Computing the Stereo Matching Cost with a Convolutional Neural Network

Seminar *Recent Trends in 3D Computer Vision*

Markus Herb
Supervisor: Benjamin Busam

Chair for Computer Aided Medical Procedures
Technische Universität München

Abstract. This paper presents a novel approach to the problem of computing the matching-cost for stereo vision. The approach is based upon a Convolutional Neural Network that is used to compute the similarity of input patches from stereo image pairs. In combination with state-of-the-art stereo pipeline steps, the method achieves top results in major stereo benchmarks. The paper introduces the problem of stereo matching, discusses the proposed method and shows results from recent stereo datasets.

1 Introduction

3D depth perception has been a long term goal within computer vision with stereo vision in particular being an area of research for several decades [1,2]. The objective in stereo vision is to reconstruct the 3D depth information of a scene from the input images of two cameras at different viewpoints and the known camera geometry.

Despite the fact that great progress has been made over the years in this field, the topic continues to be an active area of research. This is largely due to the great number of potential applications, especially in robotics, such as autonomous driving, but also for medical applications including surgical robotics or X-ray imaging. While there exist alternative approaches to depth-perception, such as RGB-D cameras including the Microsoft Kinect, stereo vision is especially appealing since it only depends on passive camera sensors, making it well suited for outdoor-use and large ranges.

This paper presents a novel approach to the problem of stereo matching, i.e. the finding of matching points in corresponding stereo image pairs, proposed by Žbontar and LeCun [3]. The method is based upon a Convolutional Neural Network (CNN), a machine learning technique used with great success in recent years to tackle challenging computer vision problems.

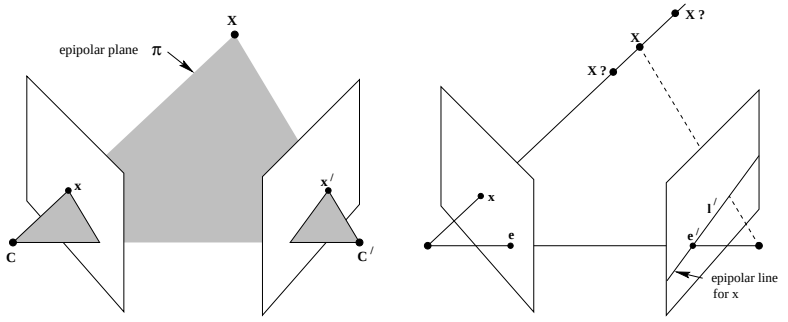
The rest of this paper is structured as follows. In section 2, a brief introduction into the underlying theory of general stereo vision techniques will be given. Section 3 contains a concise overview over the general concept of Convolutional Neural Networks. In section 4, a compact review of current state-of-the-art stereo

pipelines in general and matching cost computation in particular is given. Following that, the methodology of the approach is presented in section 5 with results following in section 6.

2 Stereo Vision

2.1 Epipolar geometry

In order to understand how the 3D scene structure can be reconstructed from a pair of stereo images, it is sensible to examine the special geometry of such a multi-view camera system first, also known as epipolar geometry. Figure 1 schematically depicts a stereo camera setup and the involved epipolar constraints.



(a) Projection of a 3D point X onto x and x' for distinct cameras C and C'

(b) For a known projection x, search for a corresponding x' is restricted to the epipolar line l'

Fig. 1: Epipolar geometry [4].

The system is built from the two cameras denoted by their respective camera centers C and C'. A 3D scene point X will be projected onto the image-planes at points x and x' respectively. Conversely, having a known pair of projections x and x' recovered solely from the captured images, one can then compute the coordinates of the original point X by finding the intersection of the two rays passing through C and x as well as C' and x' respectively, a method known as triangulation. Hence it is sufficient to have a pair of projections x and x' of a single point in space in order to recover the 3D coordinates of that point, assuming the interior and exterior camera parameters are known. Such a pair of projected points in both images is also known as a stereo correspondence.

The key challenge within stereoscopic vision thus is to find stereo correspondences in the images. While the search for such correspondences is fairly hard in general, epipolar geometry imposes certain constraints regarding where in the images such correspondences can occur.

