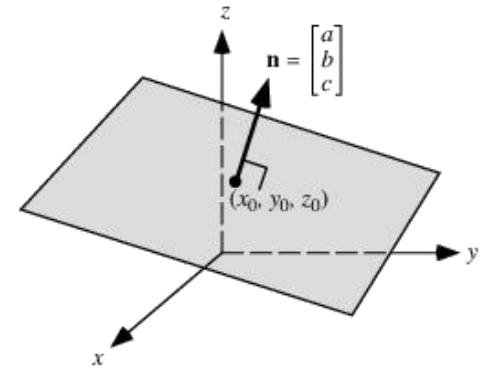
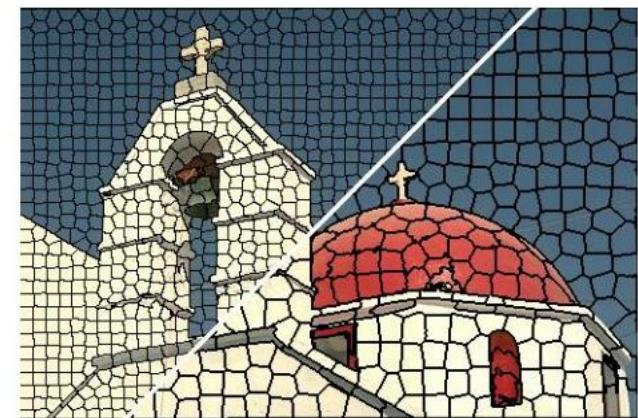
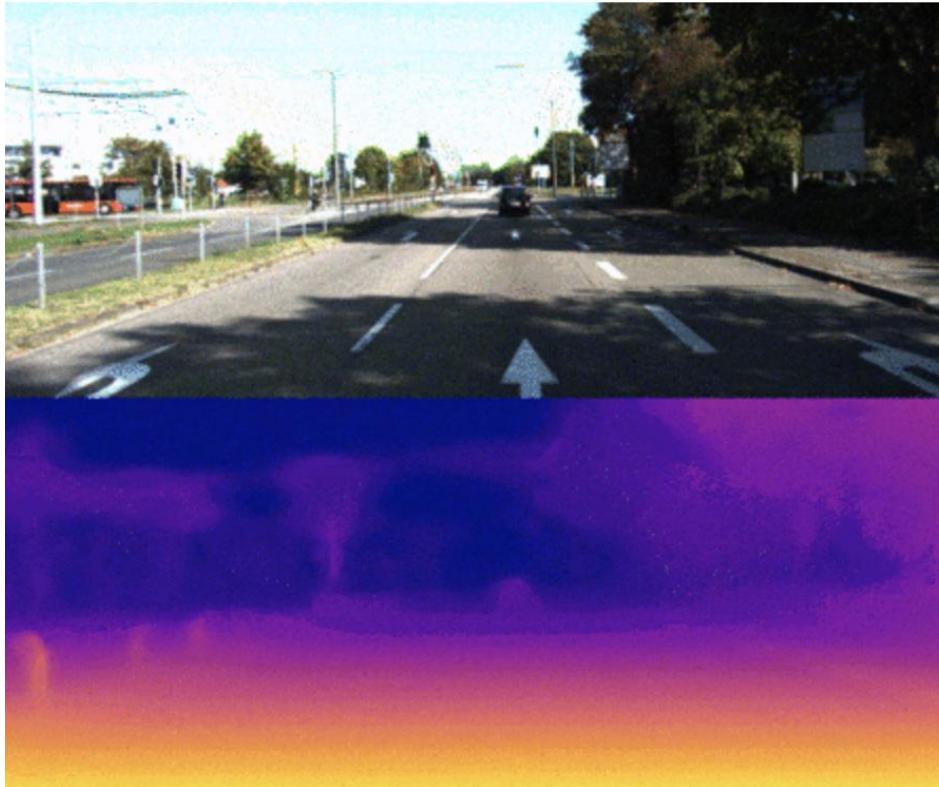


Final Presentation: Piecewise monocular depth estimation by plane fitting

Group 2: Felix Wirth, Korbinian Kunst, Helge Dörsan, Jan Ceccejus



Agenda



TECHNISCHE
UNIVERSITÄT
DARMSTADT

1. Overview and Goal
2. Architecture of Baseline NN - Godart et. al 2019
3. Improvement Approaches
 1. Iteration I: 4 Ch: RGB & SP, Normal2Depth (N2D) Block
 2. Iteration II: 4 Ch + N2D
 3. Iteration III: Superpixel Loss: Binary, Continuous
 4. Iteration IV: Normal in Loss
 5. Final: N2D + Normal Loss & Continuous Superpixel Loss
4. Results
5. Further Approaches
6. Review

1. Overview incl. Goal



Why is Depth important?

- autonomous driving, visual filter
- mono: automotive (cost efficient), medical (space restriction), mobile (space restriction)
- groundtruth data hard to obtain (lidar is sparse), stereo is rare

History

- Eigen [1] 2014 (supervised, introduced autoencoder)
- Deep 3D [6], Garg [7], Godard 2017 [8] (unsupervised, Disparity, Left Right Consistency)
- Zhou 2017 [3] (Camera Pose Estimation and novel view synthesis)
- Godard 2019 - Monodepth 2 [9] (Occlusion sensitive and edge sensitive)

Goal

1. improve depth (via surface normal and superpixels)
2. include normal estimation inside the architecture
3. show that it is possibility to directly estimate plane coefficients using CNNs

2. Architecture

Dataset: KITTI Dataset and Eigensplit

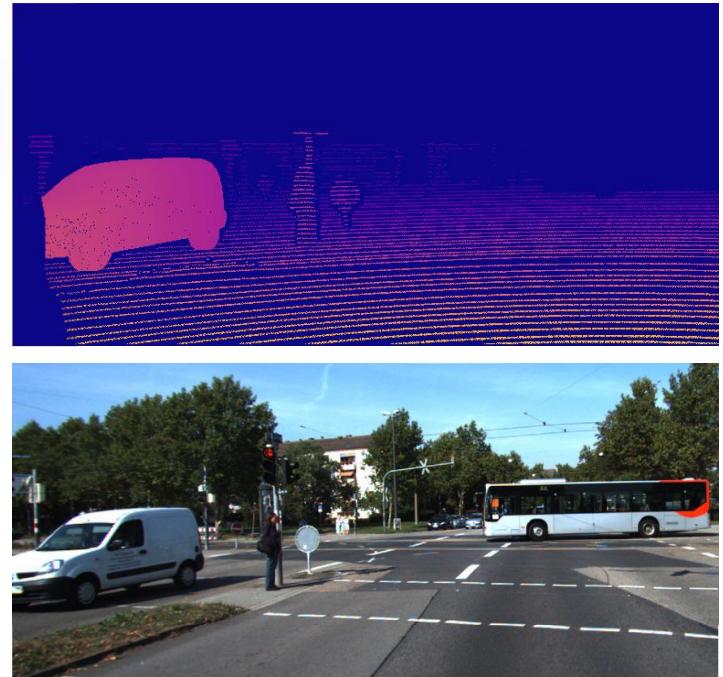
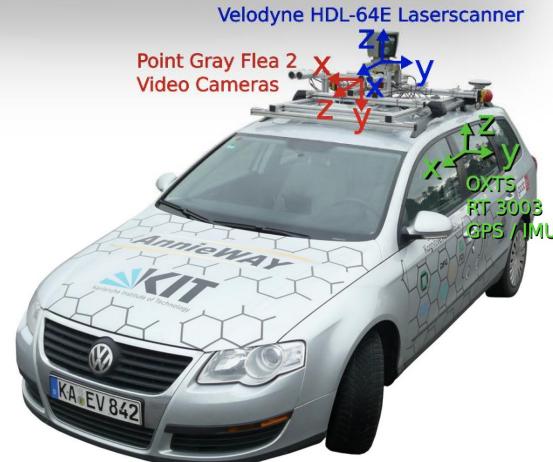


TECHNISCHE
UNIVERSITÄT
DARMSTADT

KITTI DATASET [2]

2 high - resolution color
and grayscale video
cameras

Velodyne laser
scanner



EIGEN ZHOU [3]

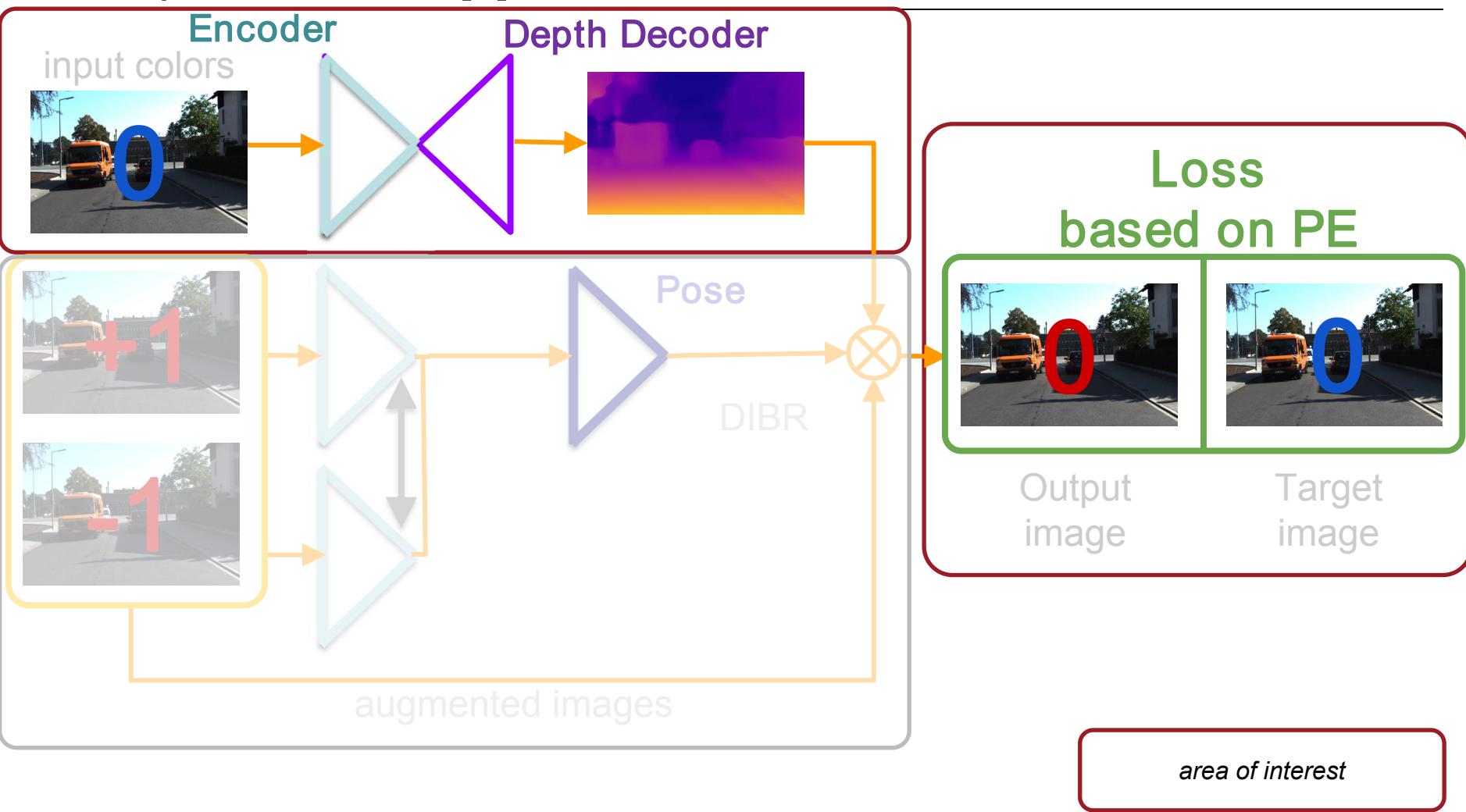
- Subset of KITTI proposed by Eigen in 2014 [1] and modified bei ZHOU in 2017 [3]
- uses 39.6K training images, 4424 validation images,
and 1000 test images,
which are resized to 640 px x 192 px

2. Architecture

Overview: Digging Into Self-Supervised Monocular Depth Estimation [1]



TECHNISCHE
UNIVERSITÄT
DARMSTADT



2. Architecture

Loss: Digging Into Self-Supervised Monocular Depth Estimation [1]



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Baseline – Loss

3. Improvements

3 pillar focus



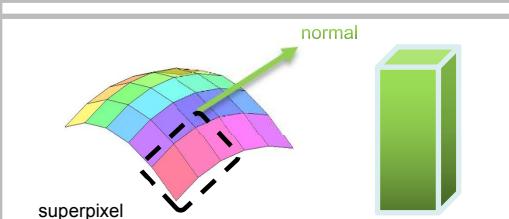
Input



Change input channels (I_t)

1. RGB (3-CH)
2. Superpixel & RGB (4-Ch)
3. RGB & S-RGB (6-Ch)

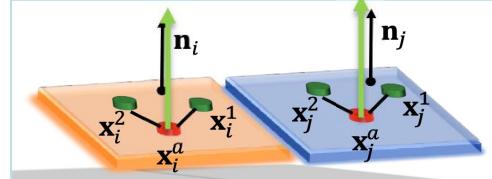
Architecture



Include transformation blocks inside the NN-flow

1. Normal-to-Depth Block
2. Averaging Normal based on SP

Loss



Let the NN learn with Normal and SP information

1. improve depth accuracy with superpixel boundary
2. enforce normal unity constraint on superpixel surface

