UCU Summer School

Alexander Shekhovtsov
Kostiantyn Antonuk

Outline

- D1 Starting toolbox for statistical recognition
- D2 Structured prediction
- D3 Learning for structured prediction
  - Structured output SVM, advanced examples
  - Cutting Plane methods

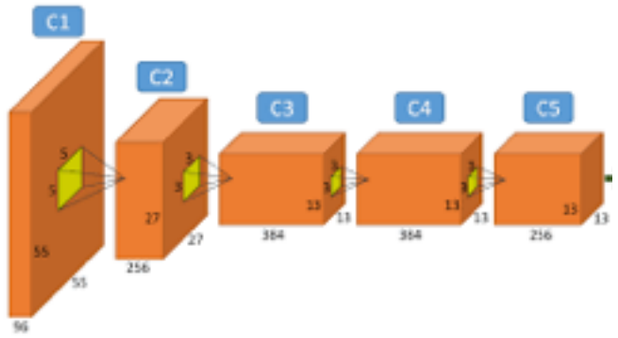
CNN+CRF Stereo

Patrick Knöbelreiter, Christian Reinbacher, Alexander Shekhovtsov, Thomas Pock

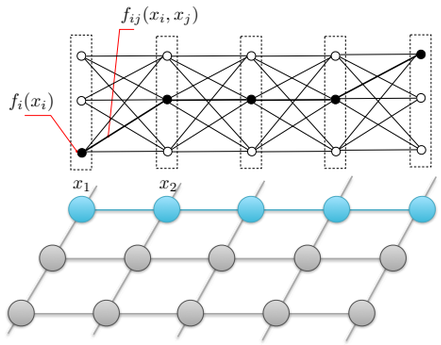
End-to-End Training of Hybrid CNN+CRF Models for Stereo

Hybrid Inference Models

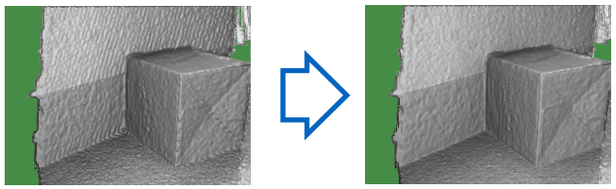
CNN features



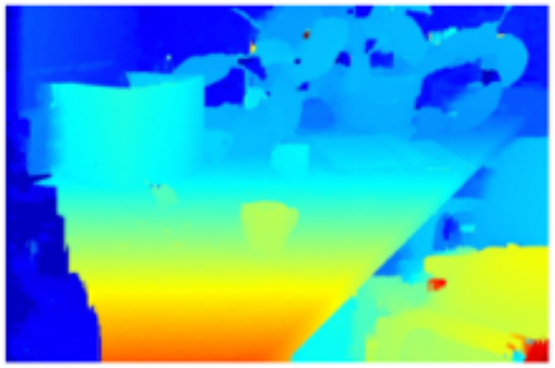
Discrete Inference

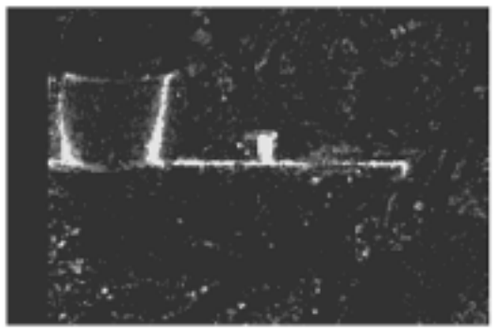


Refinement



Model Overview





Model Overview

Adaptive regularizer: $w_{ij}\rho(|x_i - x_j|)$

Commonly applied in segmentation, stereo

Contrast sensitive (gradient) model: $w_{ij} = \exp(\alpha|I_i - I_j|^\beta)$

Contrast Sensitive / Pairwise CNN

I_0 → Unary CNN

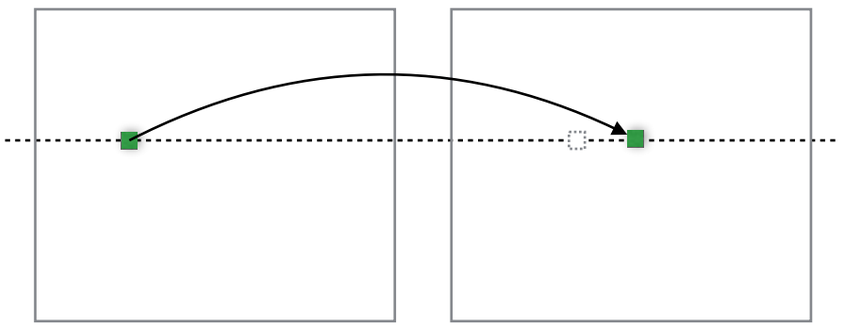
I_1 → Unary CNN

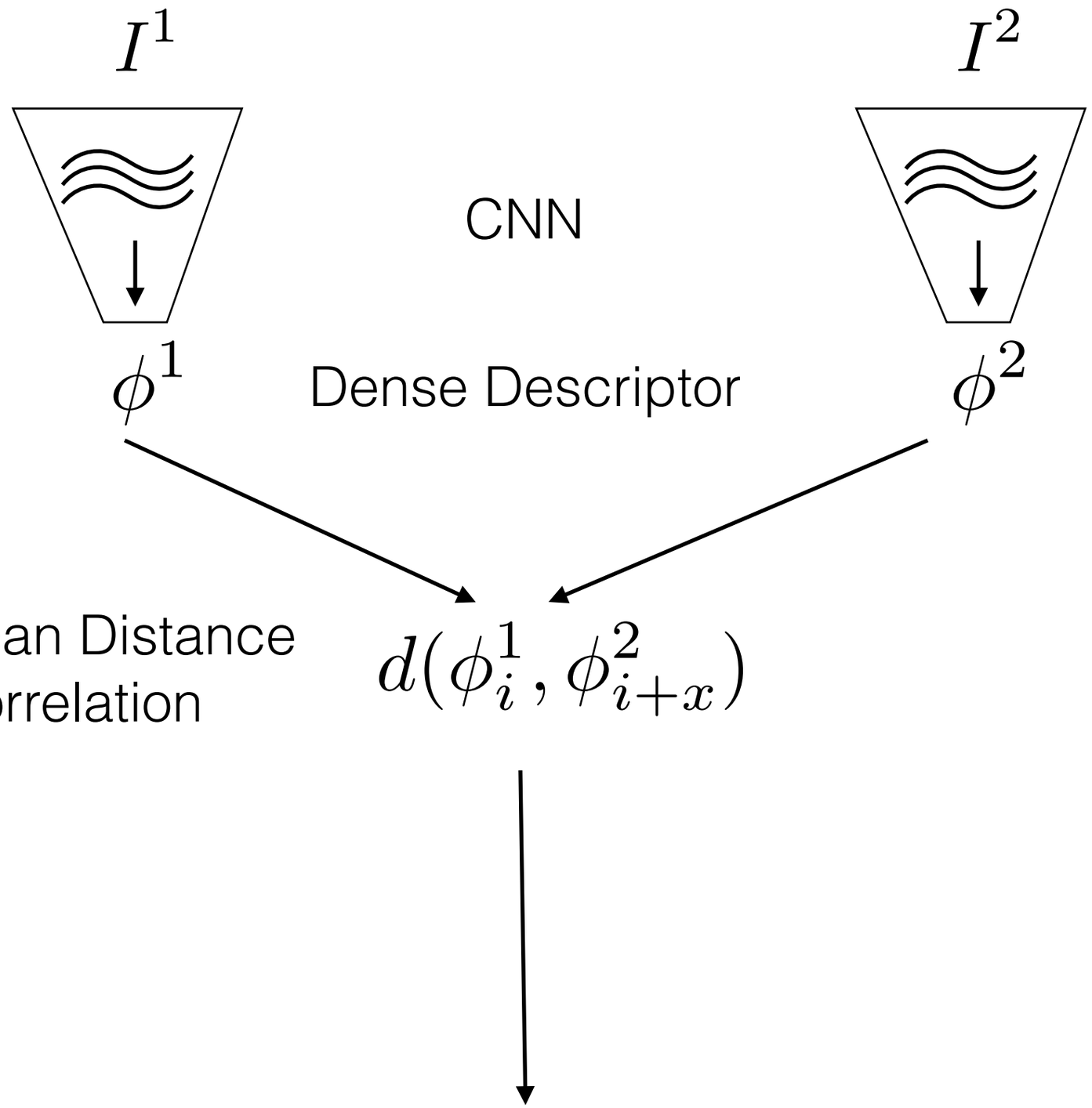
Correlation → CRF → D

Generalizes engineered features
Occurs in many matching problems: image retrieval, optical flow, stereo

Replaces post-processing such as
Cost aggregation / filtering, SGM, etc.

Local Matching Costs



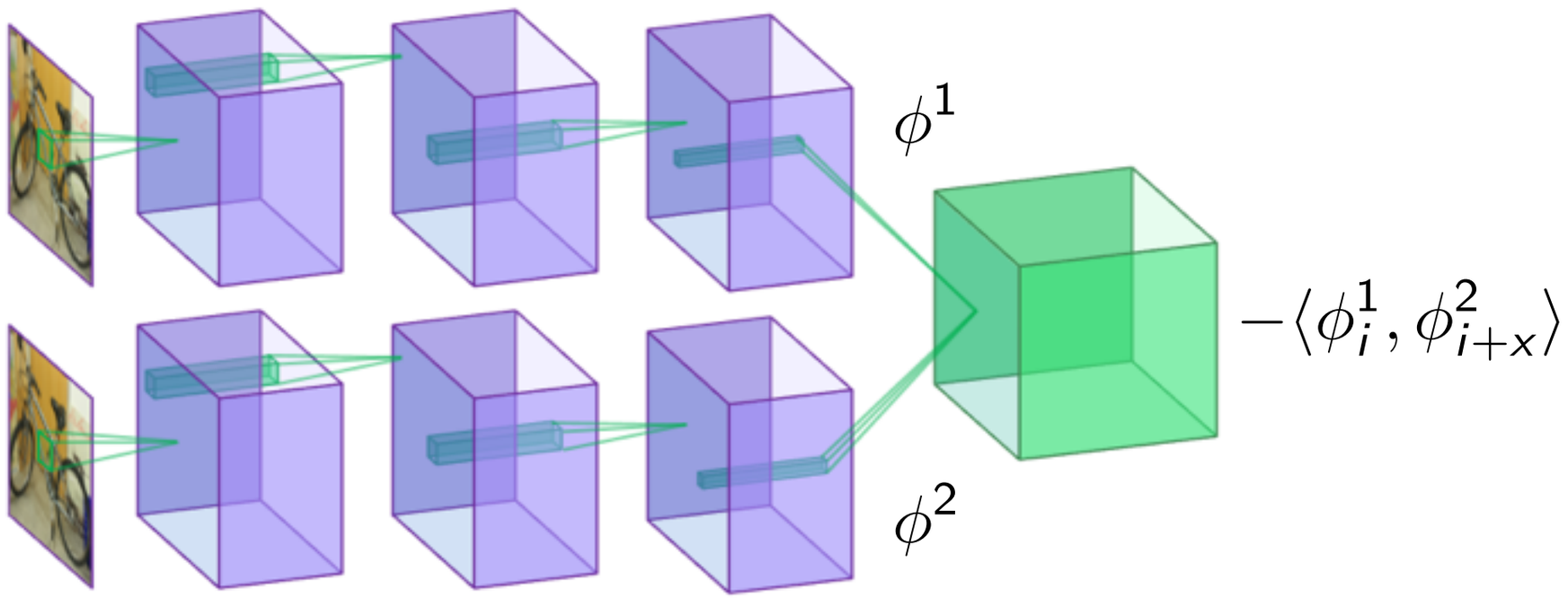


- This part generalizes engineered matching costs
- Can apply all techniques with it!

The building blocks: Unary CNN & Correlation

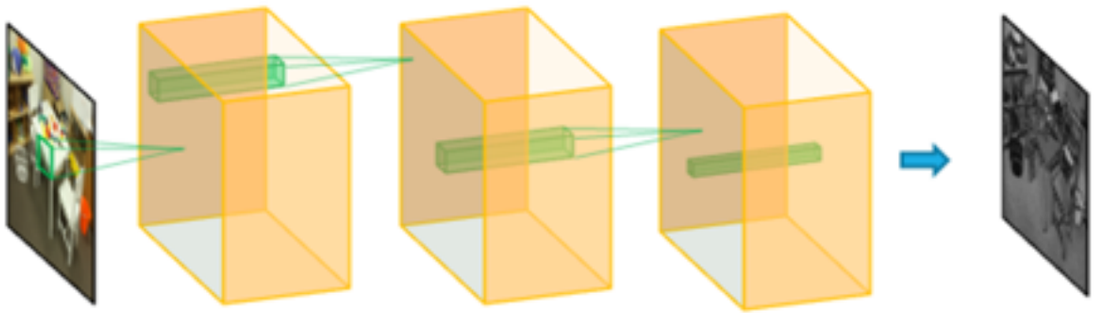
Unary CNN

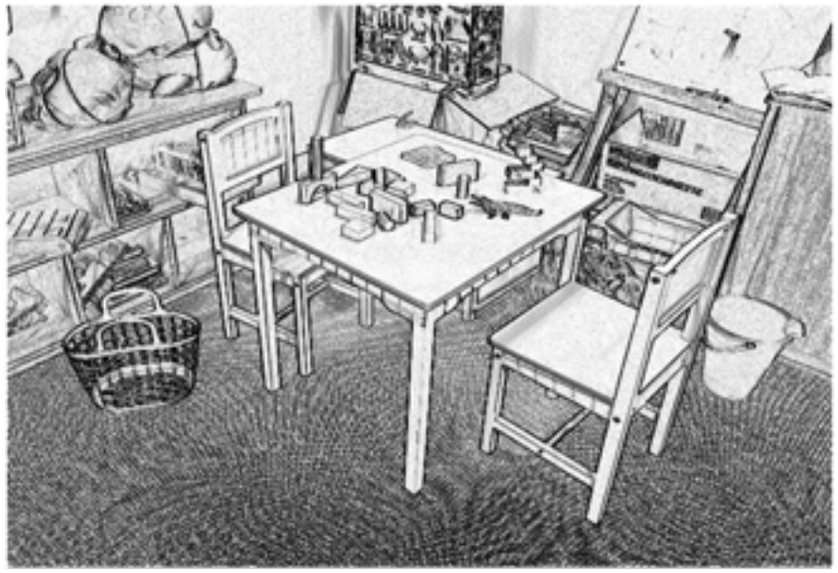
- 3-7 convolutional layers
- 83k – 243k parameters
- Learn optimal features for stereo-matching
- Parameters are shared between left and right image



