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

Patrick Knöbelreiter, Christian Reinbacher, Alexander Shekhovtsov and Thomas Pock

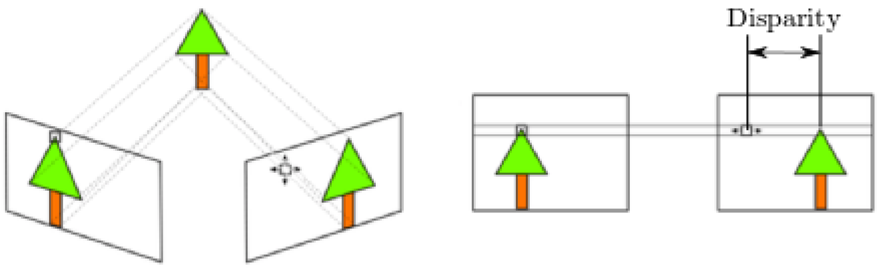
Institute for Computer Graphics and Vision
Graz University of Technology

Digital Safety & Security Department
AIT Austrian Institute of Technology

Stereo Problem

Input

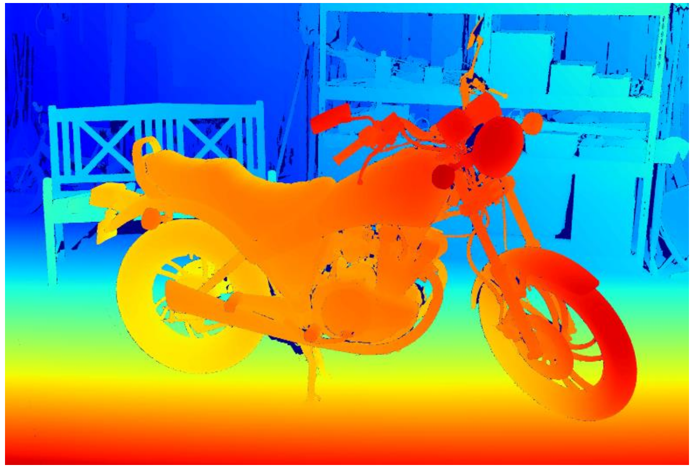
- Two images from a calibrated camera pair
- Rectified: epipolar lines correspond to image rows



Input Pair

Problem

For each pixel in the left image find the corresponding pixel in the right image

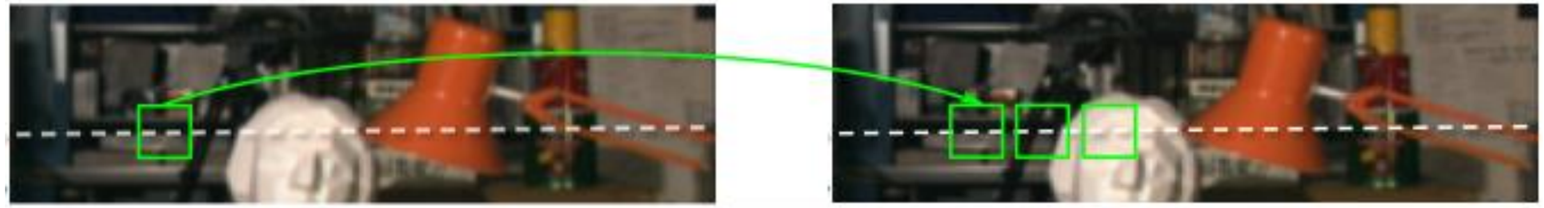


Disparity Map (GT)

Output

Dense depth (disparity) map

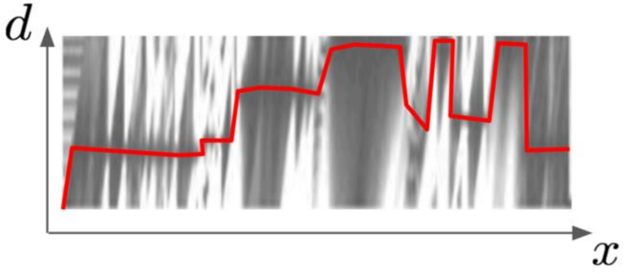
Local and Optimization-based Approaches



Local Matching Cost

E.g. SSD/SAD, cross-correlation, adaptive weights, guided filter, sampling-insensitive, Census transform, Correlation of CNN features

