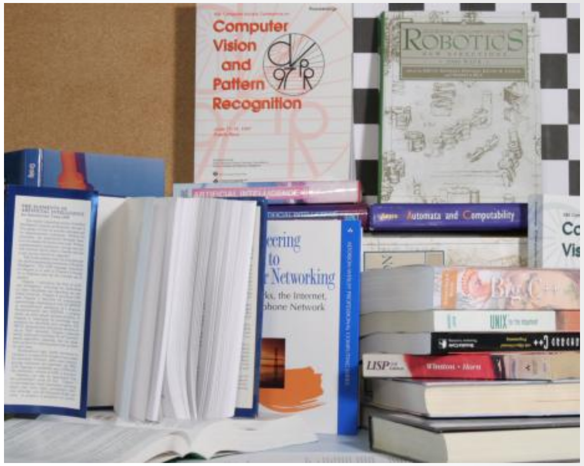
Conditional Random Fields for Dense Stereo Matching

Christopher Yeh
Los Alamitos High

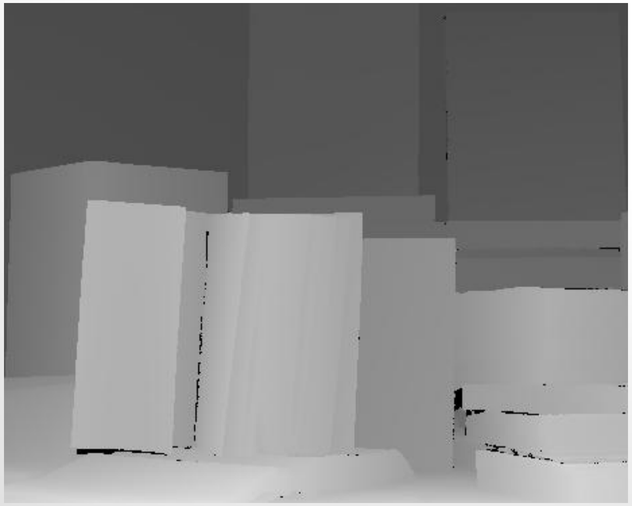
Dr. Alex Ihler (mentor)
UC Irvine

Stereo Correspondence Problem

Left Image



Right Image

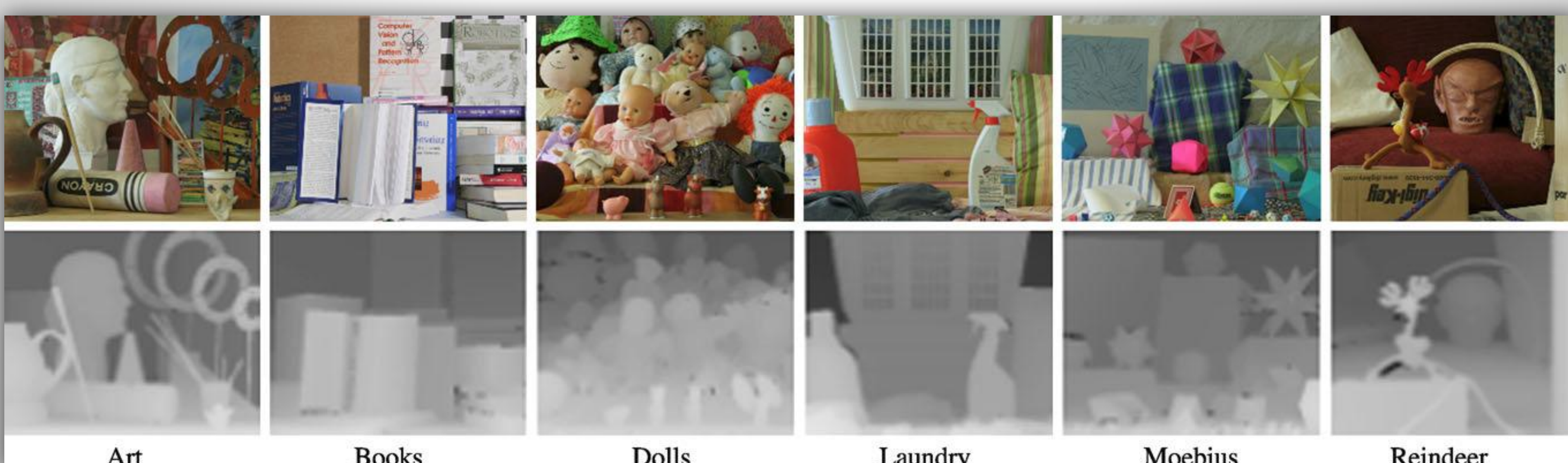


Depth Map

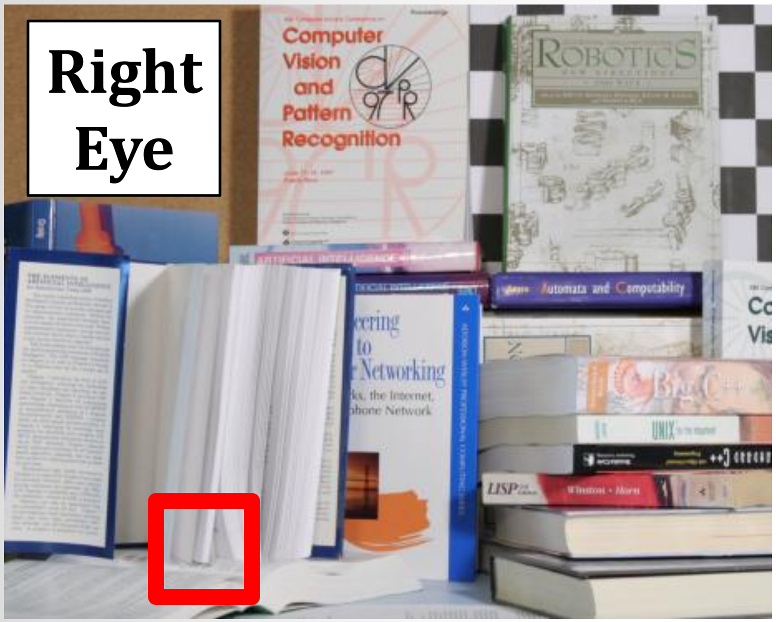
Darker colors indicate greater distance from the camera to the object.

Stereo Image Datasets

- Middlebury 2005 datasets
  - Accurate ground truth depth maps
  - Widely-used → easy to make comparisons



Local Matching



i

i

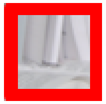
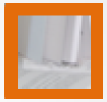
$i + x_i$

$i + x_i$

$i + x_i$

Find matching **disparity (x_i)** between individual pixels or patches of pixels in left and right images.

Local Matching Cost

Data Term: $U(x_i) = \Big|$  $-$  $\Big|$

- $U(x_i)$ measures the similarity of two patches between the left and right images.
  - Patches are similar → small value for $U(x_i)$
  - Patches are different → large value for $U(x_i)$
- Similarity measured in terms of absolute intensity difference

Smoothness

- Majority of an image is “smooth”
- Disparities for neighboring pixels should be similar

Let pixel ***i*** and pixel ***j*** be neighboring pixels. In most cases, they will have the same disparity.

