Computer Vision – TP12
Advanced Segmentation

Miguel Coimbra, Francesco Renna

U. PORTO FC

Outline

- Segmentation by Fitting
- Active Contours
- Semantic Segmentation

Topic: Segmentation by Fitting

- Segmentation by Fitting
- Active Contours
- Semantic Segmentation

Fitting and Clustering

- Another definition for segmentation:
  - Pixels belong together because they conform to some model
- Sounds like "Segmentation by Clustering"...
- Key difference:
  - The model is now **explicit**

We have a mathematical model for the object we want to segment.
Ex: A line

Hough Transform

- Elegant method for direct object recognition
- Edges need not be connected
- Complete object need not be visible
- Key Idea: Edges **VOTE** for the possible model

Image and Parameter Spaces

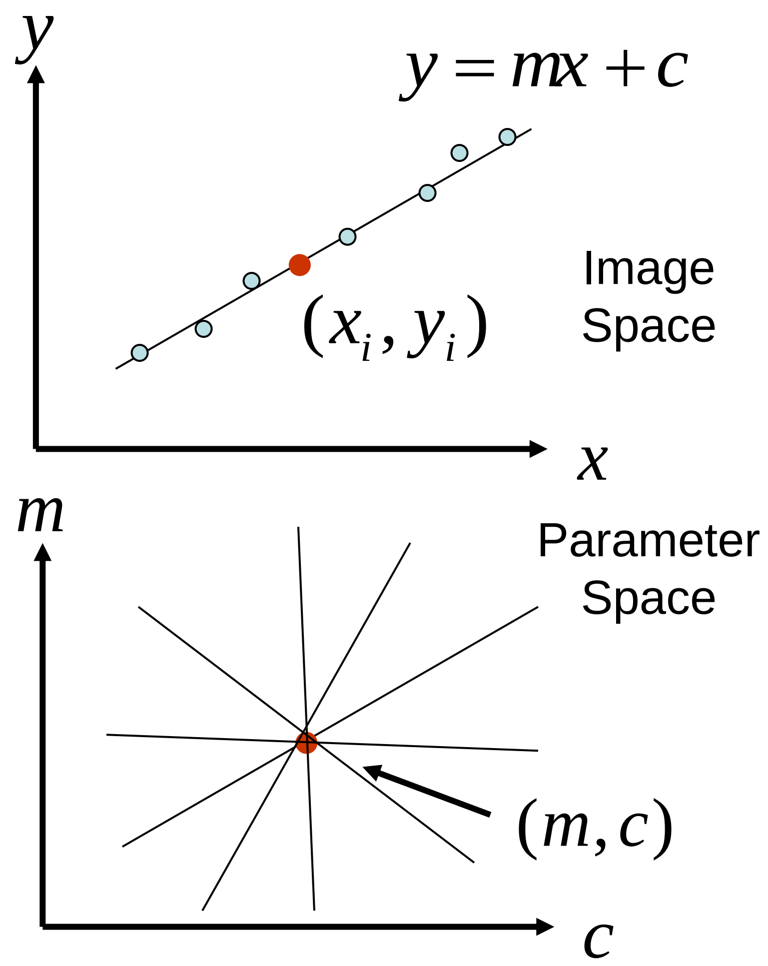
Equation of Line: $y = mx + c$

Find: (m, c)

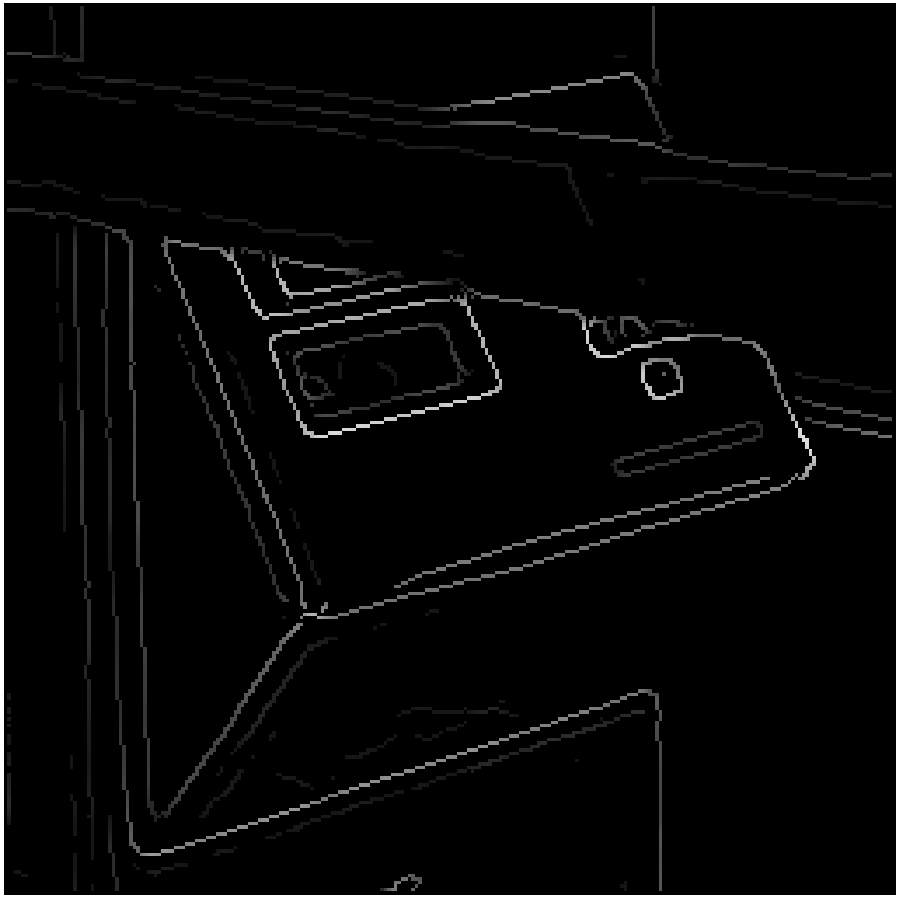
Consider point: (x_i, y_i)

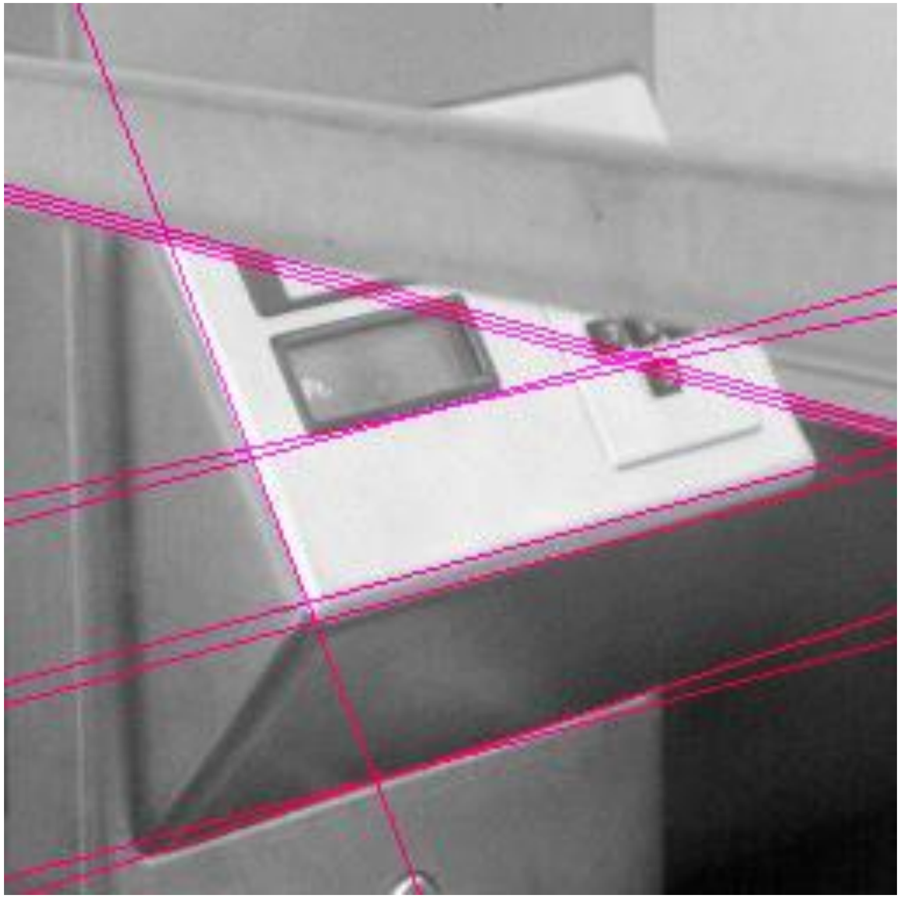
$$y_i = mx_i + c \quad or \quad c = -x_i m + y_i$$

