Segmentation

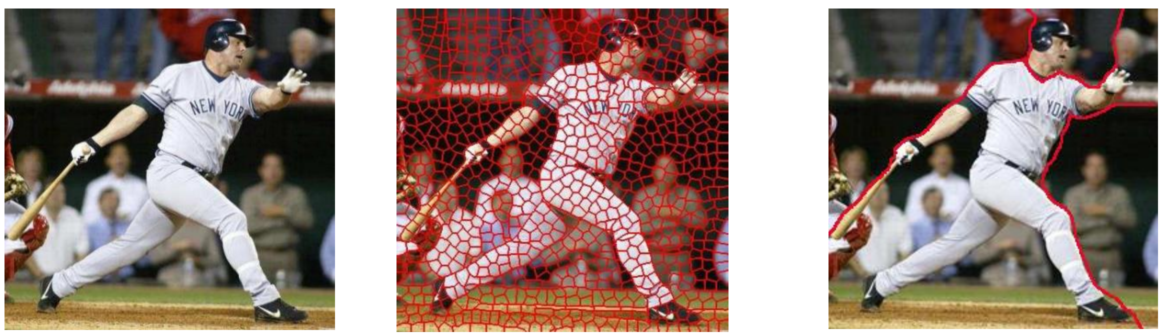
簡韶逸 Shao-Yi Chien
Department of Electrical Engineering
National Taiwan University

Outline

- Segmentation
- Image segmentation
  - Object selection with interactive segmentation
  - Super-pixel methods
  - Semantic segmentation
- Video segmentation
  - Segmentation in motion field
  - Change detection method

Segmentation

- Group pixels that share similar attributes in perception into regions
  - Over-segmentation v.s. under-segmentation



- Used as pre-processing or post-processing
- Select region-of-interest (ROI) in an image/video with/without users' inputs (ex. stroke)

What We Will Introduce Today

Image Segmentation

Object Selection

Super-pixel

Semantic Segmentation

Video Segmentation

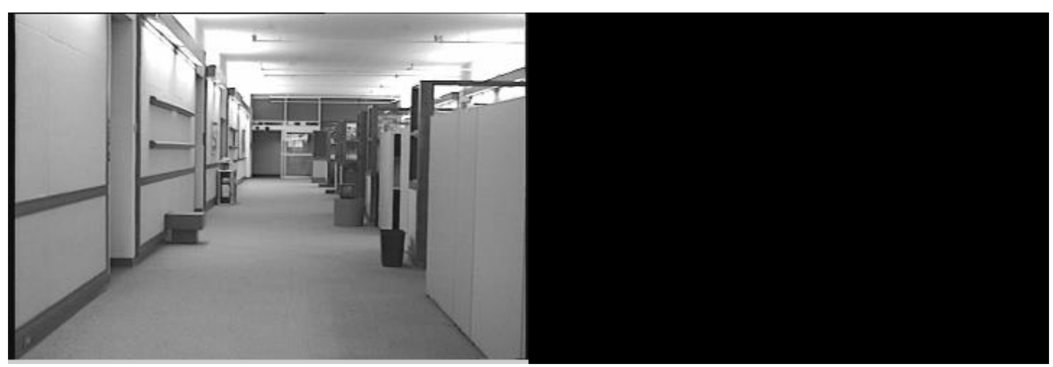


Image Segmentation: Object Selection with Interactive Segmentation

- Select region-of-interest (ROI) in an image/video with users' help

- Active contour
- Graphcut/Grabcut
- Deep interactive object selection

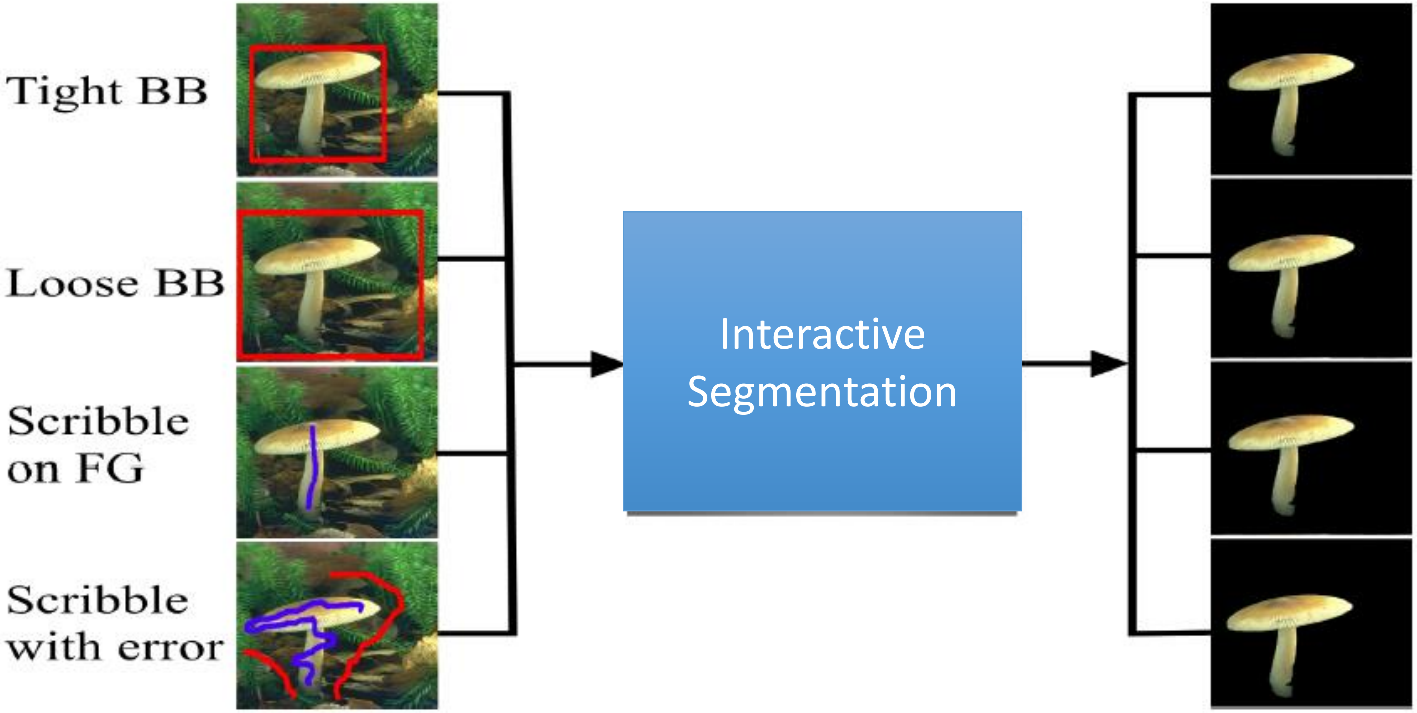
Where is the Foreground?

- Determining foreground objects is subjective
  - All people and horses, or…
  - The person in the middle

User interaction is required!

The Form of User Input

- Some examples



The Form of User Input

- Clicks

Active Contour

- To minimize the total energy of an active contour

$$\boxed{\mathcal{E}_{int}} + \boxed{\mathcal{E}_{ext}}$$

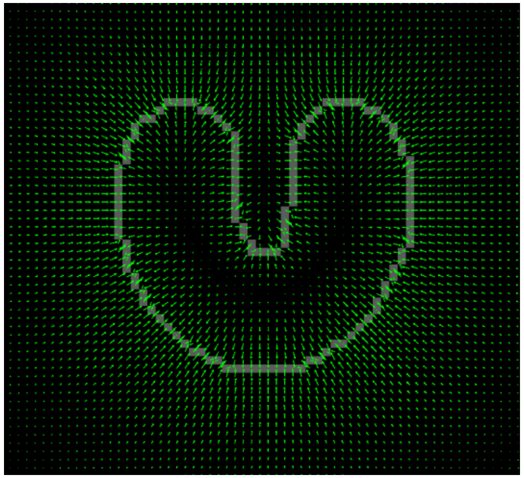
$$\mathcal{E}_{\mathrm{int}} = \int \alpha(s)\|\boldsymbol{f}_s(s)\|^2 + \beta(s)\|\boldsymbol{f}_{ss}(s)\|^2\, ds$$

$$\begin{aligned} E_{\mathrm{int}} &= \sum_i \alpha(i)\|f(i+1) - f(i)\|^2/h^2 \\ &\quad + \beta(i)\|f(i+1) - 2f(i) + f(i-1)\|^2/h^4 \end{aligned}$$

$$\mathcal{E}_{\mathrm{image}} = w_{\mathrm{line}}\mathcal{E}_{\mathrm{line}} + w_{\mathrm{edge}}\mathcal{E}_{\mathrm{edge}} + w_{\mathrm{term}}\mathcal{E}_{\mathrm{term}}$$

$$E_{\mathrm{edge}} = \sum_i -\|\nabla I(\boldsymbol{f}(i))\|^2$$

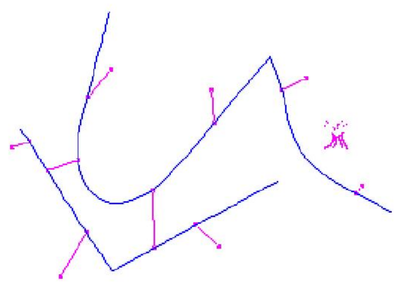
$$E_{\mathrm{spring}} = k_i\|\boldsymbol{f}(i) - \boldsymbol{d}(i)\|^2$$



[Kass, Witkin, Terzopoulos IJCV1988]

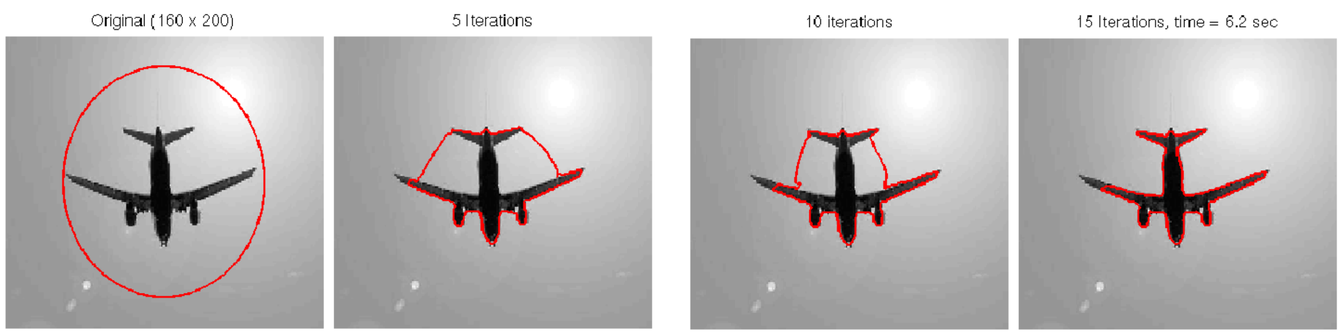
Active Contour

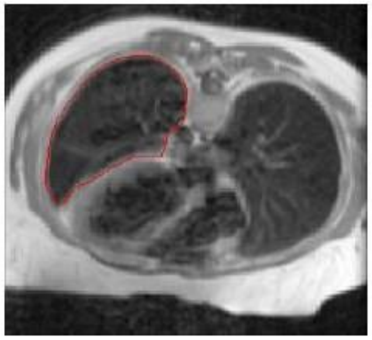
- To minimize the total energy of an active contour



(a)

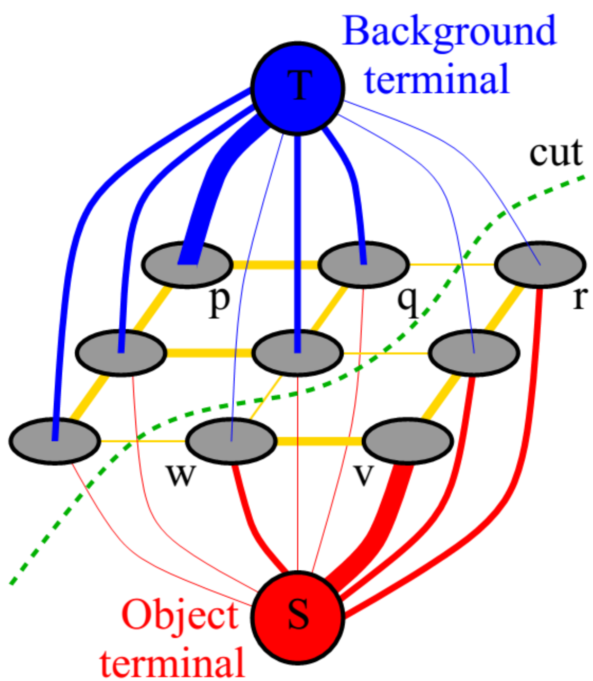
(b)





Graphcut

- Formulate the problem as a Markov-Random-Field (MRF)



$$E(A) \;=\; \lambda \cdot R(A) + B(A)$$

Region Properties Term (Data Term)

$$R(A) \;=\; \sum_{p \in \mathcal{P}} R_p(A_p)$$

Boundary Properties Term (Smooth Term)

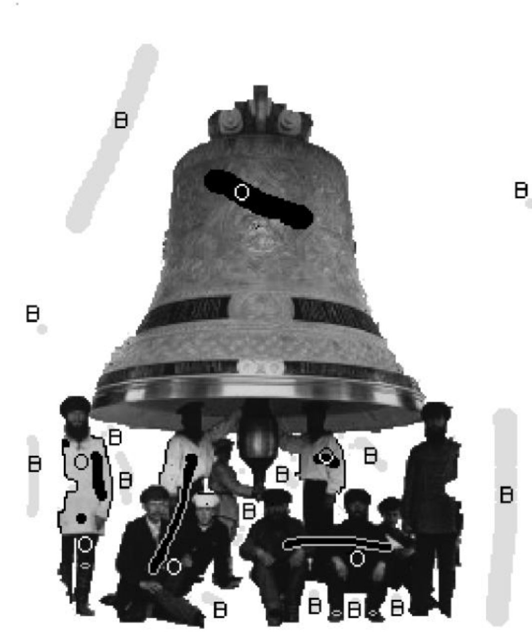
$$B(A) \;=\; \sum_{\{p,q\} \in \mathcal{N}} B_{\{p,q\}} \cdot \delta(A_p, A_q)$$

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{otherwise.} \end{cases}$$

[Boykov and Jolly ICCV 2001]

Graphcut

- An example



$$R_p(\text{“obj”}) = -\ln \Pr(I_p|\mathcal{O})$$
$$R_p(\text{“bkg”}) = -\ln \Pr(I_p|\mathcal{B})$$

Can be modeled by histogram

$$B_{\{p,q\}} \propto exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)}$$

[Boykov and Jolly ICCV 2001]

GrabCut

[Rother, Kolmogorov, Blake SIGGRAPH 2004]

1. Define graph
   - usually 4-connected or 8-connected
     - Divide diagonal potentials by sqrt(2)
2. Define unary potentials
   - Color histogram or mixture of Gaussians for background and foreground

$$unary\_potential(x) = -\log\left(\frac{P(c(x);\theta_{foreground})}{P(c(x);\theta_{background})}\right)$$

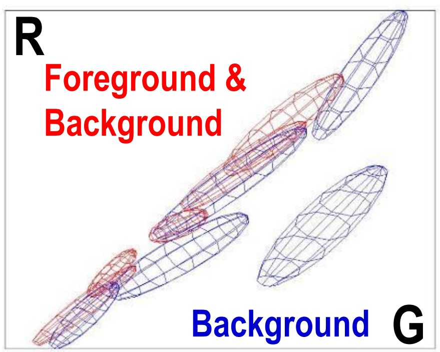
3. Define pairwise potentials

$$edge\_potential(x,y) = k_1 + k_2 \exp\left\{\frac{-\left\|c(x)-c(y)\right\|^2}{2\sigma^2}\right\}$$

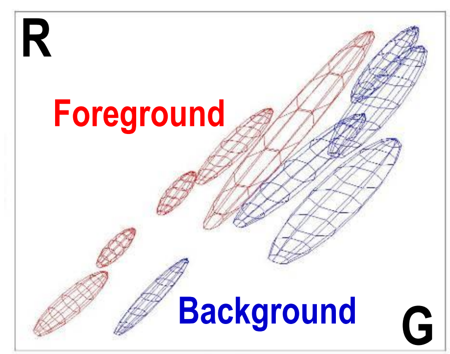
4. Apply graph cuts
5. Return to 2, using current labels to compute foreground, background models

GrabCut

- Color model



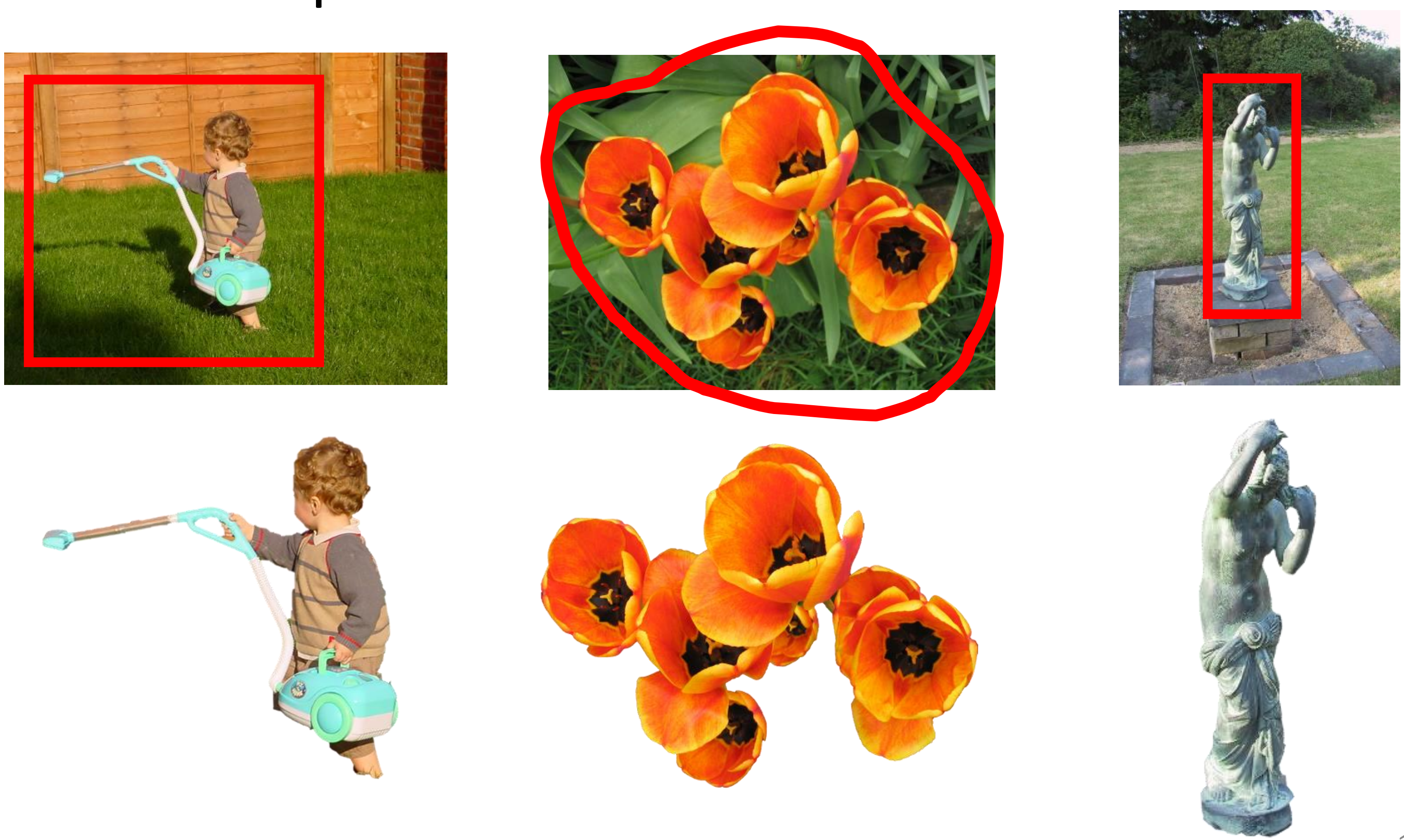
Iterated
graph cut



Gaussian Mixture Model (typically 5-8 components)

GrabCut

- Easier examples



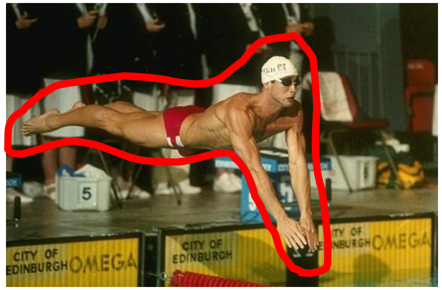
GrabCut

- More difficult examples

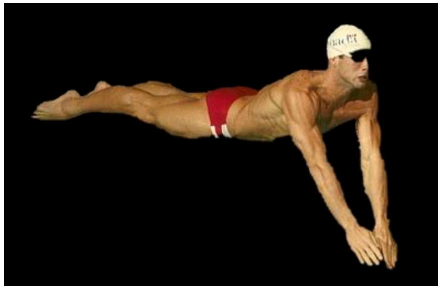
Fine structure

Harder Case

Initial Rectangle

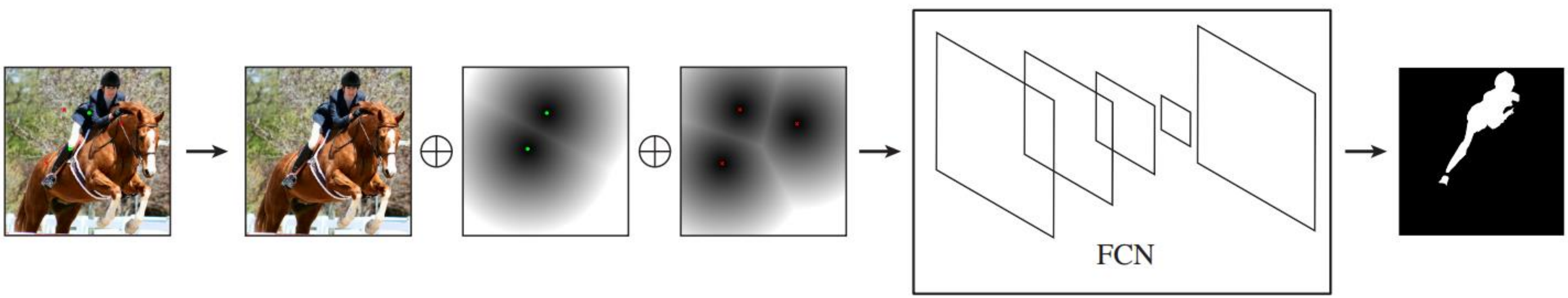


Initial Result



Deep Interactive Segmentation

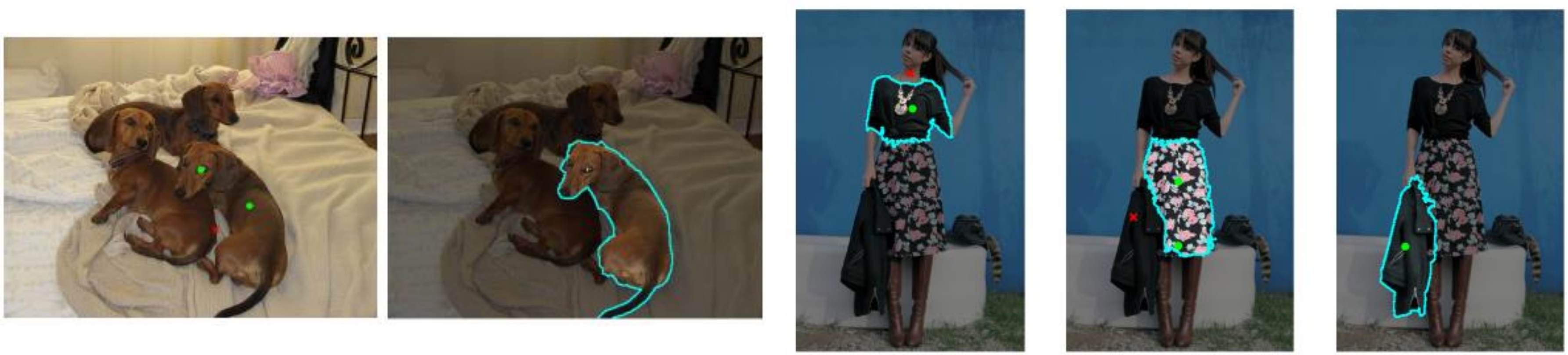
- FCN model
- User clicks are transformed into distance maps
- Input color image and the user clicks are cascaded as 5D input features