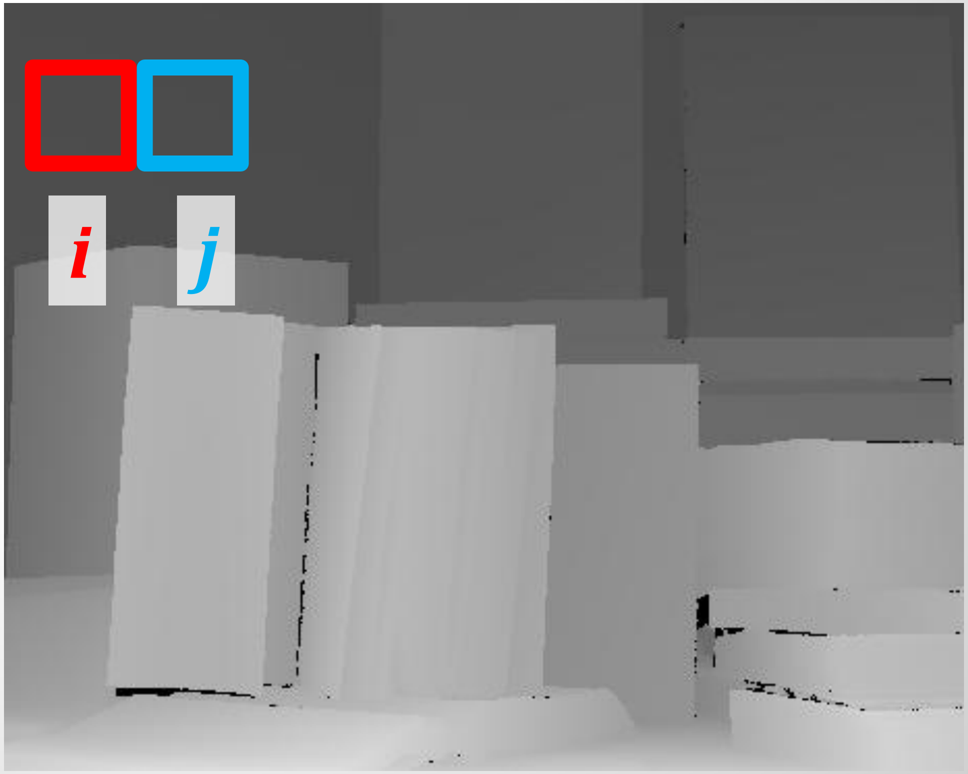
$$x_i \approx x_j$$

Smoothness

- Majority of an image is “smooth”
- Disparities for neighboring pixels should be similar

Let pixel ***i*** and pixel ***j*** be neighboring pixels. In most cases, they will have the same disparity.

$$\boldsymbol{x_i \approx x_j}$$

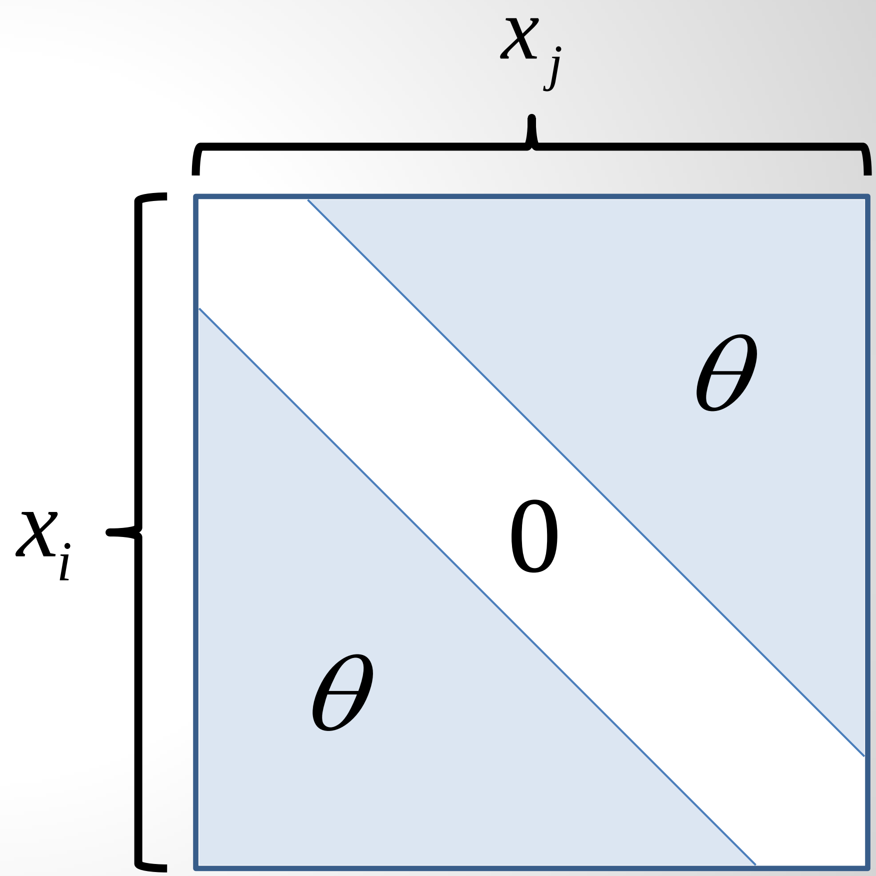


Smoothness Cost

Basic Smoothness Term

$$V(x_i, x_j) = \begin{cases} 0, & \text{if } x_i = x_j \\ \theta, & \text{if } x_i \neq x_j \end{cases}$$

- Penalize when the disparities between neighboring pixels *i* and *j* are different
- Known as <u>Potts Model</u>