The building blocks: Pairwise CNN

- 3 layers:
- Courage label jumps at strong object boundaries
- Discourage label jumps in homogenous regions





Engineered

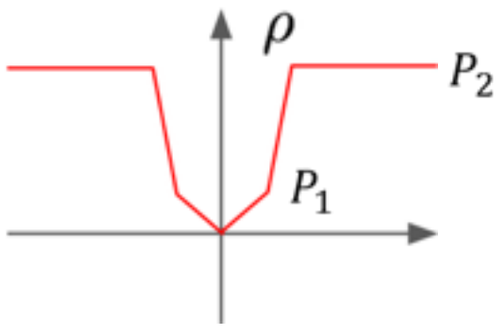
Learned

The building blocks: CRF

- Optimizes the total cost of data and regularizer on a 4-conncted pixel grid

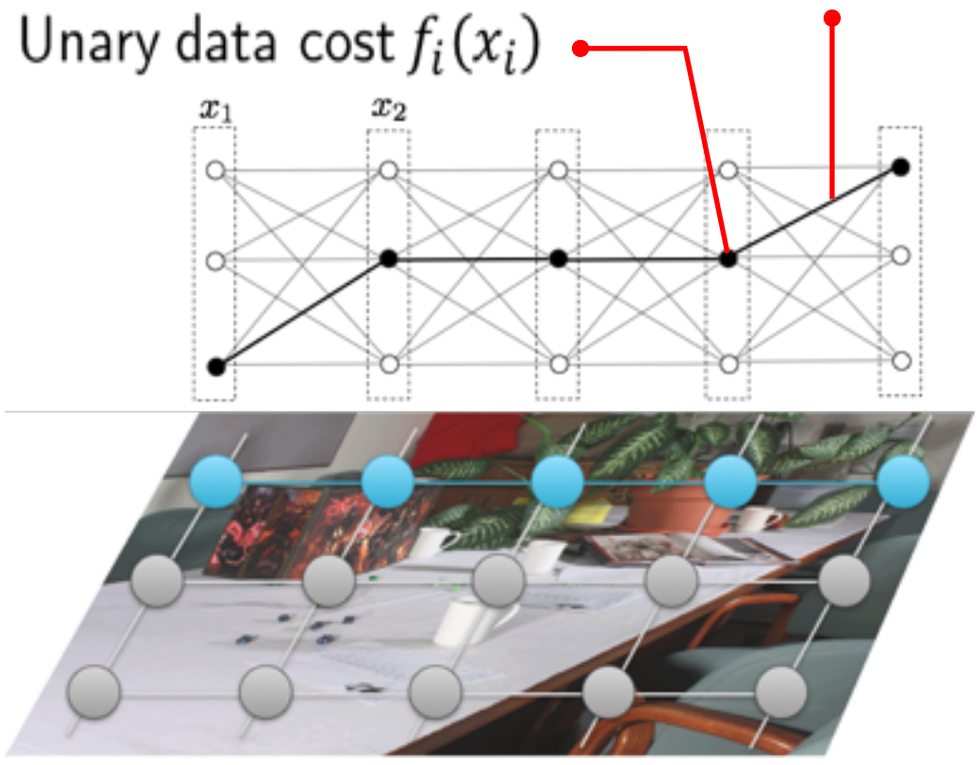
$$\min_{x \in V^L} \sum_{i \in V} f_i(x_i) + \sum_{ij \in E} f_{ij}(x_i, x_j)$$

$$\rho(d) = \begin{cases} 0 & if\ d = 0 \\ P_1 & if\ |d| = 1 \\ P_2 & otherwise \end{cases}$$



Pairwise Regularizer $f_{ij}(x_i, x_j) = w_{ij}\rho(|x_i, x_j|)$

Unary data cost $f_i(x_i)$



- Inference using Dual Minorize Maximize (DMM)
  - Similar to other LP-based approaches, but parallel, on the GPU

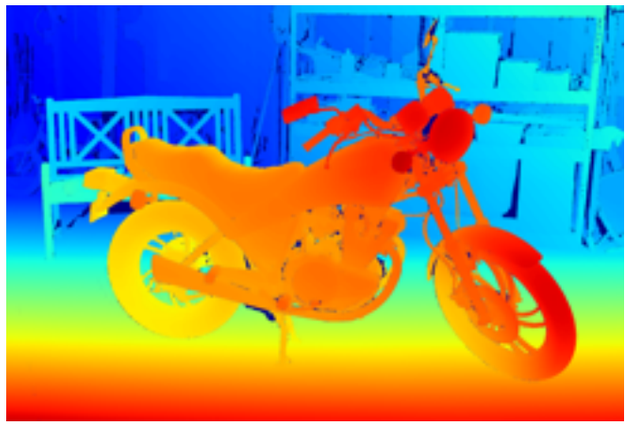
End-to-end learning

- The bilevel problem

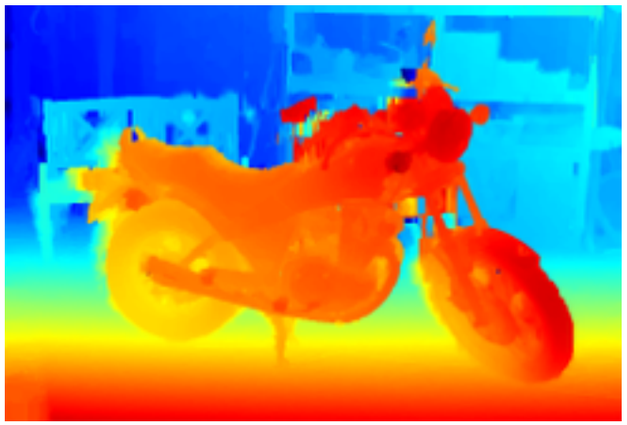
$$\min_{\theta} l(\hat{x}, x^*)$$

$$\hat{x} \in \operatorname*{argmin}_{x} E(x; \theta)$$

x^*



$\hat{x}(\theta)$



$$l(\hat{x}, x^*) = \sum_i |x_i - x_i^*|$$

  - It is in general hard to differentiate minimizers
  - Derivative of discrete minimizer in theta is zero almost everywhere

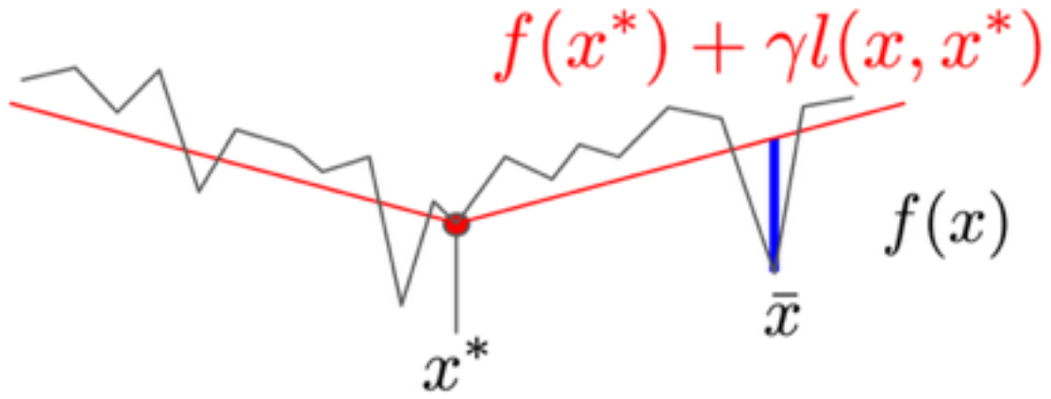
- Apply a convex proxy - margin rescaling Structured SVM
  - [Tsochantaridis et al. '05]

Structured SVM

- Want: GT disparity map x^* is better than any other solution by a margin proportional to loss

$$\exists \theta \quad \forall x \in \mathcal{V}^L \quad f(x^*) \leq f(x) - \gamma l(x, x^*)$$

Not always feasible!



- Minimize the most violated constraint:

$$\min_{\theta} \max_{x} \Big(\underbrace{f(x^*) - f(x) + \gamma l(x, x^*)}_{\text{Upper bound on the risk}} \Big)$$

- A subgradient in CRF cost vector $\bar{f}$ is given by $\delta(x^*) - \delta(\bar{x})$, where $f(x) = \langle \bar{f}, \delta(x) \rangle$ and

$$\bar{x} \in \arg\min_{x} \Big(f(x) - \gamma l(x, x^*) \Big)$$

Loss-augmented inference problem

- Back-propagate to get gradient in θ

Training

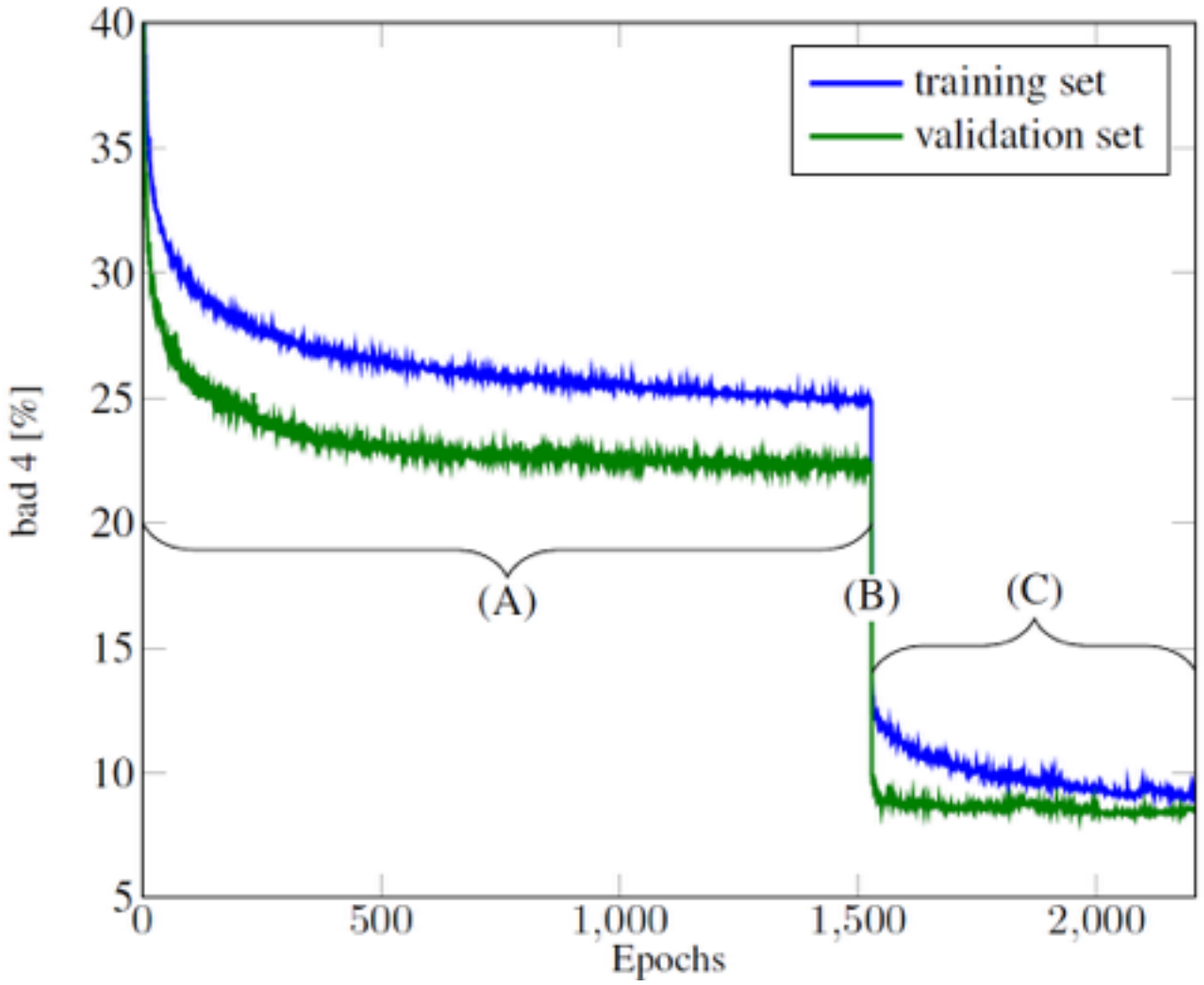
Training

- stochastic (sub)gradient descent with momentum
- unary-CNN pre-training
- joint training

Databases

- Middlebury Stereo v3

- Kitti 2015



Training curves

Middlebury Stereo v3

Comparing our models

- Standard metric: *bad4*
  - Percentage of „bad“ pixels whose error is greater than 4
- Results are computed on quarter-size images and then upsampled to full size for evaluation
- Fast: 285ms for 640px x 480px

Method	CNN	+CRF	+Joint	+Pairwise
CNN3	23.89	11.18	9.48	9.45
CNN7	18.58	9.35	8.05	7.88