Ref: Ning Xu, Brian Price, Scott Cohen, Jimei Yang, Thomas Huang. Deep Interactive Object Selection. In *CVPR* 2016

Deep Interactive Segmentation

- Select different instances
- Select different parts



Ref: Ning Xu, Brian Price, Scott Cohen, Jimei Yang, Thomas Huang. Deep Interactive Object Selection. In *CVPR* 2016

Deep Interactive Segmentation

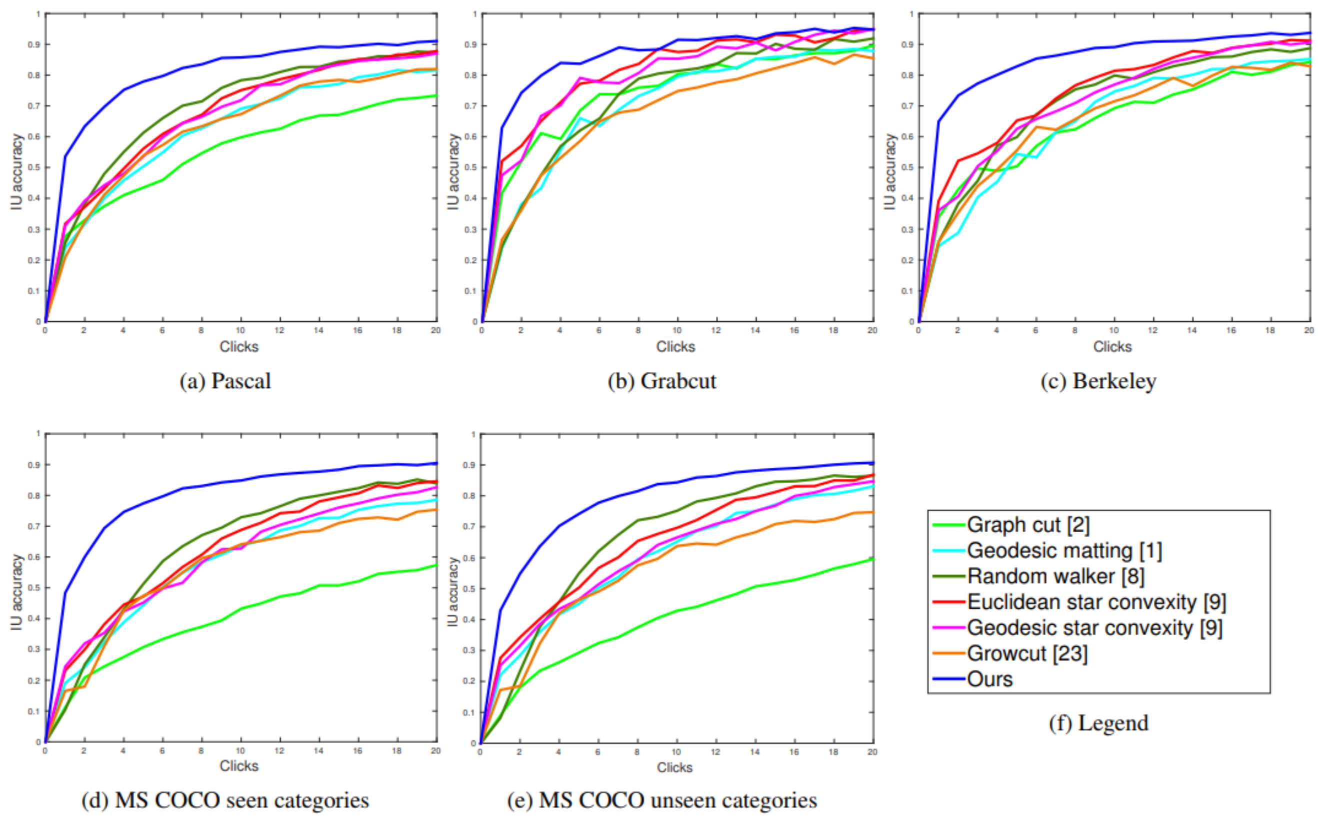
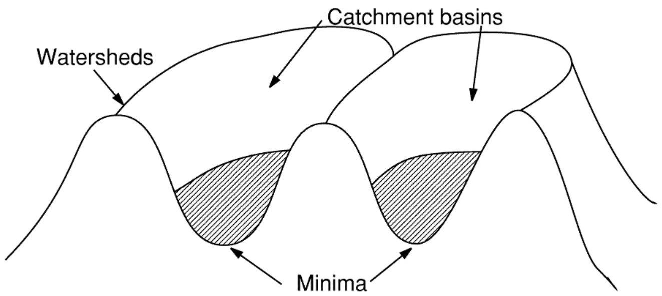


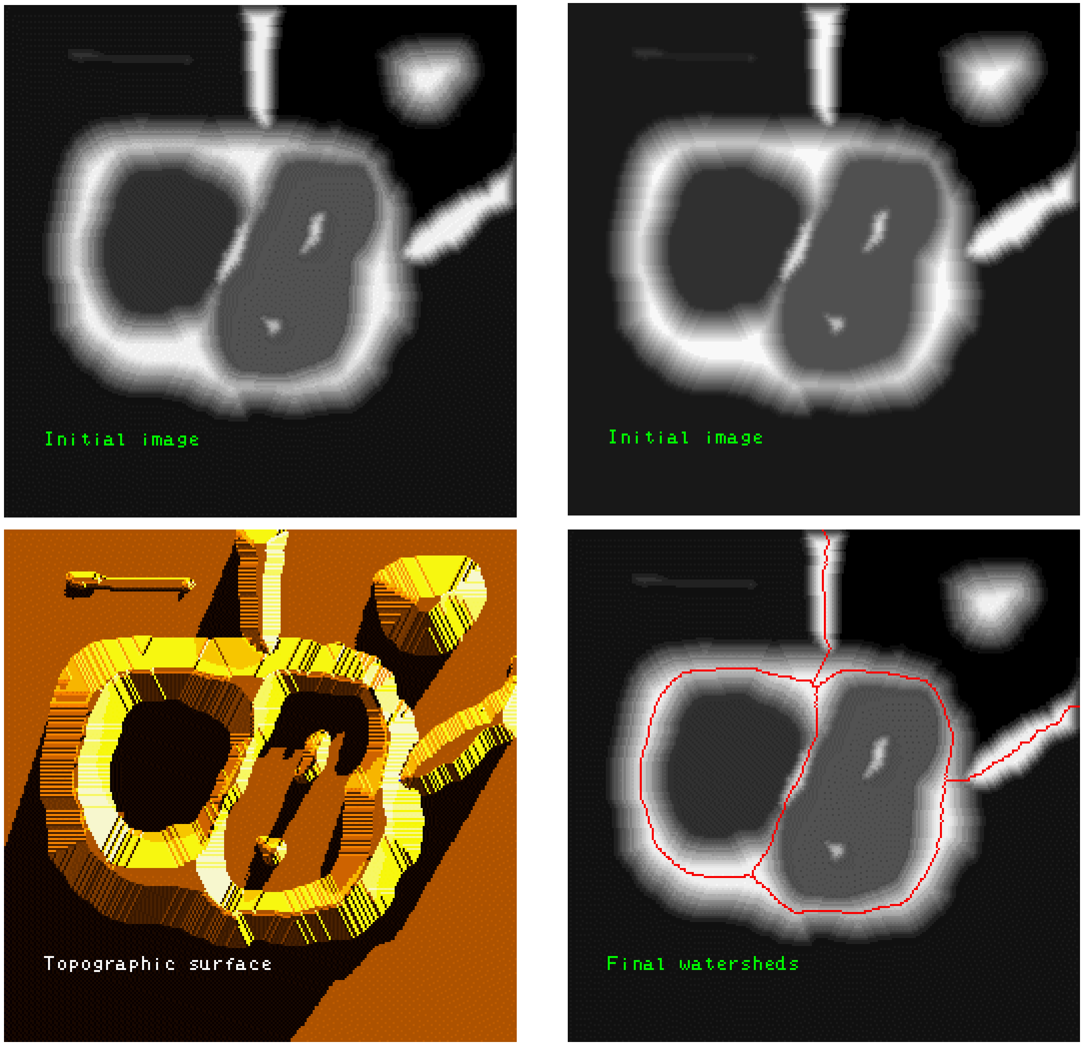
Image Segmentation: Superpixel

- Superpixels are grouping of pixels (over-segmentation)

- Watershed
- K-means
- Mean-shift
- Modern superpixel

Watershed



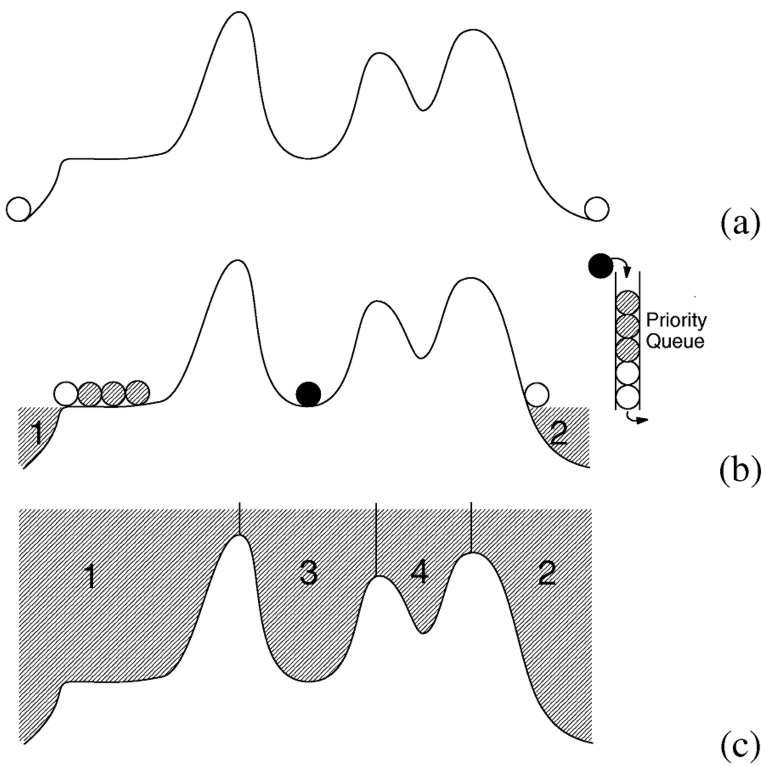


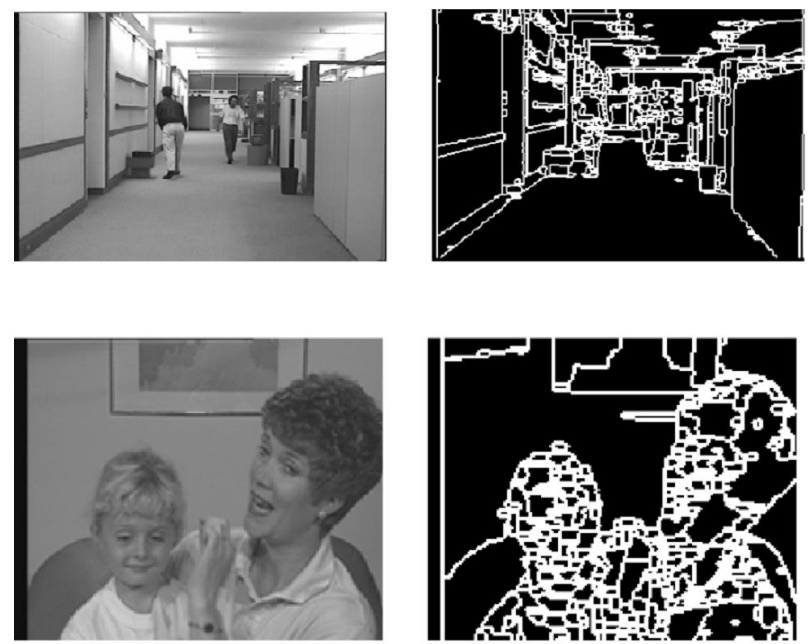
[Vincent and P. Soille PAMI91]

http://cmm.ensmp.fr/~beucher/wtshed.html

Watershed

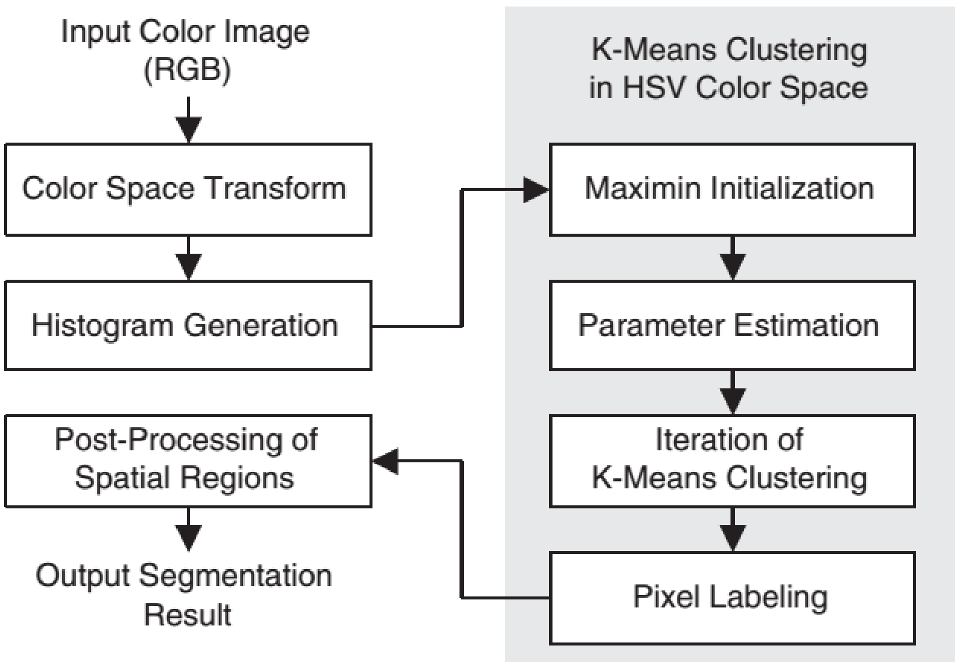
- Can be implemented efficiently





Ref: S.-Y. Chien, Y.-W. Huang, and L.-G. Chen, "Predictive Watershed: A Fast Watershed Algorithm for Video Segmentation," *IEEE T. Circuits and Systems for Video Technology*, 2003.

K-means



- K-means in HSV color space
- The H term should be handled carefully

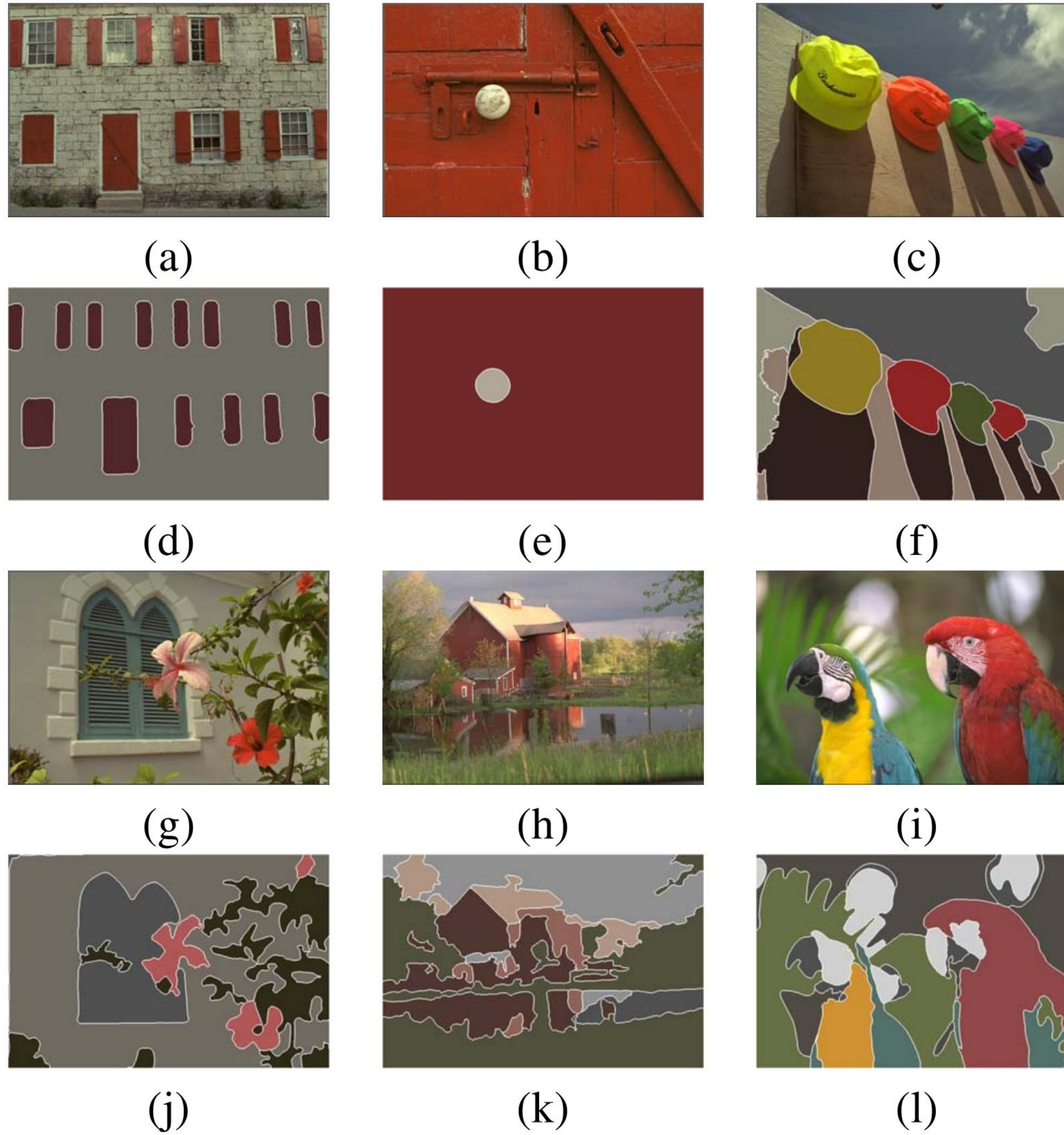
$$D^2(\boldsymbol{B}_i, \boldsymbol{C}_j^{(t)}) = D_h^2(h_i, h_j^{(t)}) + (s_i - s_j^{(t)})^2 + (v_i - v_j^{(t)})^2,$$

where

$$D_h^2(h_i, h_j^{(t)}) = \begin{cases} (\frac{360^\circ}{h_Q} - |h_i - h_j^{(t)}|)^2, & \text{if } |h_i - h_j^{(t)}| > \frac{180^\circ}{h_Q} \\ (h_i - h_j^{(t)})^2, & \text{otherwise.} \end{cases}$$

Ref: T.-W. Chen, Y.-L. Chen, and S.-Y. Chien, “Fast Image Segmentation Based on K-Means Clustering with Histograms in HSV Color Space,” MMSP2008.

K-means

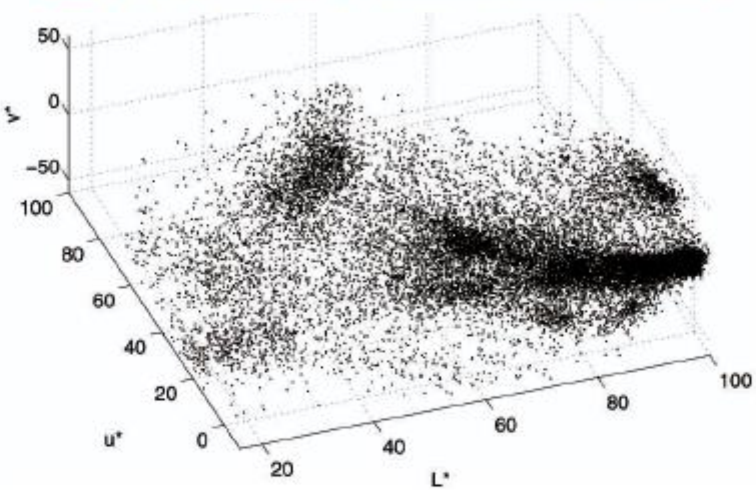


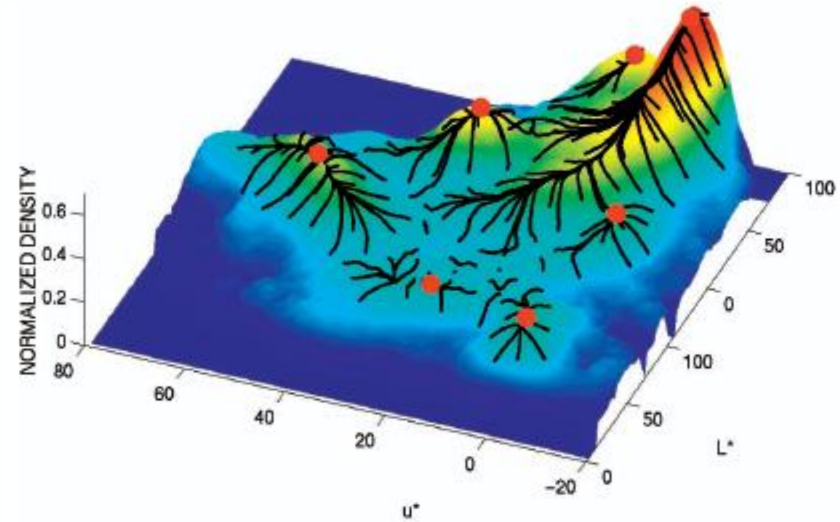
Ref: T.-W. Chen, Y.-L. Chen, and S.-Y. Chien, "Fast Image Segmentation Based on K-Means Clustering with Histograms in HSV Color Space," MMSP2008.

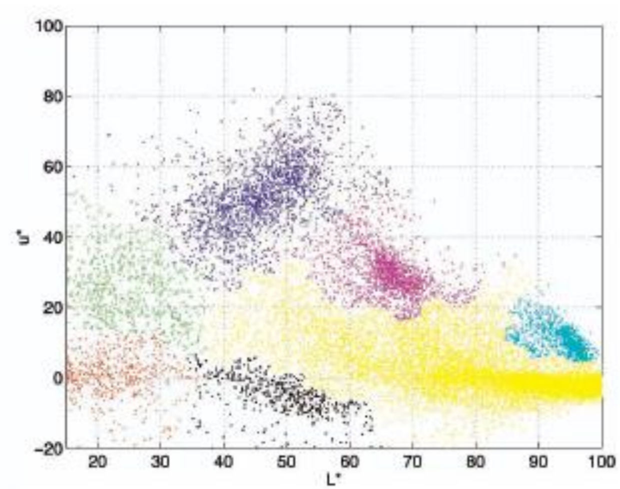
Ref: D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach toward Feature Space Analysis,” PAMI 2002.

