A PARALLEL CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE FOR STEREO VISION ESTIMATION

[1]Yao Chou, [1]Dah-Jye Lee, [2]Dong Zhang, [1]Karina Hill

[1]Department of Electrical and Computer Engineering
Brigham Young University, Utah, 84602, U.S.A.
[2]School of Electronics and Information Technology
Sun Yat-sen University, Guangzhou, Guangdong, China 510006

ABSTRACT

Extracting depth information from the stereo image pair is a commonly used method in 3-D computer vision. For robotics and unmanned vehicle applications that require real-time performance, speed is often more important than accuracy. In recent years, Convolutional Neural Networks (CNNs) have shown great success in many computer vision applications including classification, segmentation, object detection, edge detection, and stereo vision estimation. Existing network architectures for stereo vision estimation predict very little information during the forward pass and are only able to calculate the disparity for one pixel at a time. In this paper, we propose a parallel architecture to speed up disparity map computation by simultaneously processing all pixels on one horizontal line. We train and test our network on five Middlebury datasets. Our parallel architecture achieves at least a 10x speedup compared to existing networks. Its accuracy is also very competitive.

Index Terms— Stereo Vision, CNN, Deep Learning, Parallel Architecture.

1. INTRODUCTION

Stereo vision plays a critical role in many robotic vision applications. It is crucial to extract depth information of the relative position of 3D objects, which separates occluding image components [1]. Stereo vision is widely used in many fields such as aerial surveys, autonomous vehicles, etc. A stereo vision algorithm computes the disparity for each pixel from two rectified images taken from the left and right cameras. Disparity refers to the difference in horizontal location of an object in both images, as shown in Fig. 1.

The key step in calculating the disparity is to find the corresponding points between the left and right images. A point *P(x, y, z)* in 3D space that projects to $P_L(x, y)$ in the left image will also appear at $P_R(x\text{-}d, y)$ in the right image. The object depth, Z (the distance between the object and the cameras) can be calculated by using Eq. (1) if the disparity *d* for this 3D point *P(x, y, z)* is known:

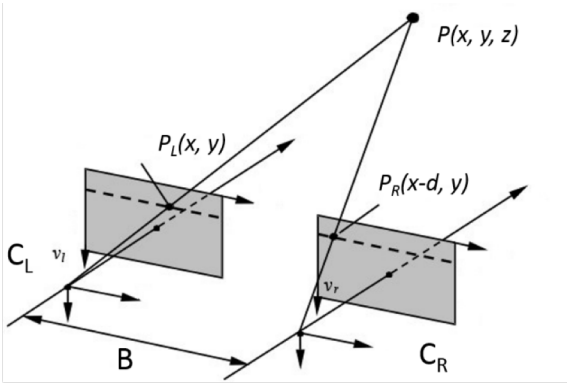

Fig. 1 Disparity calculation framework

$$z = \frac{fB}{d}, \qquad (1)$$

where f is the focal length of the camera and B is the baseline of the stereo system.

Convolutional Neural Networks (CNNs), forefront models of deep learning, are driving advancements in image analysis. CNNs improve the performance of whole-image classification [2] and feature extraction [3]. CNNs make use of the detailed features learned from training and can make a prediction for each pixel. The superior performance of CNNs demonstrates its potential to produce features for stereo vision and to solve the discontinuity problem of the traditional feature-based stereo vision algorithms.

Recently, Zagoruyko et al. [4] and Zbontar et al. [5] utilized CNNs to compute the matching cost between two small image patches extracted from the left and right images. In particular, they used a Siamese network [6] with a few fully-connected layers at the end of the network to predict the matching cost. They trained the prediction models to solve a binary classification problem, which made a decision on whether the input patch pair was a good match. Luo et al. proposed a method in 2016 to compute the matching cost

between one patch from the left image and one horizontal strip from the right image [6]. It then calculated the disparity value for the center pixel of the input image patch at the end of the path [6]. These CNN-based architectures achieved very high disparity accuracy for the benchmark datasets [4-6]. However, the computation of these methods is very slow. They match one patch to either multiple small patches or a long strip of image to find the best disparity for one pixel. It takes anywhere from 1 second to 2 minutes to obtain the disparity map of an image pair, which limits the usage of these algorithms for real-time robotics applications.

