EE 702 Computer Vision | Paper Review

Meet Pragnesh Shah | 13D070003 | meetshah1995@ee.iitb.ac.in

March 18, 2016

Paper Title: Computing the Stereo Matching Cost with a Convolutional Neural Network
Authors: Yann LeCun, Jure Zbontar
Conference: CVPR 2015 [pdf : http://arxiv.org/abs/1409.4326] [code : https://github.com/jzbontar/mc-cnn]

1 Abstract

We present a method for extracting depth information from a rectified image pair. We train a convolutional neural network to predict how well two image patches match and use it to compute the stereo matching cost. The cost is refined by cross-based cost aggregation and semiglobal matching, followed by a left-right consistency check to eliminate errors in the occluded regions. Our stereo method achieves an error rate of **2.61 %** on the KITTI stereo dataset and is currently (August 2014) the top performing method on this dataset.

2 Brief Explanation

The method presented by the authors leverages convolutional neural networks [CNN] for cost computation of the image patches for consequent stereo matching. It is currently the **top performer** on the **KITTI stereo dataset**. The method works as follows:

- A dataset is initially generated consisting of 9x9 patches of images from the left and right images such as $I_{\text{left}}(i,j)$ and $I_{\text{right}}(i-d+offset,j)$ where d is the known disparity and the $offset$ is a **signed** hyperparameter of the system. This training set is later used to train a 8 layered CNN. The CNN output is a intermediate cost of the system.
- The cost is aggregated using a cross based method, wherein a cross [] is defined for each pixel such that the arms of cross extend in all 4 directions of the pixel until the sum of absolute difference in intensities exceeds a threshold or the maximum arm length (predefined hyperparameter) is exceeded. The collection of all pixels in the crosses of neighbouring pixels is used to aggregate the cost for a single pixel. The cost is then calculated iteratively for 4 iterations to give a intermediate cost C_{CBCA}.
- Semi-global matching is then performed on the cost obtained above in which the cost is penalized differentially (using hyperparameters) for different local changes in disparities, this ensures that the jumps in disparities coincide with edges. This energy function is minimized in 4 directions and the arg min of the costs in all directions is taken as the output disparity map. Later left right consistency checks are performed to identify the mismatches and occlusions. The occlusions and mismatches are then handled by averaging and interpolating over windows of other pixels with no occlusions.
- Lastly sub-pixel enhancement is performed using quadratic curve fitting and then a median and bilateral filter is applied to the output disparity map to get the final disparity map D_{final}.

3 Critique

3.1 Merits

- This method generates a cross (over which cost is aggregated) which is adaptive (in terms of size) to each pixel and the window is larger for neighbourhoods with similar disparities and vice versa. Hence this cross-based cost aggregation helps in reducing formation of textures (disparity confusion) at such pixels which is observed in naive stereo algorithms.

- This method also incorporates textures that may arise due to directional gradients (edges) by using semi global matching in 4 directions which ensures that the cost is penalized for large differences in disparities for neighbouring pixels. Hence it ensures that jumps in disparities coincide with the edges in the images.
- A quadratic neighbourhood based curve fitting using sub-pixel enhancement helps in regaining the resolution lost in averaging.
- A very fine attempt with very little computational overhead to remove occlusions, disparity confusion (at edges) and mismatches using left right consistency, refinement, filtering and interpolation.
- The cost calculation is iterative and hence the costs have smoother gradients and hence give rise to a smoother disparity map with lesser spikes.
- The matching accuracy increases with increase in training sets as shown by the author which is a positive indicator of the robustness of the CNN learning architecture.
- This method effectively uses variants of several different techniques used by references in order to eliminate textures, occlusions and mismatches from the final disparity map.

3.2 Demerits

- The author explicitly mentions that the CNN architecture changes with the dataset. The architecture is thus not generic enough to learn features from some input and effectively use them on a wide spectrum of input images. This implies that the training of the network ($5hrs$) has to be done every time the input type changes.
- The implementation and methods suggested by the authors is not a **plug-and-play** method and needs to be trained (supervised learning) on a dataset with known ground truths before it can be used. Further more the performance of the architecture on a set of input other than the training input is yet to be explored.
- The network may be prone to over-fitting (general problem with Convnets) and might learn rot learn the features specific to the training dataset and might not perform with similar accuracy on other datasets.
- This method involves too many parameters and hyperparameters for a relatively simple problem which are dependent on the dataset and hence parameter tuning is required every time the dataset type changes.
- The training and testing was done on the GPUs taking **100s** per image pair. This method thus is not a practical method and the author is yet to explore ways to reduce the runtime in order to make it practically feasible.
- The increment in accuracy was very little when compared to the large amount of time taken. Other methods with competitive and similar accuracy perform the same task with significantly lesser time.
- Last but not the least, the method and all it's accuracy gains are obtained on grayscale images. The experiments on a RBG dataset are yet to be done. Given the $3x$ amount of data in RGB images, the time taken and accuracy gains might also suffer.

References

[1] Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. Pattern Analysis and Machine Intelligence, IEEE 2008.

[2] Yamaguchi, K., McAllester, D., and Urtasun, R. (2014). Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In Computer Vision–ECCV 2014. Springer.

[3] Zhang, K., Lu, J., and Lafruit, G. (2009). Cross-based local stereo matching using orthogonal integral images. Circuits and Systems for Video Technology, IEEE Transactions