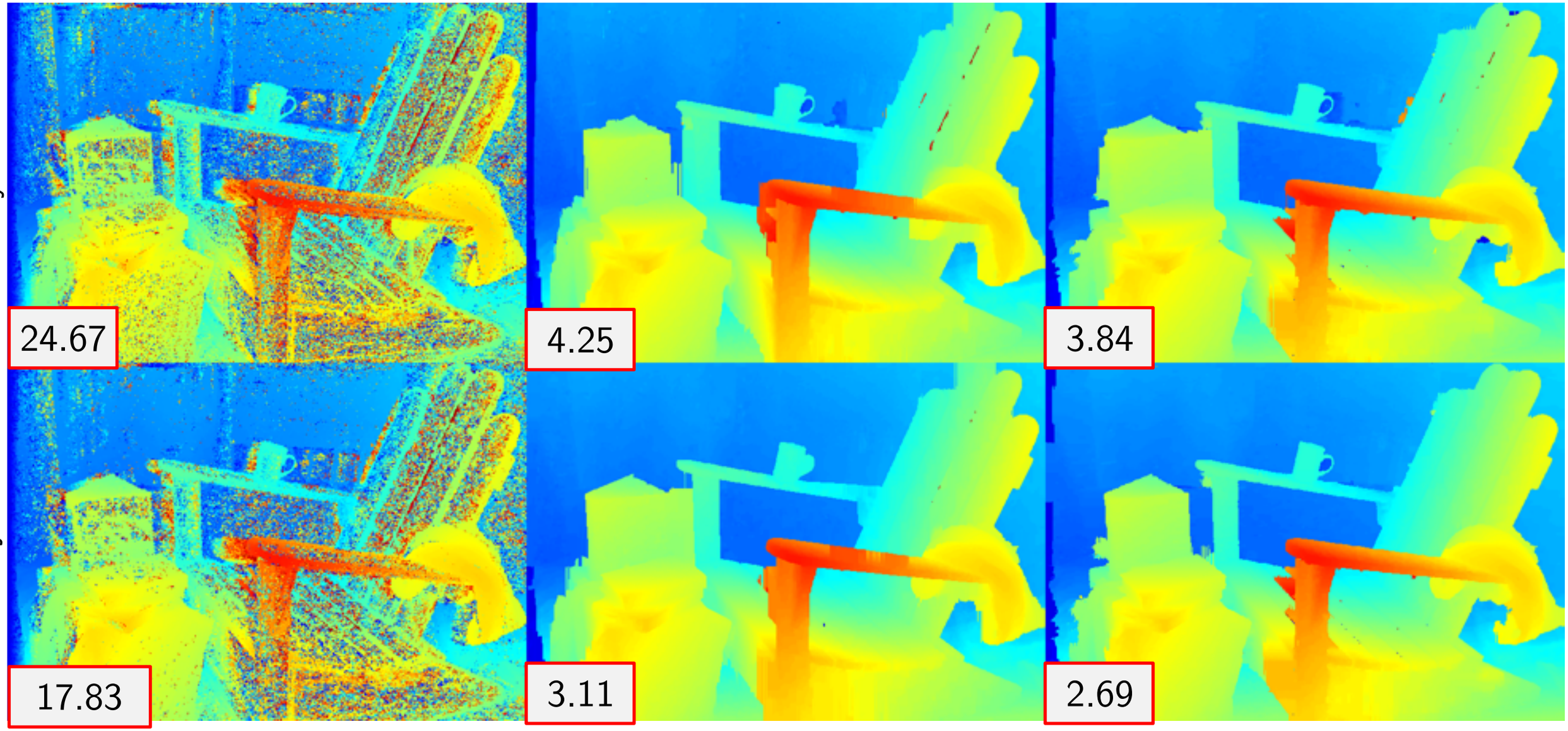
Effect of Joint Training

Unary CNN

Unary CNN + CRF

Full Joint

3 layer



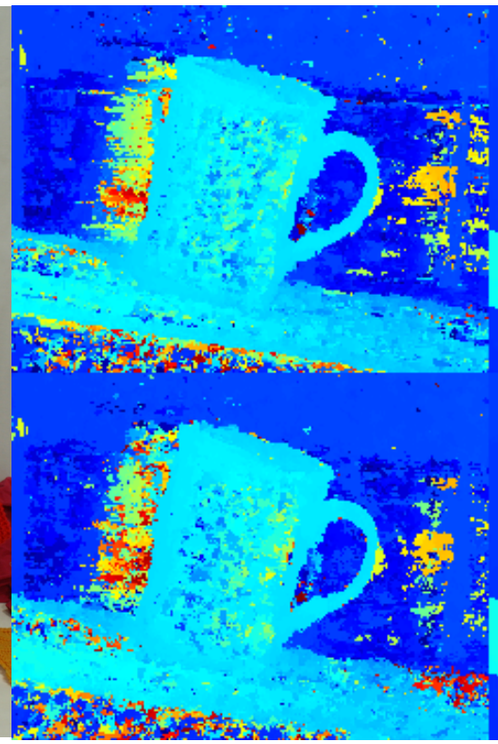
7 layer

5 iterations of DMM

Effect of Joint Training

Input

CNN



+CRF

+Joint+PW

Middlebury Stereo v3

- Error metrics
  - Bad2: Percentage of „bad“ pixels whose error is greater than 2
  - RMS: root mean-squared error
- Very large images: 3000px x 2000px
- Currently rank 1 with RMS error, rank 6 with bad2 error

Method	Ø RMS	Ø bad2	Time/MP	Parameters	Post-Processing
MC-CNN	21.3	**8.29**	112s	830k	CA, SGM, SE, MF, BF
MC-CNN + RBS	15.0	8.62	140s	830k	CA, SGM, SE, MF, BF, RBS
Ours	**14.4**	12.5	**3.69s**	**281k**	**-**

CA...Cost Aggregation, SGM...Semi-Global Matching, SE...Sublabel Enhancement, MF...Median Filtering, BF...Bilateral Filtering, RBS... Robust Bilateral Solver

