Depth from Stereo

Dominic Cheng
February 7, 2018

Agenda

1. Introduction to stereo
2. Efficient Deep Learning for Stereo Matching
   (W. Luo, A. Schwing, and R. Urtasun. In *CVPR* 2016.)
3. Cascade Residual Learning: A Two-stage Convolutional Neural Network for Stereo Matching
   (J. Pang, W. Sun, J. SJ. Ren, C. Yang, and Q. Yan. In *CVPR* 2017.)

Introduction to Stereo

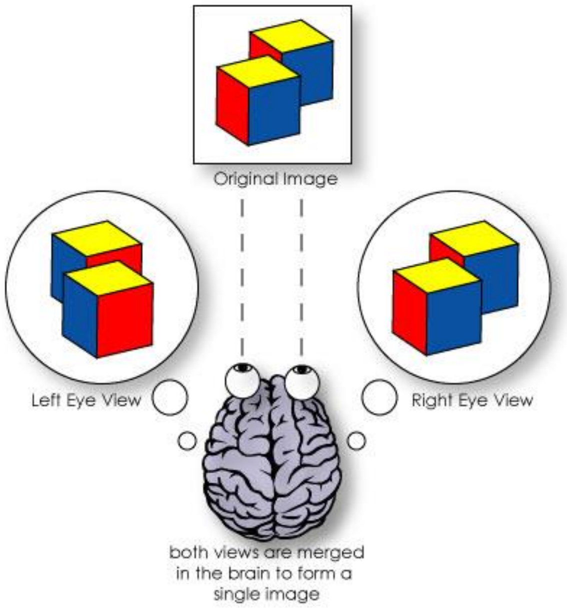
What is stereo?

Depth from images is a very intuitive ability

- Given two images of a scene from (slightly) different viewpoints, we are able to infer depth

Can we do the same using computers?

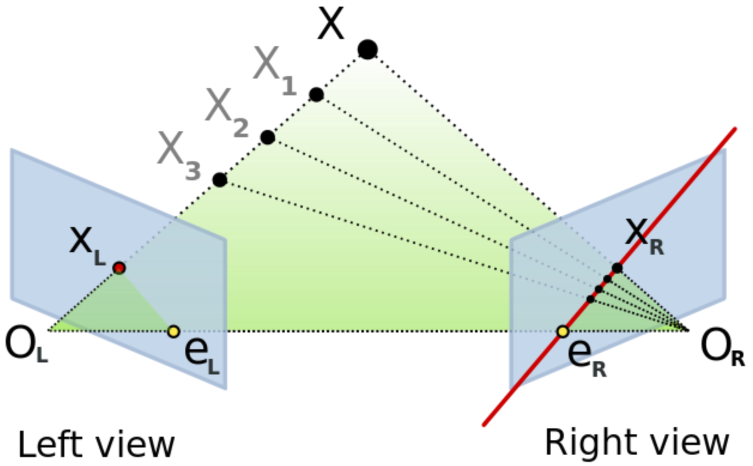
- Yes (kind of?)
- First, we need to appreciate the geometry of the situation



Source: https://s.hswstatic.com/gif/pc-3-d-brain.jpg

Geometry in stereo (a visual overview)

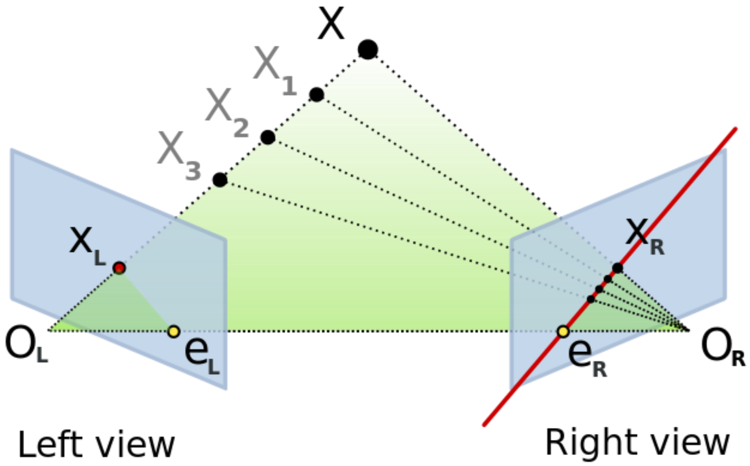
- Think of images as projections of 3D points (in the real world) onto a 2D surface (image plane)
- X_L is the projection of X, X_1, X_2, X_3, ... onto the left image
- X, X_1, X_2, X_3 will also project onto the right image



Source:
https://upload.wikimedia.org/wikipedia/commons/thumb/1/14/Epipolar_geometry.svg/640px-Epipolar_geometry.svg.png?1517775941158

Geometry in stereo (a visual overview)