Parameter space also called Hough Space

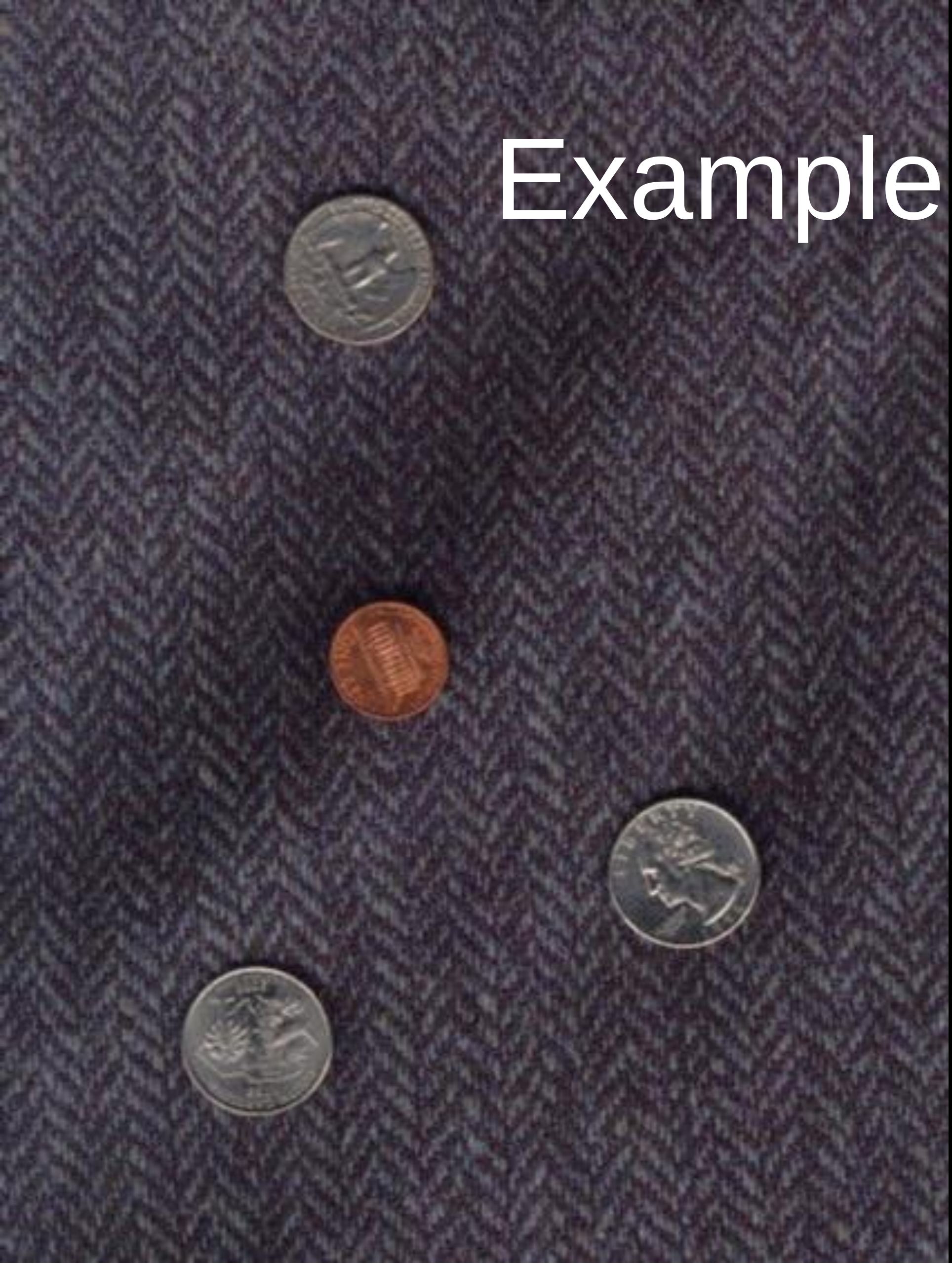


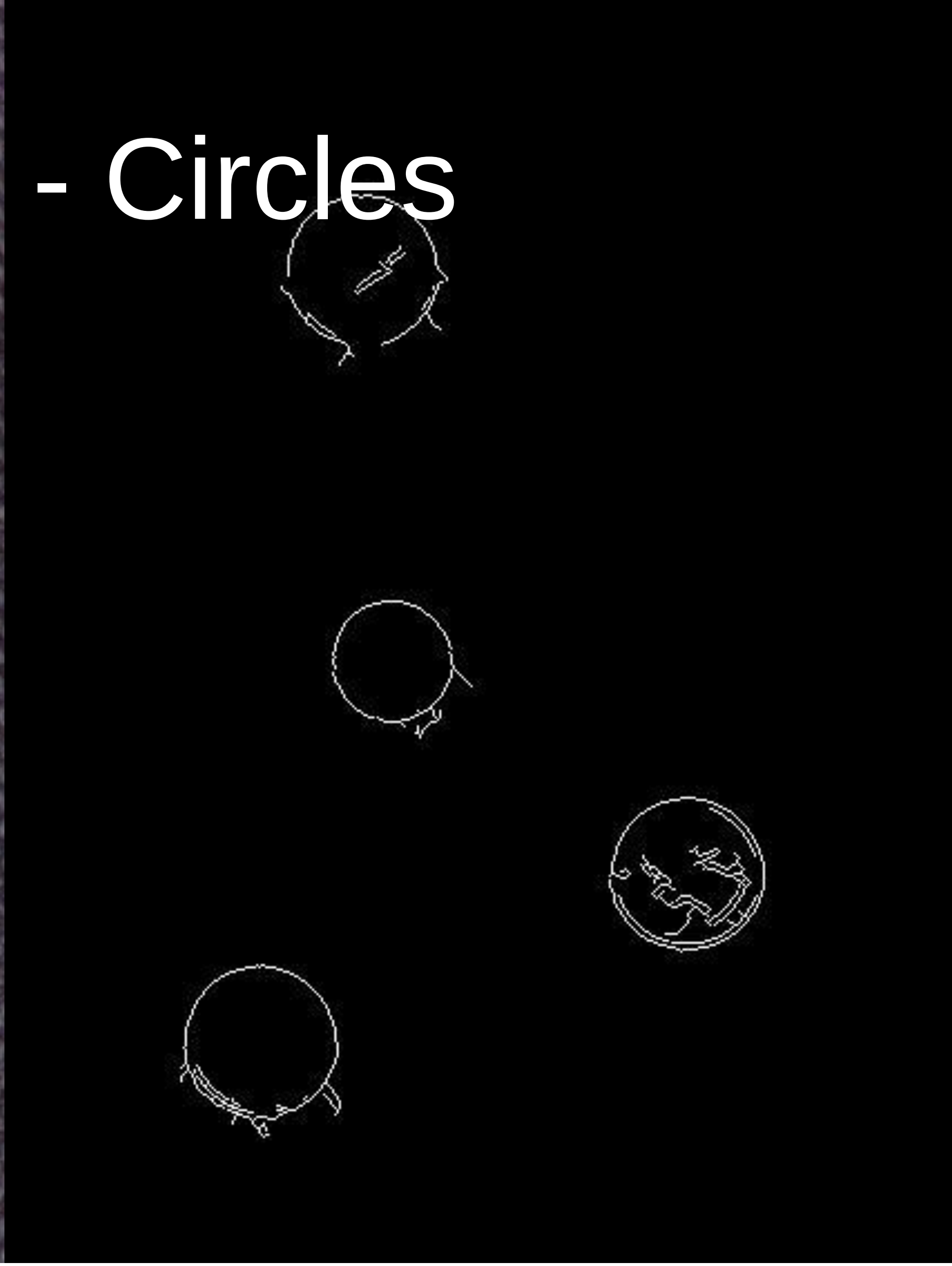
Example - Lines





Example - Circles



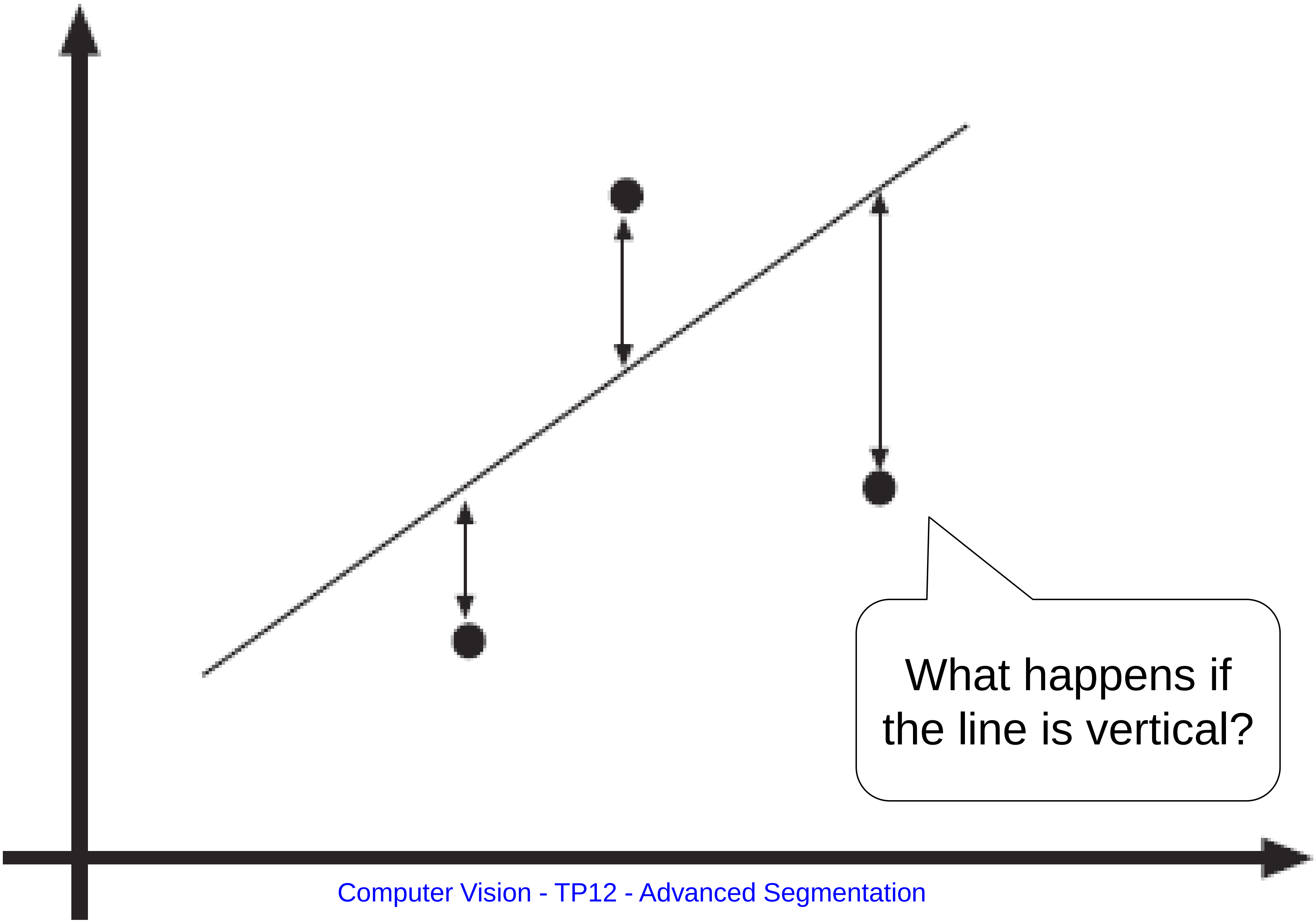


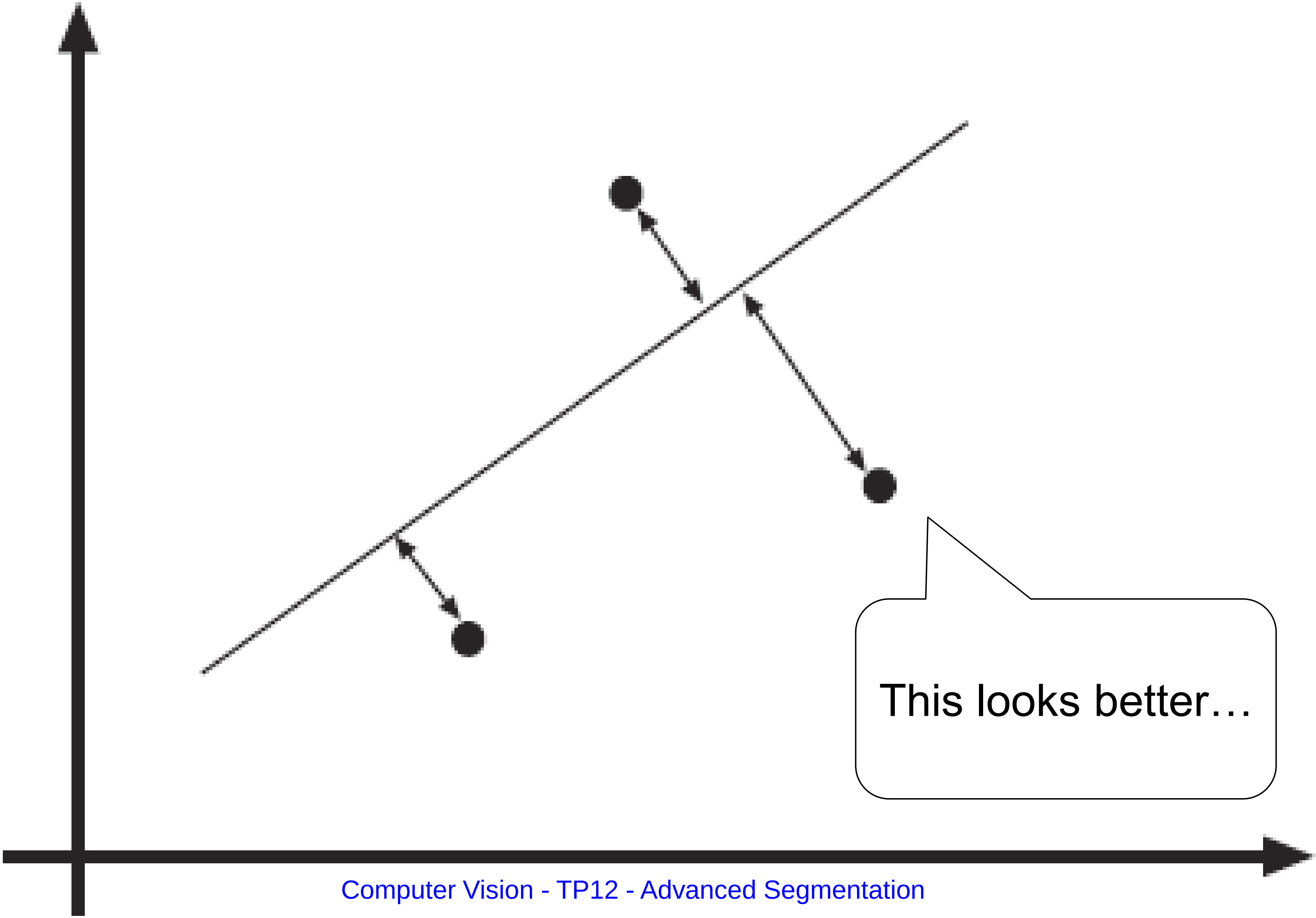
Least Squares Line Fitting

- Popular fitting procedure
- Simple but biased (why?)
- Consider a line:

$$y = ax + b$$

- What is the line that best predicts all observations (x_i, y_i)?
  – Minimize: $$\sum_i (y_i - ax_i - b)^2$$


What happens if the line is vertical?


This looks better…

Total Least Squares

- Works with the actual distance between the point and the line (rather than the vertical distance)
- Lines are represented as a collection of points where:

$$ax + by + c = 0$$

- And:

$$a^2 + b^2 = 1$$

Again... Minimize the error, obtain the line with the 'best fit'.