Cost Volume: $f_i(x_i)$

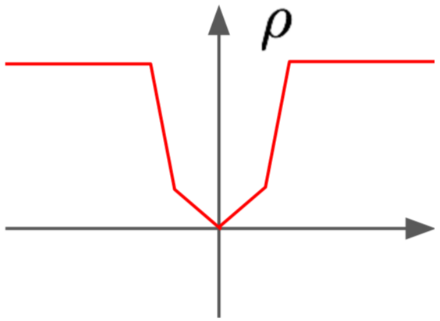


Smoothness Prior / Regularization

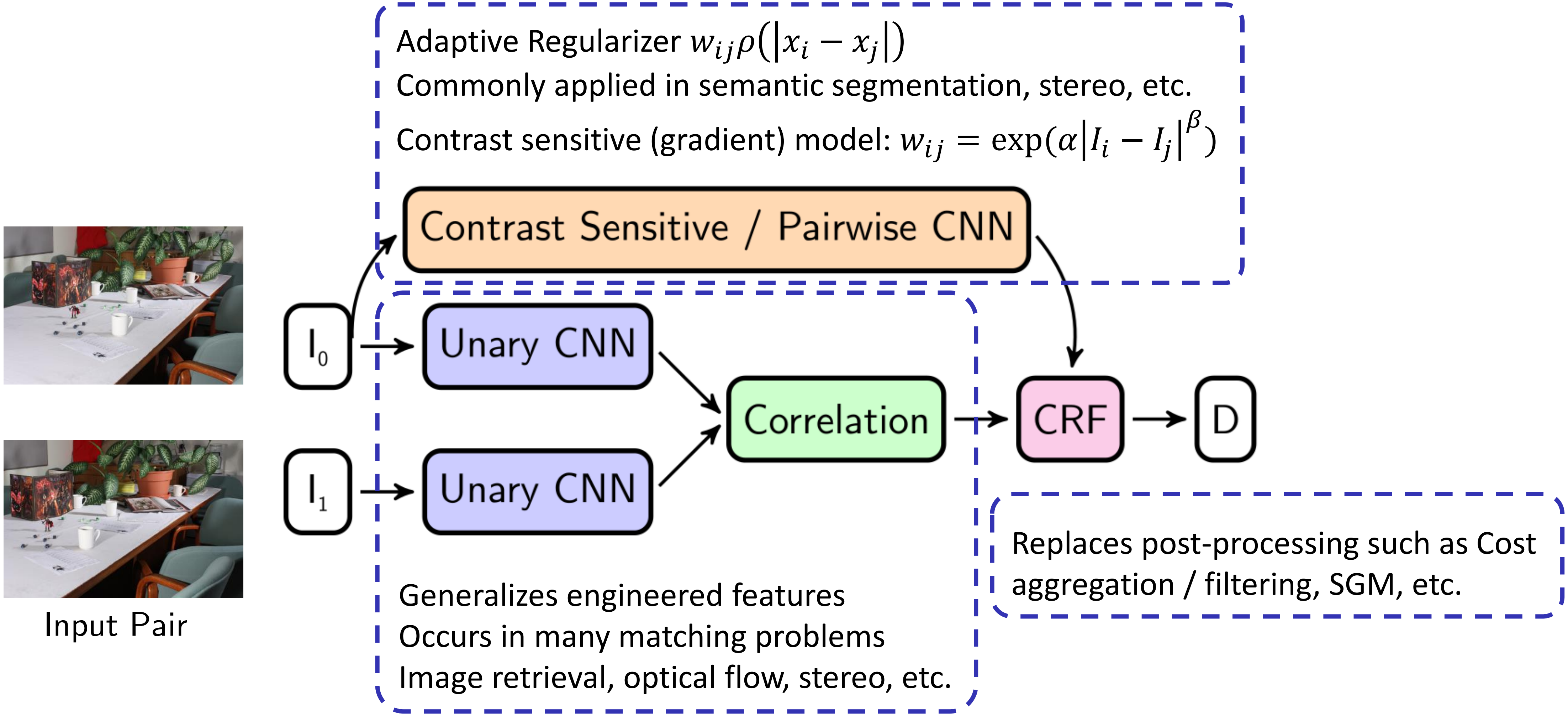
Optimizing local matching cost with regularizer