Using CRFs for Stereo

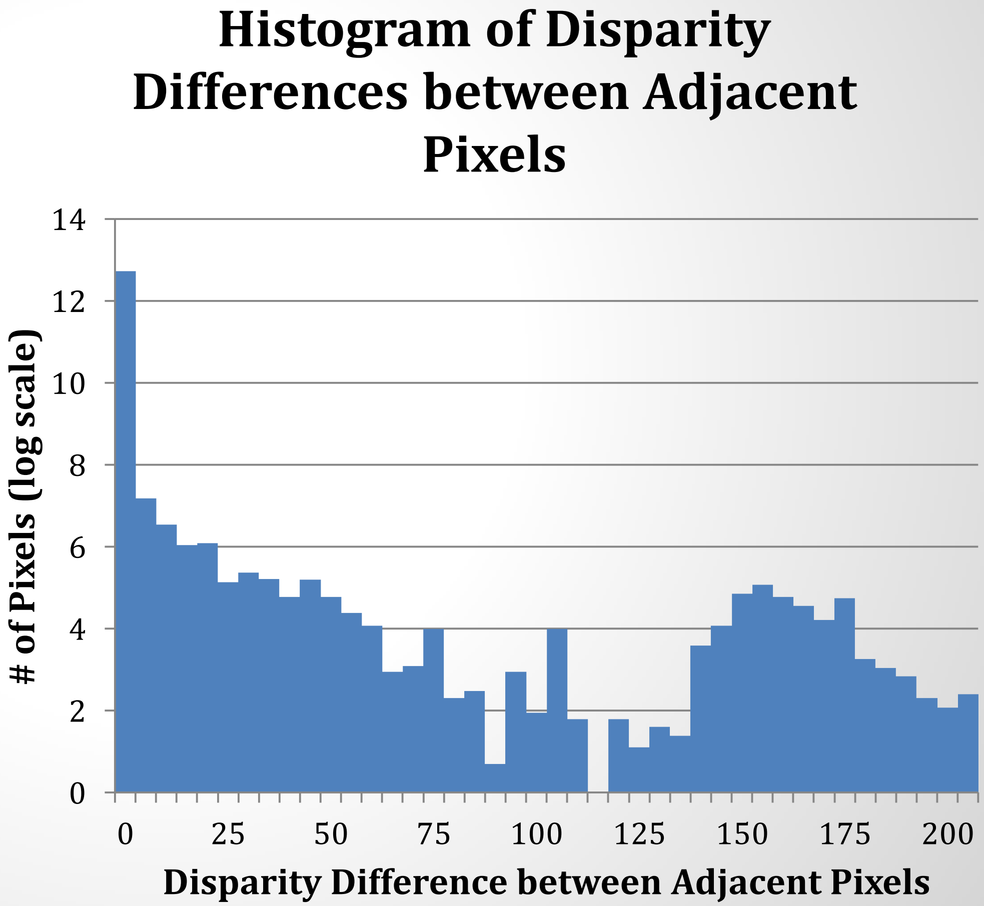
Basic Cost (Energy) Function

$$F(X) = \sum_{i} U(x_i) + \sum_{i\sim j} V(x_i, x_j)$$

- Combine local and smoothness costs into one function
- $F(X)$ = total cost for a certain configuration of disparities X
  - Goal = minimize $F(X)$
- Known as a Conditional Random Field (CRF)

Purpose of Smoothness Term

- Majority of an image is “smooth”
  - in most cases, penalize for disparity jumps
- Allow for disparity jumps at edges
  - Requires accurate edge detection



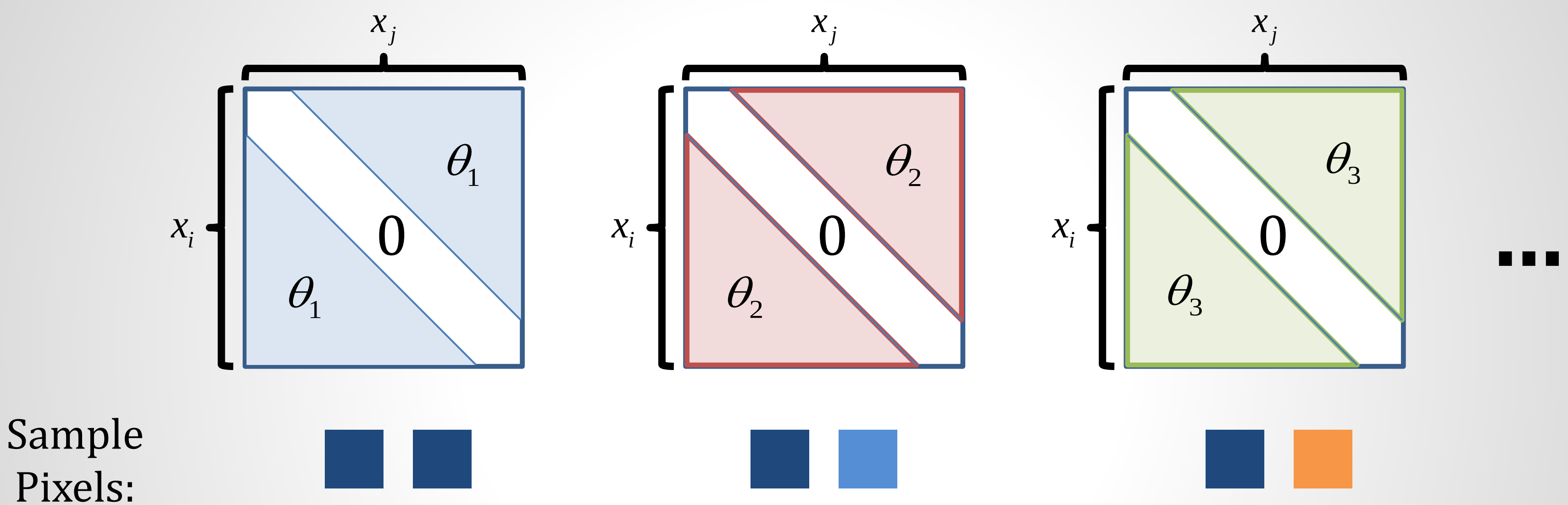
Smoothness Cost: Gradients

Gradient-Modulated Potts Model Smoothness Term

$$V(x_i, x_j) = \begin{cases} 0, & \text{if } x_i = x_j \\ \theta_z, & \text{if } x_i \neq x_j \text{ and } g_{ij} \in B_z \end{cases}$$

- Vary the cost based on color gradient g_{ij} between neighboring pixels using $z = 6$ discrete bins B_z
  - Color gradient (g_{ij}) = root mean square color difference between adjacent pixels
  - B_z = intervals on [0, 2, 4, 8, 12, 16, ∞]

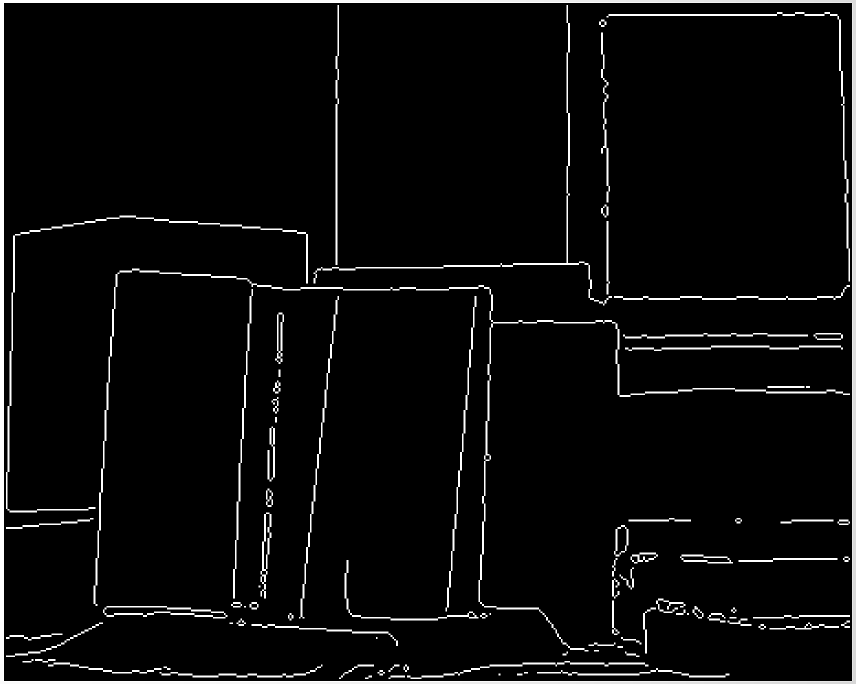
Gradient Bins



Bin (B_z)	1	2	3	...
Color Gradient (g_{ij})	[0, 2)	[2, 4)	[4, 8)	...
θ	$\theta_1 = 3.23$	$\theta_2 = 2.92$	$\theta_3 = 2.87$	...