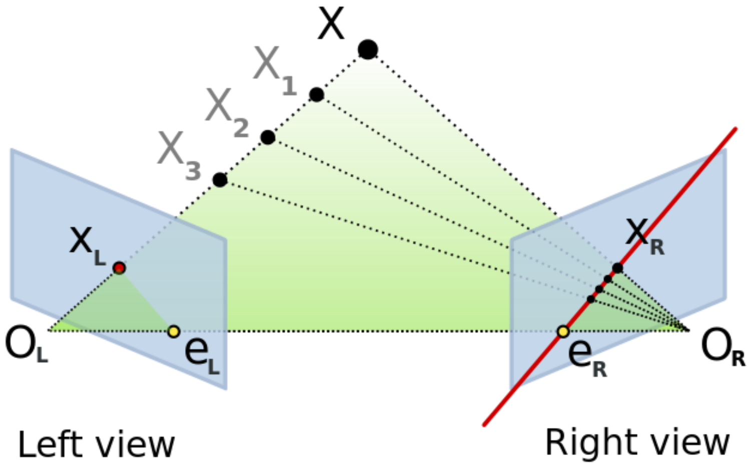
- What do you notice?
- Projections of X_1, X_2, X_3 on right image all lie on a line
- This line is known as an **epipolar line**
    - Points e_L, e_R are known as **epipoles**
    - Projections of cameras' optical centers O_L, O_R onto the images
    - All epipolar lines will intersect at epipoles
    - Left image has corresponding epipolar line
- Geometry of stereo vision also known as **epipolar geometry**



Source:
https://upload.wikimedia.org/wikipedia/commons/thumb/1/14/Epipolar_geometry.svg/640px-Epipolar_geometry.svg.png?1517775941158

Geometry in stereo (a visual overview)

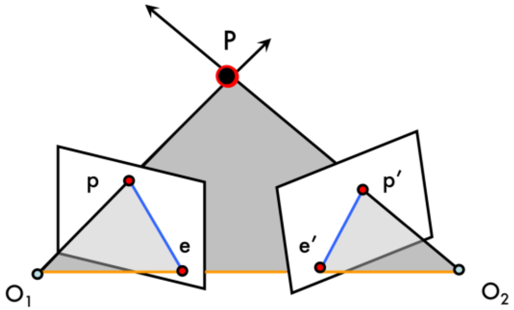
- What does this give us?
- All 3D points that could have resulted in X_L must have a projection on the right image, and must be on the epipolar line $e_R - x_R$
- Given just the left/right images and X_L, you can search on the corresponding epipolar line in the right image. **If you can find the corresponding match X_R, you can uniquely determine the 3D position of X.**

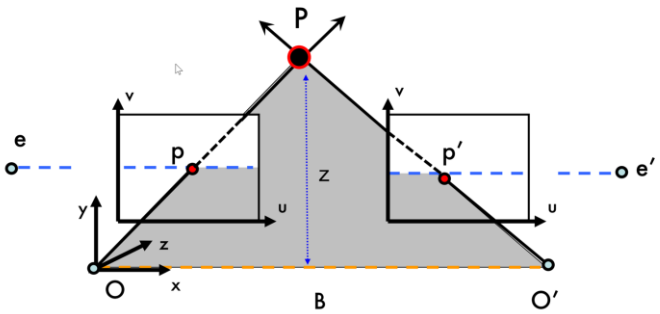


Source:
https://upload.wikimedia.org/wikipedia/commons/thumb/1/14/Epipolar_geometry.svg/640px-Epipolar_geometry.svg.png?1517775941158

Geometry in stereo (a visual overview)

- In practice ...
- Epipolar lines can be made parallel through a process called **rectification**
- Simplifies the process of finding a match and calculating the 3D point





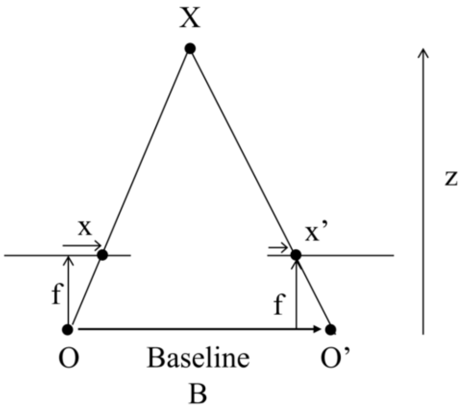
Credit: S. Savarese
Source: http://web.stanford.edu/class/cs231a/lectures/lecture6_stereo_systems.pdf

Geometry in stereo (a visual overview)

- How do you actually get depth?
- If you find correspondences x and x’, the quantity x− x’ is known as the **disparity**
- By similar triangles, you can convince yourself that **disparity is inversely proportional to depth**