- Continuous: TV, TGV, ...
- Discrete: Graph cut, CRF

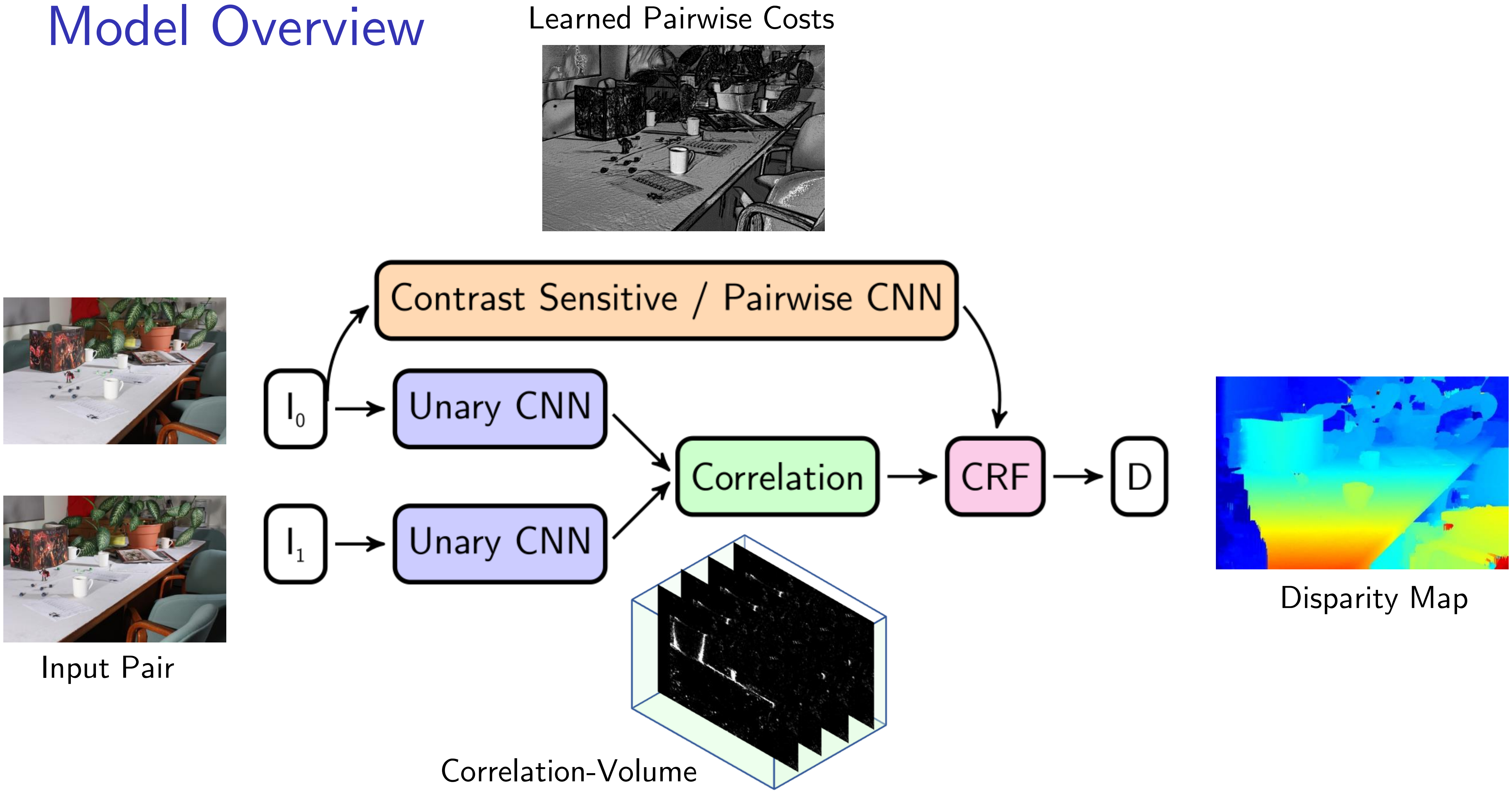
Regularizer: $+\rho(|x_i - x_j|)$



Model Overview



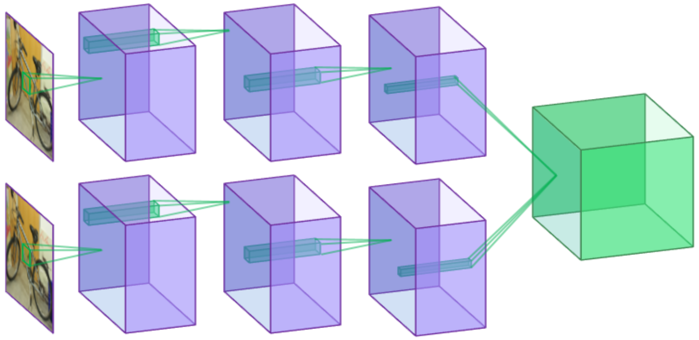
Model Overview



The building blocks: Unary CNN & Correlation

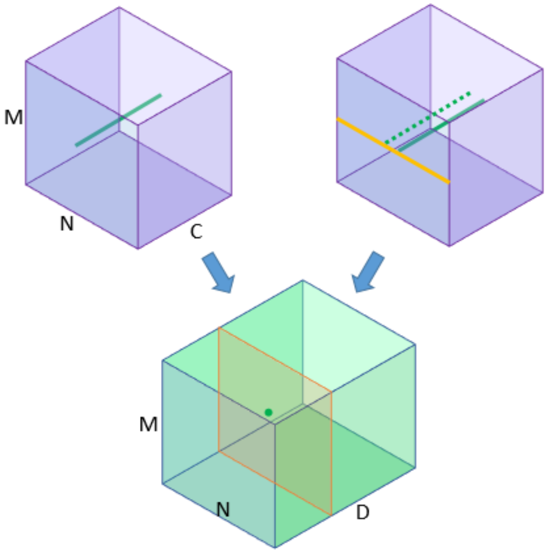
Unary CNN

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



Correlation

- Compute the correlation across the learned features for all disparities
- Each disparity creates one slice in the correlation volume

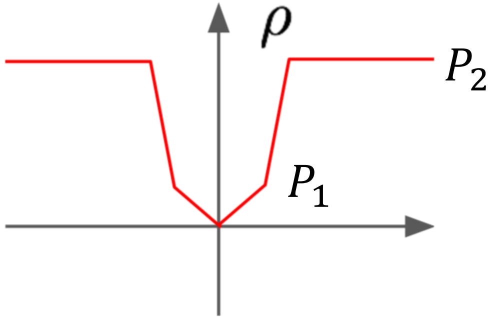


The building blocks: CRF

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

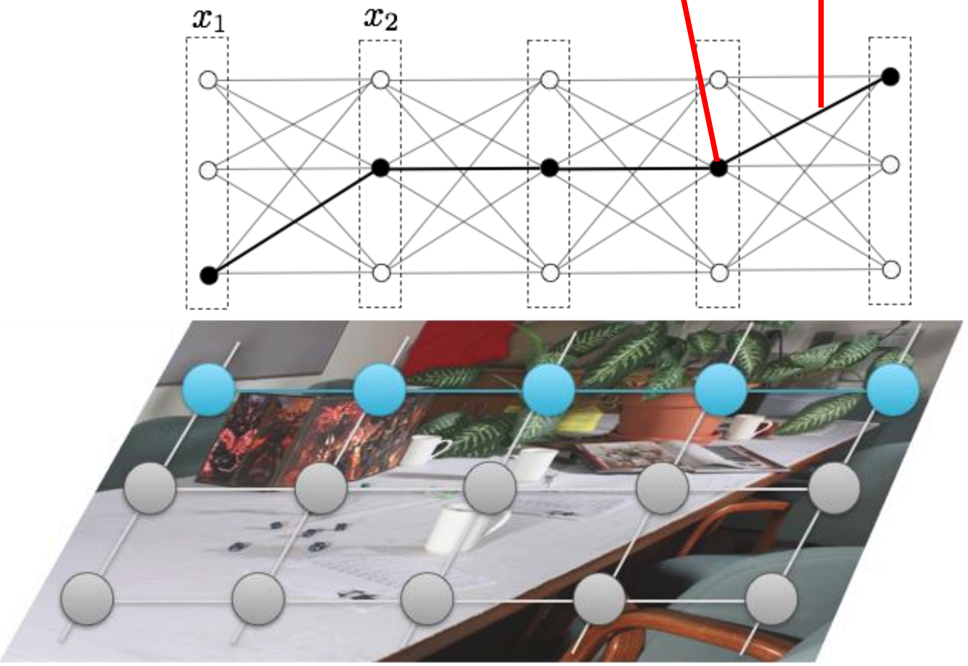
$$\min_{x \in V^L} f(x) \coloneqq \sum_{i \in V} f_i(x_i) + \sum_{ij \in E} f_{ij}(x_i, x_j)$$