Edge Detection

- Color gradient is not a true edge detector
- Separate parameters for non-edge vs. edge pixels
- Combine color gradient feature with **Canny edge detection** feature → produces more accurate results



Problems with the Potts Model

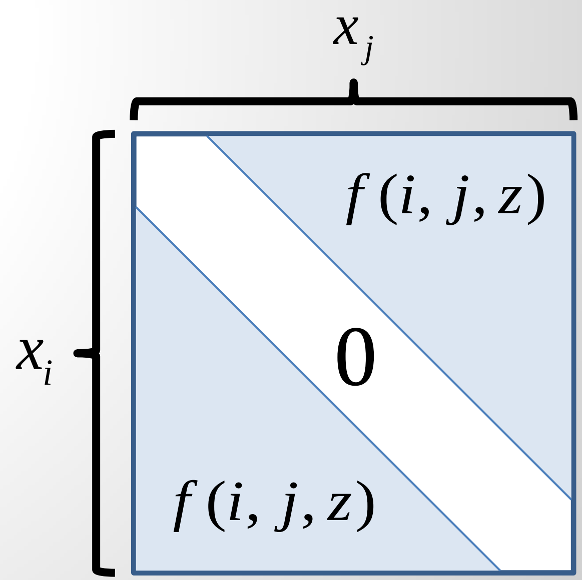
- Uses discrete costs
- Uses simple binary approach to account for differences in disparity between neighboring pixels
  - Only considers $x_i = x_j$ or $x_i \neq x_j$
  - Fails to factor in $x_i - x_j$

Smoothness Cost: Log Model

Disparity Difference and Gradient-Modulated Log Model

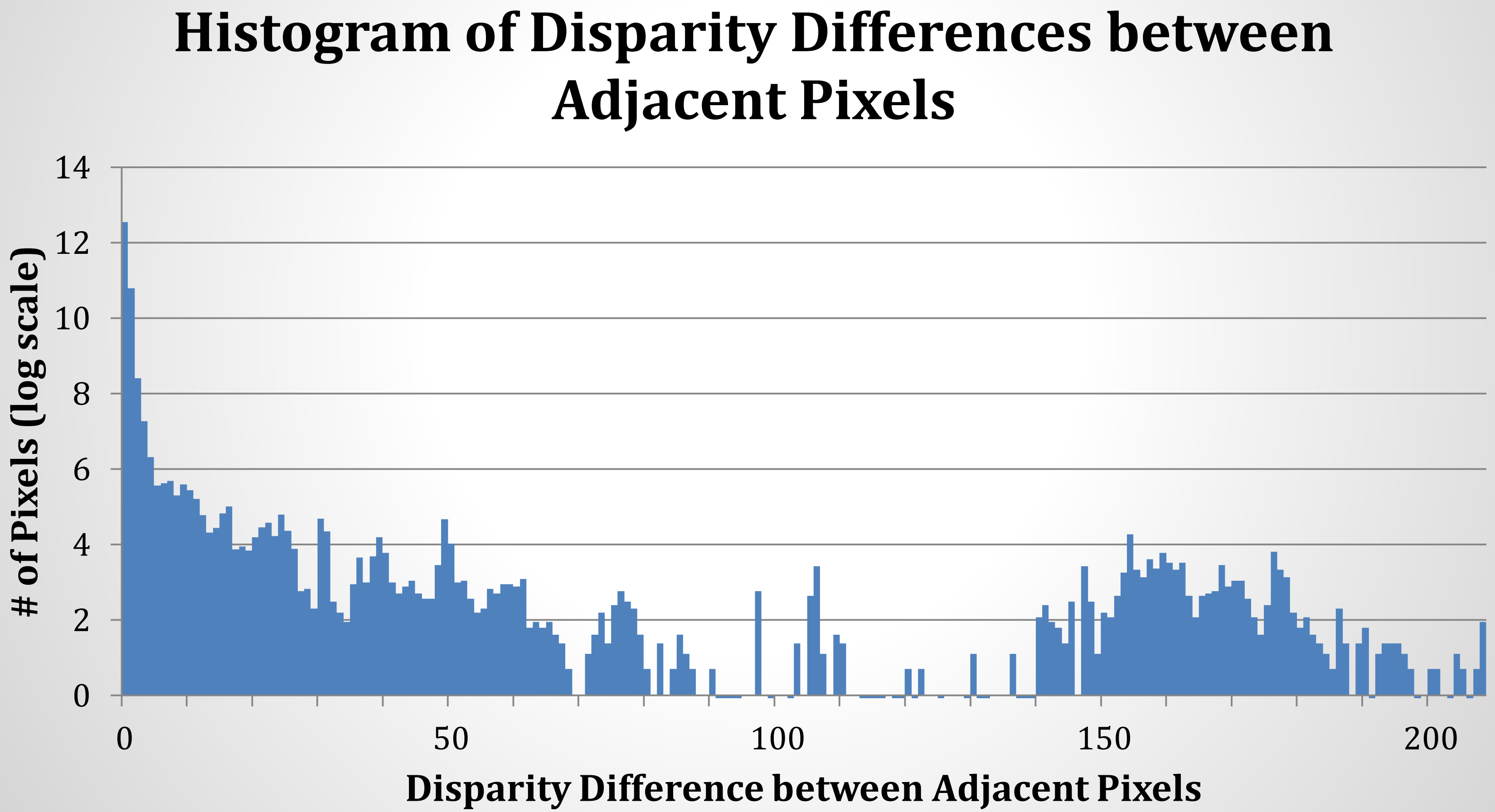
$$V(x_i, x_j) = \begin{cases} 0, & \text{if } x_i = x_j \\ f(i, j, z), & \text{if } x_i \neq x_j \text{ and } g_{ij} \in B_z \end{cases}$$