Mean-shift Algorithm

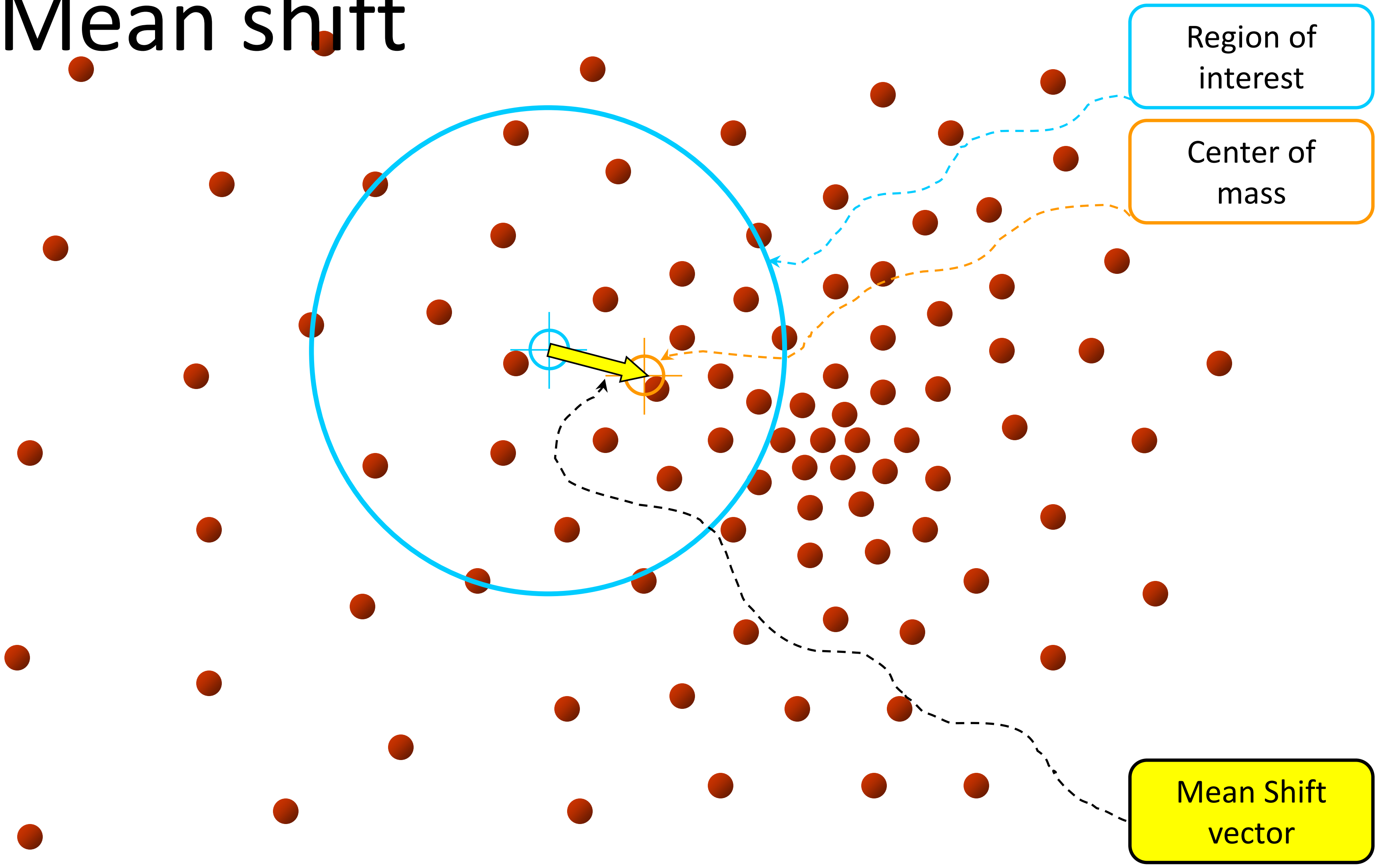
- Try to find *modes* of this non-parametric density





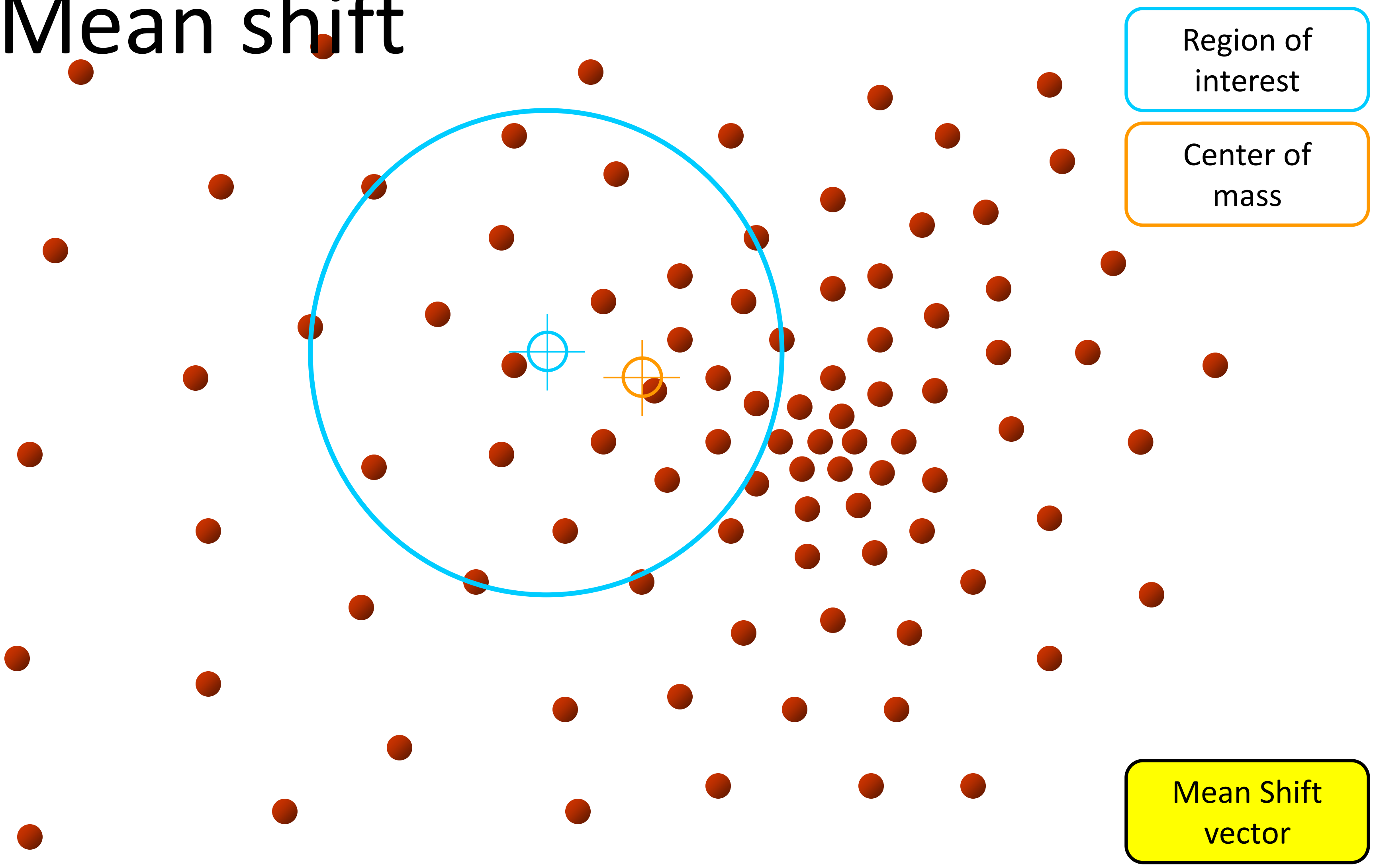


Mean shift



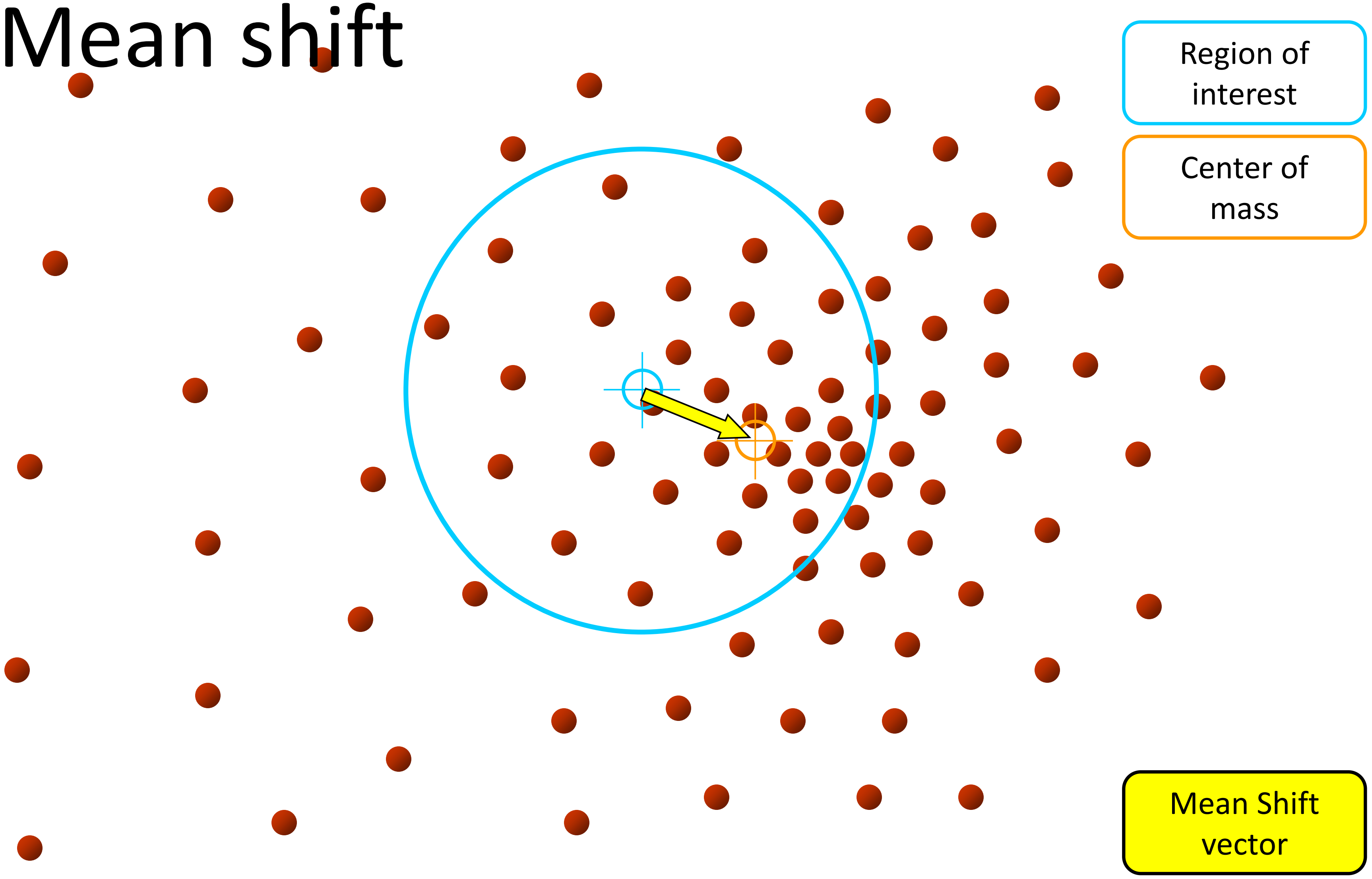
Slide by Y. Ukrainitz & B. Sarel

Mean shift



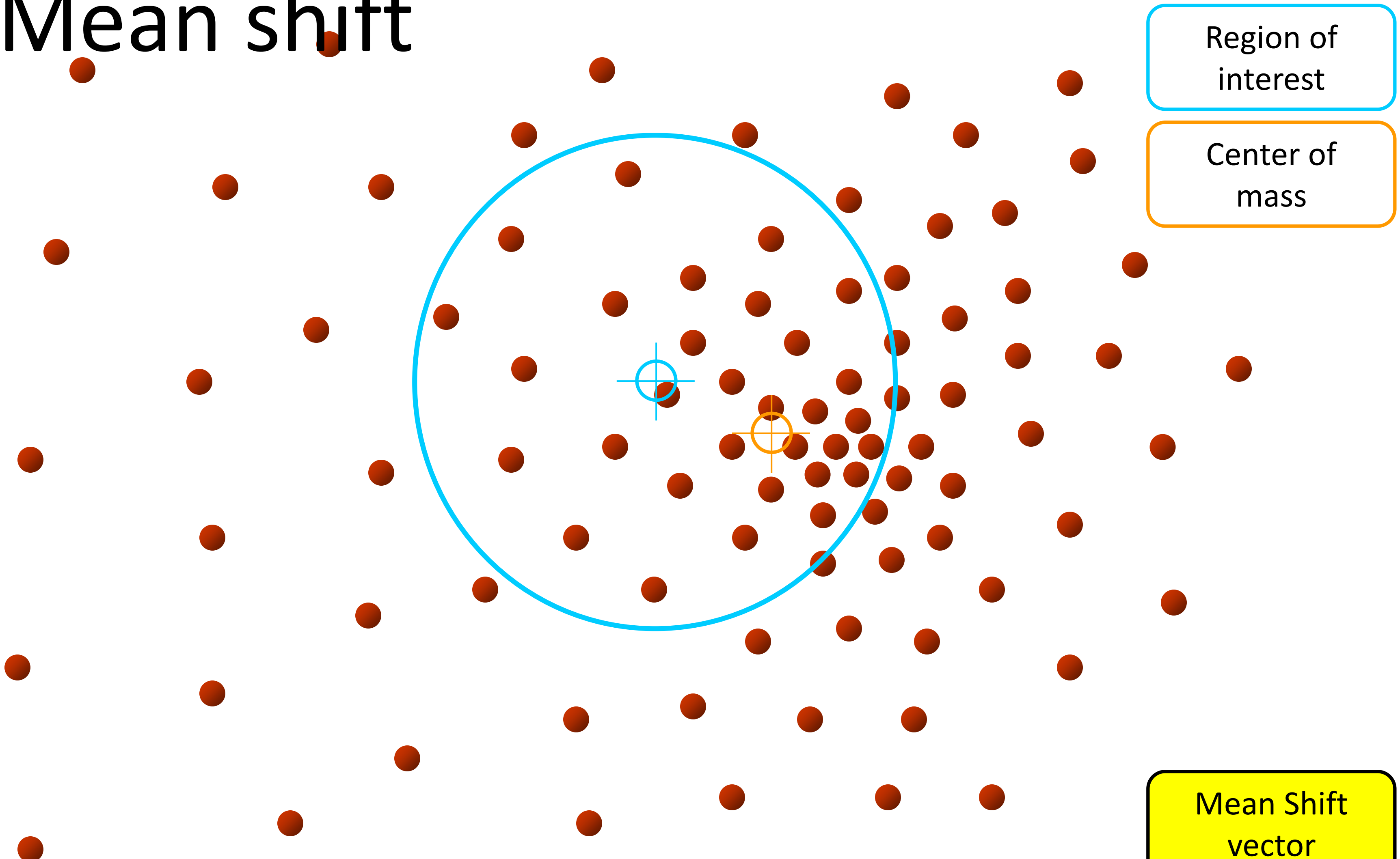
Slide by Y. Ukrainitz & B. Sarel

Mean shift



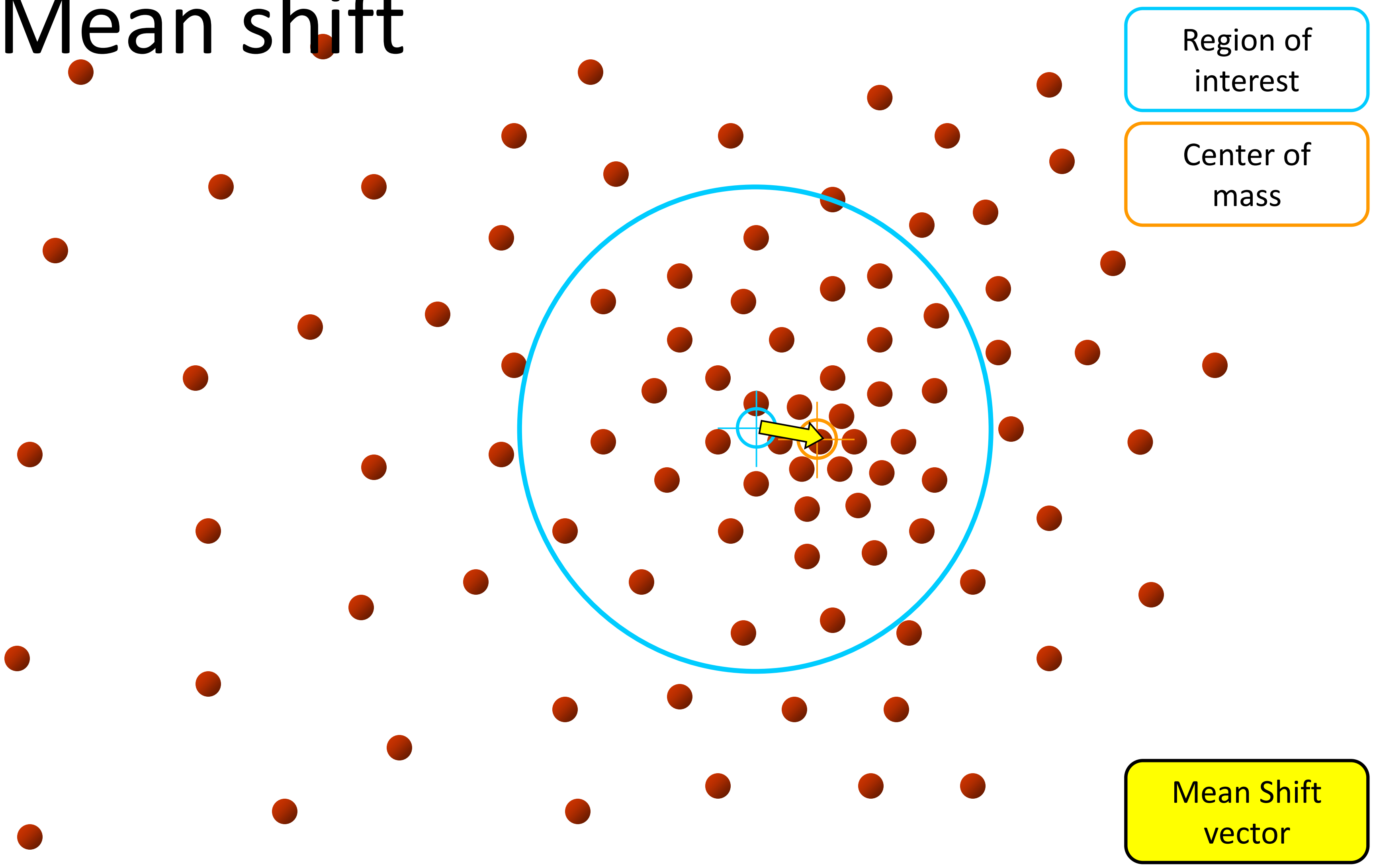
Slide by Y. Ukrainitz & B. Sarel

Mean shift



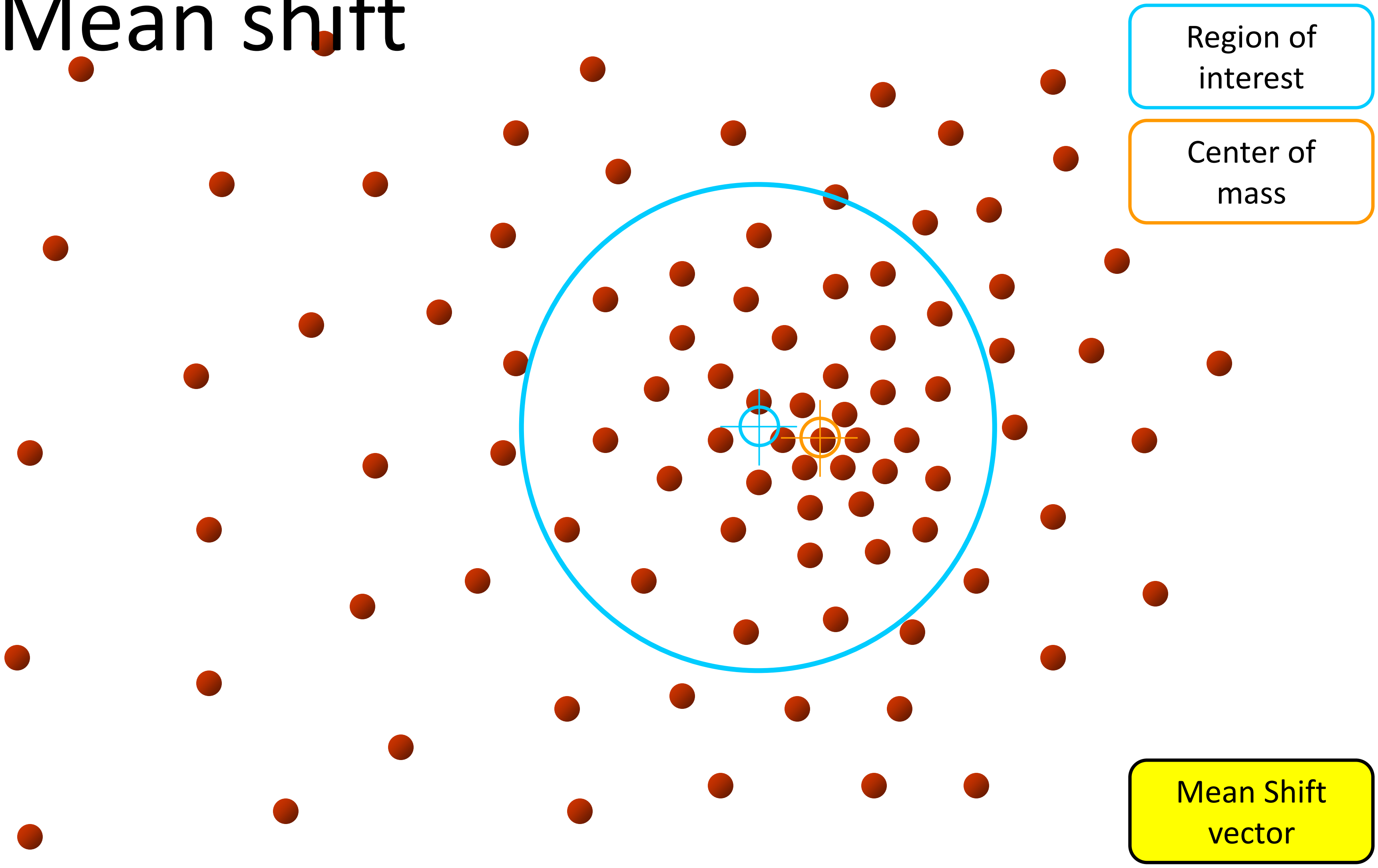
Slide by Y. Ukrainitz & B. Sarel

Mean shift



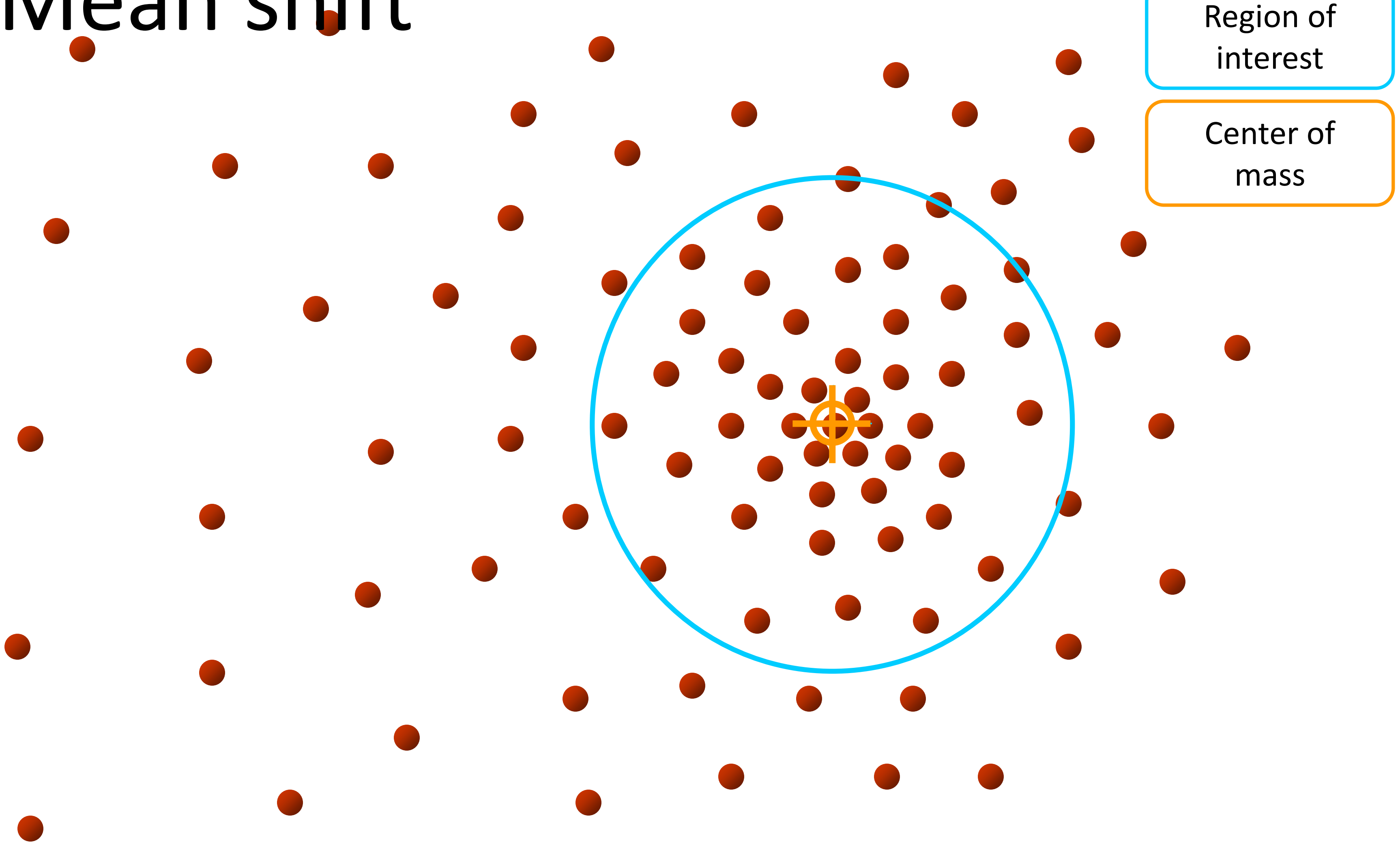
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