When selecting a stereo vision algorithm, determining acceptable trade-offs between speed and accuracy depends on the target application [7]. This work attempts to provide a perspective on the efficiency of existing CNN-based stereo vision algorithms in terms of this trade-off. Inspired by the work in [6] and taking advantage of parallel processing, we input two image strips into CNNs and simultaneously calculate the disparities of each pixel in the entire left strip, rather than matching a patch to the right strip. We use CNNs to extract features for both strips and then compute the matching cost between these two groups of features. Finally, we utilize a layer of parallel LogSoftMax functions to simultaneously estimate the disparities for all pixels in the strips. This parallel architecture significantly reduces the processing time and makes it suitable for real-time use.

Details of our estimation model architecture are introduced in Section 2. Section 3 discusses our experiment settings and results. Finally, the paper is summarized and concluded with future research directions in Section 4.

2. PARALLEL CNN ARCHITECTURE

Unlike approaches [4] and [6] that compute the disparity for one center pixel of a small image patch at a time, we focus on computing disparity values at a much larger scale. Our architecture is able to compute disparity values for all pixels on an entire horizontal line simultaneously. The input images to the system are assumed to be rectified; therefore, the epipolar lines are aligned with the horizontal image axis.

Our parallel CNN architecture is shown in Fig. 2. It is a joint learning network which consists of two sub-networks: a feature network and a metric network. The feature network maps two input images to feature representations, and the metric network calculates the disparities for all pixels.

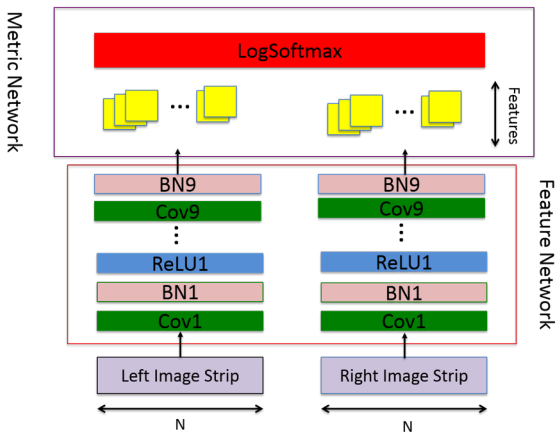

Fig. 2 The overview of the proposed parallel CNN architecture

2.1. Feature network

In many literatures, CNNs outperform other feature extraction methods. In our feature network, we chose a CNN model to extract features from the two input image strips. As shown in Fig. 2, both input images go through the same feature encoding. This network can be created by employing a Siamese architecture. Only one CNN model is needed because the features of the two images are obtained by the same model. The parallel CNNs are identical and remain together throughout the training. They share the same parameters; therefore, updates for either network will be added to their shared coefficients. We select several types of layers commonly used in image classification. We do not use pooling in our architecture. This architecture is suitable for both grayscale and color images.

Both networks within the feature network have nine convolutional layers. The first three convolutional layers have 64 convolutional filters of size 5×5. Three convolutional layers of 128 filters of size 7×7 are in the middle. The last three layers have 256 filters of size 9×9. Each convolutional layer is followed by a Batch Normalization layer and a nonlinear Rectified Linear Units (ReLU) layer. To retain the negative information, we removed the last Rectified Linear Units layer.

The feature network outputs a 256-dimension vector for each pixel in the center row in both input image strips. Each vector is wrapped into a 16×16 feature image and passed into the subsequent metric network.

2.2. Metric network

The metric network is illustrated in Fig. 3. The metric network is jointly trained from a supervised learning process. Since the last layer of the proposed model consists of parallel LogSoftMax functions, a cross-entropy error can be used as the loss function for training. By minimizing the cross-entropy loss error with respect to the weights, $\boldsymbol{w}$, as shown in Eq. (2), we are able to train a suitable model for stereo vision estimation task. We train our network using the stochastic gradient descent (SGD) back propagation.