$$f(i, j, z) = \theta_{a,z} \ln\left(1 + e^{\theta_{b,z}}(x_i - x_j)\right)$$

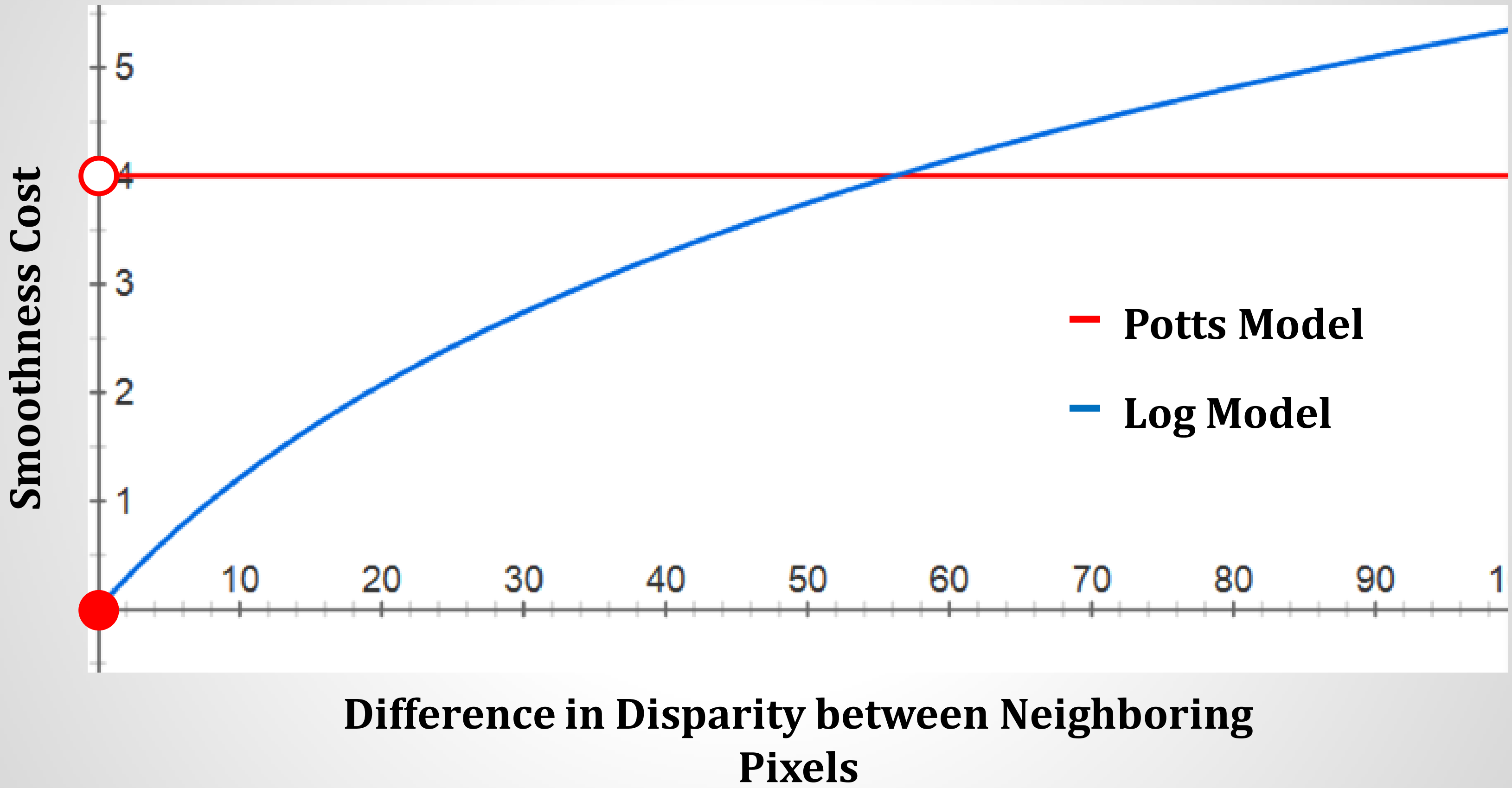


- Cost for difference in disparity modeled logarithmically
- Takes 2 parameters (θ_a and θ_b) per gradient bin
  - Twice as many as Potts Model

Why Log Model



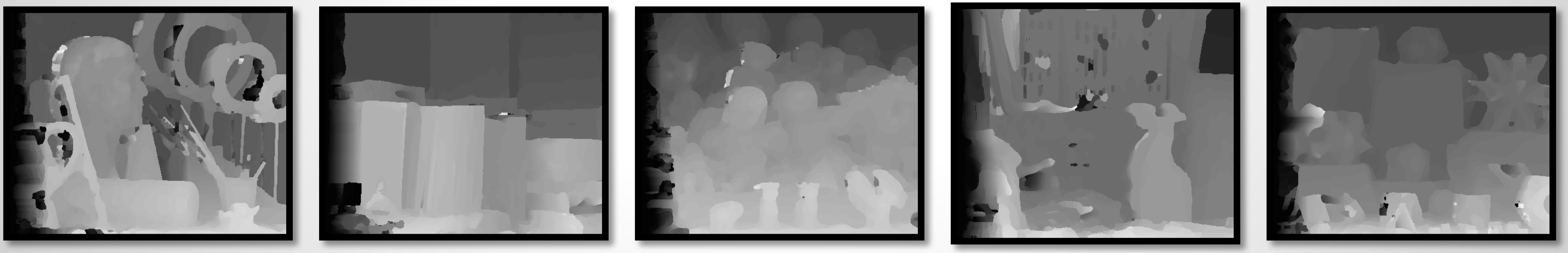
Why Log Model



Disclaimer: The equations used in graph above are $y=3*\ln(1+e^{-3}x)$ and $y=4$. These values are for demonstration purposes only. They were not actually used in the research.