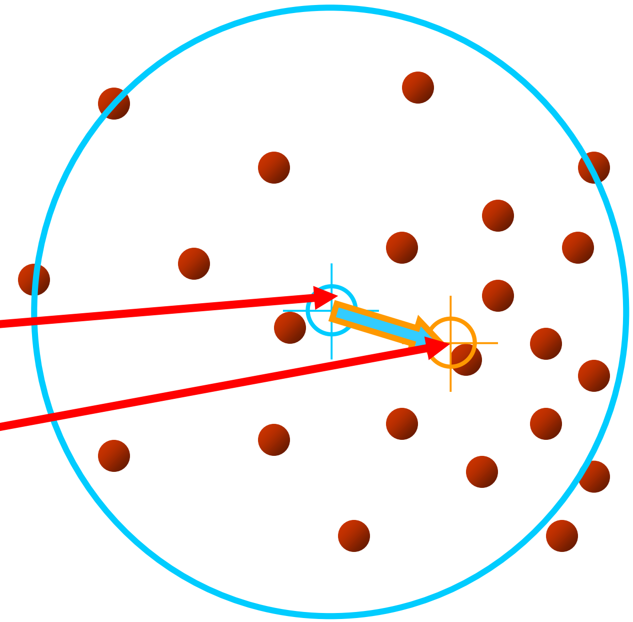
Computing the Mean Shift

<u>Simple Mean Shift procedure</u>:

• Compute mean shift vector

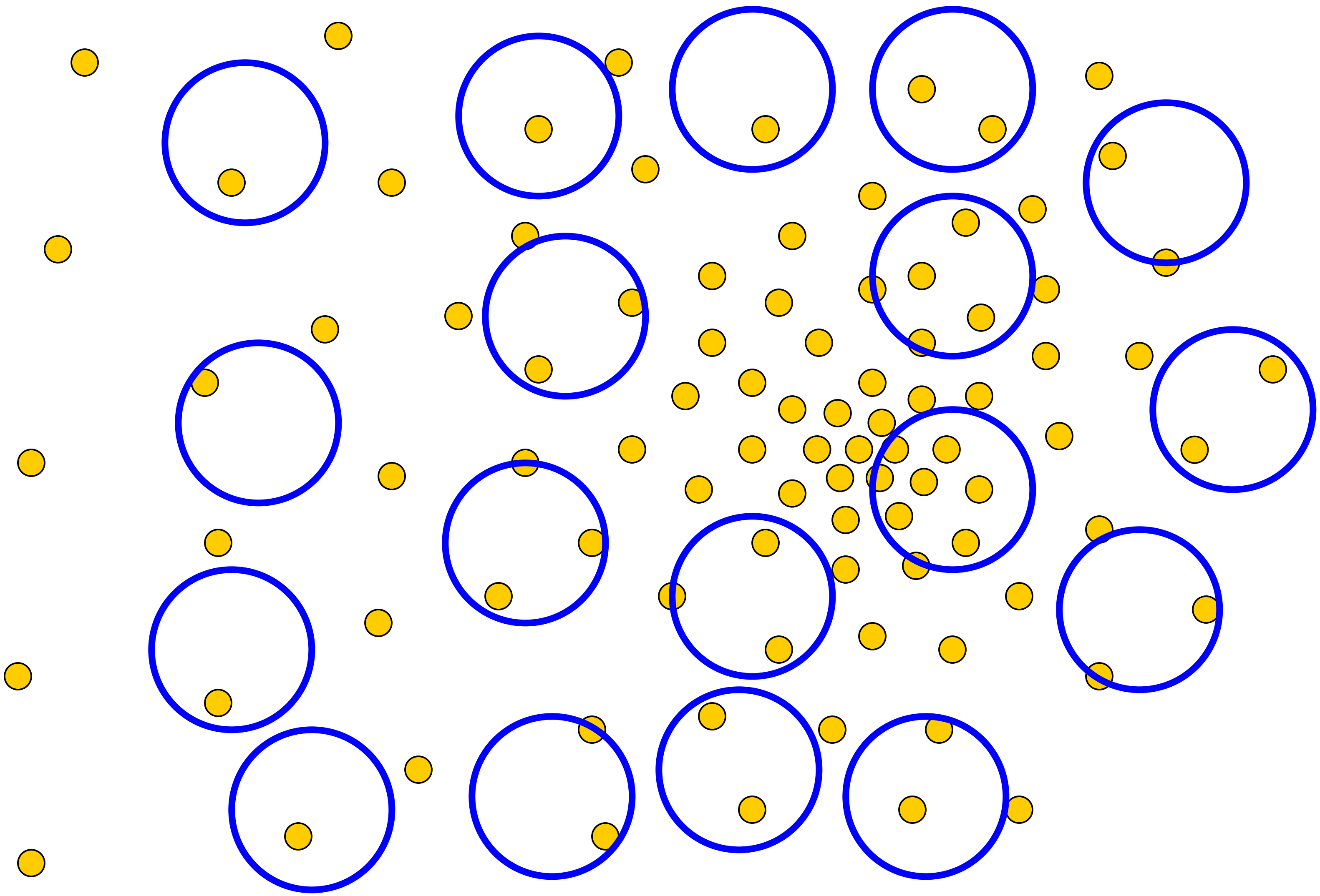
•Translate the Kernel window by **m(x)**

$$\mathbf{m}(\mathbf{x}) = \left[\boxed{\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h}\right)}} - \boxed{\mathbf{x}} \right]$$



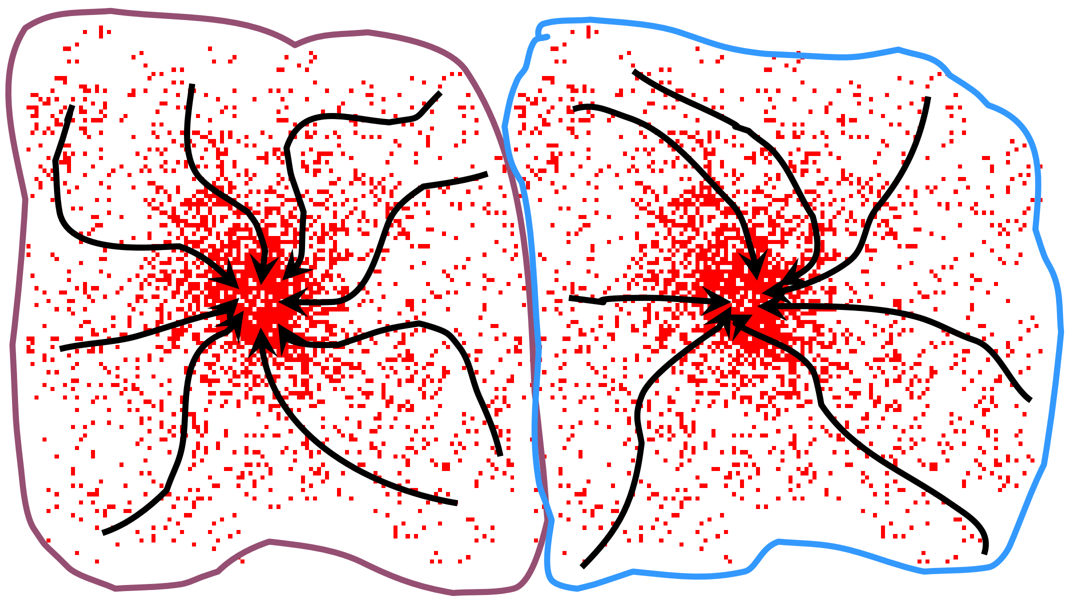
Slide by Y. Ukrainitz & B. Sarel

Real Modality Analysis



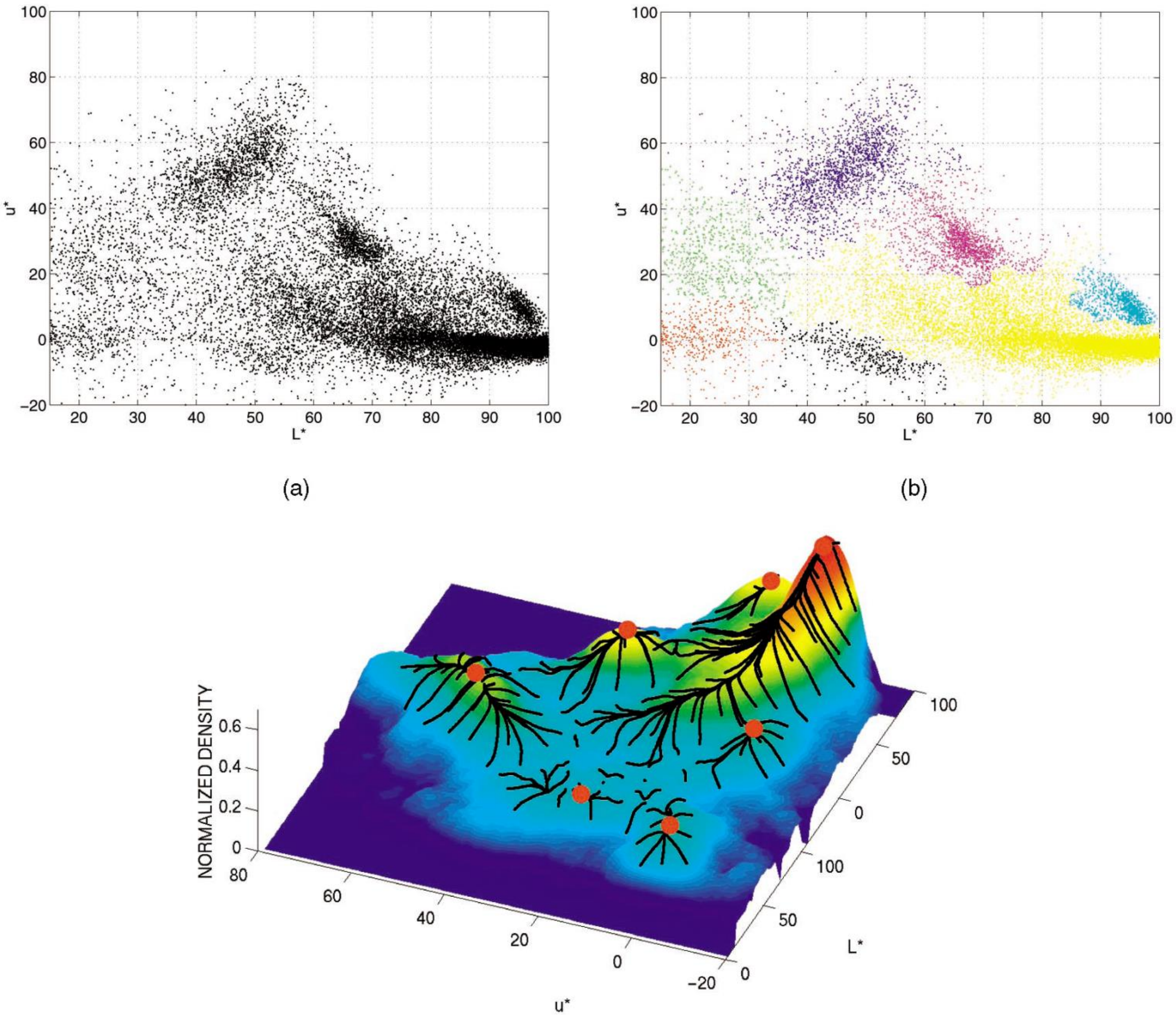
Attraction basin

- Attraction basin: the region for which all trajectories lead to the same mode
- Cluster: all data points in the attraction basin of a mode



Slide by Y. Ukrainitz & B. Sarel

Attraction basin

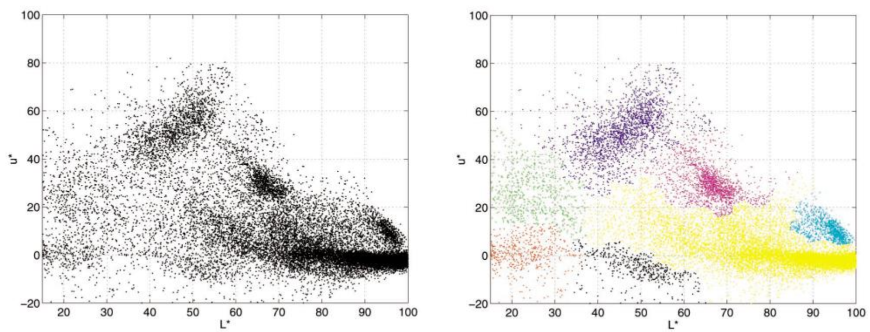


Mean shift clustering

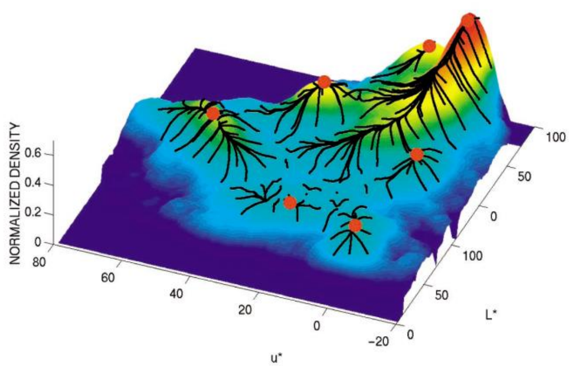
- The mean shift algorithm seeks *modes* of the given set of points
  1. Choose kernel and bandwidth
  2. For each point:
     a) Center a window on that point
     b) Compute the mean of the data in the search window
     c) Center the search window at the new mean location
     d) Repeat (b,c) until convergence
  3. Assign points that lead to nearby modes to the same cluster

Segmentation by Mean Shift

- Compute features for each pixel (color, gradients, texture, etc); also store each pixel’s position
- Set kernel size for features K_f and position K_s
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge modes that are within width of K_f and K_s

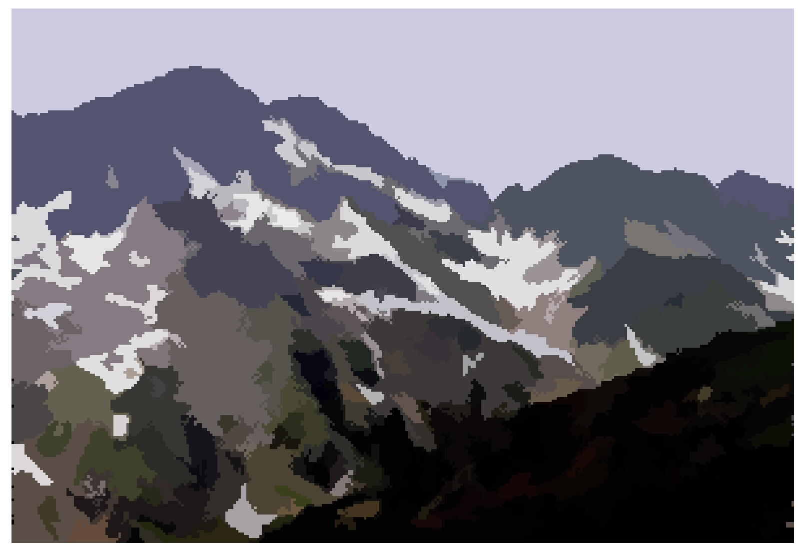


(a) (b)

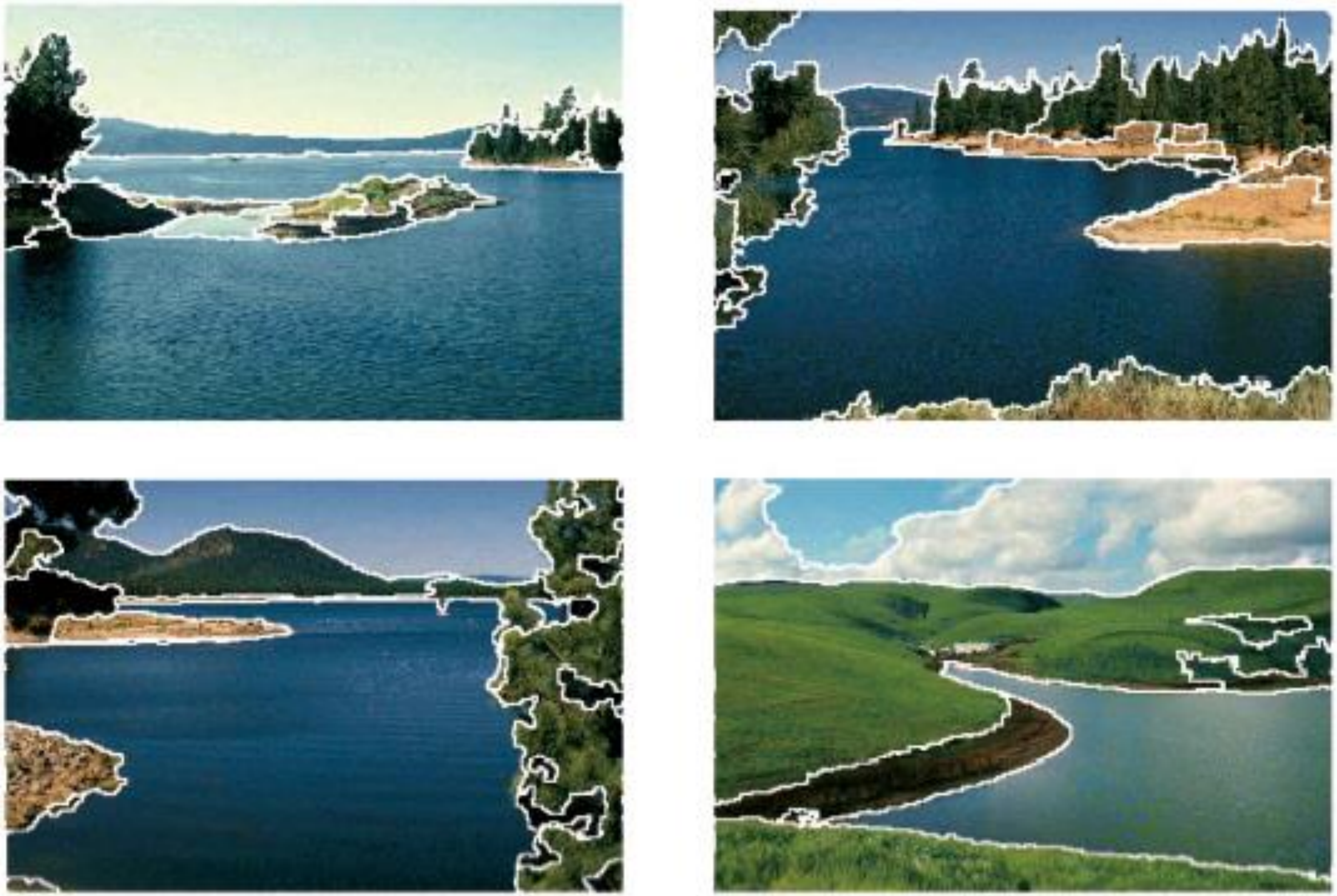


Mean shift segmentation results



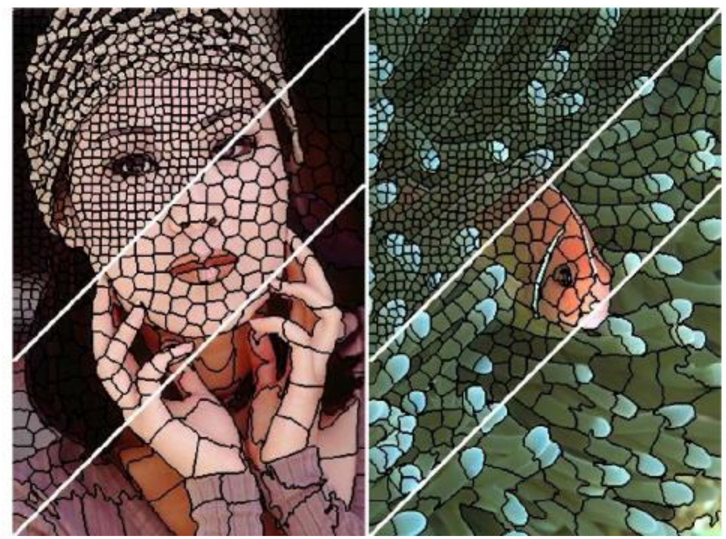
http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html



http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

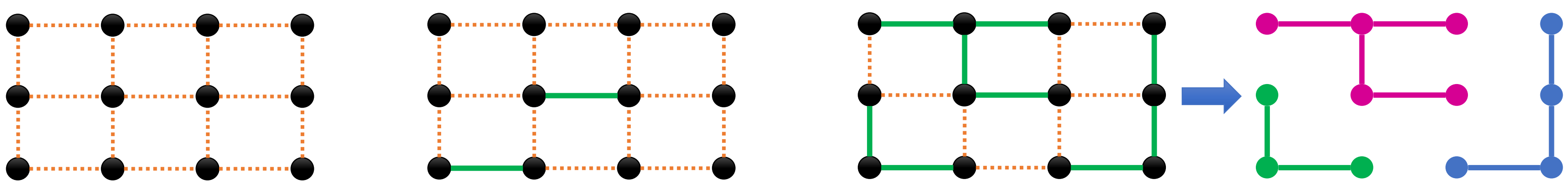
Modern Superpixel Methods
What Are Superpixels?