Denote that (x_i, y_i) are the image coordinates of the pixels in the center row of the left image strip. Their disparities will be calculated simultaneously. During training,

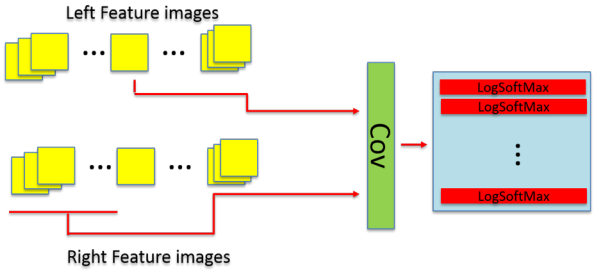

Fig. 3 The architecture of the metric network.

the goal is to minimize cross-entropy loss error with respect to the weights, $\boldsymbol{w}$, of the network

$$E = \min_{w} \sum_{i,y_i} p_{gt}(y_i) \log p_i(y_i, w) \tag{2}$$

where $p_i(y_i, \boldsymbol{w})$ represent the probabilities of all possible disparity classes (or values) for pixel (x_i, y_i), and the log $p_i(y_i, \boldsymbol{w})$ represents the final output from the i^{th} LogSoftMax function. Since we are interested in a 3-pixel to 5-pixel error metric, we use a smooth target distribution $p_{gt}(y_i)$ centered around the ground-truth y_i^{GT} as follows:

$$p_{gt}(y_i) = \begin{cases} \alpha_0 & if\ y_i = y_i^{GT} \\ \alpha_1 & if\ \|y_i - y_i^{GT}\| = 1 \\ \alpha_2 & if\ \|y_i - y_i^{GT}\| = 2 \\ \alpha_3 & if\ \|y_i - y_i^{GT}\| = 3 \\ \alpha_4 & if\ \|y_i - y_i^{GT}\| = 4 \\ \alpha_5 & if\ \|y_i - y_i^{GT}\| = 5 \\ 0 & otherwise \end{cases} \tag{3}$$

For our experiments, we set $\alpha_1 = 0.5$, $\alpha_2 = 0.2$, $\alpha_3 = 0.05$, $\alpha_4 = 0.02$, and $\alpha_5 = 0.005$.

Two groups of 16×16 feature images for the left and the right input image strips from the feature network are inputs to the metric network. We treat all feature images for the left image as kernels and convolve them with the feature images for the right image. Since the disparity value must be positive and the maximum disparity in the Middlebury dataset is 256, the feature images for the left image are only convolved with the 256 right feature images that represent pixels on the left side of the right input image strip. Applying this constraint, we significantly reduce the amount of computations and the number of required parameters in the network, which speeds up the entire stereo vision estimation process. A matching score between two pixels is then calculated from their corresponding feature images. Finally, we pass all matching scores into the LogSoftMax layer to calculate the disparity values.

Since the maximum disparity of our training images is 256, we set our metric network to have 256 classes of disparity for each LogSoftMax function. Each class represents one disparity value. We calculate matching scores between one left feature image and each of the 256 right feature images. Then, each LogSoftMax function in our parallel architecture processes these 256 scores to report the disparity class with the highest probability.

3. EXPERIMENTAL EVALUATION

KITTI [9] and Middlebury [10] are two commonly used stereo vision benchmark datasets. The two KITTI datasets were captured by driving through rural areas and highways around the mid-sized city of Karlsruhe. Many researchers have used the KITTI datasets to test their methods; however, the ground truth disparity maps of the KITTI datasets are sparse. They are not fit for our approach because the voids in the disparity maps confuse the training process of the network.

The image pairs of the Middlebury stereo datasets are indoor scenes taken under controlled lighting conditions [5]. Some scenes in the datasets were taken under varying lighting conditions and shutter speeds [11]. The datasets were published in five separate groups in the years of 2001 [10], 2003 [12], 2005 [13], 2006 [8], and 2014 [14]. In this paper, we refer to the Middlebury dataset as the concatenation of all five datasets.

