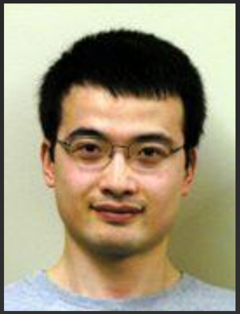
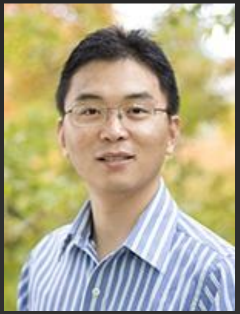
Stereo Matching with Nonparametric Smoothness Priors in Feature Space

CVPR 2009 MIAMI





Brandon M. Smith[1], Li Zhang[1], and Hailin Jin[2]

[1] University of Wisconsin – Madison

[2] Adobe Systems Incorporated

Adobe

Motivation

State-of-the-art two-view stereo methods
9 out of top 10 employ image segmentation

Stereo | Evaluation • Datasets • Code • Submit

Middlebury Stereo Evaluation - Version 2

Error Threshold = 1 (Error Threshold...)

Sort by nonocc · Sort by all · Sort by disc

Algorithm	Avg. Rank	Tsukuba ground truth nonocc	Tsukuba all	Tsukuba disc	Venus ground truth nonocc	Venus all	Venus disc	Teddy ground truth nonocc	Teddy all	Teddy disc	Cones ground truth nonocc	Cones all	Cones disc	Average percent of bad pixels (explanation)
AdaptingBP [17]	3.8	1.11 8	1.37 5	5.79 9	**0.10** 1	0.21 3	1.44 2	4.22 3	7.06 3	11.8 4	**2.48** 1	7.92 5	7.32 2	4.23
CoopRegion [41]	3.8	**0.87** 1	**1.16** 1	**4.61** 1	0.11 2	0.21 2	1.54 4	5.16 9	8.31 5	13.0 7	2.79 5	7.18 2	8.01 7	4.41
DoubleBP [35]	5.1	0.88 3	1.29 2	4.76 3	0.13 5	0.45 10	1.87 7	3.53 2	8.30 4	**9.63** 1	2.90 6	8.78 14	7.79 4	4.19
OutlierConf [42]	5.9	0.88 2	1.43 7	4.74 2	0.18 9	0.26 6	2.40 11	5.01 6	9.12 7	12.8 6	2.78 4	8.57 10	**6.99** 1	4.60
SubPixDoubleBP [30]	8.2	1.24 15	1.76 16	5.98 10	0.12 4	0.46 11	1.74 6	**3.45** 1	8.38 6	10.0 2	2.93 8	8.73 13	7.91 6	4.39
WarpMat [55]	9.5	1.16 9	1.35 4	6.04 11	0.18 10	0.24 5	2.44 12	5.02 7	9.30 8	13.0 9	3.49 13	8.47 9	9.01 17	4.98
Undr+OvrSeg [48]	12.4	1.89 31	2.22 29	7.22 26	0.11 3	0.22 4	**1.34** 1	6.51 14	9.98 9	16.4 14	2.92 7	8.00 6	7.90 5	5.39
GC+SegmBorder [57]	13.3	1.47 25	1.82 18	7.86 29	0.19 11	0.31 7	2.44 12	4.25 4	**5.55** 1	10.9 3	4.99 36	**5.78** 1	8.66 12	4.52
AdaptOvrSegBP [33]	13.8	1.69 28	2.04 26	5.64 8	0.14 6	**0.20** 1	1.47 3	7.04 20	11.1 11	16.4 16	3.60 16	8.96 16	8.84 14	5.59
SymBP+occ [7]	15.4	0.97 6	1.75 15	5.09 6	0.16 7	0.33 8	2.19 9	6.47 13	10.7 10	17.0 20	4.79 33	10.7 30	10.9 28	5.92

Motivation

Image segmentation problems

Motivation

Segmentation artifacts in video: temporal instability

1 of 5 input views

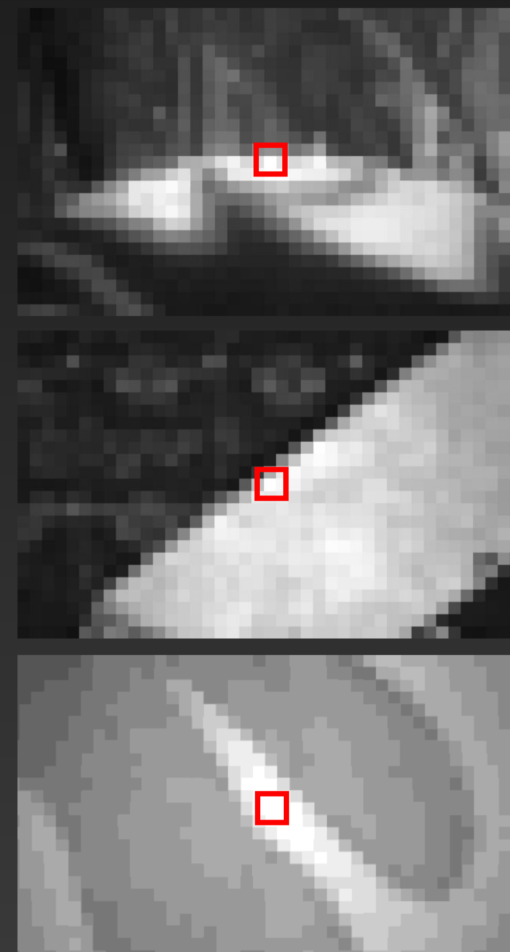
2nd-order smoothness method ***with segmentation*** [Woodford et al. CVPR ’08]

