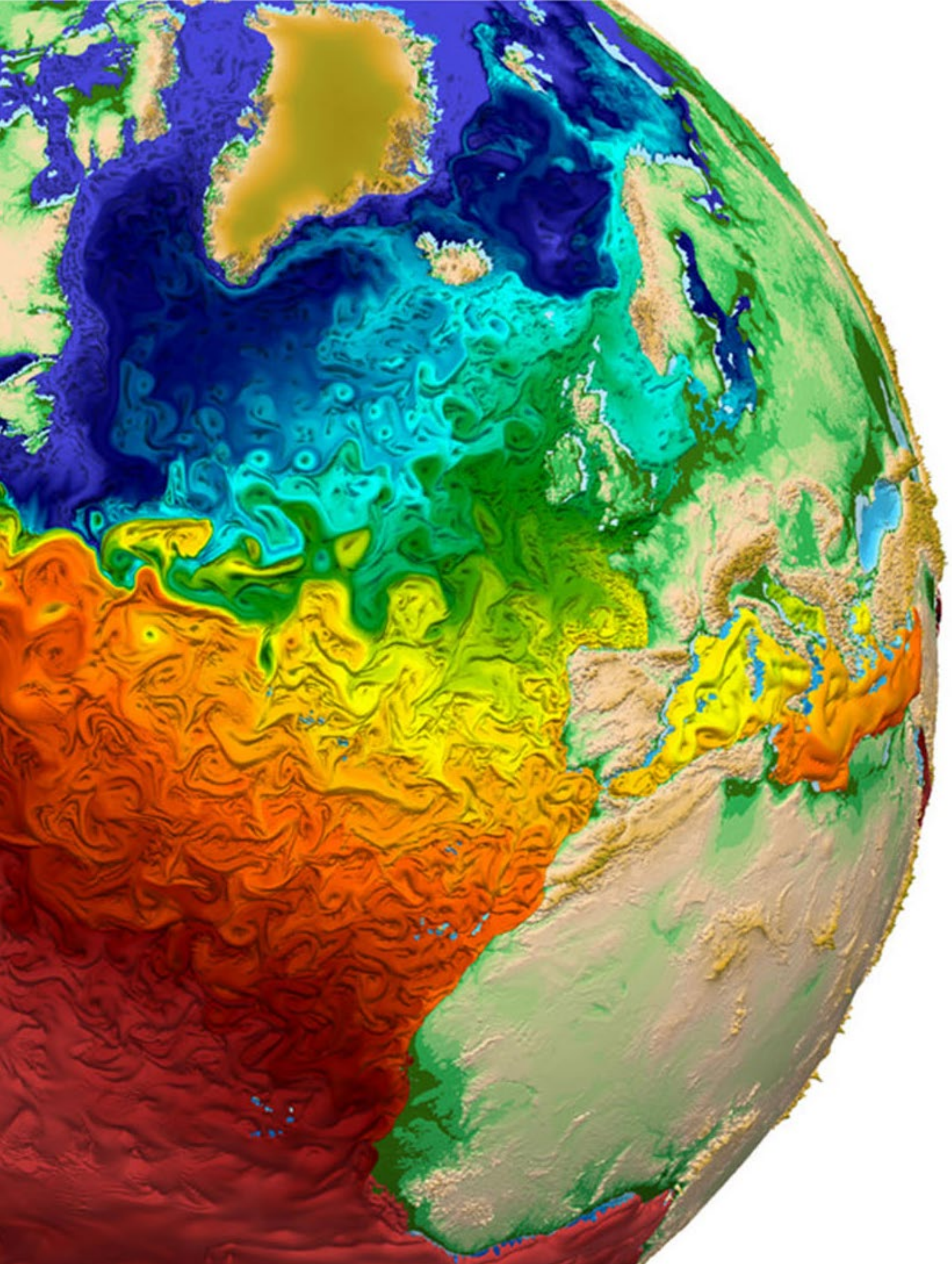


# Image Segmentation

## Interactive Graph Cuts

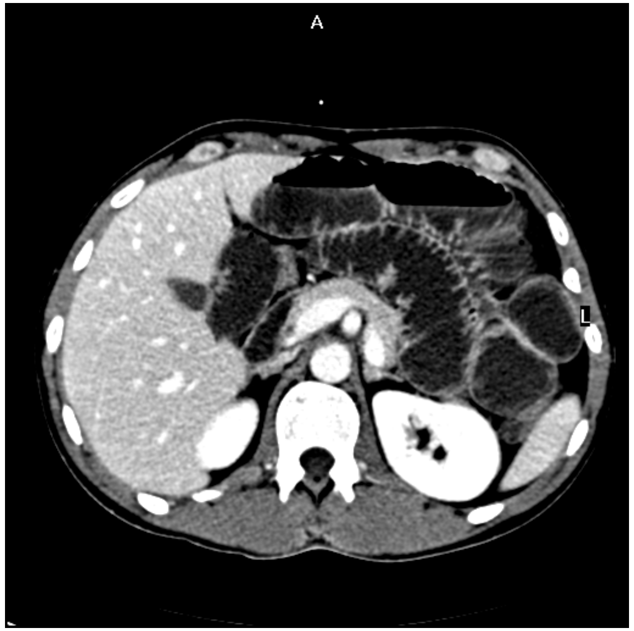
Scientific Visualization

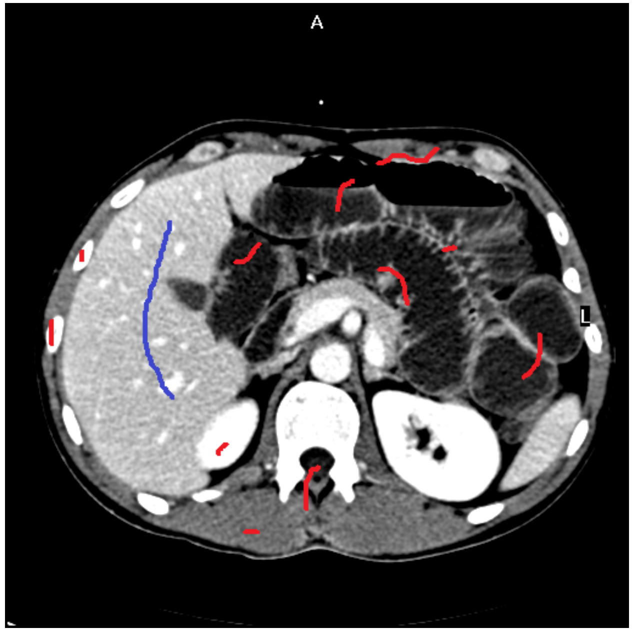
Professor Eric Shaffer

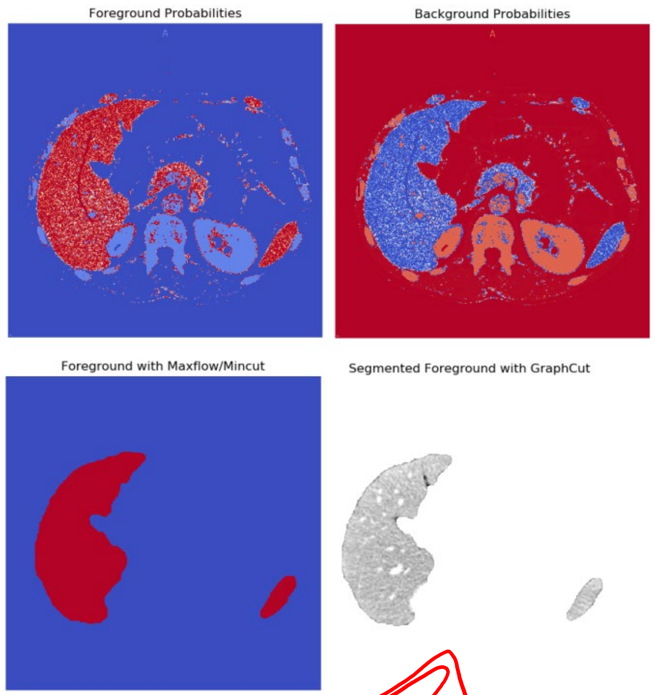
ILLINOIS

# Example: Interactive Graph Cuts

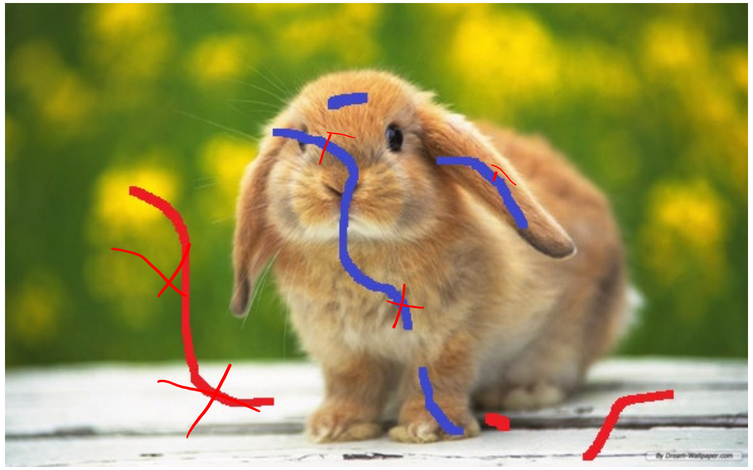
*Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images* Y. Y. Boykov and M. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images, *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*,