Some disparity maps in the Middlebury dataset contain around 1% of pixels that lack ground truth disparity. To solve this problem, we used a simple image interpolation to assign an estimated disparity value to each of those pixels. From the Middlebury dataset, we selected 200 image pairs containing ground truth disparity maps for training and 57 image pairs for testing. The height of each input image strip was 55 pixels.

When dealing with the complex regions containing occlusions, saturation, etc., the raw disparity map must go through several time-consuming and complex post-processing steps to refine the disparities. In this paper, the disparity map produced from cost matching for comparison is a raw output without any post-processing.

There are many stereo vision post-processing methods in the literature that can be used to refine the raw disparity map. Cross-based cost aggregation, Semiglobal matching, a left-right consistency check, subpixel enhancement, median filtering, and bilateral filtering, etc. [5-6] are a few examples of these powerful post-processing algorithms. Luo et al. reported that the validation error of their architecture improves from 6.61% to 3.83% on the KITTI 2012 dataset after 4 post-processing smoothing steps [6]. Zbontar et al. reported that the validation error of their architecture improves from 13.49% to 2.61% on the KITTI 2012 dataset and from 13.38% to 3.25 % on the KITTI 2015 dataset after 6 post-processing steps.

Our proposed parallel architecture generates an accurate disparity map in real-time (0.1 second using NVIDIA Tesla K80 graphics processing units), which is much faster than the existing network architectures. As shown in Table 1, our proposed method sacrificed only about 2% of the accuracy in order to achieve this accelerated speed.

Table 1. Accuracy comparison.

Middlebury	>2 pixels	>3 pixels	>4pixels	>5pixels
2001	84.87%	86.44%	87.11%	87.50%
2003	87.89%	90.00%	91.40%	92.26%
2005	82.83%	87.67%	90.22%	91.74%
2006	81.24%	82.24%	82.27%	82.83%
2014	73.04%	78.56%	83.10%	87.21%
Average raw accuracy of the proposed architecture	81.94%	84.92%	86.82%	88.30%
[6] without post processing	85.71%	86.98%	88.72%	90.31%
[6] with simple post processing	87.18%	88.24%	90.83%	91.29%

Samples of the disparity maps from the Middlebury datasets using our proposed method are presented in Fig. 4. Table 1 shows the accuracy of our method for different Middlebury datasets. The overall performance is included as the average accuracy of all five datasets. We also list the raw accuracy of the method reported in [6] as well as the refined accuracy after a very simple cost aggregation post-processing. The cost aggregation post-processing simply performs an average pooling over a window of size 3×3 to the raw disparity map. These results indicate that the accuracy can be improved with post-processing techniques; however, in most robotic vision applications, determining an acceptable trade-off between speed and accuracy is very critical. In our work, we did not include post-processing. Rather, we showed the raw results to demonstrate that the proposed architecture performs well and is suitable for real-time applications.

4. CONCLUSION

In this paper, we propose a parallel CNN architecture for stereo vision estimation. Our results on the Middlebury datasets show that our network architecture produces accurate results. Unlike the existing CNN stereo vision architectures that either determine whether a pair of small image patches is a good match, or compute the disparity for one pixel in one forward pass time, our parallel architecture can estimate disparities for all pixels in the center row of an entire image strip in one single forward pass. Our architecture significantly speeds up the processing while only sacrificing roughly 2% of the disparity accuracy to do so.

With a balance between accuracy and efficiency, our proposed method shows great potential for real-time robotic vision applications. In the future, we will focus on modification for improved speed and implementation on a field programmable gate array device for embedded vision sensor application.