Inspiration: Adaptive Support Weight

Yoon & Kweon, Locally adapt. support-weight approach for vis. corr. search, CVPR '05

Close-up views of matching window

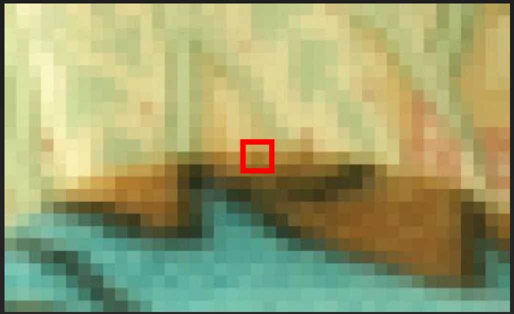
Intensity-encoded weights



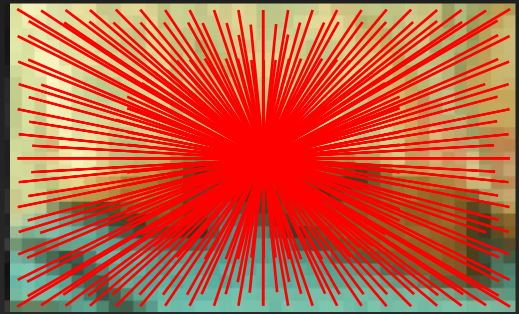
Inspiration: Adaptive Support Weight

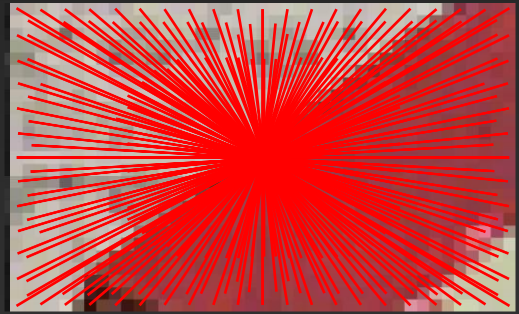
Can we incorporate this idea into a global inference algorithm?

