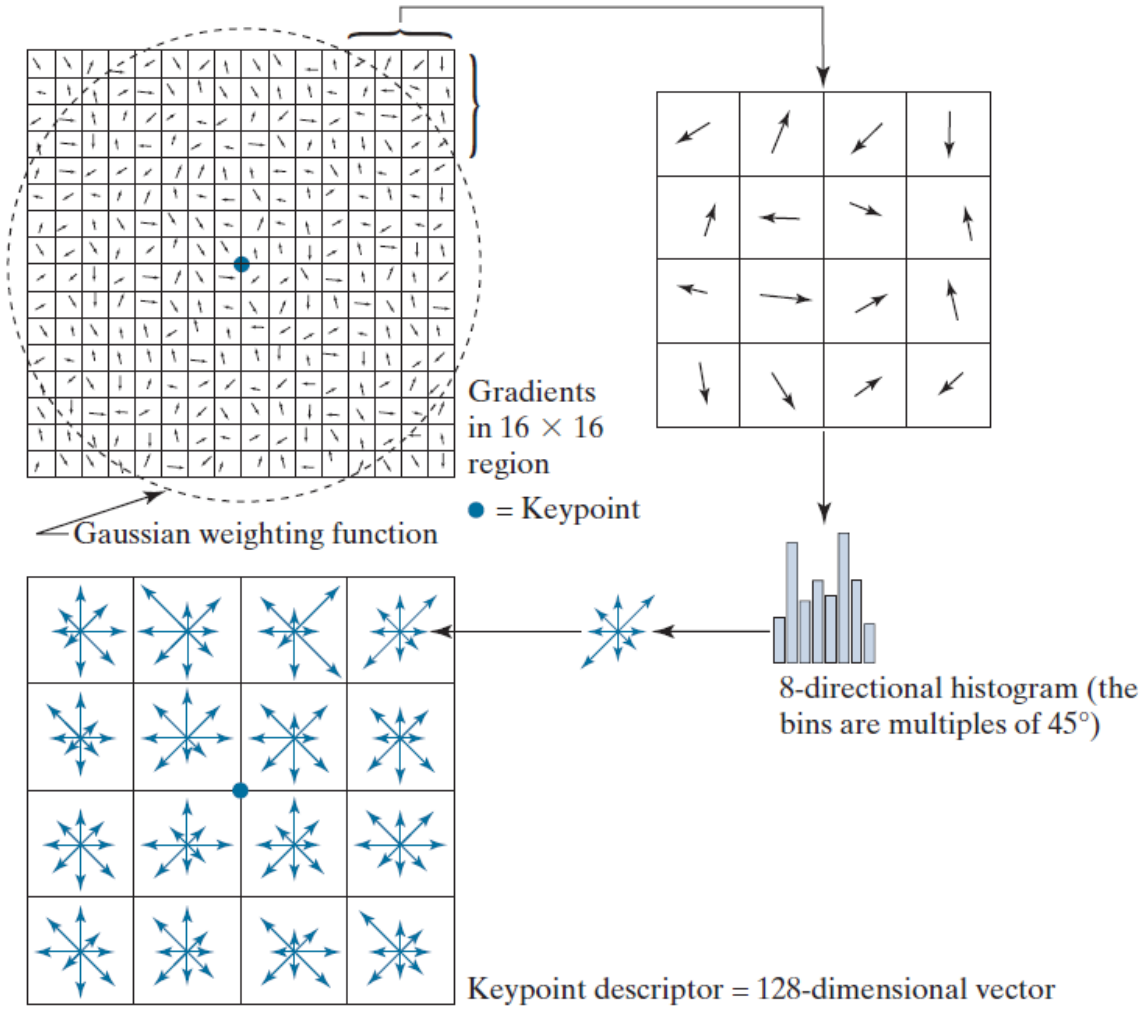

Scale
(next
octave)
4σ
$2\sqrt{2}\sigma$
2σ
$\sqrt{2}\sigma$
σ
Scale
(first
octave)
Gaussian
Difference of
Gaussian (DOG)

Scale-Invariant Feature Transform (SIFT)

- **Keypoint orientation**
  - Peaks in the orientation histogram correspond to dominant directions of local gradients.
  - Any other local peak that is within 80% of the highest peak is used to create another keypoint.

