Three-Dimensional Computer Vision
Deep Learning in stereoscopic reconstruction

Marin Oršić

UniZG-FER D307

UNIVERSITY OF ZAGREB
Faculty of Electrical
Engineering and
Computing

Table of Contents

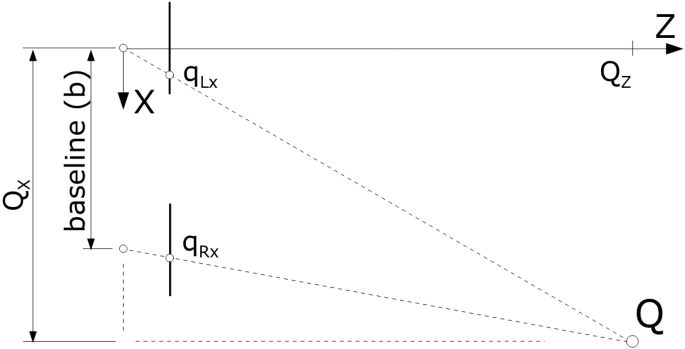
Stereo pair rectification

Computing cost volumes

Lab assignment overview

Deep learning for stereo reconstruction

Recap: rectified stereo

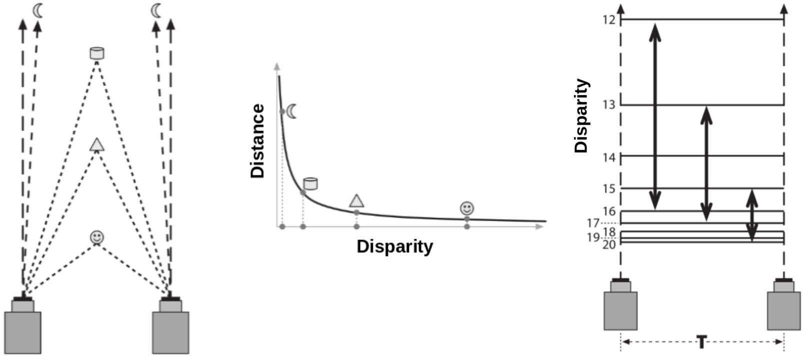


[segvic3d]

$$q_{Lx} = f \cdot \frac{Q_x}{Q_Z} \quad q_{Rx} = f \cdot \frac{Q_x - b}{Q_Z}$$

$$d = q_{Lx} - q_{Rx} = f \cdot \frac{b}{Q_Z}$$

True distance from disparity: error analysis



[kreso13ms]

Rectified stereo

Motivation:

- Constrain the stereo matching algorithm search space: searching along rectified epipolar lines is computationally more efficient.
- Off-the-shelf stereo algorithms assume rectified stereo image pairs.

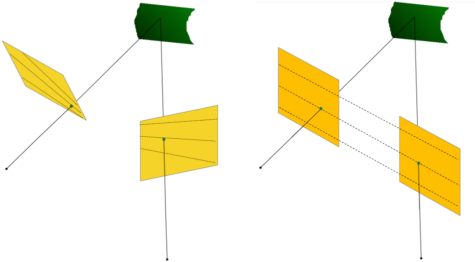
Rectified stereo

Motivation:

- Constrain the stereo matching algorithm search space: searching along rectified epipolar lines is computationally more efficient.
- Off-the-shelf stereo algorithms assume rectified stereo image pairs.

Stereo rectification goal is to:

- make epipolar lines parallel to the x-axis in both images (project epipoles to infinity), and
- have corresponding points project to the same y-axis value.



https://www.cs.cmu.edu/~16385/s17/Slides/13.1_Stereo_Rectification.pdf

Rectified stereo

We aim to obtain $\mathbf{R} = \mathbf{I}$ and $\mathbf{t}^T = [T, 0, 0]$

The essential matrix for this system equates to

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

Rectified stereo: motivation

Let's construct the rotation matrix $\mathbf{R}$

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}$$

Where:

$$\mathbf{r}_1 = \mathbf{e}_1 = \frac{\mathbf{t}}{\|\mathbf{t}\|}$$

$$\mathbf{r}_2 = \left[\mathbf{r_1}\right]_\times \left[0, 0, 1\right]^T$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