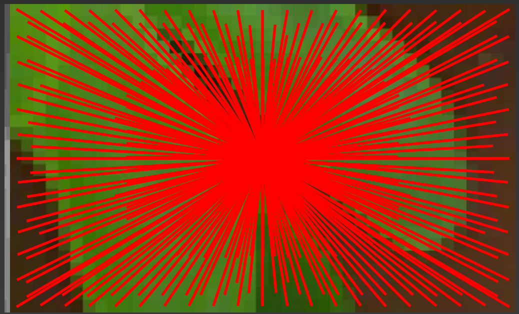
Close-up views of matching window



Large, weighted smoothness nbrhood

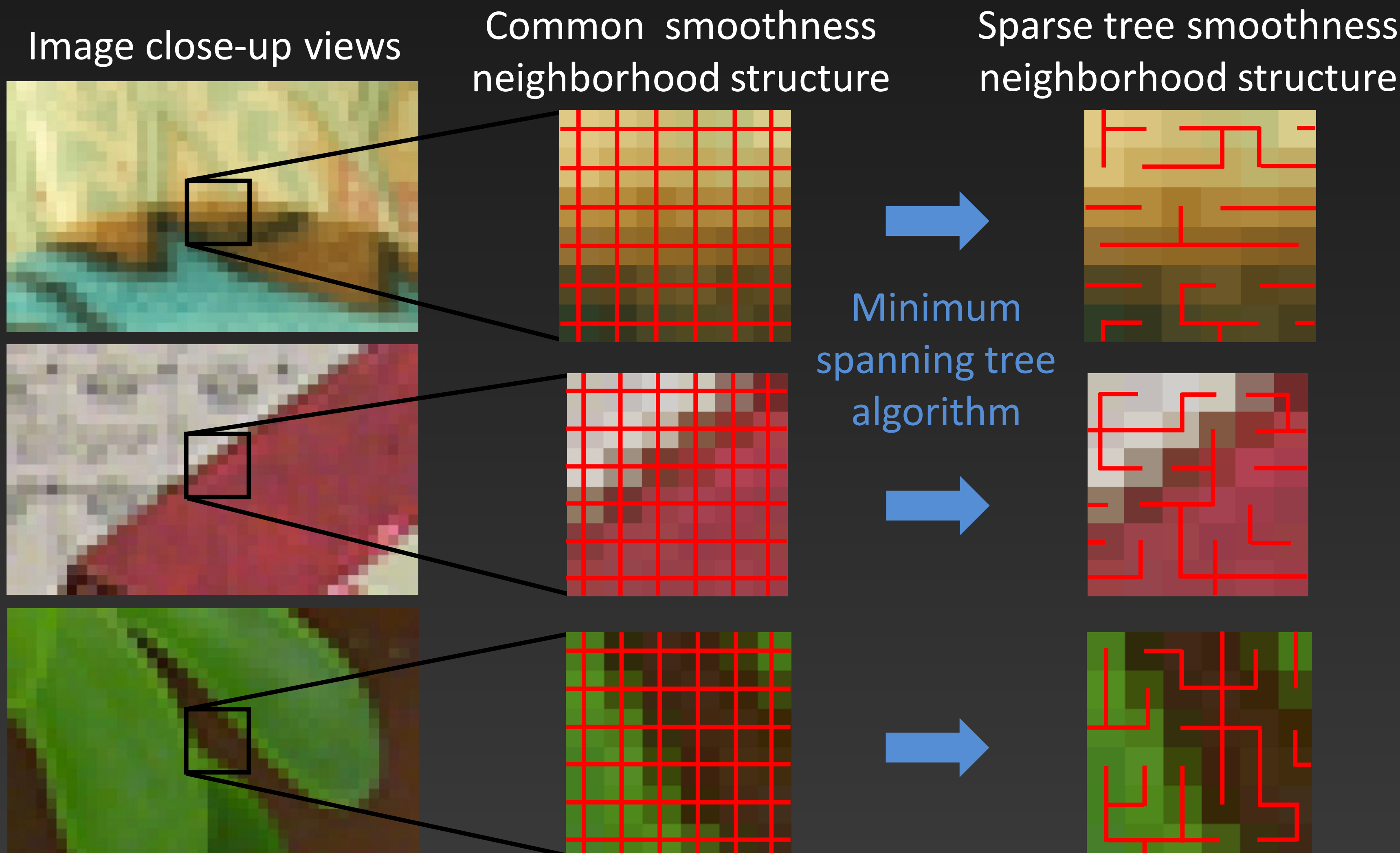






Inspiration: Sparse Neighborhoods

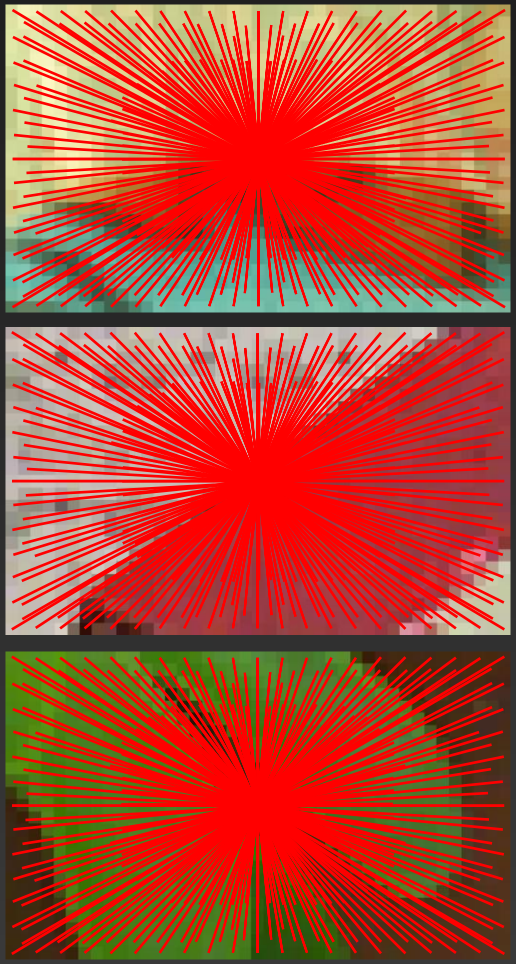
O. Veksler, Stereo Correspondence by Dynamic Programming on a Tree, CVPR '05



Our Approach

Global inference using large, sparse smoothness neighborhoods

Large, weighted smoothness nbrhood



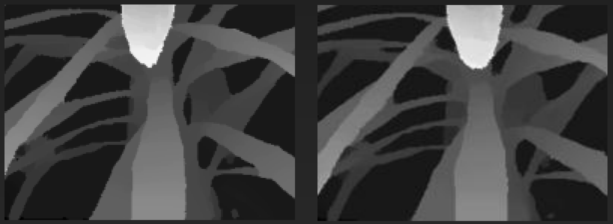
Minimum spanning tree algorithm

Most important edges

Problem Formulation

Given a stereo image pair, I_1 and I_2,

compute disparity maps, D_1 and D_2



by minimizing:

$$\Phi(D_1, D_2) = \Phi_{\mathrm{ph}}(D_1, D_2) + \Phi_{\mathrm{sm}}(D_1) + \Phi_{\mathrm{sm}}(D_2)$$