- Problem statement, reformulated: **Find the disparity for every pixel in the left (or right) image by finding matches in the right (or left) image**

$$\frac{x - x'}{O - O'} = \frac{f}{z}$$



$$disparity = x - x' = \frac{B \cdot f}{z}$$

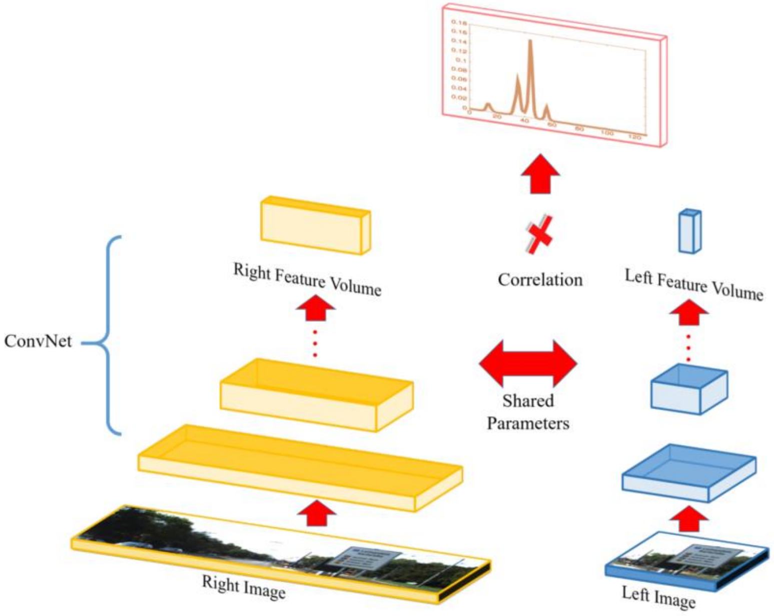
Credit: L. Shapiro
Source: https://courses.cs.washington.edu/courses/cse455/16wi/notes/11_Stereo.pdf

Practical example: KITTI

Source: KITTI Stereo 2015 Training Set [5]

Efficient Deep Learning for Stereo Matching [1]

W. Luo, A. Schwing, and R. Urtasun.
In *CVPR* 2016.



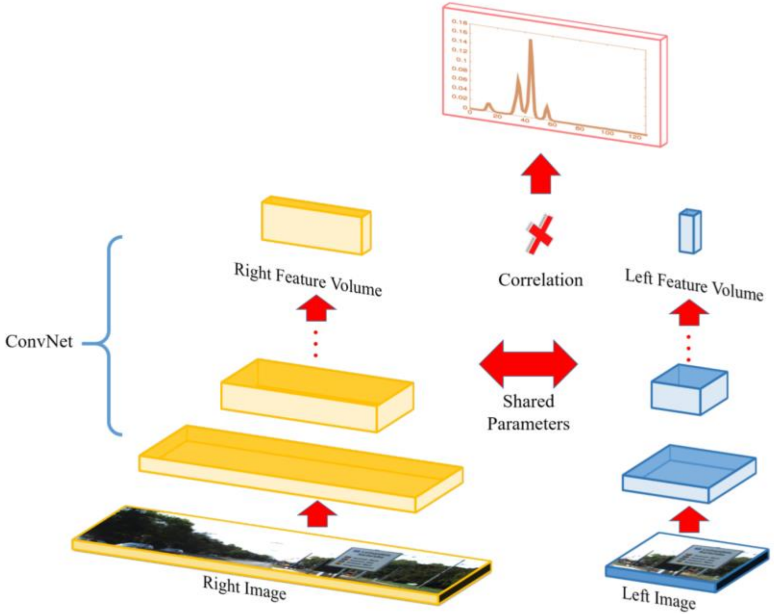
Features for stereo correspondence

- Finding a good match is hard
- What is a good feature?
- **Can we learn the features instead?**

Source: https://upload.wikimedia.org/wikipedia/en/3/3b/Stereo_empire.jpg

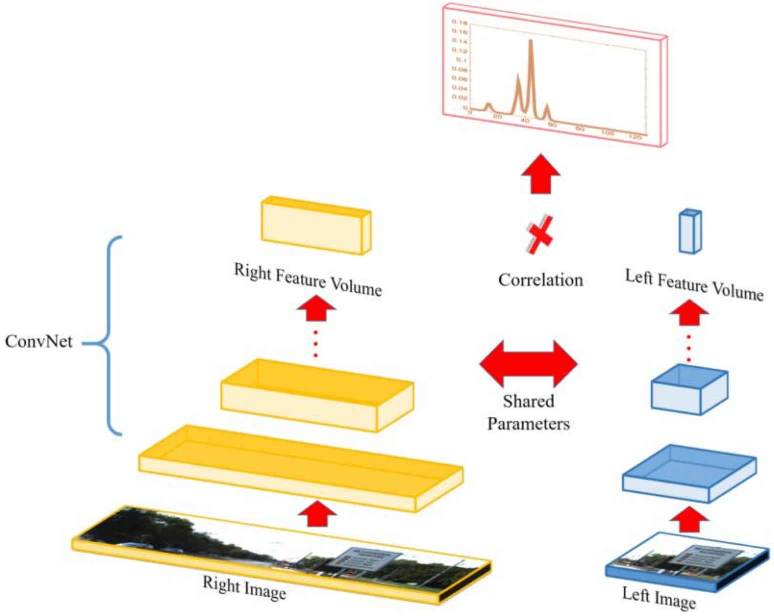
Key idea

- Construct a neural network that takes input images (left/right) and produces representative features that can be used to find stereo correspondences efficiently.