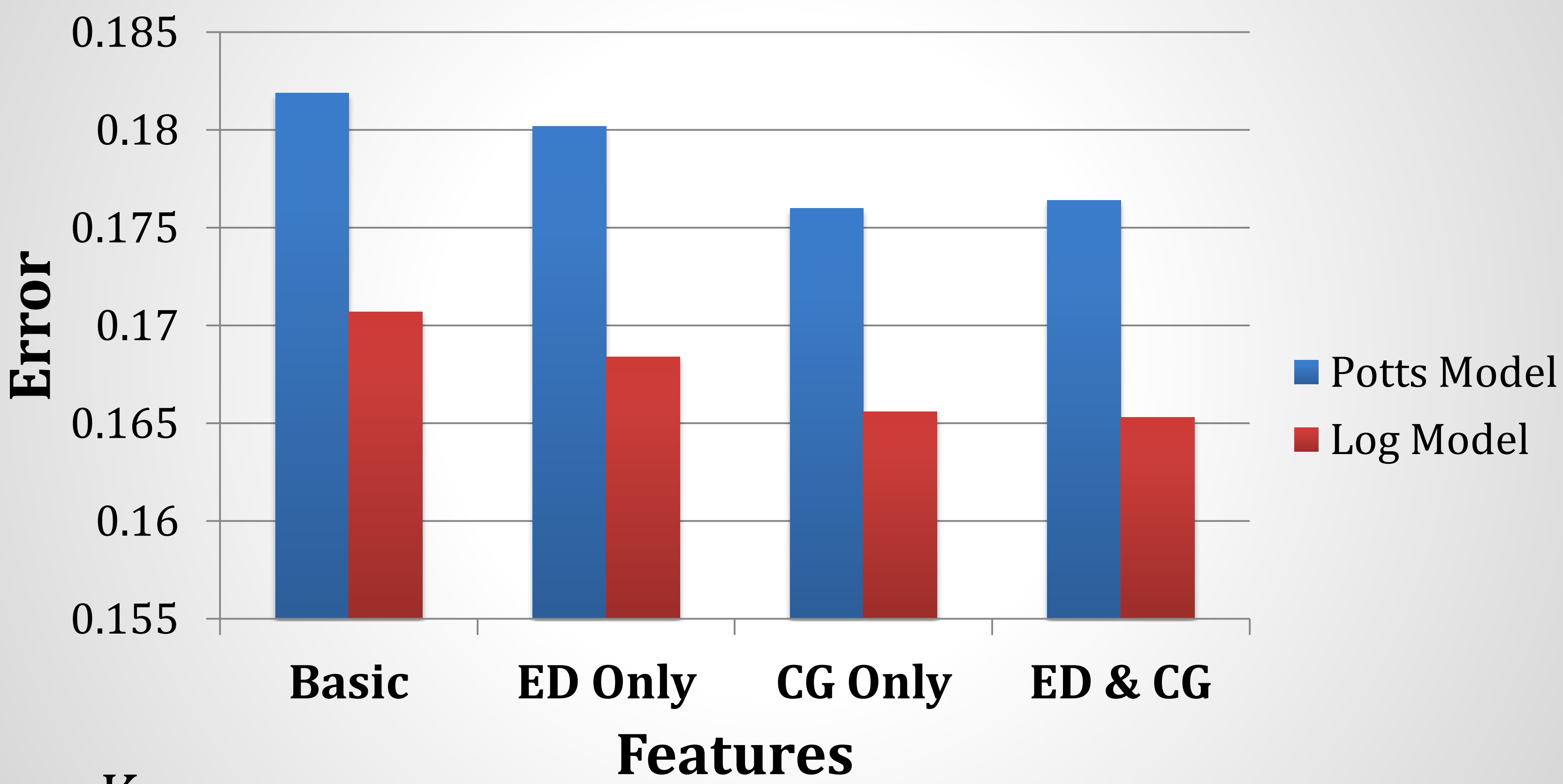
Solving Parameters: Machine Learning

- What is Machine Learning?
- 5 “Training” Datasets from Middlebury
- JGMT Toolbox for MATALB



<u>Note:</u> 1. Images above are from the log model with edge detection but no gradient bins. 2. Training was done using Justin's Graphical Models / Conditional Random Field Toolbox (JGMT), which can be found online here: http://users.cecs.anu.edu.au/~jdomke/JGMT/

Results Summary



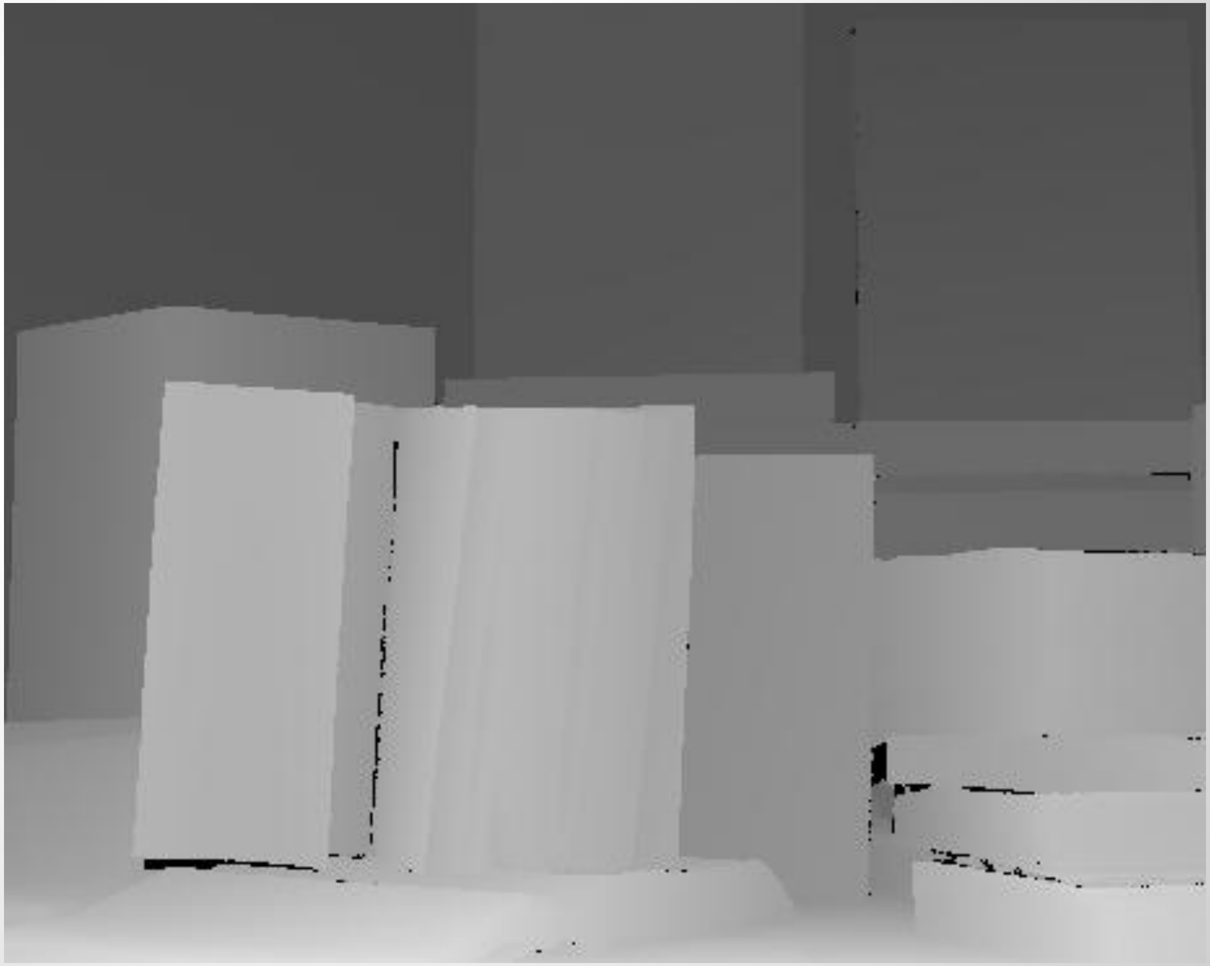
Key:

- ED = Canny **E**dge **D**etection
- EG = **C**olor **G**radients

Potts vs. Log: An Example

Ground Truth

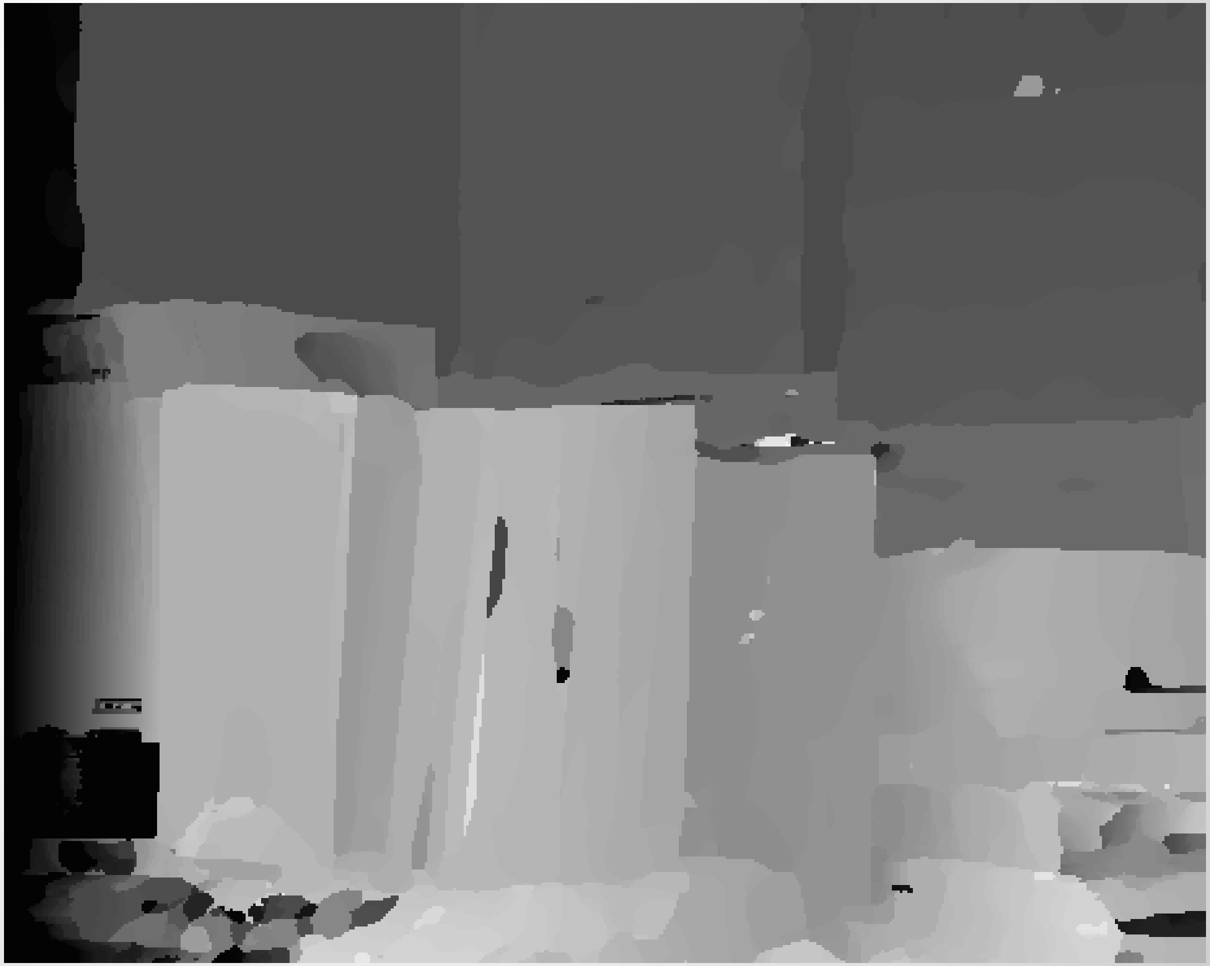
What the depth map should look like.