For a given x in one image, it is known that the corresponding x' must lie on the other image plane as well as on the same *epipolar plane* π, which is spanned by C, C' and X. The only points that fullfil both constraints lie on the intersection of both planes, which is a line known as *epipolar line* l'. Hence the search-space for potentially matching correspondences in the other image is reduced to a single line.

For a more in-depth introduction to epipolar geometry, the interested reader is referred to [4].

2.2 Rectification, Disparity & Matching

Generally, the epipolar lines within an image may occur in arbitrary directions, i.e. they are not aligned to a specific axis and not parallel [5].

To make the search along epipolar lines less cumbersome, a *rectification* can be applied to the stereo image pairs. This transforms the images in such a way that all epipolar lines are parallel to the horizontal axis and vertically aligned with the corresponding epipolar lines of the other image [5]. The rationale for working with rectified images is that search for correspondences can be performed within each scanline of the image, as all pixels within one horizontal row of pixels lie on the same epipolar line. Thus the computational effort for searching correspondences along epipolar lines is greatly reduced.

Another benefit gained from using rectified images is that matching points can be specified by three parameters only. Instead of specifying the full image coordinates of the corresponding points in both images, it is sufficient to indicate the horizontal offset of the projections between both images, since the vertical coordinates are ensured to be identical due to the vertical alignment of corresponding epipolar lines. Hence a pair of point correspondences can be defined using the pixel location in one image $x = (u, v)$ and a corresponding horizontal offset, the *disparity* d, to the corresponding projection $x' = (u + sd, v)$ in the other image, with $s \in \{-1, 1\}$ chosen to ensure d is always positive [2].

From the disparity of a pair of projections, the distance or *depth* of the original scene point can be reconstructed using

$$z = \frac{f \cdot B}{d}$$

where z denotes the depth, f the focal length, B the distance between the camera centers and d the disparity [3]. Thus, the depth is inversely proportional to the disparity, which leads to the fact that it is sufficient to recover the disparity value for each point in the image in order to be able to reconstruct the depth.

3 Convolutional Neural Networks

The concept of Convolutinal Neural Networks (CNNs) has been proposed by LeCun *et al.* [6,7] and can be considered an extension to classical Neural Networks. In addition to these, CNNs also contain *convolutional layers* and may have additional sub-sampling layers.

In contrast to fully-connected neural network layers, where each neuron of one layer is connected to all neurons of the previous layer, each convolutional layer neuron is connected to a spatially connected subset of neurons in the previous layer. By sharing the connection-weights among sets of neurons and arranging them spatially to form *feature-maps*, the network effectively learns the filters used for a convolution operation, hence the name *Convolutional* Neural Network. Due to their convolutional nature, CNNs are well suited for image processing tasks. The interested reader is referred to [7] for a deeper introduction to the concept of CNNs.

With the advent of powerful GPUs for general purpose computations, Convolutional Neural Networks have gained a lot of traction within the computer vision community. CNNs have since been used with great success for tasks such as image classification [8] and more recently also for segmentation [9] or to predict optical flow [10].

4 Related Work

4.1 Stereo Pipeline

According to Scharstein and Szeliski [2], a stereo pipeline can usually be decomposed into four steps, namely (1) matching cost computation, (2) cost aggregation, (3) disparity computation and finally (4) disparity refinement.

In the matching-cost step, for each pixel (x, y) and each disparity d under consideration, a matching cost is computed to measure the similarity of the point (x, y) in one image and $(x + sd, y)$ in the other. This cost is then stored in a 3D cost volume $C(x, y, d)$, which is also known as *Disparity Space Image* (DSI) [11].

After that, the cost in the DSI is usually aggregated within small support windows around each pixel in order to make the cost computation more robust.

In the third step, the DSI is then reduced to a single disparity estimate for each pixel. Depending on the method, there may be some optimization of the whole cost volume first, in order to enforce a smooth disparity and eliminate errors. Following that, for each pixel (x, y), one looks for the disparity d at which the cost $C(x, y, d)$ is minimized and stores that disparity in the disparity map at $D(x, y)$.