Rectified stereo: motivation

Let's construct the rotation matrix $\mathbf{R}$

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}$$

Where:

$$\mathbf{r}_1 = \mathbf{e}_1 = \frac{\mathbf{t}}{\|\mathbf{t}\|}$$

$$\mathbf{r}_2 = \left[\mathbf{r_1}\right]_\times \left[0, 0, 1\right]^T$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$\mathbf{e}_1$ projects to infinity.

$$\mathbf{R}\mathbf{e}_1 = \begin{bmatrix} \mathbf{r}_1^T\mathbf{e}_1 \\ \mathbf{r}_2^T\mathbf{e}_1 \\ \mathbf{r}_2^T\mathbf{e}_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Rectified stereo

The new camera projection is $\mathbf{Q}_{\text{rect}} = \mathbf{KR}$.
$\mathbf{K}$ can be set to a $\mathbf{K} = \frac{\mathbf{K}_1 + \mathbf{K}_2}{2}$ if the cameras are similar.

We are interested in finding a rectifying homography $\mathbf{H}$ for each camera:

$$\mathbf{x}_{rect} = \mathbf{Q}_{rect}\mathbf{Q}_o^{-1}\mathbf{x}_o = \underbrace{\mathbf{KR}(\mathbf{K_o R_o})^{-1}}_{\mathbf{H}}\mathbf{x}_o$$

We have $\mathbf{H}_1 = \mathbf{KRK}_1^{-1}$ and $\mathbf{H}_2 = \mathbf{KR}(\mathbf{K}_2\mathbf{R})^{-1}$

Rectified stereo: example on KITTI

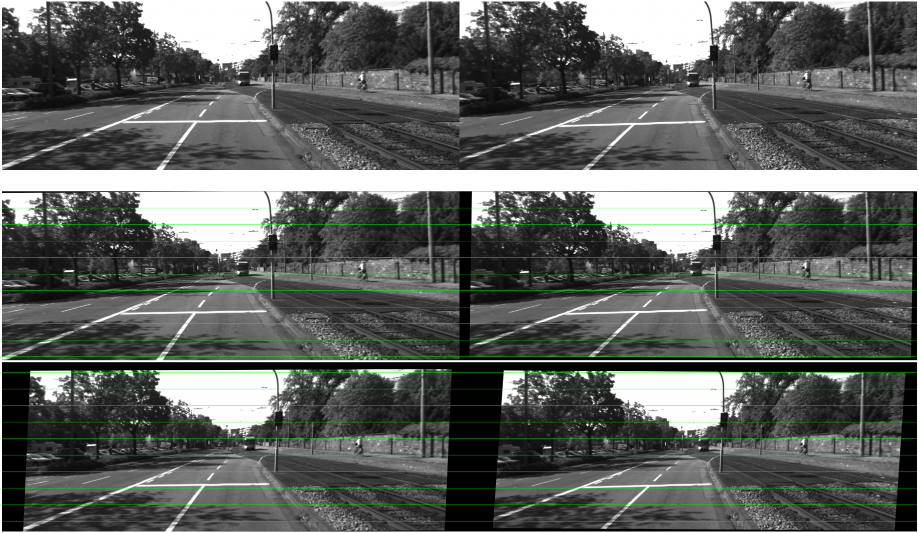


Table of Contents

Stereo pair rectification

Computing cost volumes

Lab assignment overview

Deep learning for stereo reconstruction

Constructing a cost volume

- Rectification constrains the correspondence search along the x image axis.
- For dense stereo, we can explicitly compute all possible correspondence values.
- Such structure is called cost volume.

Constructing a cost volume

Cost volume $\mathbf{V}_{dij}$ is a three-dimensional array. Element at position (d, i, j) corresponds to:

$$\mathbf{V}_{dij} = -\mathbf{l}_{fij}\mathbf{r}_{fi(j-d)}$$

l_{fij} and r_{fij} are f-dimensional descriptors suitable for computing a matching cost.

Constructing a cost volume

Cost volume $\mathbf{V}_{dij}$ is a three-dimensional array. Element at position (d, i, j) corresponds to:

$$\mathbf{V}_{dij} = -\mathbf{l}_{fij}\mathbf{r}_{fi(j-d)}$$

l_{fij} and r_{fij} are f-dimensional descriptors suitable for computing a matching cost. There are multiple methods that enable matching cost computation

(Census, learning-based).

