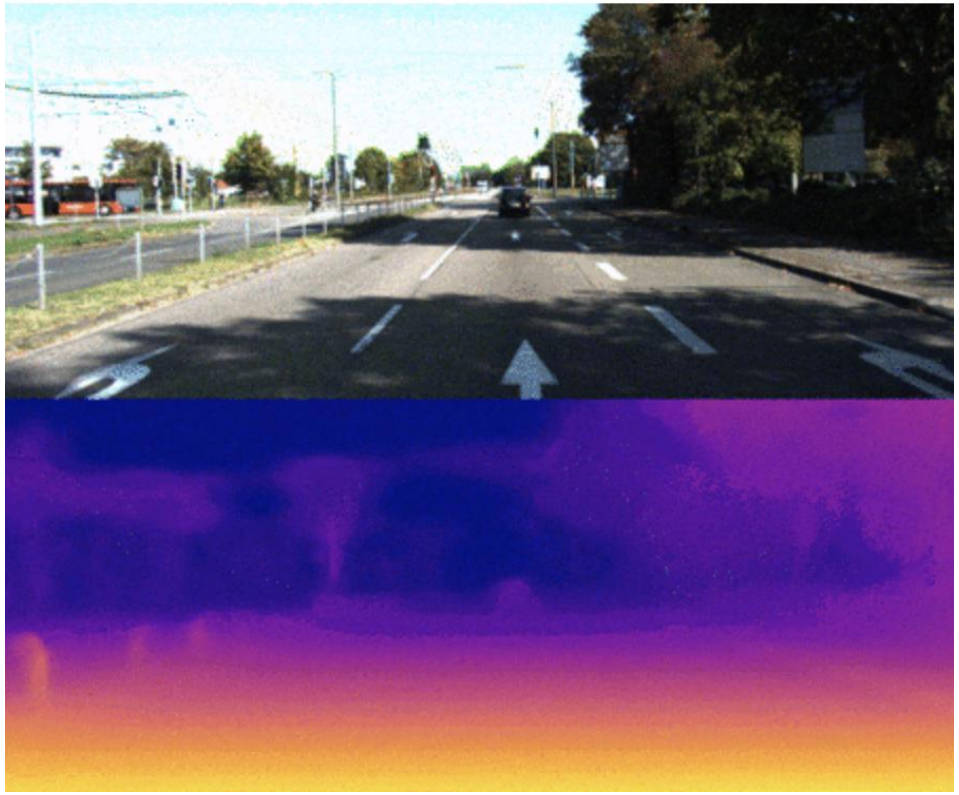
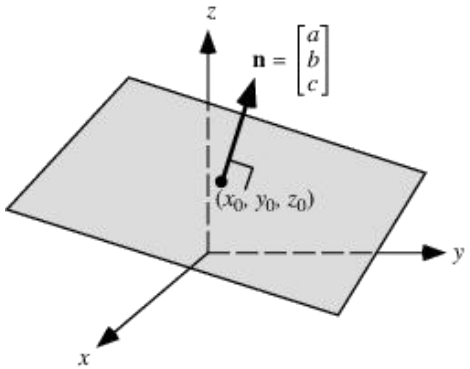
Final Presentation:
Piecewise monocular depth estimation by plane fitting

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

TECHNISCHE UNIVERSITÄT DARMSTADT





visual inference

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

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

1. Overview incl. Goal

TECHNISCHE UNIVERSITÄT DARMSTADT

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* [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

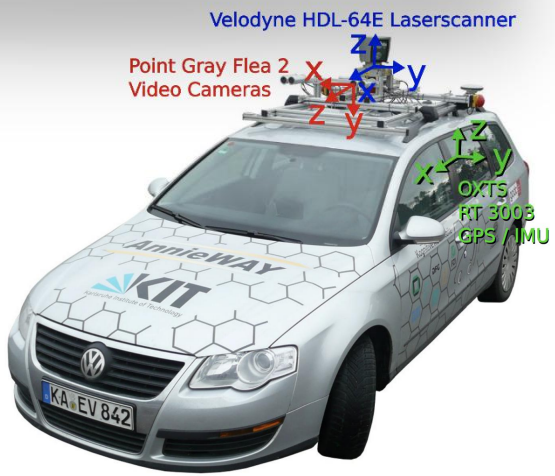
Dataset: KITTI Dataset and Eigensplit

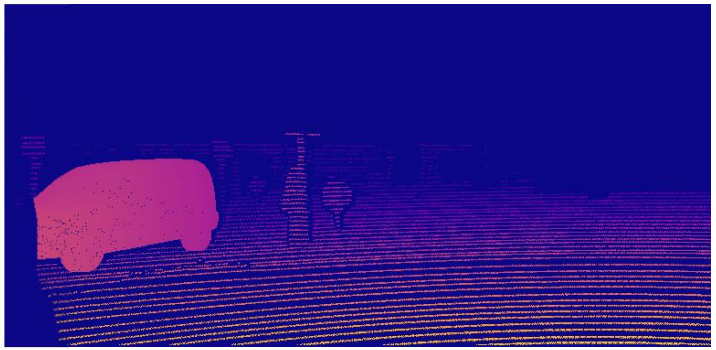
TECHNISCHE UNIVERSITÄT DARMSTADT

KITTI DATASET [2]

2 high - resolution color and grayscale video cameras

Velodyne laser scanner





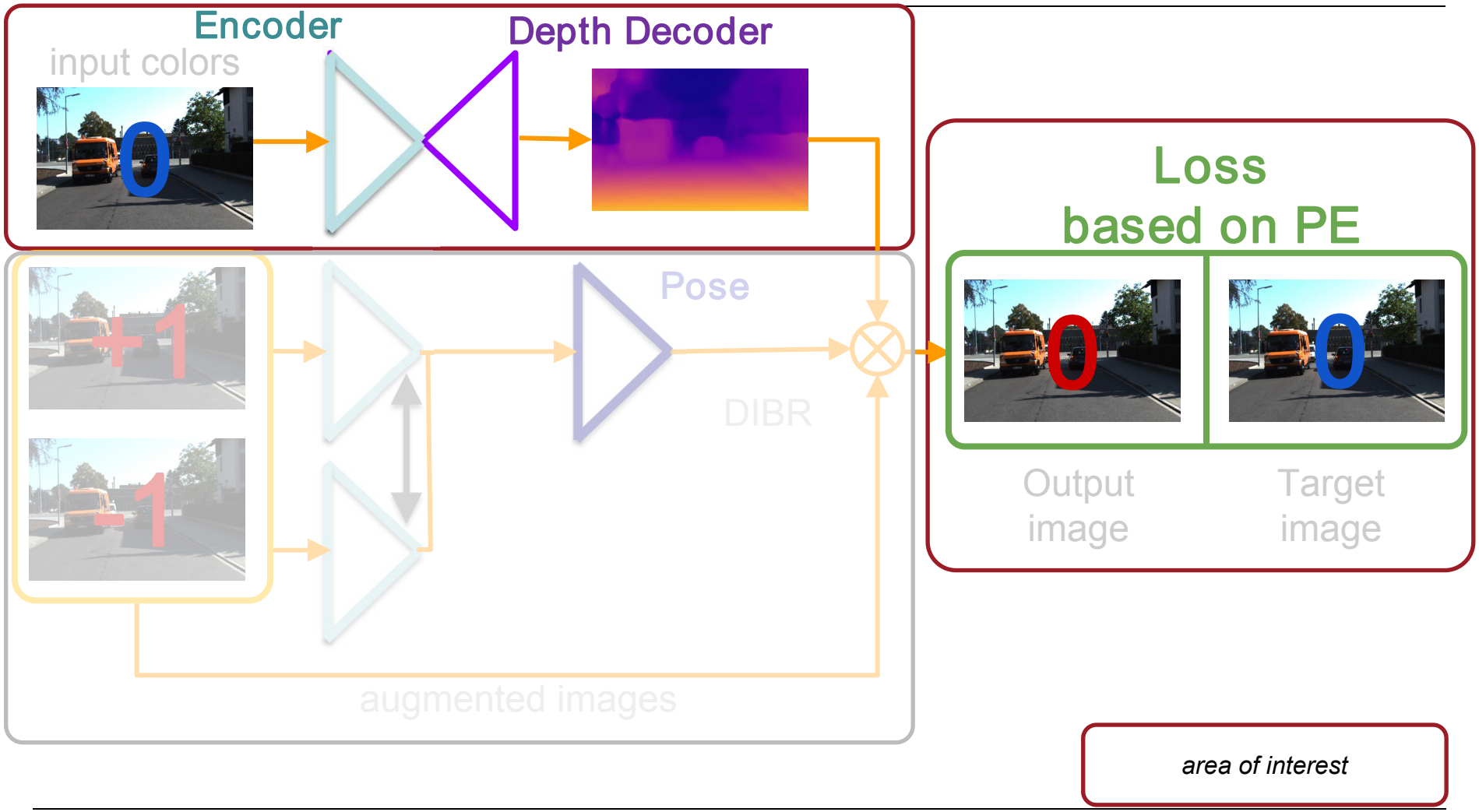
EIGEN ZHOU [3]

- 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**

visual inference

2. Architecture

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

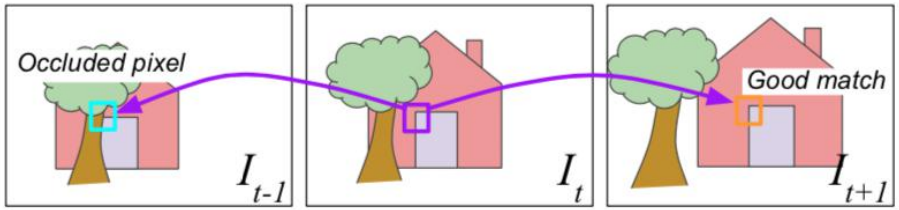


2. Architecture
Loss: Digging Into Self-Supervised Monocular Depth Estimation [1]

TECHNISCHE UNIVERSITÄT DARMSTADT

Baseline – Loss

Minimum Photometric Error



L_p: Calculated Error between image and warped image based on L1-Norm & SSIM

$$L = \underbrace{\mu \min\left(PE\left(I_t, I_{t\to t'}\right)\right)}_{L_p} + \underbrace{\lambda \left(\left|\partial_x d_t\right| * e^{\{-|\partial_x I_t|\}} + \left|\partial_y d_t\right| * e^{\{-|\partial_y I_t|\}} \right)}_{L_s}$$

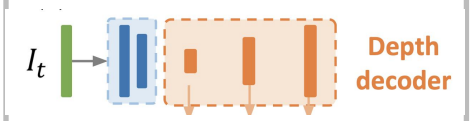
L_s :Ensures smooth gradient expect on edges

visual inference

3. Improvements
3 pillar focus

TECHNISCHE UNIVERSITÄT DARMSTADT

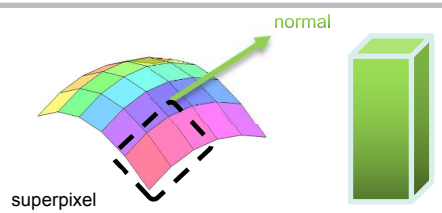
Input



change input channels (I_t)