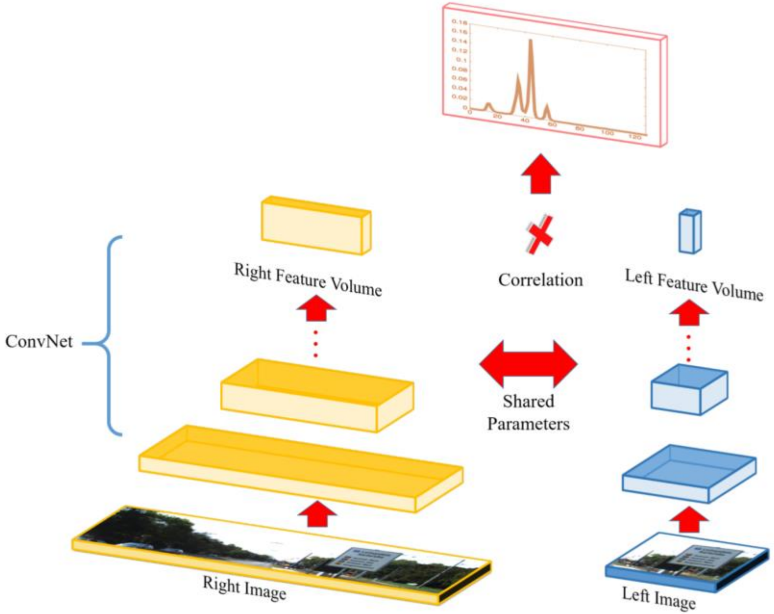
Network architecture

- Siamese network
  - Shared weights enforce similar features are learned on both left/right images
- Several convolution layers
  - Paper implements a fairly vanilla network
  - Several variants are tested; the key behind the choices of kernel size / stride is the effective receptive field



Training

- Pose this as a multi-class classification problem
  - Differentiating from earlier work which poses as binary classification [3]
- Left image patch is equal to the receptive field
  - Final feature volume after passing through the network is 1 x 1 x 64 (H x W x C)
- Right patch is larger to accommodate more context across range of possible disparities
  - Final feature volume is 1 x S x 64 (S is total number of search locations)
- **Inner product** of left feature with every spatial location of right feature ⇒ S scores



Training

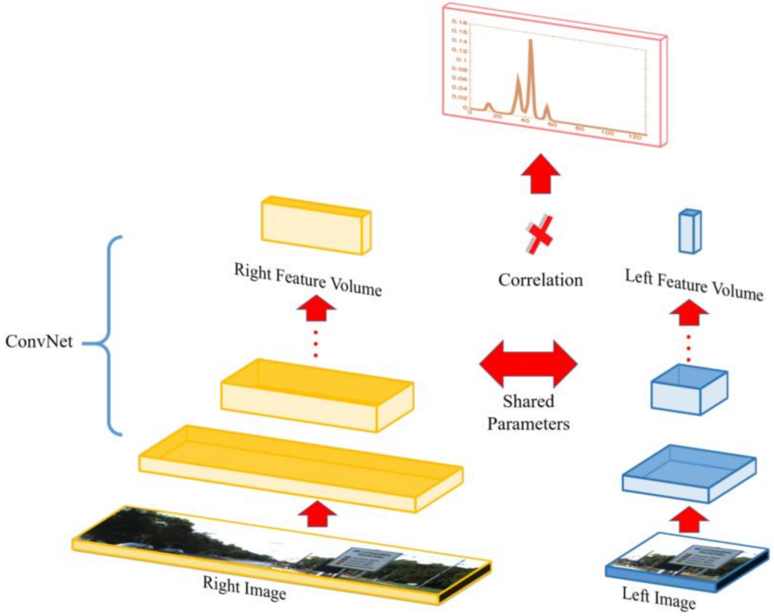
- Multi-class cross entropy loss over these S scores
- Each class is an actual spatial bin
- Probability mass is diffused across ground truth bin +/- 2 bins, to allow for some ambiguity

$$\min_{\mathbf{w}} \sum_{i,y_i} p_{\text{gt}}(y_i) \log p_i(y_i, \mathbf{w})$$

$$p_{\text{gt}}(y_i) = \begin{cases} \lambda_1 & \text{if } y_i = y_i^{GT} \\ \lambda_2 & \text{if } |y_i - y_i^{GT}| = 1 \\ \lambda_3 & \text{if } |y_i - y_i^{GT}| = 2 \\ 0 & \text{otherwise} \end{cases}$$

Testing

- Does not have to take the same form as training
- **Efficiency comes from enforcing that similarity between features is measured by their inner product**
- Can compute all these features at once on left/right images
- Produce a **cost volume** by computing similarity across multiple disparities
  - H x W x D, where D is number of disparity candidates