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



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

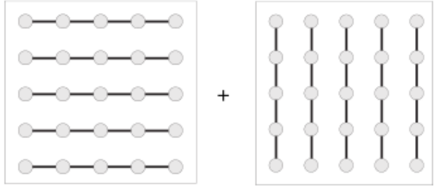
Unary data cost $f_i(x_i)$



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

Inference in CRF – Dual Majorize-Maximize

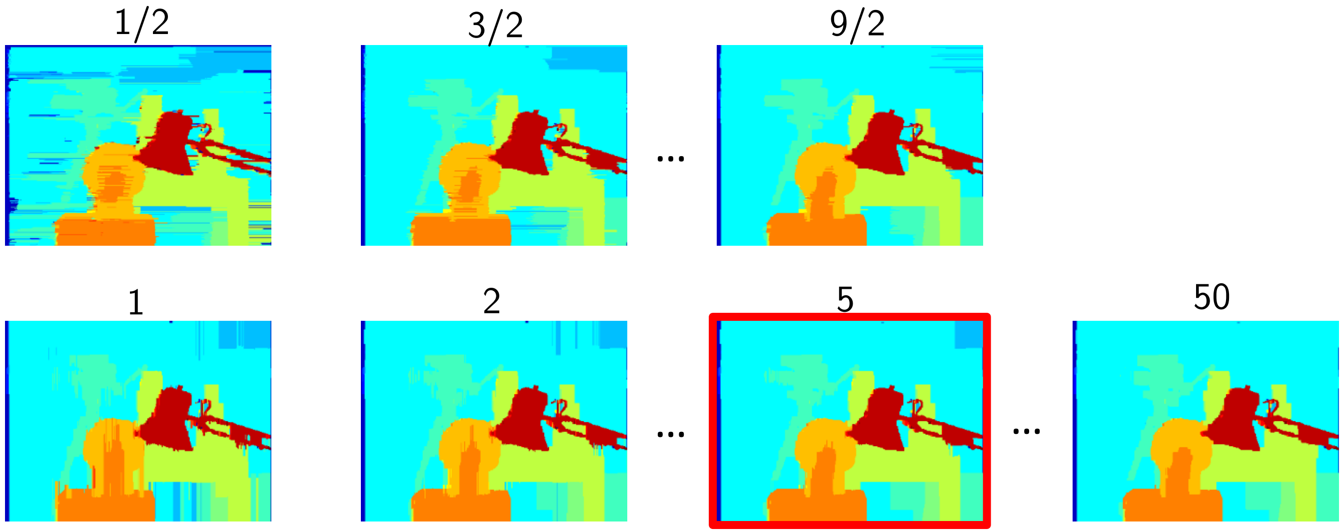
- Sum of chain sub-problems: $f = f^1 + f^2$

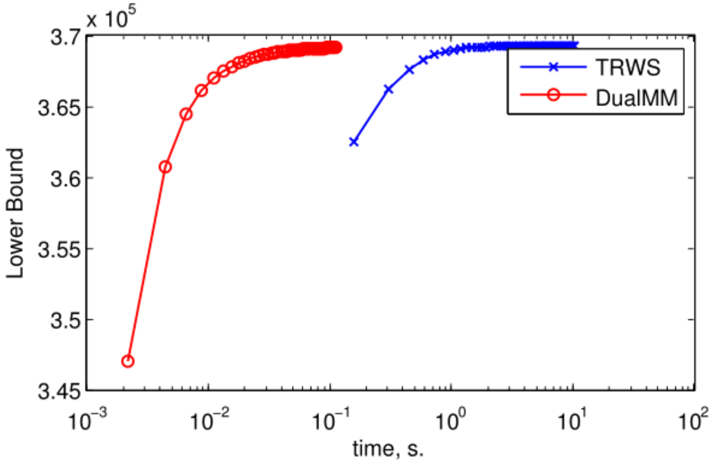


- Lagrange decomposition

$$\max_{\varphi}[\min_{x}(f^1 + \varphi)(x) + \min_{x}(f^2 - \varphi)(x)] \qquad \text{(LP Relaxation Dual)}$$

- Lagrange multiplier φ ensures consistent solutions of sub-problems

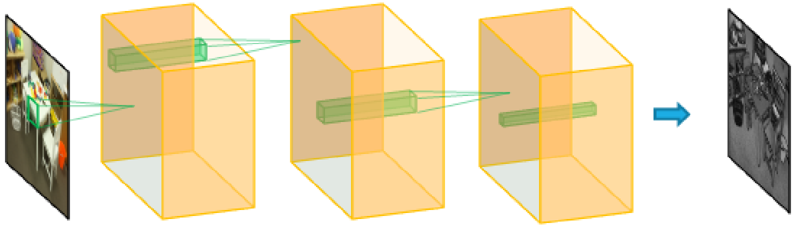


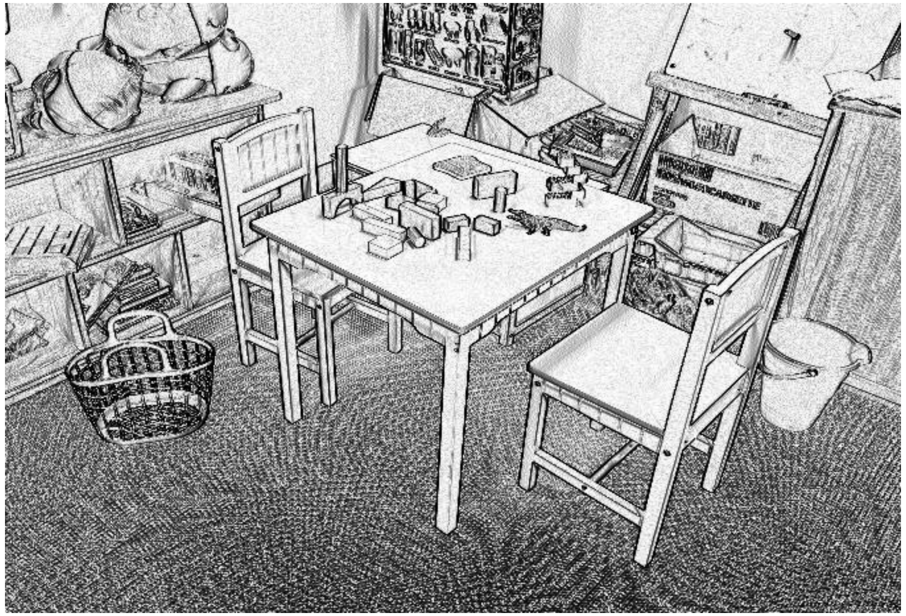


LB vs time (DMM on GPU)