Step 4 refines the computed disparity estimate, for instance through additional consistency checks and filtering. The refined disparity map then contains the most likely disparity for each pixel in the image, i.e. the disparity map represents the estimated mapping from points in one image to corresponding points in the other image.

Most recent works in the area of stereo vision focused on novel methods for disparity calculation, optimization and refinement, rather than the computation of the matching cost itself. In contrast to that, the MC-CNN approach specifically focuses on the matching-cost step, while state-of-the-art techniques are used for the rest of the pipeline.

4.2 Matching Cost Computation

The goal of the matching-cost computation step is to measure the similarity of two points in the images, hence a similarity measure is needed in order to compute the cost. Common similarity measures that can also be used for matching cost computation are absolute intensity differences (AD) or squared intensity differences (SD) [12], as well as Normalized cross-correlation (NCC) [13]. Other cost-computation methods include the census-transform [14] or the probabilistic Mutual Information approach [15]. Recent approaches also use a weighted combination of different measures, such as absolute differences and census-transform [16] or gradient and census-transform [17].

5 MC-CNN

The proposed MC-CNN is a novel way to compute the cost in the matching cost computation step as part of a full stereo pipeline. Figure 2 depicts the pipeline of the approach in a slightly simplified manner. Within the pipeline, the MC-CNN network serves the purpose of computing the matching cost for each point and each potential match in the other image, so for each pixel (x, y) and each disparity d. The matching cost for a pair of points is computed by taking patches centered around the points of consideration and feeding these to the network, with the output cost being stored in the Disparity Space Image.

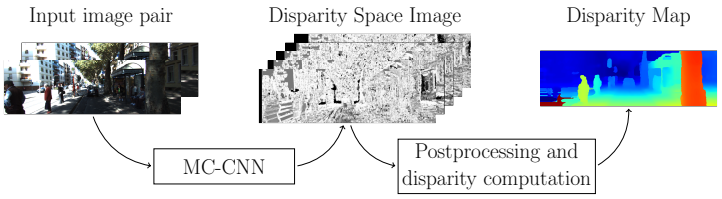


Fig. 2: MC-CNN pipeline from input images to final disparity map output.

The MC-CNN network layout is discussed in the following subsection, with the postprocessing and disparity computation steps being outlined in subsection 5.3.

5.1 Network Architecture

Figure 3 shows the general network architecture. Inputs of the network are two image patches from left and right image respectively, for which the similarity or matching cost shall be computed. Output of the network is a probability distribution over the two classes *good match* and *bad match*, of which the bad match probability is taken as the final matching cost.

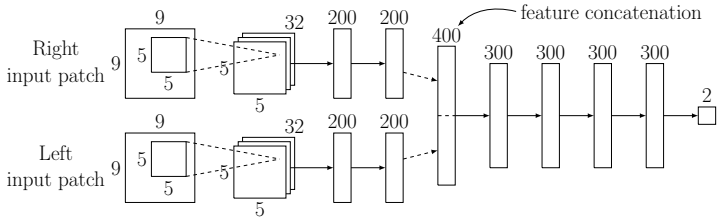


Fig. 3: MC-CNN network architecture [3].

The network is divided into two major parts. The first stage up to the concatenation layer makes up the feature extraction part, where features are extracted from the left and right patch. Each image patch is handled by an independent path containing first a convolutional layer with two fully-connected layers following afterwards. It is important to note that the layer-weights in both paths are tied, meaning that the weights are the same in both paths, hence the identical feature extraction takes place in both paths, but for different input data.

After the feature-extraction part, the computed feature vectors are concatenated to a combined feature-vector twice the size. This combined vector is then passed into the second network stage containing four fully-connected network layers ending up in a softmax-classifier output layer. The second part is responsible for computing the similarity of the feature vectors of the individual patches. After all layers in the first and second stage, except for the softmax output, a rectified linear activation function [18] is applied.

5.2 Network Training

As the used Convolutional Neural Network is a supervised machine learning technique, it is necessary to train the network before use. Ground-truth data to teach the network what comprises a *good match* versus a *bad match* is obtained from recent stereo datasets such as the KITTI 2012 [19] dataset.