Point correspondence

- We can estimate a line but, **which points are on which line**?
- Usually:
  - We are fitting lines to edge points, so…
  - Edge directions can give us hints!
- What if I only have isolated points?
- Let's look at two options:
  - Incremental fitting
  - Allocating points to lines with K-means

Incremental Fitting

- Start with connected *curves* of edge points
- Fit *lines* to those points in that curve
- Incremental fitting:
  - Start at one end of the *curve*
  - Keep fitting all points in that curve to a line
  - Begin another line when the fitting deteriorates too much
- Great for closed curves!

```
Put all points on curve list, in order along the curve
empty the line point list
empty the line list

Until there are two few points on the curve
  Transfer first few points on the curve to the line point list
  fit line to line point list

  while fitted line is good enough
    transfer the next point on the curve
    to the line point list and refit the line
  end

  transfer last point back to curve
  attach line to line list
end
```

K-means allocation

- What if points carry no hints about which line they lie on?
- Assume there are *k* lines for the *x* points.
- Minimize:

$$\sum_{\text{lines}} \sum_{\text{points}} dist(\text{line}, \text{point})^2$$

- Iteration:
  - Allocate each point to the closest line
  - Fir the best line to the points allocated to each line

Hypothesize k lines (perhaps uniformly at random)

or

hypothesize an assignment of lines to points
and then fit lines using this assignment

Until convergence
  allocate each point to the closest line
  refit lines

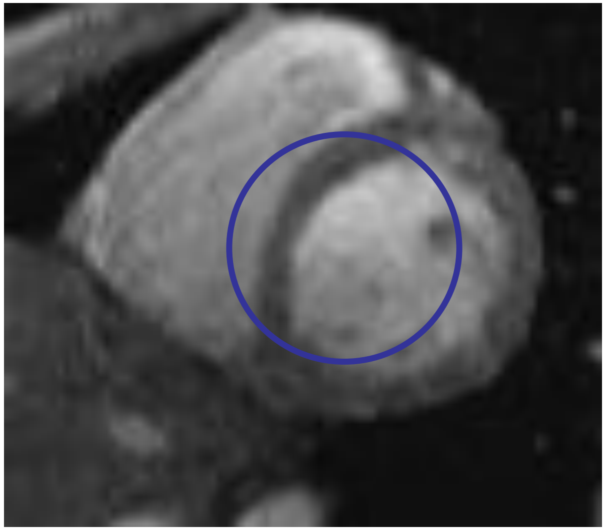
Topic: Active Contours

- Segmentation by Fitting
- Active Contours
- Semantic Segmentation

Acknowledgement: Slides adapted from Kristen Grauman

Active Contours

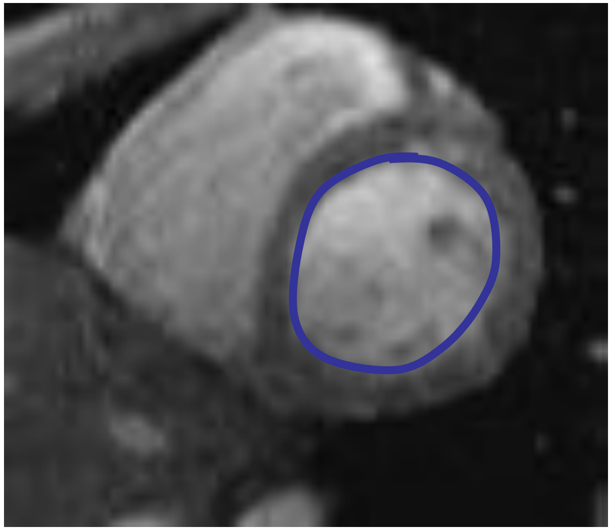
- Given: initial contour (model) near desired object



Acknowledgement: Slides adapted from Kristen Grauman

Active Contours

- Goal: evolve the contour to fit exact object boundary

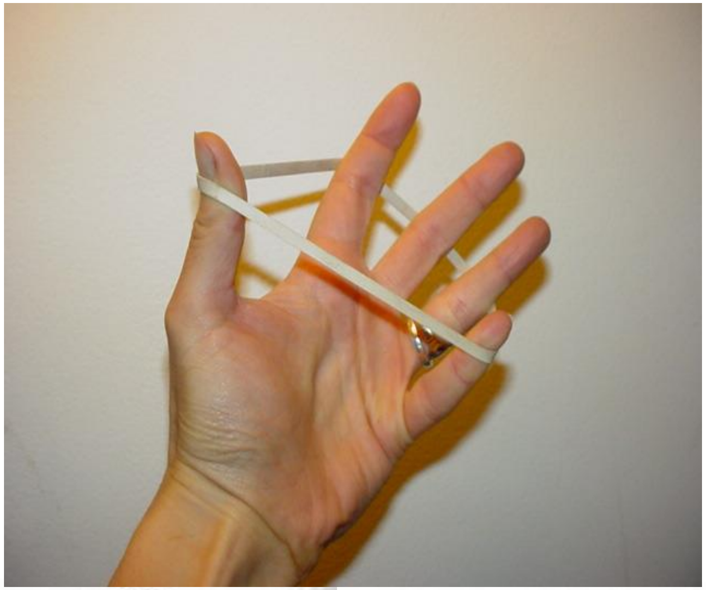


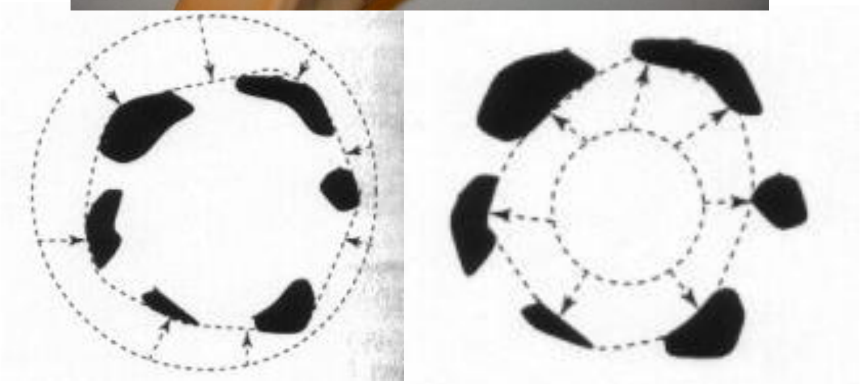
- How?
  - Reward solutions next to high image gradients
  - Punish solutions that deform shape too much
  - Iteratively find the ‘best’ solution to these requirements