Energy Minimization Function

$$\Phi_{\mathrm{ph}}\left(D_1, D_2\right) \quad \Phi_{\mathrm{sm}}\left(D_1\right) \quad \Phi_{\mathrm{sm}}\left(D_2\right)$$

Photo consistency term
[Kolmogorov & Zabih ECCV '02]

Smoothness (regularization) terms

$$\Phi\left(D_1, D_2\right) = \Phi_{\mathrm{ph}}\left(D_1, D_2\right) + \Phi_{\mathrm{sm}}\left(D_1\right) + \Phi_{\mathrm{sm}}\left(D_2\right)$$

Spatial Smoothness Terms

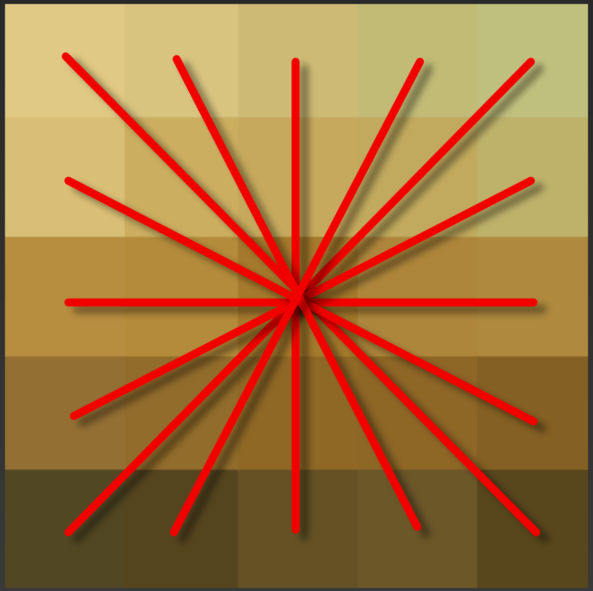
$$\Phi(D_1, D_2) = \Phi_{ph}(D_1, D_2) + \Phi_{sm}(D_1) + \Phi_{sm}(D_2)$$

Previous global methods:

- 1^{st}-order smoothness priors
- 2^{nd}-order smoothness priors

Another approach:

- Kernel density estimation



Large neighborhood

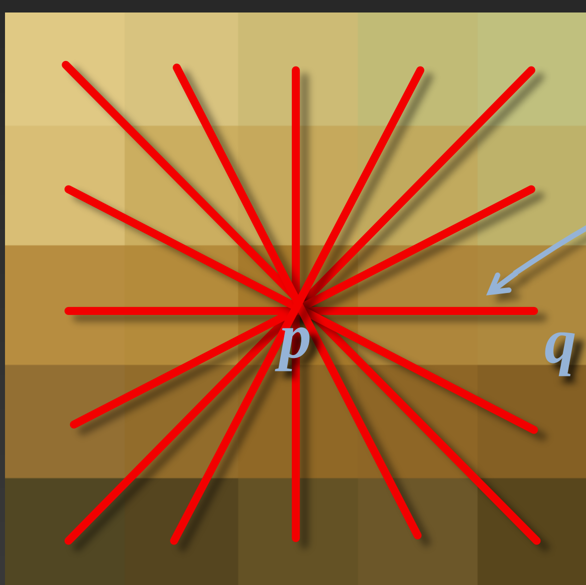
Kernel Density Estimation

$$\Phi(D_1, D_2) = \Phi_{\mathrm{ph}}(D_1, D_2) + \Phi_{\mathrm{sm}}(D_1) + \Phi_{\mathrm{sm}}(D_2)$$

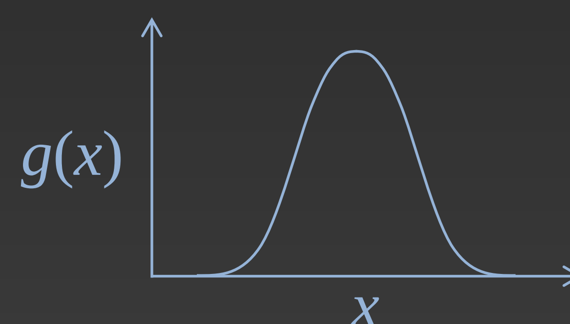
Weight based on proximity, color between p,q [Yoon & Kweon CVPR`05]

$$\sum_{q \in \mathcal{N}_p} w_{p,q} \; \overbrace{g_d\left(\frac{d_p - d_q}{\sigma_d}\right)}^{\text{Kernel function for disparity}}$$

$$w_{p,q} = \frac{1}{|\mathcal{N}_p|} \, g_{\mathrm{x}}\left(\frac{\mathbf{x}_p - \mathbf{x}_q}{\sigma_{\mathrm{x}}}\right) \, g_{\mathrm{c}}\left(\frac{\mathbf{c}_p - \mathbf{c}_q}{\sigma_{\mathrm{c}}}\right)$$



Large neighborhood



Upperbound Approximation

$$\Phi(D_1, D_2) = \Phi_{\mathrm{ph}}(D_1, D_2) + \Phi_{\mathrm{sm}}(D_1) + \Phi_{\mathrm{sm}}(D_2)$$