From the ground-truth disparities, positive and negative training examples are generated, where each example consists of a pair of patches. Positive examples are computed by taking patches at the true disparity and adding a slight random horizontal offset to increase robustness. Negative examples are obtained similarly, but using a larger random offset such that the patches are still roughly from the same area, but do not match directly anymore.

5.3 Post-processing and Disparity Computation

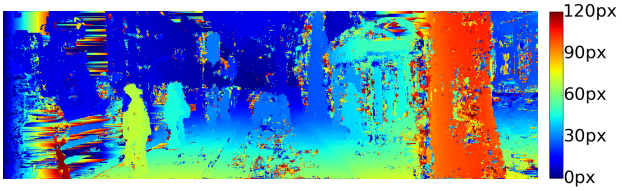
While the Disparity Space Image generated by the CNN could be used directly to predict the disparity map by minimizing the cost along the disparity direction in the DSI, a number of post-processing steps are applied in order to improve the results. All of these post-processing steps are state-of-the-art and are

not a particular contribution of the paper. The postprocessing pipeline is very much inspired by the approach presented by Mei *et al.* [16] including cross-based cost-aggregation (CBCA) [20] and semi-global matching (SGM) [21]. Additional checks and refinements such as a left-right consistency check are also applied.

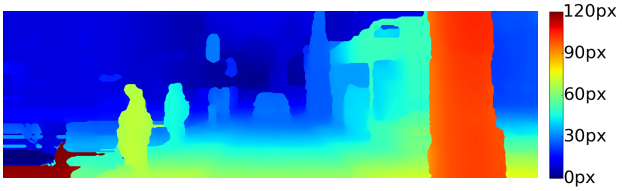
6 Results

An example for the disparity prediction obtained using the MC-CNN method is shown in figure 4. Figure 4b shows the resulting disparity map computed directly from the DSI-output of the CNN. In this intermediate result, a large amount of artifacts and mismatches is visible, especially in the overexposed areas on the left (street, building) due to low texture in these regions. Nevertheless the disparity estimate is already quite accurate in many areas.

(a) Left input image from dataset.



(b) Disparity map $D(x, y)$ directly after CNN (without post-processing).



(c) Final disparity map $D(x, y)$ after all post-processing.

Fig. 4: Disparity results for an example image pair from KITTI 2015 dataset [22]. Results have been computed using the slightly revised MC-CNN-acrt journal architecture [23].

The final and improved result after all post-processing steps is depicted in figure 4c. The disparity estimate is heavily smoothed and practically all mismatches have been removed, resulting in an excellent disparity estimate where all foreground objects are clearly distinguishable.

Table 1: KITTI 2012 [19] benchmark results (Top 5) as of November 5, 2015

Rank	Method	Out-Noc	Avg-Noc	Runtime	Environment
1	MC-CNN-acrt [23]	2.43 %	0.7 px	67 s	Nvidia GTX Titan X
2	Displets [24]	2.47 %	0.7 px	265 s	>8 cores @ 3.0 Ghz
3	MC-CNN [3]	2.61 %	0.8 px	100 s	Nvidia GTX Titan
4	PRSM [25]	2.78 %	0.7 px	300 s	1 core @ 2.5 Ghz
5	SPS-StFl [26]	2.83 %	0.8 px	35 s	1 core @ 3.5 Ghz

In addition to the good subjective visual results, the method has been submitted to a number of state-of-the-art stereo vision benchmarks and ranks very well in them. As displayed in table 1, MC-CNN currently ranks third in the KITTI 2012 benchmark with an error of more than 3px in 2.65% of the pixels in non-occluded areas. It is important to note that the second-best method *Displets* also uses the MC-CNN cost computation, but applies a specialized post-processing to increase the accuracy. Finally, the top-ranking method is currently MC-CNN-acrt, which is an improved journal version of MC-CNN by the same authors. At the time of writing, this method also ranks first in the KITTI 2015 [22] and Middlebury 2014 [27] benchmarks.

7 Conclusions

The presented paper proposes an entirely new approach for matching cost computation in a stereo vision pipeline. It is the first method to take an existing stereo vision pipeline and replace one step thereof with a Convolutional Neural Network. While the network achieves stunning results as-is, additional state-of-the-art pipeline steps are still needed as post-processing steps in order to achieve competitive results. Including this post-processing, the method achieves top-results in all recent major stereo vision benchmark suites.