3. Improvements

3 pillar focus



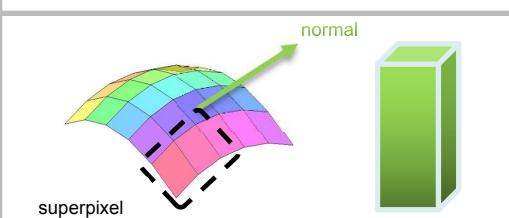
Input



change input channels (I_t)

1. RGB (3-CH)
2. Superpixel & RGB (4-Ch)
3. RGB & S-RGB (6-Ch)

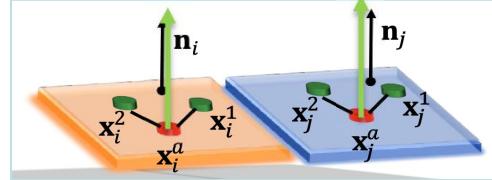
Architecture



include transformation blocks inside the NN-flow

1. Normal-to-Depth Block
2. Averaging Normal based on SP

Loss



let the NN learn with Normal and SP information

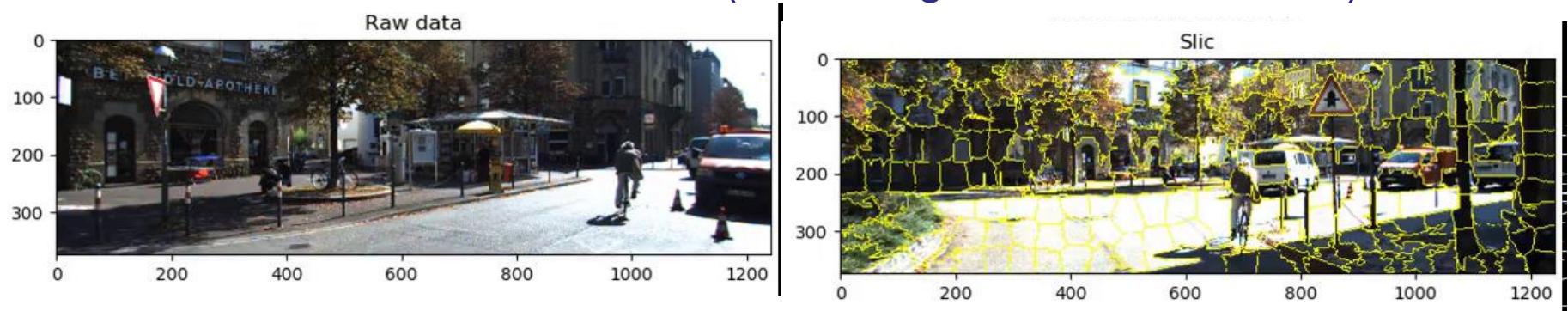
1. improve depth accuracy with superpixel boundary
2. enforce normal unity constraint on superpixel surface

3. Improvements

Iteration I: Basic I - Superpixel

- Def: superpixel can be defined as a group of pixels that share common characteristics (e.g. pixel intensity)

current favorite: **SLIC (Scale, Sigma, Minimum Size)**



Conclusion

superpixel can be usefull additional information for the net to approximate depth since they model surfaces and reduce image complexity into a composition of multiple plane

3. Improvements

Iteration I: 4 Channel Input (Superpixel & RGB)



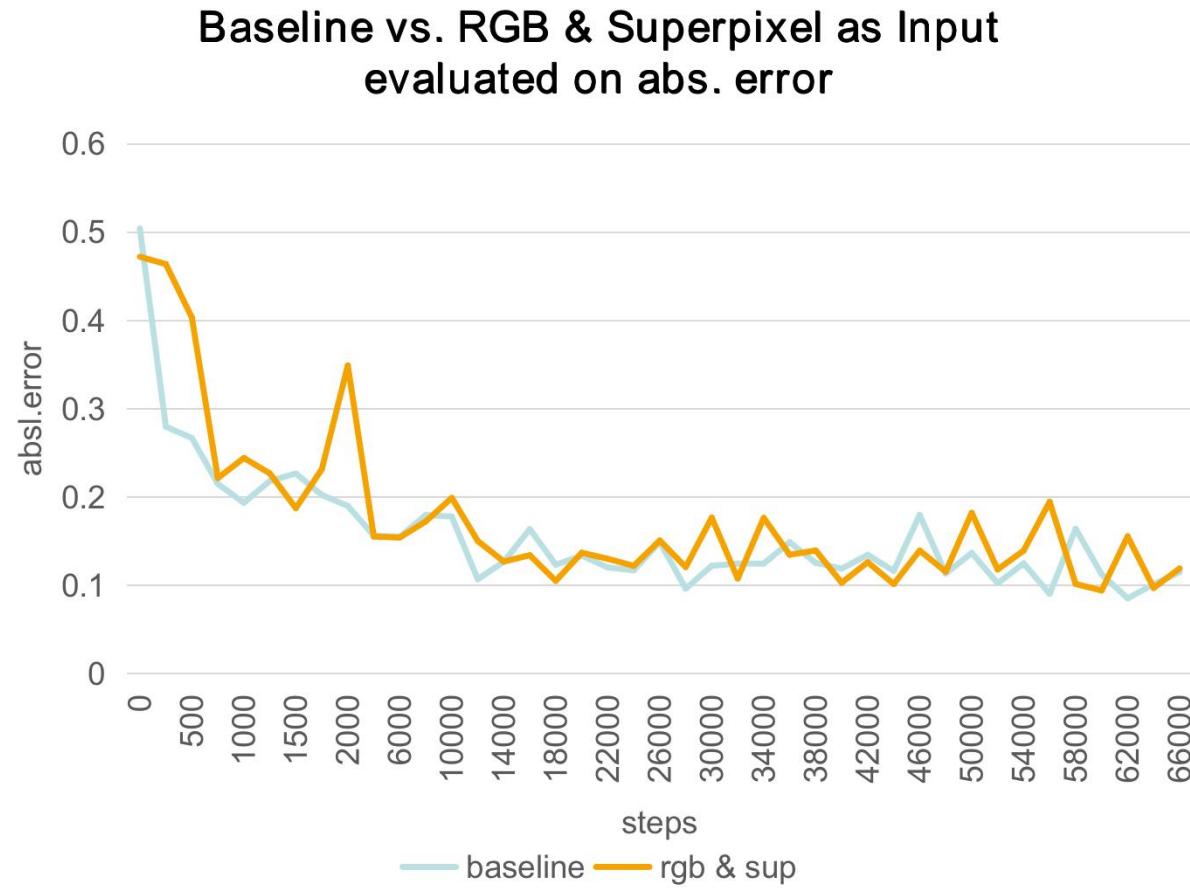
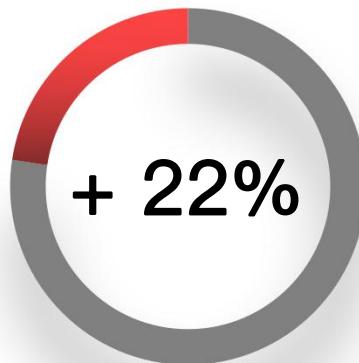
TECHNISCHE
UNIVERSITÄT
DARMSTADT

Testing:

Abs. rel. Error

Baseline: 0.115

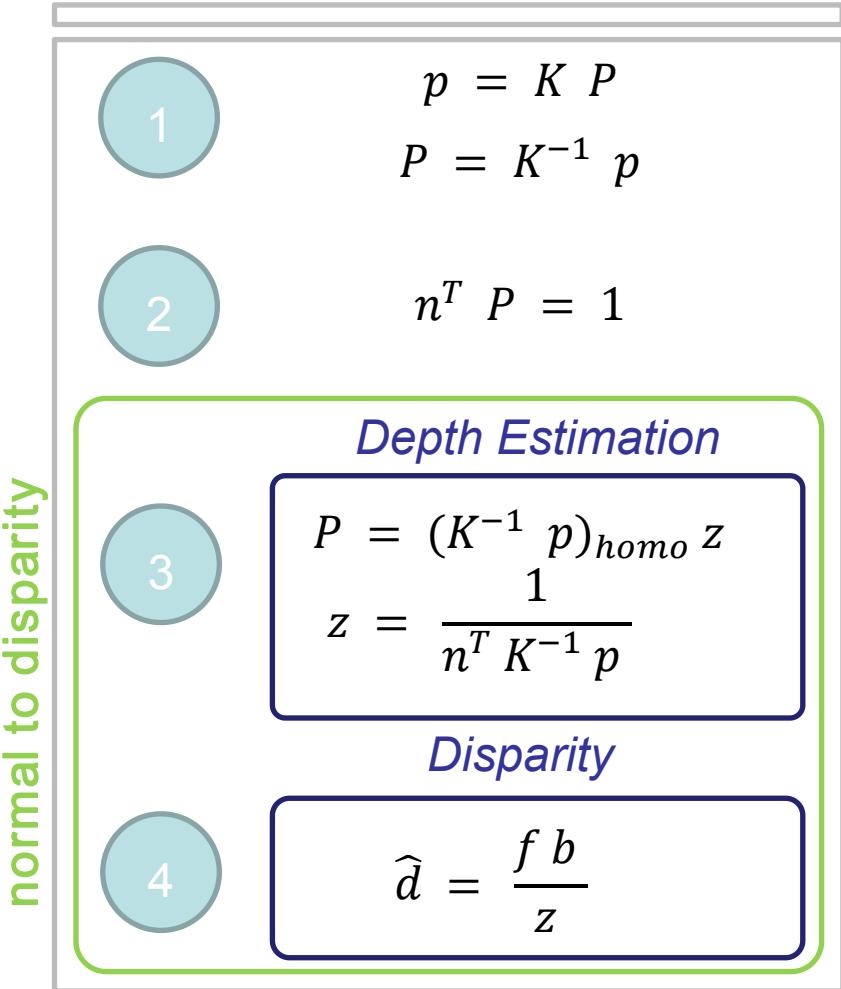
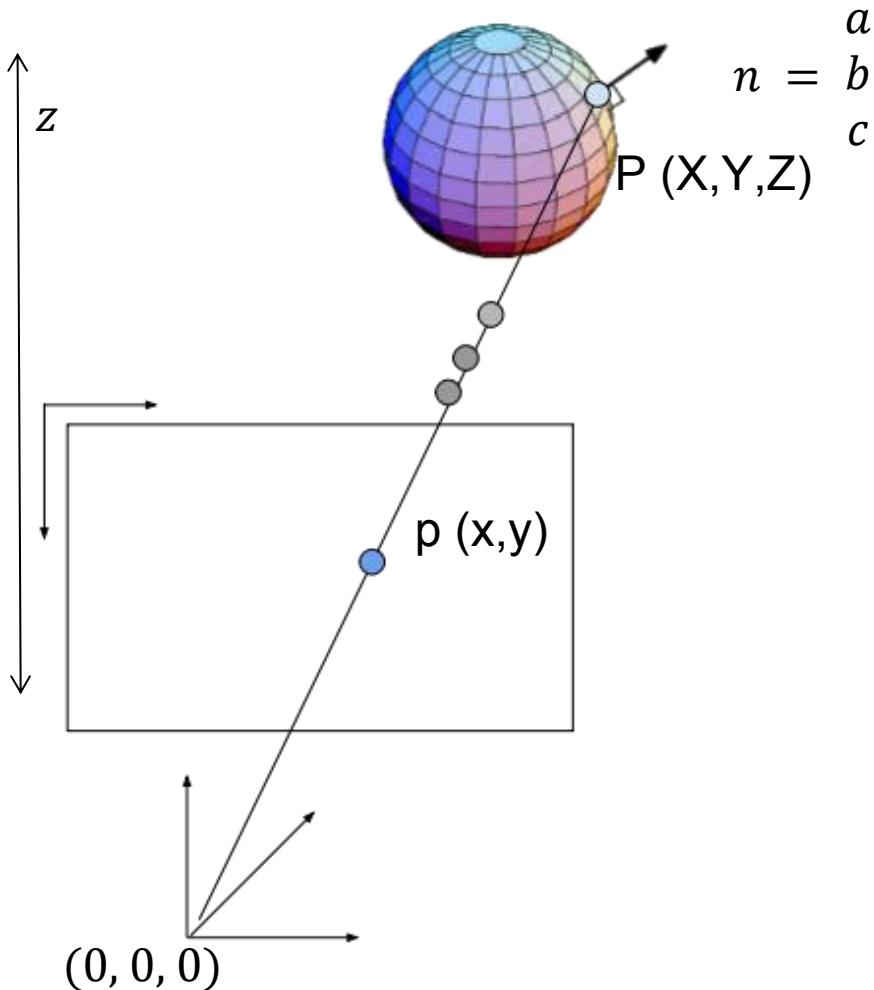
Current: 0.141



Absolute relative difference: $\text{rel} = \frac{1}{T} \sum_{i,j} |y_{i,j} - y_{i,j}^*| / y_{i,j}^*$