Smoothing

- How to get final result?
- Could just take most likely assignments across this volume
- Drawback: These predictions tend to be rough (no smoothness prior)
- Can smooth in various ways through averaging, energy minimization (semi-global block matching), slanted-plane, and other post-processing techniques

Unary	CA	SGM[30]	Post[30]	Slanted[27]	Ours(9)	Ours(19)	Ours(29)	Ours(37)	MC-CNN-acrt[29]	MC-CNN-fast[29]
✓					15.25	8.95	7.23	7.13	12.45	14.96
✓	✓				11.43	8.00	6.60	6.58	7.78	-
✓	✓	✓			5.18	4.74	4.62	4.73	3.48	5.05
✓	✓	✓	✓		4.41	4.23	4.31	4.38	3.10	4.74
✓	✓	✓	✓	✓	4.25	4.20	4.14	4.19	3.11	4.79

Evaluation

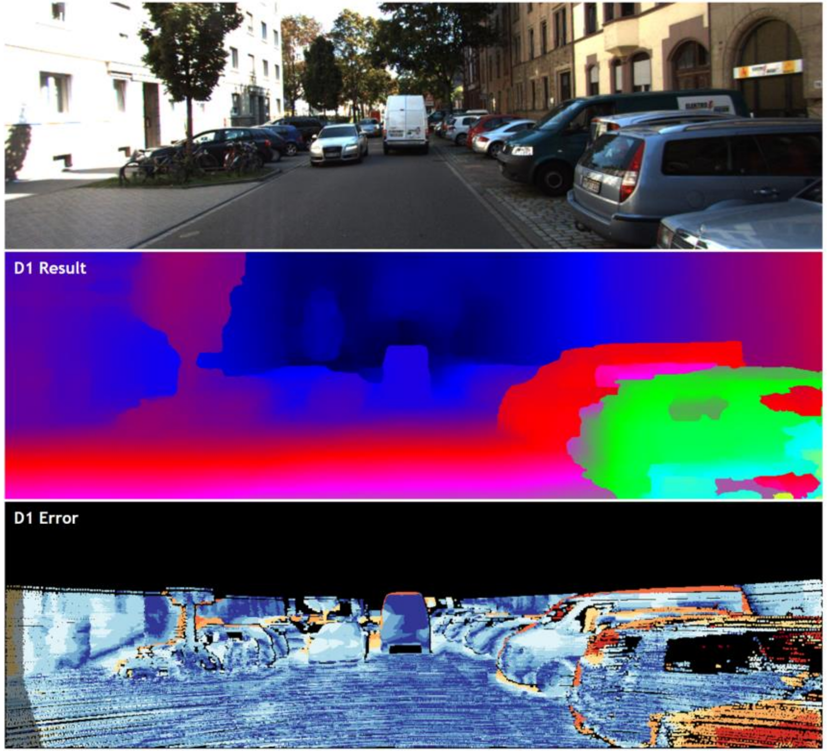
- Train and test on KITTI only (training set has 200 image pairs)
- Very straightforward training procedure
- Competitive results (on D1 error reported by testbench) with significant speed-up
  - Highlighting similar approach of [3]

	All/All			All/Est			Noc/All			Noc/Est			Runtime (s)
	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	D1-bg	D1-fg	D1-all	
MBM [9]	4.69	13.05	6.08	4.69	13.05	6.08	4.33	12.12	5.61	4.33	12.12	5.61	0.13
SPS-St [27]	3.84	12.67	5.31	3.84	12.67	5.31	3.50	11.61	4.84	3.50	11.61	4.84	2
MC-CNN [30]	2.89	8.88	3.89	2.89	8.88	3.88	2.48	7.64	3.33	2.48	7.64	3.33	67
Displets v2 [12]	3.00	5.56	3.43	3.00	5.56	3.43	2.73	4.95	3.09	2.73	4.95	3.09	265
Ours(37)	3.73	8.58	4.54	3.73	8.58	4.54	3.32	7.44	4.00	3.32	7.44	4.00	1

Source:http://www.cvlibs.net/datasets/kitti/eval_scene_flow_detail.php?benchmark=stereo&result=b54624a9eed52b4c8e6c76b411179dce4bd7d4d8

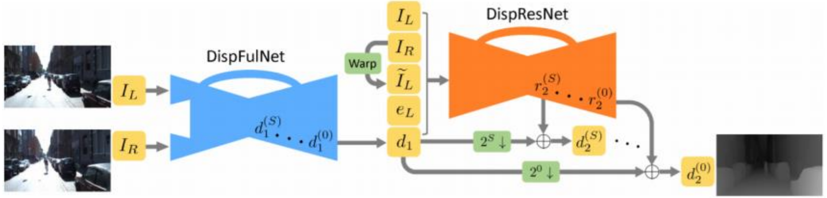
Sample output

- From submission to KITTI 2015 stereo benchmark
- Middle is prediction, bottom is error
- Even small differences in prediction can result in large disparity errors