Constructing a cost volume

Cost volume $\mathbf{V}_{dij}$ is a three-dimensional array. Element at position (d, i, j) corresponds to:

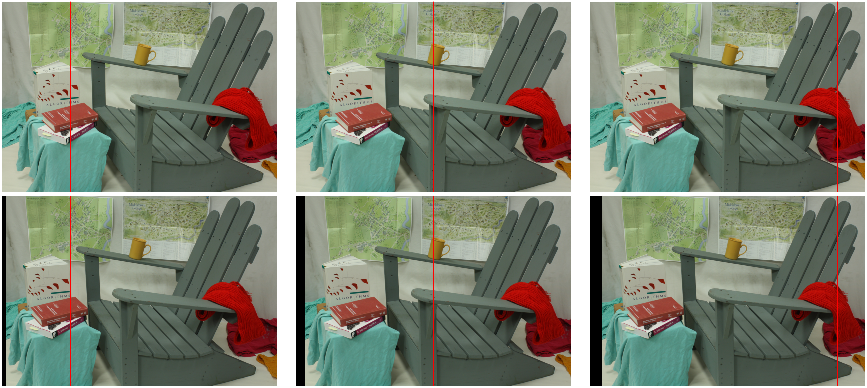
$$\mathbf{V}_{dij} = -\mathbf{l}_{fij}\mathbf{r}_{fi(j-d)}$$

l_{fij} and r_{fij} are f-dimensional descriptors suitable for computing a matching cost. There are multiple methods that enable matching cost computation

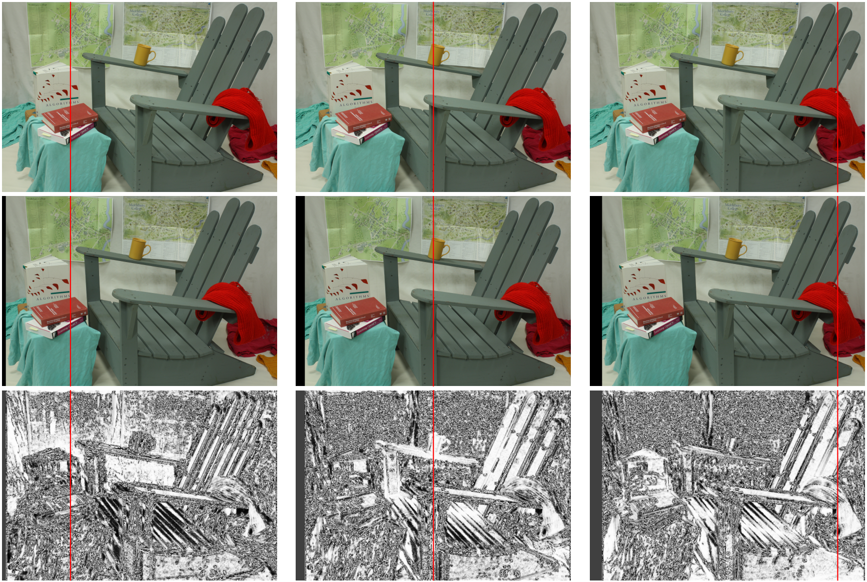
(Census, learning-based).

Cost volumes are used in most dense stereo algorithms [scharstein02ijcv].

Constructing a cost volume



Constructing a cost volume



Constructing a cost volume

By taking $\arg\min_{d} \mathbf{V}_{dij}$, we can obtain a disparity rough estimate:

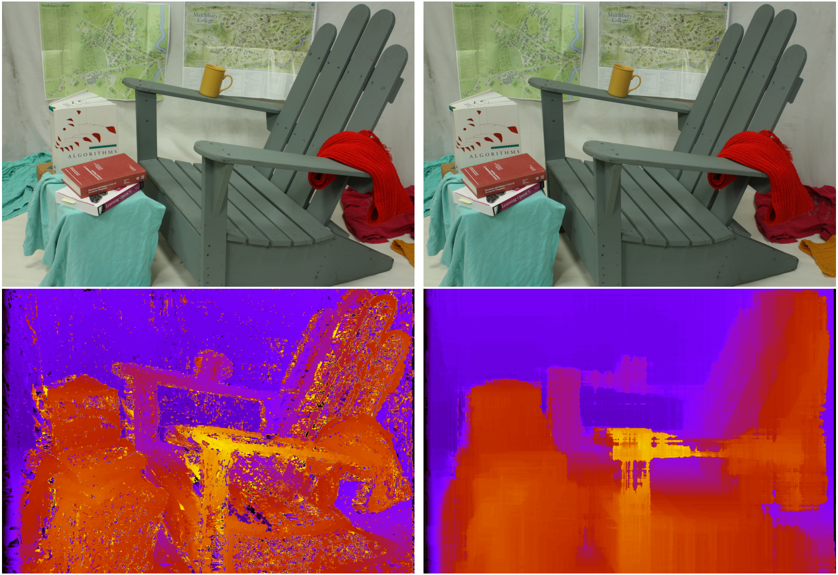


Table of Contents

Stereo pair rectification

Computing cost volumes

Lab assignment overview

Deep learning for stereo reconstruction

Lab assignment

`https://gitlab.com/FER-D307/Nastava/3d-racvid/3drv-lab2`

Table of Contents

Stereo pair rectification

Computing cost volumes

Lab assignment overview

Deep learning for stereo reconstruction

Deep learning for stereo

- We mentioned that *classical* stereo reconstruction algorithms rely on heuristics for i) expressing the matching cost and ii) cost volume refinement.
- Both heuristics can be replaced by *neuralizing* parts of the stereo reconstruction algorithm.

MC-CNN (Matching cost convolutional neural network)

Objective: Leverage CNNs to learn a matching cost between image patches.

MC-CNN (Matching cost convolutional neural network)

Objective: Leverage CNNs to learn a matching cost between image patches.

$$L(r_f, p_f, q_f) = max(0, r_f p_f - r_f q_f + m)$$

MC-CNN (fast architecture)

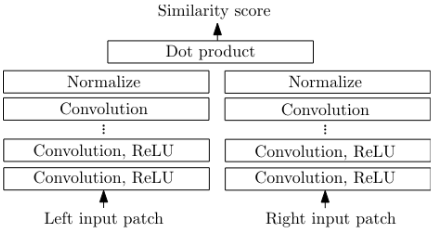


Figure 2: The fast architecture is a siamese network. The two sub-networks consist of a number of convolutional layers followed by rectified linear units (abbreviated "ReLU"). The similarity score is obtained by extracting a vector from each of the two input patches and computing the cosine similarity between them. In this diagram, as well as in our implementation, the cosine similarity computation is split in two steps: normalization and dot product. This reduces the running time because the normalization needs to be performed only once per position (see Section 3.3).

[zbontar15jmlr]

MC-CNN (accurate architecture)

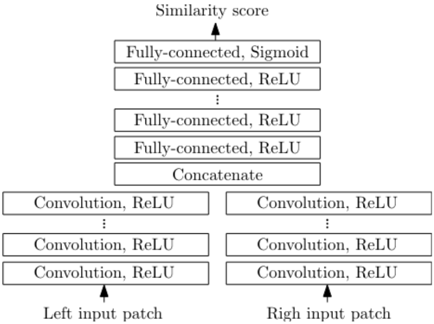
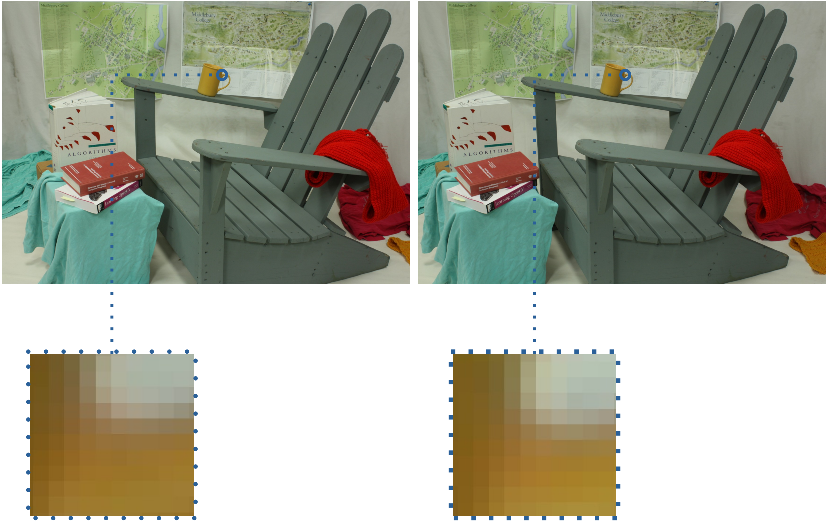


Figure 3: The accurate architecture begins with two convolutional feature extractors. The extracted feature vectors are concatenated and compared by a number of fully-connected layers. The inputs are two image patches and the output is a single real number between 0 and 1, which we interpret as a measure of similarity between the input images.