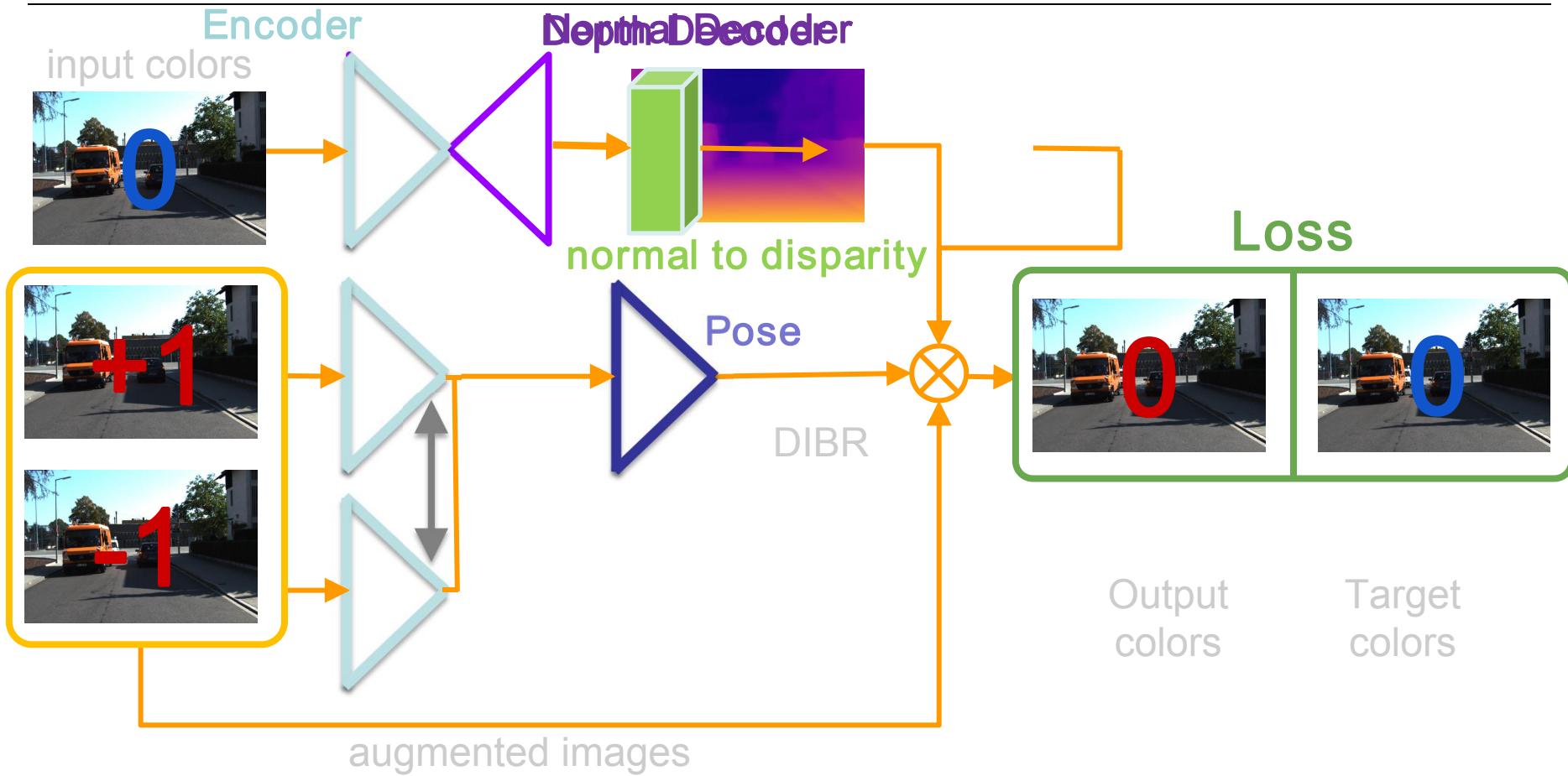
3. Improvements

Iteration I: Basic II - Normal Computation



3. Improvements

Iteration II: Introducing N2D Block



3. Improvements

Iteration I: Normal 2 Depth Block



Testing:

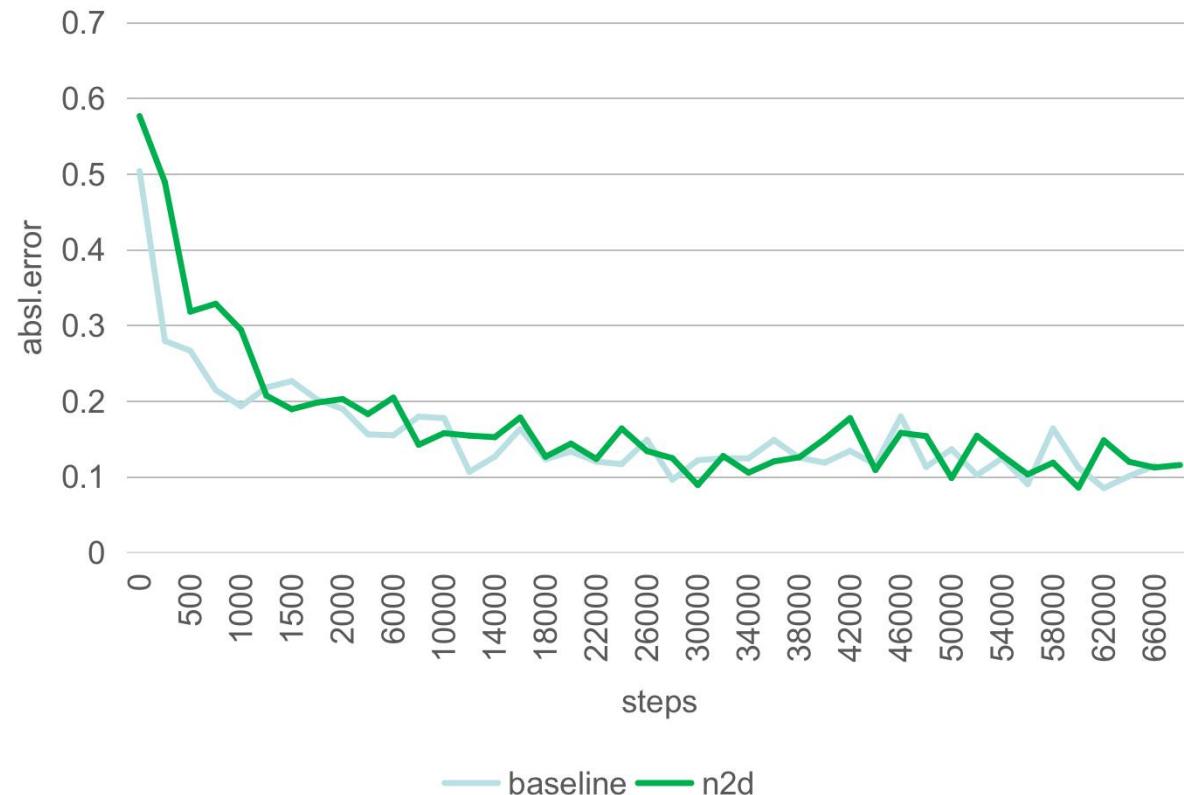
Abs. rel. Error

Baseline: 0.115

Current: 0.123



Baseline vs. Normal 2 Depth Block
evaluated on abs. error



3. Improvements

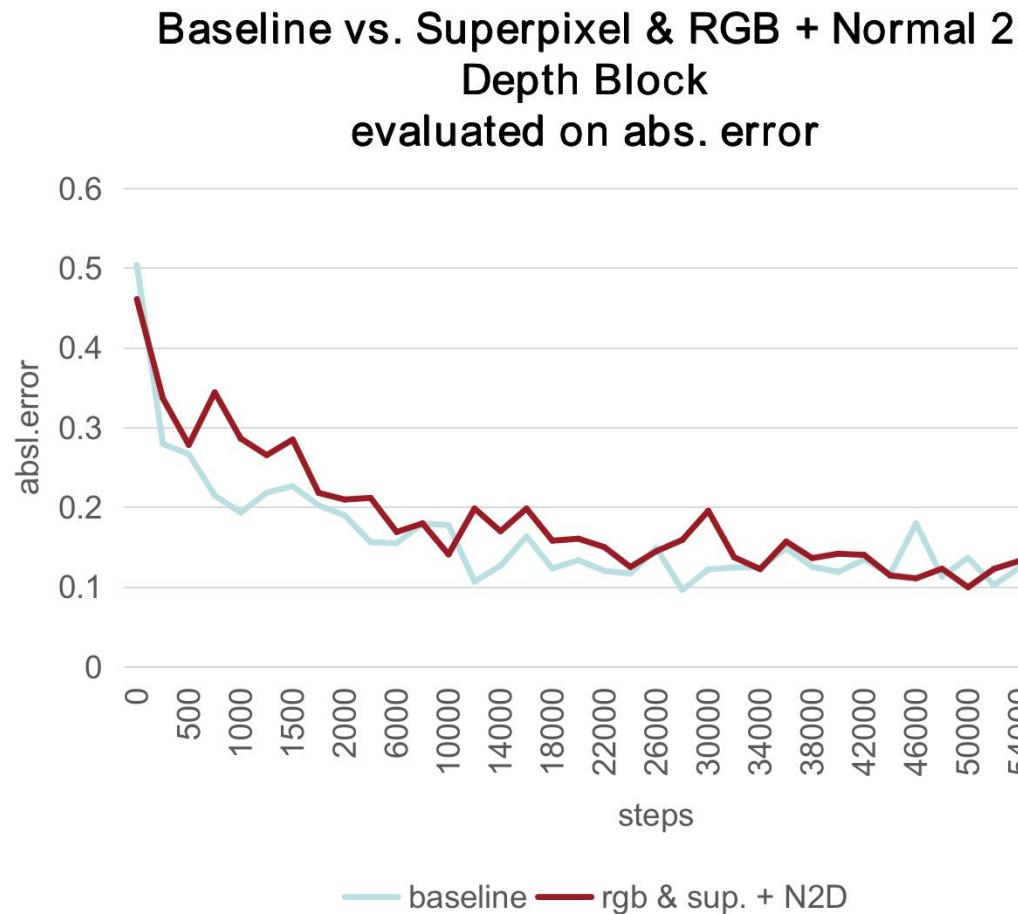
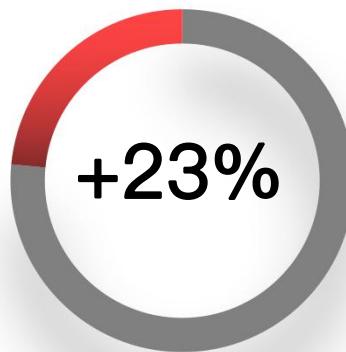
Iteration II: 4 Ch + N2D Block