One of largest disadvantages of the proposed method is the fairly high computational effort required to compute the cost, taking up to minutes on recent high-end GPUs. This issue has already been addressed in a revised journal version of the presented paper, which features an improved feature extraction and an additional *fast* network variant that reduces the computation time to the range of seconds while achieving almost the same accuracy.

Despite the great success in current benchmarks, there are various opportunities to increase the accuracy even further. An approach could be to use multi-scale CNNs [28] to incorporate more global information into the cost computation. As also suggested by the authors, using larger training sets might be beneficial. This could be done by using synthetically generated ground-truth data in order to pre-train the network first and subsequently fine-tune the weights on real-world training data.

References

1. Barnard, S.T., Fischler, M.A.: Computational stereo. ACM Comput. Surv. **14**(4) (December 1982) 553–572
2. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: Stereo and Multi-Baseline Vision, 2001. (SMBV 2001). Proceedings. IEEE Workshop on. (2001) 131–140
3. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2015)
4. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press (2003)
5. Loop, C., Zhang, Z.: Computing rectifying homographies for stereo vision. In: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. Volume 1. (1999) 131 Vol. 1
6. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4) (December 1989) 541–551
7. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. (1998) 2278–2324
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., Weinberger, K., eds.: Advances in Neural Information Processing Systems 25. Curran Associates, Inc. (2012) 1097–1105
9. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. CVPR (to appear) (November 2015)
10. Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning Optical Flow with Convolutional Networks. In: IEEE International Conference on Computer Vision (ICCV). (December 2015)
11. Intille, S., Bobick, A.: Disparity-space images and large occlusion stereo. In Eklundh, J.O., ed.: Computer Vision — ECCV '94. Volume 801 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1994) 179–186
12. Hirschmuller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on. (June 2007) 1–8
13. Sinha, S.N., Scharstein, D., Szeliski, R.: Efficient high-resolution stereo matching using local plane sweeps. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. CVPR '14, Washington, DC, USA, IEEE Computer Society (2014) 1582–1589
14. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In Eklundh, J.O., ed.: Computer Vision — ECCV '94. Volume 801 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1994) 151–158
15. Kim, J., Kolmogorov, V., Zabih, R.: Visual correspondence using energy minimization and mutual information. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. (Oct 2003) 1033–1040 vol.2
16. Mei, X., Sun, X., Zhou, M., shaohui Jiao, Wang, H., Zhang, X.: On building an accurate stereo matching system on graphics hardware. In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. (Nov 2011) 467–474

17. Zhang, C., Li, Z., Cheng, Y., Cai, R., Chao, H., Rui, Y.: Meshstereo: A global stereo model with mesh alignment regularization for view interpolation. In: IEEE International Conference on Computer Vision (ICCV). (2015)
18. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In Fürnkranz, J., Joachims, T., eds.: Proceedings of the 27th International Conference on Machine Learning (ICML-10), Omnipress (2010) 807–814
19. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2012)
20. Zhang, K., Lu, J., Lafruit, G.: Cross-based local stereo matching using orthogonal integral images. Circuits and Systems for Video Technology, IEEE Transactions on **19**(7) (July 2009) 1073–1079
21. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30**(2) (Feb 2008) 328–341
22. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2015)
23. Žbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. arXiv preprint arXiv:1510.05970 (2015)
24. Güney, F., Geiger, A.: Displets: Resolving stereo ambiguities using object knowledge. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4165–4175
25. Vogel, C., Schindler, K., Roth, S.: 3d scene flow estimation with a piecewise rigid scene model. International Journal of Computer Vision (2015) 1–28
26. Yamaguchi, K., McAllester, D., Urtasun, R.: Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., eds.: Computer Vision – ECCV 2014. Volume 8693 of Lecture Notes in Computer Science. Springer International Publishing (2014) 756–771
27. Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nesic, N., Wang, X., Westling, P.: High-resolution stereo datasets with subpixel-accurate ground truth. In: 36th German Conference on Pattern Recognition (GCPR). (September 2014)
28. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. CoRR **abs/1406.2283** (2014)