[zbontar15jmlr]

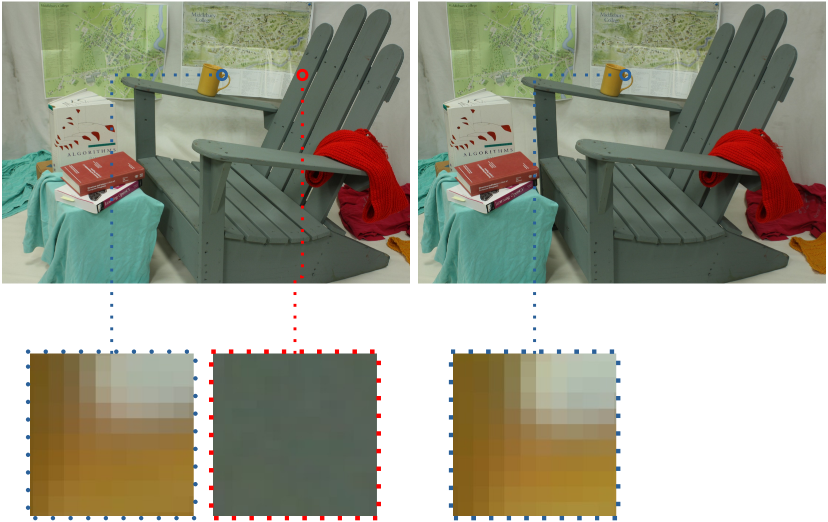
MC-CNN

Let r_f and p_f be embeddings from corresponding image patches.



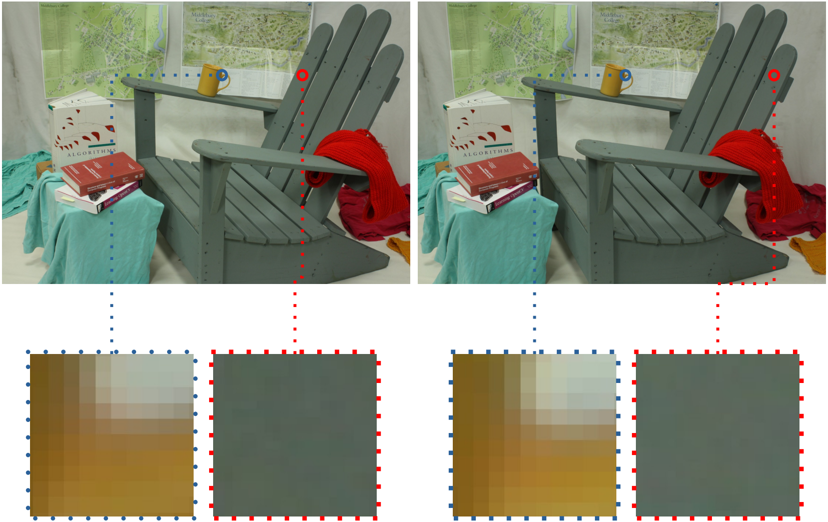
MC-CNN

Formally, the goal is to obtain $r_f p_f > r_f q_f, \forall(r_f, p_f, q_f)$



MC-CNN

Unfortunately, processing image patches alone does not solve everything.



GC-Net

Possible solution: describe the entire stereo pipeline using a deep model.

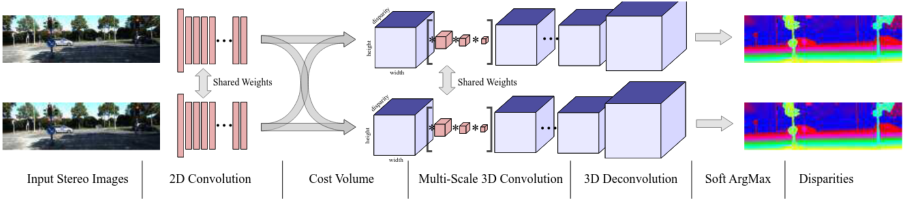


Figure 1: **Our end-to-end deep stereo regression architecture, GC-Net** (Geometry and Context Network).