- Most image processing algorithms use the pixel grid as the underlying representation.
  - Processing time grows with the number of pixels.
- Superpixels are grouping of pixels.
  - Pixels in the same superpixel are near and visually similar (local and edge-preserving)
  - A favor superpixel segmentation algorithm should be efficient
  - Processing time depends on the number of superpixels (regardless of image resolution)



Graph-Based Algorithms

- FH [Felzenszwalb and Huttenlocher, IJCV 2004]
- GBVS [Grundmann et al., CVPR 2010]
- ERS [Liu et al., CVPR 2011]



N pixels as N disjoint sets

After 2 merges, we have $N-2$ sets

To obtain K superpixels,
we do $N-K$ merges ($K=3$ here)

- P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004
- M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010
- M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy-rate superpixel segmentation. In *CVPR*, 2011

Graph-Based Algorithms

- Graph-based methods are able to generate **superpixel hierarchy**

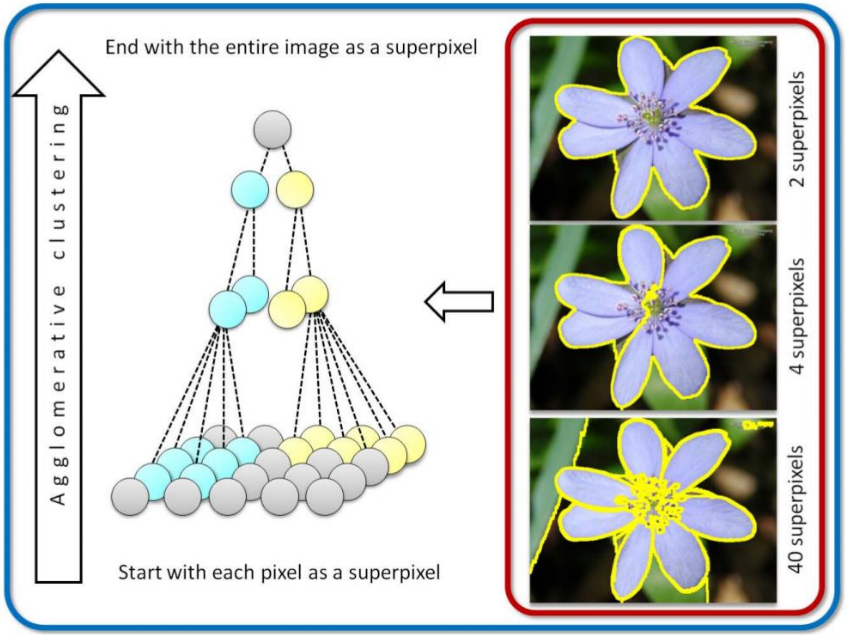
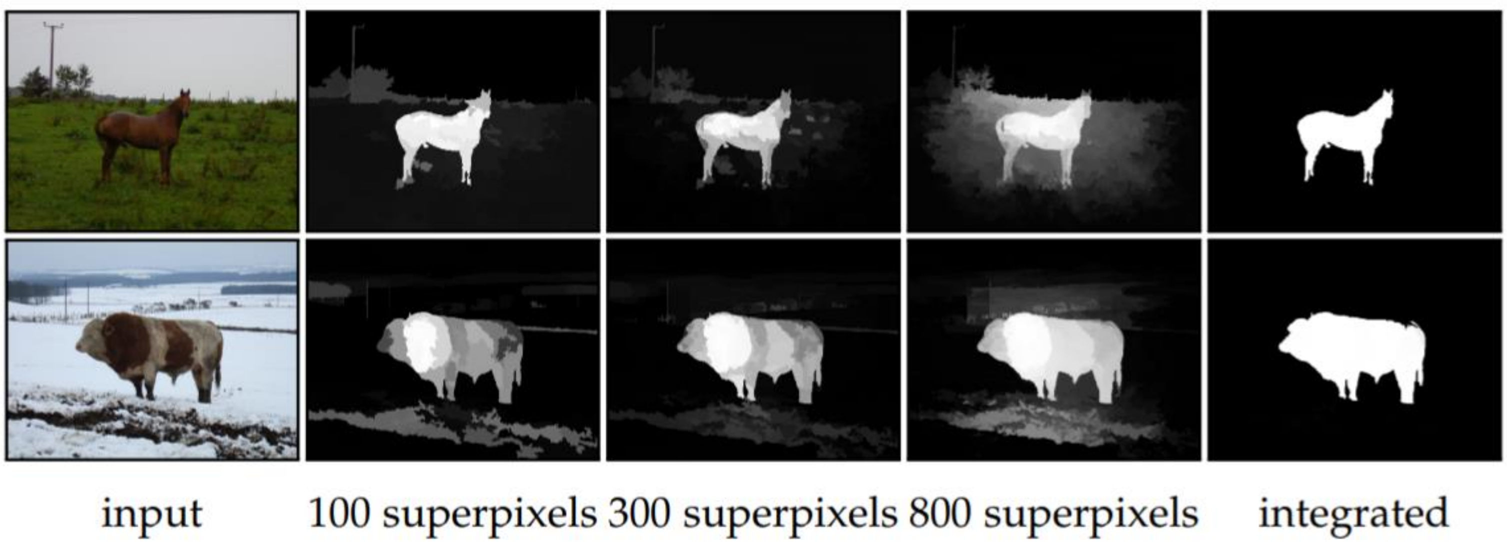


Figure from ERS paper

Graph-Based Algorithms

- Graph-based methods are able to generate **superpixel hierarchy**



Example of salient object segmentation based on the superpixel hierarchy

Clustering-Based Algorithms

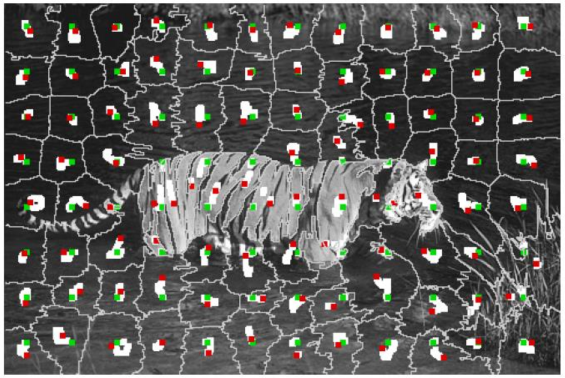
- SLIC (Simple Linear Iterative Clustering)
  - RGB → CIELab
  - 5D feature (L, a, b, x, y)
  - Initialize the K superpixel centers on the uniform grid
  - Localized K-means clustering in 2S x 2S region

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$

$$D_s = d_{lab} + \frac{m}{S} d_{xy} \,,$$

m is a constant



Localized k-means

- R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. "SLIC superpixels compared to state-of-the-art superpixel methods." *TPAMI*, 2012

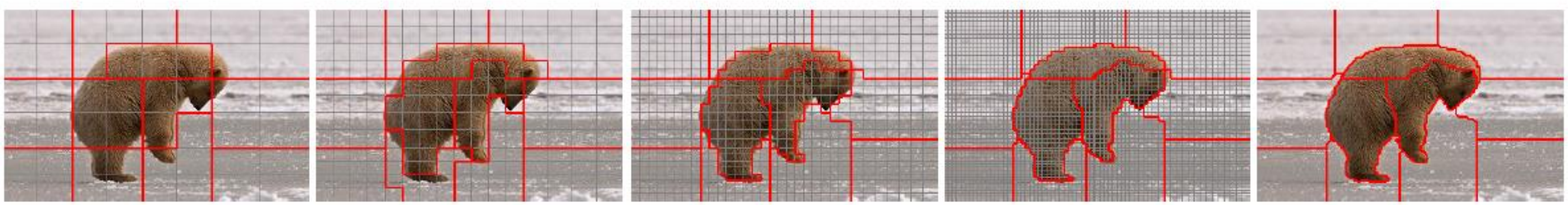
Other SLIC-Like Algorithms

- LSC [Li and Chen, CVPR 2015]
  - 10D feature + localized K-means
- Manifold-SLIC [Liu et al., CVPR 2016]
  - Project 5D feature to a 2D space + localized K-means
- SNIC [Achanta and Susstrunk, CVPR 2017]
  - 5D feature + iteration free clustering

- Z. Li and J. Chen. Superpixel segmentation using linear spectral clustering. In *CVPR*, 2015
- Yong-Jin Liu, Cheng-Chi Yu, Min-Jing Yu, and Ying He. Manifold slic: A fast method to compute content-sensitive superpixels. In *CVPR,* 2016
- R. Achanta and S. Susstrunk. Superpixels and polygons using simple non-iterative clustering. In *CVPR*, 2017

Grid-Based Algorithms

- SEEDS [Van den Bergh et al., IJCV 2015]
  - Superpixels as an energy optimization (color consistency, boundary shape, …)
  - Switch nearby blocks if it makes the total energy lower
  - Coarse to fine strategy

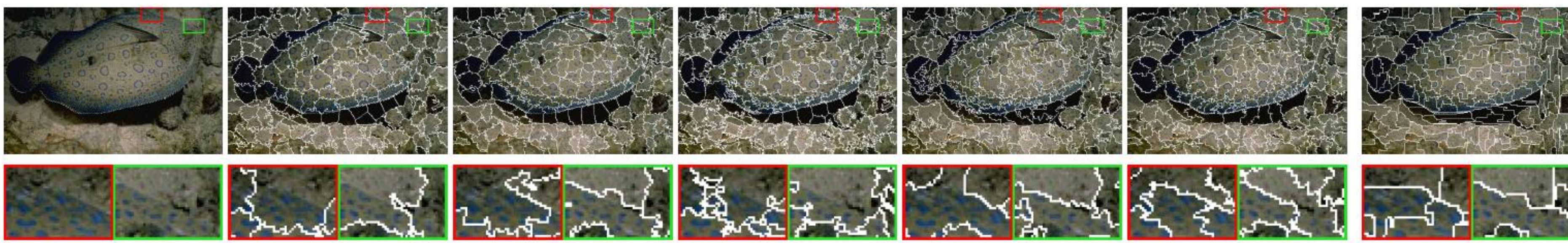


Multi-scale block switching

- M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool. SEEDS: Superpixels extracted via energy-driven sampling. *IJCV*, 2015

Drawbacks of Existing Methods

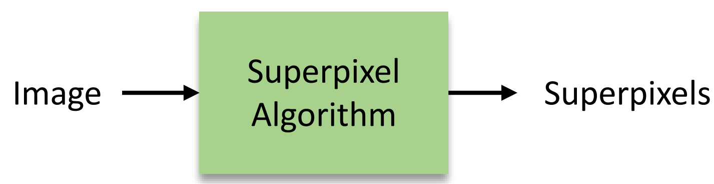
- All above methods are based on hand-crafted features to compute pixel distances/affinities
  - They often fail to preserve weak object boundaries



Input SLIC SNIC LSC SEEDS ERS Ours

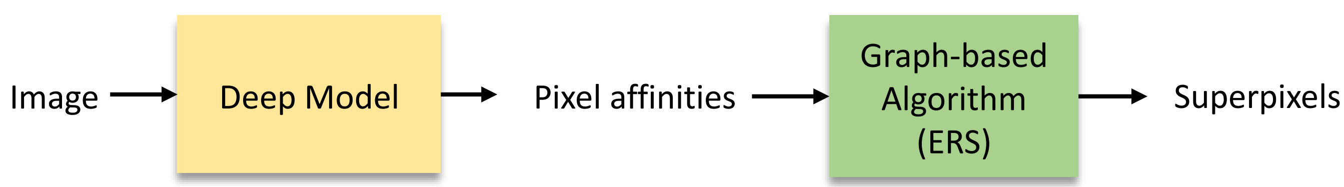
Superpixels Meet Deep Learning

- Supervised learning is not easy
  - There is no ground-truth
  - Label indices are interchangeable
  - Superpixel algorithms are non-differentiable



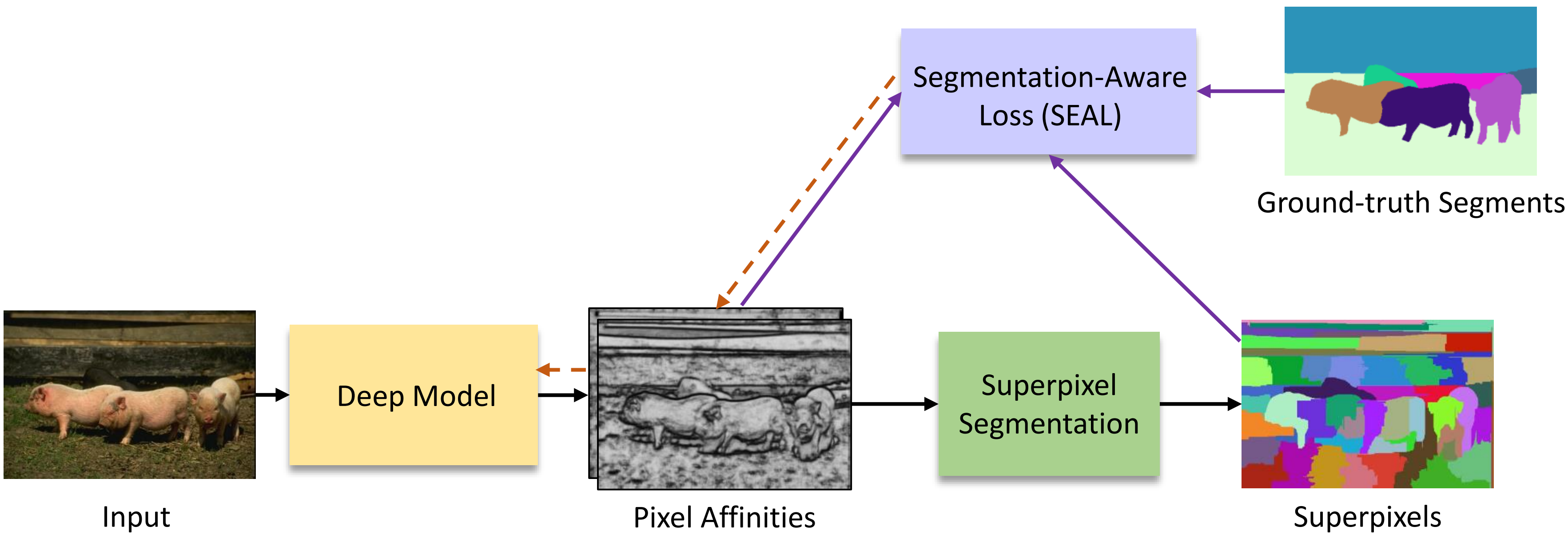
Superpixels Meet Deep Learning

- Supervised learning is not easy
  - There is no ground-truth
  - Label indices are interchangeable
  - Superpixel algorithms are non-differentiable
- Our main idea: learning pixel affinities (distances) for the graph-based algorithms
  [Tu et al., CVPR 2018]



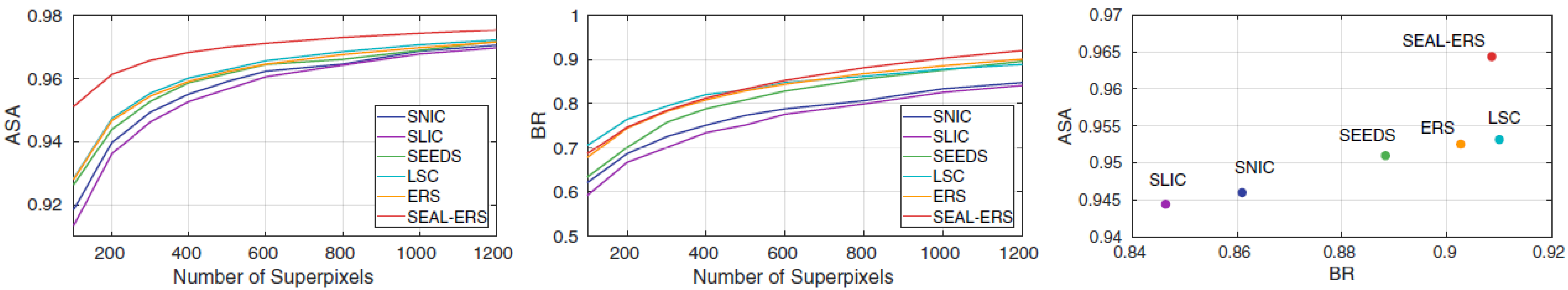
Wei-Chih Tu, Ming-Yu Liu, Varun Jampani, Deqing Sun, Shao-Yi Chien, Ming-Hsuan Yang, Jan Kautz.
Learning superpixels with segmentation-aware affinity loss. In *CVPR*, 2018

Segmentation-Aware Loss



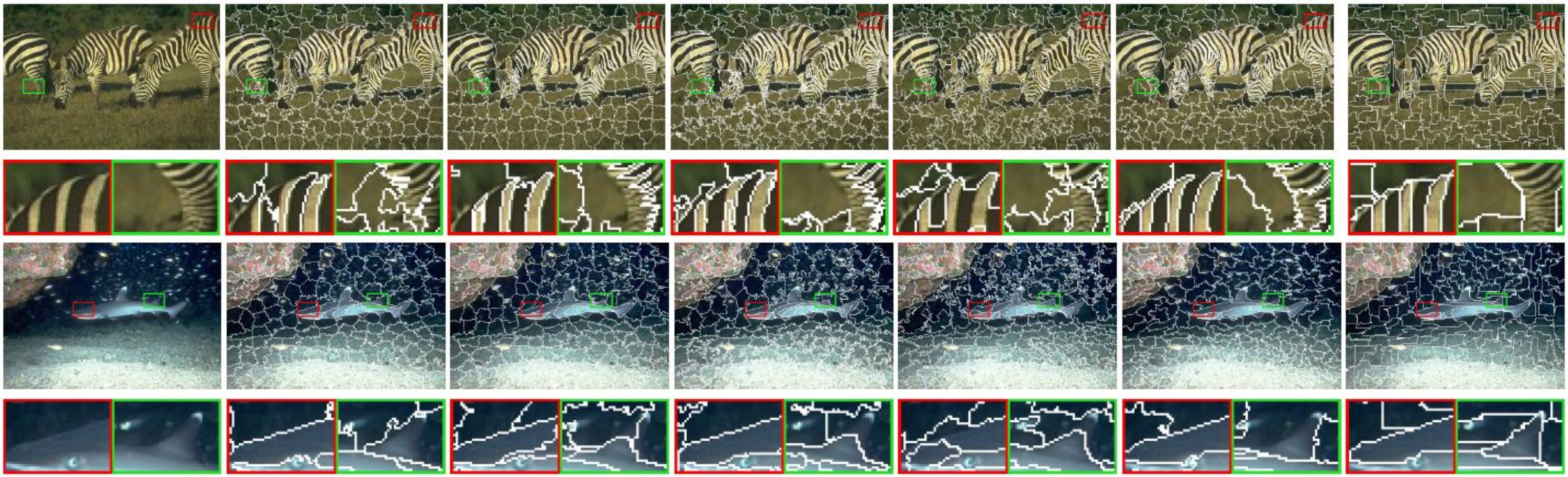
Comparisons with the State-of-the-Arts

- Results on BSDS500
  - SEAL-ERS = learned affinities + ERS algorithm (proposed)



Wei-Chih Tu, Ming-Yu Liu, Varun Jampani, Deqing Sun, Shao-Yi Chien, Ming-Hsuan Yang, Jan Kautz. Learning superpixels with segmentation-aware affinity loss. In *CVPR*, 2018

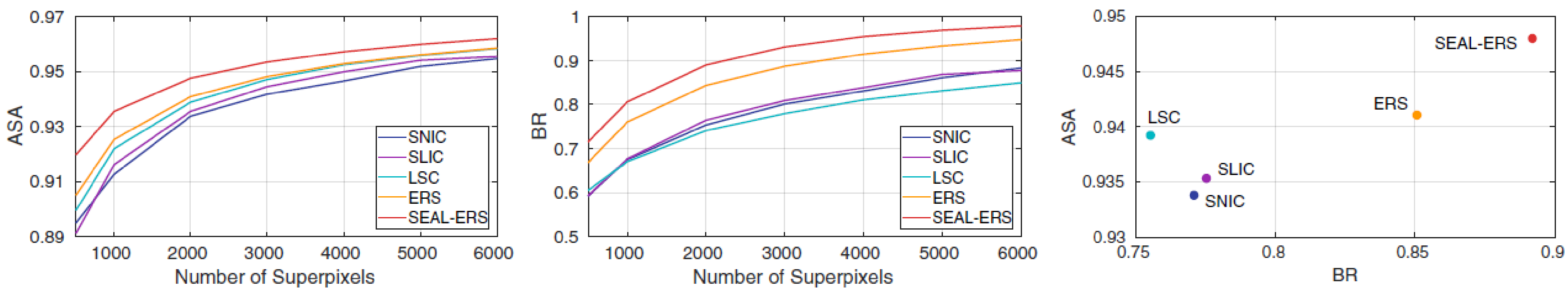
Comparisons with the State-of-the-Arts



Input SLIC SNIC LSC SEEDS ERS Ours

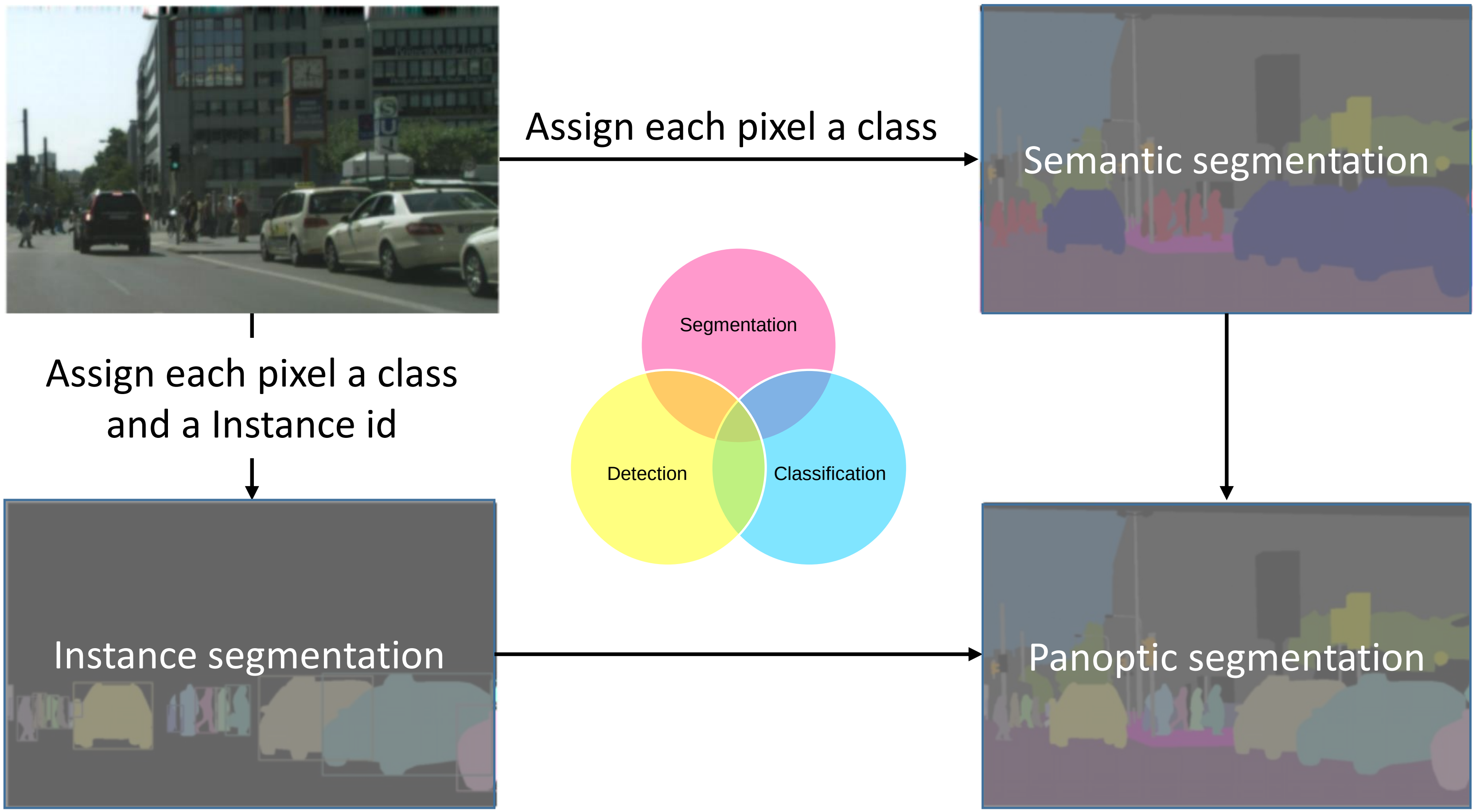
Comparisons with the State-of-the-Arts

- Results on Cityscapes



Wei-Chih Tu, Ming-Yu Liu, Varun Jampani, Deqing Sun, Shao-Yi Chien, Ming-Hsuan Yang, Jan Kautz. Learning superpixels with segmentation-aware affinity loss. In *CVPR*, 2018

Image Segmentation: Semantic Segmentation



Ref: A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, “Panoptic Segmentation,” CVPR 2019.