The building blocks: Pairwise CNN

- Pairwise CNN
- 3 layers:
  - 2 layers extract features
  - Last layer maps to weights
- 38k parameters
- Learn image-dependent weighting costs
- Courage label jumps at strong object boundaries
- Discourage label jumps in homogenous regions

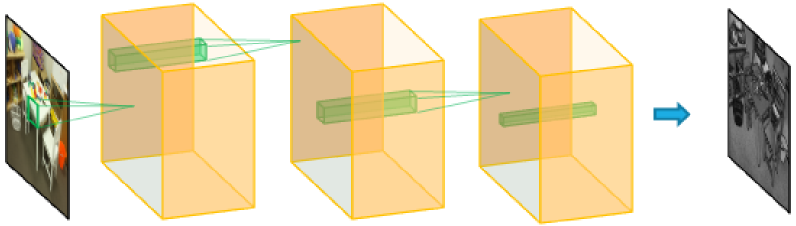




Fixed Edges

The building blocks: Pairwise CNN

- Pairwise CNN
- 3 layers:
  - 2 layers extract features
  - Last layer maps to weights
- 38k parameters
- Learn image-dependent weighting costs
- Courage label jumps at strong object boundaries
- Discourage label jumps in homogenous regions



Learned Edges

Key Challenge

Algorithm for training everything jointly, i.e., „end-to-end“

How can we learn all parameters End-to-End?

Bi-Level Optimization Problem

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

$$s.t. \;\; x \in \arg\min_{x \in X} f(x; \theta)$$

„Learn parameters θ of CNNs, such that the minimizer of the CRF model minimizes a certain loss function“

Challenge

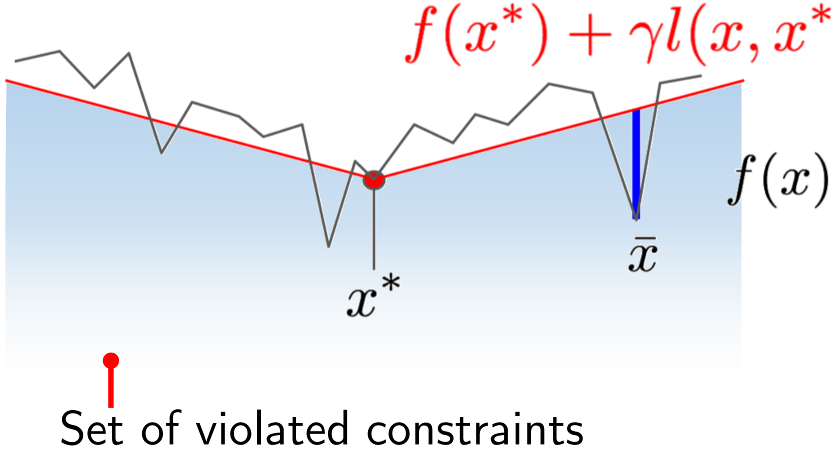
Directly back-propagating the error of the loss function to the model parameters does not work

Structured SVM [Taskar, Tsochantaridis]

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

$$(\exists\theta)\ (\forall x \in V^L)\ \ f(x^*;\theta) \leq f(x;\theta) - \gamma l(x, x^*)$$

Not always feasible!



Minimize the most violated constraint

$$\min_{\theta} \underbrace{\max_{x} \left(f(x^*;\theta) - f(x;\theta) + \gamma l(x, x)\right)}_{\text{Upper bound on the original loss}}$$

A subgradient is given by $\delta(x^*) - \delta(\bar{x})$

$\bar{x} \in \arg\min_{x}\left(f(x;\theta) - \gamma l(x, x^*)\right)$ "Loss-augmented inference problem"