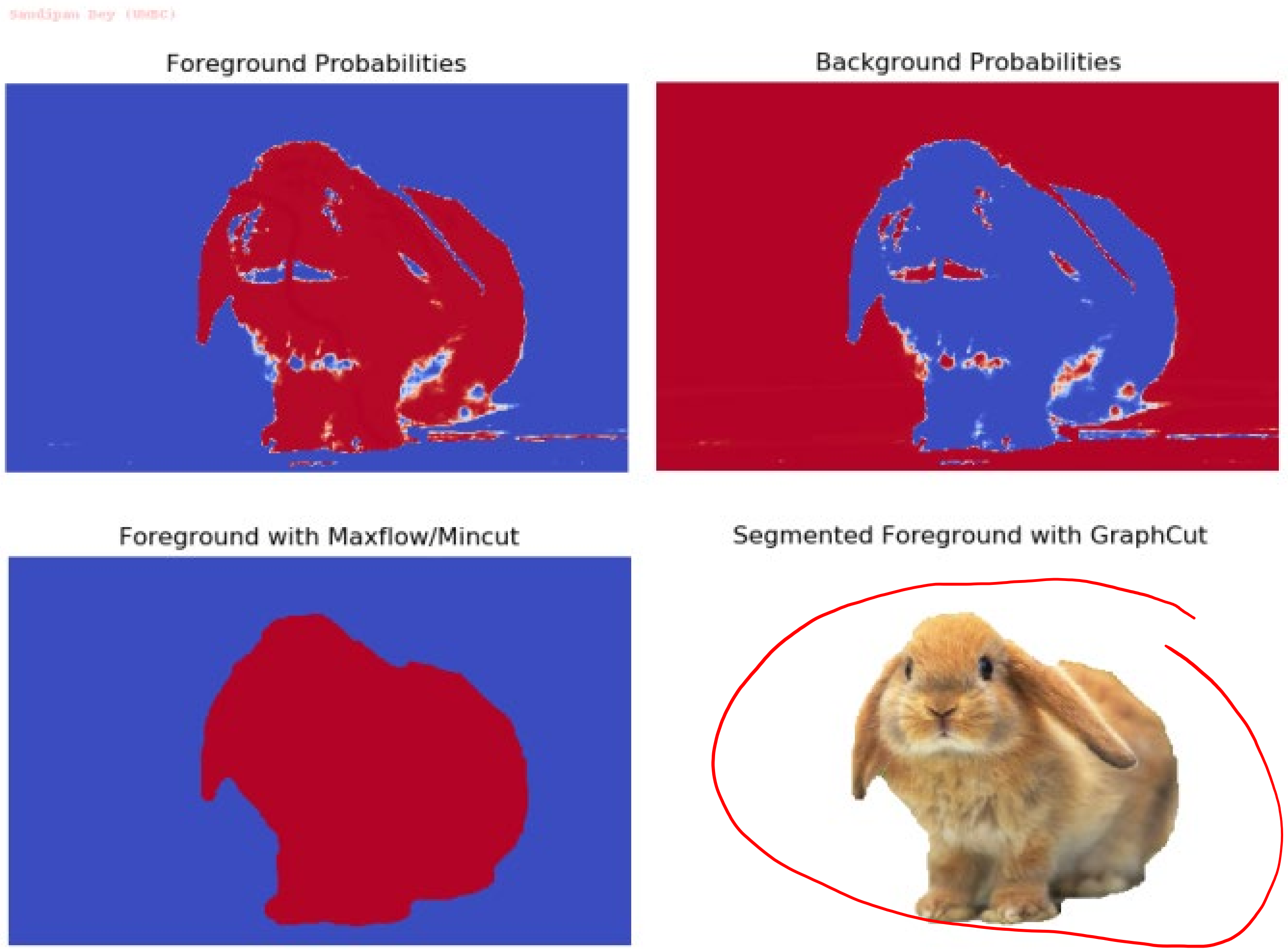





Foreground Probabilities
Background Probabilities
Foreground with Maxflow/Mincut
Segmented Foreground with GraphCut

# ...another example





Sandipan Dey (UMBC)