Testing:
Abs. rel. Error

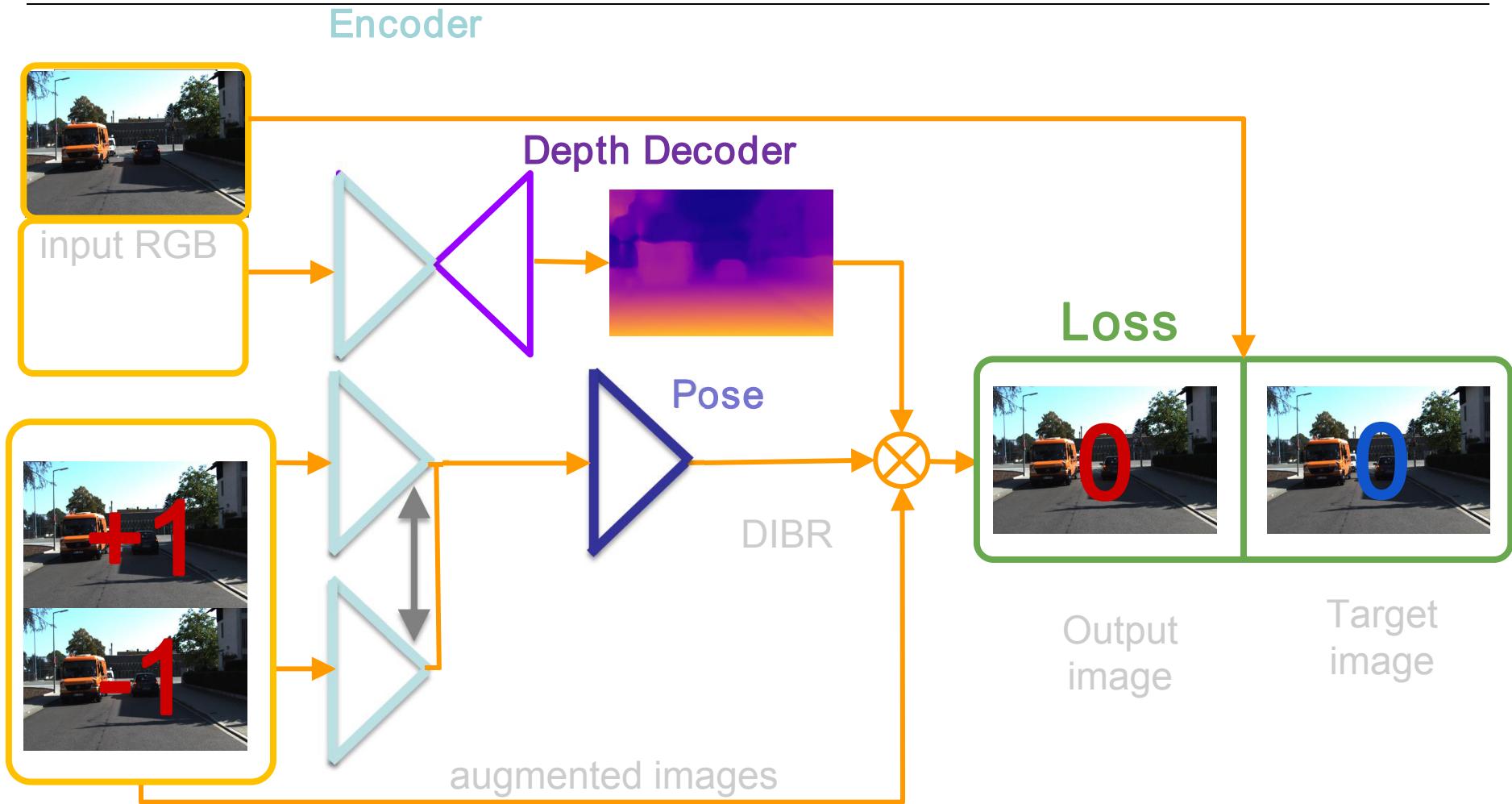
Baseline: 0.115

Current 0.142



3 . Improvements

Iteration III: Superpixel in Loss Function

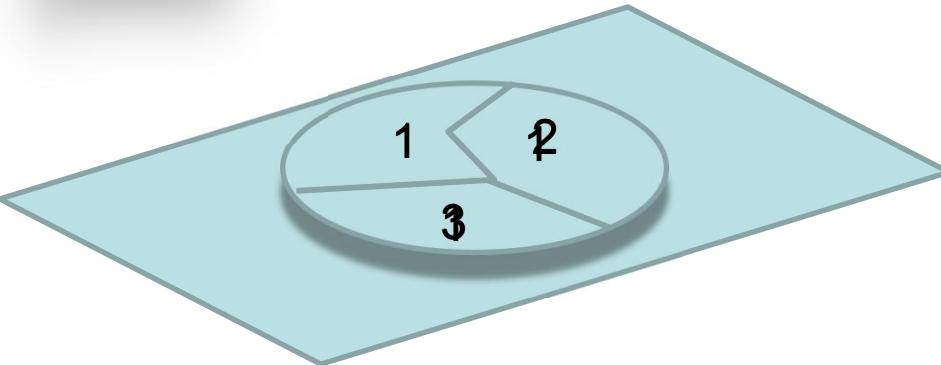


3 . Improvements

Iteration III: Superpixel in Loss Function



Concept Idea



use Superpixel information in implementing loss function

Problem: Oversegmentation:

- one plane is represented multiple superpixel segments

	1	1	1	
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
	1	1	1	

- Join Superpixels on same surface
- Minimize the disparity gradient in y and x direction in one plane
→ Let edges be rough

3 . Improvements

Iteration III: Binary & Continuous Superpixel Loss



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Mathematical
Implementation
Superpixel – Loss



3. Improvements

Iteration III: Binary Superpixel Loss



TECHNISCHE
UNIVERSITÄT
DARMSTADT

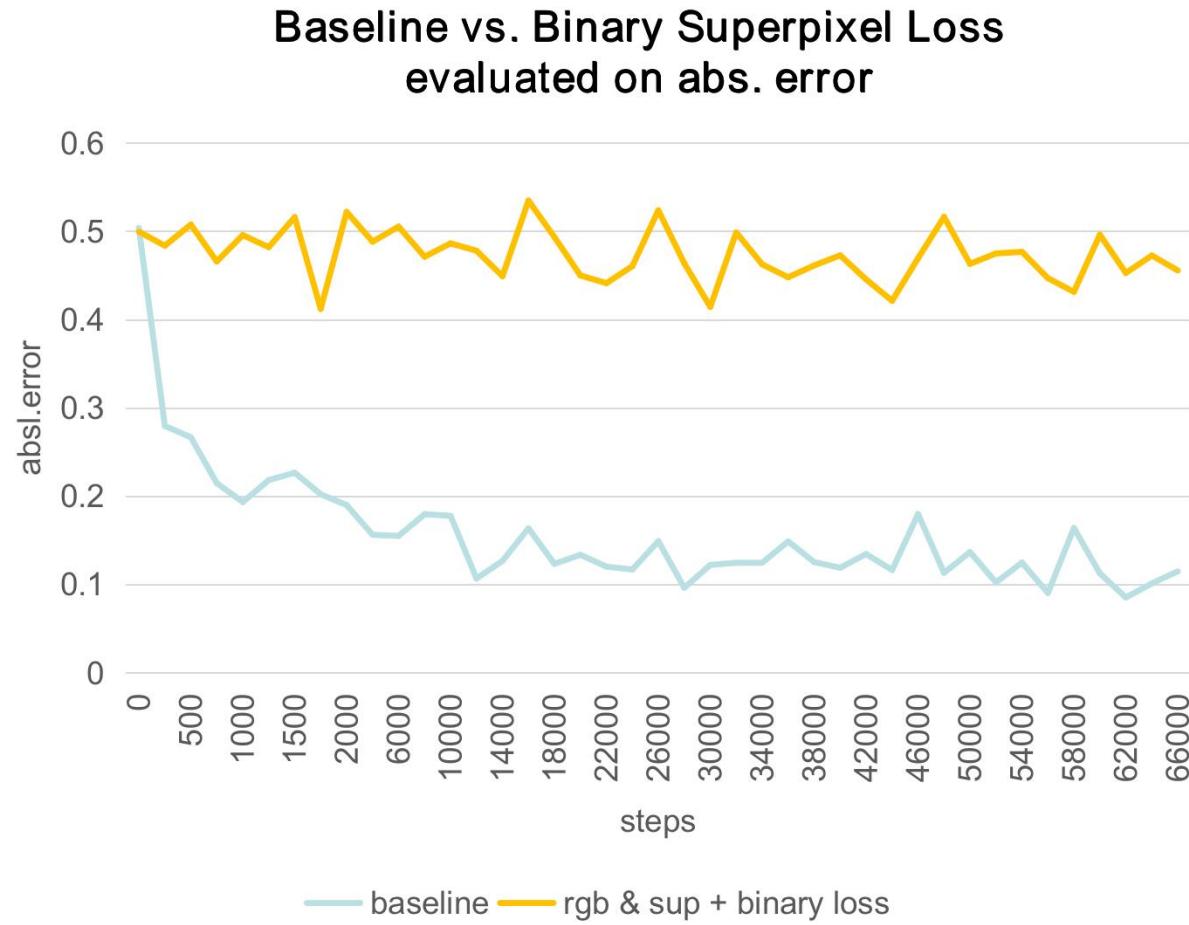
Testing:

Abs. rel. Error

Baseline: 0.115

Current: 0.443

+398%



3. Improvements

Iteration III: Continuous Superpixel Loss

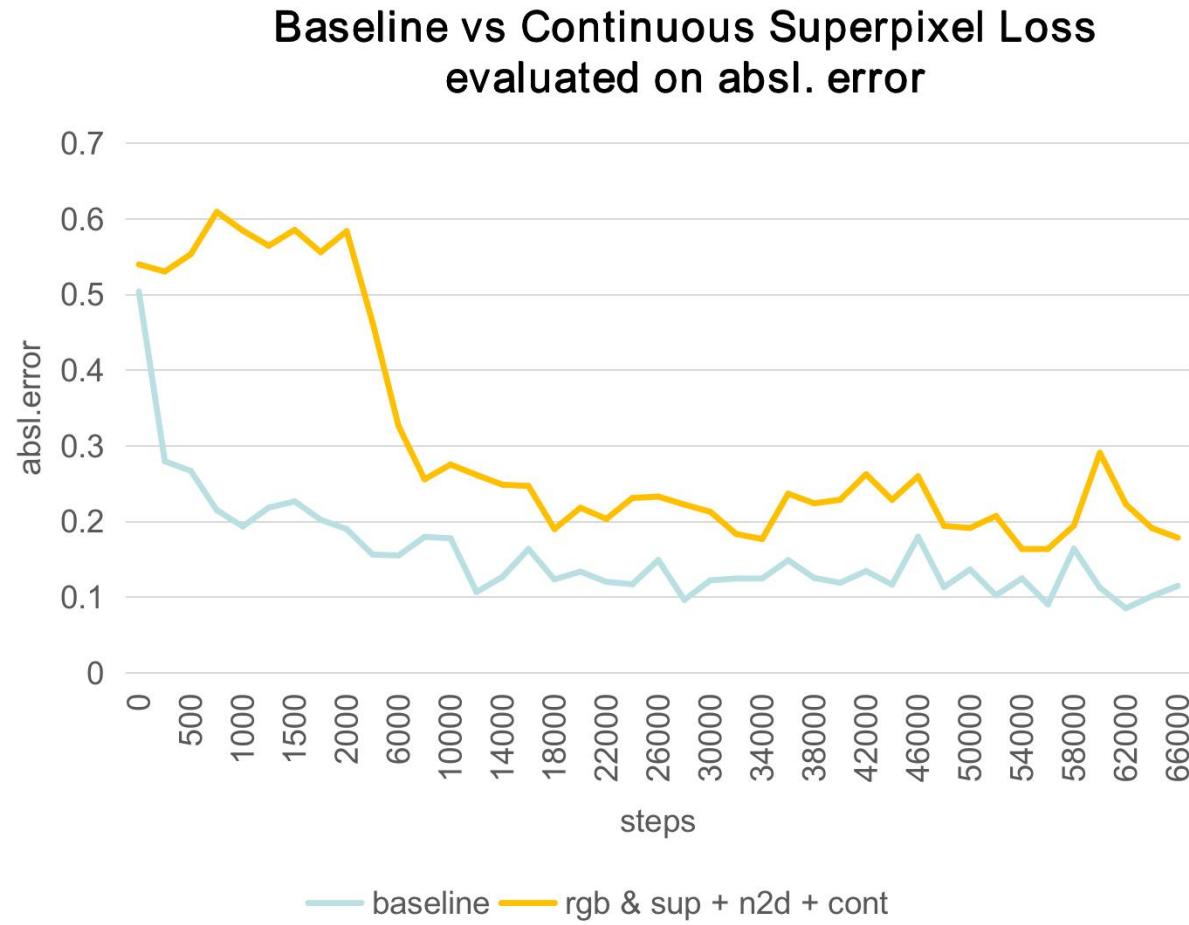


TECHNISCHE
UNIVERSITÄT
DARMSTADT

Testing:
Abs. rel. Error

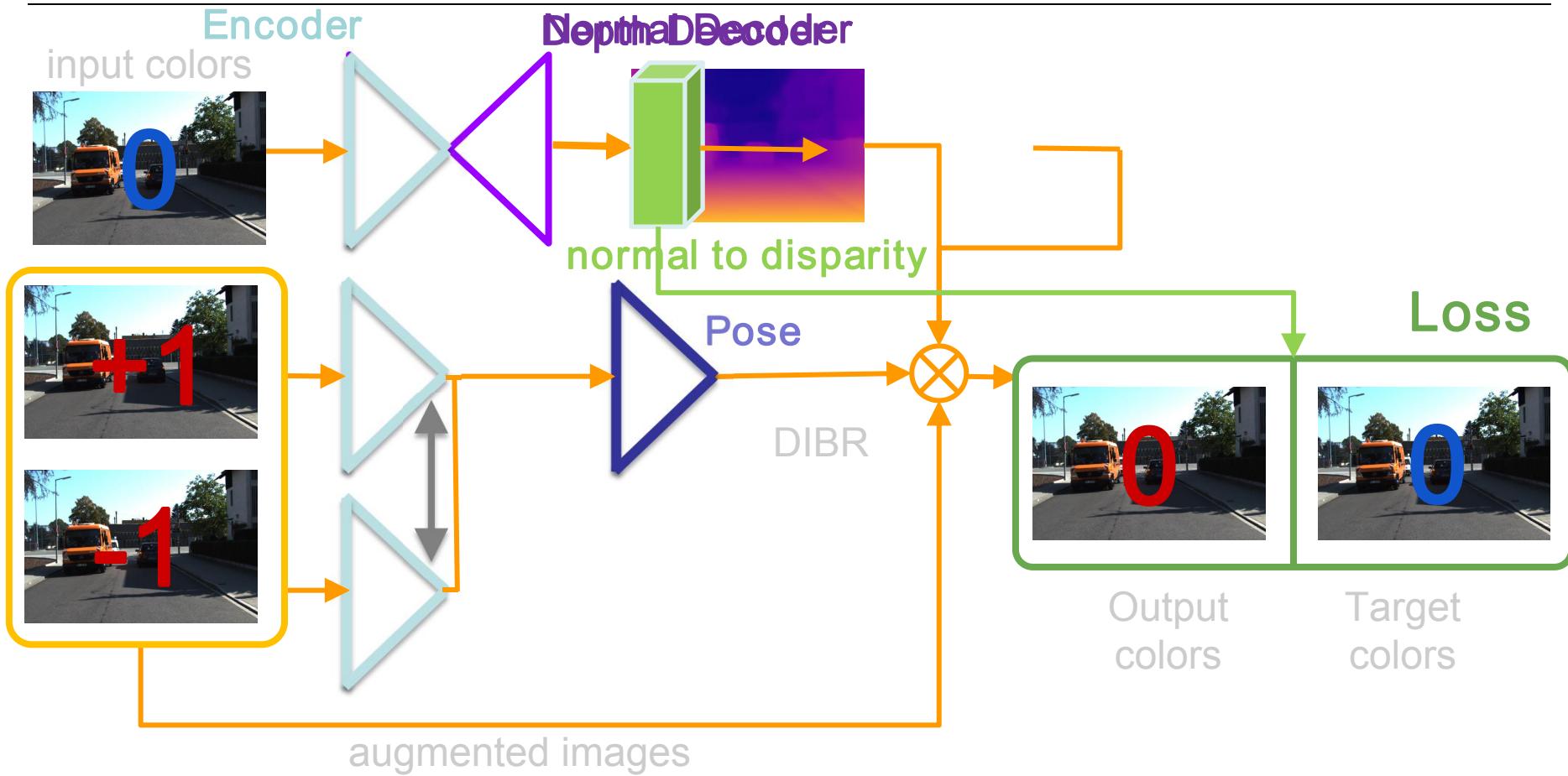
Baseline: 0.115

Current: 0.138



3. Improvements

Iteration IV: Normals in Loss Function



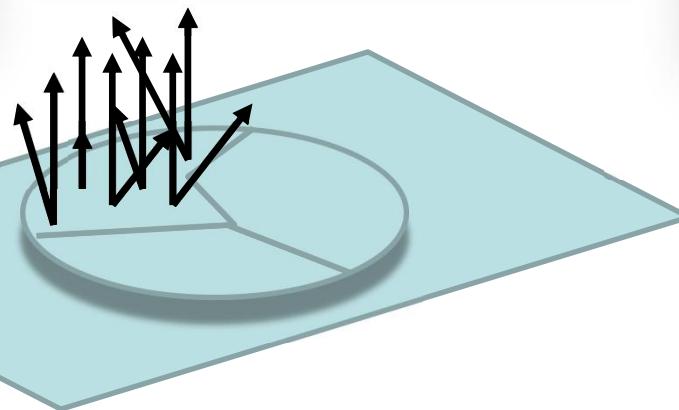
Loss Function

Normals in Loss Function



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Concept Idea



Implementation

- Force normals to be equal inside SPs
- Treat superpixels as planes represented by one normal vector