Image Segmentation: Semantic Segmentation

- Fully convolutional networks (FCN)
- DeepLab

What is Semantic Segmentation?

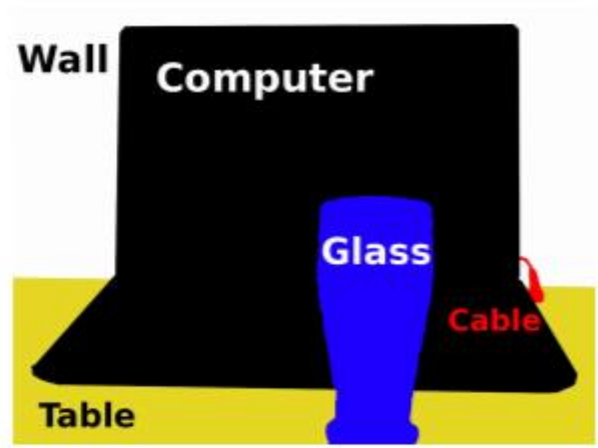
- Segmentation + labeling

(a) Image (b) Ground Truth

Example from ADE20K dataset.

Why Semantic Segmentation?

- As a vision aid for the blind



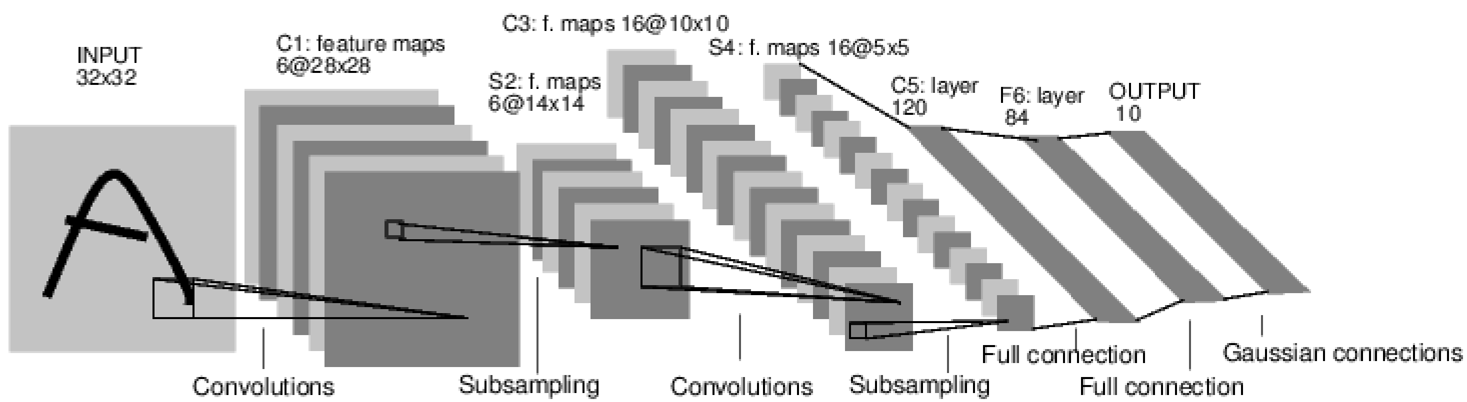
https://arxiv.org/pdf/1602.06541.pdf

Why Semantic Segmentation?

- Autonomous driving

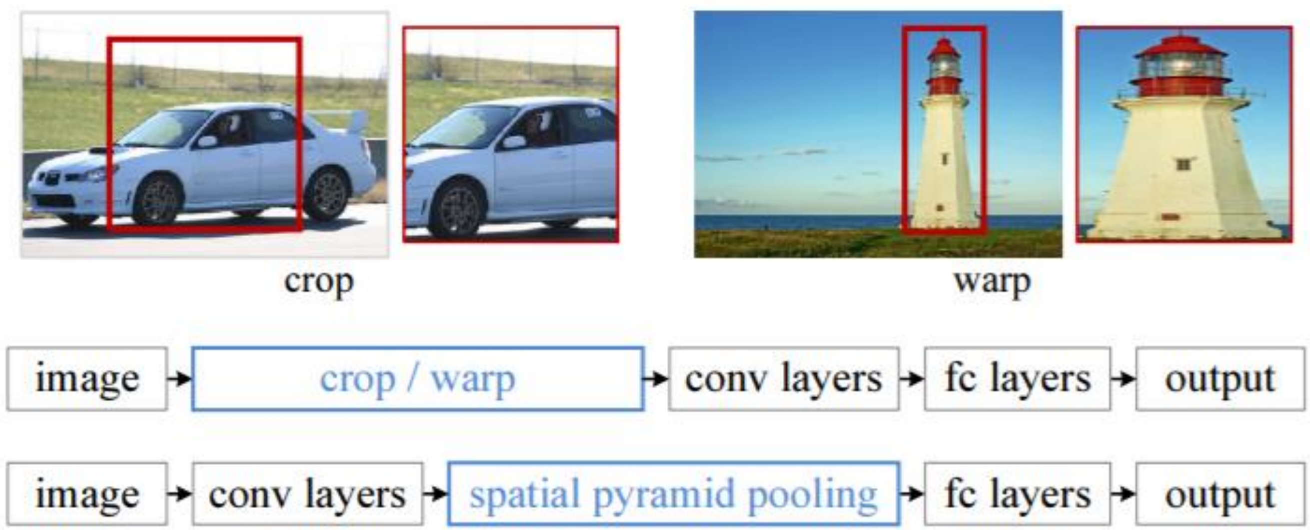
Previous Image Recognition Networks

- LeNet, AlexNet or their successors take fixed size input and produce non-spatial outputs.



Previous Image Recognition Networks

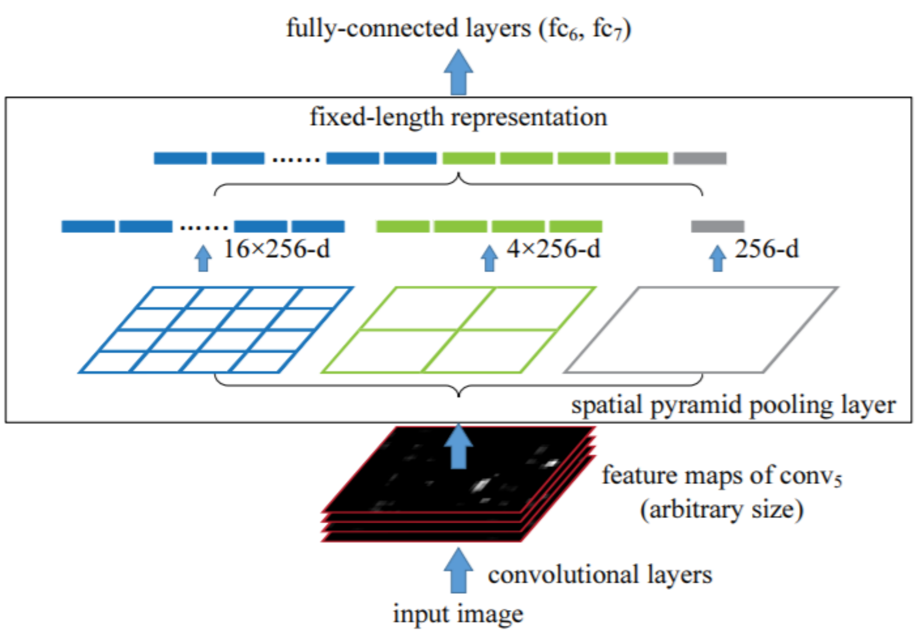
- Spatial pyramid pooling can take arbitrary size input but still no spatial output.



Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, ECCV 2014

Previous Image Recognition Networks

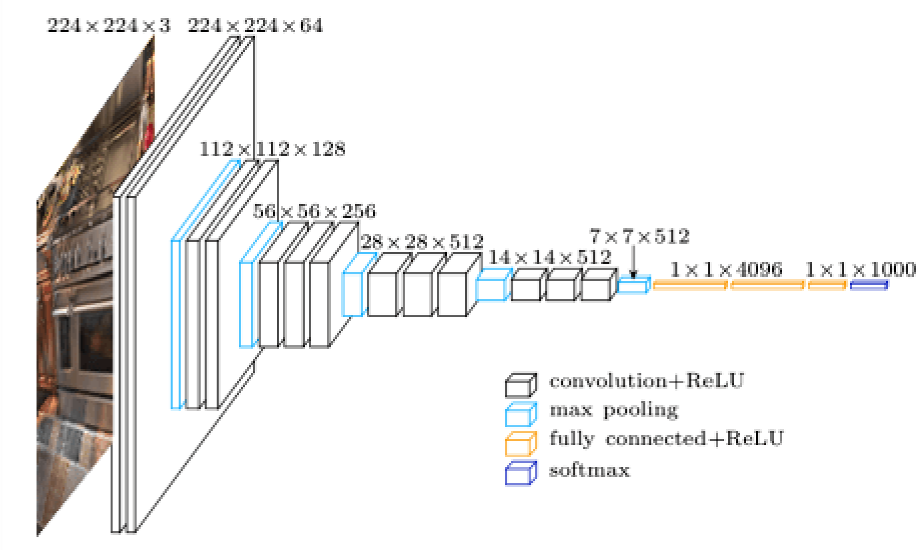
- Spatial pyramid pooling can take arbitrary size input but still no spatial output.



Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, ECCV 2014

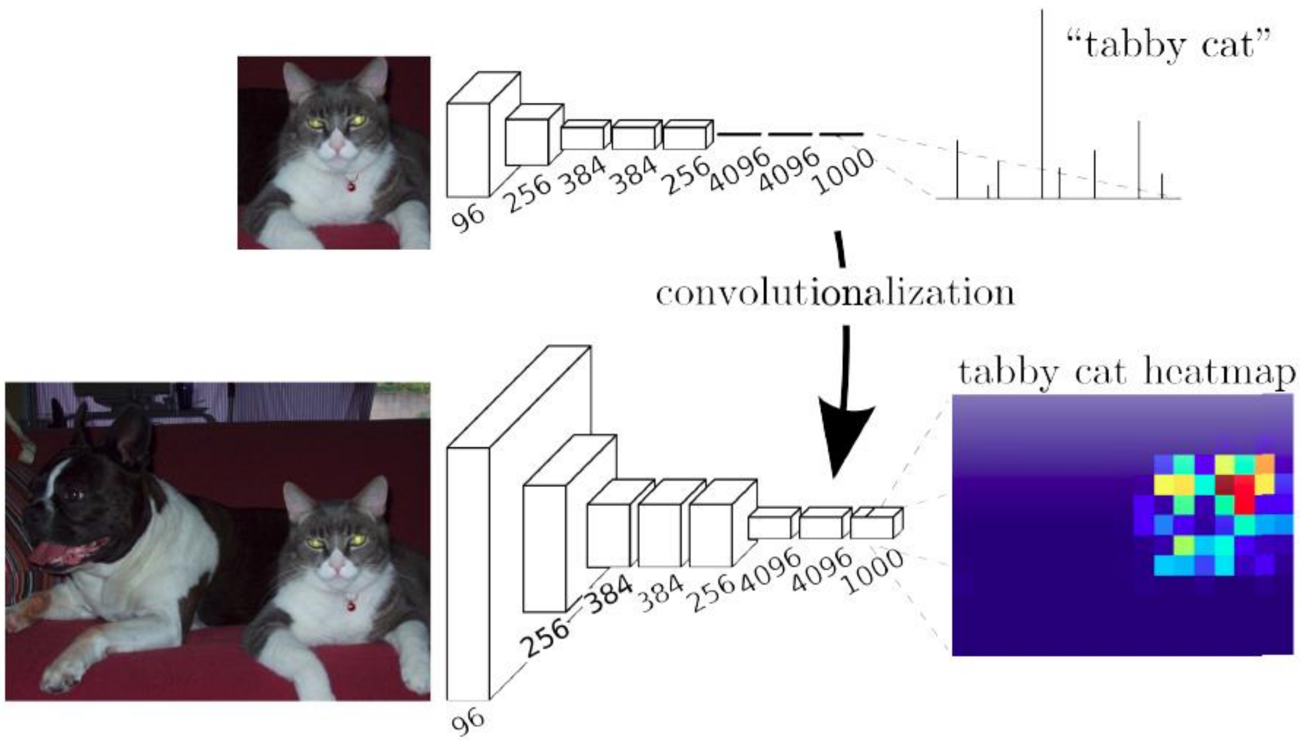
VGG16 Model

- Pre-trained on image classification



Fully Convolutional Networks (FCN)

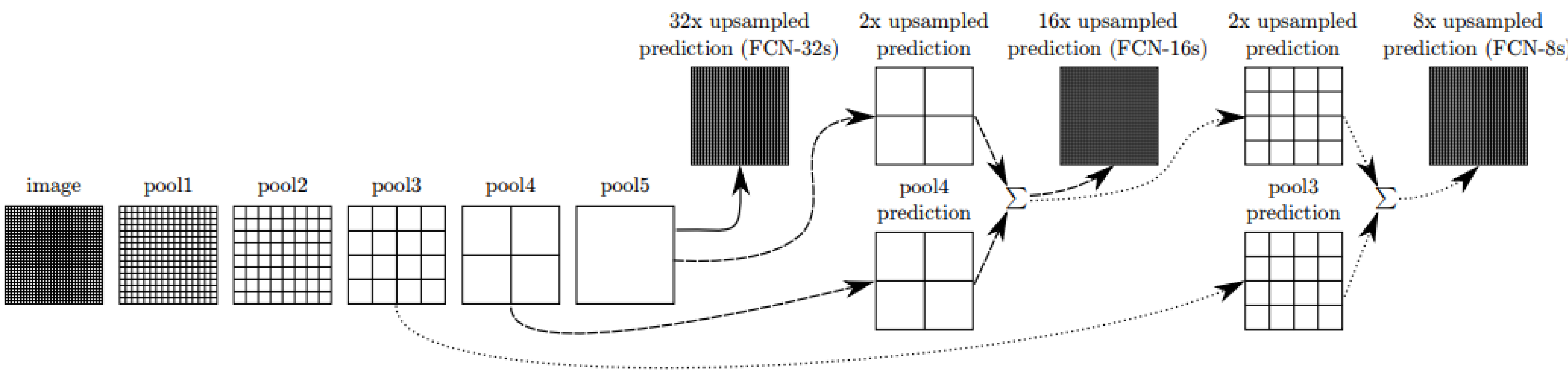
- Fully connected layers can also be viewed as convolutions with kernels that cover their entire input regions



Fully convolutional networks for semantic segmentation, CVPR 2015

FCN Architecture

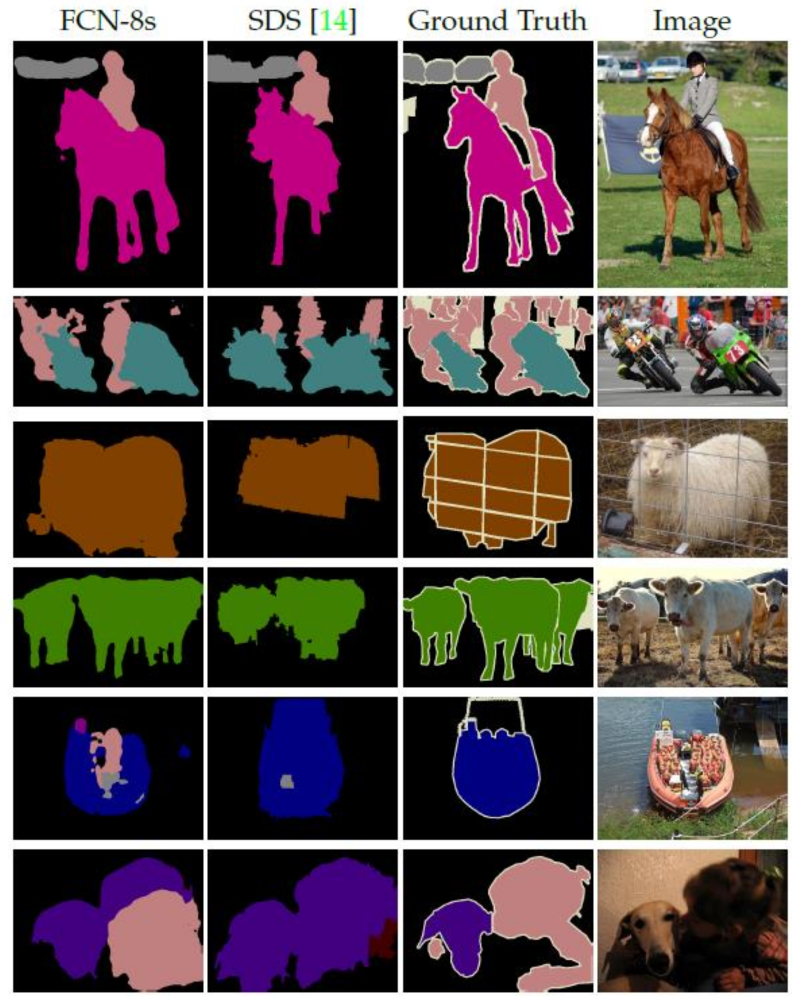
- Fully connected layers are replaced by convolutions
- Append 1x1 convolution with channel dimension 21 in the end (20 classes + 1 background class)



Fully Convolutional Networks (FCN)

- Results

	mean IU VOC2011 test	mean IU VOC2012 test
R-CNN [5]	47.9	-
SDS [14]	52.6	51.6
FCN-8s	67.5	67.2

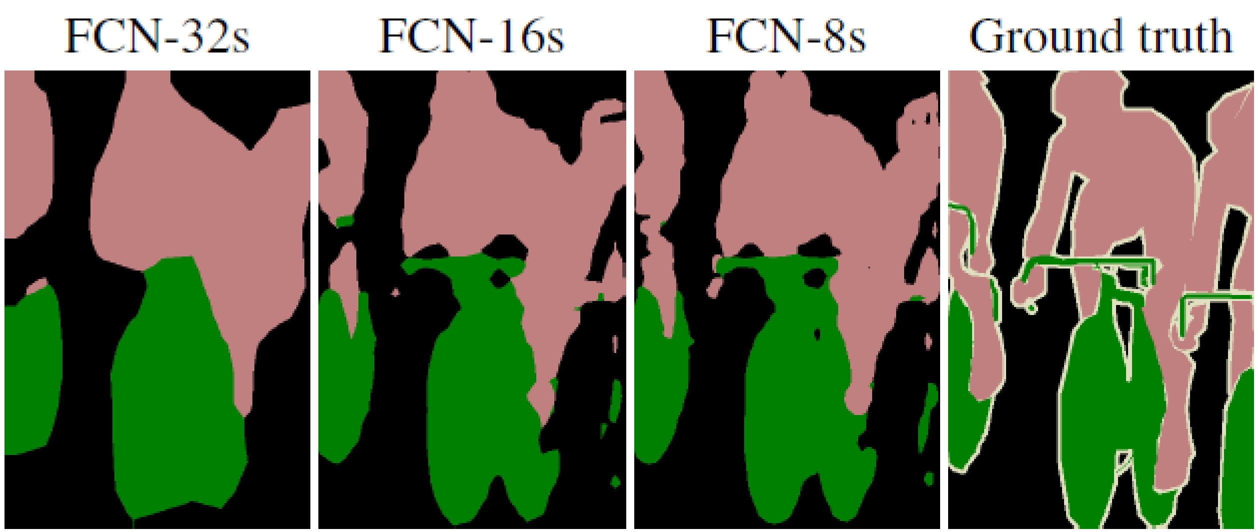


- Definition
  - n_{ij}: number of pixels in class i predicted to be class j
  - $t_i = \sum_j n_{ij}$ be the total number of pixels in class i
  - n_{cl}: number of classes
- Pixel accuracy
  - $\sum_i n_{ii} / \sum_i t_i$
- Mean accuracy
  - $\frac{1}{n_{cl}} \sum_i n_{ii}/t_i$
- Mean IU (intersection over union)
  - $\frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}$

Fully convolutional networks for semantic segmentation, CVPR 2015

Fully Convolutional Networks (FCN)

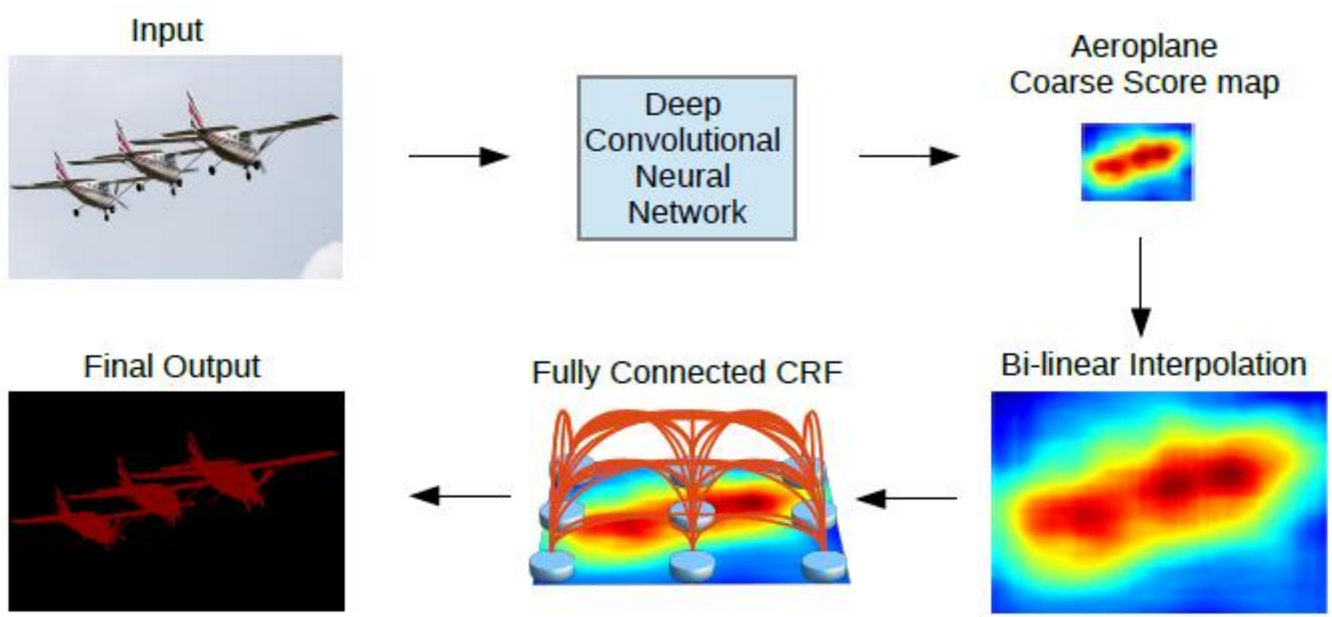
- FCN is still not good at segmenting objects…