Foreground Probabilities

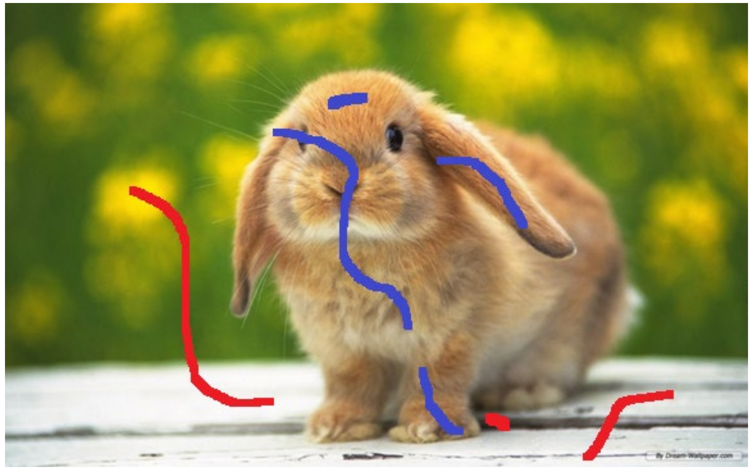
Background Probabilities

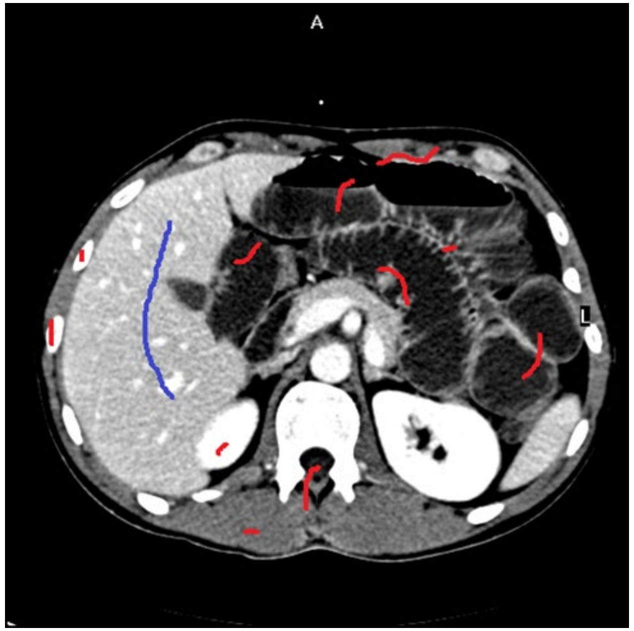
Foreground with Maxflow/Mincut

Segmented Foreground with GraphCut