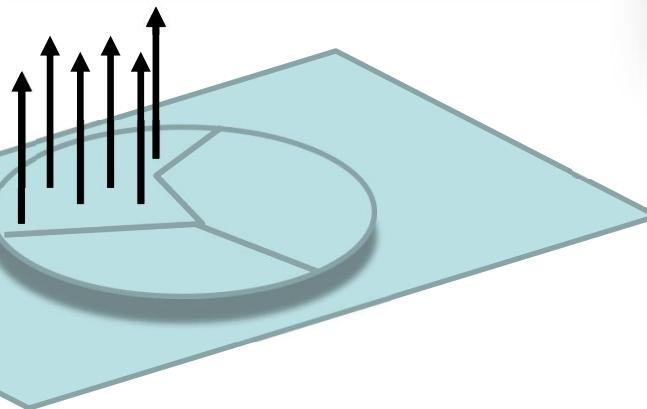
Loss Function

Normals in Loss Function



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Concept Idea



Implementation

$$L_N = \sum_{i=0}^{N(SP)} (\sigma_{x_i} + \sigma_{y_i} + \sigma_{z_i})$$

$$L = \mu L_p + \lambda L_s + \alpha L_N$$

- Force normals to be equal inside SPs
- Treat superpixels as planes represented by one normal vector

L_p : Photometric Loss

L_s : Smoothness Loss

L_N : Normal Loss

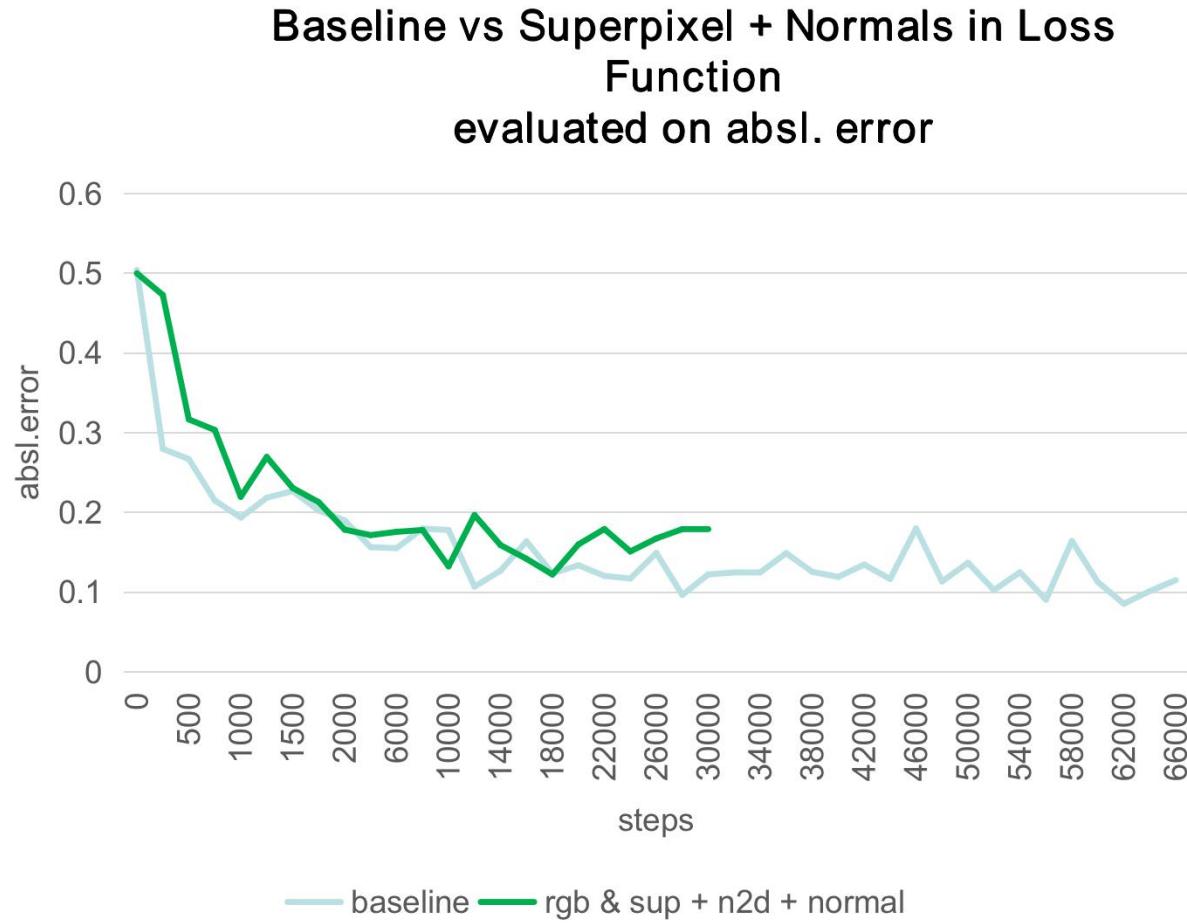
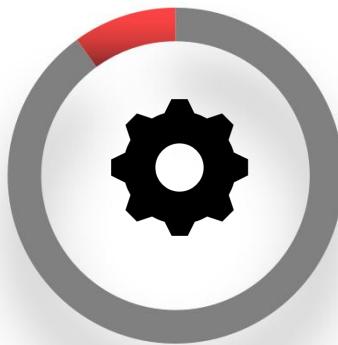
3. Improvements

Iteration IV: Normals in Loss Function



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Testing:
Abs. rel. Error
Baseline: 0.115
Current: still running

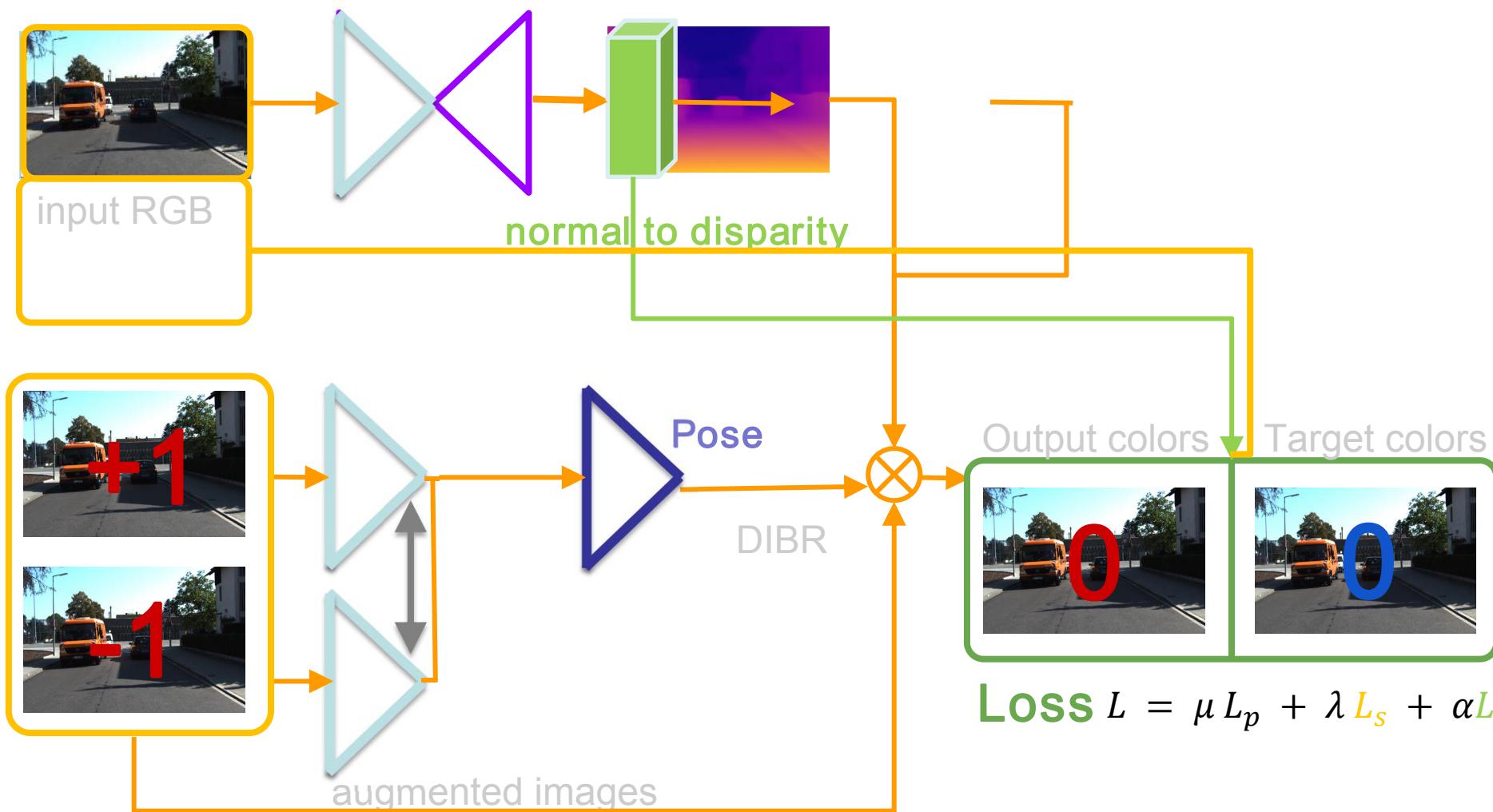


3. Improvements

Iteration V: Normal Loss & Continuous Superpixel Loss



TECHNISCHE
UNIVERSITÄT
DARMSTADT



3. Improvements

Iteration V: Normal Loss & Continuous Superpixel Loss



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Testing:

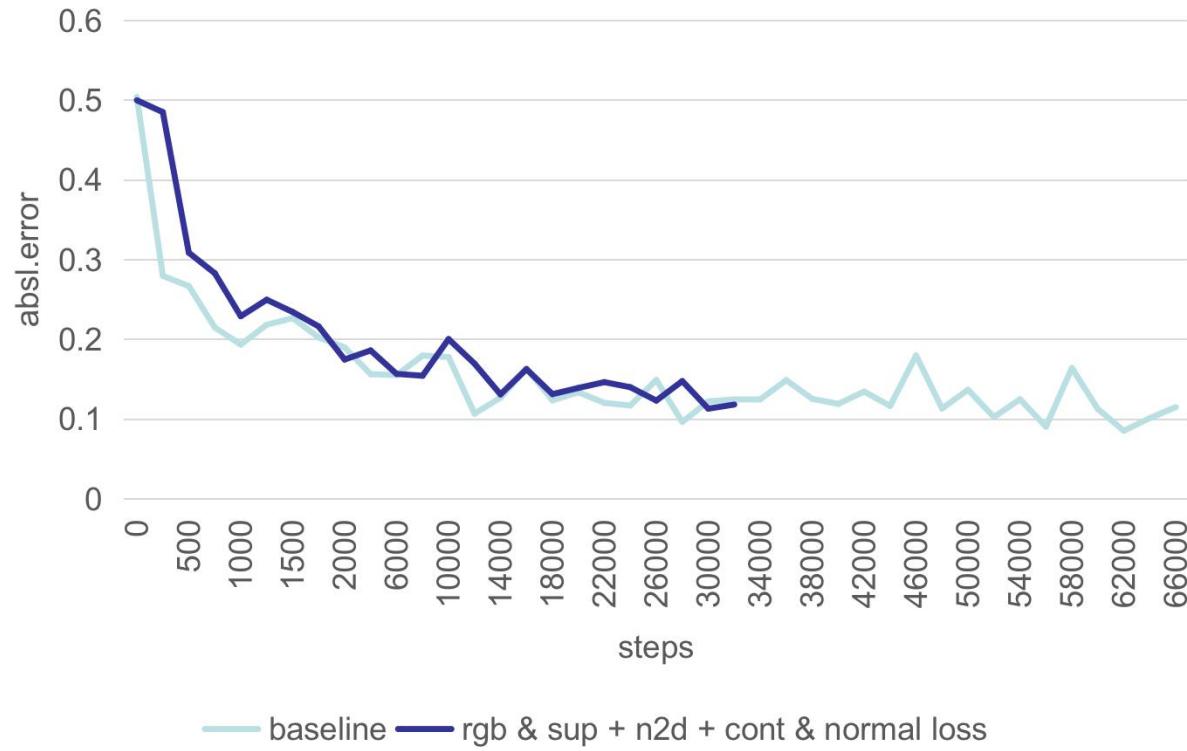
Abs. rel. Error

Baseline: 0.115

Current: still running



Baseline vs Normal Loss & Continuous
Superpixel
evaluated on absl. error



4. Results

All Approaches



TECHNISCHE
UNIVERSITÄT
DARMSTADT

$$\text{Absolute relative difference: } \text{rel} = \frac{1}{T} \sum_{i,j} |y_{i,j} - y_{i,j}^*| / y_{i,j}^*$$