Cascade Residual Learning: A Two-stage Convolutional Neural Network for Stereo Matching [2]

J. Pang, W. Sun, J. SJ. Ren, C. Yang, and Q. Yan. In *CVPR* 2017.

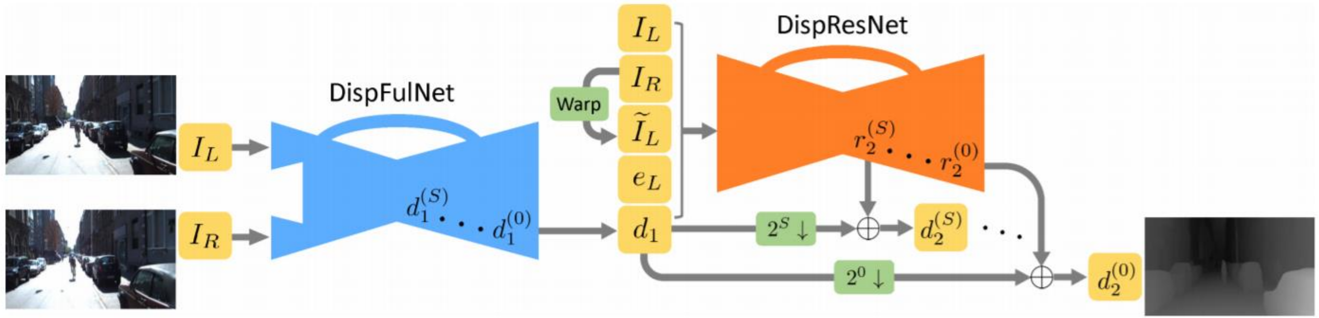


Another approach

- This can be posed as a classification problem, why not regression?
  - Based on idea of DispNet presented in [4]
  - Feed two images in, get dense disparity prediction out
- Advantage:
  - Note that in previous approach, smoothing was still necessary for good results
  - We could try to make the entire prediction process end-to-end learnable
- Disadvantage:
  - Do not get to explicitly incorporate geometric priors

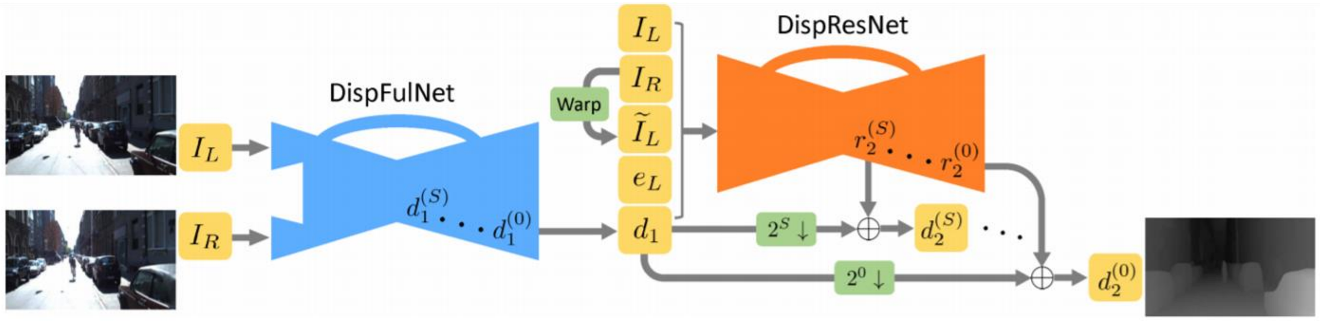
Architecture

- Two parts
  - DispFulNet: Predict initial disparity
  - DispResNet: Refine the prediction



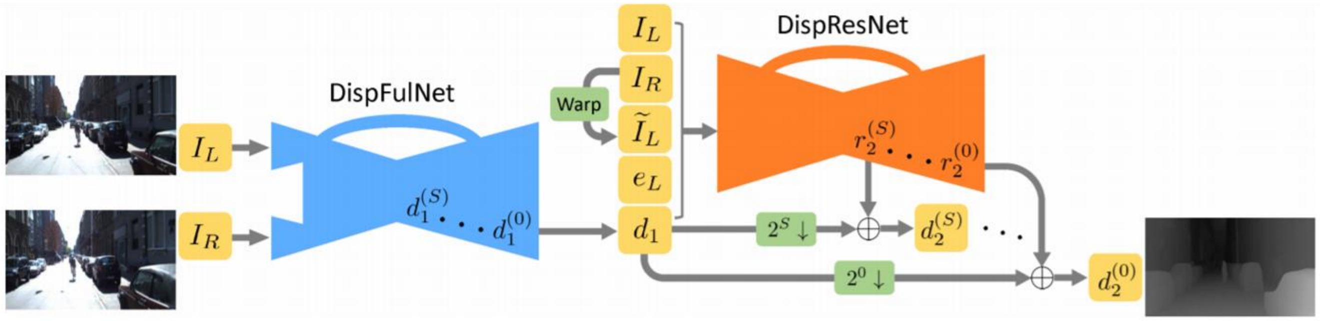
Architecture

- DispFulNet
  - Based on DispNet [4]
  - Encoder/decoder architecture; take left/right images as input, share lower level features, combine, predict
  - Train with L_1 loss against ground truth disparity map
  - Make predictions at multiple scales during decode ($d_1^{(S)}$, …, $d_1^{(0)}$)
  - Produce initial disparity map d_1