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

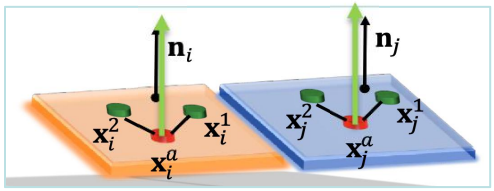
Architecture



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

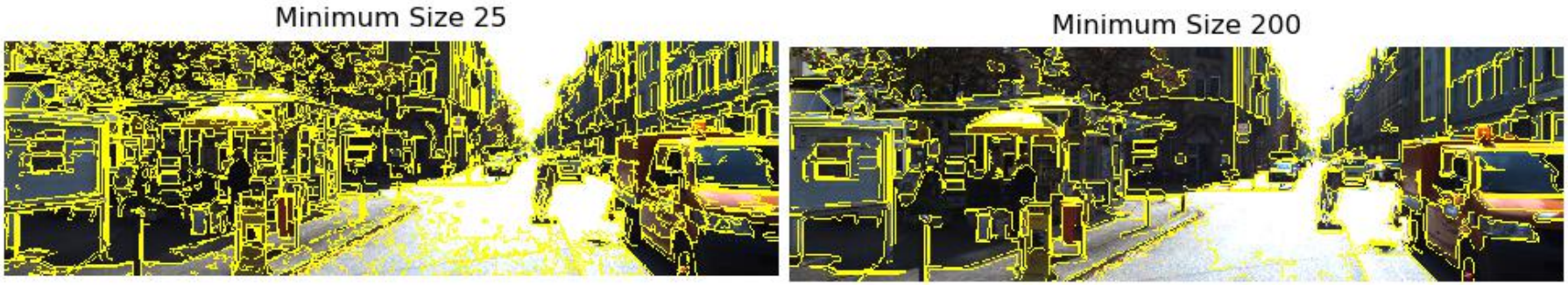
visual inference

3. Improvements
Basic I: Superpixel

TECHNISCHE UNIVERSITÄT DARMSTADT

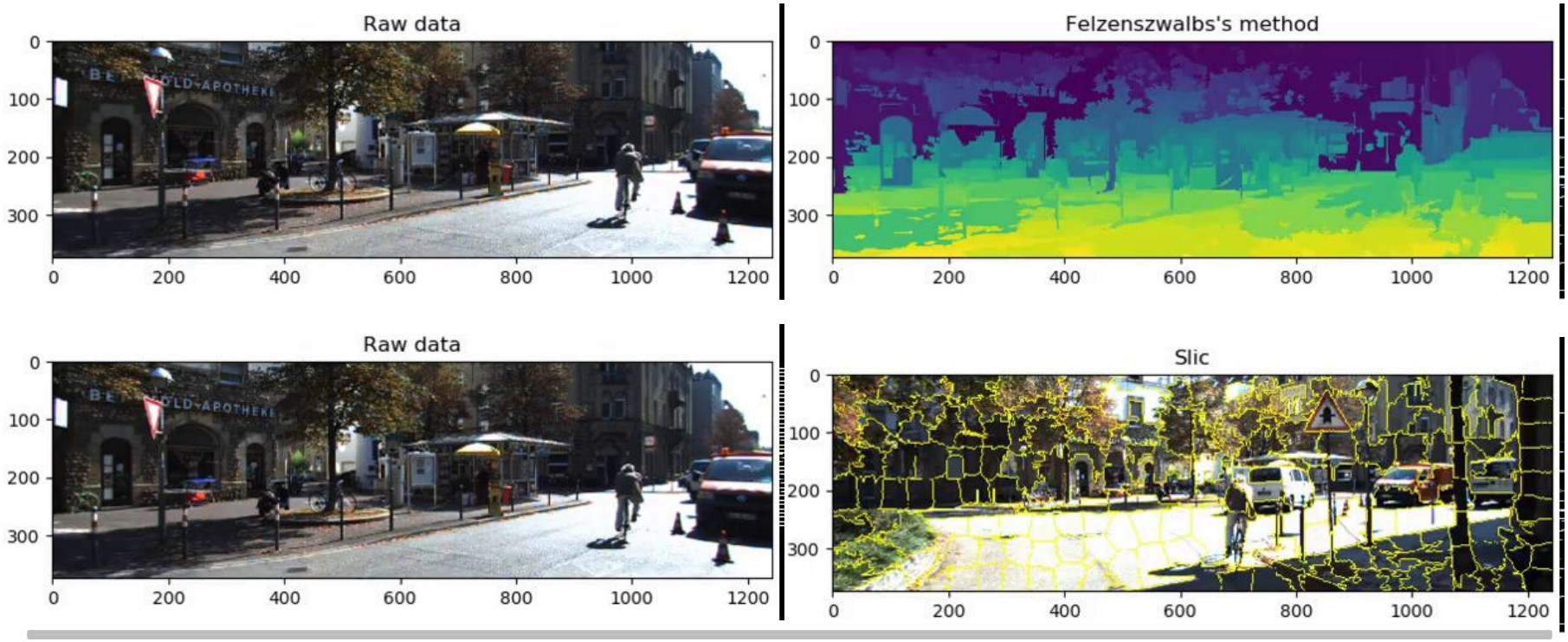
- Def: superpixel can be defined as a group of pixels that share common characteristics (e.g. pixel intensity)
- Huge variety exists catgegorized in Graph-, Clustering-, Watershed-, Density-, Contour-, Energy- and Wavelet-based
- Benchmarks exists [4] [5], evaluate on criteria like: Boundary Recall, Undersegmentation Error, Explained Variation, Compactness

current favorite: **Felzenshalb** (Scale, Sigma, Mimimum Size)



visual inference

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

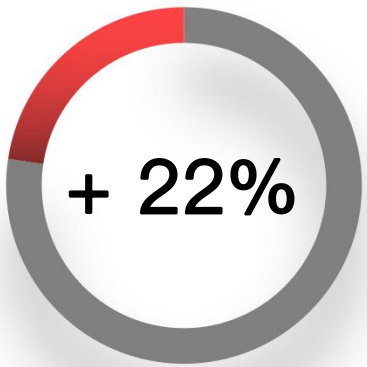
3. Improvements
Iteration I: 4 Channel Input (Superpixel & RGB)