Fully convolutional networks for semantic segmentation, CVPR 2015

DeepLab

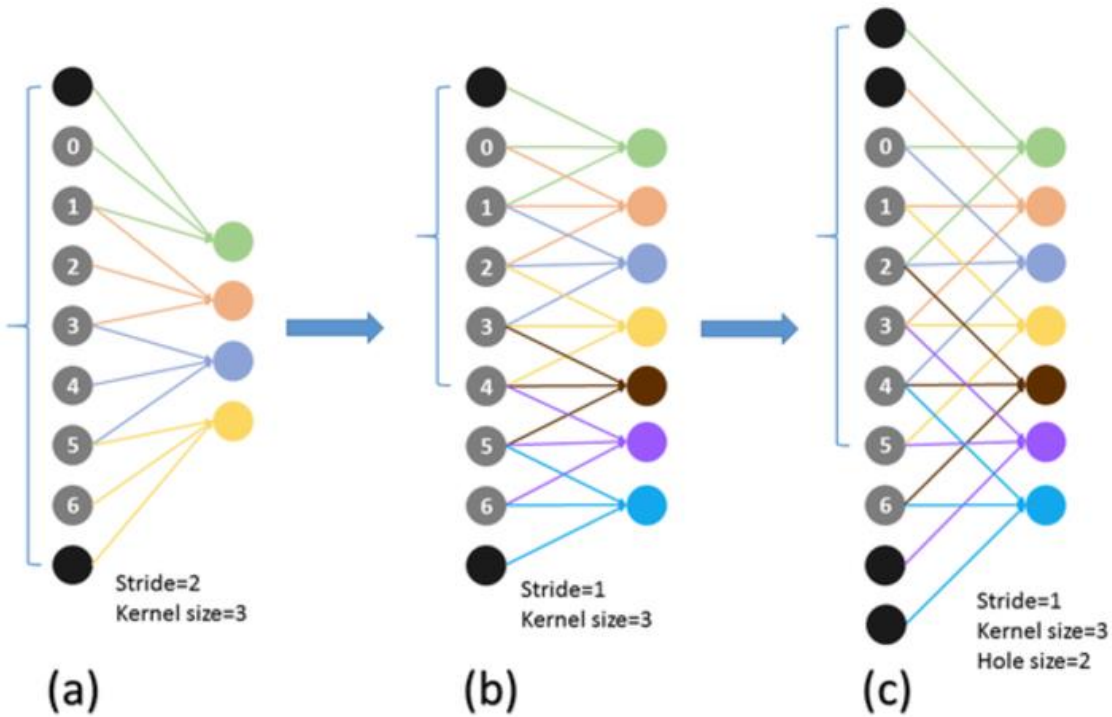
- FCN + Atrous convolution + dense CRFs (conditional random field)



Semantic image segmentation with deep convolutional nets and fully connected CRFs, ICLR 2015

DeepLab

- Atrous convolution (dilated convolution)



Semantic image segmentation with deep convolutional nets and fully connected CRFs, ICLR 2015

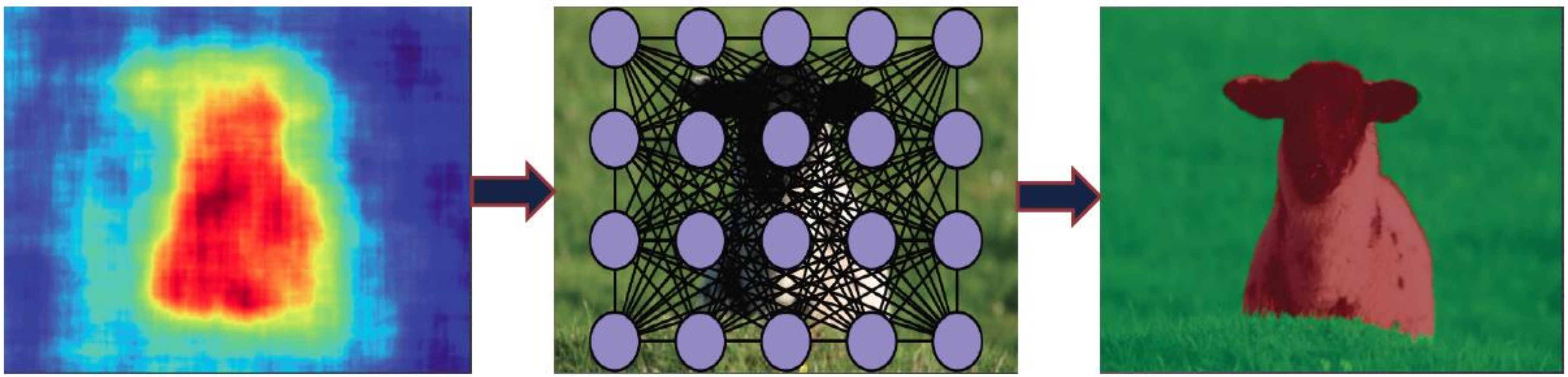
DeepLab

- Dense CRFs

From FCN output

From input image

$$E(\mathbf{x}) = \sum_i \underbrace{\psi_u(x_i)}_{\text{unary term}} + \sum_i \sum_{j \in \mathcal{N}_i} \underbrace{\psi_p(x_i, x_j)}_{\text{pairwise term}}$$



Coarse output from the pixel-wise classifier

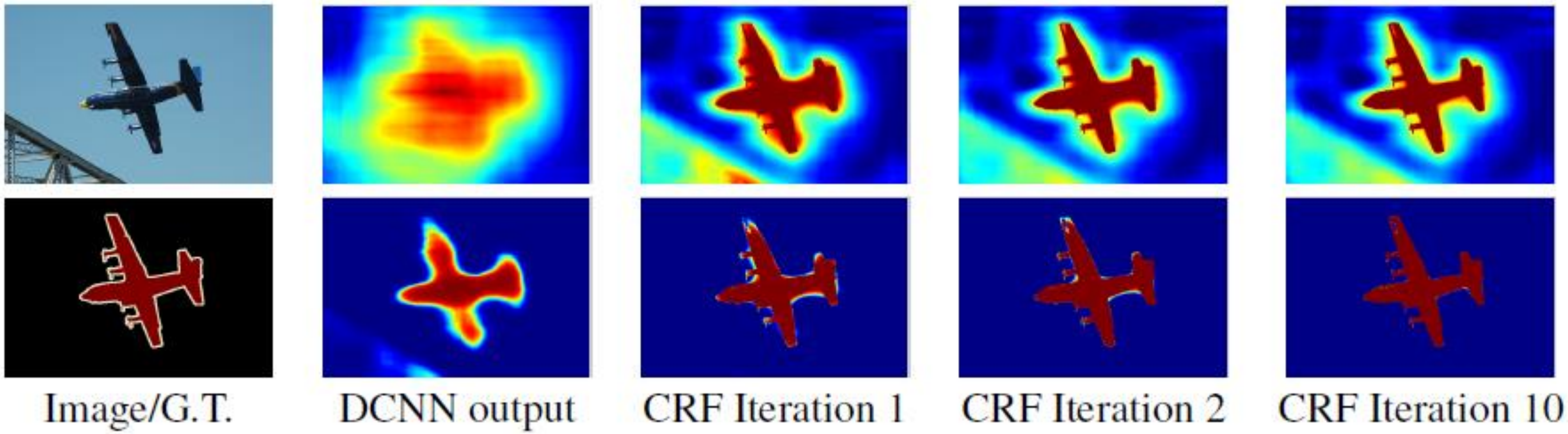
CRF modelling

Output after the CRF inference

Efficient inference in fully connected CRFs with Gaussian edge potentials, NIPS 2011

DeepLab

- Effect of dense CRF refinement



Problem:

1. No joint training
2. More number of iterations means longer inference time

Semantic image segmentation with deep convolutional nets and fully connected CRFs, ICLR 2015

DeepLab

- Results on PASCAL VOC 2012 test set

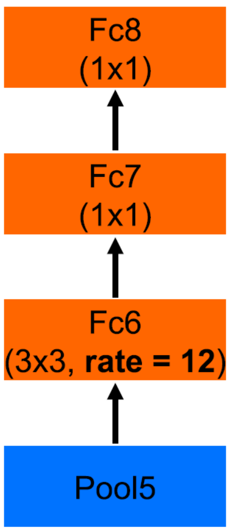
Method	mean IOU (%)
DeepLab	59.80
DeepLab-CRF	63.74
DeepLab-MSc	61.30
DeepLab-MSc-CRF	65.21
DeepLab-7x7	64.38
DeepLab-CRF-7x7	67.64
DeepLab-LargeFOV	62.25
DeepLab-CRF-LargeFOV	67.64
DeepLab-MSc-LargeFOV	64.21
DeepLab-MSc-CRF-LargeFOV	68.70

Method	mean IOU (%)
MSRA-CFM	61.8
FCN-8s	62.2
TTI-Zoomout-16	64.4
DeepLab-CRF	66.4
DeepLab-MSc-CRF	67.1
DeepLab-CRF-7x7	70.3
DeepLab-CRF-LargeFOV	70.3
DeepLab-MSc-CRF-LargeFOV	71.6

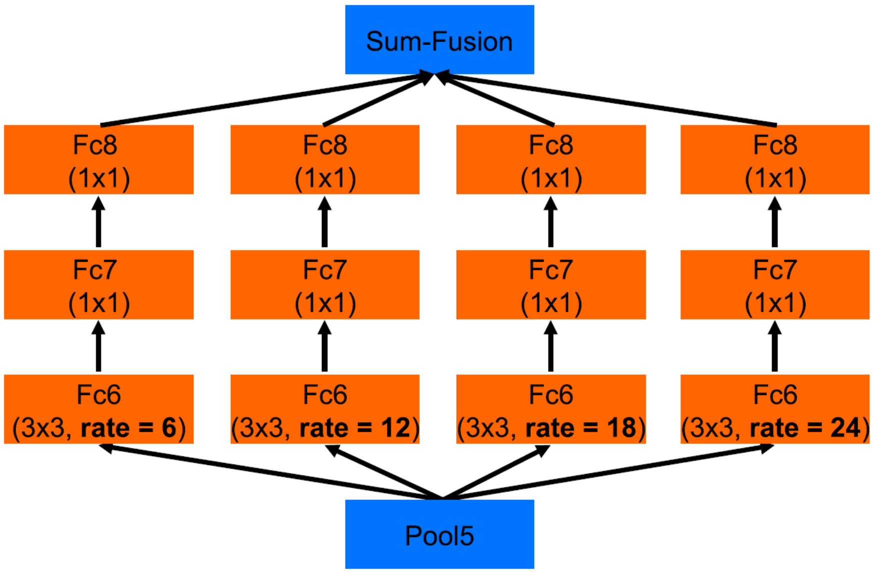
Semantic image segmentation with deep convolutional nets and fully connected CRFs, ICLR 2015

DeepLabv2

Based o VGG16



(a) DeepLab-LargeFOV



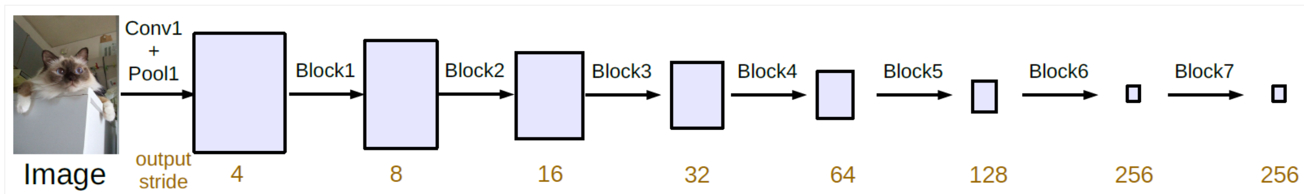
(b) DeepLab-ASPP (Atrous Spatial Pyramid Pooling)

- The other improvement strategy is to replace the backbone with ResNet-101

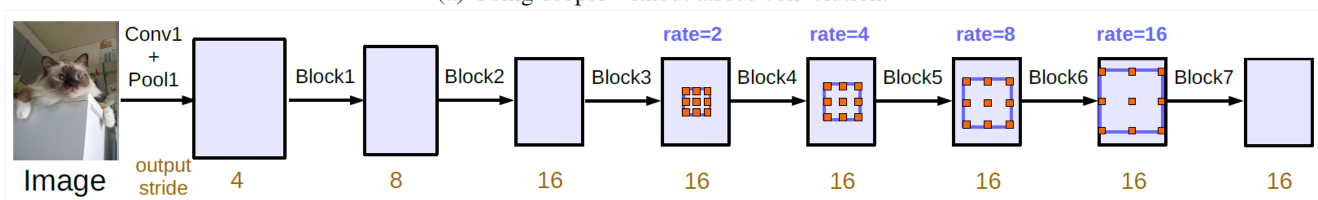
L.-C. Chien, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolution nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, Apr. 2018.

DeepLabv3

- Based on ResNet, go deeper

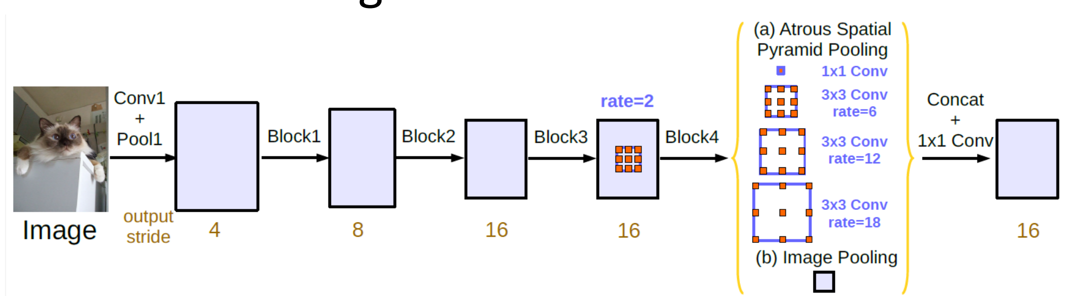


(a) Going deeper without atrous convolution.



(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

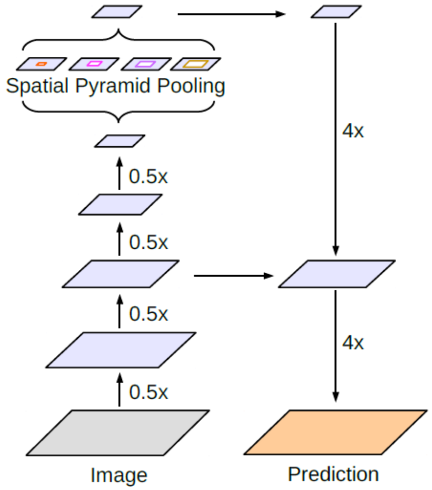
- ASPP augmented with image-level features

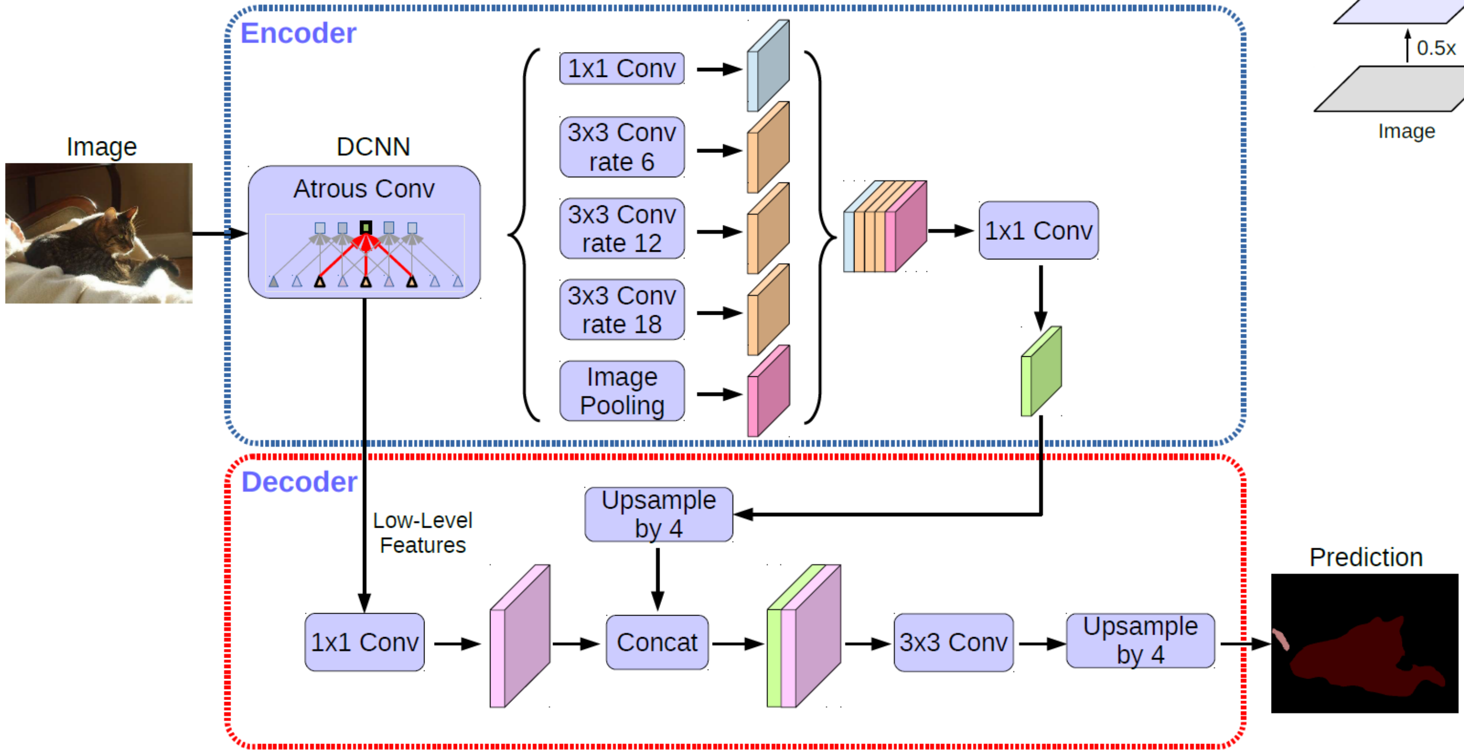


L.-C. Chien, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic segmentation,” *arXiv:1706.05587v3*, Dec. 2017.

DeepLabv3+

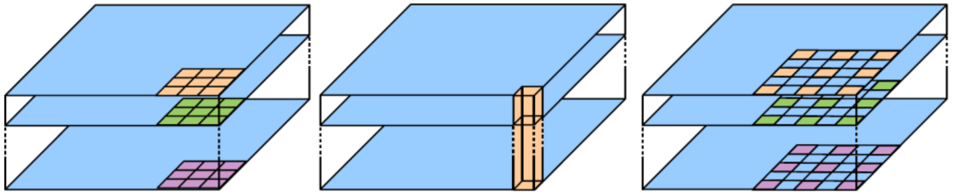
- Encoder-decoder architecture



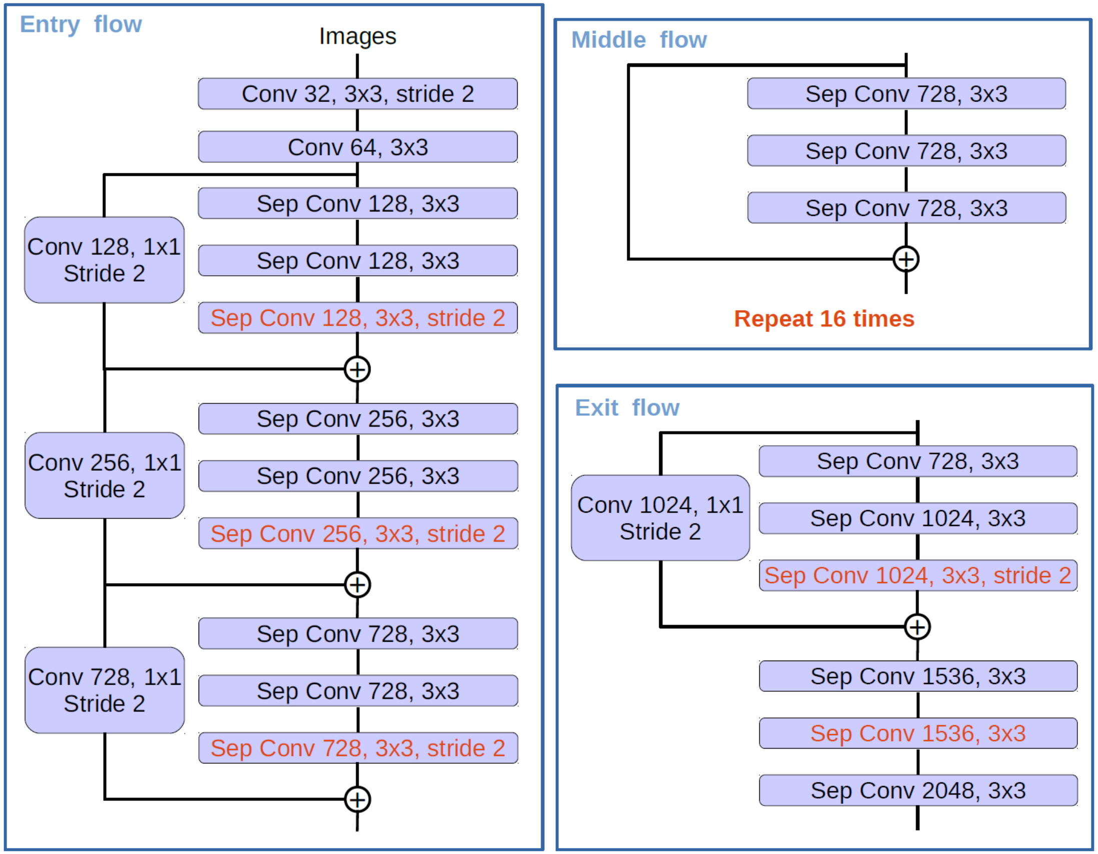


L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *ECCV 2018*.

DeepLabv3+



- Xception



L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *ECCV 2018*.

DeepLabv3+

- Test results on PASCAL VOC 2012

Method	mIOU
Deep Layer Cascade (LC) [82]	82.7
TuSimple [77]	83.1
Large_Kernel_Matters [60]	83.6
Multipath-RefineNet [58]	84.2
ResNet-38_MS_COCO [83]	84.9
PSPNet [24]	85.4
IDW-CNN [84]	86.3
CASIA_IVA_SDN [63]	86.6
DIS [85]	86.8
DeepLabv3 [23]	85.7
DeepLabv3-JFT [23]	86.9
DeepLabv3+ (Xception)	87.8
DeepLabv3+ (Xception-JFT)	89.0

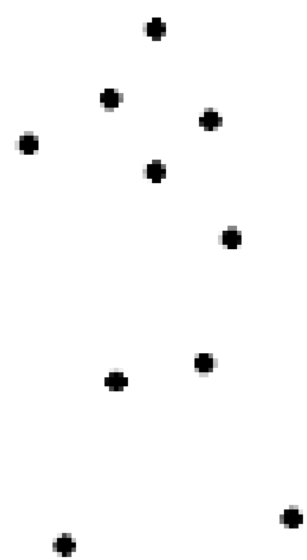
L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *ECCV 2018*.