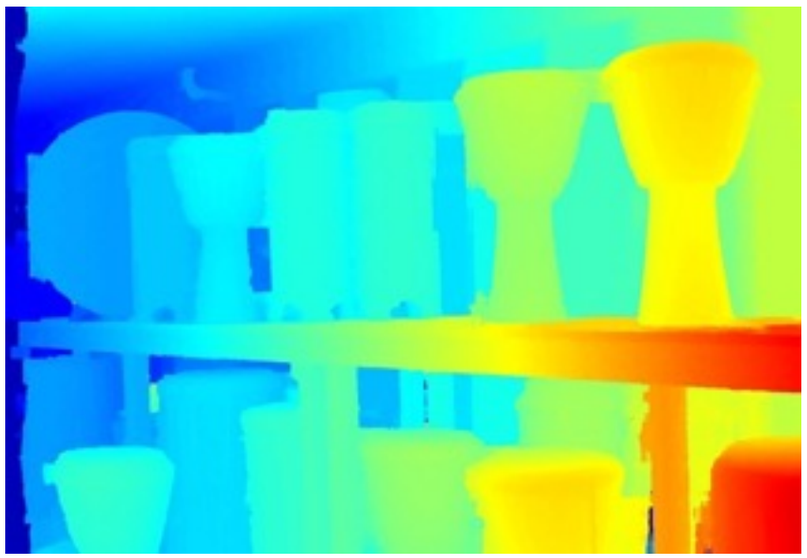
Middlebury Stereo v3 Test Results

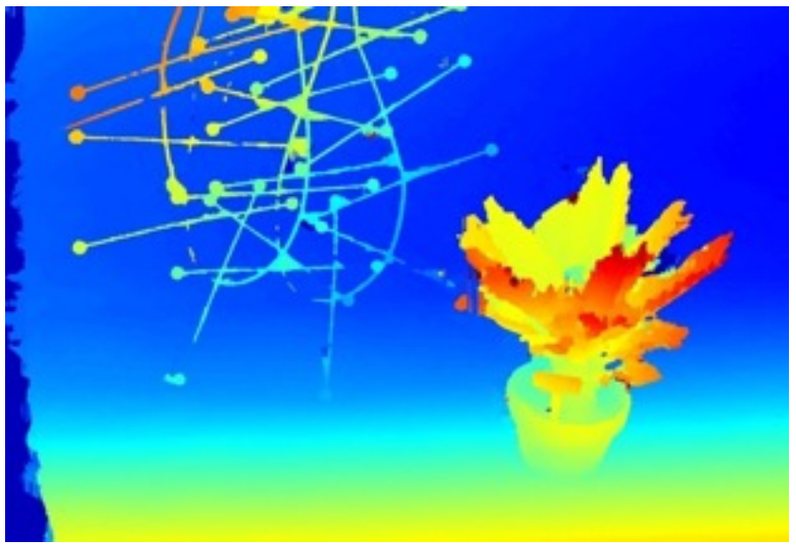
Djembe

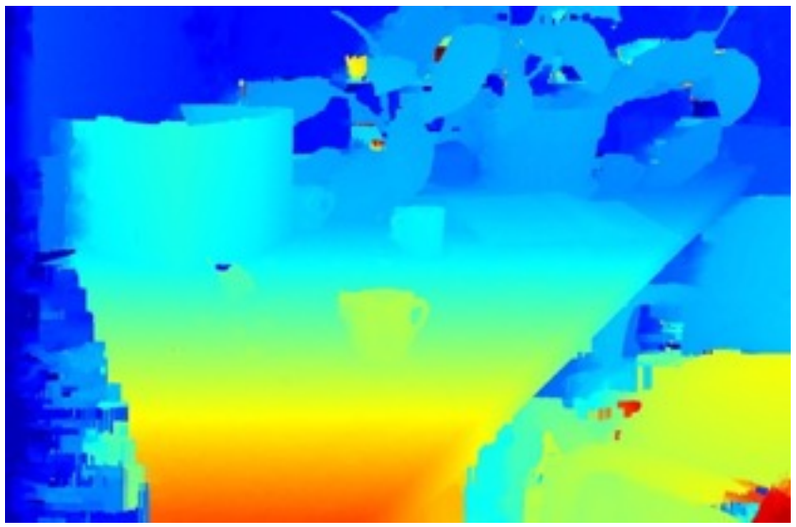
Australia

Crusade

Result

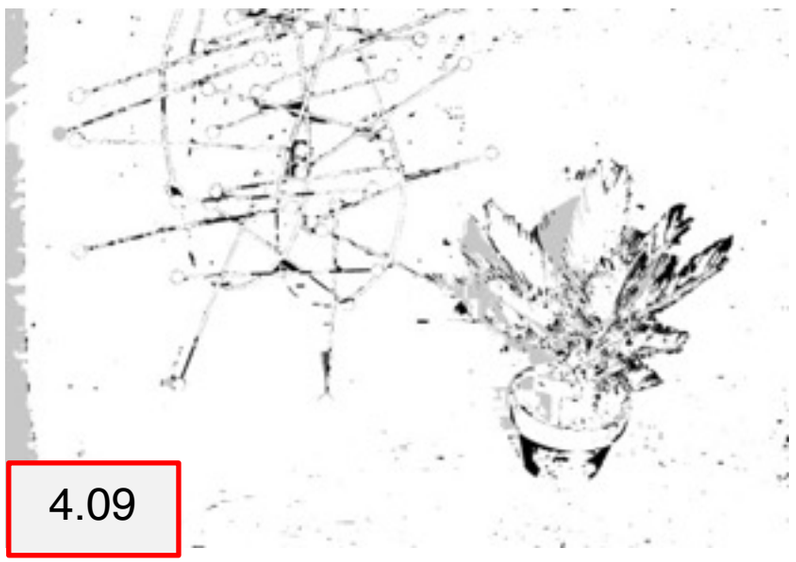






Error

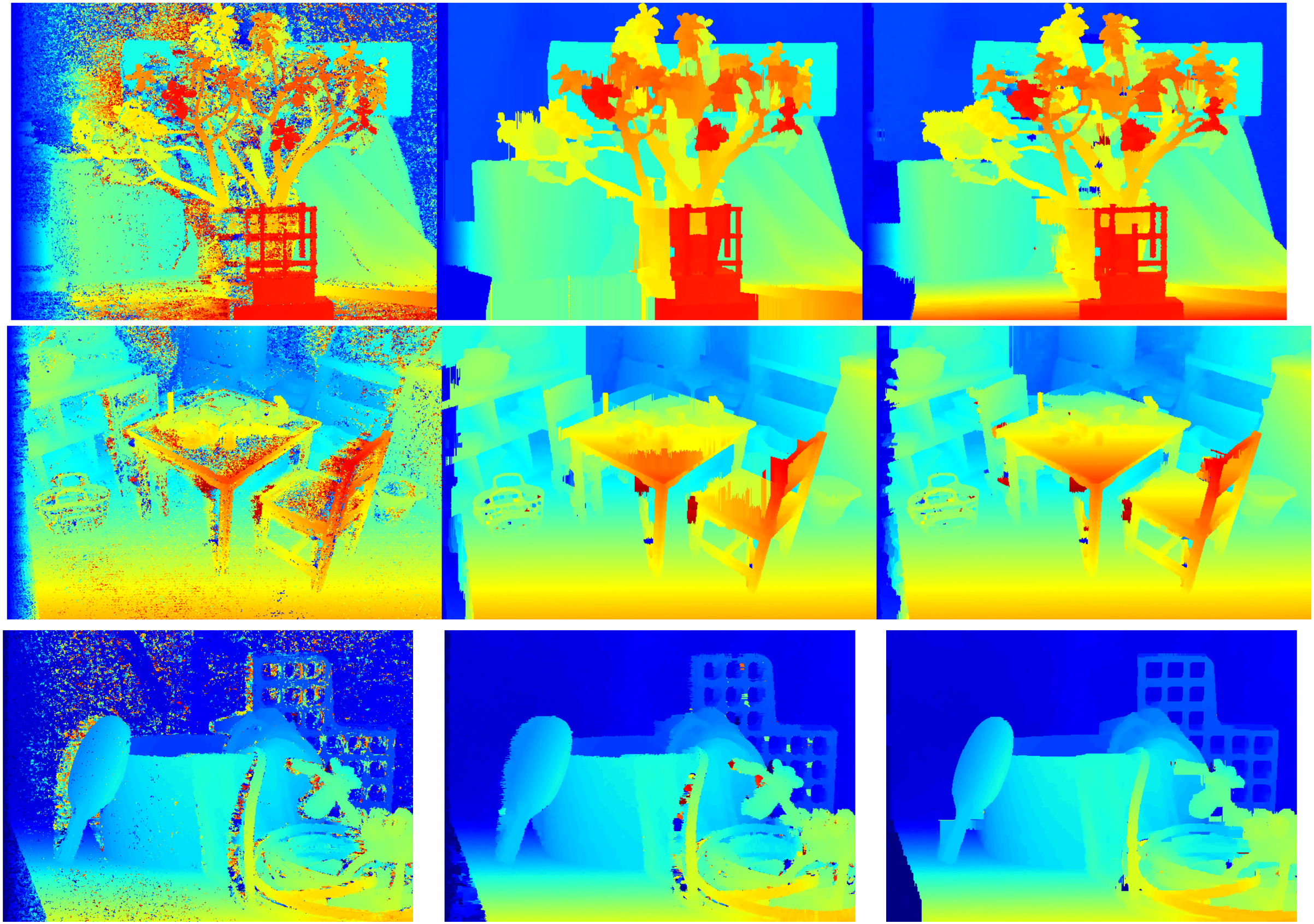
3.66



4.09

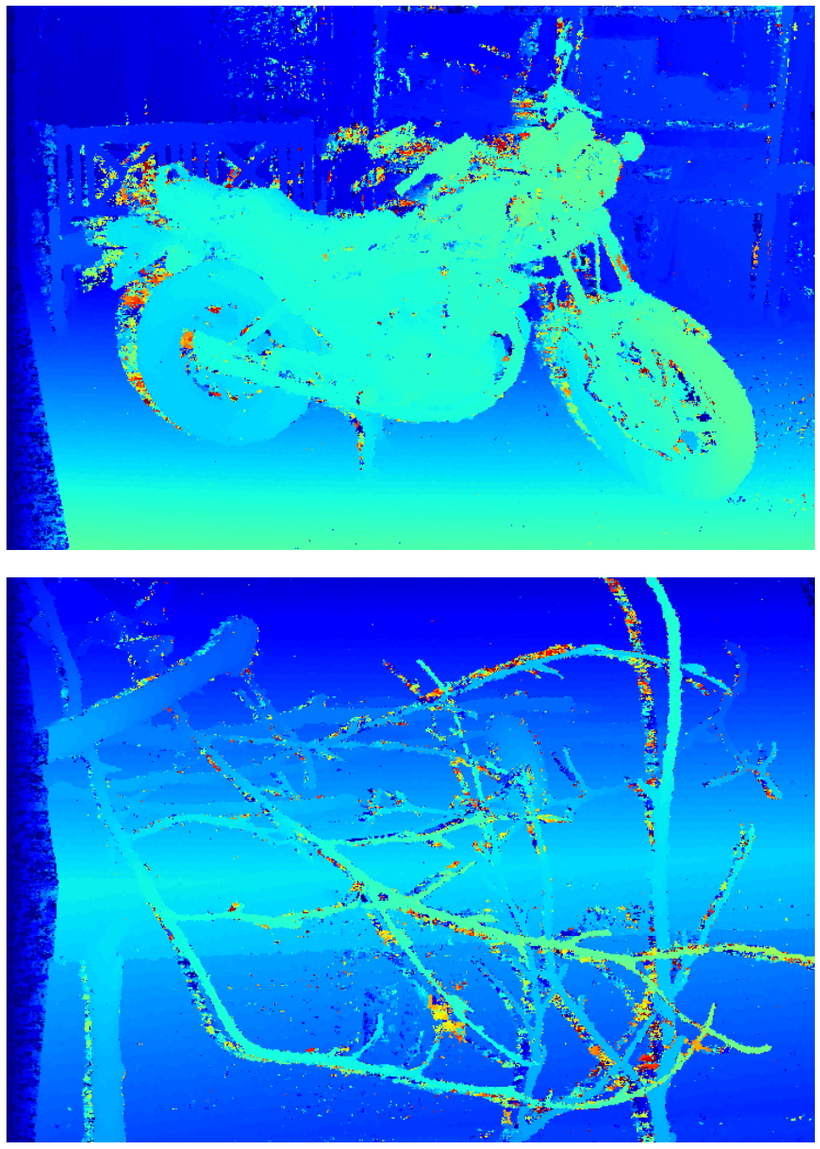
19.5

CNN+CRF

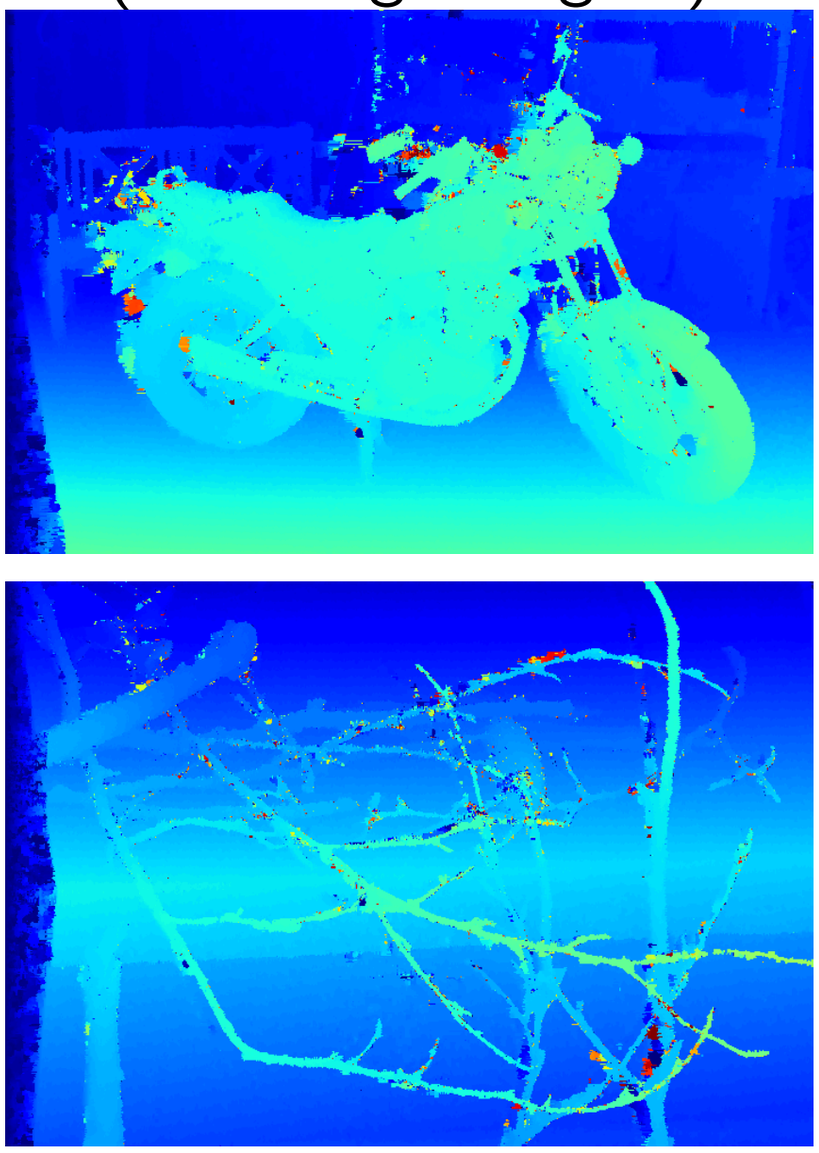


CNN+CRF

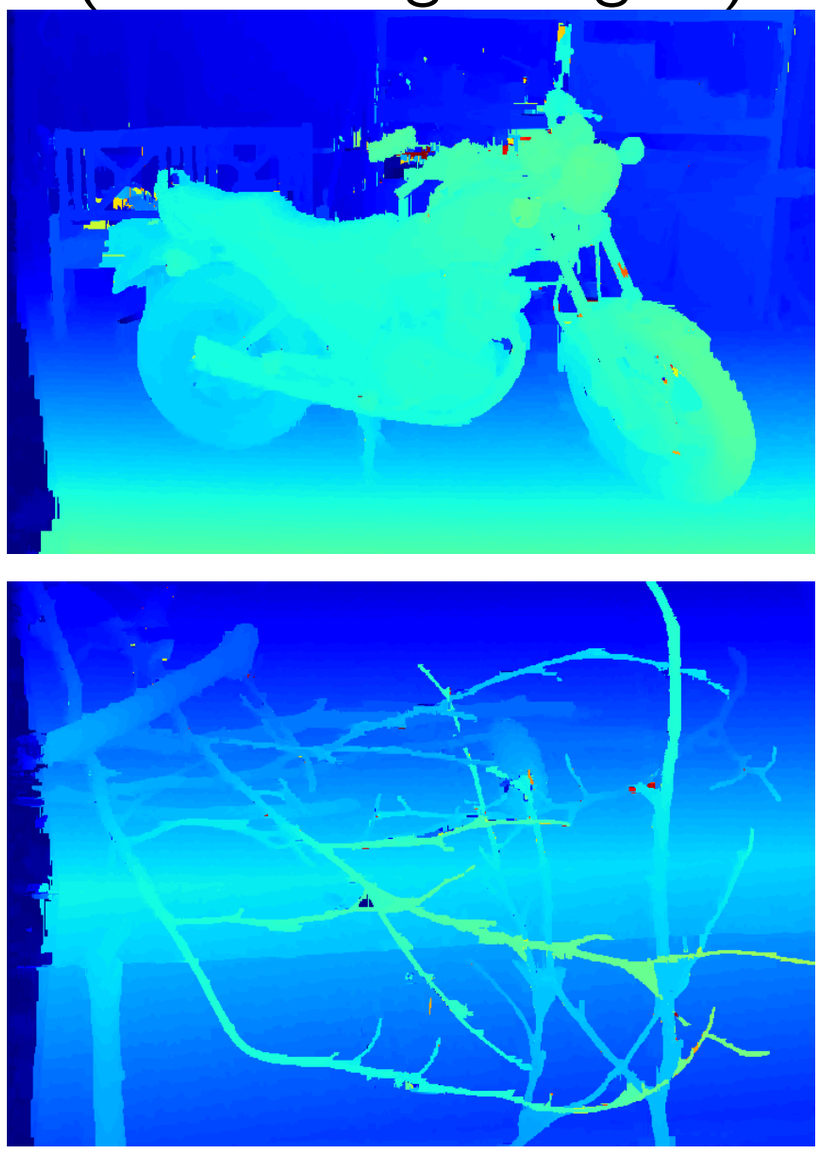
CNN-only



CNN + CRF
(fixed edge weights)



CNN + CRF
(trained edge weights)



Experiments – Kitti 2015

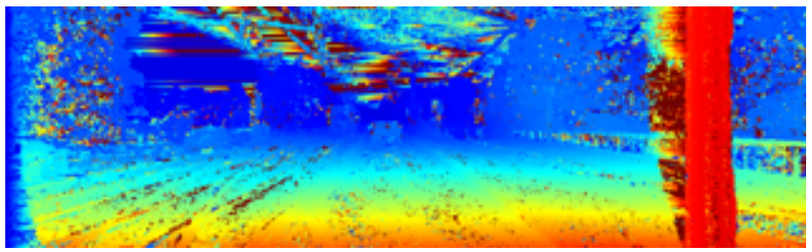
- Standard metric: *bad3*
  - Percentage of „bad“ pixels whose error is greater than 3
- Dataset specific for autonomous driving
- Currently rank 8 of published algorithms

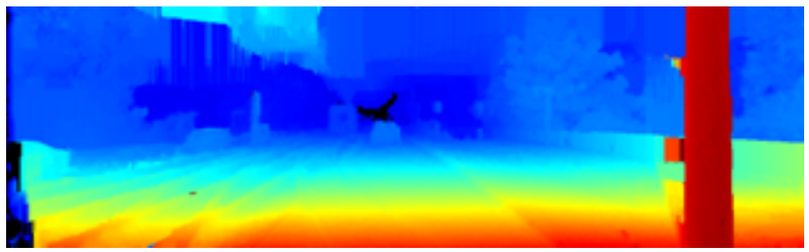
Method	Non-occ	All	#Parameters	Time	Post-Processing
MC-CNN	3.33	3.89	830k	67s	CA, SGM, SE, MF, BF
ContentCNN	4.00	4.54	700k	1s	CA, SGM, LR, SE, MF, BF, RBS
Ours	4.84	5.50	281k	1.3s	-

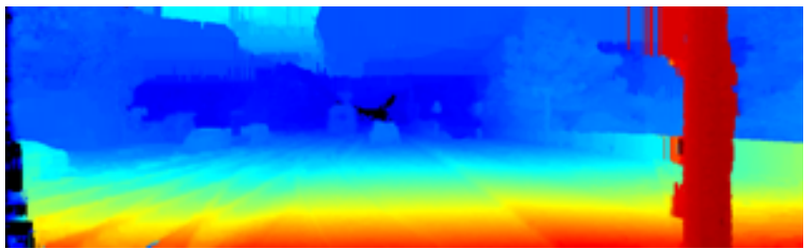
CA...Cost Aggregation, SGM...Semi-Global Matching, SE...Sublabel Enhancement, LR...Left-Right Check, MF...Median Filtering, BF...Bilateral Filtering, RBS...Robust Bilateral Solver