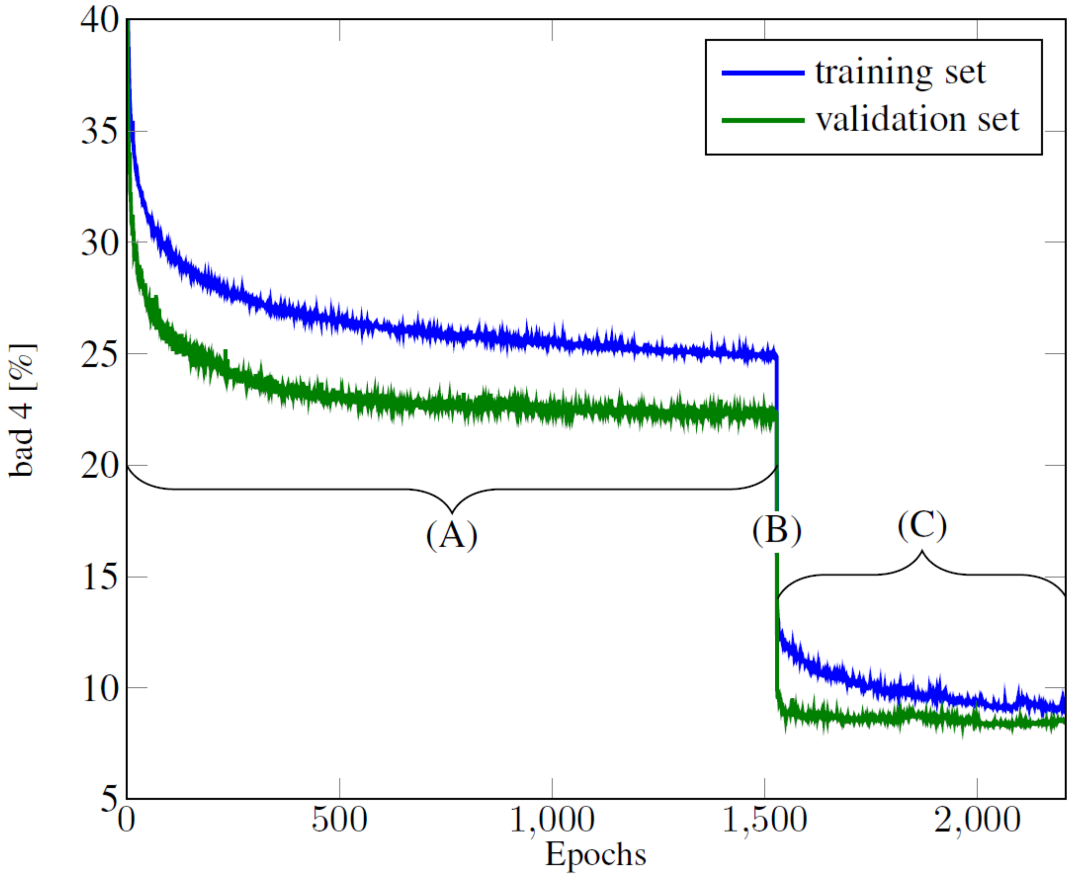
Training

Training

- Training is performed using stochastic subgradient descent with momentum
- First, we perform a Unary-CNN pre-training, followed by a joint training

Databases

- Middlebury Stereo v3
- Kitti 2015



Training-Curves

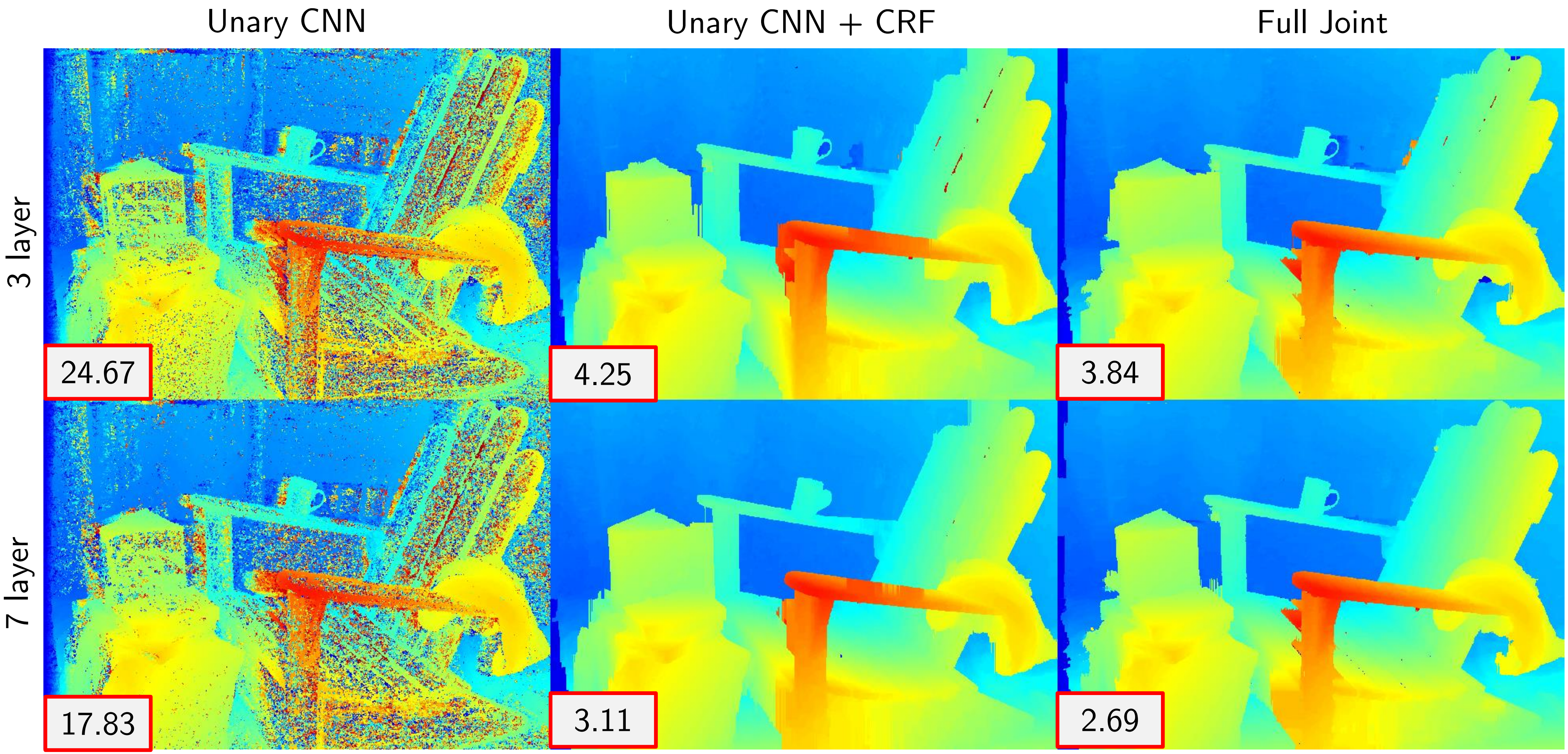
Middlebury Stereo v3

Comparing our models

- Disparity error on quarter-size images in %
- Deeper Unary CNN reduces the error
- Pairwise interactions decreases the error
- Joint training decreases the error