Potts vs. Log: An Example

<u>Potts Model</u>
w/ ED + CG

Error:
0.18425



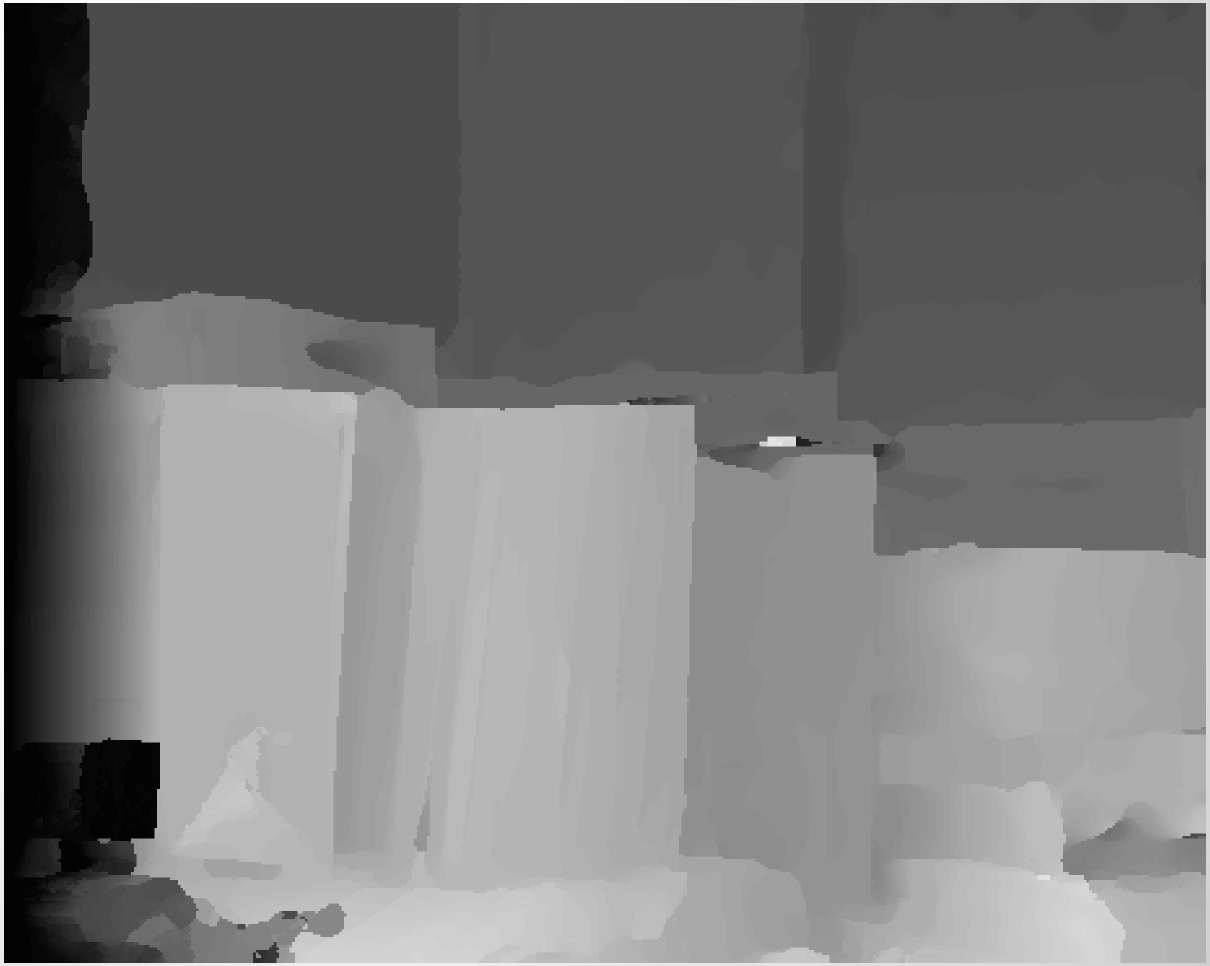
Potts vs. Log: An Example

Log. Model

w/ ED + CG

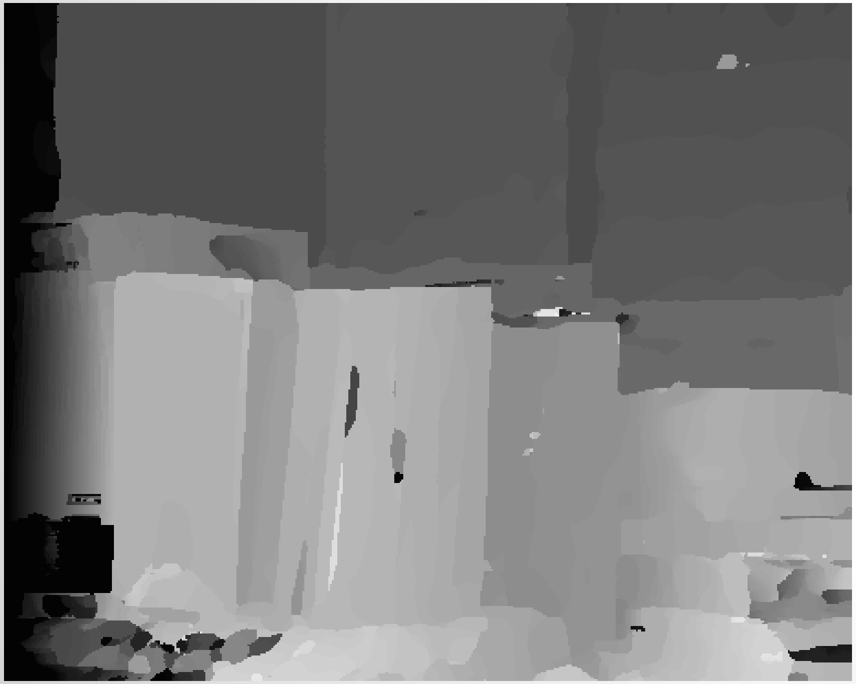
Error:

0.17602

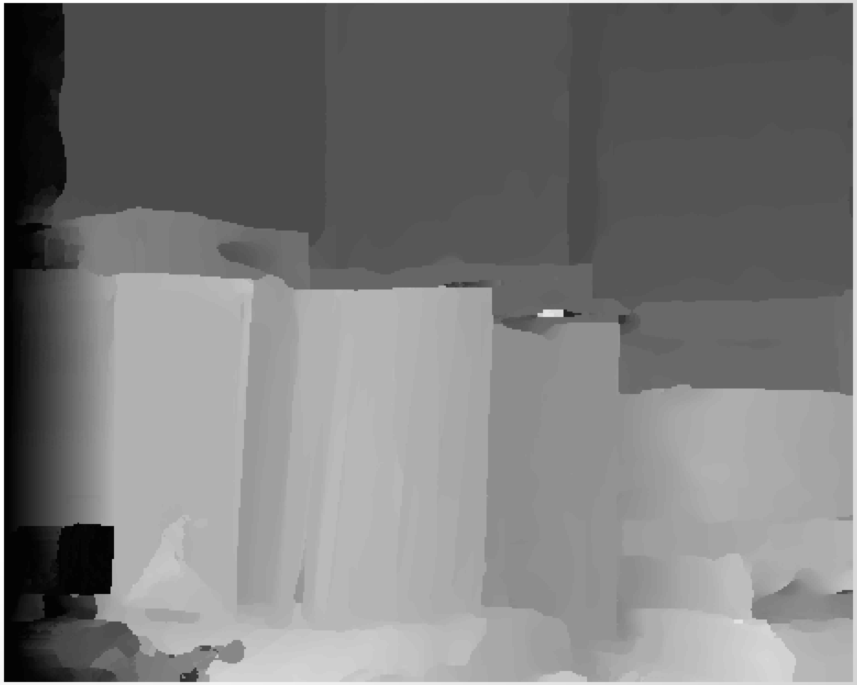


Potts vs. Log: An Example

Potts Model



Log Model



Number of Parameters

	# of Local Matching Parameters	# of Smoothness Parameters
Potts Model	1	1
Potts Model + ED	1	2
Potts Model + CG	1	6
Potts Model + ED + CG	1	12
Log Model	1	2
Log Model + ED	1	4
Log Model + CG	1	12
Log Model + ED + CG	1	24

Key:

- ED = Canny **E**dge **D**etection
- EG = **C**olor **G**radients

Normalized Results

2 Parameters

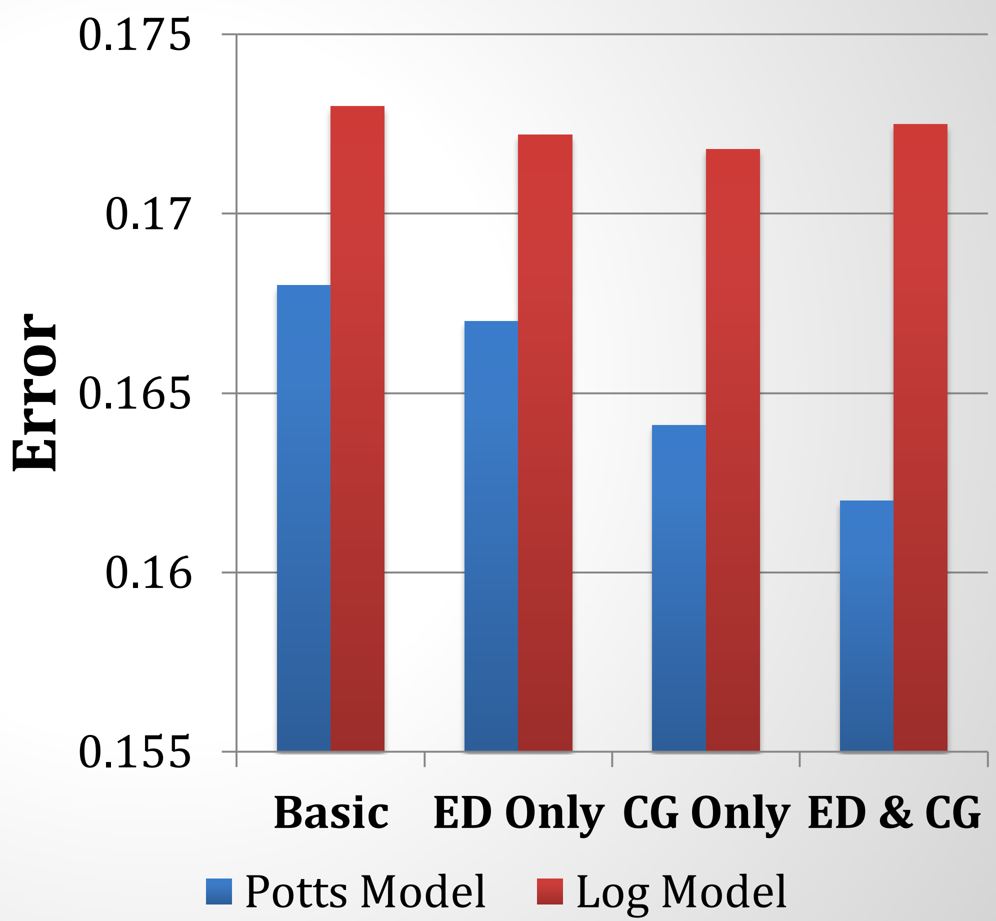
- Potts Model w/ ED
  - Error: 0.1802
- Basic Log Model
  - Error: **0.1707**
- Improvement: 5.3%

12 Parameters

- Potts Model w/ ED and CG
  - Error: 0.1764
- Log Model w/ CG
  - Error: **0.1656**
- Improvement: 6.1%

Testing on a Different Set

- Potts Model actually does better here
- Suggests that log model may be over-fitting the data
  - Solution: train on more datasets



Conclusions + Future Work

- Logarithmic Model is able to assign more accurate smoothness costs than a Potts Model
- Future work
  - Train model on more data to avoid over-fitting
  - Try different models

THANK YOU!