Architecture

- DispResNet
  - Idea from ResNet
  - Given initial prediction, have another network predict the **residuals**
  - Again, produce predictions at multiple scales to incorporate more supervision
  - Output is final disparity



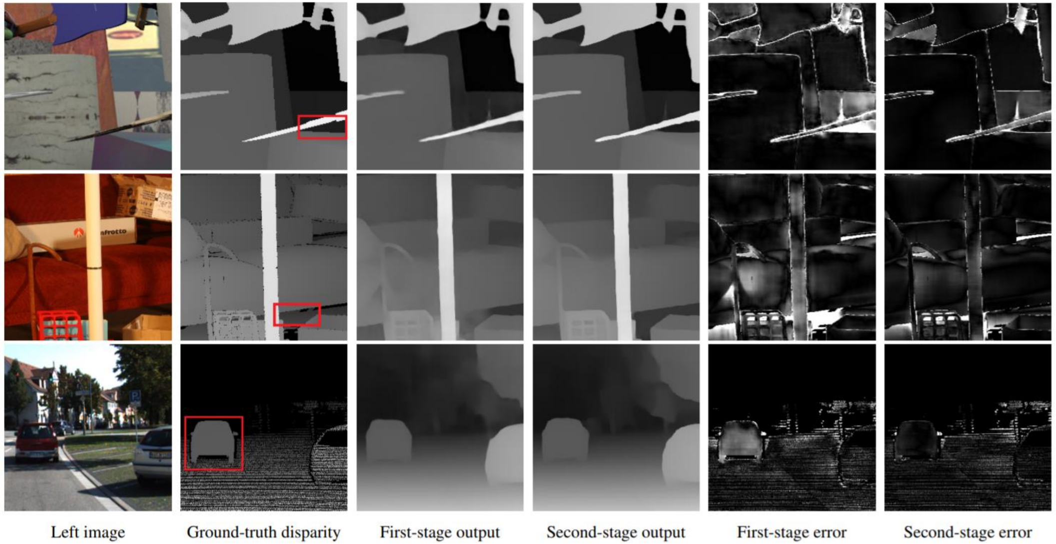
Evaluation

- Train on a lot of data
    - FlyingThings3D: Synthetic dataset with 22k+/4k+ train/test examples
    - Finetuning on KITTI
- Test on FlyingThings, Middlebury, and KITTI
- Currently #8 on KITTI 2015 stereo leaderboard!
    - Keep in mind submitted March 2017

	Method	Setting	Code	D1-bg	D1-fg	D1-all	Density	Runtime	Environment	Compare
1	PSMNet			**1.86 %**	4.62 %	**2.32 %**	100.00 %	0.41 s	Nvidia GTX Titan X	
2	Kandao			2.14 %	3.45 %	2.36 %	100.00 %	0.22 s	Nvidia GTX 1080 Ti	
3	iResNet-i2e2			2.10 %	3.64 %	2.36 %	100.00 %	0.25 s	Nvidia Titan X (Pascal)	
Z. Liang, Y. Feng, Y. Guo and H. Liu: Learning Deep Correspondence through Prior and Posterior Feature Constancy. arXiv preprint arXiv:1712.01039 2017.										
4	pVGG			2.25 %	3.40 %	2.44 %	100.00 %	0.15 s	Nvidia titan x (Python)	
5	SegStereo			2.16 %	4.02 %	2.47 %	100.00 %	0.6 s	Nvidia GTX Titan X	
6	iResNet-i2			2.35 %	**3.23 %**	2.50 %	100.00 %	0.12 s	Nvidia Titan X (Pascal)	
7	DeepStereo			2.16 %	4.72 %	2.59 %	100.00 %	0.9 s	Titian X	
8	CRL		code	2.48 %	3.59 %	2.67 %	100.00 %	0.47 s	Nvidia GTX 1080	
J. Pang, W. Sun, J. Ren, C. Yang and Q. Yan: Cascade residual learning: A two-stage convolutional neural network for stereo matching. ICCV Workshop on Geometry Meets Deep Learning 2017.										
9	DeepStereo			2.15 %	5.88 %	2.77 %	100.00 %	0.9 s	Titian X	
10	3DResStereo			2.27 %	5.50 %	2.80 %	100.00 %	1.3 s	Titian X	
11	GC-NET			2.21 %	6.16 %	2.87 %	100.00 %	0.9 s	Nvidia GTX Titan X	
A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach and A. Bry: End-to-End Learning of Geometry and Context for Deep Stereo Regression. Proceedings of the International Conference on Computer Vision (ICCV) 2017.										
12	LRCR			2.55 %	5.42 %	3.03 %	100.00 %	49.2 s	1 core @ 2.5 Ghz (C/C++)	
ERROR: Wrong syntax in BIBTEX file.										
13	RecResNet			2.46 %	6.30 %	3.10 %	100.00 %	.1 s	@ (Nvidia GTX Titan X)	
14	DRR			2.58 %	6.04 %	3.16 %	100.00 %	0.4 s	Nvidia GTX Titan X	
S. Gidaris and N. Komodakis: Detect, Replace, Refine: Deep Structured Prediction For Pixel Wise Labeling. arXiv preprint arXiv:1612.04770 2016.										
15	MS-GANs			2.53 %	6.64 %	3.21 %	100.00 %	0.9 s	Nvidia GTX Titan X	
16	E2ES-Net			2.61 %	6.43 %	3.25 %	100.00 %	0.5 s	Nvidia GTX 1080 (Python)	
17	SSMFCR			3.16 %	4.11 %	3.32 %	100.00 %	0.09 s	1 core @ 2.5 Ghz (C/C++)	
18	gcn			2.62 %	6.85 %	3.32 %	100.00 %	0.9 s	1 core @ 2.5 Ghz (C/C++)	
19	SsSMnet			2.70 %	6.92 %	3.40 %	100.00 %	0.8 s	P100	
Y. Zhong, Y. Dai and H. Li: Self-Supervised Learning for Stereo Matching with Self-Improving Ability. arXiv:1709.00930 2017.										
20	L-ResMatch		code	2.72 %	6.95 %	3.42 %	100.00 %	48 s	1 core @ 2.5 Ghz (C/C++)	
A. Shaked and L. Wolf: Improved Stereo Matching with Constant Highway Networks and Reflective Loss. arXiv preprint arxiv:1701.00165 2016.										