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

Experiments – Middlebury Stereo v3



Middlebury Stereo v3

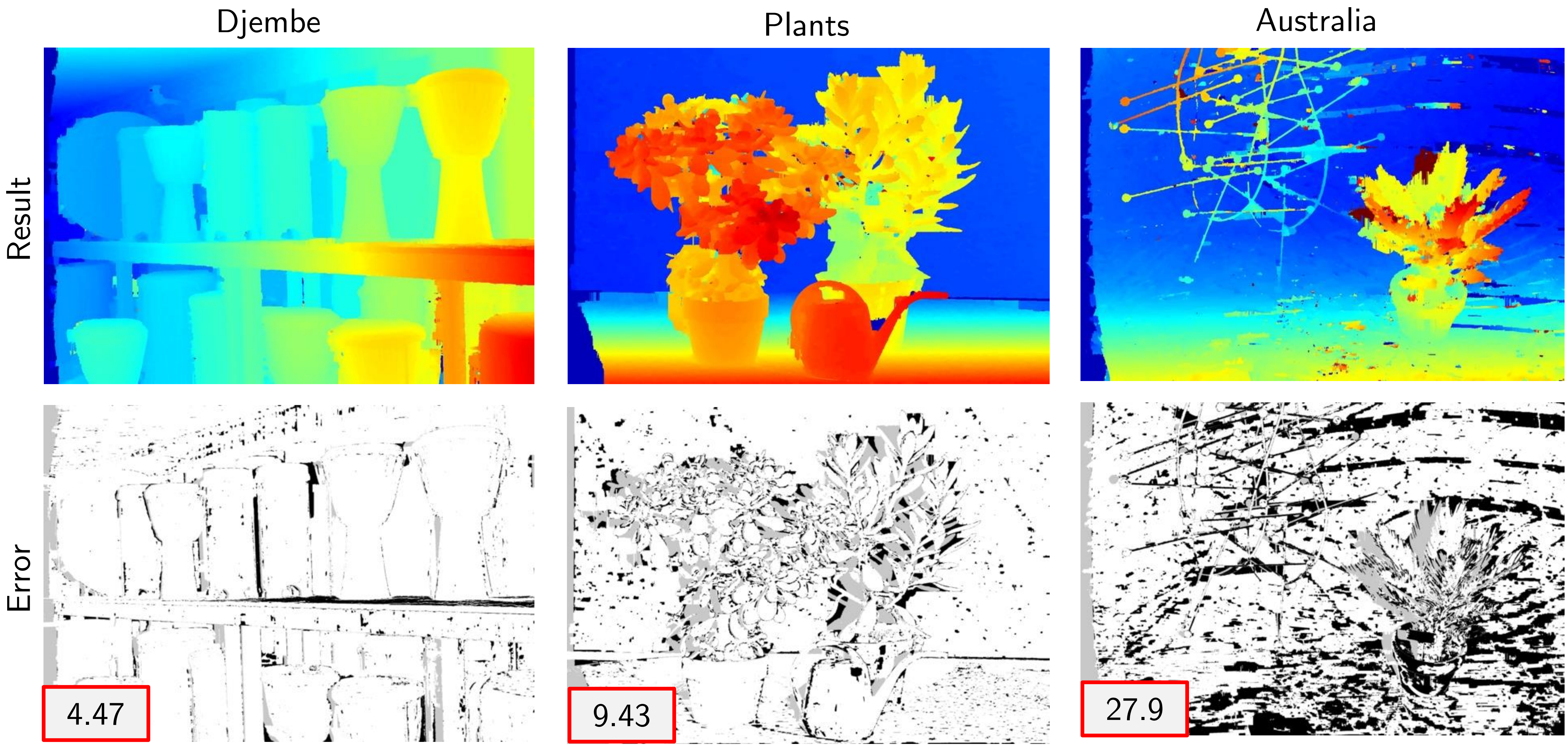
Comparison with state of the art methods

- Currently rank 7 of published algorithms

Method	Average Performance	Time/MP	Parameters	Post-Processing
MC-CNN	4.93	112s	830k	CA, SGM, SE, MF, BF
MC-CNN + RBS	5.10	140s	830k	CA, SGM, SE, MF, BF, RBS
Ours	9.71	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



Experiments – Kitti 2015

Comparison with state of the art methods

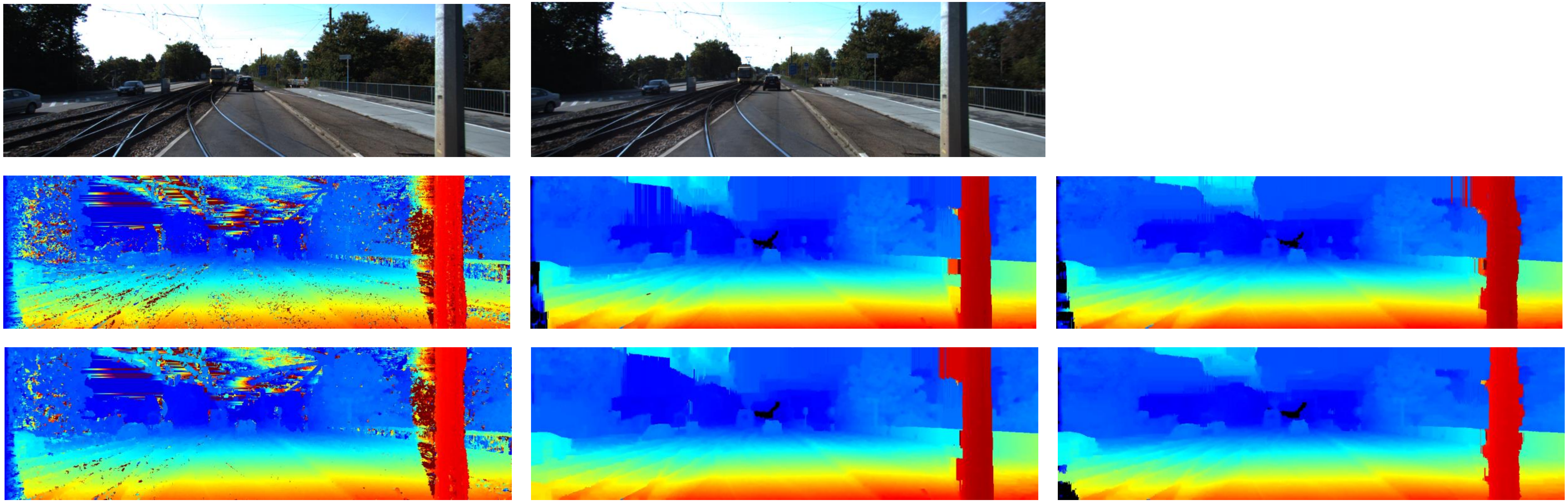
- Dataset specific for autonomous driving
- Currently rank 8 of published algorithms

