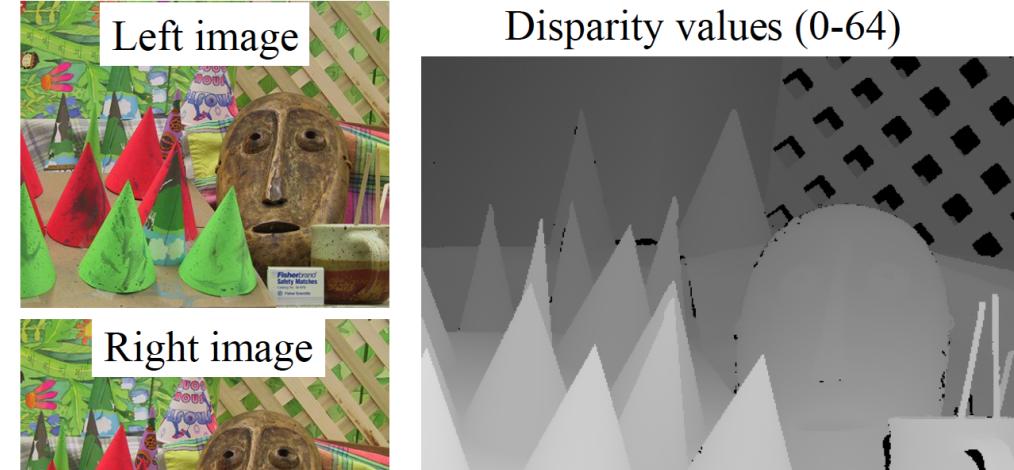


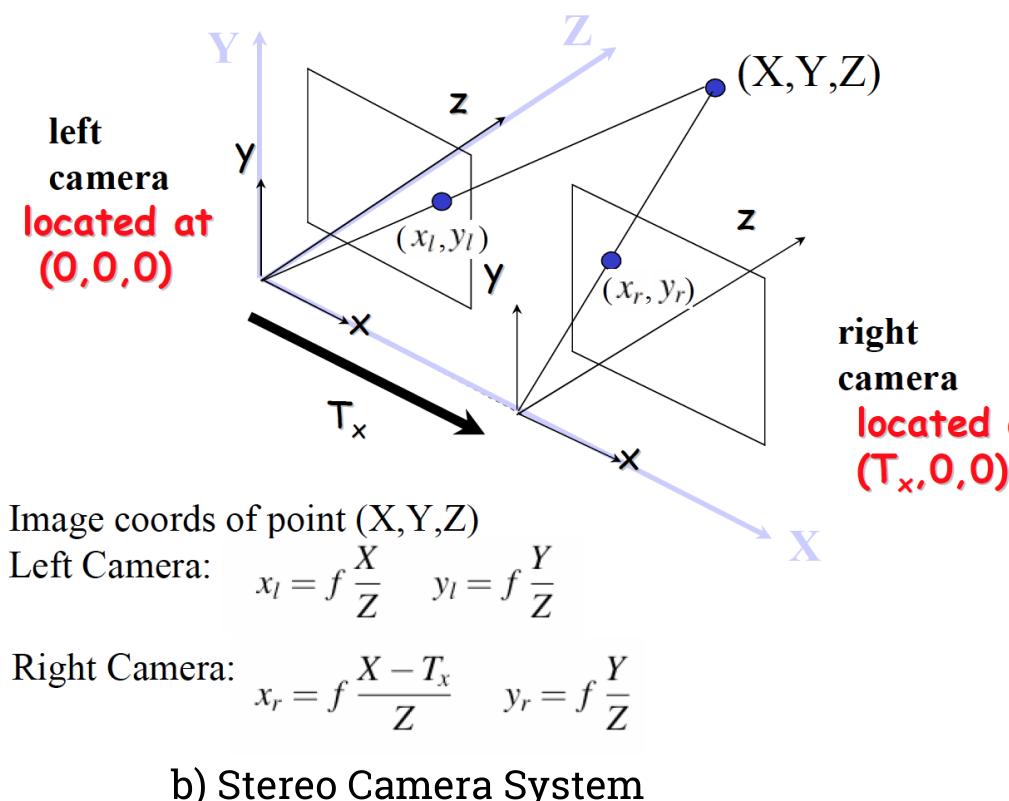
Self-Driving Cars to See Through Fog with GPU Imaging and Deep Learning

