

Review of a paper-Stereo Matching by Training a Convolutional Neural Network

Qingwei (David) Wu

Columbia University

qw2208@columbia.edu

January 29, 2017


Overview

- 1 Derive Depth Map from Disparity Map
- 2 Convolutional Neural Network Model
 - Matching Cost
 - Data Set
 - Network Architecture
 - Accurate Architecture
 - More on Matching Cost
- 3 Stereo Method & Post-processing Steps
 - Cross-based Cost Aggregation
 - Semiglobal Matching
 - Compute the Disparity Image

Rank in KITTI

Evaluation ground truth All pixels

Evaluation area All pixels

	Method	Setting	Code	D1-bg	D1-fg	D1-all	Density	Runtime	Environment	Compare
1	E2EDSR			2.21 %	6.16 %	2.87 %	100.00 %	0.7 s	Nvidia GTX Titan X	<input type="checkbox"/>
2	DRR			2.58 %	6.04 %	3.16 %	100.00 %	0.4 s	Nvidia GTX Titan X	<input type="checkbox"/>
3	L-ResMatch		code	2.72 %	6.95 %	3.42 %	100.00 %	48 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
A. Shaked and L. Wolf: Improved Stereo Matching with Constant Highway Networks and Reflective Loss . arXiv preprint arxiv:1701.00165 2016.										
4	Displets v2		code	3.00 %	5.56 %	3.43 %	100.00 %	265 s	>8 cores @ 3.0 Ghz (Matlab + C/C++)	<input type="checkbox"/>
F. Gunev and A. Geiger: Displets: Resolving Stereo Ambiguities using Object Knowledge . Conference on Computer Vision and Pattern Recognition (CVPR) 2015.										
5	CNNF+SGM			2.78 %	7.69 %	3.60 %	100.00 %	71 s	TESLA K40C	<input type="checkbox"/>
6	PBCEP			2.58 %	8.74 %	3.61 %	100.00 %	68 s	Nvidia GTX Titan X	<input type="checkbox"/>
A. Seki and M. Pollefeys: Patch Based Confidence Prediction for Dense Disparity Map . British Machine Vision Conference (BMVC) 2016.										
7	SN			2.66 %	8.64 %	3.66 %	100.00 %	67 s	Titan X	<input type="checkbox"/>
8	MC-CNN-acrt		code	2.89 %	8.88 %	3.89 %	100.00 %	67 s	Nvidia GTX Titan X (CUDA, Lua/Torch7)	<input type="checkbox"/>
J. Zbontar and Y. LeCun: Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches . Submitted to JMLR .										
9	CNN-SPS			3.30 %	7.92 %	4.07 %	100.00 %	80 s	GPU @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
L. Chen, J. Chen and L. Fan: A Convolutional Neural Networks based Full Density Stereo Matching Framework . .										
10	PRSM		code	3.02 %	10.52 %	4.27 %	99.99 %	300 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>

Derive Depth Map from Disparity Map

- Description of stereo matching: Given two images taken by cameras at different horizontal positions, we wish to compute the disparity d for each pixel in the left image. Disparity refers to the difference in horizontal location of an object in the left and right image, say, an object at (x, y) in the left image appears at position $(x - d, y)$ in the right image.
- If we know the disparity of an object we can compute its depth z using the following relation

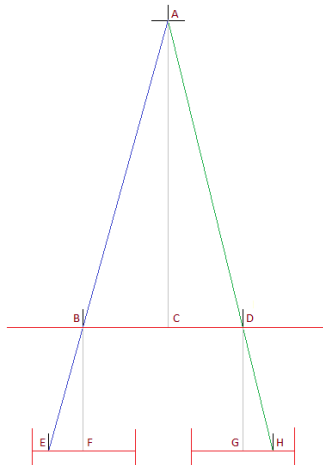
$$z = \frac{f \cdot B}{d} \quad (1)$$

Derive Depth Map from Disparity Map (Continue)

Math Derivation BF and DG
are focal distances.

$$\begin{aligned}d &= EF + GH \\&= BF \cdot \left(\frac{EF}{BF} + \frac{GH}{BF} \right) \\&= BF \cdot \frac{BC + CD}{AC} \\&= \frac{BF}{AC} \cdot BD\end{aligned}$$

d is the disparity. BD is how we place the cameras. AC represents the depth of the object!