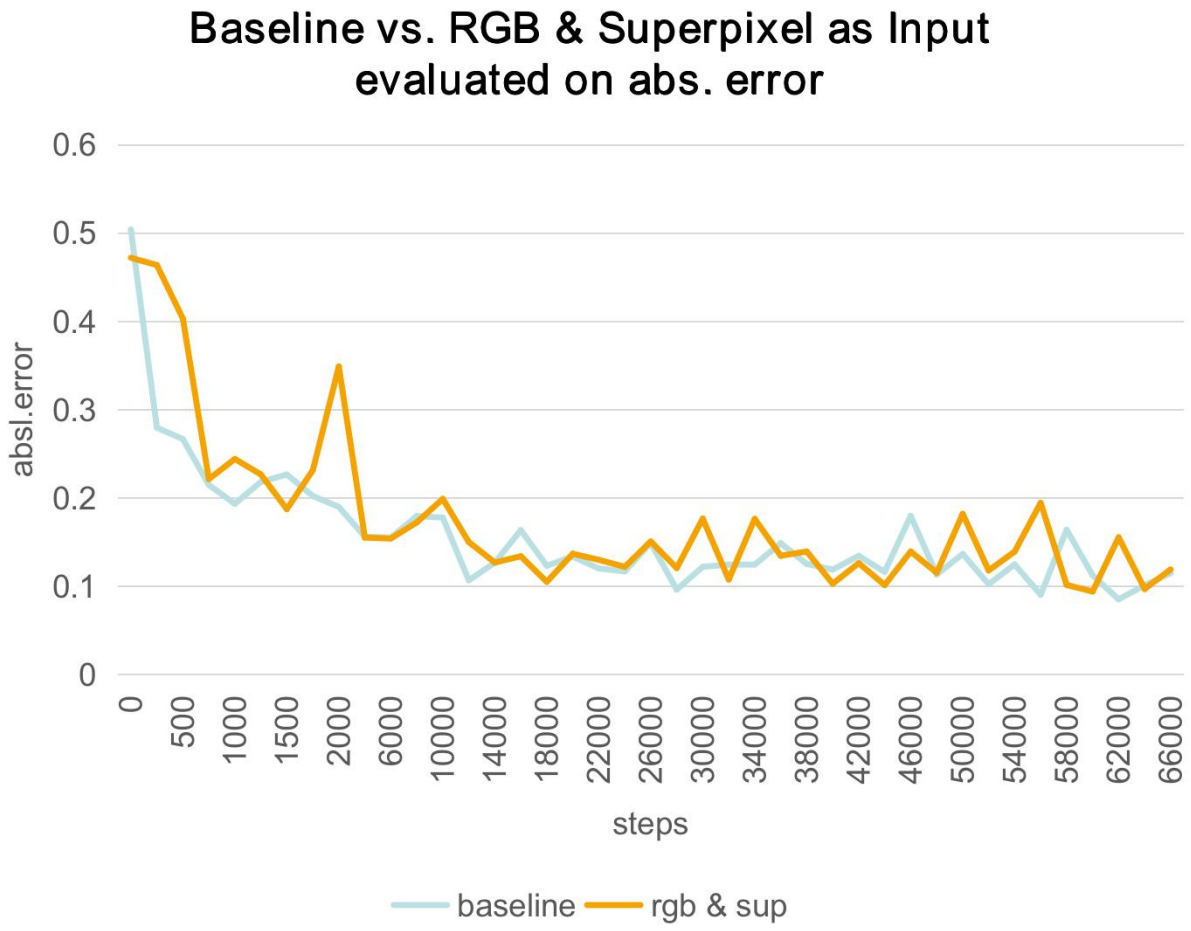
TECHNISCHE UNIVERSITÄT DARMSTADT

Abs. rel. Error

Baseline: 0.115

Current: 0.141

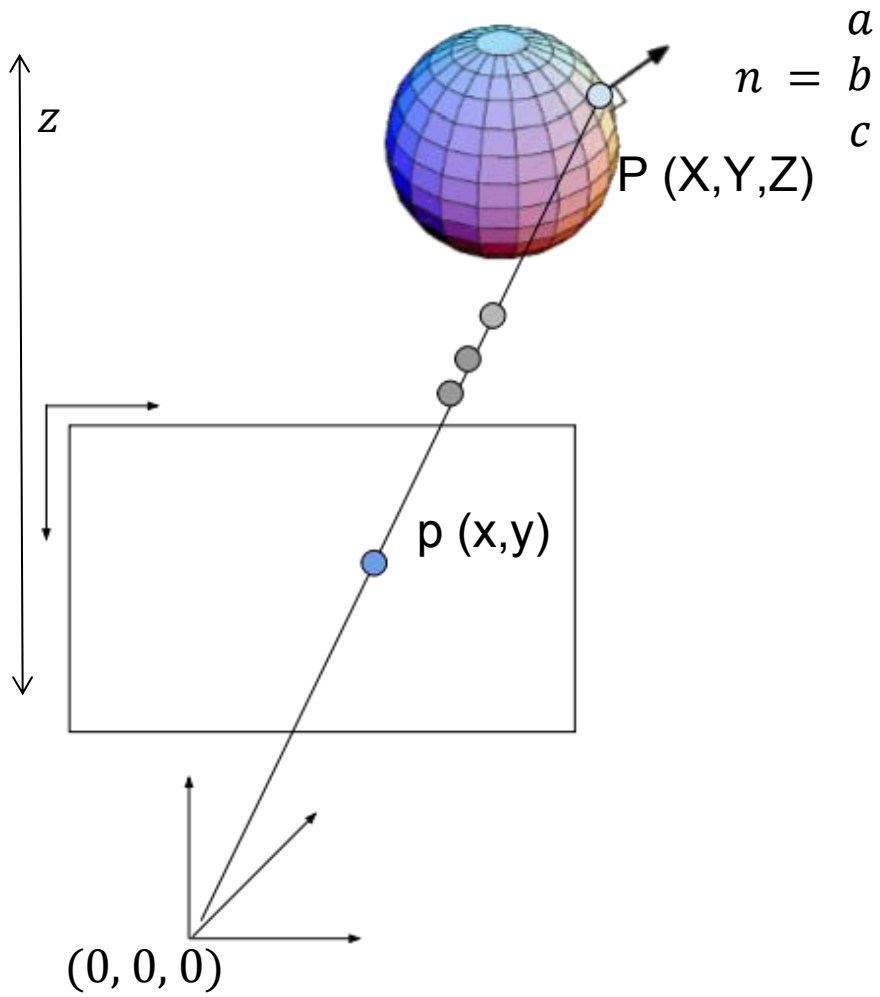




visual inference

3. Improvements

Basic II: Normal Computation



normal to disparity

1 $p = K\ P$

$P = K^{-1}\ p$

2 $n^T\ P = 1$

Depth Estimation

3 $P = (K^{-1}\ p)_{homo}\ z$

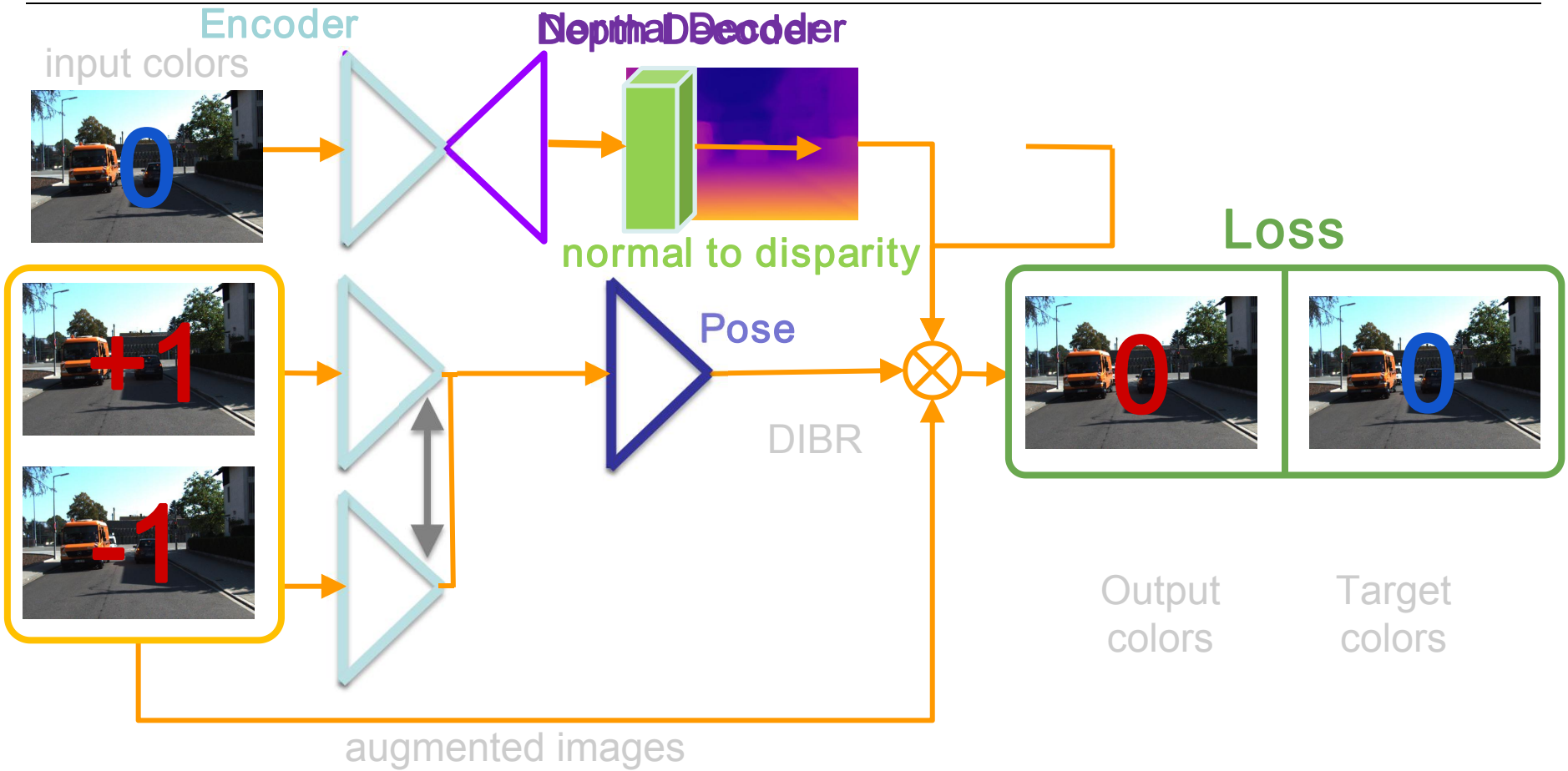
$z = \frac{1}{n^T\ K^{-1}\ p}$

Disparity

4 $\hat{d} = \frac{f\ b}{z}$

3. Improvements

Iteration II: Introducing N2D Block



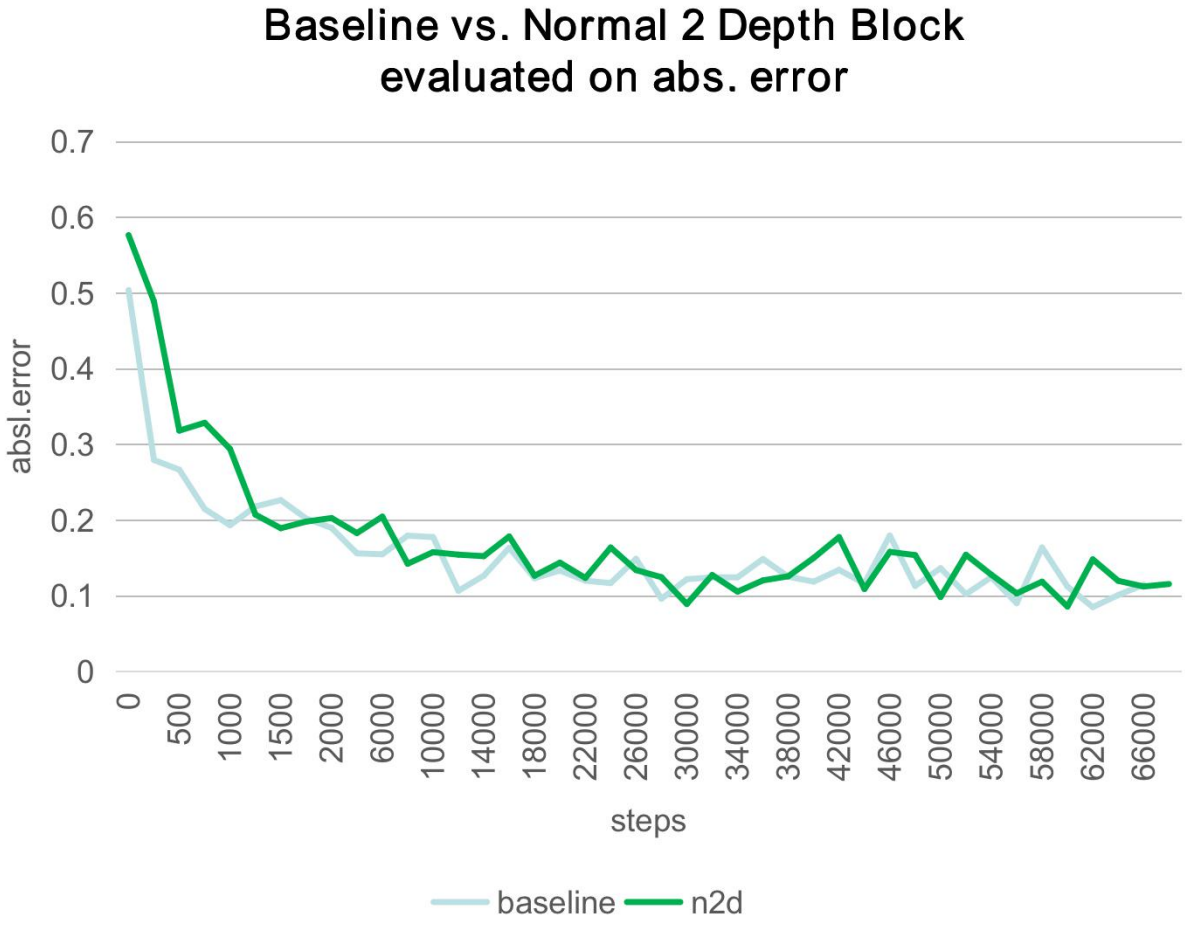
3. Improvements
Iteration II: Normal 2 Depth Block

TECHNISCHE UNIVERSITÄT DARMSTADT

Abs. rel. Error

Baseline: 0.115

Current: 0.123



visual inference

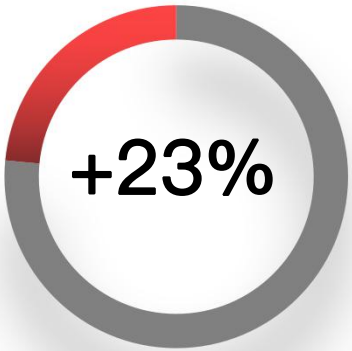
3. Improvements
Iteration II: 4 Ch + N2D Block