Depth from Stereo Disparity

- We approach the dehazing problem by extracting a depth map, which is used to relate the hazy scene to the clear scene.
- Stereo vision:** We compute depth using two cameras and observing the "disparity" between corresponding pixels.



a) Disparity from Structured Light



Left camera

$$x_l = f \frac{X}{Z} \quad y_l = f \frac{Y}{Z}$$

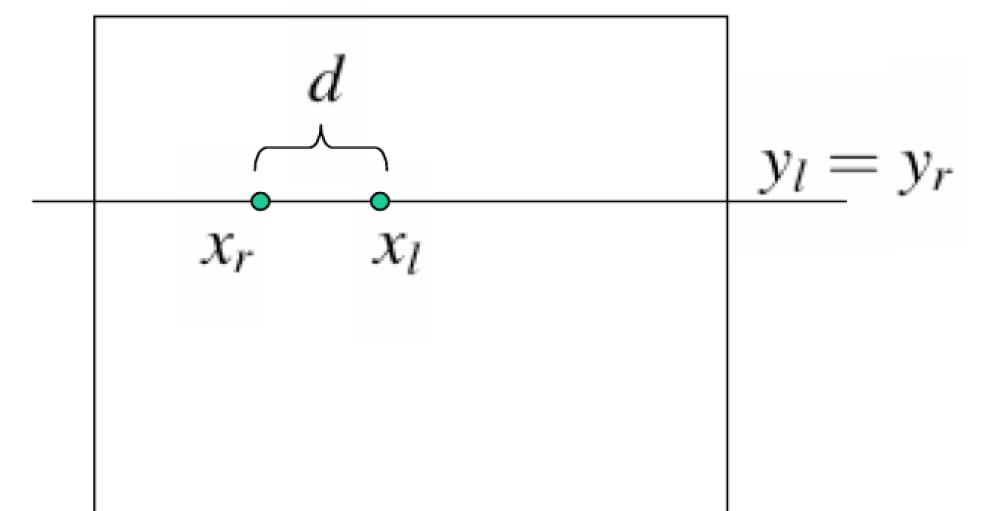
Right camera

$$x_r = f \frac{X - T_x}{Z} \quad y_r = f \frac{Y}{Z}$$

Stereo Disparity

$$d = x_l - x_r = f \frac{X}{Z} - (f \frac{X}{Z} - f \frac{T_x}{Z})$$

$$d = \frac{f T_x}{Z}$$



Important equation!