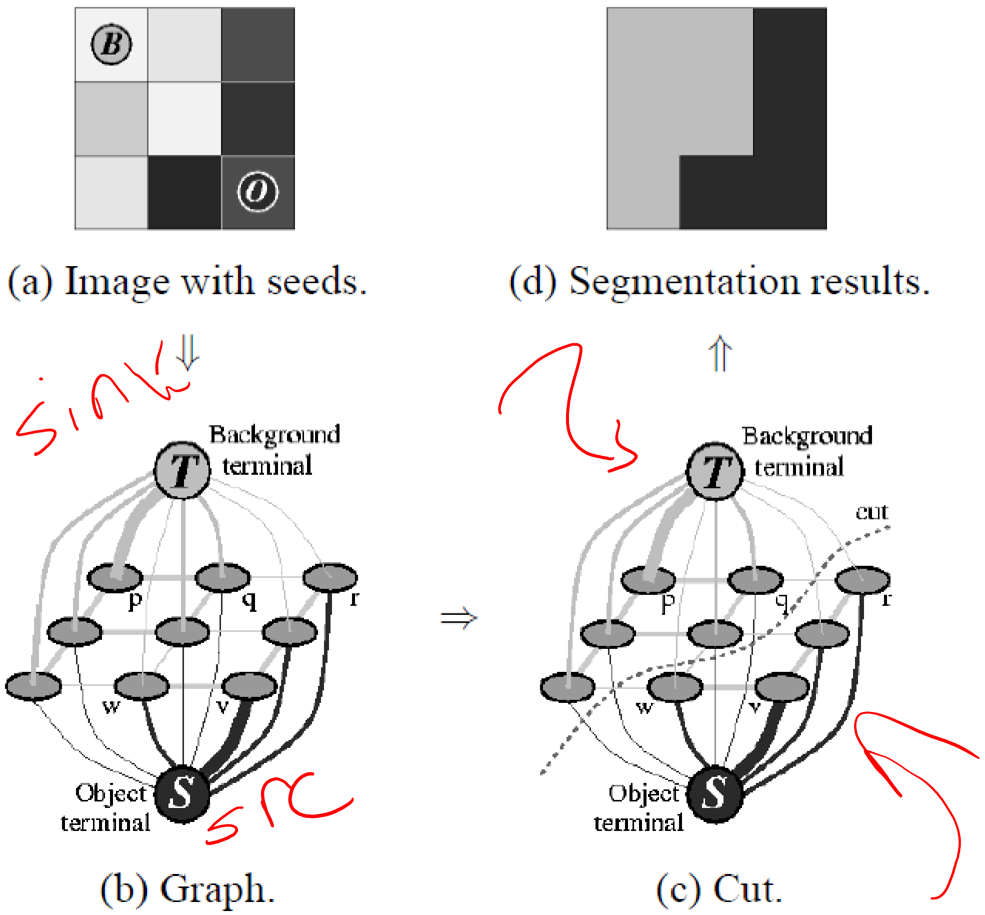
# User Input

user scribbles provide indication of foreground versus background





# Graph Construction



(a) Image with seeds.

(d) Segmentation results.

(b) Graph.

(c) Cut.