Weight based on proximity, color between p,q [Yoon & Kweon CVPR`05]

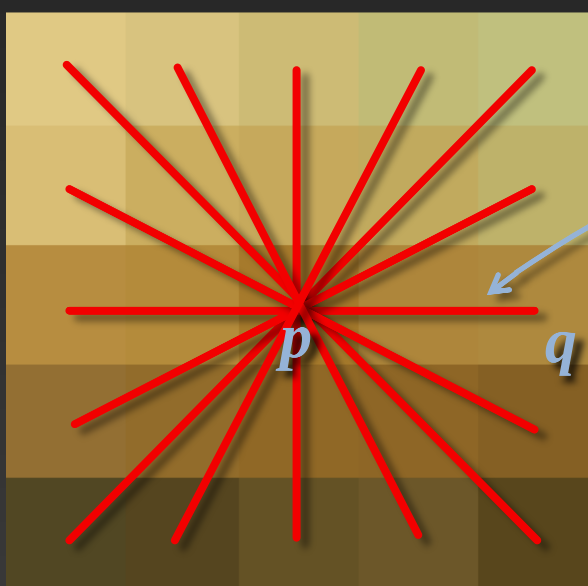
Kernel function for disparity difference

Truncated linear function

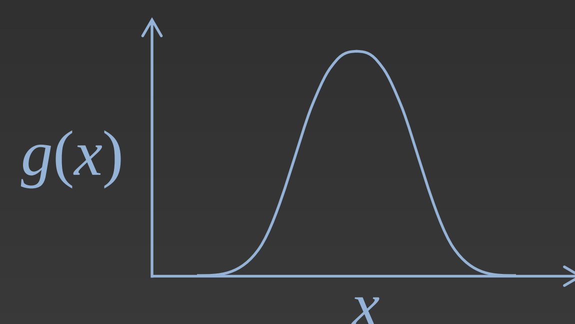
$$\Phi_{\mathrm{sm}}(D) = \sum_{p \in I} \sum_{q \in \mathcal{N}_p} w_{p,q} \, g_d\left(\frac{d_p - d_q}{\sigma_d}\right)$$

$$\Phi_{\mathrm{sm}}(D) = \sum_{p \in I} \sum_{q \in \mathcal{N}_p} w_{p,q} \min(\lambda |d_p - d_q|, \tau)$$

$$\frac{1}{|\mathcal{N}_p|} g_{\mathrm{x}}\left(\frac{\mathbf{x}_p - \mathbf{x}_q}{\sigma_{\mathrm{x}}}\right) g_{\mathrm{c}}\left(\frac{\mathbf{c}_p - \mathbf{c}_q}{\sigma_{\mathrm{c}}}\right)$$



Large neighborhood

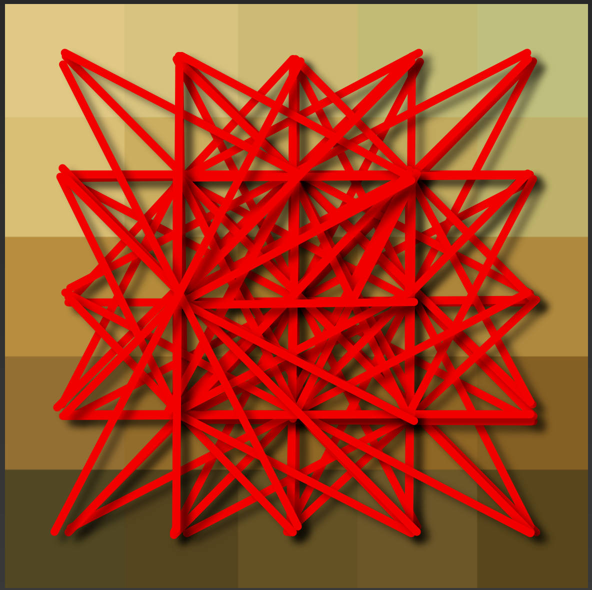


Optimization Challenge

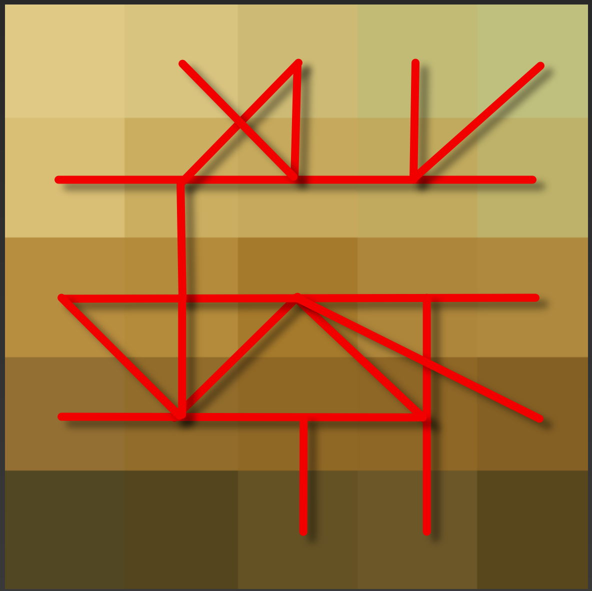
Φ_{sm} is dense, expensive to minimize

Solution: use a sparse approximation

$$\Phi_{sm}(D) = \sum_{p \in I} \sum_{q \in \mathcal{N}_p} w_{p,q} \min\left(\lambda \left|d_p - d_q\right|, \tau \right)$$