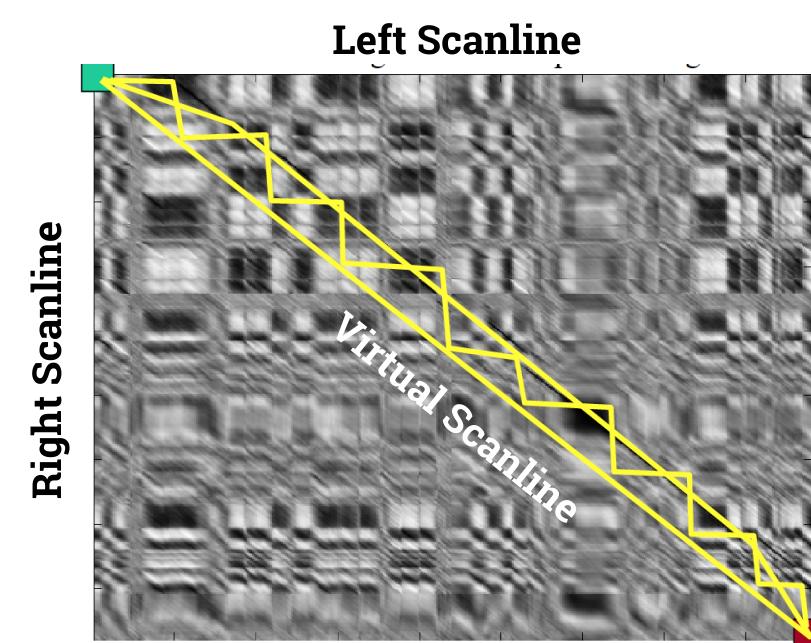
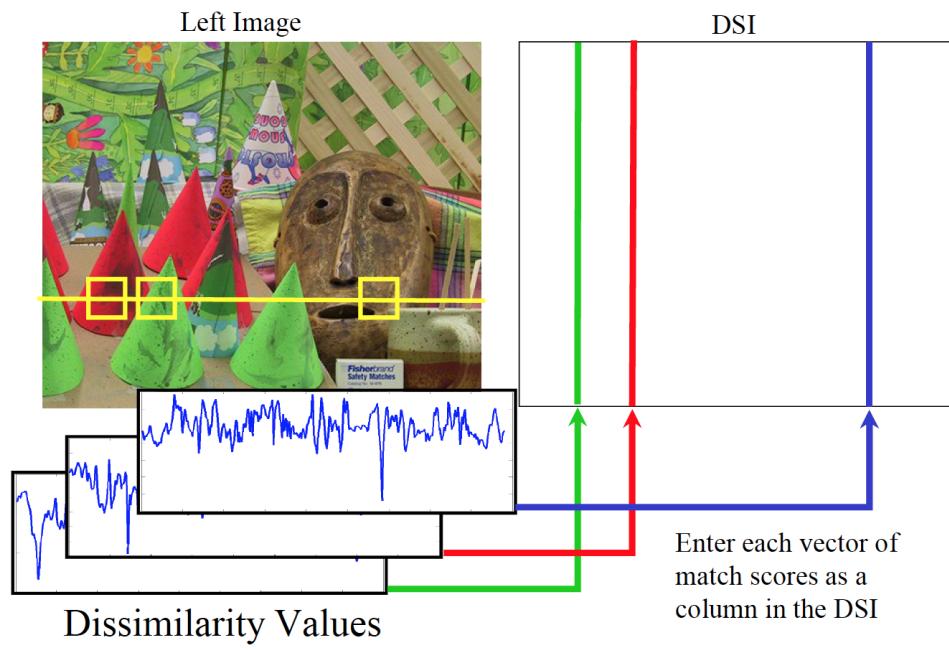
Efficient Disparity Map Extraction

- Problem:** Given two stereo images, compute the disparity map.
- Define a naive match score (cost) of two pixels using the Normalized Cross-Correlation (NCC). We define a minimum threshold for this match score, α .

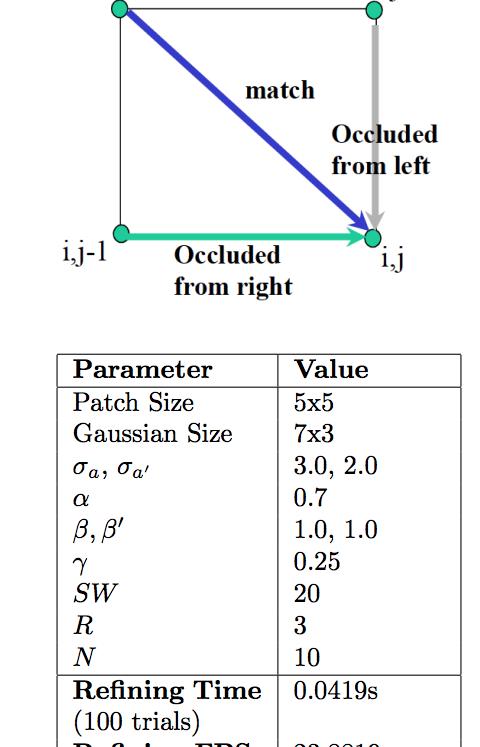
$$M(l, r) = 0.5(1 - NCC)$$

$$M(l, r) = 0.5\left(1 - \frac{\sum_{\sigma \in \Omega}(I_{pl+\sigma}^l - \bar{I}_{pl}^l)\sum_{\sigma \in \Omega}(I_{pr+\sigma}^r - \bar{I}_{pr}^r)}{\sqrt{\sum_{\sigma \in \Omega}(I_{pl+\sigma}^l - \bar{I}_{pl}^l)^2 \sum_{\sigma \in \Omega}(I_{pr+\sigma}^r - \bar{I}_{pr}^r)^2}}\right)$$