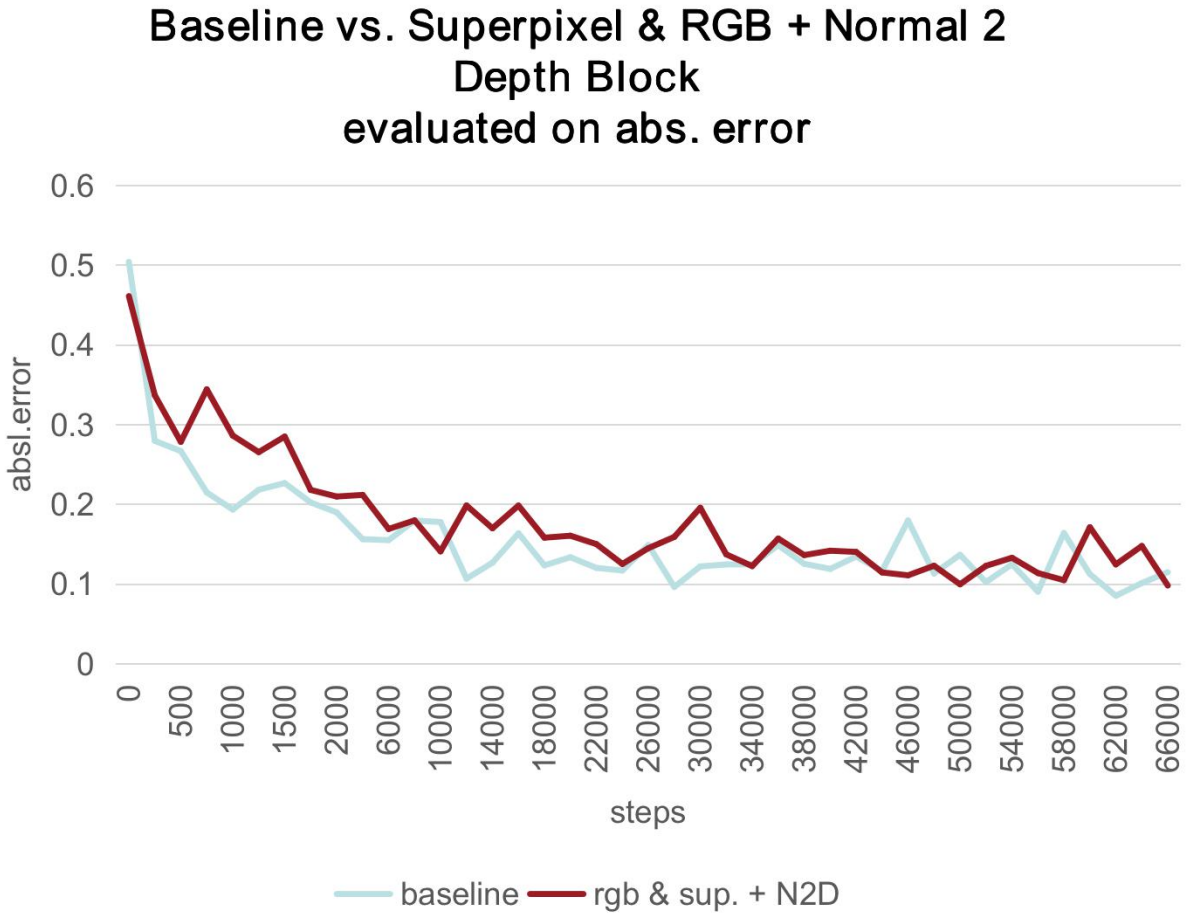
TECHNISCHE UNIVERSITÄT DARMSTADT

Abs. rel. Error

Baseline: 0.115

Current 0.142

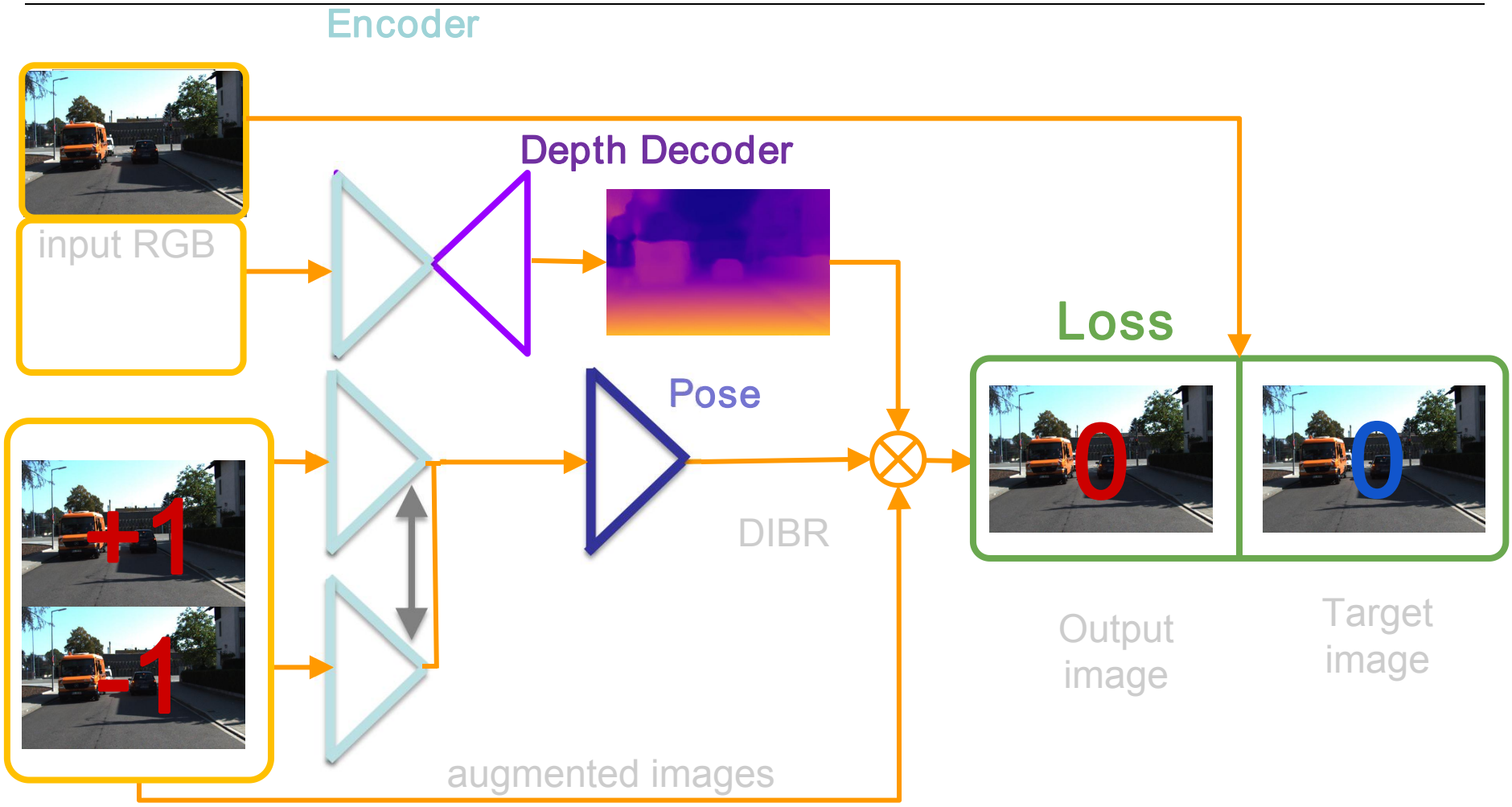




visual inference

3 . Improvements

Iteration III: Superpixel in Loss Function

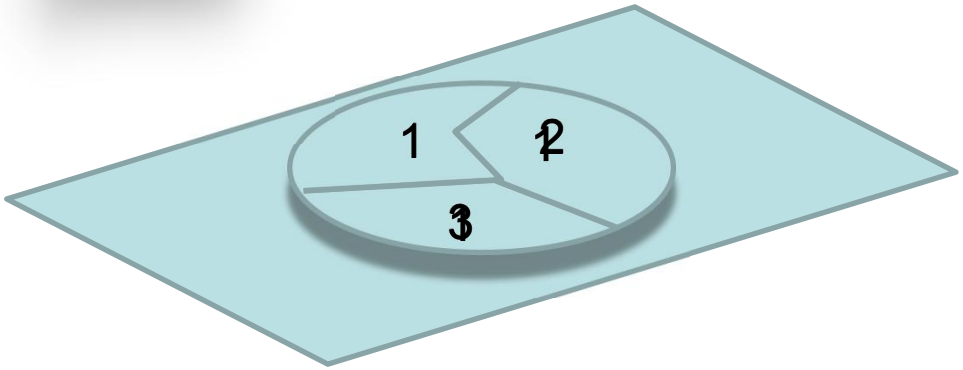


3 . Improvements

Iteration III: Superpixel in Loss Function

TECHNISCHE UNIVERSITÄT DARMSTADT

Concept Idea



use Superpixel information in implementing loss function

Problem: Oversegmentation:
- one plane is represented multiple superpixel segements

		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

visual inference

3 . Improvements

Iteration III: Binary & Continuous Superpixel Loss

TECHNISCHE UNIVERSITÄT DARMSTADT

Mathematical Implementation Superpixel – Loss

visual inference

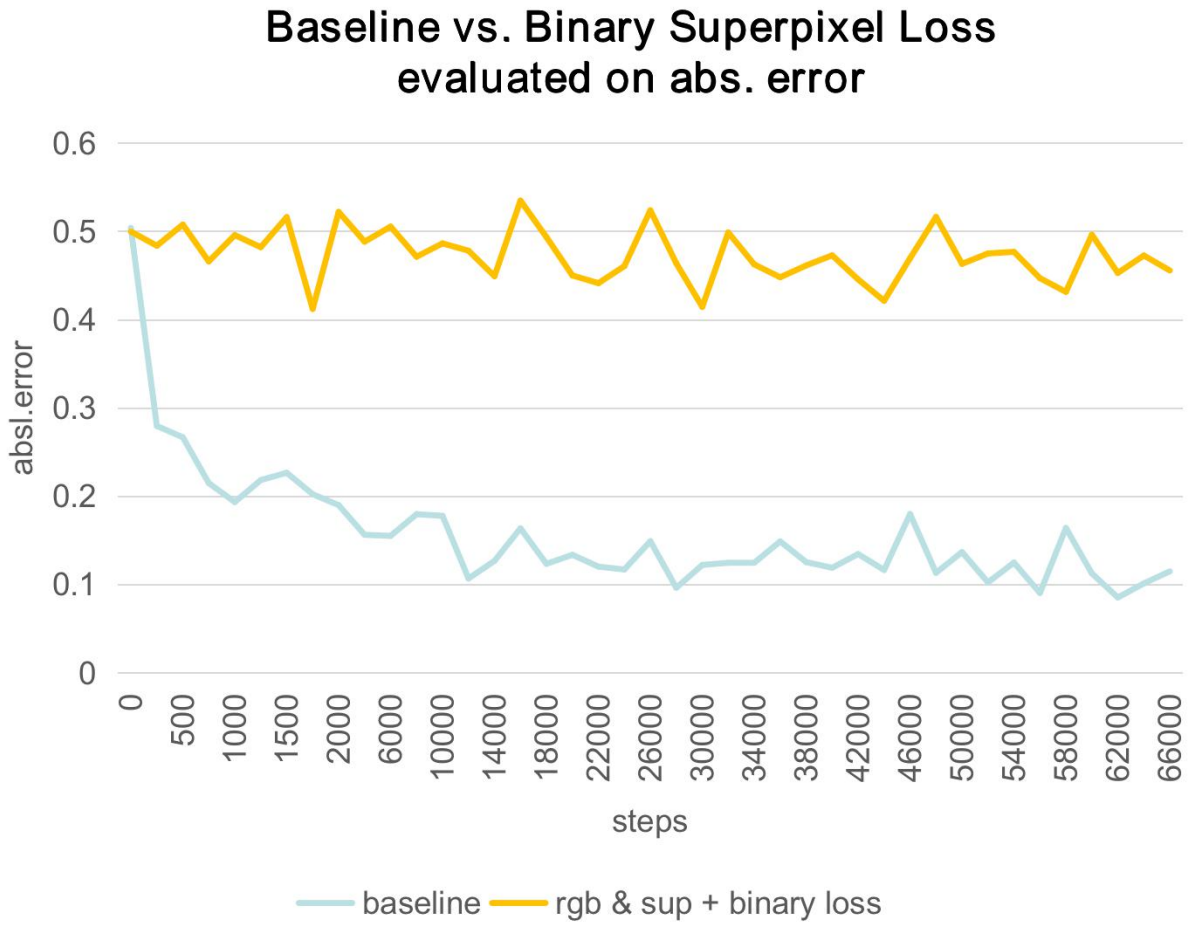
3. Improvements
Iteration III: Binary Superpixel Loss

TECHNISCHE UNIVERSITÄT DARMSTADT

Abs. rel. Error

Baseline: 0.115

Current: 0.443



visual inference

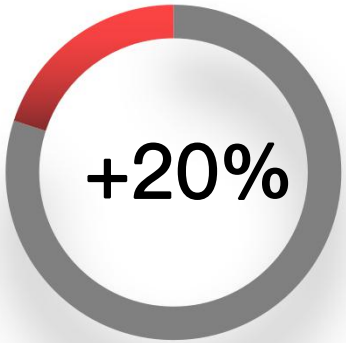
3. Improvements
Iteration III: Continuous Superpixel Loss