Large neighborhood

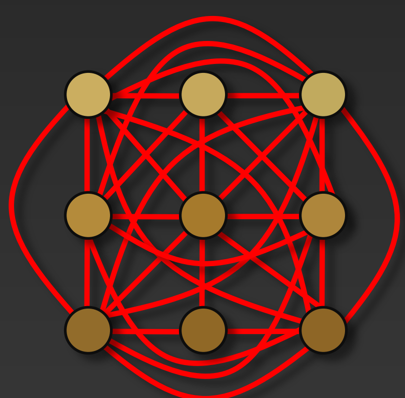


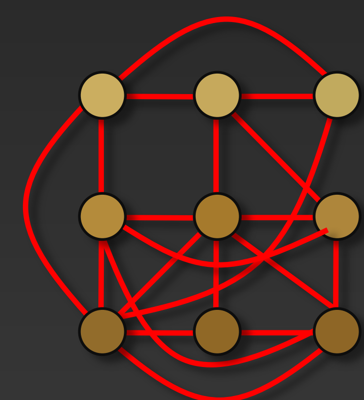
Sparse Graph Approximation

Φ_{sm} is dense, expensive to minimize

Solution: use a sparse graph approximation

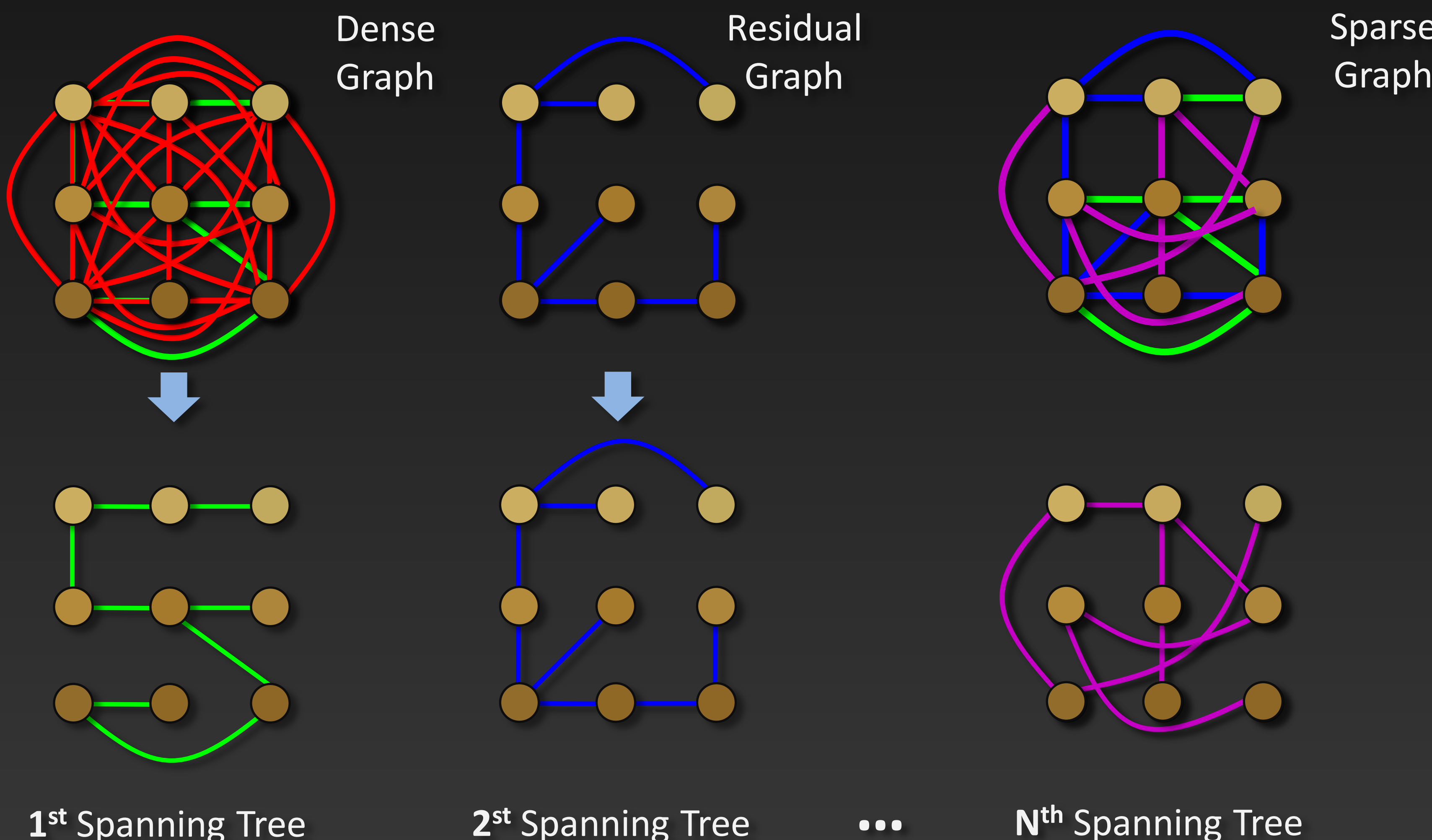
$$\Phi_{sm}(D) = \sum_{p \in I} \sum_{q \in \mathcal{N}_p} w_{p,q} \min\left(\lambda \, |d_p - d_q|, \tau \right)$$