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

Comparison of our models



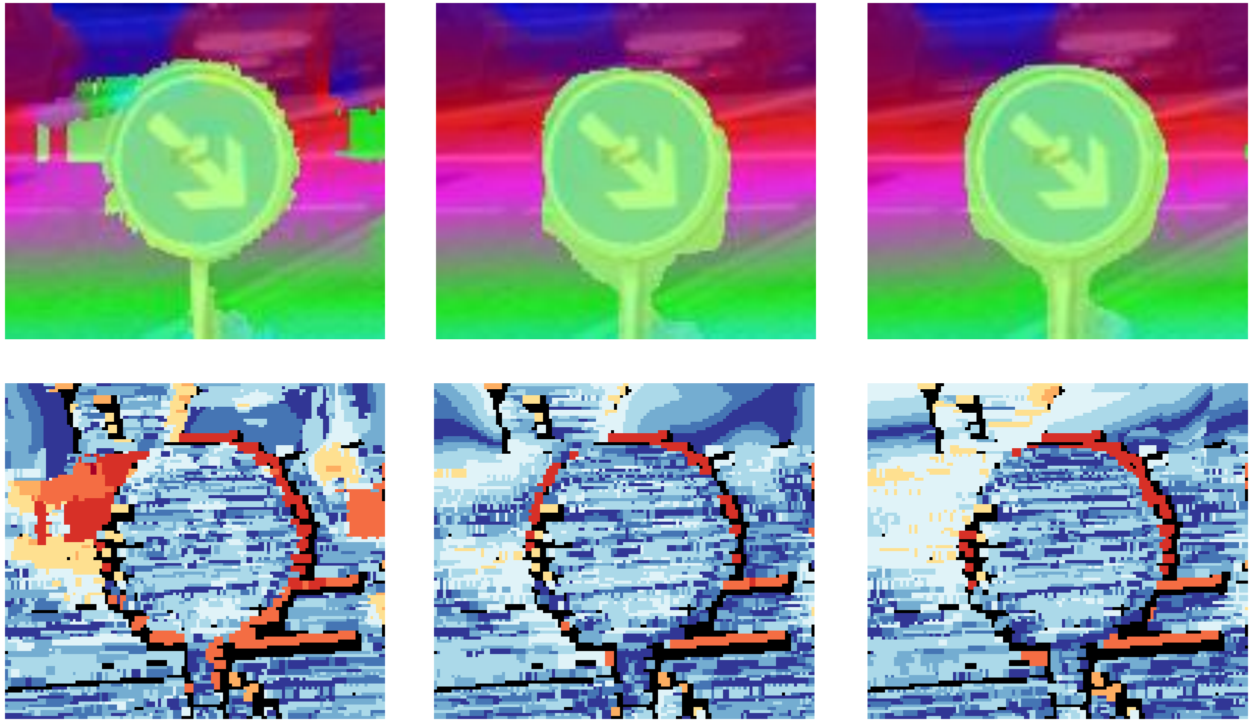
3 layer

7 layer

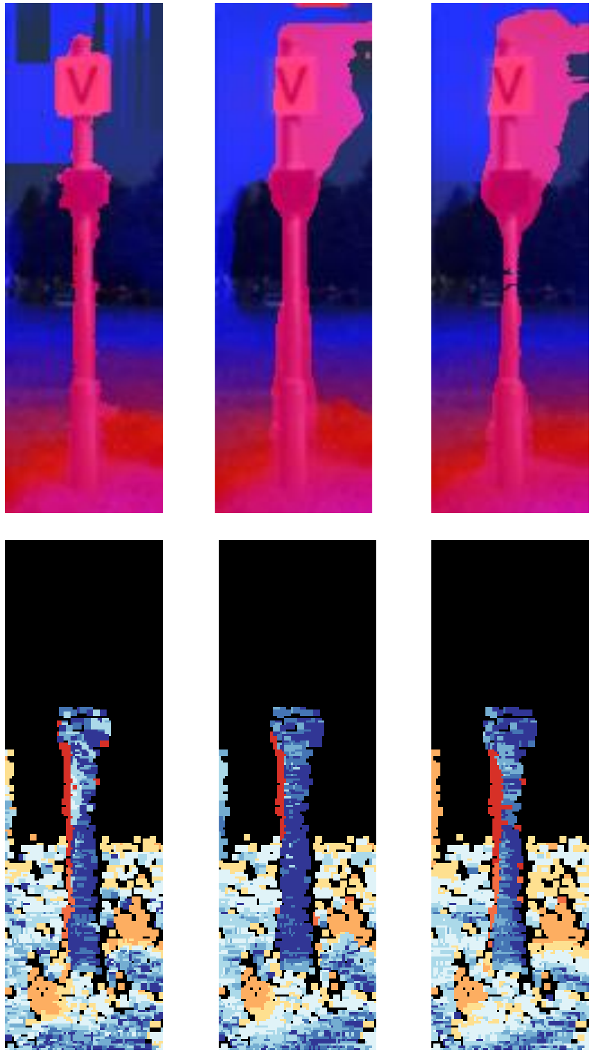
Unary CNN | Unary CNN + CRF | Full Joint

Experiments – Kitti 2015

Where do we make errors?



Ours MC-CNN ContentCNN



Ours MC-CNN ContentCNN

Conclusion & Future Work

Conclusion

- Fully trainable hybrid CNN+CRF model for stereo
- We showed how our model can be trained jointly
- It always pays off to replace hand-crafted features by learned features
- Joint training always decreases the error
- Even small models yield competitive performance when trained jointly

Future Work

- Gradient of unrolled inference
- Model occlusions explicitly
- Trainable continuous refinement

Thank you for your attention!