Acknowledgement: Slides adapted from Kristen Grauman

Intuition - Elastic Band

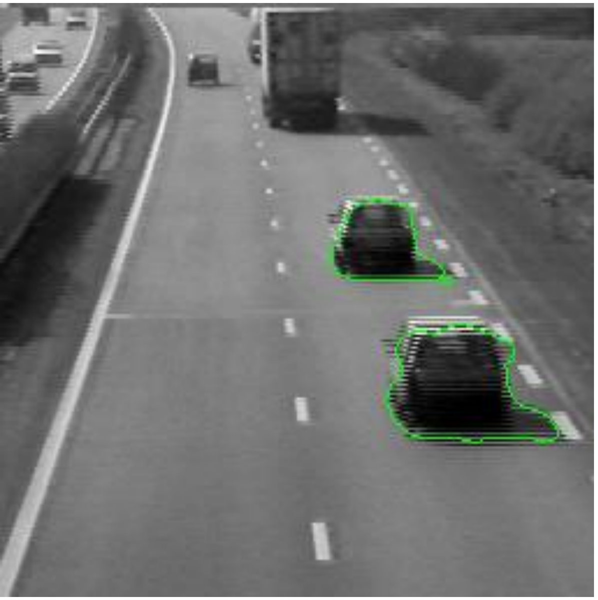
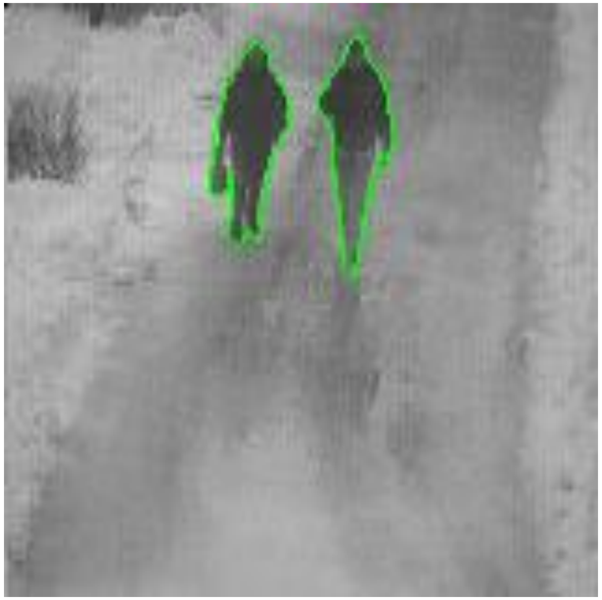
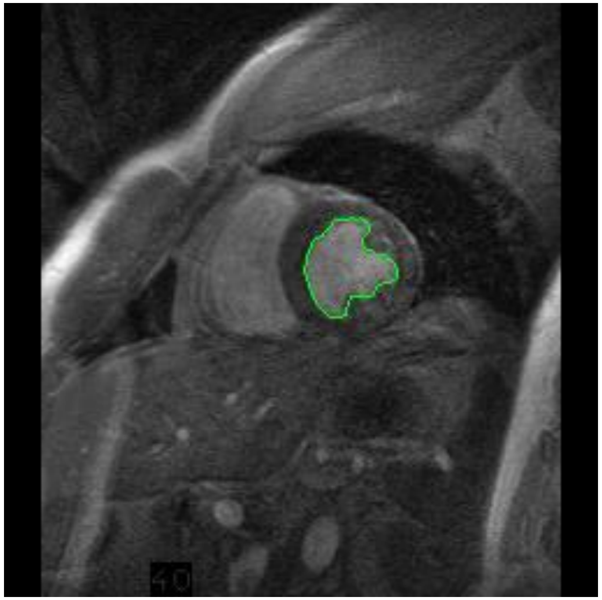
- Contour evolves to a low-energy solution, but is hindered by obstacles
- Better intuition: Gravity
  - Contour is ‘attracted’ to specific image features
  - Contour resists to any deformation of its shape





Acknowledgement: Slides adapted from Kristen Grauman

Strong motivation – Moving deformable objects

Acknowledgement: Slides adapted from Kristen Grauman

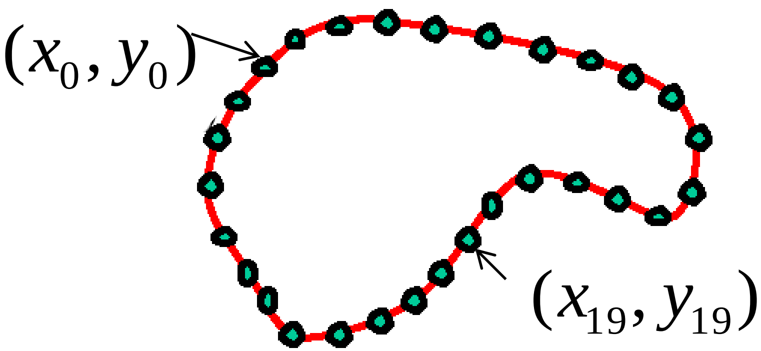
Things we need to consider

- Representation of the contours
- Defining the energy functions
  - External
  - Internal
- Minimizing the energy function

Acknowledgement: Slides adapted from Kristen Grauman

Representation

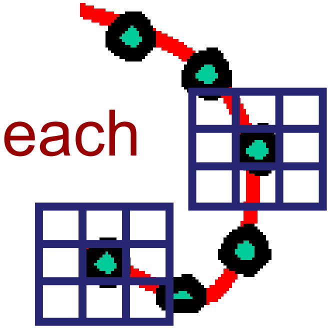
- We'll consider a discrete representation of the contour, consisting of a list of 2d point positions ("vertices")



$$v_i = (x_i, y_i),$$

$$\text{for} \quad i = 0, 1, \ldots, n-1$$

- At each iteration, we'll have the option to move each vertex to another nearby location ("state")

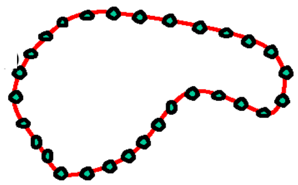


Acknowledgement: Slides adapted from Kristen Grauman

Energy function

The total energy (cost) of the current snake is defined as:

$$E_{total} = E_{external} + E_{internal}$$

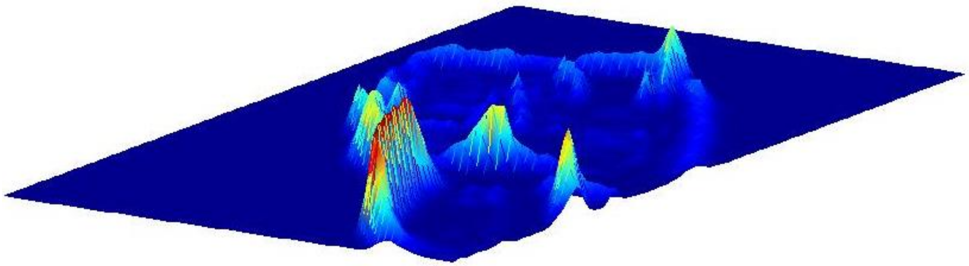


- External energy: encourage contour to fit on places where specific image structures exist
- Internal energy: encourage prior shape preferences

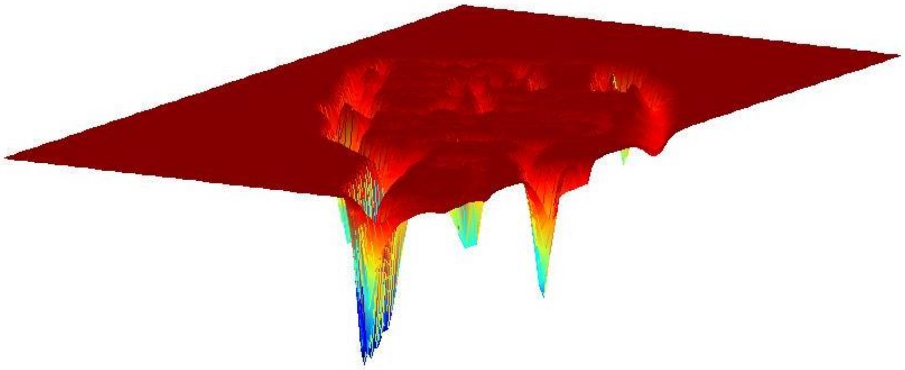
A good fit between the current deformable contour and the target shape in the image will yield a low value for this cost function

Acknowledgement: Slides adapted from Kristen Grauman

External image energy



Magnitude of gradient $G_x(I)^2 + G_y(I)^2$

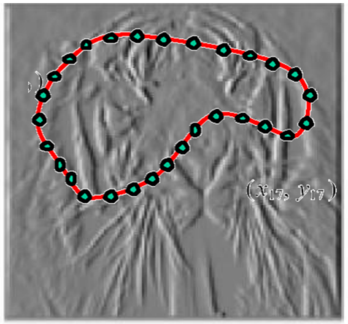


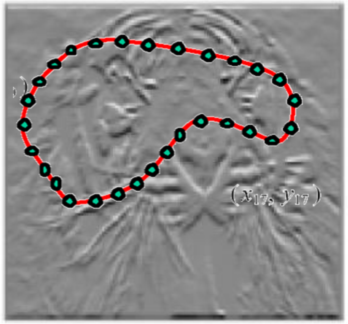
- (Magnitude of gradient) $-\left(G_x(I)^2 + G_y(I)^2\right)$

Acknowledgement: Slides adapted from Kristen Grauman

External image energy

- Gradient images $G_x(x, y)$ and $G_y(x, y)$





- External energy at a point on the curve is:

$$E_{external}(\nu) = -(\,|\,G_x(\nu)\,|^2 + |\,G_y(\nu)\,|^2\,)$$

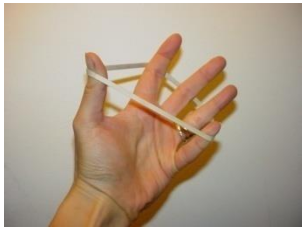
- External energy for the whole curve:

$$E_{external} = \; -\sum_{i=0}^{n-1} |\,G_x(x_i, y_i)\,|^2 + |\,G_y(x_i, y_i)\,|^2$$