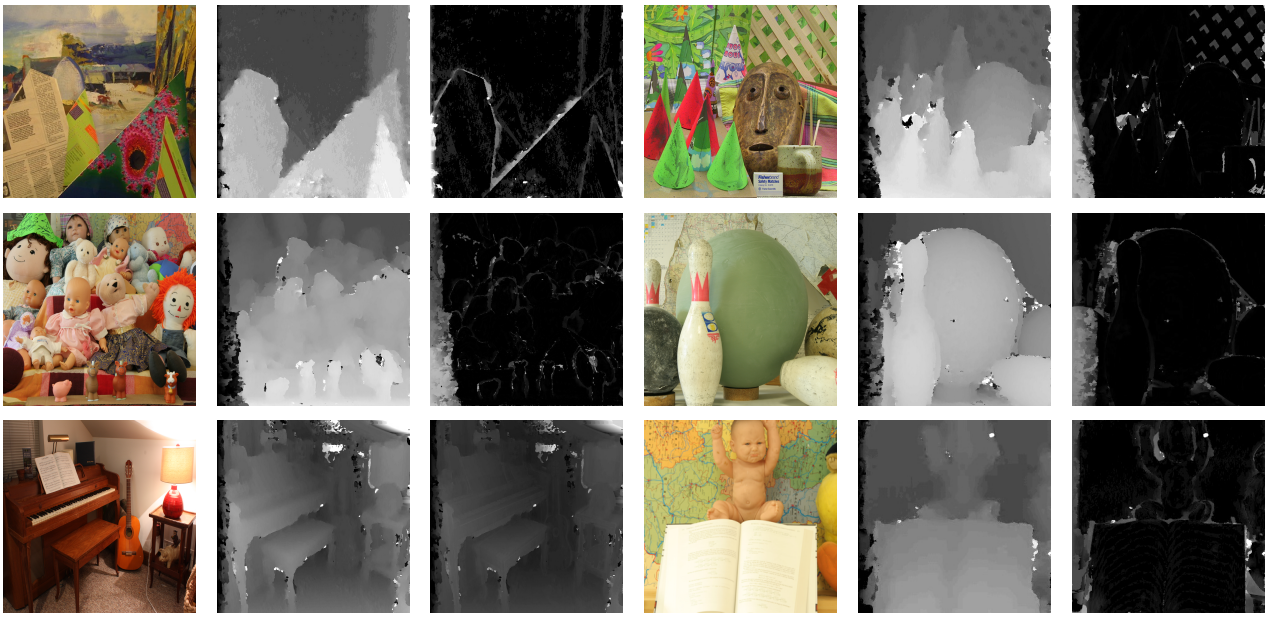

Fig 4. The results of Middlebury dataset. Columns 1 and 4: original left image, Columns 2 and 5: stereo estimate, Columns 3 and 6: disparity error.

REFERENCES

[1] Jo˜ao MF Rodrigues, Jaime A Martins, Roberto Lam, and JM Hans du Buf, "Cortical multiscale line-edge disparity model," in *International Conference Image Analysis and Recognition*. Springer, 2012, pp. 296–303.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[4] Sergey Zagoruyko and Nikos Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4353–4361.

[5] Jure Zbontar and Yann LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1592–1599.

[6] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun, "Efficient deep learning for stereo matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5695–5703.

[7] Beau Tippetts, Dah Jye Lee, Kirt Lillywhite, and James Archibald, "Efficient stereo vision algorithms for resource-limited systems," *Journal of Real-Time Image Processing,* vol. 10, no. 1, pp. 163–174, 2015.

[8] Heiko Hirschmuller and Daniel Scharstein, "Evaluation of cost functions for stereo matching," in Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007, pp. 1–8.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 3354–3361.

[10] Daniel Scharstein and Richard Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[11] Moritz Menze and Andreas Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3061–3070.

[12] Daniel Scharstein and Richard Szeliski, "High-accuracy stereo depth maps using *structured light," in Computer Vision and Pattern Recognition, 2003. Proceedings.* 2003 IEEE Computer Society Conference on. IEEE, 2003, vol. 1, pp. I–I.

[13] Daniel Scharstein and Chris Pal, "Learning conditional random fields for stereo," in *Computer Vision and Pattern Recognition, 2007. CVPR'07*. IEEE Conference on. IEEE, 2007, pp. 1–8.

[14] Daniel Scharstein, Heiko Hirschm¨uller, York Kitajima, Greg Krathwohl, Nera Ne˘si´c, XiWang, and PorterWestling, "High-resolution stereo datasets with subpixel accurate ground truth," in *German Conference on Pattern Recognition*. Springer, 2014, pp. 31–42.