Run	Input	Architecture	Loss Function	Run-Time	absolute relative error*
0	RGB	Baseline	Baseline	12 h	0,115
1	RGB+Superpixel	Baseline	Baseline	22 h	0,141
2	RGB	Normal-to-Depth Block	Baseline	14 h	0,123
3	RGB+Superpixel	Normal-to-Depth Block	Baseline	24 h	0,142
4	RGB+Superpixel	Normal-to-Depth Block	Binary	15 h	0,443
5	RGB	Normal-to-Depth Block	Binary	17 h	0,443
6	RGB+Superpixel	Normal-to-Depth Block	Continous	14 h	0,138
7	RGB	Normal-to-Depth Block	Continous	15 h	0,443
8	RGB+Superpixel	Normal-to-Depth Block	Normal loss		Still running
9	RGB+Superpixel	Normal-to-Depth Block	Cont + normal		Still running
10	RGB+Superpixel	Normal-to-Depth Block	Binary + normal		Still running
11	RGB+Superpixel	Averaging Normal based on SP	baseline		Still running
12	RGB+Superpixel	Normal-to-Depth Block + Averaging Normal based on SP	baseline		Still running

4. Results

Demo: Compare Results



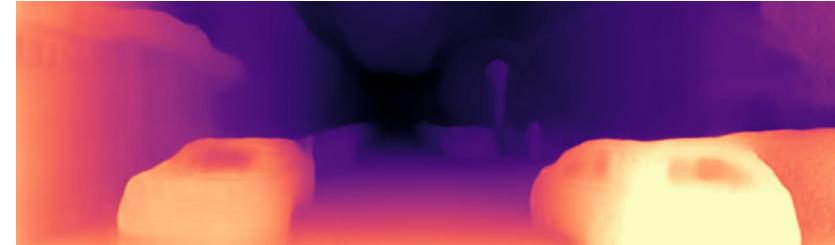
TECHNISCHE
UNIVERSITÄT
DARMSTADT

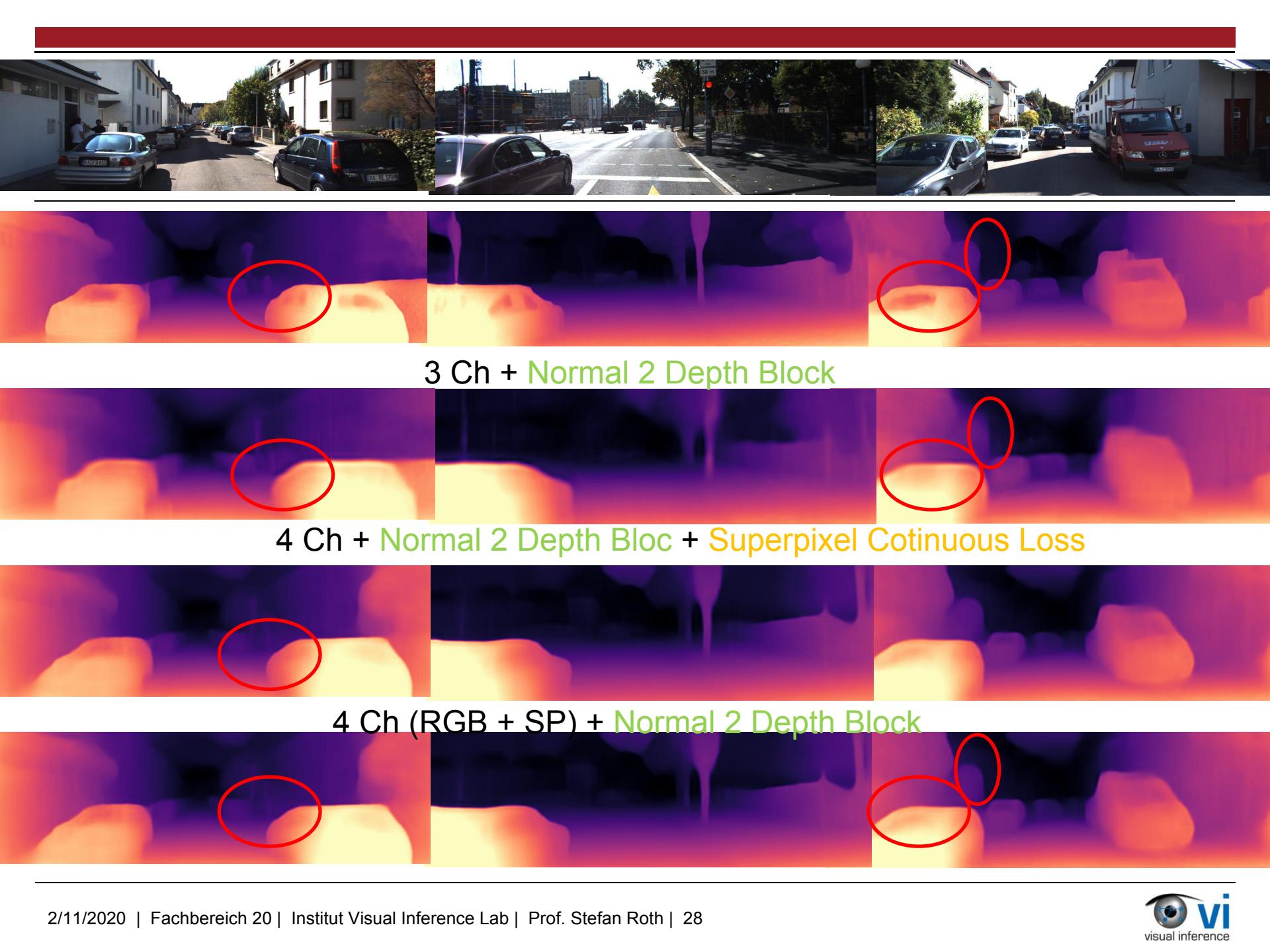
Evaluation of our result based on three test images

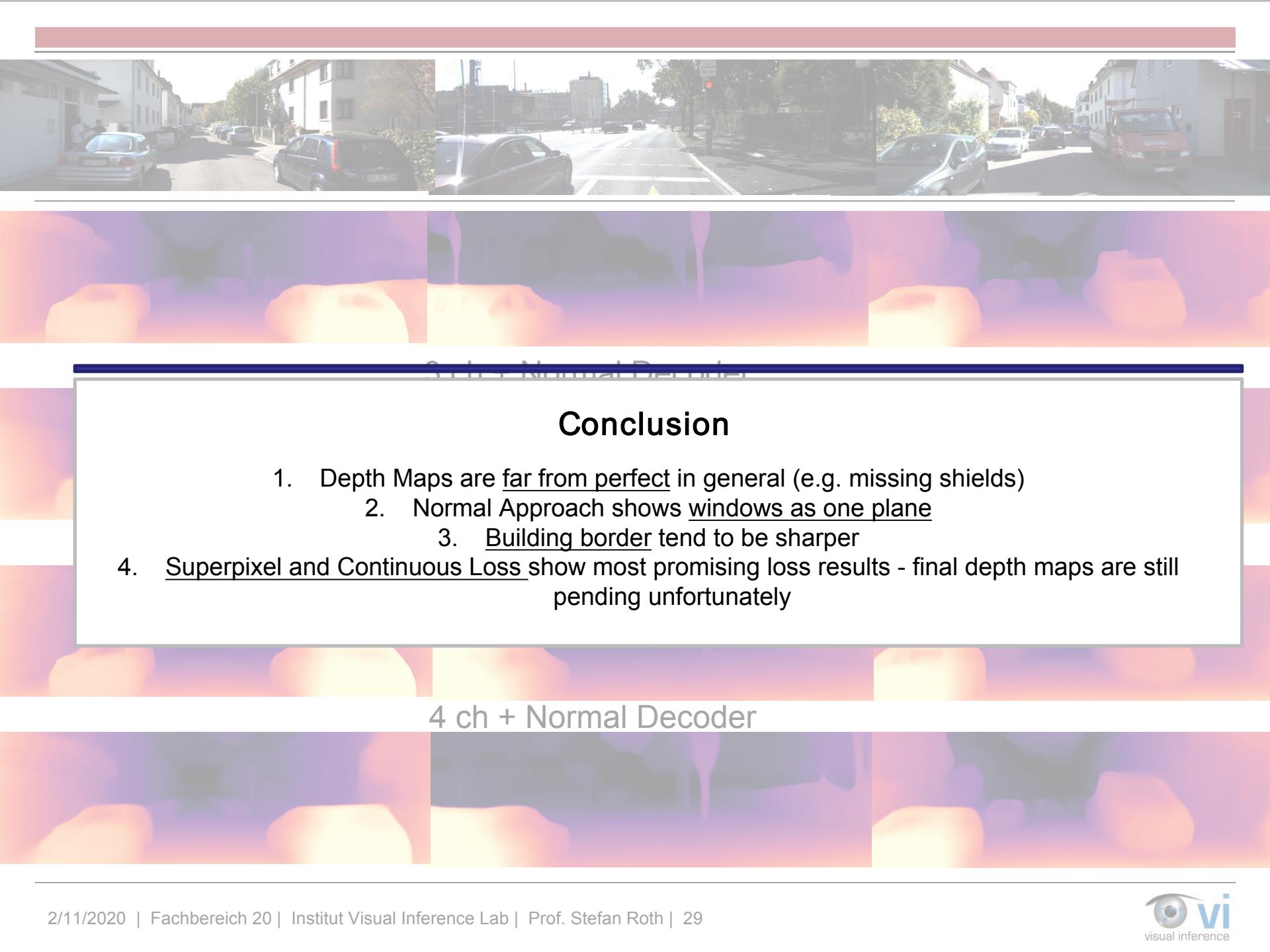
chosen on criteria: close objects, open streets, buildings, street shields



Baseline





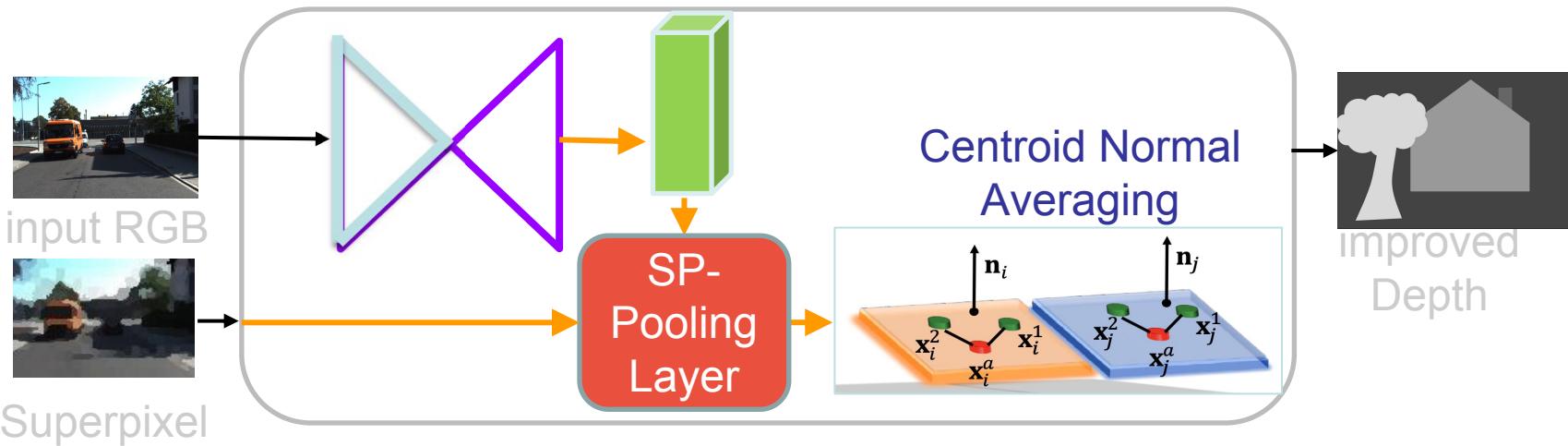


5. Further approaches

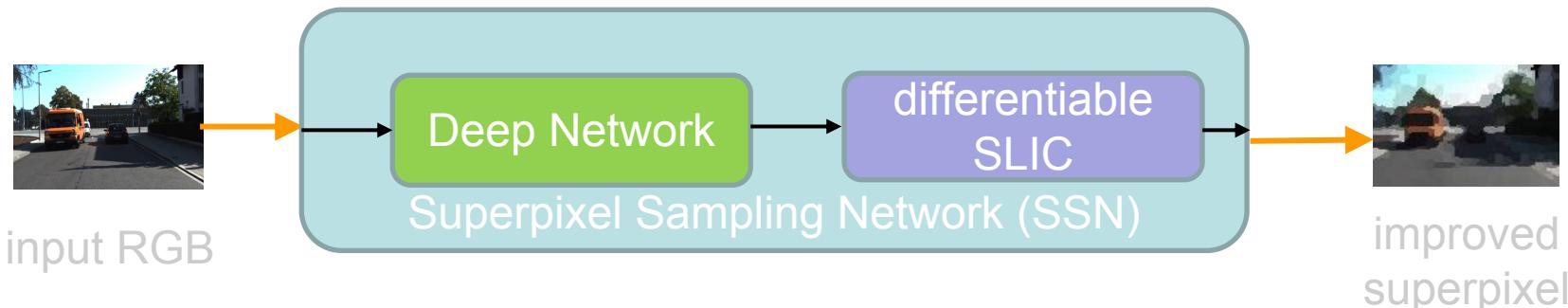
Improvement of Loss



1. Use centroid information for superpixel surface modification



2. Learn Superpixel with hyperparameter and learning part itself



6. Review



- Understanding the literature and getting acquainted with a deep learning library (PyTorch) ✓
- Prepare the dataset and run baseline ✓
- Choose preferred superpixel generation algorithm and generate new dataset ✓
- create new loss functions ✓
- train the network to estimate the normal coefficients ✓
- Evaluate the result on public benchmark datasets ✓
- beat current network error score ✗
- Writing of a report ✗