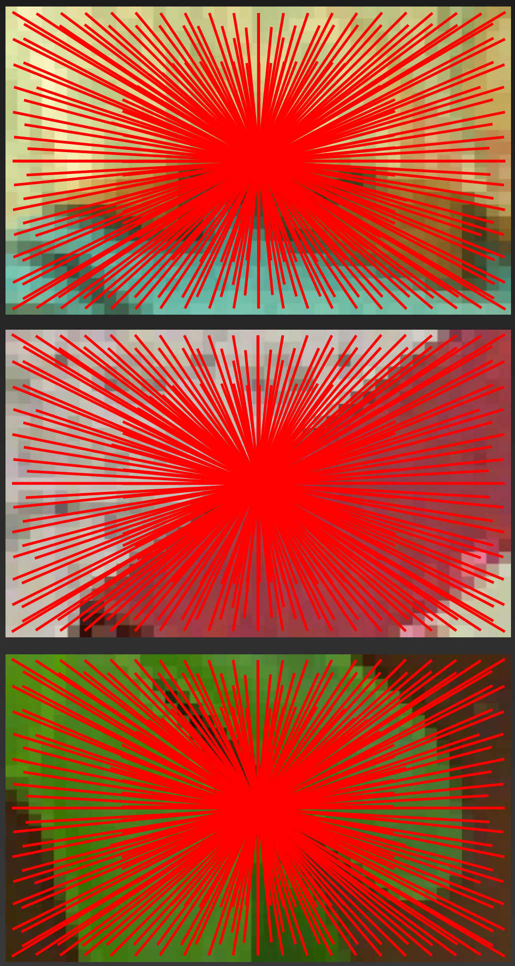
Sparse Graph Approximation

CVPR 2009 MIAMI



Graph Edges On Real Images

Dense smoothness neighborhood



Minimum spanning tree algorithm

Sparse smoothness neighborhood

Connection to Image Segmentation

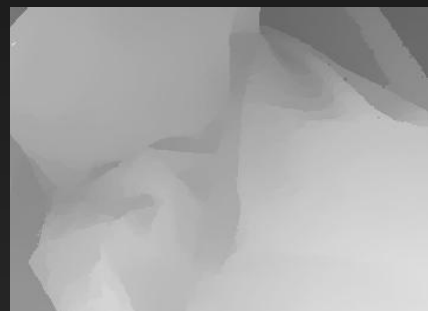
Kruskal's minimum spanning tree algorithm

0% pixels connected | 82% pixels connected | 95% pixels connected | 98% pixels connected | 100% pixels connected

C. Zahn. Graph-theoretic methods for detecting and describing gestalt clusters. IEEE Trans. on Computing, 1971.

Results on Stereo Images

Left input image



Our result

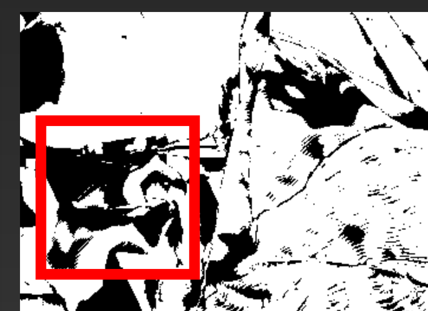
Multi-view graph cuts result [Kolmogorov & Zabih, '02]

Second-order smoothness [Woodford et al. '08]

Ground truth

3.41% bad pixels

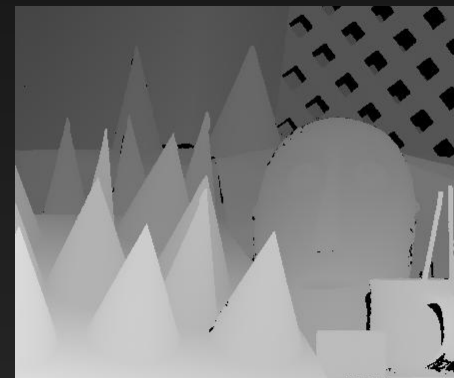
4.82% bad pixels



25.06% bad pixels

Results on Stereo Images

Left input image



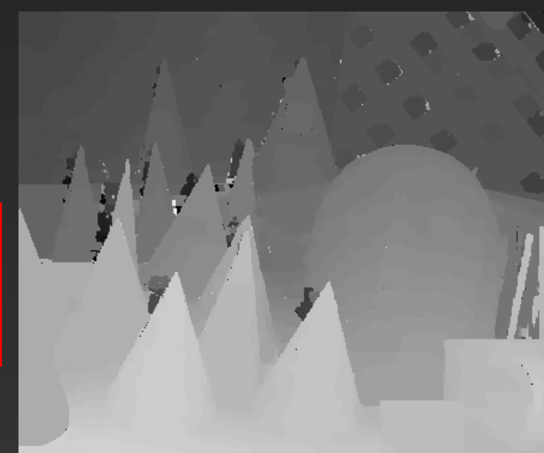
Ground truth depth

3.29% bad pixels

Our results
(tailored parameters)