Matching Cost - Goal

- Matching cost at each position \mathbf{p} for all disparities d under consideration, i.e.,

$$C_{SAD} = \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} |I^L(\mathbf{q}) - I^R(\mathbf{q} - \mathbf{d})| \quad (2)$$

where I^L and I^R are image intensities at its position in the left and right image respectively. $\mathcal{N}_{\mathbf{p}}$ is the set of locations with a fixed rectangular window center at \mathbf{p} .

- $\mathbf{d} = (d, 0)$.
- (1) can be measuring the cost associate with matching a patch from the left image, centered at position \mathbf{p} , with a patch from the right image centered at $\mathbf{p} - \mathbf{d}$.

Construct the Data Set

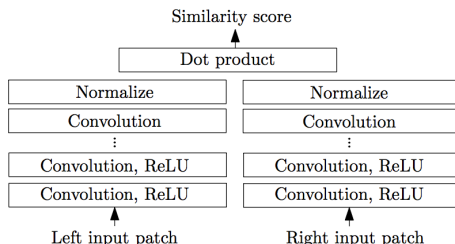
- Use ground truth disparity maps from KITTI and Middleburg stereo data sets \rightarrow Training set.
- $\langle \mathcal{P}_{n*n}^L(\mathbf{p}), \mathcal{P}_{n*n}^R(\mathbf{q}) \rangle$ denote a pair of pathes. d denotes the correct disparity at position \mathbf{p} . $\mathcal{P}_{n*n}^R(\mathbf{q})$ is an $n * n$ patch from the right image centered at position \mathbf{q} .
- A negative example:
 - Set the center of the right patch to $\mathbf{q} = (x - d + o_{neg}, y)$, where $o_{neg} \in [dataset_neg_low, dataset_neg_high] \cup [-dataset_neg_high, -dataset_neg_low]$
 - The resulting image patches are not centered around the same 3D point.
- A positive example:
 - $\mathbf{q} = (x - d + o_{pos}, y)$, $o_{pos} \in [-dataset_pos, dataset_pos]$ where $dataset_pos \leq 1$
 - Instead of setting o_{pos} to zero.

Fast Architecture

- Trained by minimizing a hinge loss:
 s_+ is output of the network for positive examples. s_- is output of the network for negative examples. Define hinge loss:

$$\text{hinge} = \max(0, m + s_- - s_+) \quad (3)$$

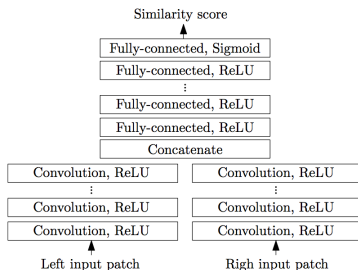
- Set the margin m to 0.2.
- Hypermeters include *num_conv_layers*, *conv_kernel_size*, etc..(See paper)



Accurate Architecture

- The last fully-connected layer produces a single number which after transformed with sigmoid nonlinearity, s , is interpreted as the similarity score between the input patches. t denote the class of the training example.
- Cross Entropy Loss: (Different from the paper)

$$-[t \cdot \log(s) + (1 - t) \cdot \log(1 - s)] \quad (4)$$



Hyper parameters see paper.

More on Matching Cost

The output of the network used to initialize the matching cost:

$$C_{CNN} = -s(\langle \mathcal{P}^L(\mathbf{p}), \mathcal{P}^R(\mathbf{p} - \mathbf{d}) \rangle) \quad (5)$$

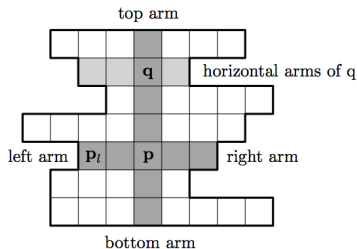
Naively, we'll have to perform the forward pass for each image location and each disparity under consideration.

- ① Outputs of the two sub-networks need to be computed only once per location.
- ② Outputs of two sub-networks can be computed for all pixels in a single forward pass instead of small image patches.
- ③ The fully-connected part of the network needs to be run d times (A bottleneck!).