[kendall17cvpr]

GC-Net

CNN processes the input image pair

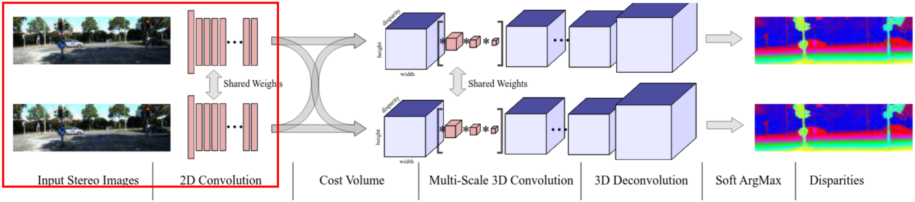


Figure 1: **Our end-to-end deep stereo regression architecture, GC-Net** (Geometry and Context Network).

`[kendall17cvpr]`

GC-Net

Correlation volume is processed using 3D convolutions (»1GB memory).

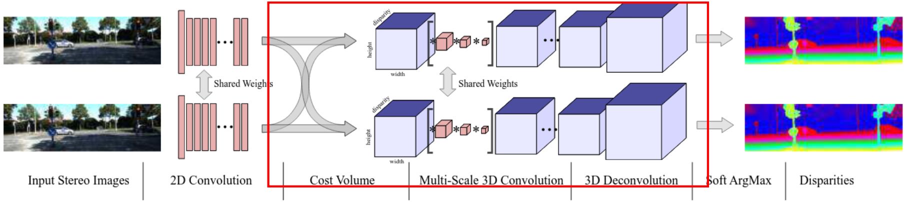


Figure 1: **Our end-to-end deep stereo regression architecture, GC-Net** (Geometry and Context Network).

`[kendall17cvpr]`

GC-Net

Output is computed using softargmin $:= \sum_{d=0}^{D} d \times \sigma(-c_d)$

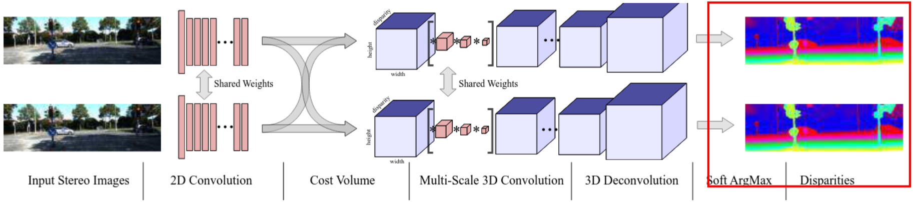
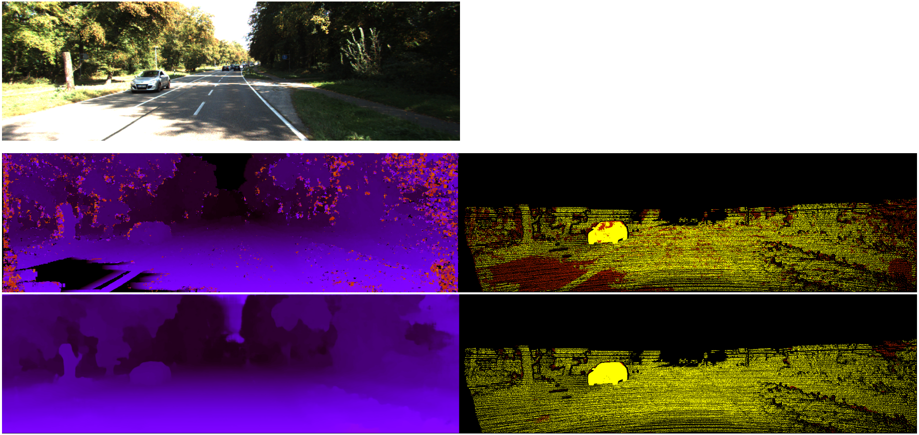


Figure 1: **Our end-to-end deep stereo regression architecture, GC-Net** (Geometry and Context Network).