Experiments – Kitti 2016

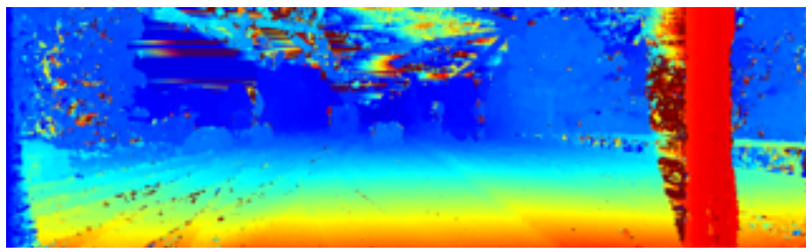
3 layer

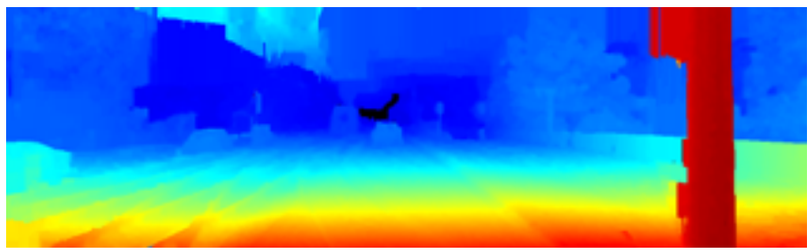


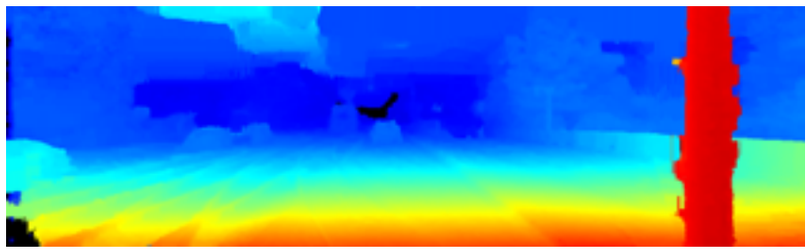




7 layer





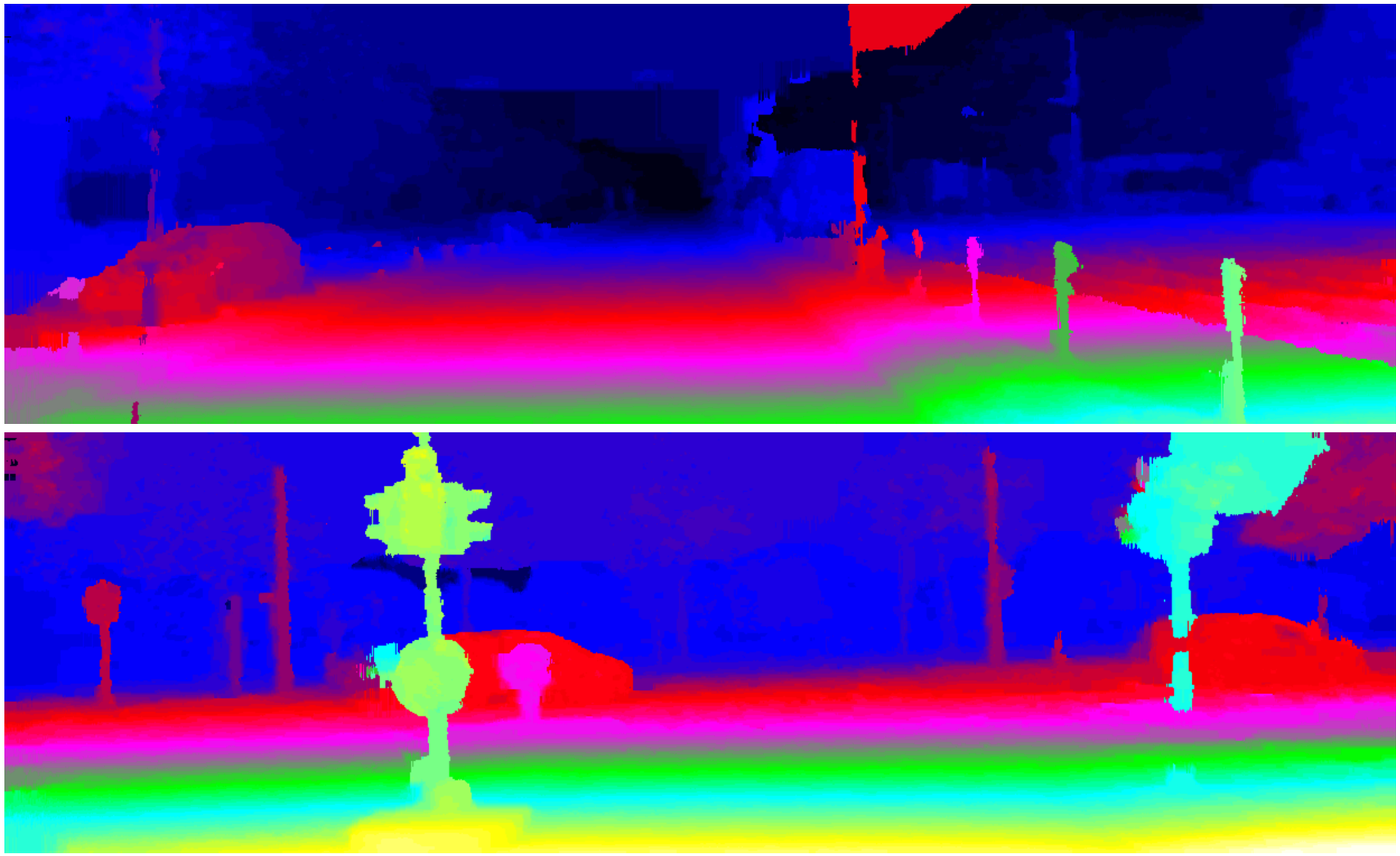


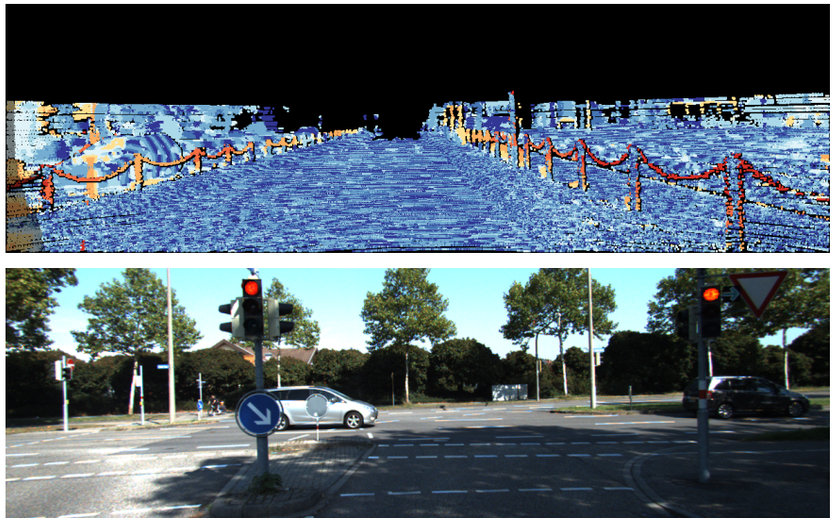
Unary CNN

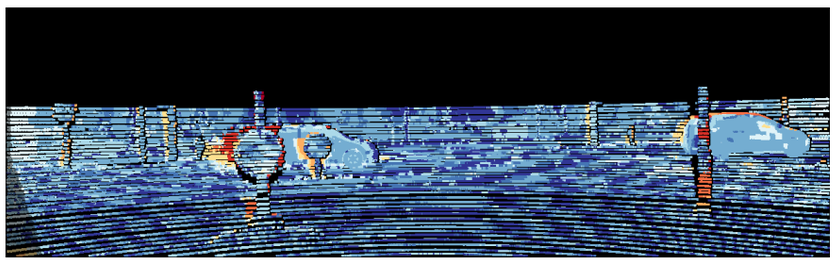
Unary CNN + CRF

Full Joint

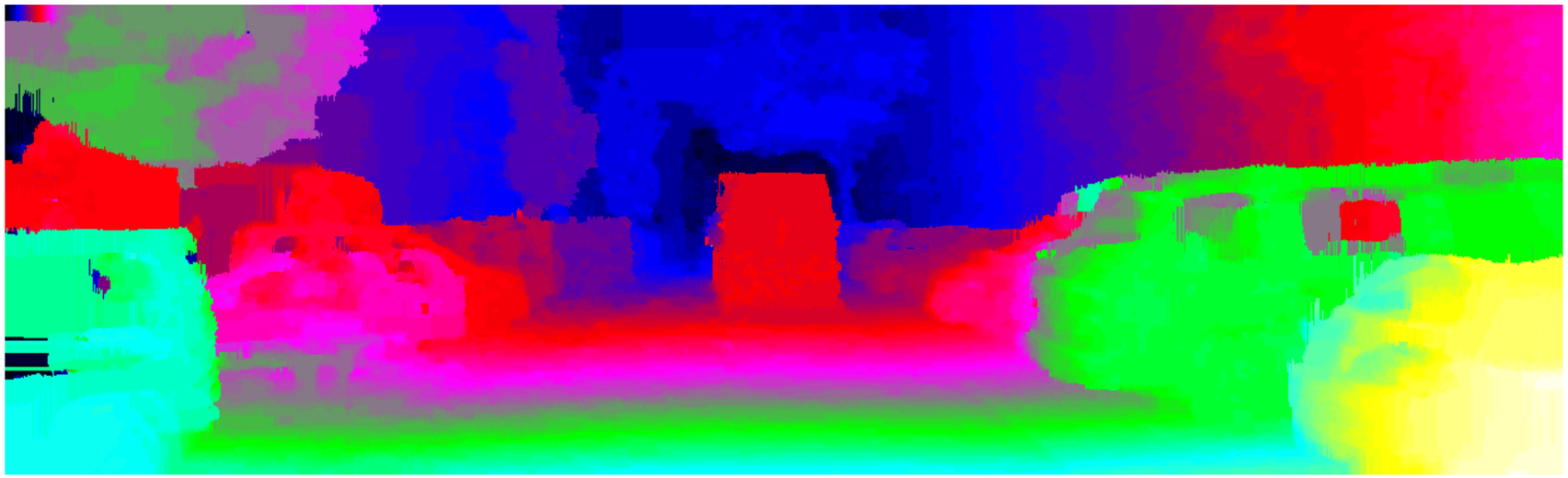
Experiments - Kitti 2016

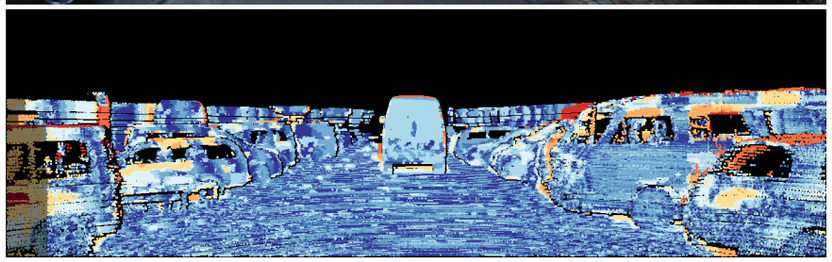


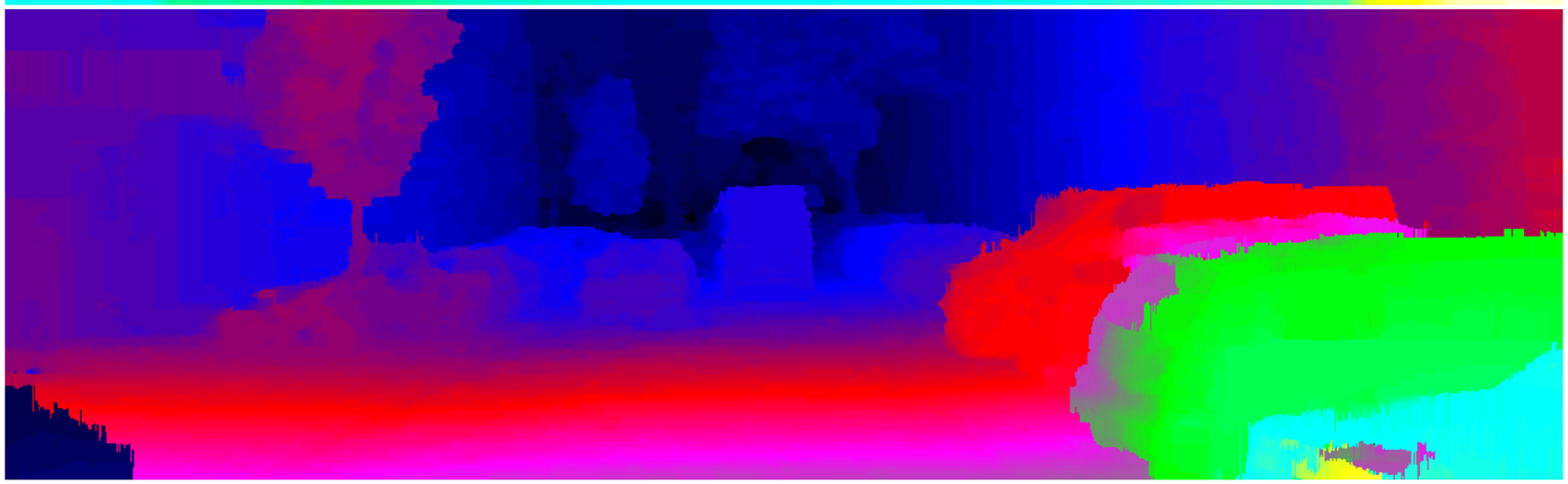