2.48% bad pixels

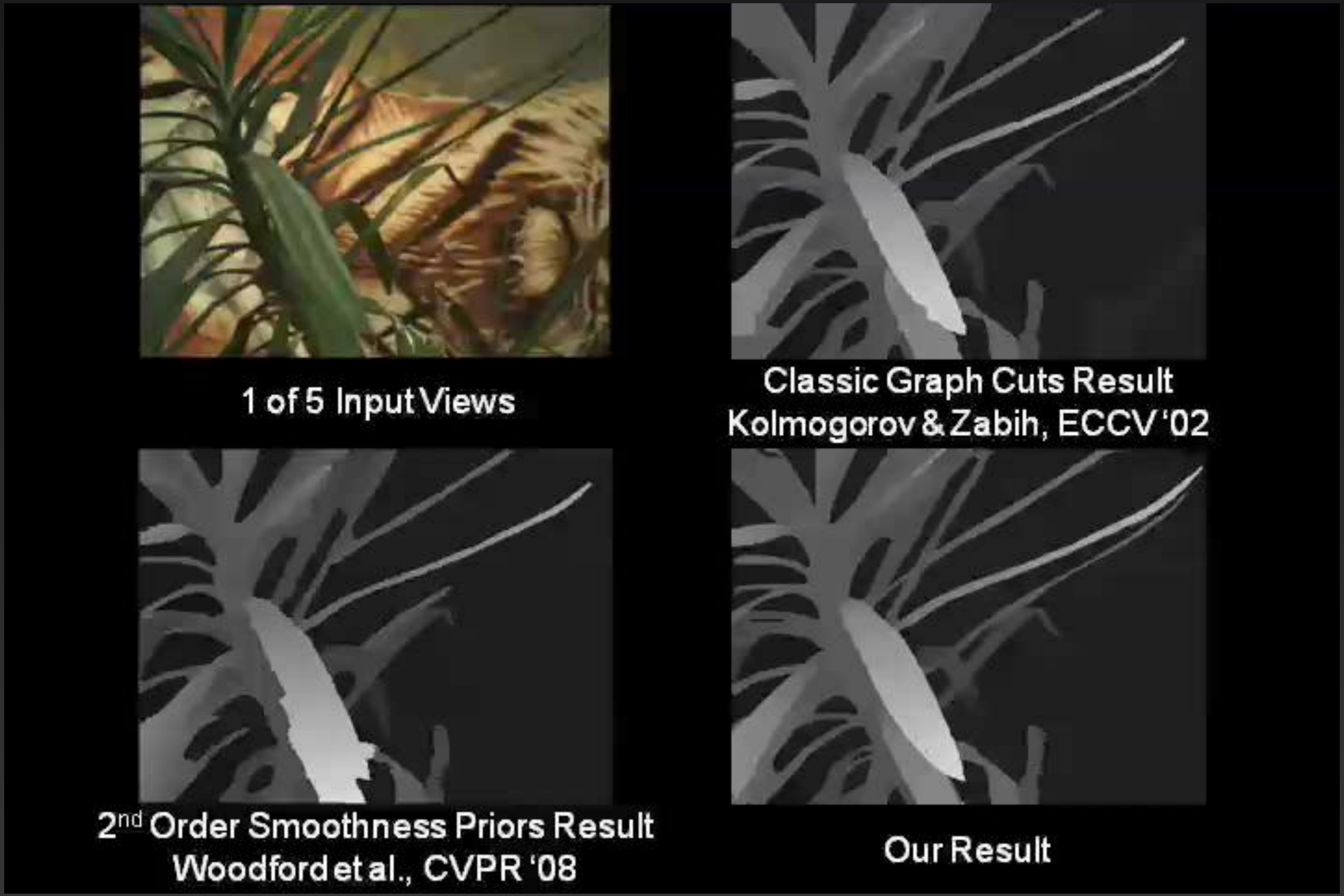
Klaus et al. ‘06 results



4.21% bad pixels

Multi-view graph cuts
[Kolmogrov & Zabih ‘02]
(tailored parameters)

Results on Videos



Future Work

- $w_{p,q}$ generalizes to any feature vector (not just x,y,r,g,b) $\longrightarrow$ explore other feature vectors

- Automatic parameter estimation (scale in kernel function)

- Better handle View-dependent brightness inconsistencies