Acknowledgement: Slides adapted from Kristen Grauman

Internal energy

For a *continuous* curve, a common internal energy term is the “bending energy”



At some point *v(s)* on the curve, this is:

$$E_{internal}(\nu(s)) = \alpha \left| \frac{d\nu}{ds} \right|^2 + \beta \left| \frac{d^2\nu}{d^2 s} \right|^2$$

Tension, Elasticity

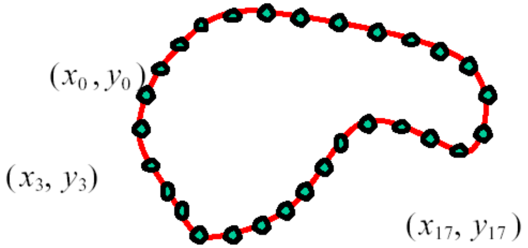
Stiffness, Curvature

Acknowledgement: Slides adapted from Kristen Grauman

Internal energy

- For our discrete representation:

$$v_i = (x_i, y_i) \qquad i = 0 \ldots n-1$$



$$\frac{dv}{ds} \approx v_{i+1} - v_i \qquad \frac{d^2v}{ds^2} \approx (v_{i+1} - v_i) - (v_i - v_{i-1}) = v_{i+1} - 2v_i + v_{i-1}$$

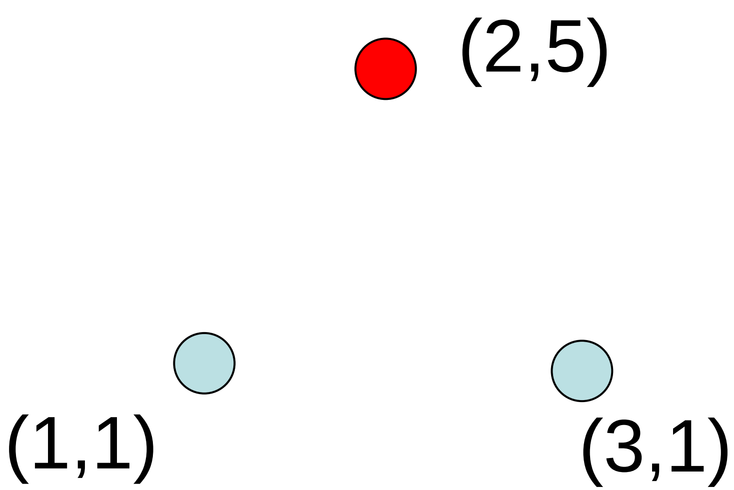
- Internal energy for the whole curve:

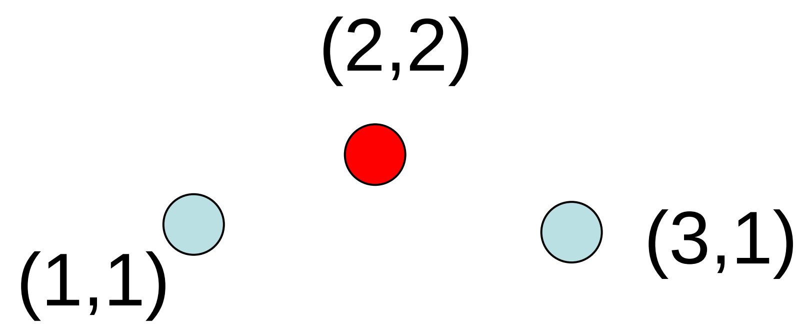
$$E_{internal} = \sum_{i=0}^{n-1} \alpha \left\| v_{i+1} - v_i \right\|^2 + \beta \left\| v_{i+1} - 2v_i + v_{i-1} \right\|^2$$

Acknowledgement: Slides adapted from Kristen Grauman

Example: compare curvature

$$E_{curvature}(v_i) = \left\| v_{i+1} - 2v_i + v_{i-1} \right\|^2$$
$$= (x_{i+1} - 2x_i + x_{i-1})^2 + (y_{i+1} - 2y_i + y_{i-1})^2$$





$$(3-2(2)+1)^2 + (1-2(5)+1)^2$$
$$= (-8)^2 = 64$$

$$(3-2(2)+1)^2 + (1-2(2)+1)^2$$
$$= (-2)^2 = 4$$

Acknowledgement: Slides adapted from Kristen Grauman

Penalizing elasticity

- Current elastic energy definition uses a discrete estimate of the derivative:

$$E_{elastic} = \sum_{i=0}^{n-1} \alpha \left\| v_{i+1} - v_i \right\|^2$$

- This rewards very small shapes!
- Instead -> Reward an ‘average distance ***d*** between pairs of points’

$$= \quad \alpha \cdot \sum_{i=0}^{n-1} \left((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 - \bar{d} \right)^2$$

Acknowledgement: Slides adapted from Kristen Grauman

Total energy

$$E_{total} = E_{internal} + \gamma E_{external}$$

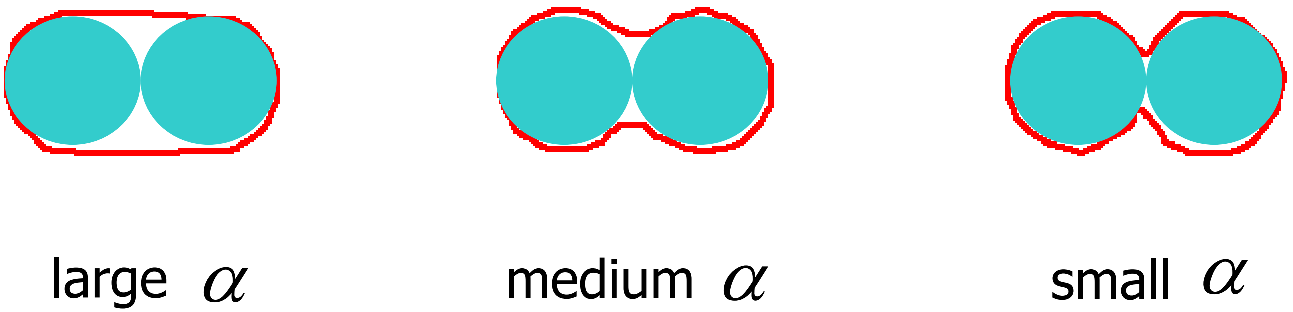
$$E_{external} = -\sum_{i=0}^{n-1} |G_x(x_i, y_i)|^2 + |G_y(x_i, y_i)|^2$$

$$E_{internal} = \sum_{i=0}^{n-1} \alpha\left(\bar{d} - \|v_{i+1} - v_i\|\right)^2 + \beta\|v_{i+1} - 2v_i + v_{i-1}\|^2$$

Acknowledgement: Slides adapted from Kristen Grauman

Energy weights

- e.g., α weight controls the penalty for internal elasticity



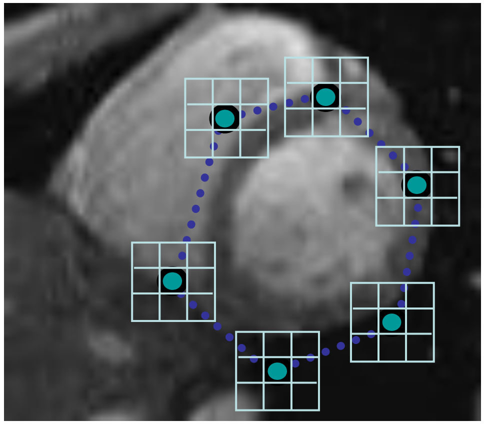
Acknowledgement: Slides adapted from Kristen Grauman

Energy minimization: greedy

- For each point, search window around it and move to where energy function is minimal
  - Typical window size, e.g., 5 x 5 pixels

- Stop when predefined number of points have not changed in last iteration, or after max number of iterations