Add source s vertex and connect to all pixels

Add sink t vertex connected to all pixels

…doing this in order to use a max flow algorithm

# Source and Sink Edge Weights

Connect source to all pixels

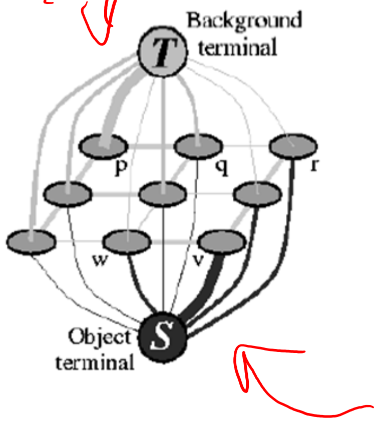
Foreground scribble pixels to foreground(src) vertex = infinity (max float)

Foreground scribble pixels to background(sink) vertex = 0

Connect the sink to all pixels

Background scribble pixels to background vertex(sink) = infinity (max float)

Background scribble pixels to foreground vertex(src) = 0



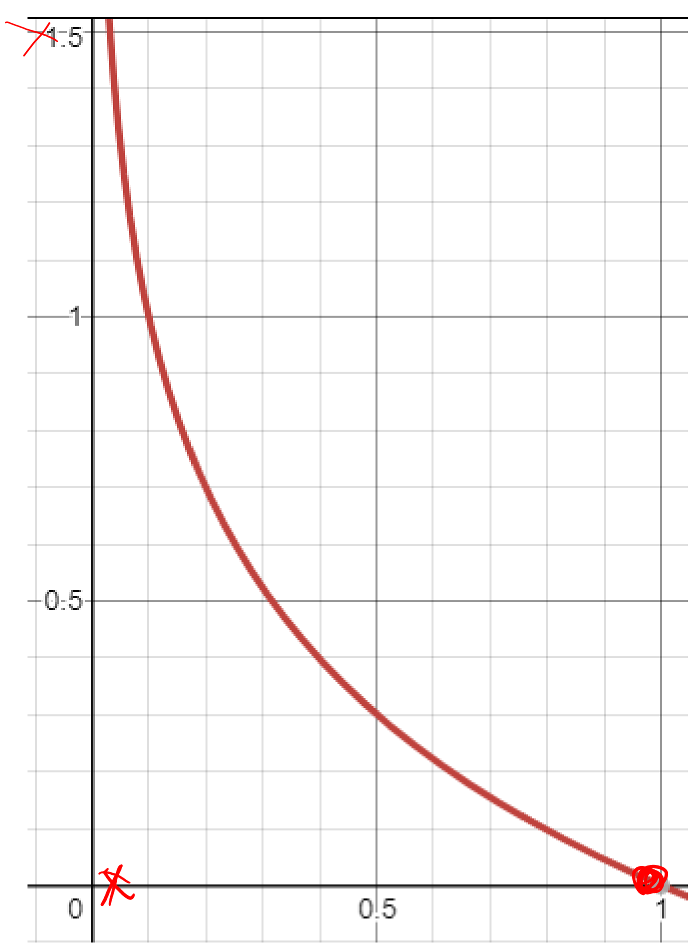
# Edge Weights

Source and Sink to Non-Scribble Vertices $p$

[0, 1]

Weight to foreground $\underline{affinity_{foreground}}(p) = \underline{-\lambda \log P_B(p)}$

Weight to background $affinity_{background}(p) = \underline{-\lambda \log P_F(p)}$



$-\log x$

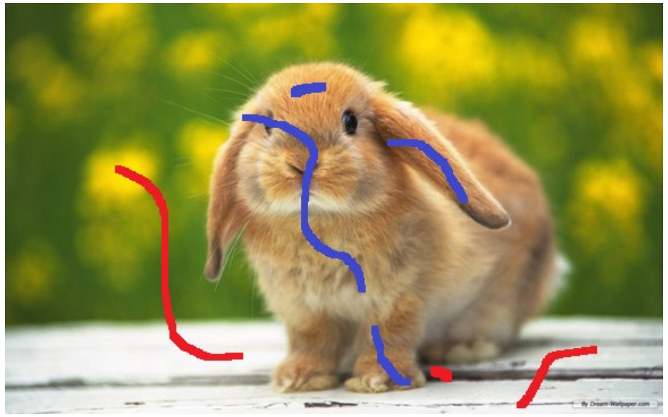
# Foreground and Background Probability Distribution