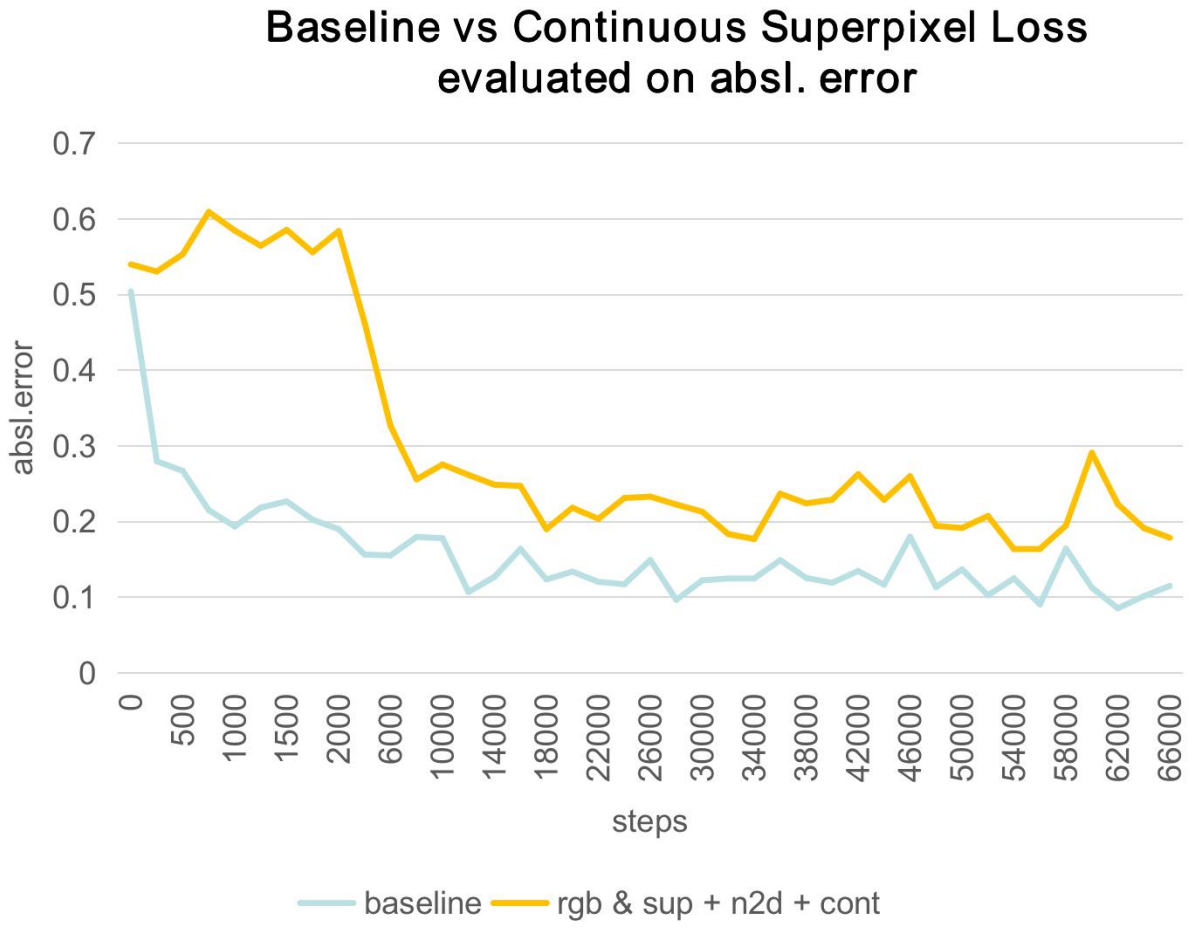
TECHNISCHE UNIVERSITÄT DARMSTADT

Abs. rel. Error

Baseline: 0.115

Current: 0.138

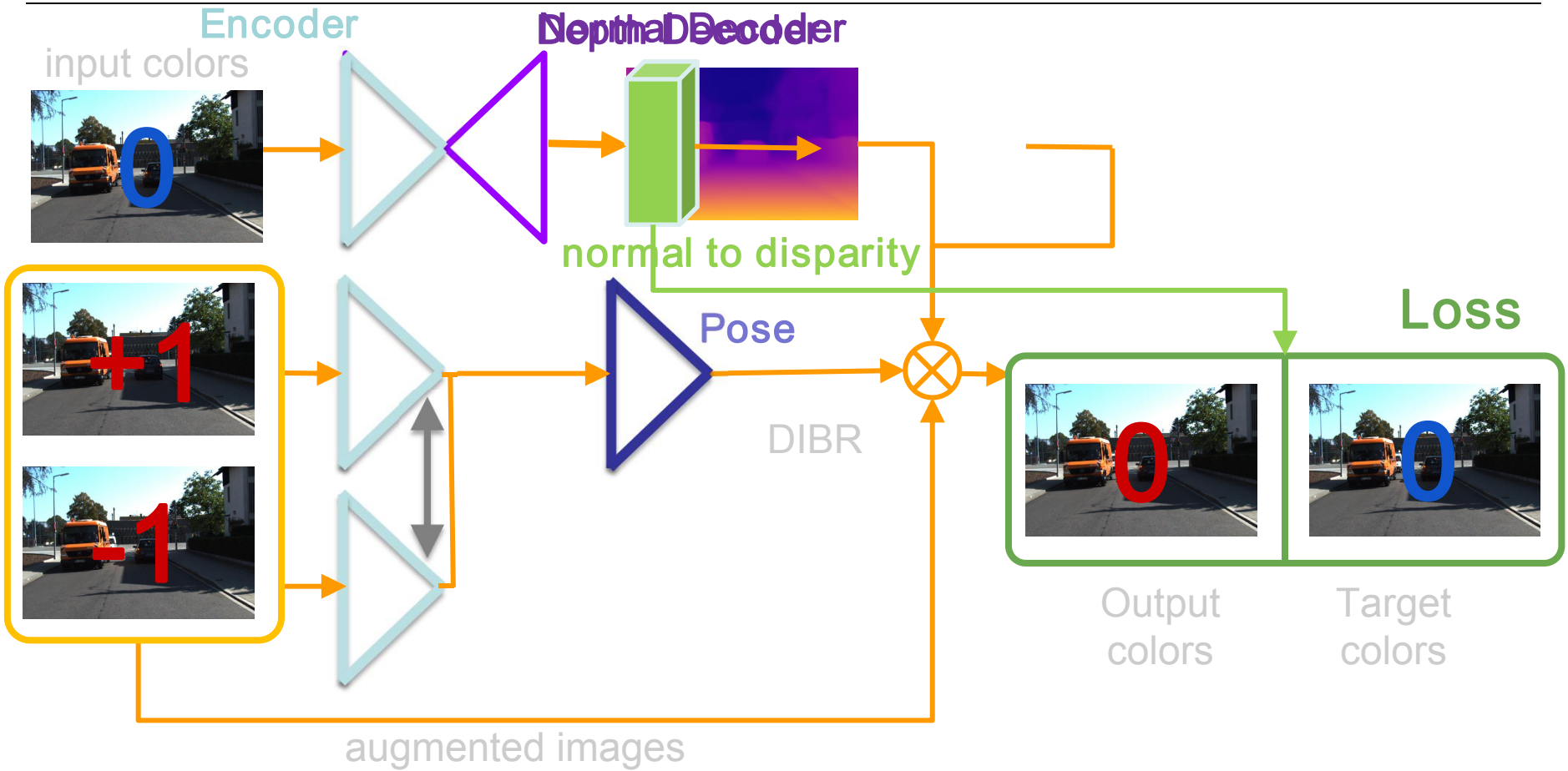




visual inference

3. Improvements

Iteration IV: Normals in Loss Function

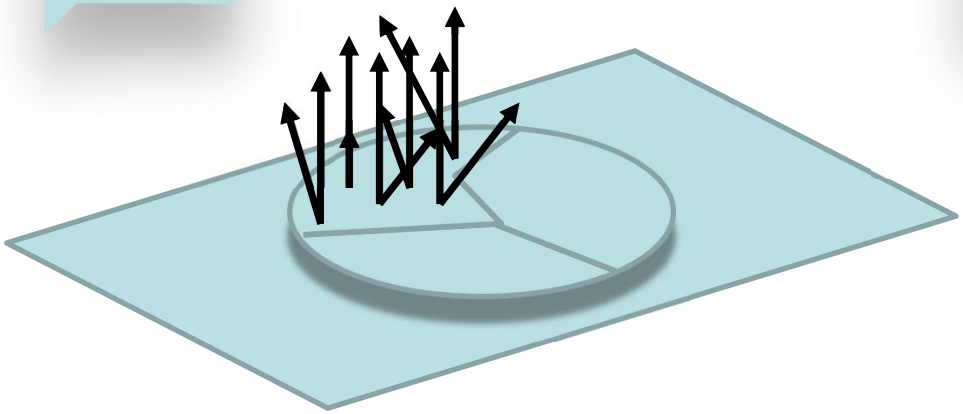


Loss Function
Normals in Loss Function

TECHNISCHE UNIVERSITÄT DARMSTADT

Concept Idea

Implementation



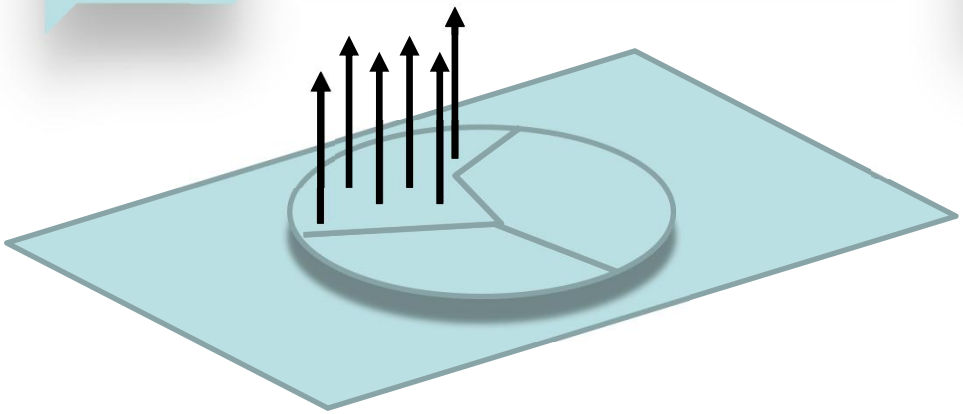
- Force normals to be equal inside SPs
- Treat superpixels as planes represented by one normal vector

visual inference

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*

visual inference

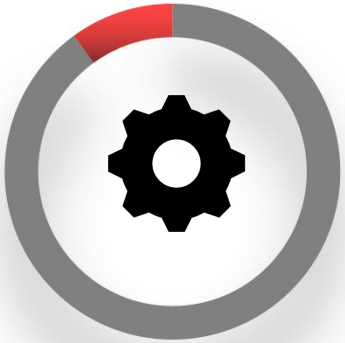
3. Improvements
Iteration IV: Normals in Loss Function