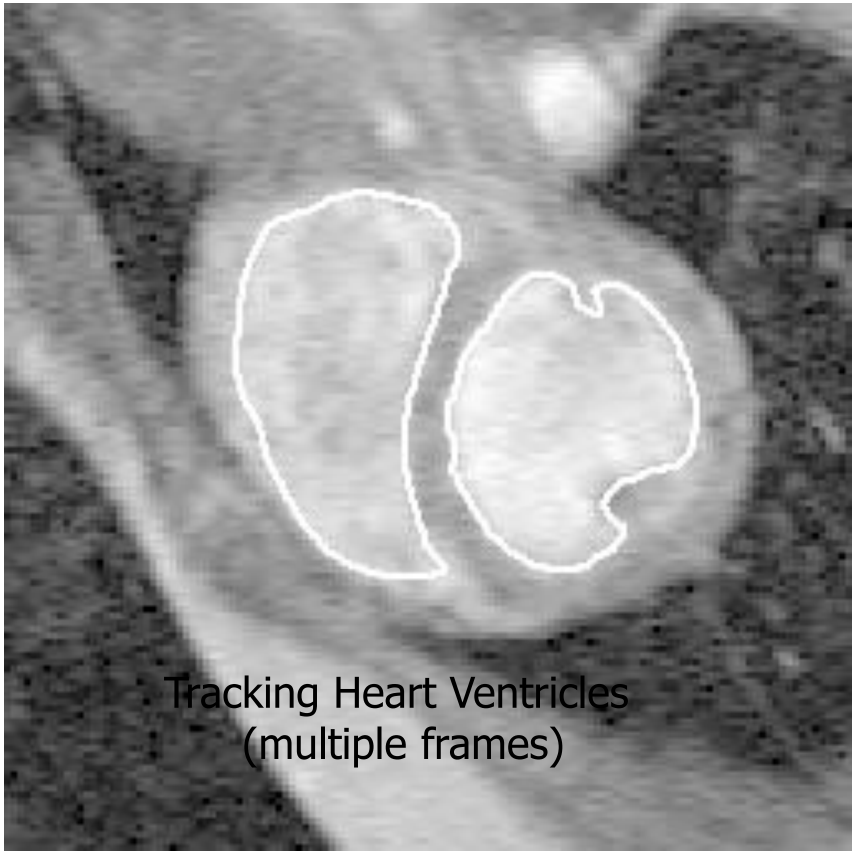
- Note:
  - Convergence not guaranteed
  - Need decent initialization



Acknowledgement: Slides adapted from Kristen Grauman

Tracking via deformable contours

1. Use final contour/model extracted at frame t as an initial solution for frame t+1
2. Evolve initial contour to fit exact object boundary at frame t+1
3. Repeat, initializing with most recent frame



Acknowledgement: Slides adapted from Kristen Grauman

Deformable contours: pros and cons

Pros:

- Useful to track and fit non-rigid shapes
- Contour remains connected
- Possible to fill in “subjective” contours
- Flexibility in how energy function is defined, weighted.

Cons:

- Must have decent initialization near true boundary, may get stuck in local minimum
- Parameters of energy function must be set well based on prior information

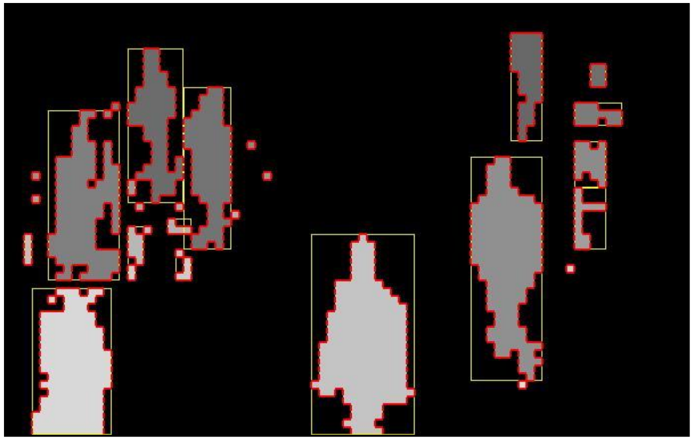
Acknowledgement: Slides adapted from Kristen Grauman

Topic: Semantic Segmentation

- Segmentation by Fitting
- Active Contours
- Semantic Segmentation

Remember 'Segmentation'?

- Separation of the image in different areas
  - Objects
  - **Areas with similar visual** or semantic **characteristics**

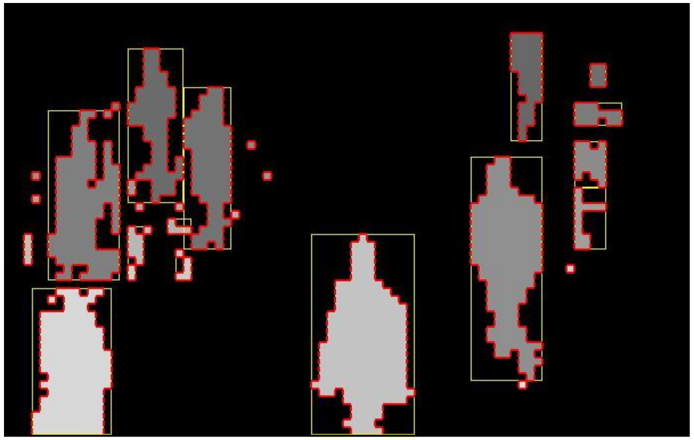


First form regions based on visual characteristics, then find the semantics of each region

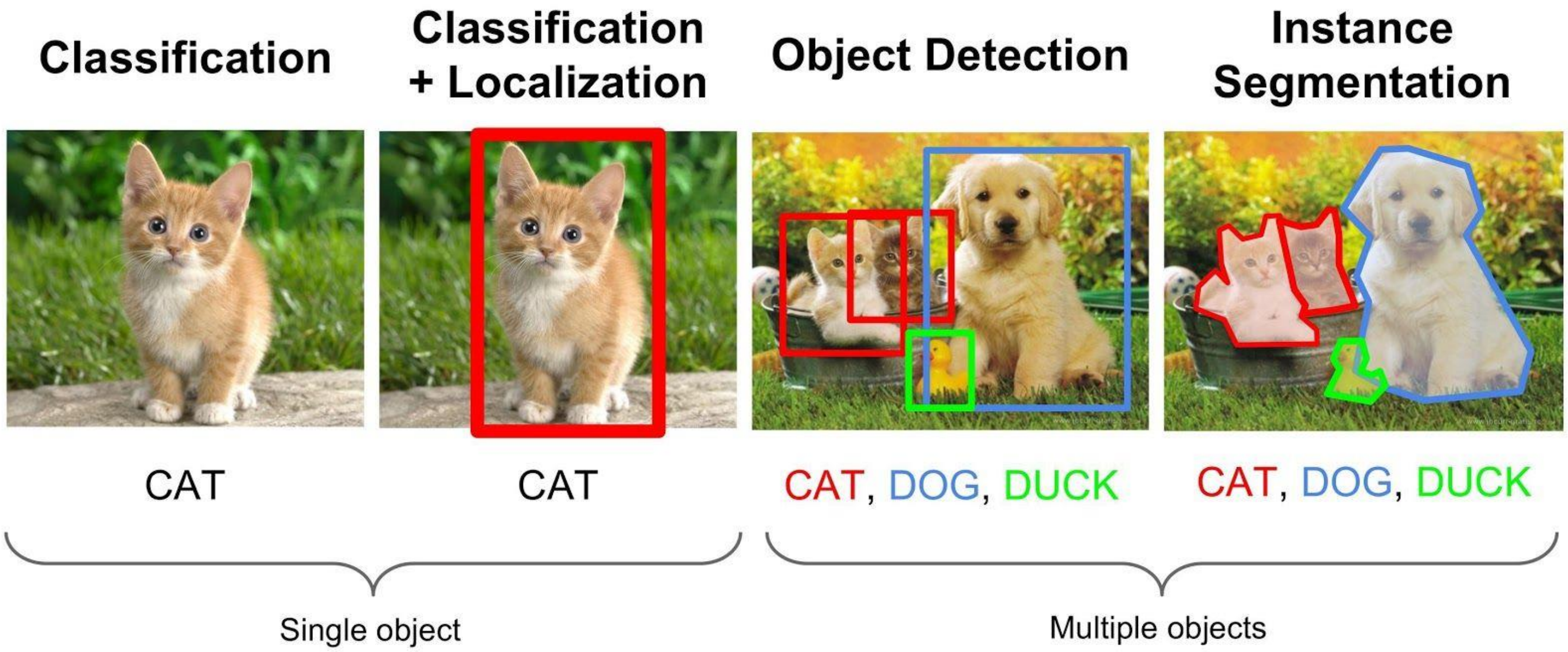
Semantic Segmentation

- Separation of the image in different areas
  - Objects
  - **Areas with similar** visual or **semantic characteristics**

First classify each pixel, and only then form regions (much harder!!)