Scale-Invariant Feature Transform (SIFT)

- **Keypoint Descriptors**
  - 128 dimensions in total
    - ✓ Based on 16×16 patches
    - ✓ 4×4 subregions
    - ✓ 8 bins in each subregion
  - The coordinates and the gradient orientations are rotated relative to the keypoint orientation.


Gradients
in 16 × 16
region
● = Keypoint
Gaussian weighting function
8-directional histogram (the
bins are multiples of 45°)
Keypoint descriptor = 128-dimensional vector

643 keypoints

54 keypoints

Rotate

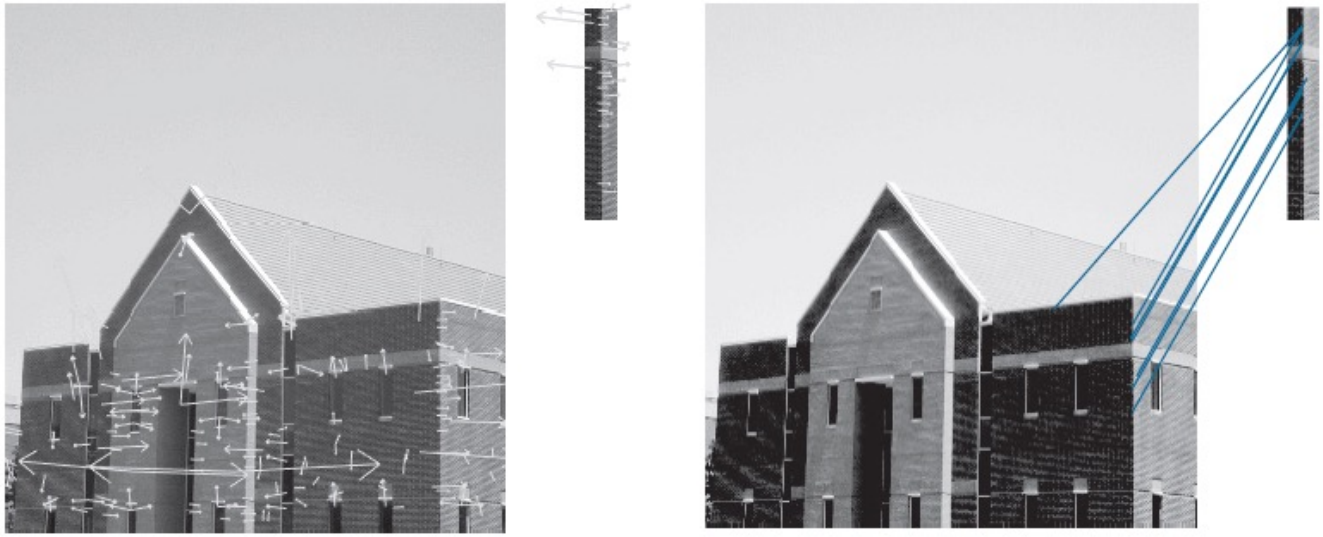
547 keypoints

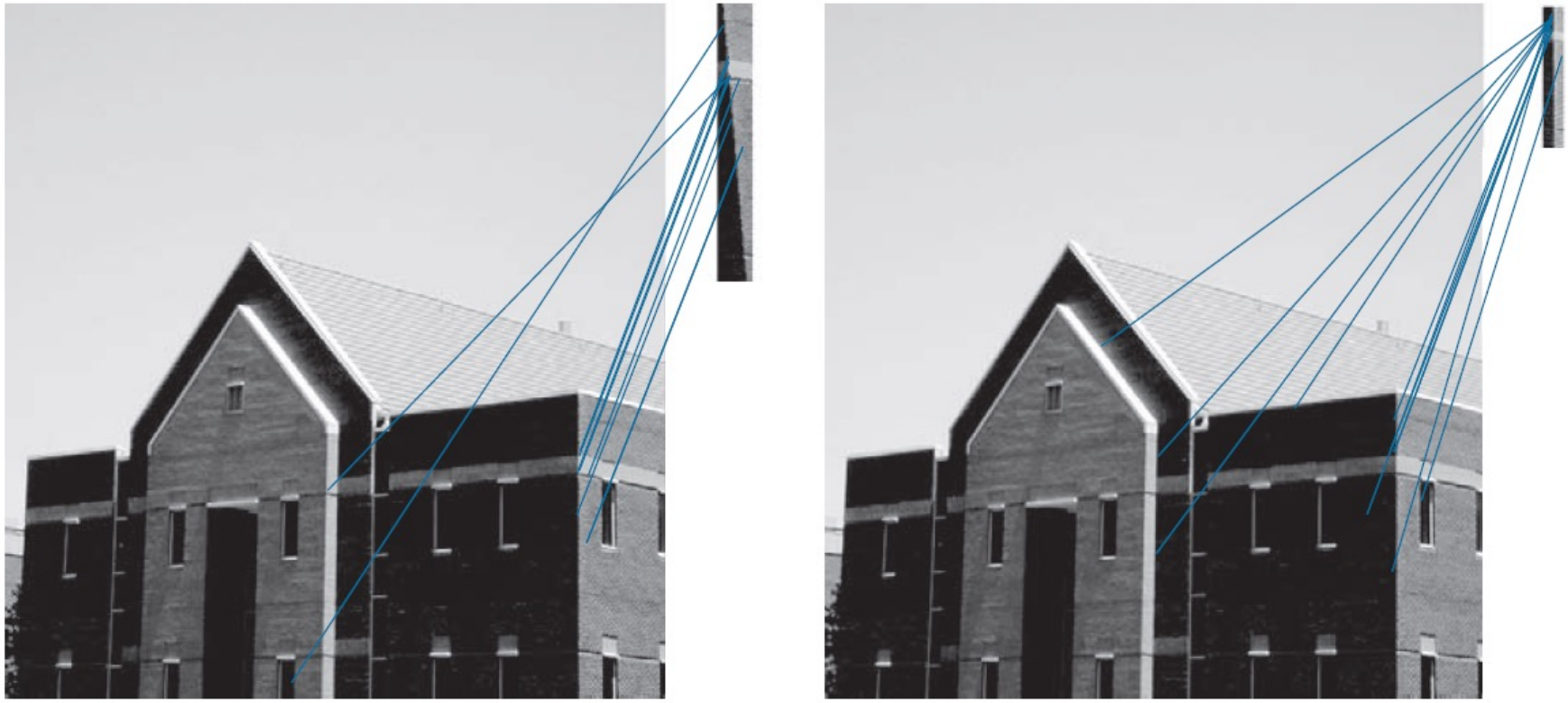
49 keypoints

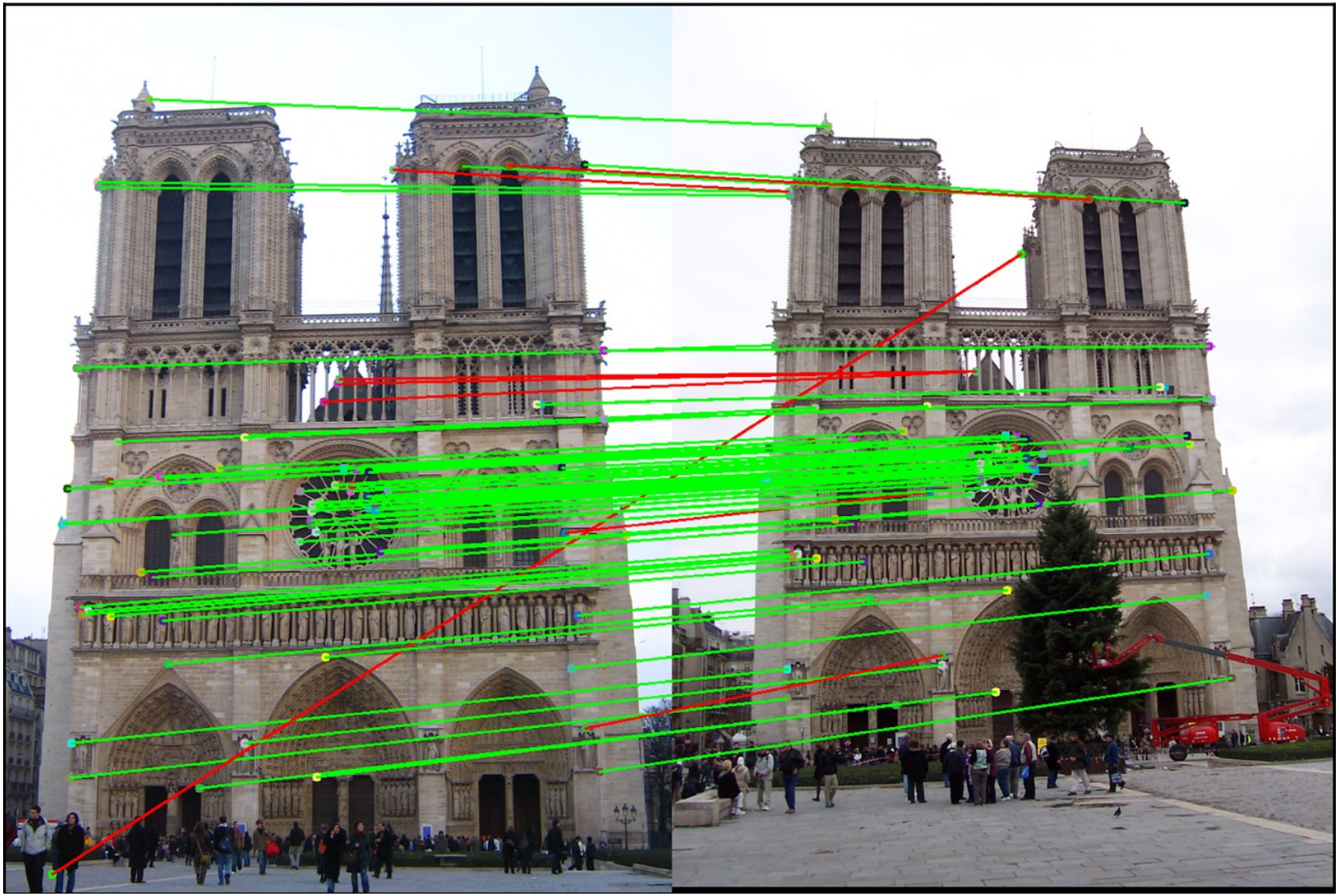
Scale

195 keypoints

24 keypoints







https://medium.com/@loboateresa/understanding-structure-from-motion-algorithms-fc034875fd0c

Student: "What are the three most important problems in computer vision?"
Takeo Kanade: "Correspondence, correspondence, correspondence!"

Digital Image Processing

Takeo Kanade

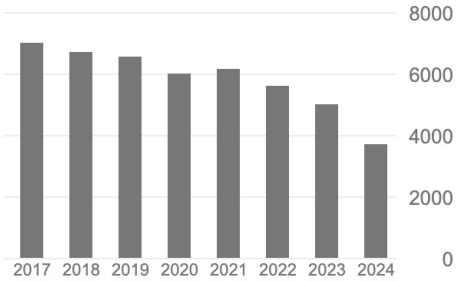
FOLLOW

Carnegie Mellon University