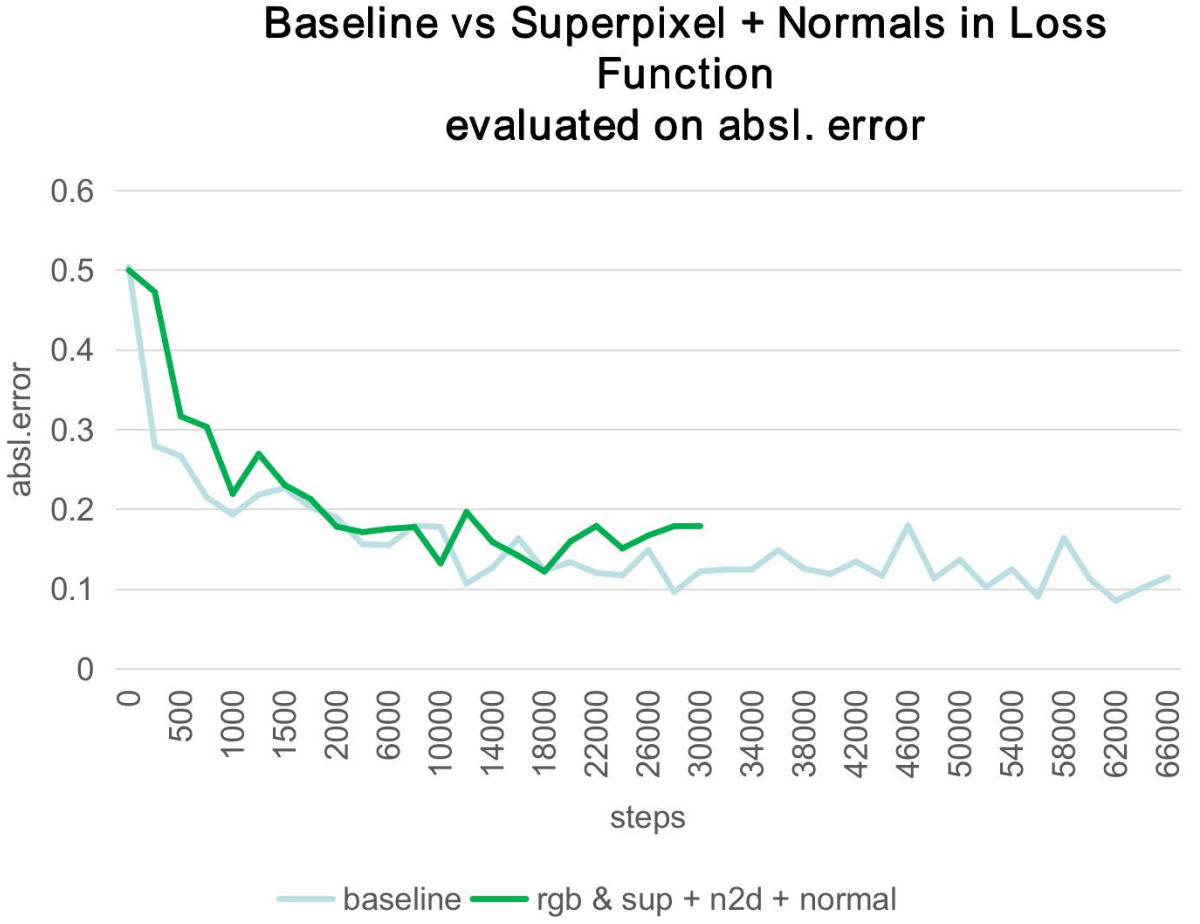
TECHNISCHE UNIVERSITÄT DARMSTADT

Abs. rel. Error

Baseline: 0.115

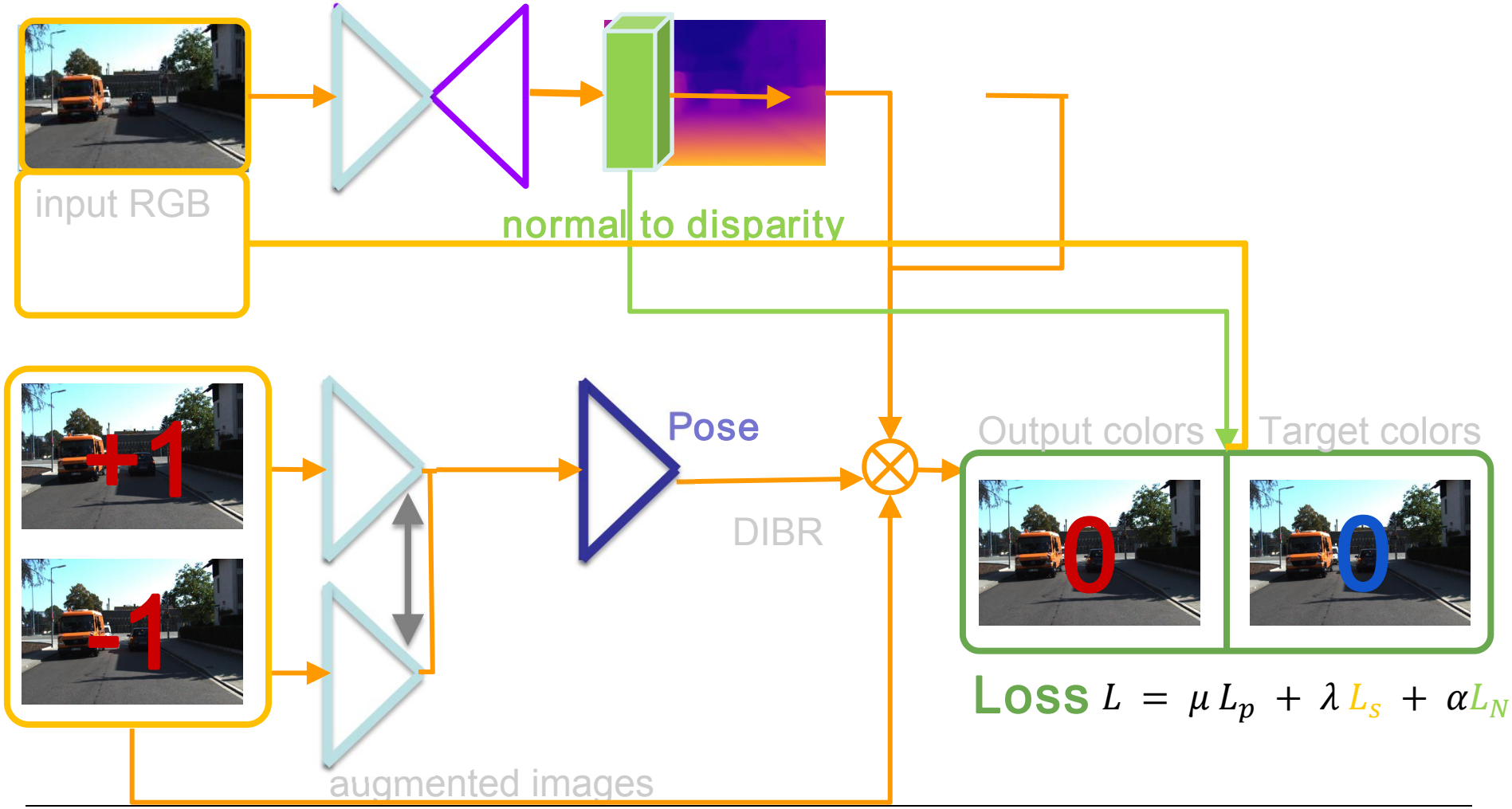
Current: still running





visual inference

3. Improvements
Iteration V: Normal Loss & Continuous Superpixel Loss



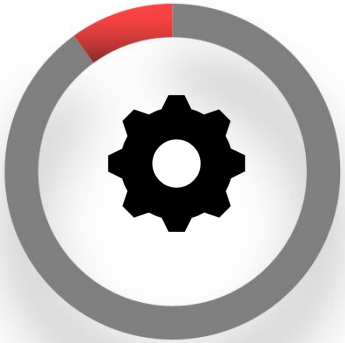
3. Improvements
Iteration V: Normal Loss & Continuous Superpixel Loss

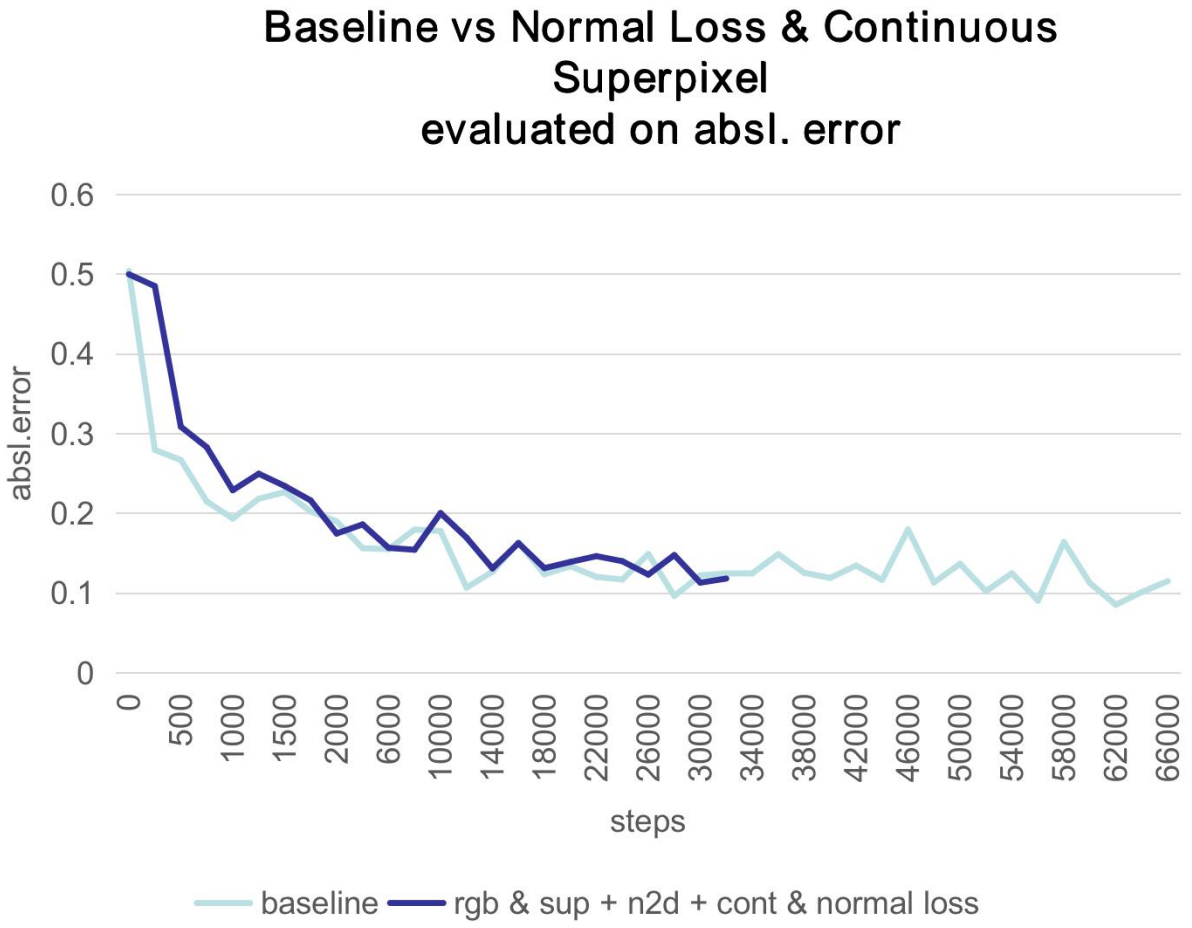
TECHNISCHE UNIVERSITÄT DARMSTADT

Abs. rel. Error

Baseline: 0.115

Current: still running





visual inference

4. Results

All Approaches

TECHNISCHE UNIVERSITÄT DARMSTADT

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

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	

visual inference

4. Results

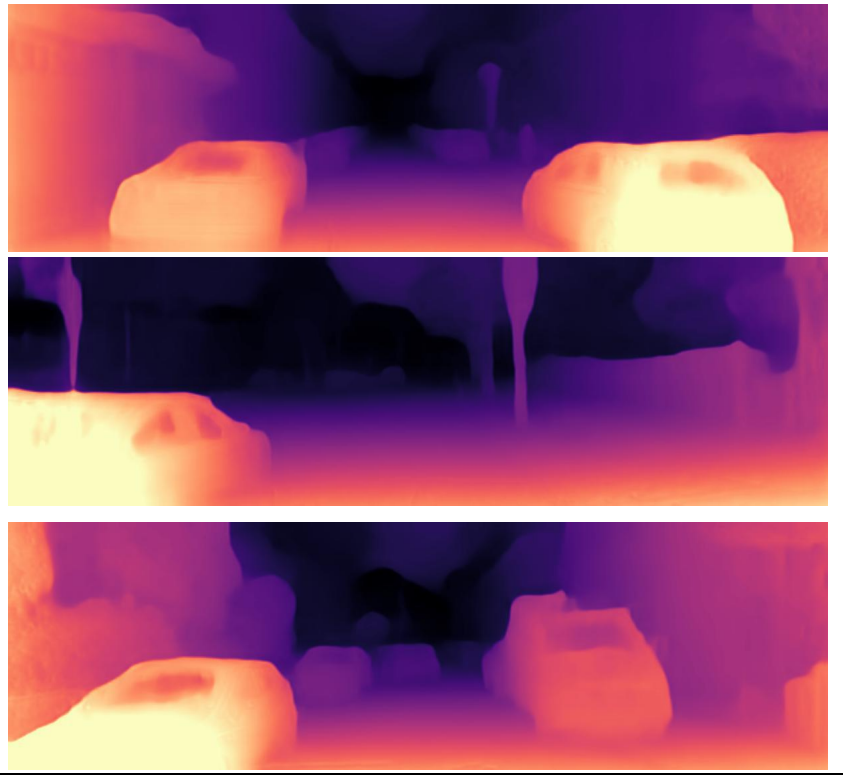
Demo I Compare Results

TECHNISCHE UNIVERSITÄT DARMSTADT

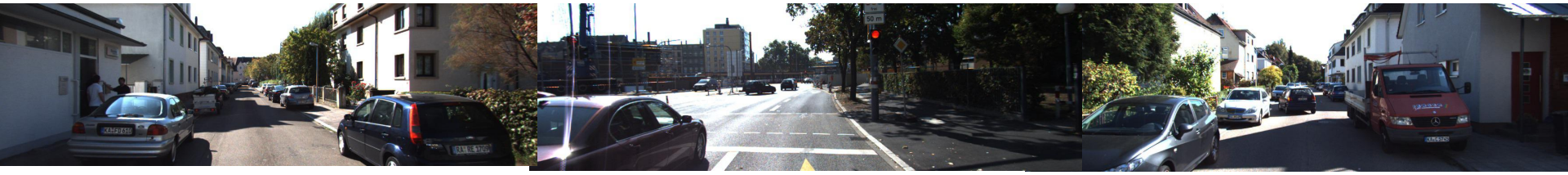
Evaluation of our result based on three test images