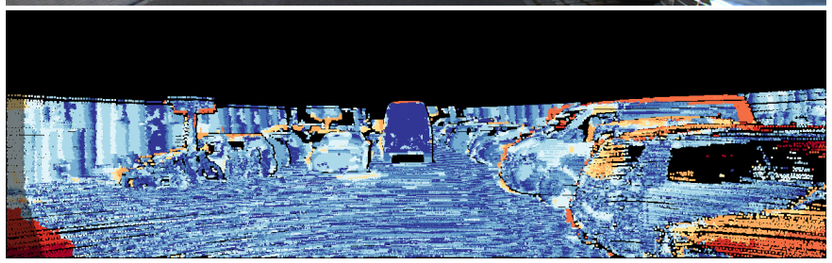


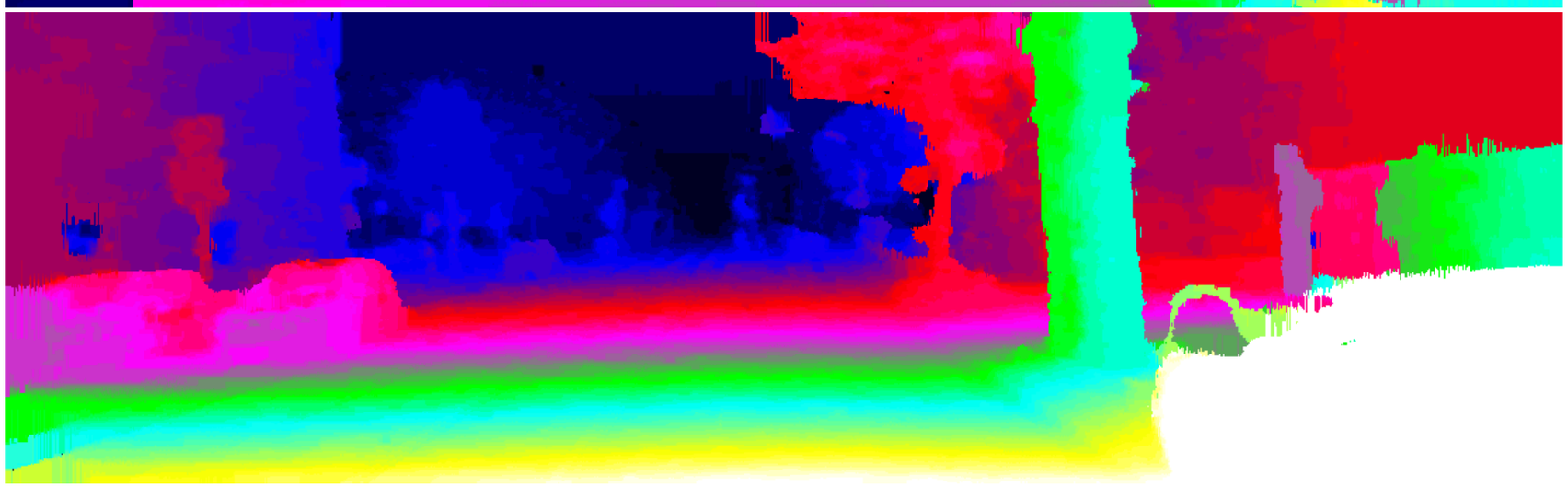
Experiments - Kitti 2016

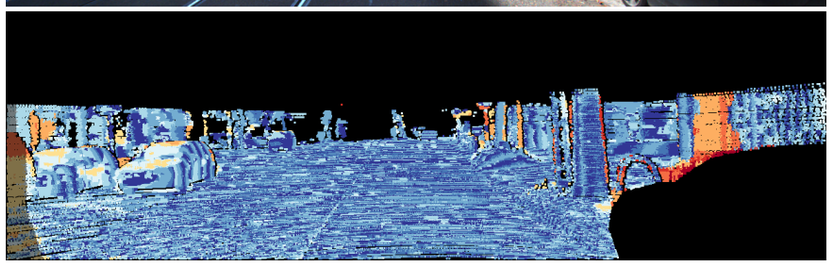






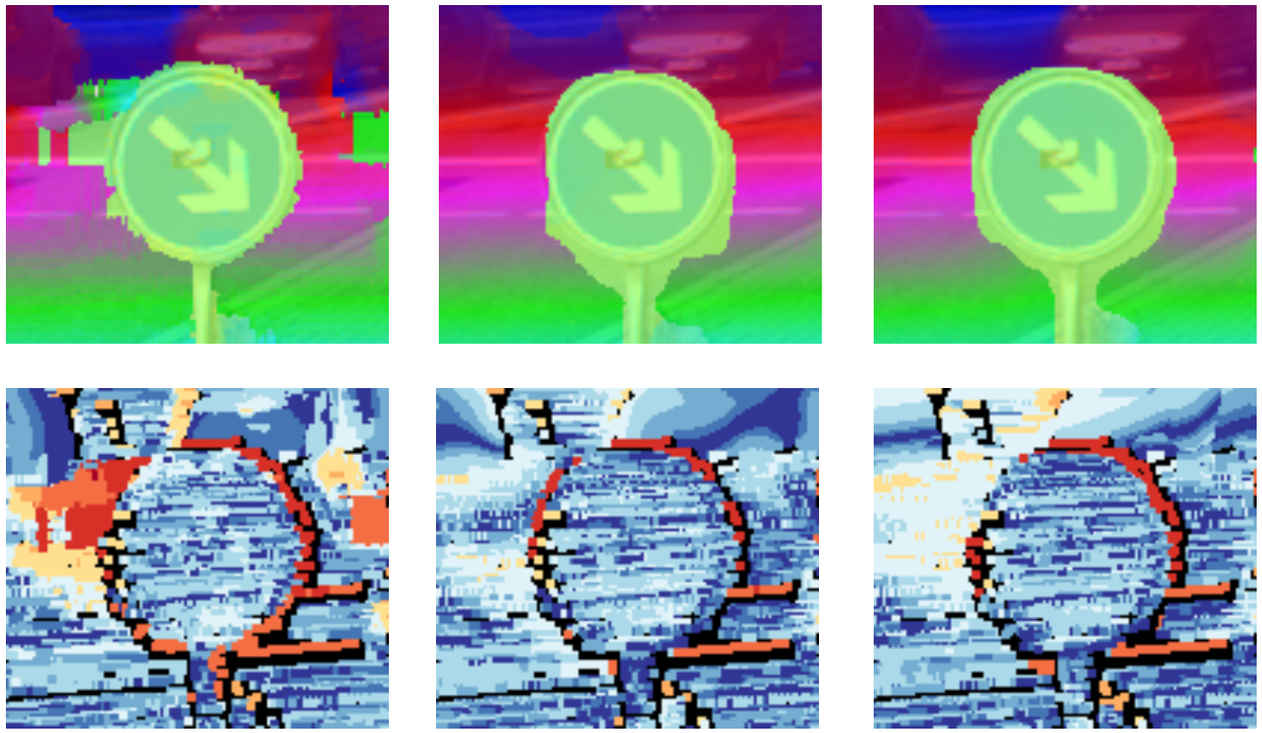




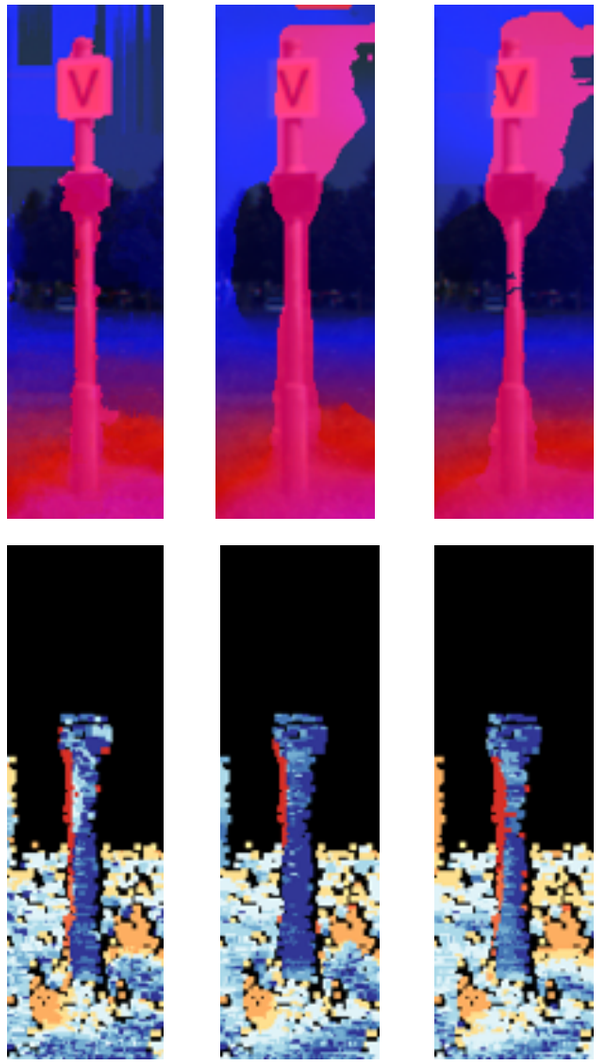


Experiments – Kitti 2015

Where do we make errors?

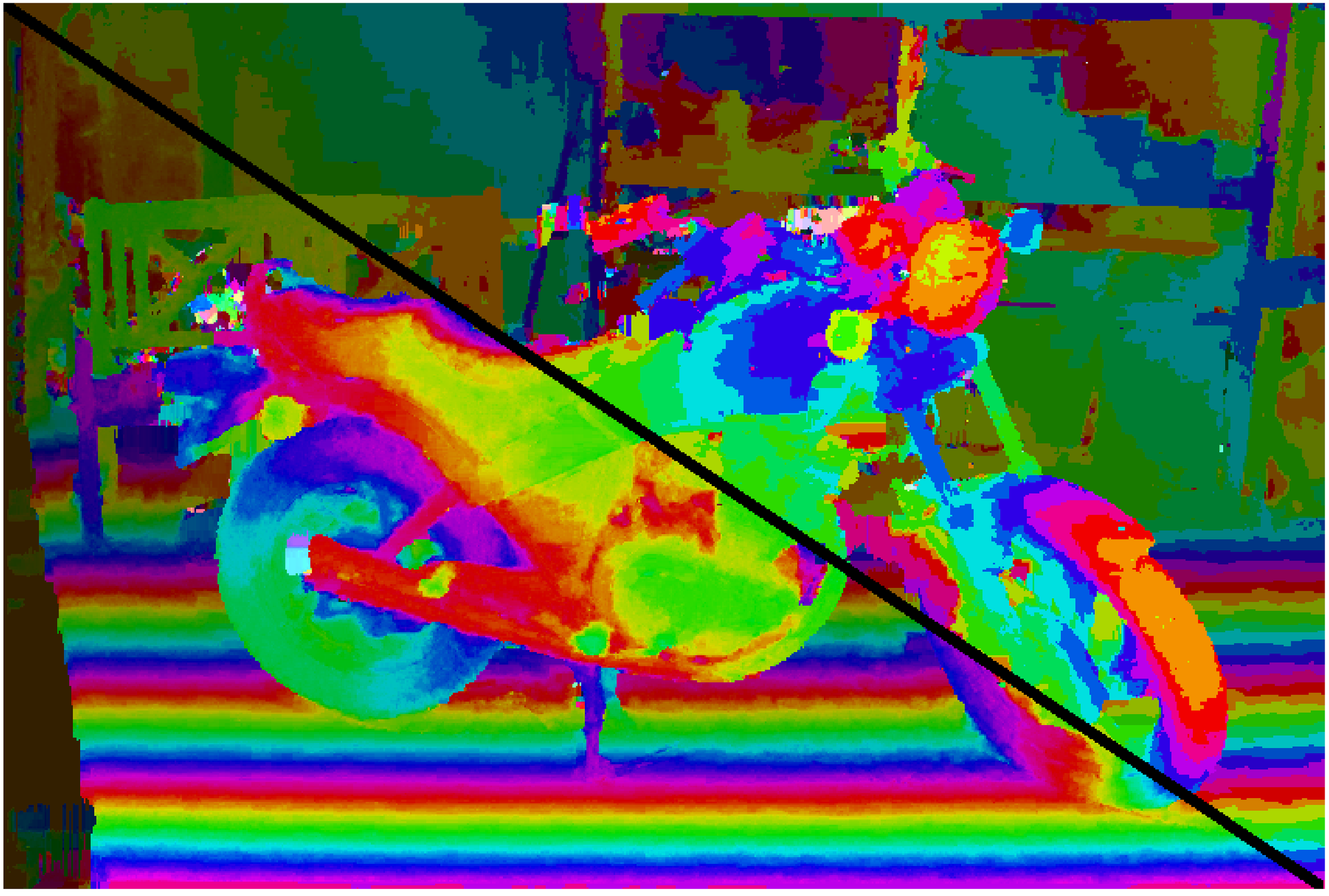


Ours MC-CNN ContentCNN



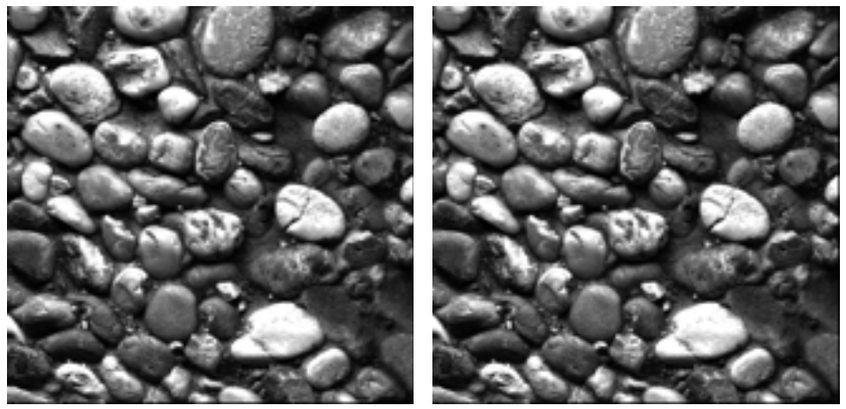
Ours MC-CNN ContentCNN

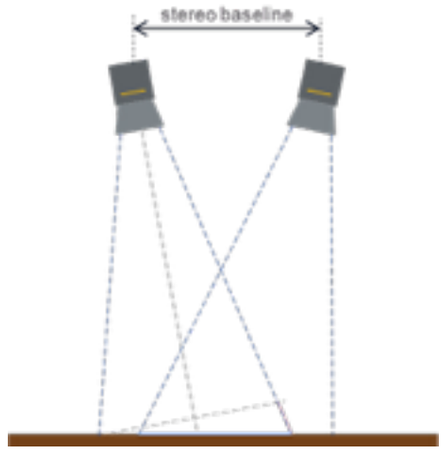
Discretization Artifacts — Sublabel Enhancement