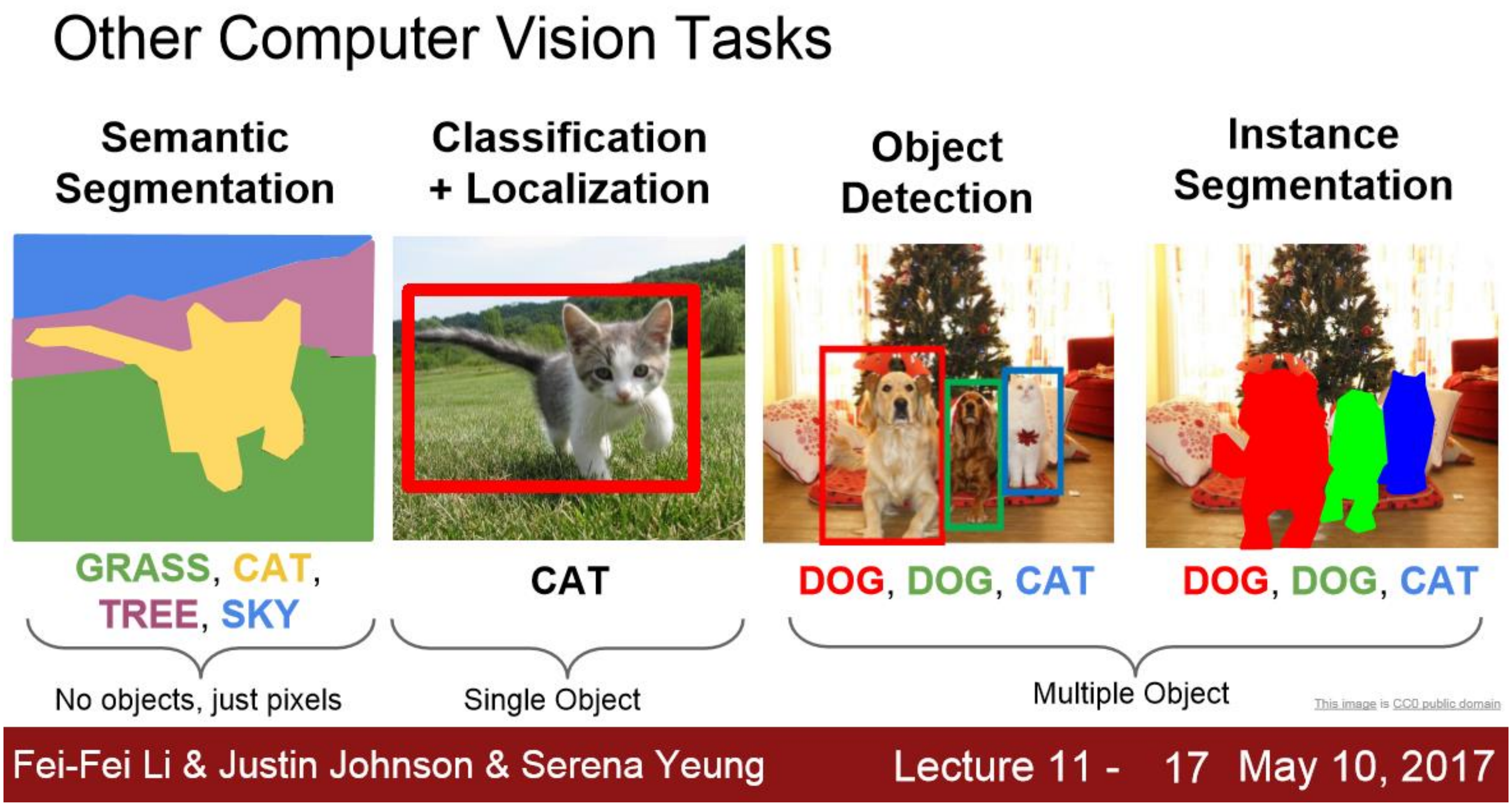


Classification and Segmentation



Source http://cs224d.stanford.edu/index.html

Semantic Segmentation



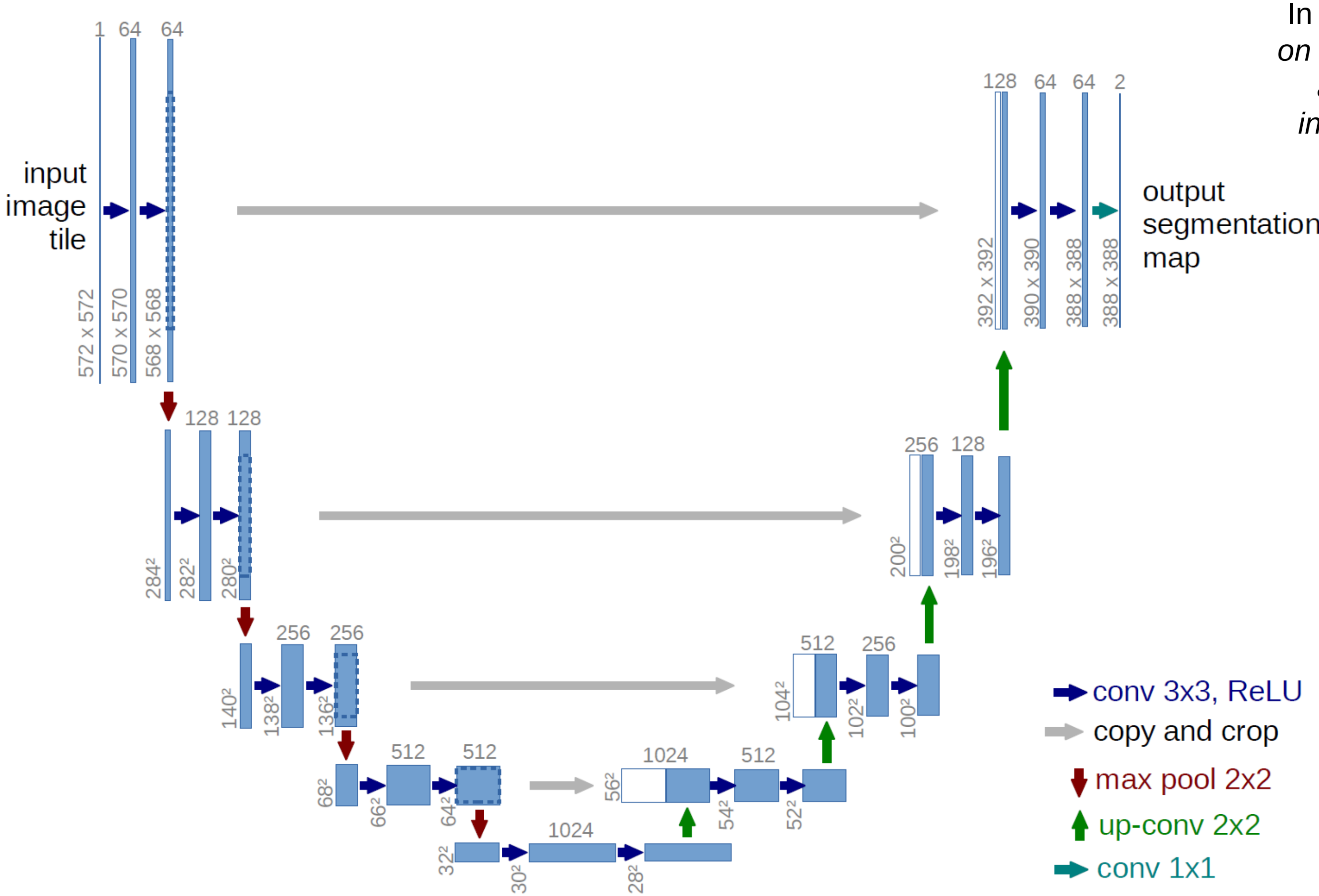
Source http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

Semantic Segmentation

- Requires sophisticated pixel-level classification algorithms to be effective
- Powerful data-based approach to segmentation
- Fueled by recent advances in deep neural networks, such as U-NET

U-Net

O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation." In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241. Springer, Cham, 2015.



- Encoder-decoder structure

Resources

- Szeliski, "Computer Vision: Algorithms and Applications", Springer, 2011
  - Chapter 5 – "Segmentation"