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

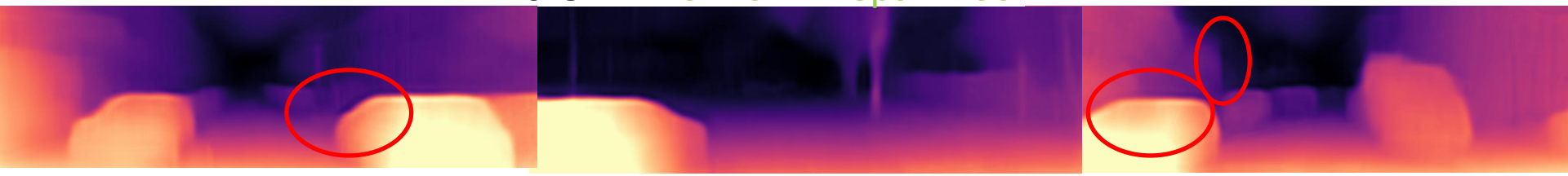
Baseline



visual inference



3 Ch + Normal 2 Depth Block



4 Ch + Normal 2 Depth Bloc + Superpixel Cotinuous Loss

4 Ch (RGB + SP) + Normal 2 Depth Block

4 ch + Normal Decoder

Conclusion

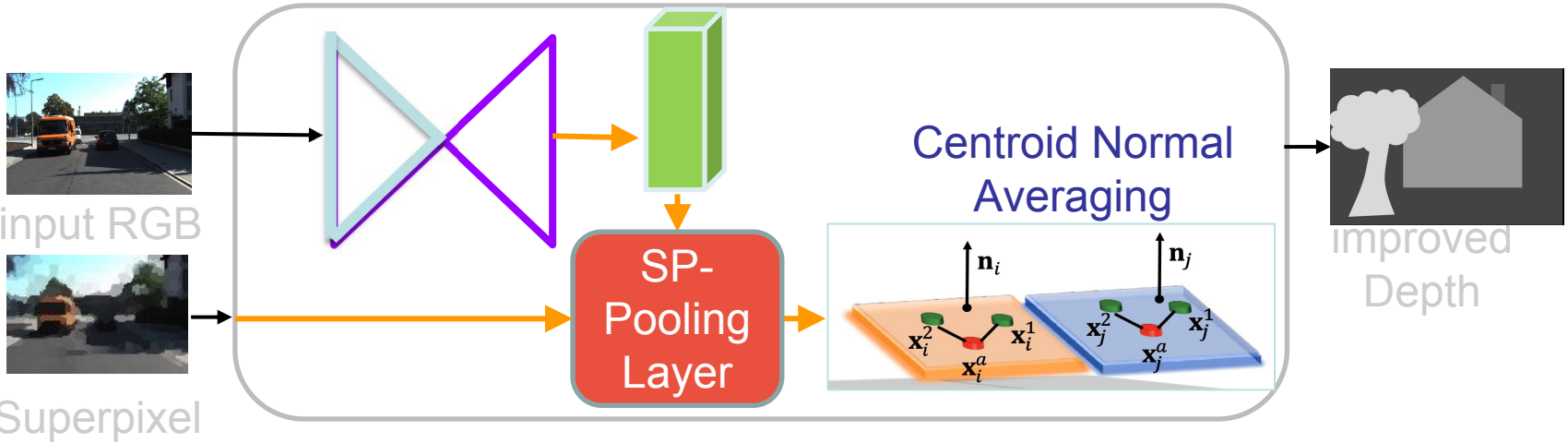
1. Depth Maps are far from perfect in general (e.g. missing shields)
2. Normal Approach shows windows as one plane
3. Building border tend to be sharper
4. Superpixel and Continuous Loss show most promising loss results - final depth maps are still pending unfortunately

4 ch + Normal Decoder

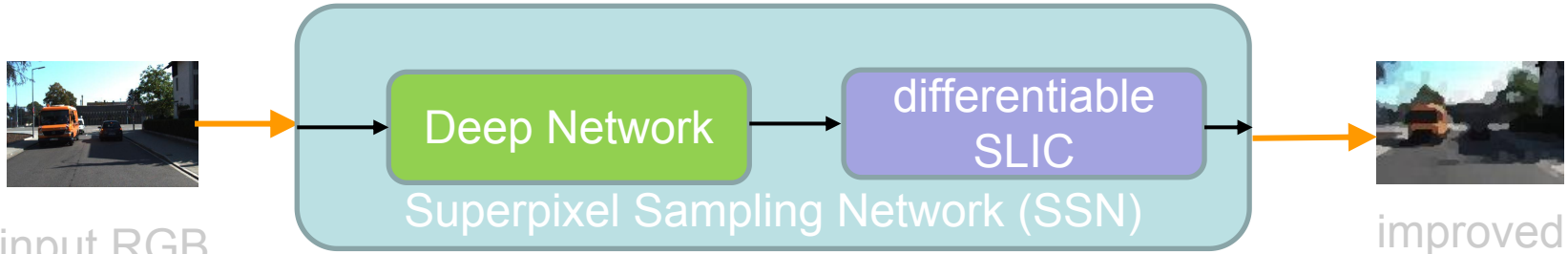
5. Further approaches

Improvement of Loss

1. Use centroid information for superface surface modifikation



2. Learn Superpixel with hyperparameter an learning part itself



6. Review

TECHNISCHE UNIVERSITÄT DARMSTADT

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

visual inference

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

TECHNISCHE UNIVERSITÄT DARMSTADT

- Net Detail View
- Superpixel Benchmark Paramter
- Error Metrics

visual inference

2. Architecture

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

TECHNISCHE UNIVERSITÄT DARMSTADT

Photometric Error L_p

- SSIM
- L1- Norm

Depth

Pose

$$I_{t\to t'} = I\_t < proj(\boldsymbol{D_t}, \boldsymbol{T_{t\to t'}}, K) >$$

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

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

Smoothness Loss L_s

- $\boldsymbol{\partial_i d_t}$: *Gradient smoothness*
- $e^{\{\partial_x |I_t|\}}$: *Edge aware smoothness*

Disparity

visual inference

Loss Function
Superpixel Binary Loss Function

TECHNISCHE UNIVERSITÄT DARMSTADT

Mathematical Implementation Binary SP–Loss

as part of the smoothness loss

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

Goal

$$L_s = \lambda \left(|\partial_x d_t| * \boldsymbol{\mu_x} + |\partial_y d_t| * \boldsymbol{\mu_y} \right)$$

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

Calculate gradient of SP-labels

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

If $|\partial_{x/y} \boldsymbol{SP}|$ ~= 0 & $|\partial_{x/y} I_t| < threshold$:

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

else:

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

visual inference

Loss Function
Superpixel Continuous Loss Function

TECHNISCHE UNIVERSITÄT DARMSTADT

Mathematical Implementation Continuous SP–Loss

as part of the smoothness loss

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

Goal

$$L_s = \lambda \left(|\partial_x d_t| * \boldsymbol{\mu_x} + |\partial_y d_t| * \boldsymbol{\mu_y} \right)$$

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

Calculate gradient of SP-labels

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

If $|\partial_{x/y} \boldsymbol{SP}| \sim= 0$:

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

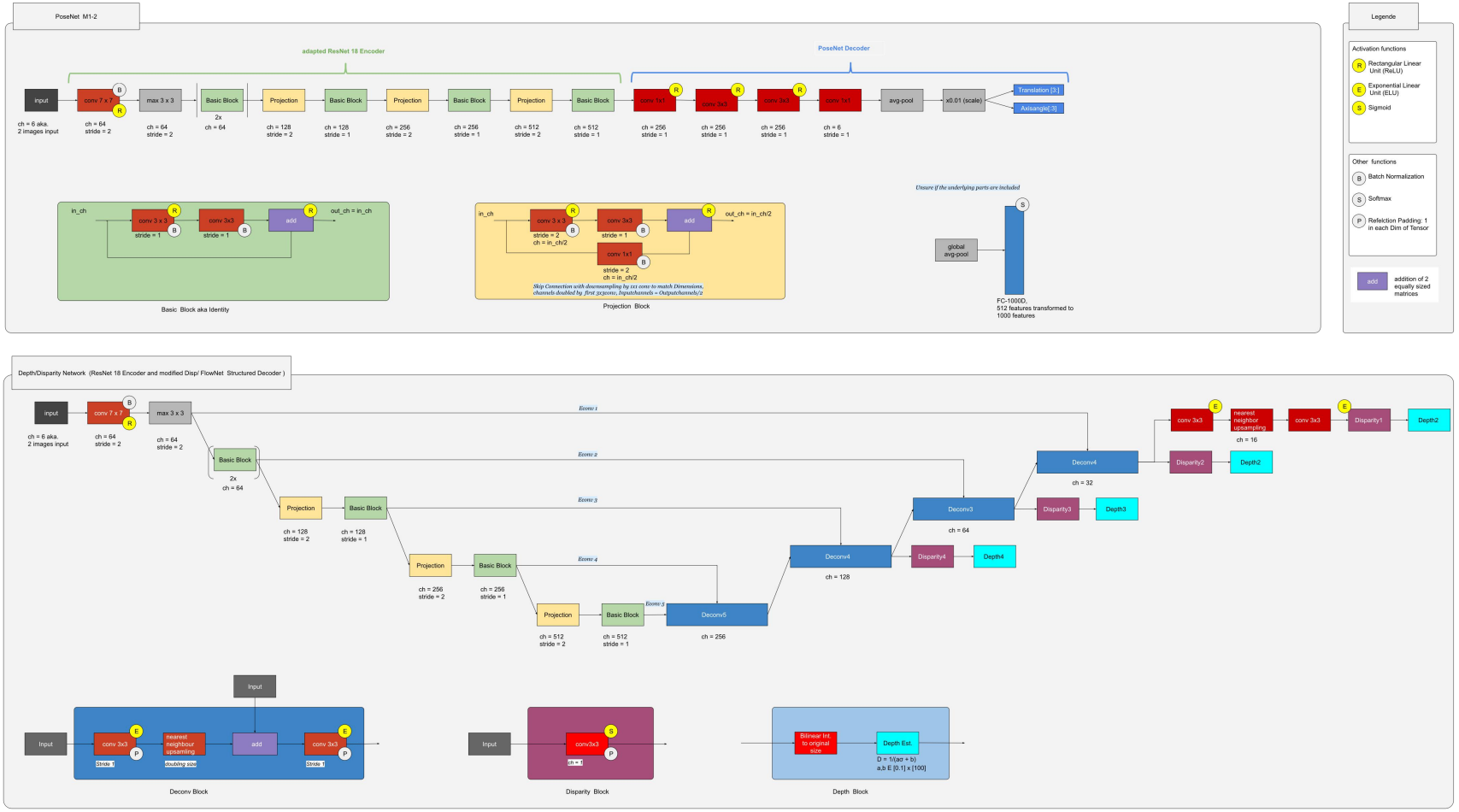
else:

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

visual inference

2. Architecture

Detail View: Digging deeper into Monocular View [1]



Superpixel Benchmark Parameters

TECHNISCHE UNIVERSITÄT DARMSTADT

Boundary Recall (Rec)

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

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

visual inference

Superpixel Benchmark Parameters

TECHNISCHE UNIVERSITÄT DARMSTADT

Undersegmentation Error (UE)

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

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

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

visual inference

Superpixel Benchmark Parameters

TECHNISCHE UNIVERSITÄT DARMSTADT

Explained Variation (EV)

- quanties the quality of a superpixel segmentation without relying on ground truth
- μ (Sj) and μ(I) are the mean color of superpixel Sj and the image I

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

visual inference

Superpixel Benchmark Parameters

TECHNISCHE UNIVERSITÄT DARMSTADT

Compactness (CO)

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

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

visual inference

Benchmark of depth estimation Algorithm some commonly used error metrics:

TECHNISCHE UNIVERSITÄT DARMSTADT

Error Metrics

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

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

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

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

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

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

Groundtruh

- For example outdoor datasets (Hight depth range / less depth resolution)
  - KITTI – Depth by LIDAR
  - Make3D – Depth by LIDAR

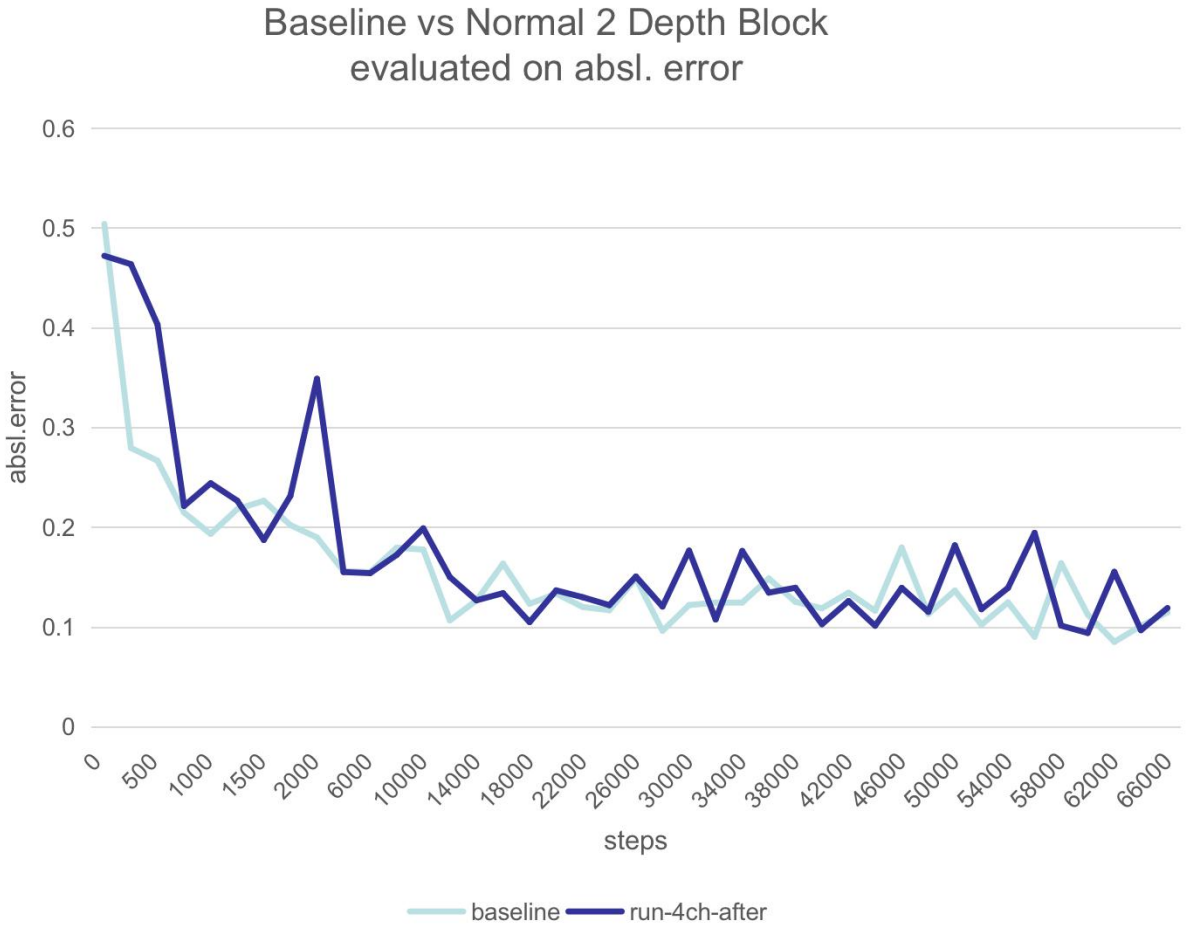
visual inference

3. Improvements
Iteration I: Normal 2 Depth Block

TECHNISCHE UNIVERSITÄT DARMSTADT

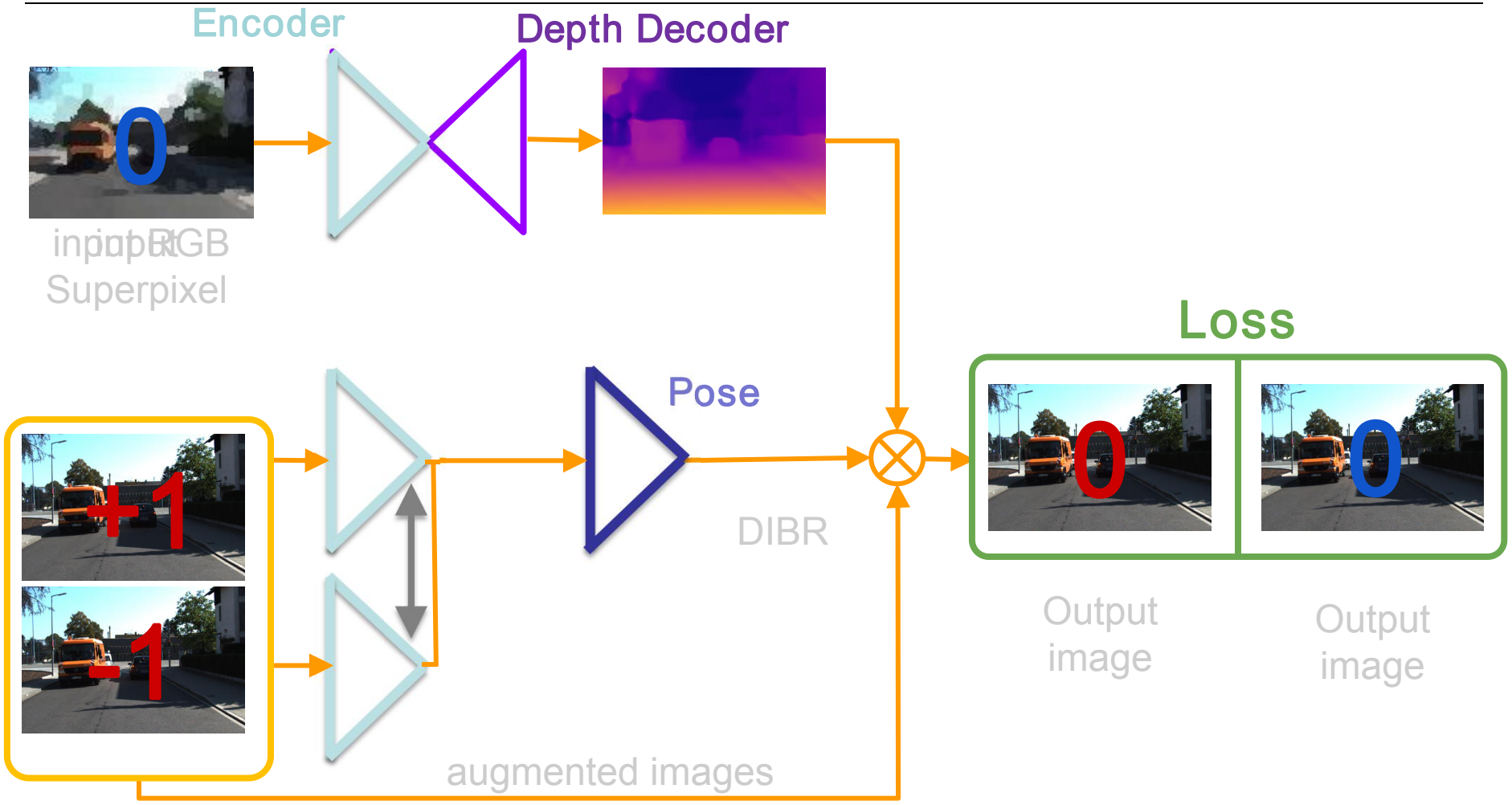
4 CH Input

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



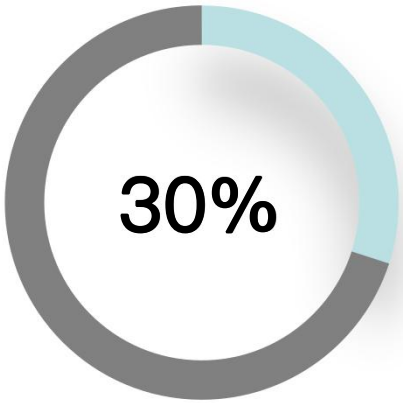
visual inference

5. Architecture
New Input: RGB



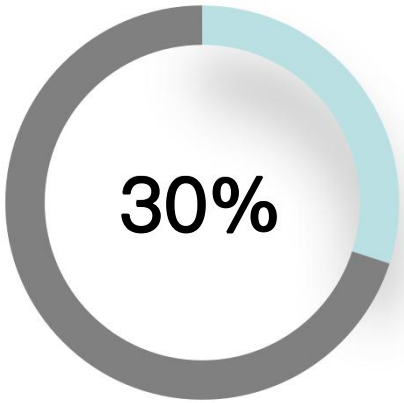
4. Results

Focus on three significant approaches



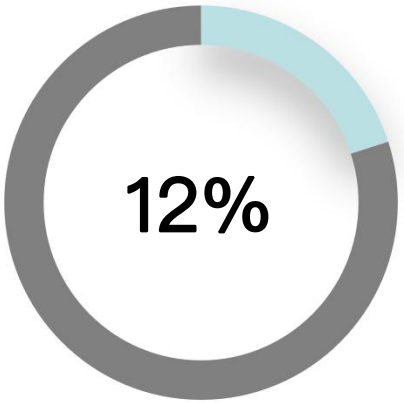
Data title here

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

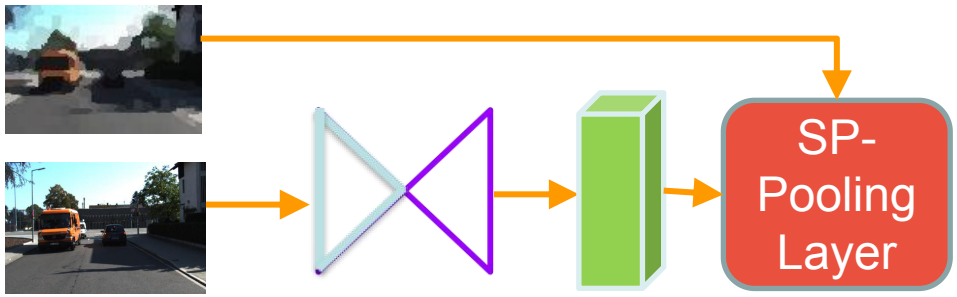
in architecture	in loss

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.



visual inference