`[kendall17cvpr]`

GC-Net compared to MC-CNN



[orsic17ms]

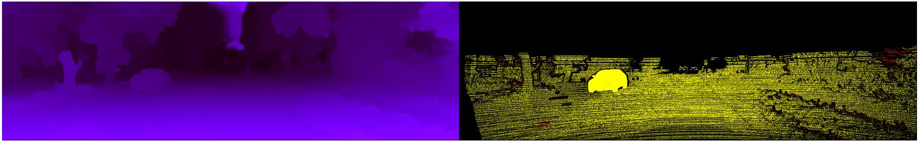
GC-Net

Pros:

- End-to-end trainable method
- Geometrical priors in the network
- Flexible wrt. max disparity and image size

Cons:

- Heavy processing requirements
- Memory allocation of the 4-D cost volume tensor



[orsic17ms]

RAFT-Stereo

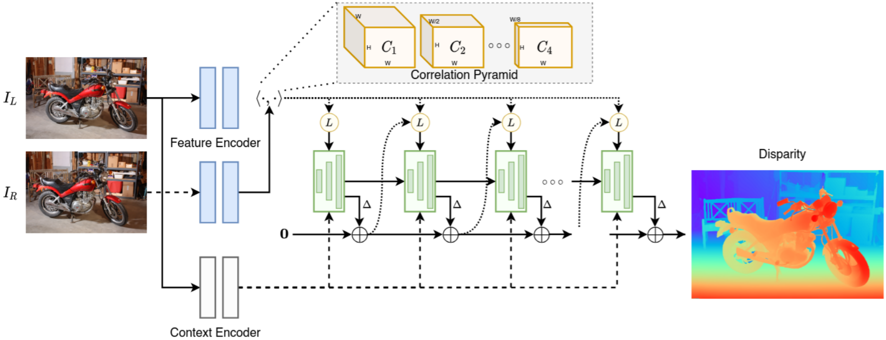


Figure 1. Correlation features (blue) are extracted from each of the images and are used to construct the correlation pyramid. ”Context” image features (white) and an initial hidden state are also extracted from the context encoder. The disparity field is initialized to zero. Every iteration, the GRU(s) (green) use the current disparity estimate to sample from the correlation pyramid. The resulting correlation features, initial image features and current hidden state(s) are used by the GRU(s) to produce a new hidden state and an update to the disparity.

[lipson21ic3dv]

Correlation pyramid

$$\mathbf{C}^0_{ijk} = \sum_f \mathbf{f}_{ijh} \cdot \mathbf{g}_{ikh}, \mathbf{C}^0 \in \mathbb{R}^{H\times W\times W}$$

Correlation pyramid

$$\mathbf{C}^0_{ijk} = \sum_f \mathbf{f}_{ijh} \cdot \mathbf{g}_{ikh}, \mathbf{C}^0 \in \mathbb{R}^{H \times W \times W}$$

A correlation pyramid consists of $\mathbf{C}^0$ pooled in the last dimension ($C^{k+1} \in \mathbb{R}^{H \times W \times W/2^k}$).

Correlation pyramid

$$\mathbf{C}^0_{ijk} = \sum_f \mathbf{f}_{ijh} \cdot \mathbf{g}_{ikh}, \mathbf{C}^0 \in \mathbb{R}^{H \times W \times W}$$

A correlation pyramid consists of $\mathbf{C}^0$ pooled in the last dimension ($C^{k+1} \in \mathbb{R}^{H \times W \times W/2^k}$). Correlation pyramid is indexed with current disparity estimate using bilinear interpolation.

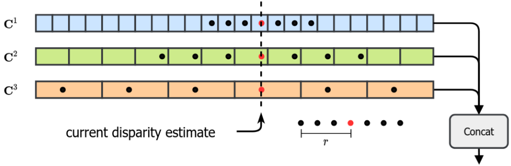


Figure 2. Lookup from the correlation pyramid. We use the current estimate of disparity to retrieve values from the each level of the correlation pyramid. We index from each level in the pyramid by linear interpolating at the current disparity estimate and at integer offsets, whose size depends on the correlation pyramid level.

[lipson21ic3dv]

RAFT-Stereo - convolutional GRU

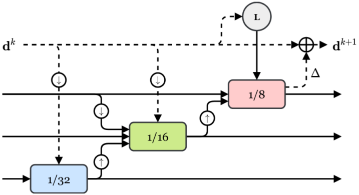


Figure 3. Multilevel GRU. We use a 3-level convolutional GRU which acts on feature maps at 1/32, 1/16, and 1/8 the input image resolution. Information is passed between GRUs at adjacent resolutions using upsampling and downsampling operations. The GRU at the highest resolution (red) performs lookups from the correlation pyramid and updates the disparity estimate.