Practical application: depth reconstruction for road surface inspection

Stereo images

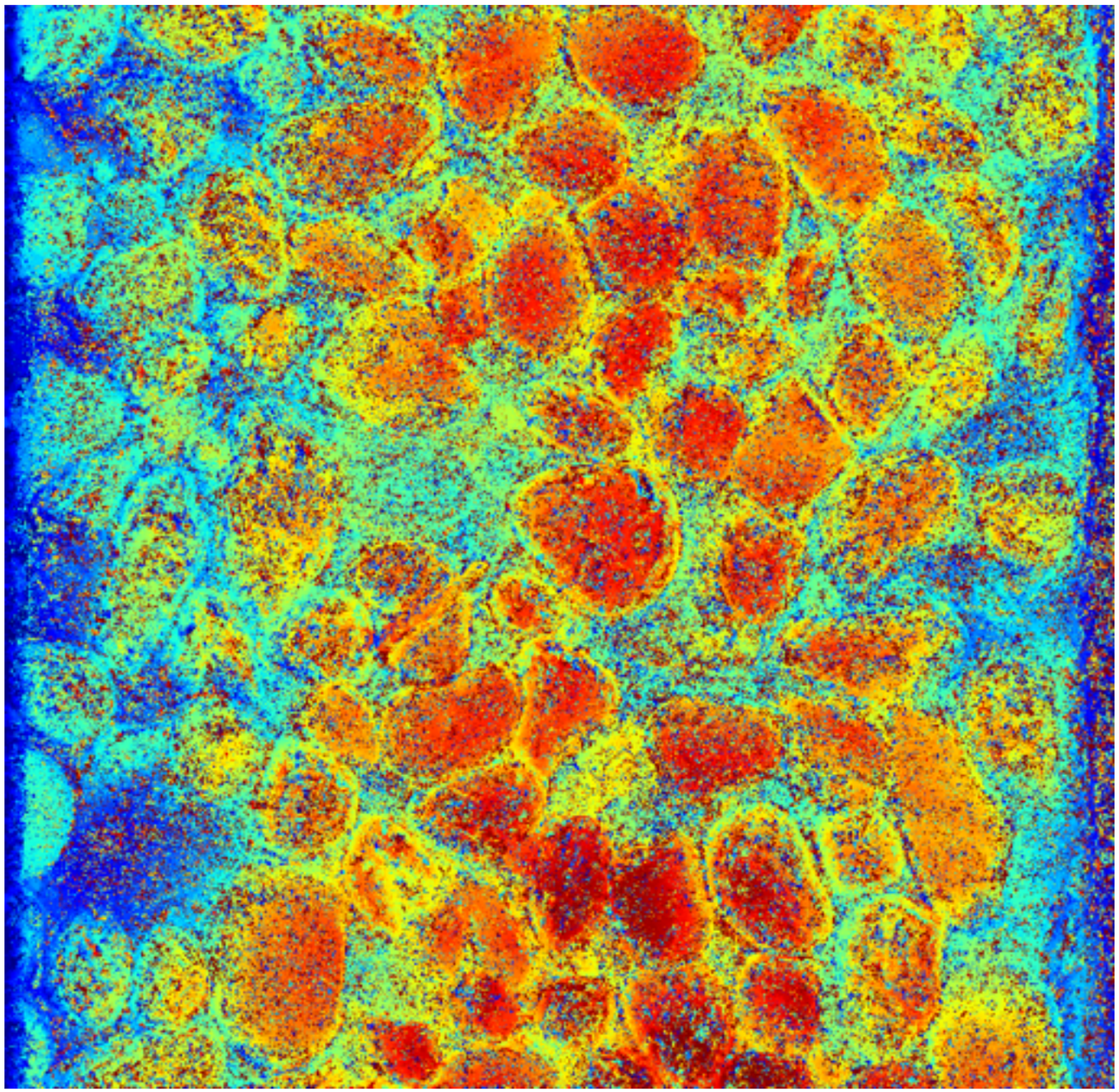




- We used the stereo model trained on the Middlebury dataset
- Imperfect rectifications can be handled by a small minimum cost search orthogonal to the epipolar lines