- We set the two cameras to be level, allowing us to assume an equipolar constraint: each pixel only matches to a pixel on the same row in the corresponding image
- For every row, we can construct a disparity space image (DSI), where every entry $DSI[i, j]$ corresponds to the matching value of $IMG[r, i]$ and $IMG[r, j]$



$$c(l, r) = \min \begin{cases} c(l-1, r) + \beta \\ c(l-1, r-1) + M(l, r) \\ c(l, r-1) + \beta \end{cases}$$



- The disparity values for each row is found by constructing the min cost path through the DSI: backtrack using pointers
- Path cannot double back on itself
- Cox et al. (1996) proposed a dynamic programming solution
- Path is always on the far side of the right scanline as the left image always shifted to the left
- Propagate DP F entries past virtual scanline, where F is the max disparity. Reduces DSI memory to $F \times \text{COLS}$.

```
Algorithm 1 Proposed Refinement Algorithm
1: function REFINING(dismap, img)
2:   edges ← cannyEdgeDetection(img)
3:   edge ← dilate(edge)
4:   labels, components ← connectedComponents(edge)
5:   labels ← labelEdges(labels, edge)
6:   assignPixelsValues(labels, components)
7:   for comp in component do
8:     SW ← 20
9:     histogram ← getHistogram(comp)
10:    ps ← preSum(histogram)
11:    if ps[255] - ps[0] < A then
12:      continue
13:      em ← em
14:      window ← maxSWSum(histogram, ps, SW)
15:      mx ← findMax(window)
16:      lb ← findLeftValley(mx, R, N)
17:      rb ← findRightValley(mx, R, N)
18:      for pixel in comp do
19:        if dismap.pixel.value not in [lb, rb] then
20:          dismap.pixel.value = nearest(lb, rb)
21:        end if
22:      end for
23:    end for
24:  end function
```

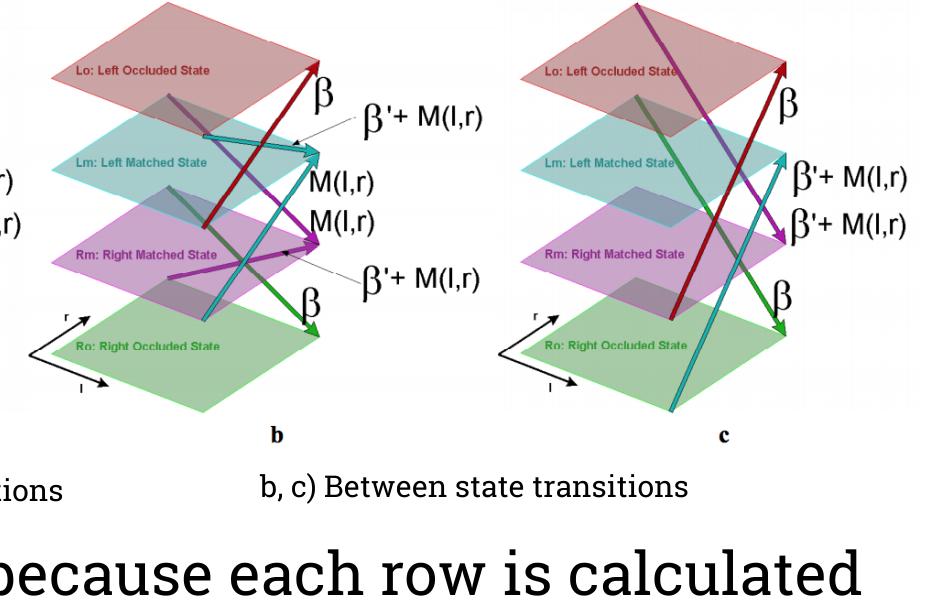
- DP is parallelizable for real-time applications as each row of the disparity map is calculated independently
- Initial three state DP approach did not produce accurate disparity maps. We utilize a four state DP algorithm proposed by researchers at Microsoft that builds to the three state model.
- Distinguishes between a left match and a right match. A true match is a consecutive left and right match. Results in less false occlusions.

$$c_{Lo}[l, r] = \min \begin{cases} c_{Lo}[l, r-1] + \alpha \\ c_{Lm}[l, r-1] + \beta \\ c_{Rm}[l, r-1] + \beta \end{cases}$$