Verified email at cs.cmu.edu - Homepage
Computer Vision

TITLE	CITED BY	YEAR
An iterative image registration technique with an application to stereo vision BD Lucas, T Kanade IJCAI'81: 7th international joint conference on Artificial intelligence 2 …	20074	1981
Neural network-based face detection HA Rowley, S Baluja, T Kanade IEEE Transactions on pattern analysis and machine intelligence 20 (1), 23-38	6447	1998
The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression P Lucey, JF Cohn, T Kanade, J Saragih, Z Ambadar, I Matthews 2010 ieee computer society conference on computer vision and pattern …	5216	2010
Shape and motion from image streams under orthography: a factorization method C Tomasi, T Kanade International journal of computer vision 9, 137-154	4381	1992
Detection and tracking of point C Tomasi, T Kanade Int J Comput Vis 9 (137-154), 3	4119	1991

Cited by VIEW ALL

	All	Since 2019
Citations	154270	33115
h-index	172	72
i10-index	649	293



Public access VIEW ALL

0 articles 22 articles

not available available

Based on funding mandates

https://www.books.com.tw/products/0010636964

Lukas-Kanede Method: Optical Flow

Takeo Kanade

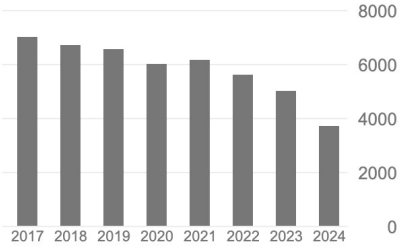
FOLLOW

Carnegie Mellon University
Verified email at cs.cmu.edu - Homepage
Computer Vision

TITLE	CITED BY	YEAR
An iterative image registration technique with an application to stereo vision BD Lucas, T Kanade IJCAI'81: 7th international joint conference on Artificial intelligence 2 …	20074	1981
Neural network-based face detection HA Rowley, S Baluja, T Kanade IEEE Transactions on pattern analysis and machine intelligence 20 (1), 23-38	6447	1998
The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression P Lucey, JF Cohn, T Kanade, J Saragih, Z Ambadar, I Matthews 2010 ieee computer society conference on computer vision and pattern …	5216	2010
Shape and motion from image streams under orthography: a factorization method C Tomasi, T Kanade International journal of computer vision 9, 137-154	4381	1992
Detection and tracking of point C Tomasi, T Kanade Int J Comput Vis 9 (137-154), 3	4119	1991