Comparisons

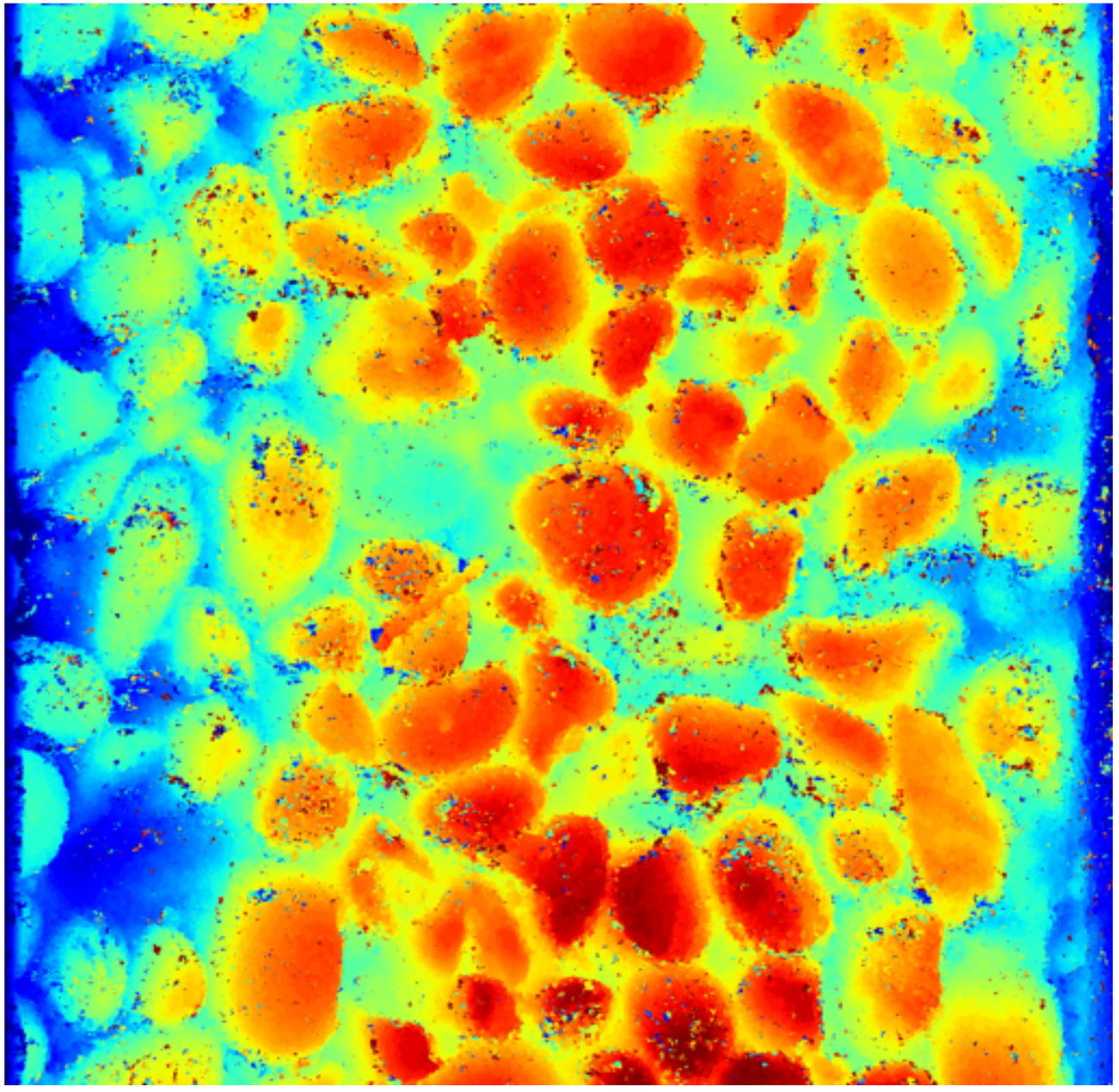
Input image



CENSUS

Comparisons

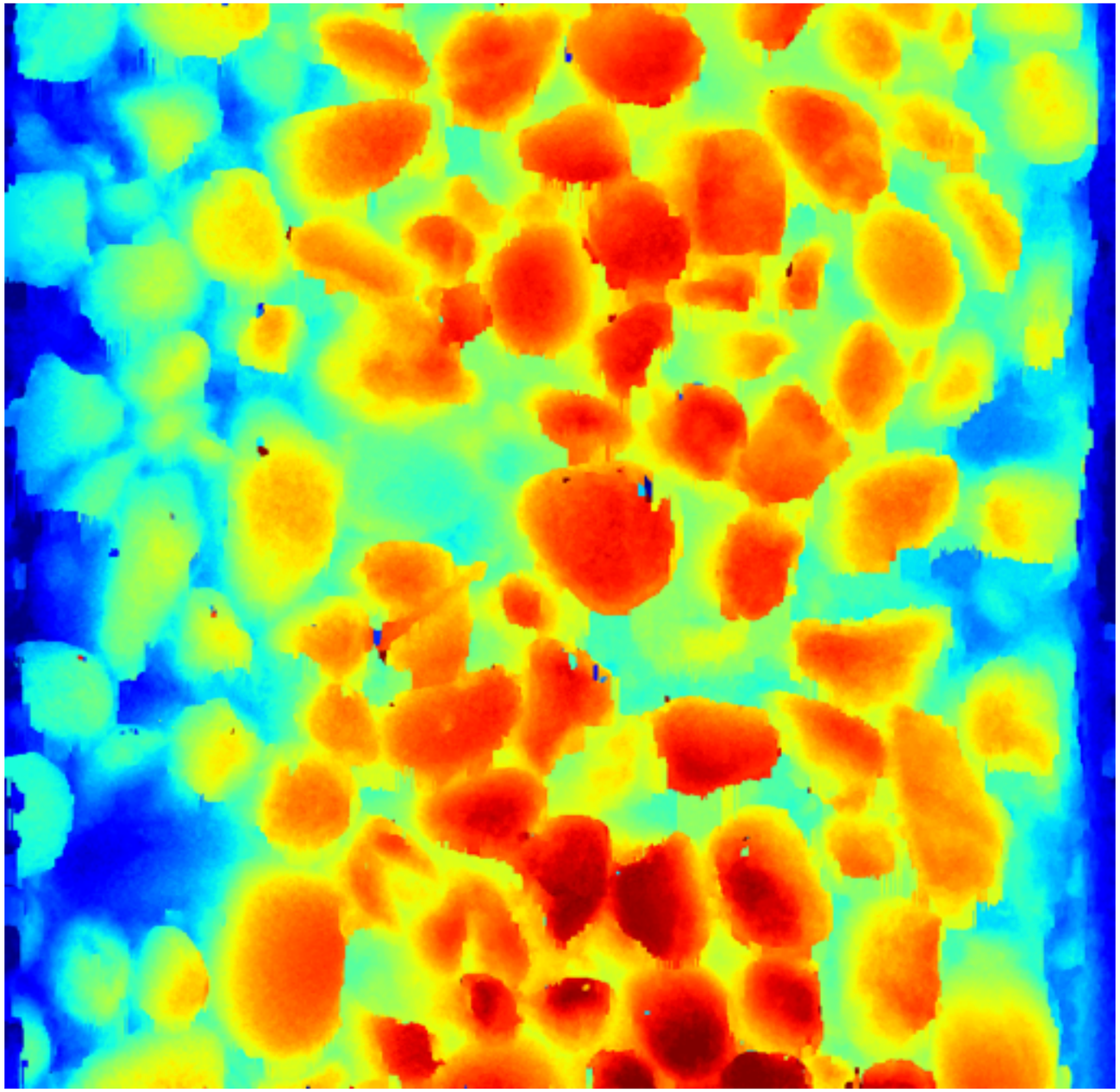
Input image



CNN

Comparisons

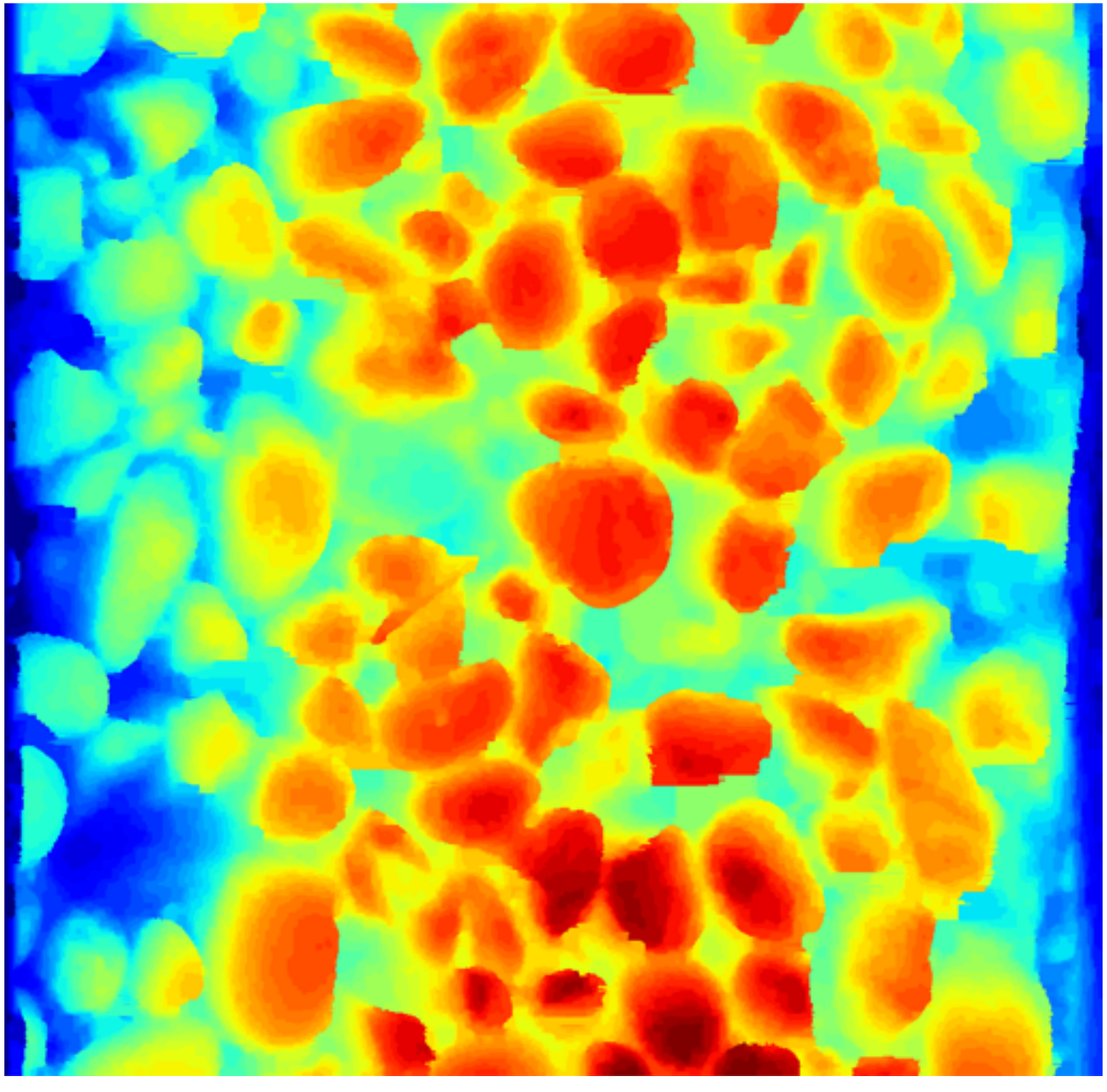
Input image



CENSUS+CRF

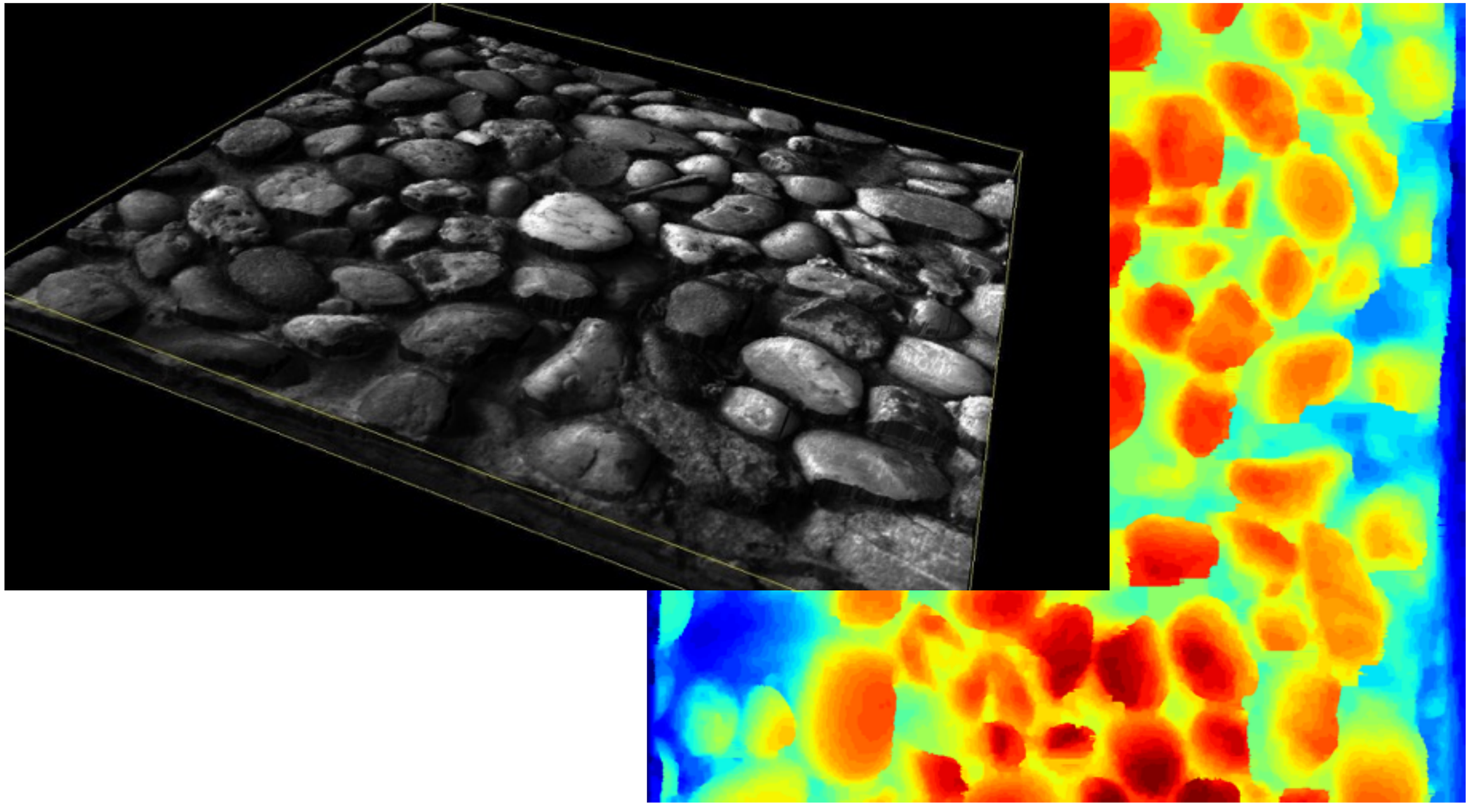
Comparisons

Input image



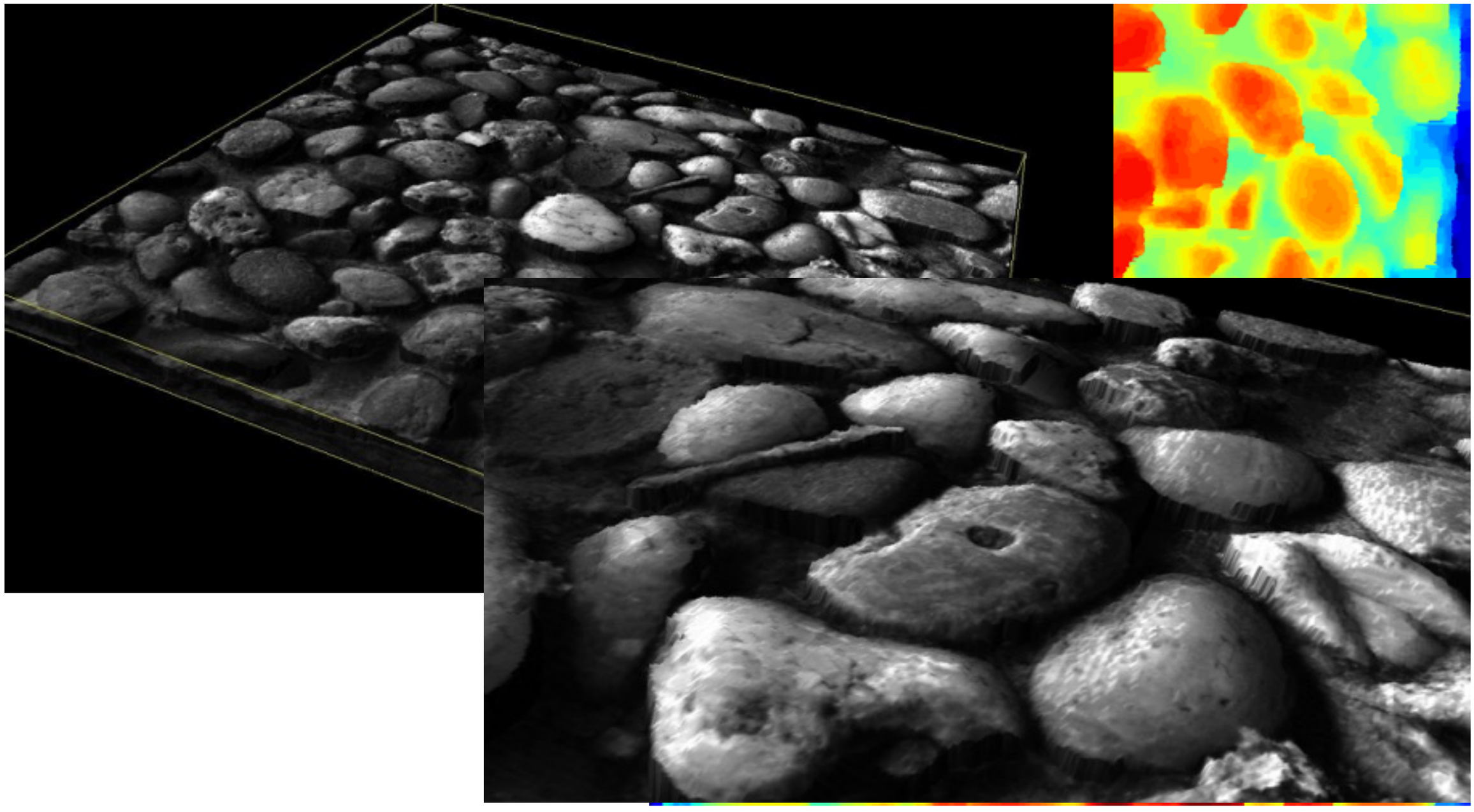
HDSM

Comparisons



HDSM

Comparisons



HDSM

Training Unary CNN: Related Loss Functions

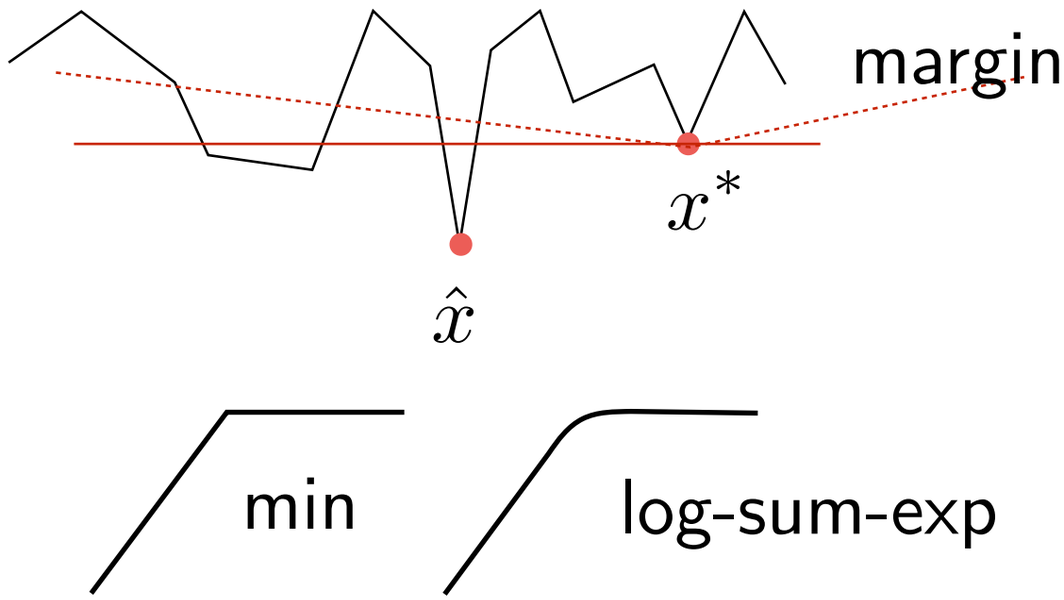
Predictor: $\hat{x} \in \arg\min_x d(x;\theta)$

SVM Margin Rescaling

Hinge loss: $d(x^*) - \min_x d(x)$

SVM with smooth min

$d(x^*) + \log \sum_x \exp(-d(x))$



Maximum Likelihood

softmax

Probabilistic model: $p(x) = \exp(-d(x)) / \sum_x \exp(-d(x))$

$-\log p(x^*)$ approaches hinge depending on scale

Cross-Entropy

$-\sum_x p^*(x) \log p(x)$

Forward KL divergence

$KL(p^*||p) = \sum_x p^*(x) \log \frac{p^*(x)}{p(x)}$

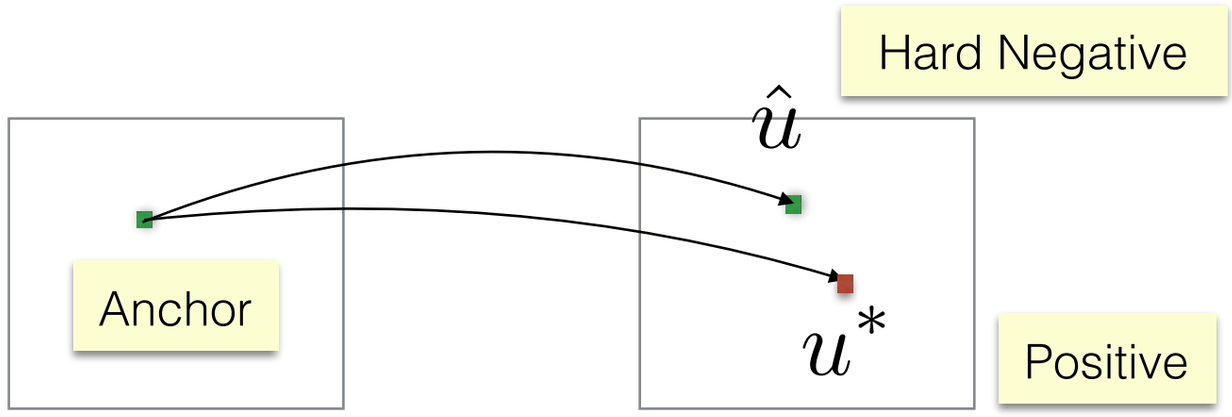
Training Unary CNN: Related Loss Functions

Predictor: $\hat{x} \in \arg\min_x d(x;\theta)$

SVM Margin Rescaling

Hinge loss: $d(x^*) - \min_x d(x)$

Triplet Loss



2 candidate matches = triplet

Want: $d(\text{anchor}, \text{positive}) \leq d(\text{anchor}, \text{negative})$

$\min_x d(\phi_i^1, \phi_{i+x}^2)$ - distance to hardest negative

- Many related problems:
  - Image-based search, image retrieval
  - Matching patches for localization
  - etc.