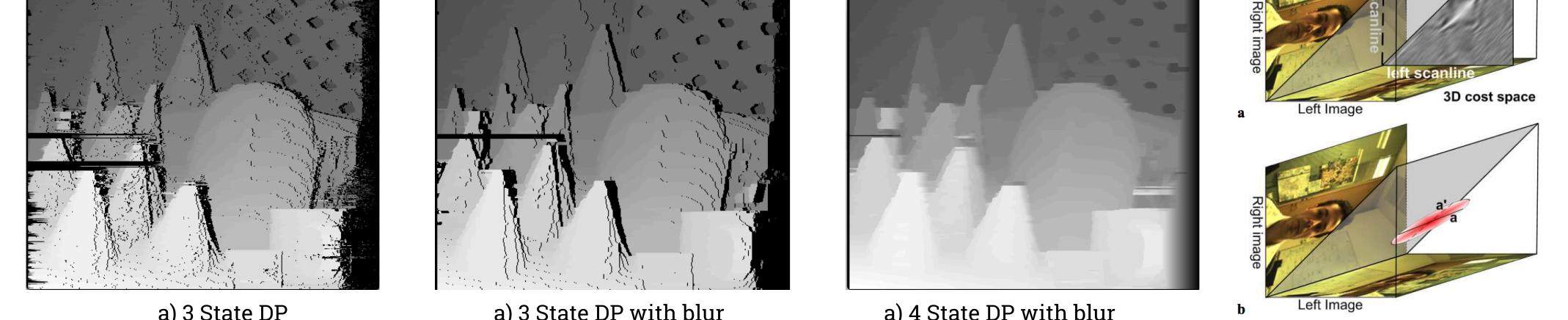
$$c_{Lm}[l, r] = M(l, r) + \min \begin{cases} c_{Lo}[l, r-1] + \beta' \\ c_{Lm}[l, r-1] + \gamma \\ c_{Rm}[l, r-1] + \beta' \end{cases}$$

Transitions for Ro and Rm can be determined through symmetry

a) State preserving transitions

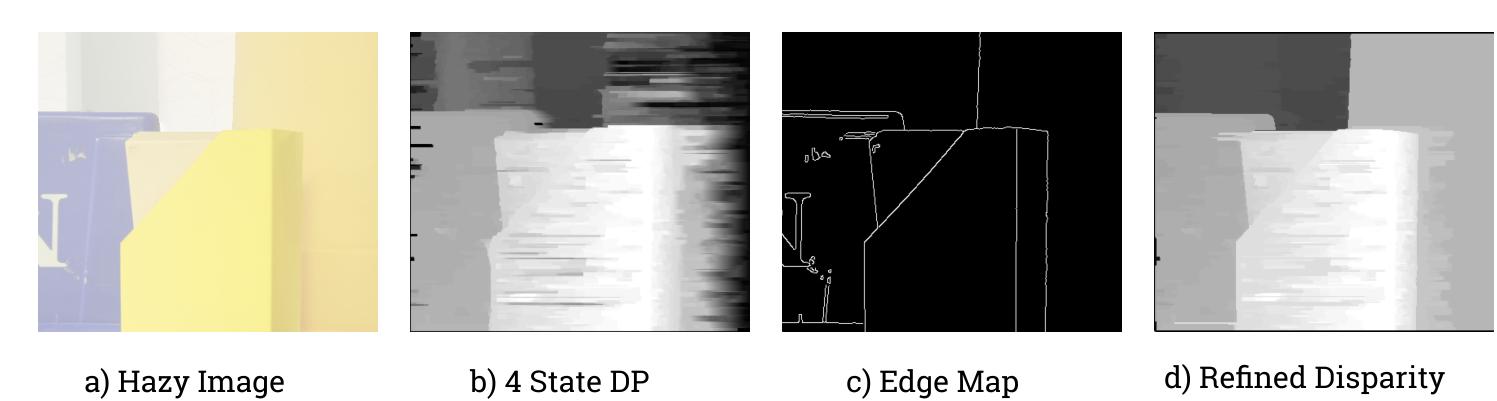


- We have resolved false occlusions, but because each row is calculated independently, any DP algorithm produces inconsistent 'streaks'. To remedy this while preserving parallelism, we propagate information across rows by stacking the DSIs and running a Gaussian Blur across the virtual scanline.