Evaluation

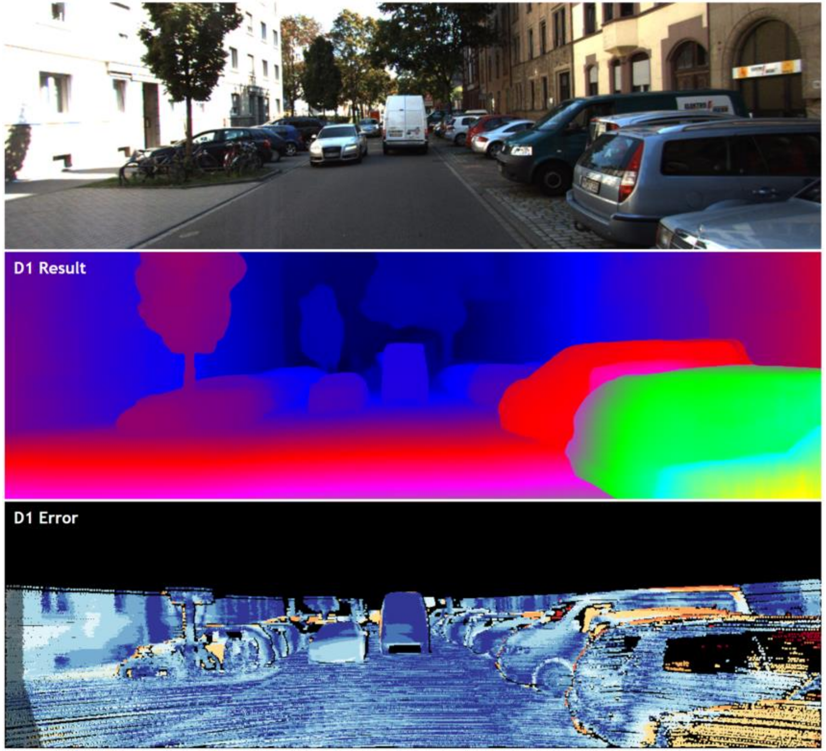
- Qualitative assessment of refinement



Source:http://www.cvlibs.net/datasets/kitti/eval_scene_flow_detail.php?benchmark=stereo&result=f791987e39ecb04c1eee821ae3a0cd53d5fd28c4

Sample output

- From submission to KITTI 2015 stereo benchmark
- Middle is prediction, bottom is error
- Generally smoother outputs with ability to define sharp boundaries for objects



Questions

References

[1] W. Luo, A. Schwing, and R. Urtasun, “Efficient deep learning for stereo matching,” in International Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[2] J. Pang, W. Sun, J. S. Ren, C. Yang, and Q. Yan, “Cascade residual learning: A two-stage convolutional neural network for stereo matching,” in ICCV Workshop on Geometry Meets Deep Learning, Oct 2017.

[3] J. Zbontar and Y. LeCun, “Stereo matching by training a convolutional neural network to compare image patches,” Journal of Machine Learning Research, vol. 17, pp. 1–32, 2016.

[4] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox, “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation,” in IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016. arXiv:1512.02134.

[5] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in Conference on Computer Vision and Pattern Recognition (CVPR), 2015.