[lipson21ic3dv]

$$
\begin{aligned}
z_t &= \sigma(\mathrm{conv}_{3\times 3}([h_{t-1}, x_t], W_z)) \\
r_t &= \sigma(\mathrm{conv}_{3\times 3}([h_{t-1}, x_t], W_r)) \\
\tilde{h}_t &= \tanh(\mathrm{conv}_{3\times 3}([r_t \odot h_{t-1}, x_t]), W_h) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \ \tilde{h}_t
\end{aligned}
$$

RAFT-Stereo - loss

$$L = \sum_{i=1}^{N} \gamma^{N-1} \|\mathbf{d}_{gt} - \mathbf{d}_i\|_1, \gamma = 0.9$$

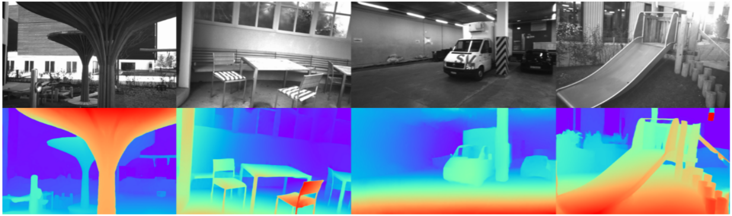


Figure 4. Results on the ETH3D stereo dataset. RAFT-Stereo is robust to difficulties like textureless surfaces and overexposure.

[lipson21ic3dv]

Conclusions

- Stereo pair rectification greatly simplifies the matching algorithm.
- Deep learning improves parts of the stereo reconstruction pipeline.
- RAFT-Stereo is a great choice for near real-time reconstruction.

Rectified stereo

We, consider the calibrated stereo case, i.e. $\mathbf{P}_{c1}$ and $\mathbf{P}_{c2}$ are known:

$$\mathbf{P}_{c1} = \mathbf{K}_1 \left[\mathbf{R}_1 | \mathbf{t}_1\right] = \left[\mathbf{Q}_1 | \tilde{\mathbf{q}_1}\right]$$

Projection of the optical centre can be expressed as:

$$\begin{aligned}\mathbf{c}_1 &= -\,\mathbf{Q}_1^{-1}\tilde{\mathbf{q}_1} = -\mathbf{Q}_1^{-1}\mathbf{K}_1\mathbf{t}_1 \\ &= -\,(\mathbf{K}_1\mathbf{R}_1)^{-1}\mathbf{K}_1\mathbf{t}_1 = -\mathbf{R}_1^{-1}\mathbf{K}_1^{-1}\mathbf{K}_1\mathbf{t}_1 \\ &= -\,\mathbf{R}_1^{-1}\mathbf{t}_1\end{aligned}$$

$\mathbf{P}_{c2}$ and $\mathbf{c}_2$ are analogously expressed for the second camera.

Rectified stereo: motivation

Let's construct the rotation matrix $\mathbf{R}$

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}$$

Where:

$$\mathbf{r}_1 = (\mathbf{c}_1 - \mathbf{c}_2)/\|\mathbf{c}_1 - \mathbf{c}_2\|$$

$$\mathbf{r}_2 = \mathbf{k} \times \mathbf{r}_1$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$\mathbf{k}$ is arbitrary and can be set to a reasonable value, e.g. last row in $\mathbf{R}_1$.

Rectified stereo: uncalibrated case

Let:

$$\hat{\mathbf{m}} = \mathbf{H}\mathbf{m}, \hat{\mathbf{m}'} = \mathbf{H}\hat{'}\mathbf{m}'$$

from the epipolar constraint, we have:

$$\mathbf{m}'^{T} \underbrace{\mathbf{H}'^{T} \hat{\mathbf{F}} \mathbf{H}}_{\mathbf{F}} \mathbf{m} = 0$$

There are multiple solutions for $\mathbf{H}$ and $\mathbf{H}'$: we aim to minimize image distortion.

Rectified stereo: uncalibrated case

Consider the following fundamental matrix: $\hat{\mathbf{F}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$

The following holds:

- All epipolar lines are parallel to the x-axis,
- corresponding points have identical y-coordinates.