Literature

- [1] D Eigen, C Puhrsch, and R Fergus. Depth map prediction from a single image using a multi-scale deep network. In Advances in Neural Information Processing Systems, 2014.
- [2] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 3354-3361.
- [3] Tinghui Zhou, Matthew Brown, Noah Snavely, David Lowe, "Unsupervised Learning of Depth and Ego-Motion" from Video in CVPR 2017
- [4] D. Stutz, A. Hermans, B. Leibe, Superpixels: An Evaluation of the State-of-the-Art
- [5] Murong Wang, Xiabi Liu, Yixuan Gao, Xiao Ma, Nouman Q. Soomro, Superpixel segmentation: A benchmark, Signal Processing: Image Communication, Volume 56, 2017, Pages 28-39,
- [6] Junyuan Xie and Ross Girshick and Ali Farhadi, Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks, 2016
- [7] Ravi Garg and Vijay Kumar BG and Gustavo Carneiro and Ian Reid, Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue, 2016
- [8] Clément Godard and Oisin Mac Aodha and Gabriel J. Brostow, Unsupervised Monocular Depth Estimation with Left-Right Consistency, 2016
- [9] Clement Godard and Oisin Mac Aodha and Michael Firman and Gabriel J. Brostow, Digging into Self-Supervised Monocular Depth Prediction, 2019

7. Backup Slides

- Net Detail View
- Superpixel Benchmark Paramter
- Error Metrics

2. Architecture

LOSS: Digging Into Self-Supervised Monocular Depth Estimation [1]



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Photometric Error L_p

- SSIM
- L1- Norm

Depth

Pose

$$I_{t \rightarrow t'} = I_t < \text{proj}(\mathbf{D}_t, \mathbf{T}_{t \rightarrow t'}, K) >$$

$$L = \mu * \min \left(\frac{\alpha}{2} \left(1 - \text{SSIM}(I_t, I_{t \rightarrow t'}) \right) + (1 - \alpha) \|I_t - I_{t \rightarrow t'}\| \right)$$

$$+ \lambda (|\partial_x \mathbf{d}_t| * e^{\{-|\partial_x I_t|\}} + |\partial_y \mathbf{d}_t| * e^{\{-|\partial_y I_t|\}})$$

Smoothness Loss L_s

- $\partial_i \mathbf{d}_t$: Gradient smoothness
- $e^{\{\partial_x |I_t|\}}$: Edge aware smoothness

Disparity

Loss Function

Superpixel Binary Loss Function



Mathematical Implementation Binary SP–Loss

Goal

$$L_s = \lambda (|\partial_x d_t| * \mu_x + |\partial_y d_t| * \mu_y)$$

$$\mu_{x/y} = \begin{cases} 0 & , \text{on edges} \\ 1 & , \text{inside plane} \end{cases}$$

as part of the smoothness loss

$$\lambda (|\partial_x \mathbf{d}_t| * e^{\{-|\partial_x I_t|\}} + |\partial_y \mathbf{d}_t| * e^{\{-|\partial_y I_t|\}})$$

Calculate gradient of SP-labels

- $|\partial_x \mathbf{SP}|$ & $|\partial_y \mathbf{SP}|$

If $|\partial_{x/y} \mathbf{SP}| \approx 0$ & $|\partial_{x/y} I_t| < \text{threshold}$:

$$|\partial_{x/y} \mathbf{SP}| = 0$$

else:

$$|\partial_{x/y} \mathbf{SP}| = 1$$

Loss Function

Superpixel Continuous Loss Function



Mathematical
Implementation
Continuous SP-Loss

Goal

$$L_s = \lambda (|\partial_x d_t| * \mu_x + |\partial_y d_t| * \mu_y)$$

$$\mu_{x/y} = \begin{cases} e^{-|\partial_{x/y} I_t|} & , \text{on SP-edges} \\ 1 & , \text{inside SP} \end{cases}$$

as part of the smoothness loss

$$\lambda (|\partial_x d_t| * e^{-|\partial_x I_t|} + |\partial_y d_t| * e^{-|\partial_y I_t|})$$

Calculate gradient of SP-labels

- $|\partial_x \mathbf{SP}|$ & $|\partial_y \mathbf{SP}|$

If $|\partial_{x/y} \mathbf{SP}| \approx 0$:

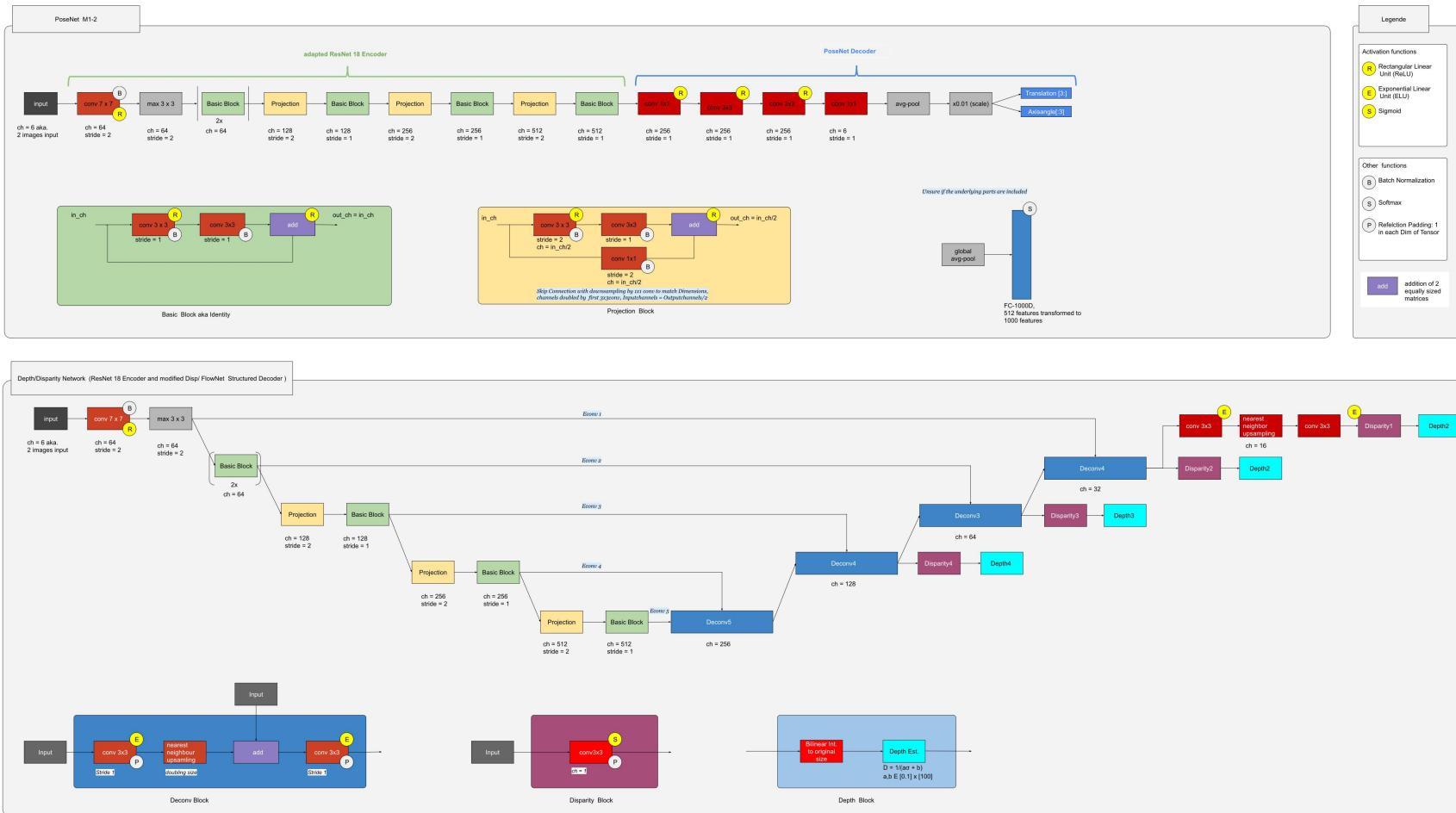
$$|\partial_{x/y} \mathbf{SP}| = e^{-|\partial_{x/y} I_t|}$$

else:

$$|\partial_{x/y} \mathbf{SP}| = 1$$

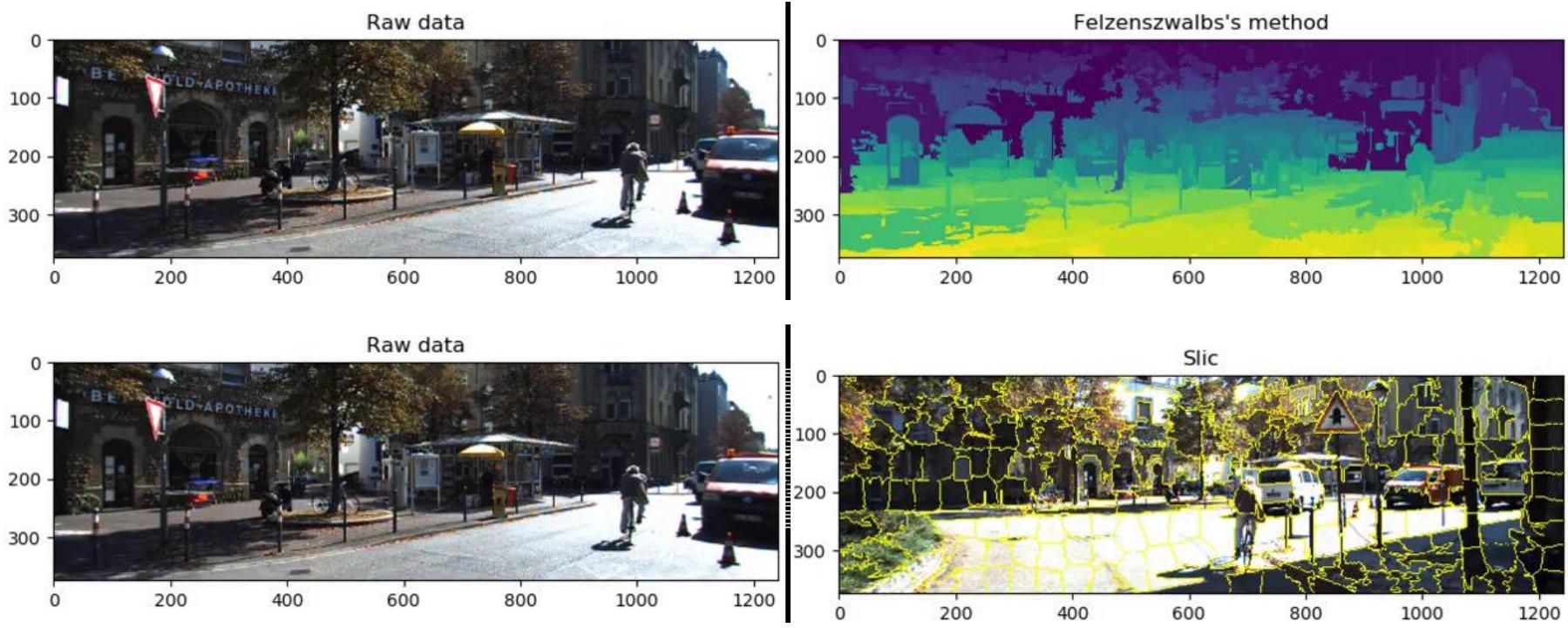
2. Architecture

Detail View: Digging deeper into Monocular View [1]



3. Improvements

Basic I: Superpixel - Demo on KITTI



Conclusion

superpixel can be usefull additional information for the net to approximate depth since they model surfaces and reduce image complexity into a composition of multiple plane