Motion and Perceptual Organization

- Sometimes, motion is foremost cue

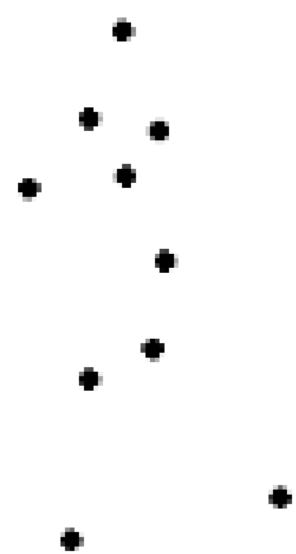
Motion and Perceptual Organization

- Even “impoverished” motion data can evoke a strong percept



Motion and Perceptual Organization

- Even “impoverished” motion data can evoke a strong percept



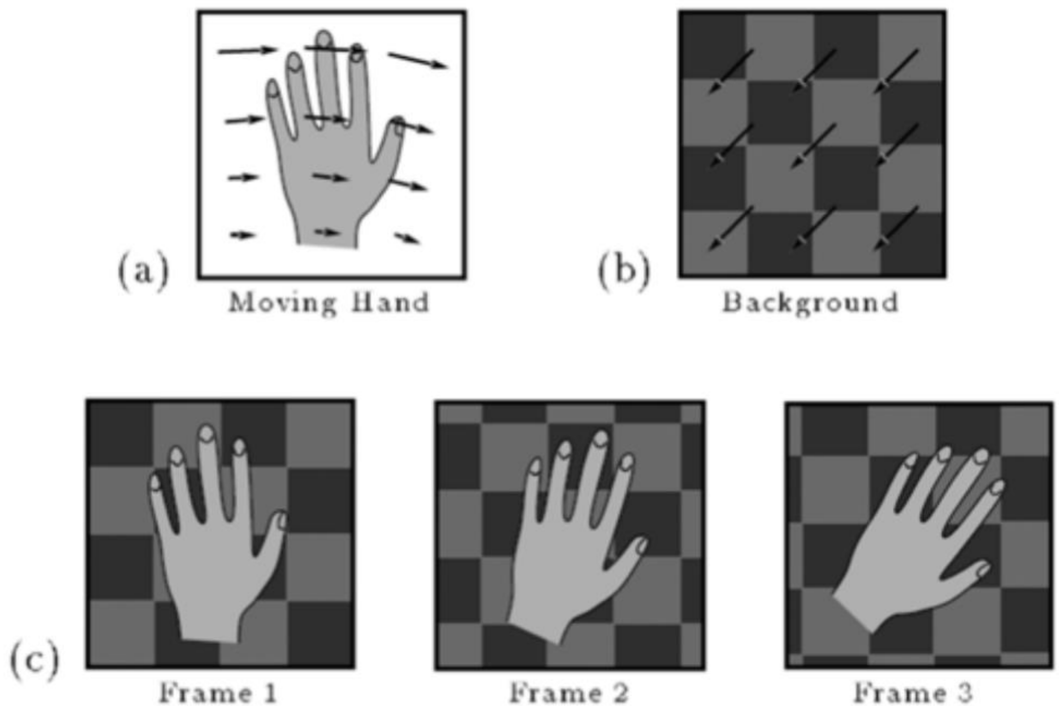
Video Segmentation: Segmentation in Motion Field

- Break image sequence into “layers” each of which has a coherent (affine) motion

Ref: J. Wang and E. Adelson, “Layered Representation for Motion Analysis,” CVPR 1993

Video Segmentation: Segmentation in Motion Field

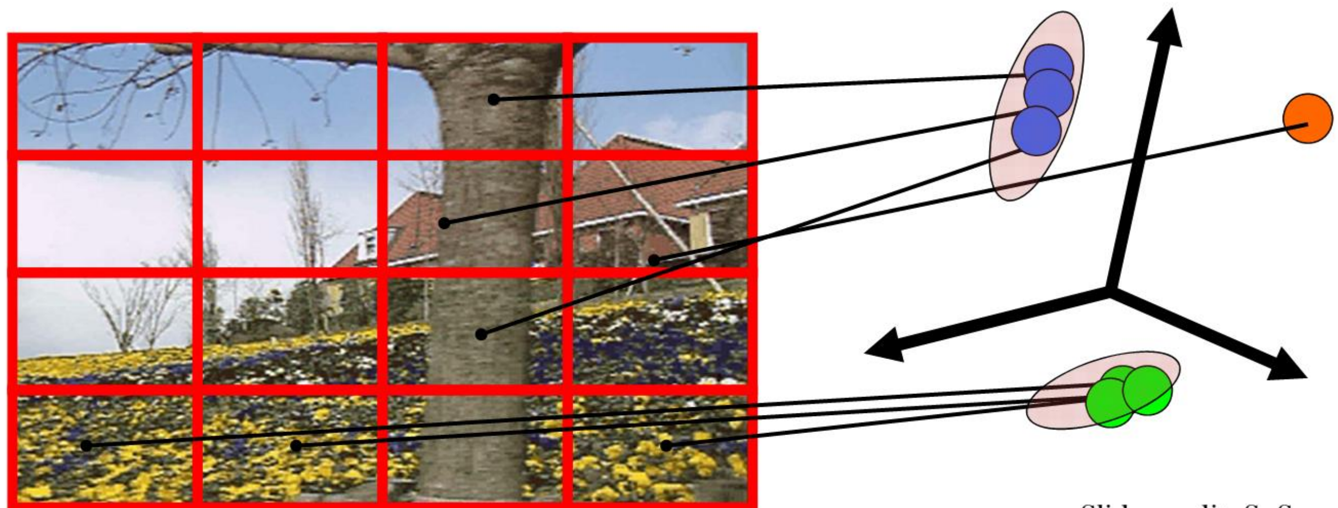
- What are layers?
  - Each layer is defined by an alpha mask and a motion model (such as affine model)



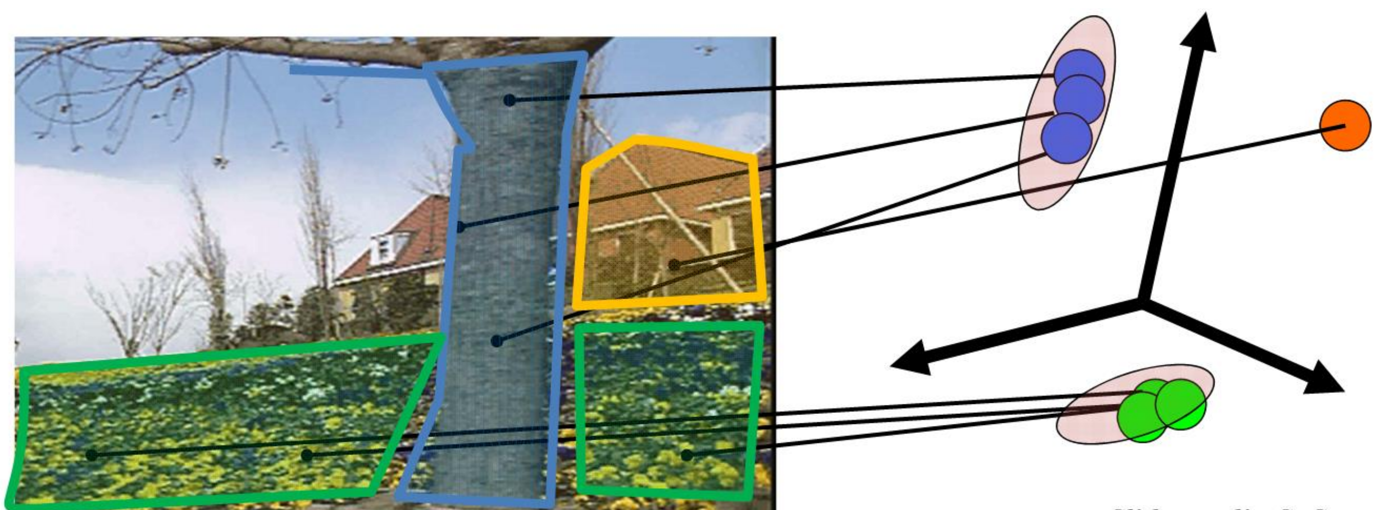
Video Segmentation: Segmentation in Motion Field

- 1. Obtain a set of initial affine motion hypotheses
  - Divide the image into blocks and estimate affine motion parameters in each block by least squares
    - Eliminate hypotheses with high residual error
  - Map into motion parameter space
  - Perform k-means clustering on affine motion parameters
    - Merge clusters that are close and retain the largest clusters to obtain a smaller set of hypotheses to describe all the motions in the scene
- 2. Iterate until convergence:
  - Assign each pixel to best hypothesis
    - Pixels with high residual error remain unassigned
  - Perform region filtering to enforce spatial constraints
    - Re-estimate affine motions in each region

Video Segmentation: Segmentation in Motion Field

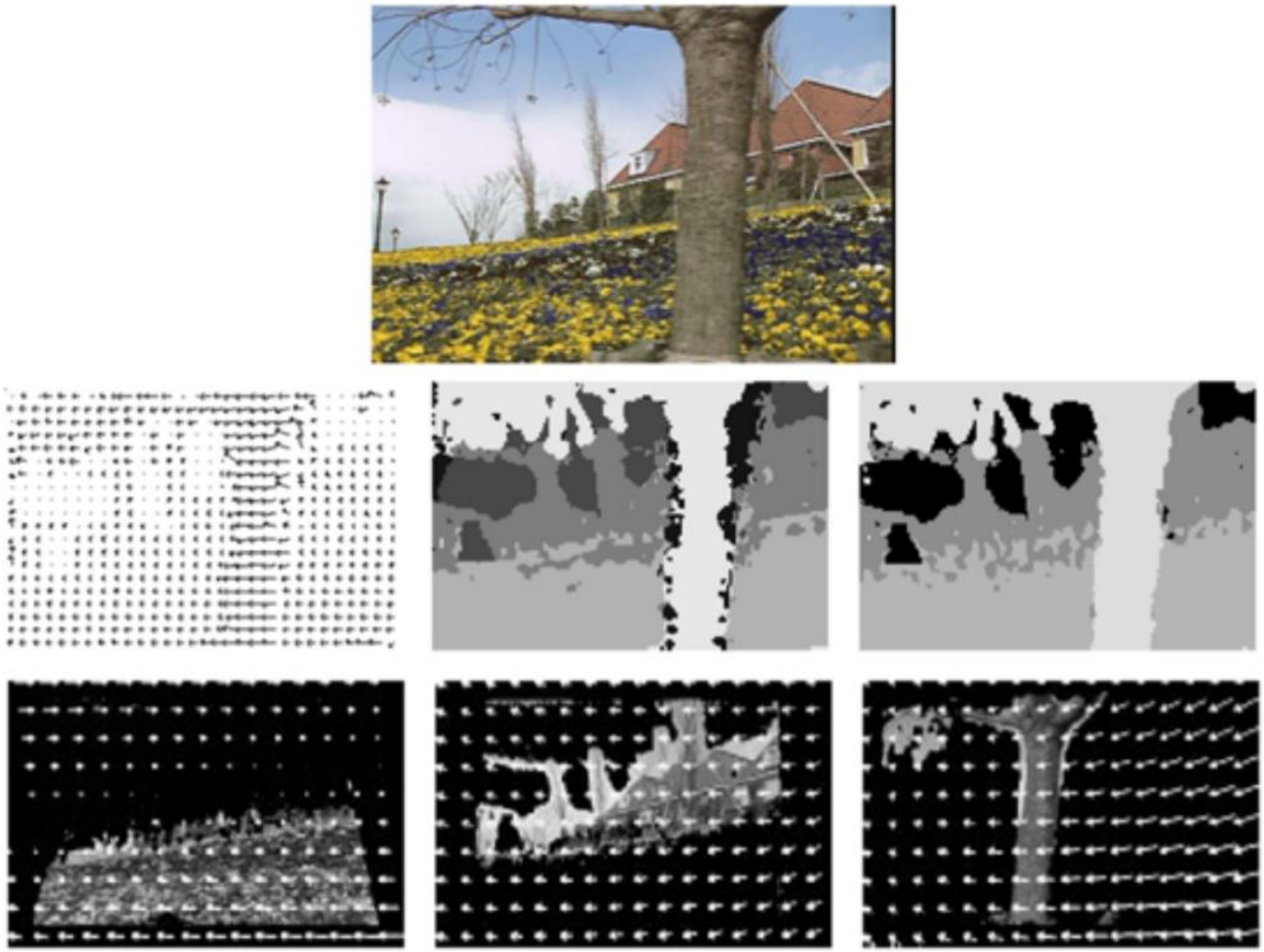


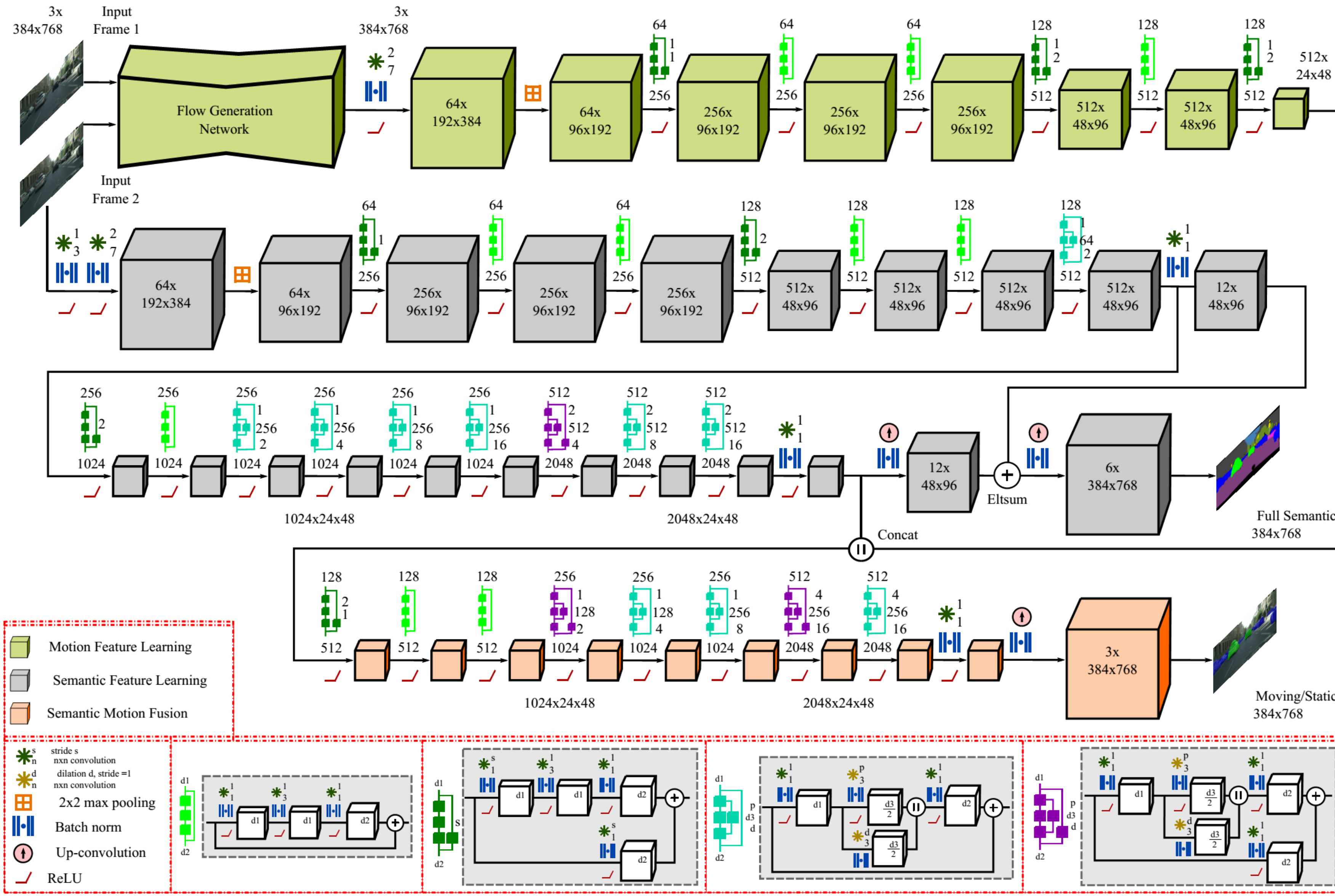
Slide credit: S. Savarese



Slide credit: S. Savarese

Video Segmentation: Segmentation in Motion Field

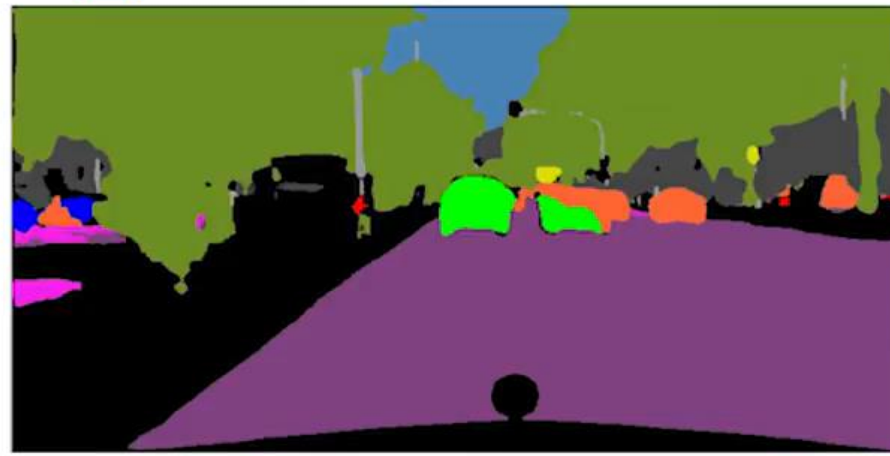




Ref: J. Vertens, A. Valada, and W. Burgard, "SMSnet: Semantic Motion Segmentation using Deep Convolutional Neural Networks," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, Canada, 2017.

Qualitative Results- Cityscapes

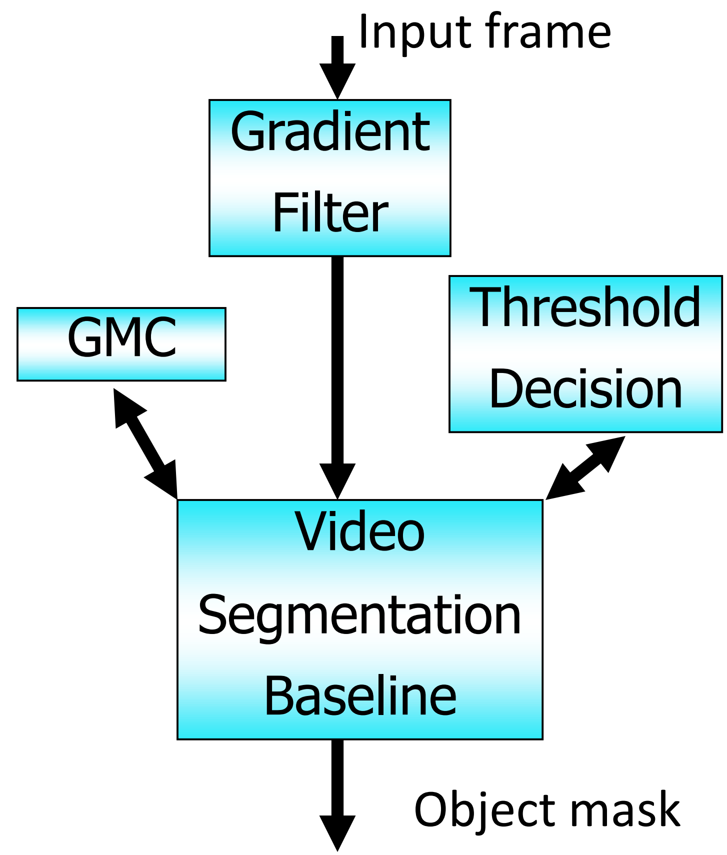
Driving Car   Static Car   Car behind max-range



Model trained with a maximum-range of 40m and EFS. All presented results are achieved by training SMSnet on City-KITTI-Motion.

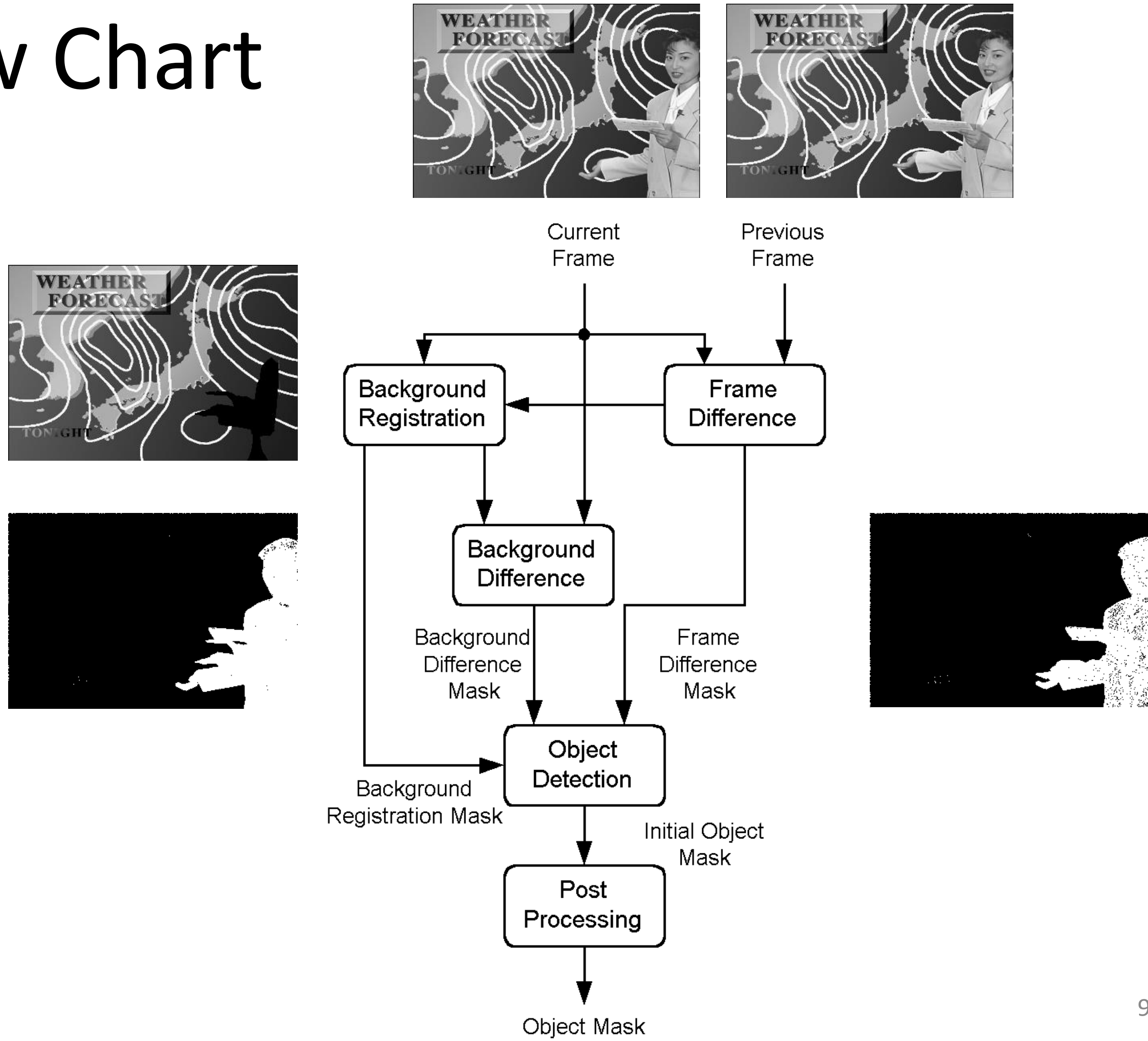
Video Segmentation: Change Detection Method

- Background substraction
- 4 modes
  - Baseline mode
  - Shadow cancellation mode (SC mode)
  - Global motion compensation mode (GMC mode)
  - Adaptive threshold mode (AT mode)



Ref: Shao-Yi Chien, Yu-Wen Huang, Bing-Yu Hsieh, Shyh-Yih Ma, and Liang-Gee Chen, "Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques," *IEEE Transactions on Multimedia*, vol. 6, no. 5, pp. 732--748, Oct 2004.
Shao-Yi Chien, Shyh-Yih Ma, and Liang-Gee Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 577 –586, July 2002.

Flow Chart


Current Frame
Previous Frame
Background Registration
Frame Difference
Background Difference
Background Difference Mask
Frame Difference Mask
Object Detection
Background Registration Mask
Initial Object Mask
Post Processing
Object Mask

Background Registration

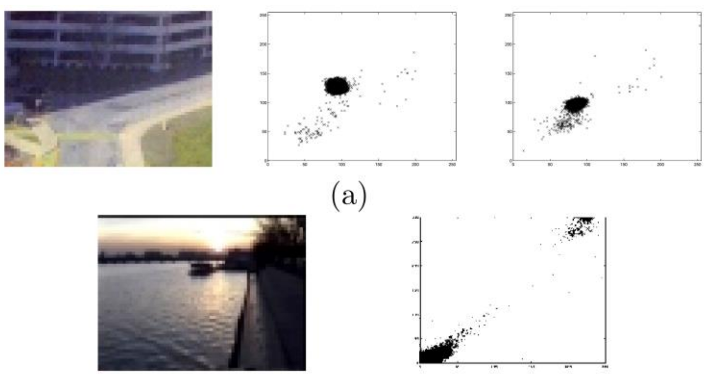
Segmentation Results

Segmentation Results

Video Segmentation: Change Detection Method

- Background modeling with Gaussian Mixture Model (GMM)

Variation of background information



(a)

(b)

- Background information is modeled as:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

- Every new pixel value, X_t, is checked against the existing K Gaussian distributions, until a match is found. A match is defined as a pixel value within 2.5 standard deviations of a distribution.

- Background model updating:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t)$$

where

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k)$$

Ref: Chris Stauffer W.E.L G rimson, “Adaptive b ackground mixture mo dels for real-time tracking,” CVPR1998.