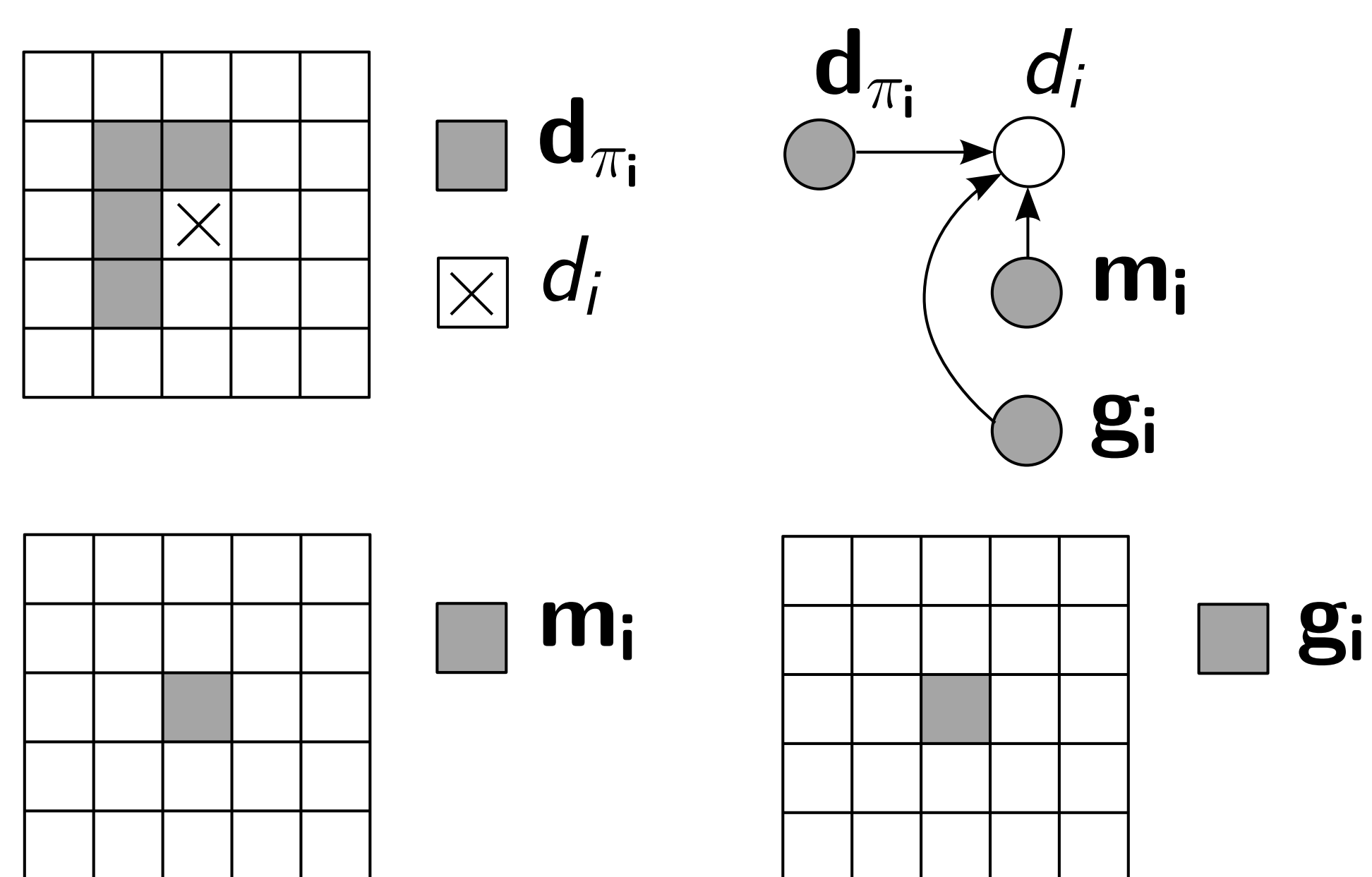


Who Killed the Directed Model (of Stereo)?

Justin Domke, Alap Karapurkar, and Yiannis Aloimonos, University of Maryland

Stereo Representation



$$p(d_i | \mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i) = \frac{\exp(f_d(d_i, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \theta^d) + f_m(d_i, \mathbf{m}_i, \theta^m))}{Z(\theta^d, \theta^m, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i)}$$

$$Z(\theta^d, \theta^m, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i) = \sum_d \exp(f_d(d, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \theta^d) + f_m(d, \mathbf{m}_i, \theta^m))$$

The functions f_d and f_m are lookup tables.

$$f_d(d_i, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \theta^d) = \theta_a^d, \quad a = \text{dcase}(d_i, \mathbf{d}_{\pi_i}, \mathbf{g}_i)$$

$$f_m(d, \mathbf{m}_i, \theta^m) = \theta_b^m, \quad b = \text{mcase}(d, \mathbf{m}_i)$$

dcase

$\text{dcase}(d_i, \mathbf{d}_{\pi_i}, \mathbf{g}_i)$ computes cases based on neighboring disparities and gradients. A naive representation of $(d_i, \mathbf{d}_{\pi_i})$ is impractical. Instead, for each parent d_j , consider 3 cases:

- 1) $|d_i - d_j| = 0$
- 2) $|d_i - d_j| = 1$
- 3) $|d_i - d_j| > 1$

Results in $81 = 4^3$ bins.

Compute intensity derivatives in each direction with 3 bins.