Superpixel Benchmark Parameters

Boundary Recall (Rec)

- is the most commonly used metric to asses boundary adherence given ground truth
- $\text{FN}(G; S)$ be the number of false negative boundary pixels
- $\text{TP}(G; S)$ be the number of true positive boundary pixels

$$\text{Rec}(G, S) = \frac{\text{TP}(G, S)}{\text{TP}(G, S) + \text{FN}(G, S)}.$$

Superpixel Benchmark Parameters



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Undersegmentation Error (UE)

- Measures the “leakage” of superpixels with respect to G
- Different approaches
- Original (Levinshtein et al.) $\text{UE}_{\text{Levin}}(G, S) = \frac{1}{|G|} \sum_{G_i} \frac{\left(\sum_{S_j \cap G_i \neq \emptyset} |S_j| \right) - |G_i|}{|G_i|}$
- Thresholded leakage (van den Bergh et al. and Neubert and Protzel)

$$\text{UE}_{\text{Bergh}}(G, S) = \frac{1}{N} \sum_{S_j} |S_j - \arg \max_{G_i} |S_j \cap G_i||,$$

$$\text{UE}_{\text{NP}}(G, S) = \frac{1}{N} \sum_{G_i} \sum_{S_j \cap G_i \neq \emptyset} \min\{|S_j \cap G_i|, |S_j - G_i|\},$$

Superpixel Benchmark Parameters



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Explained Variation (EV)

- quantifies the quality of a superpixel segmentation without relying on ground truth
- $\mu(S_j)$ and $\mu(I)$ are the mean color of superpixel S_j and the image I

$$EV(S) = \frac{\sum_{S_j} |S_j|(\mu(S_j) - \mu(I))^2}{\sum_{x_n} (I(x_n) - \mu(I))^2}$$

Superpixel Benchmark Parameters



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Compactness (CO)

- compares the area $A(S_j)$ of each superpixel S_j with the area of a circle (the most compact 2-dimensional shape) with same perimeter $P(S_j)$, i.e. higher is better.

$$\text{CO}(G, S) = \frac{1}{N} \sum_{S_j} |S_j| \frac{4\pi A(S_j)}{P(S_j)}.$$

Benchmark of depth estimation Algorithm some commonly used error metrics:



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Error Metrics

Threshold: percentage of y such that $\max\left(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}\right) = \sigma < thr$

Absolute relative difference: $rel = \frac{1}{T} \sum_{i,j} |y_{i,j} - y_{i,j}^*| / y_{i,j}^*$

Squared relative difference: $srel = \frac{1}{T} \sum_{i,j} |y_{i,j} - y_{i,j}^*|^2 / y_{i,j}^*$

RMS (linear): $RMS = \sqrt{\frac{1}{T} \sum_{i,j} |y_{i,j} - y_{i,j}^*|^2}$

RMS (log): $\log_{10} = \sqrt{\frac{1}{T} \sum_{i,j} |\log y_{i,j} - \log y_{i,j}^*|^2}$

With Y = predicted depth map, Y^* = ground truth depth image and T = pixel depth

Groundtruh

- For example outdoor datasets (High depth range / less depth resolution)
 - [KITTI](#) – Depth by LIDAR
 - [Make3D](#) – Depth by LIDAR

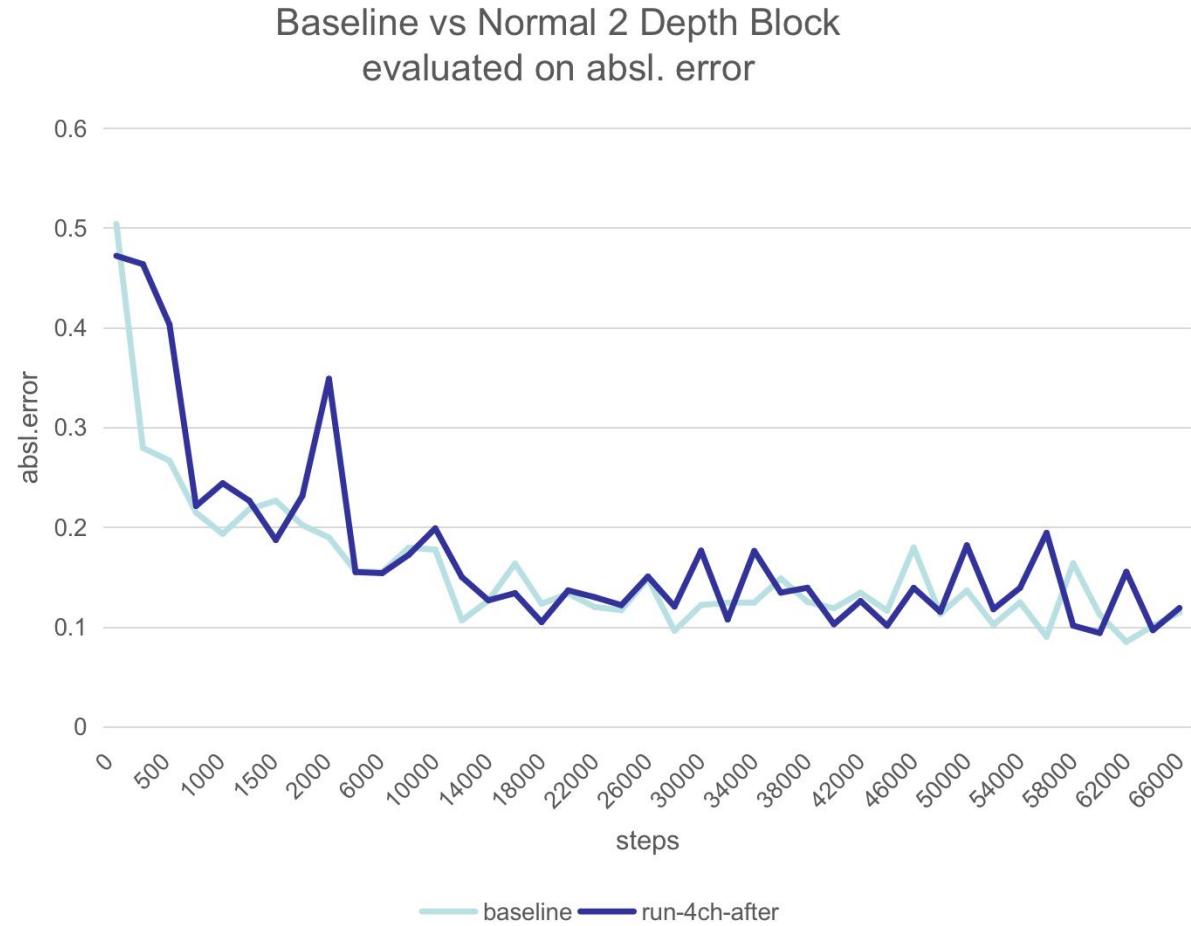
3. Improvements

Iteration I: Normal 2 Depth Block



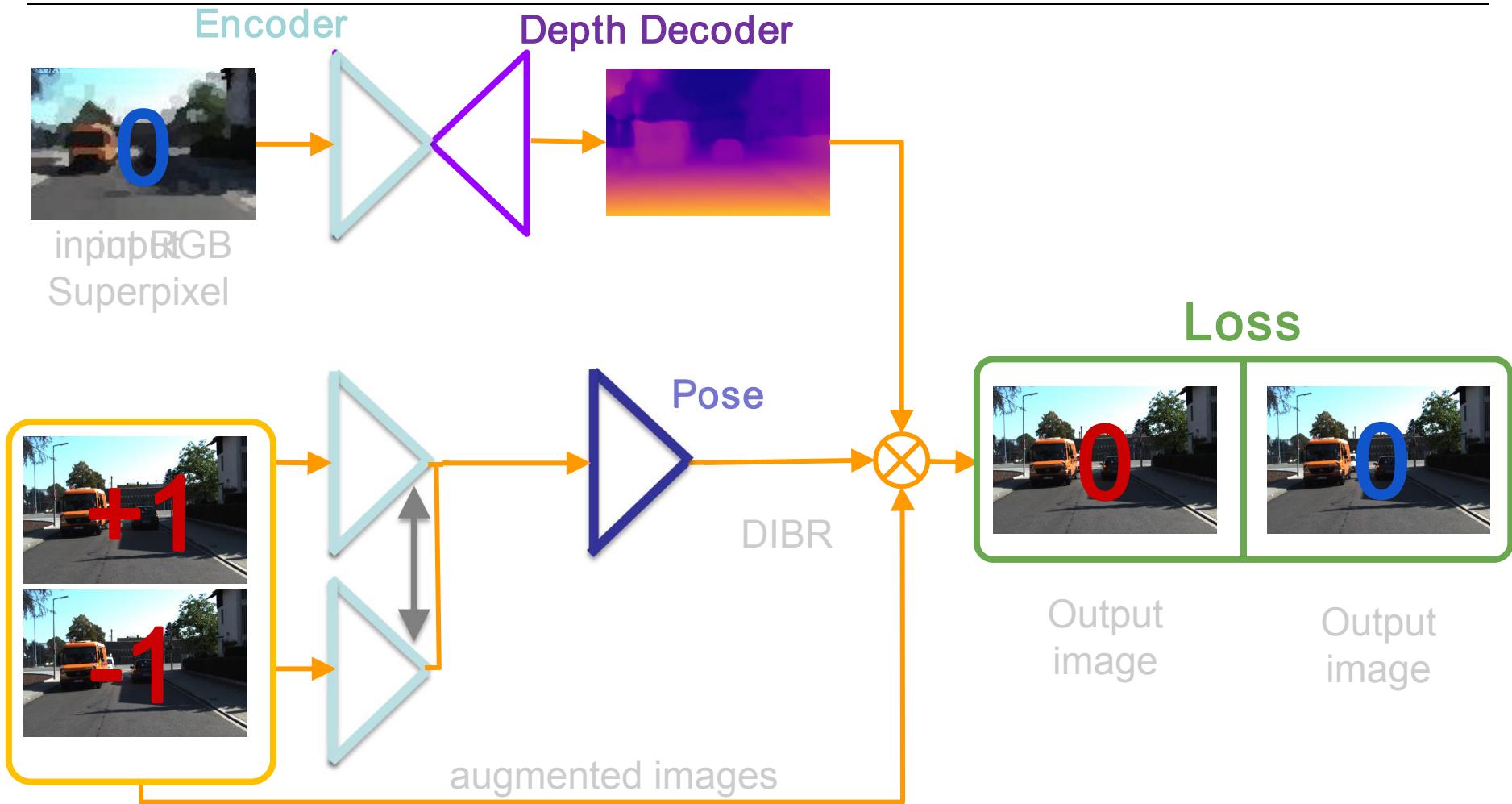
4 CH Input

superpixel can be useful additional information for the net to approximate depth since they model surfaces and reduce image complexity into a composition of multiple plane



5. Architecture

New Input: RGB

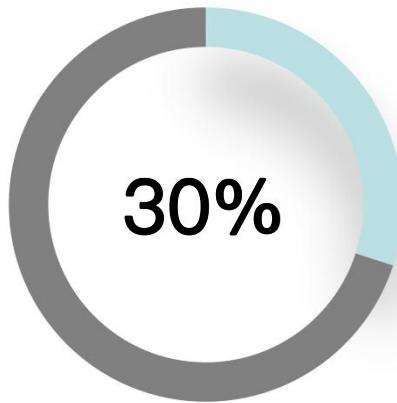


4. Results

Focus on three significant approaches

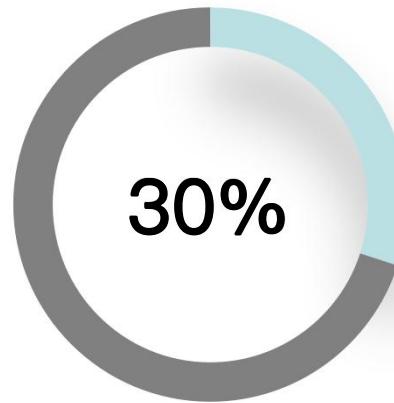


TECHNISCHE
UNIVERSITÄT
DARMSTADT



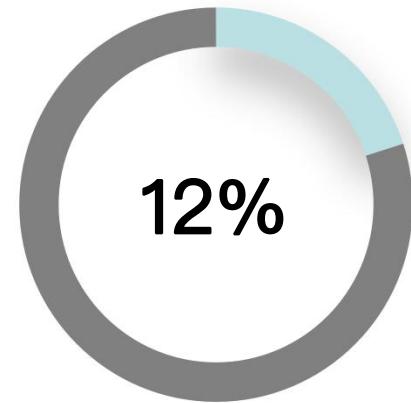
Data title here

A peep at some distant orb has power to raise and purify our thoughts like a strain of sacred music



Data title here

A peep at some distant orb has power to raise and purify our thoughts like a strain of sacred music



Data title here

A peep at some distant orb has power to raise and purify our thoughts like a strain of sacred music

3. Improvements

Improvement normals with Superpixel



TECHNISCHE
UNIVERSITÄT
DARMSTADT

use Superpixel information for uniforming normals

Goal: ensure sharp edges of surface in depth map

in architecture

in loss

Timeline Title

A peep at some distant orb has
power to raise and purify our thoughts
like a strain.

Timeline Title

A peep at some distant orb has
power to raise and purify our thoughts
like a strain.