Summary: Run the sub-networks once on each image and run the fully-connected layers d times (d is the maximum disparity under consideration.)

Cross-based Cost Aggregation

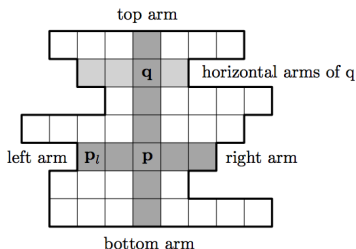
- Traditional method: Average the matching cost over a fixed window.
Caveat: Fail near depth discontinuities!
- Think of an method that adaptively selects the neighborhood for each pixel. Support region is collected only from pixels of the same physical object.
- Construct an upright cross at each position.



Cross-based Cost Aggregation (Continue)

- The left arm \mathbf{p}_l at position \mathbf{p} extends left as long as:
 - $|I(\mathbf{p} - I(\mathbf{p}_l))| < cbca_intensity$
 - $\|\mathbf{p} - \mathbf{p}_l\| < cbca_distance$
- The right, bottom and top arms are constructed analogously. The aggregation should consider the support regions of both images in a stereo pair. The combined support region U_d is denoted as:

$$U_d(\mathbf{p}) = \{\mathbf{q} | \mathbf{q} \in U^L(\mathbf{p}), \mathbf{q} - \mathbf{d} \in U^R(\mathbf{p} - \mathbf{d})\} \quad (6)$$



Cross-based Cost Aggregation (Continue)

The matching cost is averaged over the combined support region:

$$\begin{aligned} C_{CBCA}^0(\mathbf{p}, d) &= C_{CNN}(\mathbf{p}, d) \\ C_{CBCA}^i(\mathbf{p}, d) &= \frac{1}{|U_d(\mathbf{p})|} \sum_{\mathbf{q} \in U_d(\mathbf{p})} C_{CBCA}^{i-1}(\mathbf{q}, d) \end{aligned}$$

where i is the iteration number.

Semi-global Matching

- Refine the matching cost by enforcing smoothness constraints on the disparity image. Define an energy function $E(D)$ that depends on the disparity image D .

$$\begin{aligned} E(D) = & \sum_{\mathbf{p}} (C_{BCA}(\mathbf{p}, D(\mathbf{p})) + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_1 \cdot \mathbb{1}\{|D(\mathbf{p}) - D(\mathbf{q})| = 1\}) \\ & + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_2 \cdot \mathbb{1}\{|D(\mathbf{p}) - D(\mathbf{q})| > 1\}) \end{aligned}$$

- Minimize the energy in a single direction, repeat for several directions (two horizontal and two vertical directions) and average to obtain the final result.

Semi-global Matching (Continue)

- To minimize $E(D)$ in direction \mathbf{r} , define a matching cost $C_r(\mathbf{p}, d)$ with the recurrence relation:

$$\begin{aligned} C_r(\mathbf{p}, d) = & C_{BCA}^4(\mathbf{p}, d) - \min_k C_r(\mathbf{p} - \mathbf{r}, k) \\ & + \min\{C_r(\mathbf{p} - \mathbf{r}, d), C_r(\mathbf{p} - \mathbf{r}, d - 1) + P_1, \\ & C_r(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \min_k C_r(\mathbf{p} - \mathbf{r}, k) + P_2\} \end{aligned}$$

- The final cost $C_{SGM}(\mathbf{p}, d)$ is computed by taking the average across all four directions:

$$C_{SGM}(\mathbf{p}, d) = \frac{1}{4} \sum_{\mathbf{r}} C_r(\mathbf{p}, d)$$

- Repeat cross-based cost aggregation. Do cross-based cost aggregation before and after semi global matching.

Compute the Disparity Image

- The disparity image $D(\mathbf{p})$ is computed by the winner-takes-all strategy, i.e.,

$$D(\mathbf{p}) = \operatorname{argmin}_d C(\mathbf{p}, d) \quad (7)$$

- Others: Interpolation, median filter and sub pixel enhancement. (The paper has provided a brief introduction.)

References



Zbontar, Jure and LeCun, Yann (2016)

Stereo matching by training a convolutional neural network to compare image patches

Journal of Machine Learning Research 17(1-32), 2.

Thanks!