Cited by VIEW ALL

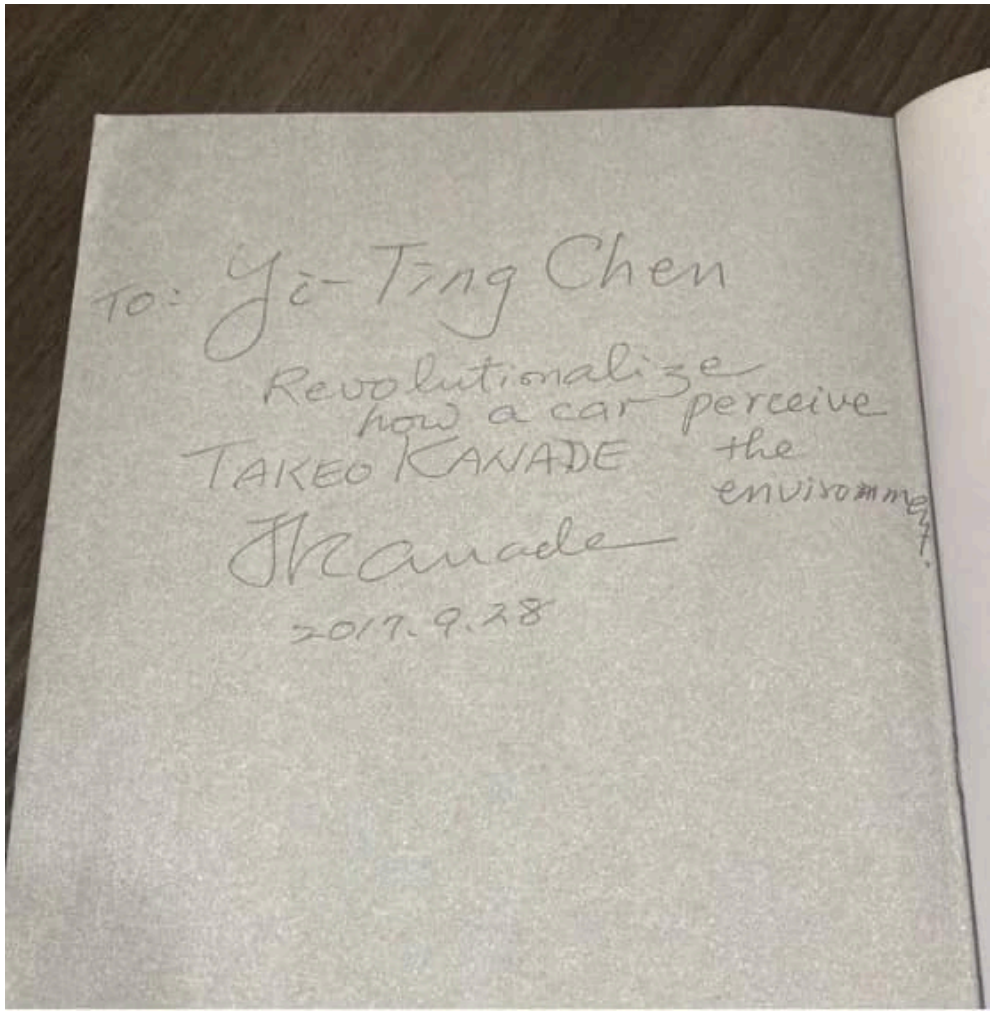
	All	Since 2019
Citations	154270	33115
h-index	172	72
i10-index	649	293



Public access VIEW ALL

0 articles	22 articles
not available	available

Based on funding mandates


To: Yi-Ting Chen
Revolutionalize how a car perceive the environment.
TAKEO KANADE
2017.9.28

Prof. Takeo Kanade

https://youtu.be/iRwzoJyBGBc?si=h24IJqYkJzsncUZQ

Eyevision

https://youtu.be/D7r3-fHXvRU?si=QhCGnm3P7mM7T4Uj&t=404

CMU Dome

https://www.wired.com/video/watch/go-inside-the-dome-that-could-give-robots-super-senses