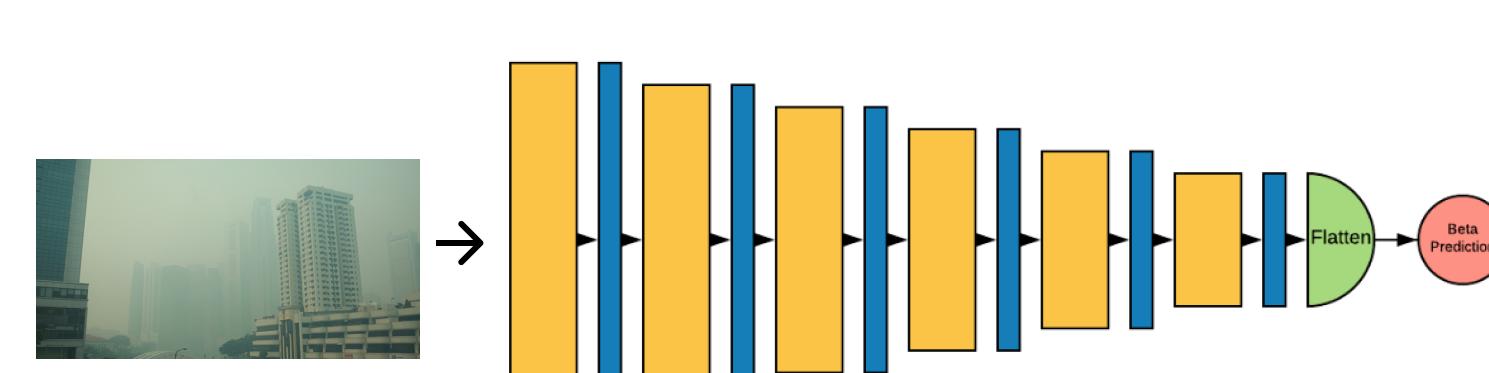
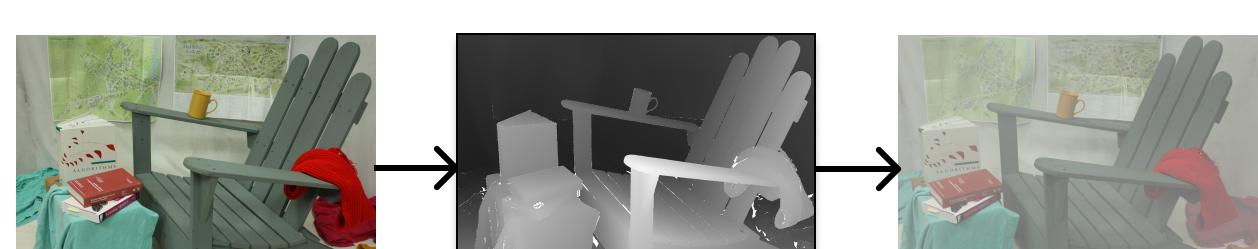
- Four state DP fails when stereo images are hazy. We propose a fast $O(N)$, refinement algorithm that greatly improves disparity map quality.

- Refinement:** We account for 1) discrete histogram peaks (non-max sliding-window is noise) and 2) continuous change in disparity.



Data:

- No natural haze datasets. Use Atmospheric Scattering Model to generate data.
- Middlebury Datasets 2005, 2014: Contains indoor stereo imagery for depth estimation.
- Given depth, we induce haze in the scenes for $\alpha \in (0.75, 1.0)$, $\beta \in (0, 1.5)$, which was recommended by Cai et al. (Dehazenet)
- 21 scenes, 80 hazy images/scene, 10 128x128 crops/image, totaling 16800 crops



- Regression using Convolutional Neural Network
- Model Input: 128x128 RGB crop of hazy scene
- Model Output: Scattering coefficient, β
- The β predictions will be used to quantify how much airlight to subtract from the hazy scene.

Hyperparameters		
activation	ReLU	
optimizer	Adam	
number of kernels	32	
kernel size	3x3	
pooling operator	2x2 max	
epochs	75	
batch size	32	
loss	mean absolute error	

Train	0.067
Validation	0.089
Test	0.081