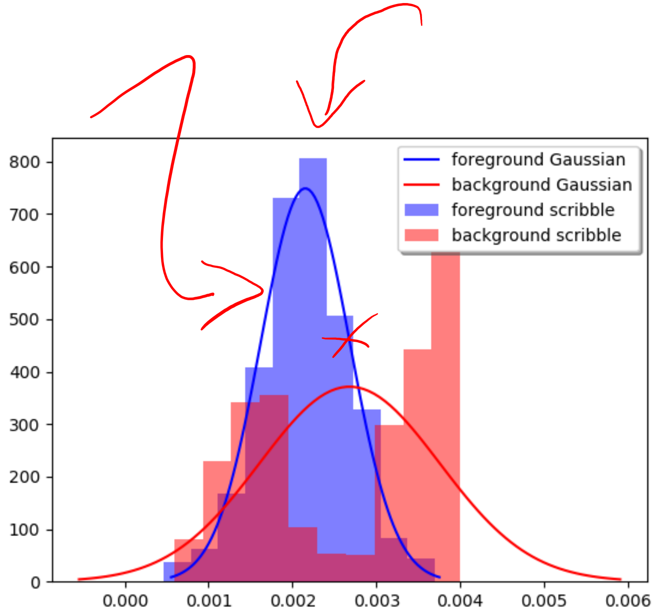
Can use intensity of foreground pixels and compute histogram

Fit a Gaussian to that histogram

Use that distribution to compute $P_F(p) = P_F(I_p)$

$I_p$ is intensity of pixel p



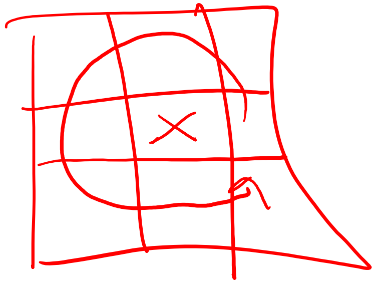


# Edge Weights

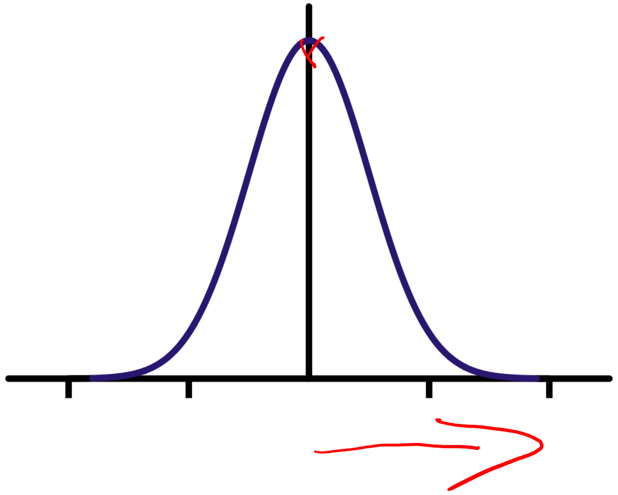
Connect neighboring image vertices with edges (4 or 8 connected)

Edge weights:

$$\boxed{affinity(p_i, p_j) = e^{-\frac{\|f(p_i)-f(p_j)\|^2}{2\sigma^2}}}$$



$f(p_i)$ is intensity in original paper



Same intensity → $affinity(p_i, p_j) = 1$

Different intensity → $affinity(p_i, p_j) \approx 0$

# Generalizes to nD Volumes

- Can be applied to video data (each frame is a 2D slice)
- Or  a volume (e.g. MRI)
  - Uses a 26-connected neighborhood
- In general, fastest max-flow requires O(VE) time
  - In practice….more complicated
  - Algorithms with a higher time-bound can perform better for image data
- Boykov max flow algorithm allows fast updating
  - Add scribbles and adjust existing flow quickly
- Algorithm will generally segment 1920×1080 image in a few seconds