Total of $729 = 81 \cdot 3 \cdot 3$ cases.

mcase

$\text{mcase}(d_i, \mathbf{m}_i)$ computes cases based on matching costs. Use both Birchfield-Tomasi (BT) cost and the L_1 norm of the difference of the gradients for disparity d_i at pixel i . Both are averaged over a 3×3 window. This results in the vector of matching costs \mathbf{m}_i . To compute $\text{mcase}(d_i, \mathbf{m}_i)$, divide BT costs into 18 bins, gradient differences into 20. Results in $360 = 20 \cdot 18$ cases.

Learning

$$l = \sum_{\{\hat{d}, \hat{g}, \hat{m}\}} \sum_i (f_d(\hat{d}_i, \hat{\mathbf{d}}_{\pi_i}, \mathbf{g}_i, \theta^d) + f_m(d_i, \mathbf{m}_i, \theta^m) - \log Z(\theta^d, \theta^m, \mathbf{d}_{\pi_i}, \mathbf{g}_i, \mathbf{m}_i))$$

Due to parameterization, cannot be learned by counting. Gradients of the likelihood with respect to the parameters are complicated to write down (see paper), but not hard to compute.

Use 100,000 samples from various images from Middlebury dataset.

Learning took about 30 minutes using L-BFGS.

Full maximum likelihood learning gives poor results.

Instead,

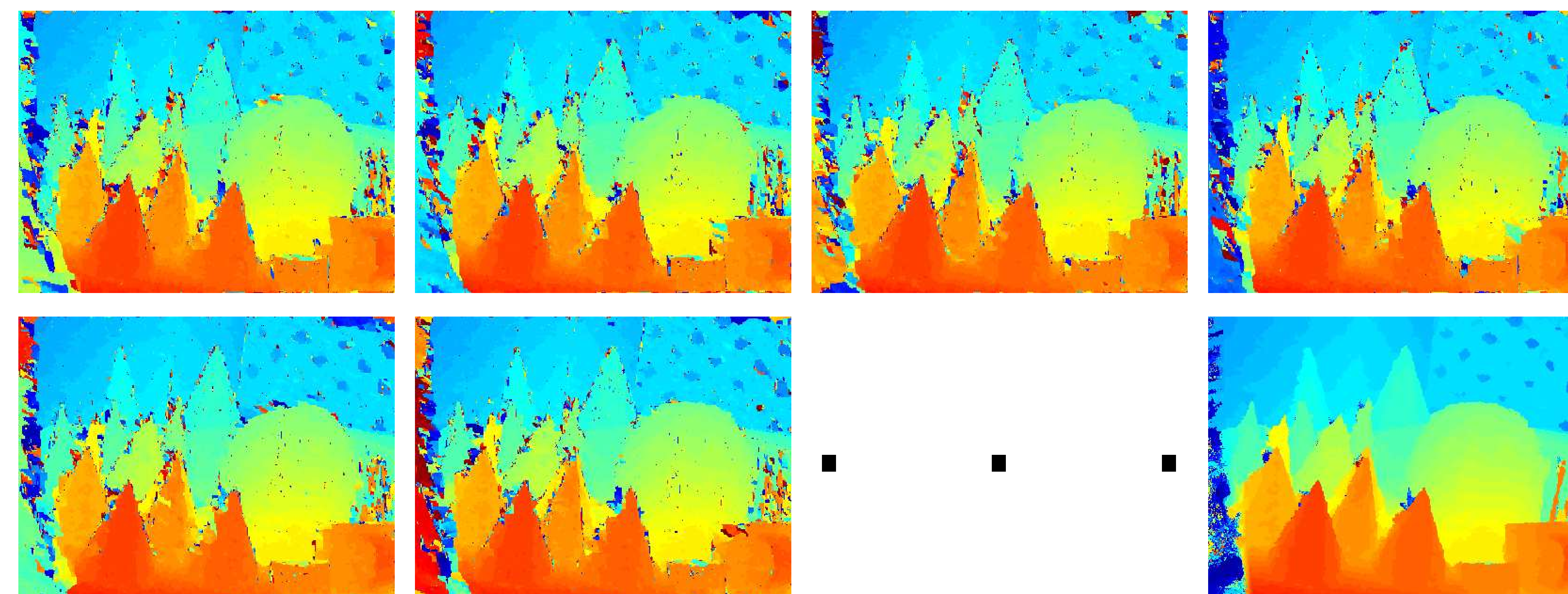
- 1) Train matching parameters θ^m with $f_d = 0$.
- 2) Train the smoothness parameters with θ^m held constant.
- 3) "Regularize" the smoothness term by changing the learnt θ^d to θ^d / λ , where $\lambda = 1.5$.

Why would this help? (See Justin's UAI 2008 paper.)

Inference

We want, for all i , $\arg \max_{d_i} p(d_i | \text{ims})$. This is not equivalent to seeking $\arg \max_{\mathbf{d}} p(\mathbf{d} | \text{ims})$. This can be approximated arbitrarily well by simply drawing samples from $p(\mathbf{d} | \text{ims})$, and counting how often each disparity occurs at each pixel. Can draw a perfect sample in $O(Nd)$, with N pixels and d disparities. Takes 0.1 seconds for Tsukuba (16 disparities), and 0.4 seconds for cones (64 disparities).

Illustration of Inference Process



Results



Results of Middlebury evaluation. (Numbers indicate the percentage of incorrect disparities.)

	Tsukuba	Venus	Teddy	Cones	Mean
Directed Sampling	3.9	3.6	10.5	4.2	5.6
Graph Cuts	1.9	1.8	16.5	7.7	7.0
CRFs ($K = 2$)	2.2	1.6	11.3	10.7	6.5