BRIEF Stereo: Fast Stereo Matching using Multiscale Binary Feature

Sung Su Lim
HKUST
namasteism@gmail.com

Yu-Wing Tai
SenseTime Group Limited
yuwing@gmail.com

Jimmy Ren
SenseTime Group Limited
jimmy.sj.ren@gmail.com

Shengen Yan
SenseTime Group Limited
yanshengen@sensetime.com

Yun Liang
SenseTime Group Limited
ericlyun@pku.edu.cn

Abstract

In this paper, we propose a stereo matching algorithm based on multiscale binary feature. Contrary to state-of-the-art methods that utilize convolutional neural networks (CNN) for matching, our binary feature is hand-craft. A major benefit of our method is that our binary feature can be computed using very simple arithmetic operations, and the disparity map can be estimated in a very fast manner by using Hamming Distance to evaluate the matching cost. In our experiments, we carefully analyze the effectiveness of major components of our method. Results on the KITTI 2015 stereo benchmark show that our method is comparable to some state-of-the-art CNN based methods.

1. Introduction

Stereo matching is a fundamental task in computer vision. Its goal is to estimate a disparity map given a pair of rectified images. Recent advances in stereo matching are all based on convolutional neural networks. Representative works include [29, 5, 6, 3, 16, 17, 22].

While the CNN-based methods produce good results in stereo matching, they are computationally expensive which prohibit their usage in practice especially for applications that require real-time feedback such as autonomous driving [4]. Another limitation of CNN-based methods is that the trained network may overfit to training data. For example, DispNet [17] directly learns the disparity map given a pair of stereo images. When the baseline of the stereo pairs change, the trained network needs to be retrained or finetuned in order to produce correct results. However, it is very common that the baseline of stereo pairs changes across platforms, and the corresponding training data may not always be available.

In this paper, we present a novel and practical stereo matching algorithm based on multi-scale binary features. Our binary feature is designed to achieve two major goals. First, the developed binary features need to be general and robust enough in order to adapt to different challenging scenarios. Second, the developed binary features need to be simple enough in order to achieve fast computation. After deep analyses of the effectiveness of different local binary feature descriptors, we found that the BRIEF feature [2] best fits in our scenario. The BRIEF feature is chosen because it has the minimum computation costs among BRIEF [2], ORBS[19], BRISK [14] and FREAK [1] features. It is also highly robust to illumination variations and camera parameter changes across the left and right images of stereo pairs since the BRIEF feature only computes the relative intensity between two points in the same image, and the local image structure is binary encoded. However, the BRIEF feature does not generalize well in regions with thin objects or smooth area. Therefore, we extend the BRIEF feature to match the disparity across multi-scale. The multi-scale strategy also speeds up the matching process by avoiding the comparisons that are far from the correct disparity. Following previous works in stereo matching, our method also includes a series of post-processing steps including left-right consistency check and weighted least squares filtering to improve the accuracy of our disparity maps.

2. Related work

The stereo matching problem has been studied for decades with rich literature. A compete review is beyond the scope of this paper. Therefore, only the representative works are reviewed in this section.

The stereo matching problem consists of two major components: matching cost calculation, and spatial cost aggregation [21]. In matching cost calculation, common measurements include sum of squared difference (SSD), sum of absolute difference (SAD), zero mean normalized cross correlation(ZNCC), and robust square difference [20]. In some cases, not only the intensity information is used for

matching cost calculation, but also the image gradient and other high level statistices are used for matching cost calculation. A survey which evaluates the performance of different cost function for stereo matching is presented in [10]. In recent deep learning based methods, Zbontar and Yann [29] propose to use a convolutional neural network to learn the matching cost of two patches, and have demonstrated outstanding performance compared with conventional measurement methods. Follow-up methods such as Displets [5], Ensemble method [6], Embedding model [3], Content-CNN [16] and PBCP [22] consider different network architectures and have demonstrated better performance than the work by Zbontar and Yann.

In spatial cost aggregation, traditional methods formulate this problem as an energy minimization problem, where the data cost is the matching cost, and the neighboring cost is the spatial smoothness of disparity label. The energy minimization problem can be solved using superpixel/non-local aggregation [13, 27], dynamic programming [11], belief propagation [24], graph-cut [12], guided filtering [7], and semi-global block matching [9]. Some recent methods [26, 22] also include a confidence map in the spatial aggregation process. The confidence map evaluates the confidence of matching cost computation, while the occlusion map and left-right consistency check can also be included in the energy minimization framework of cost aggregation [18, 28]. A survey which evaluates the performance of different cost aggregation methods can be found in [25]. The aforementioned deep learning methods [29, 5, 6, 3, 16, 22] also include spatial aggregation as their post-processing methods.

3. Stereo Matching by BRIEF feature

In this section, we first review the BRIEF feature. Then, we describe our BRIEF stereo algorithm, followed by implementation details about post-processing methods.

3.1. Review of BRIEF

The BRIEF (Binary Robust Independent Elementary Features) [2] has been widely used as a feature point descriptor for image matching. Compared with other feature point descriptors, the BRIEF feature offers a major advantage in terms of speed. Not only is the feature itself very efficient to compute, but the descriptor similarity can also be evaluated using the Hamming distance, which is very efficient compared with the L1-norm/L2-norm distances. BRIEF feature is also highly discriminative even though it uses relatively few bits to represent local image patches.

BRIEF feature extracts the local image structure of a patch by first sampling a lot of random point pairs as illustrated in Figure 1. Each line segment in Figure 1 represents a pair of random sampled points. For each pair of random points, a 1-bit descriptor is computed as follows:

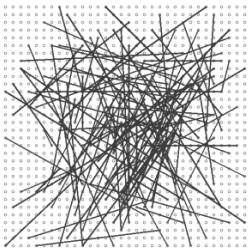

Figure 1. BRIEF descriptor generated by Gaussian Distribution. As reported in [2], random sampling which follows the Gaussian distribution, with the zero mean located at the patch center has the highest discriminative power.

$$\tau(\mathrm{p};\mathrm{x},\mathrm{y}) := \begin{cases} 1, & \text{if} \quad I(\mathrm{x}-\mathrm{p}) < I(\mathrm{y}-\mathrm{p}) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\{\mathrm{x},\mathrm{y}\}$ are the sampled random point pair, p is the center pixel coordinate of a patch, and τ denotes the 1-bit descriptor.

The BRIEF descriptor combines k number of 1-bit descriptor, and form a k_d-dimensional bitstring defined as

$$f_{k_d}(\mathrm{p}) := \sum_{1 \le i \le k_d} 2^{i-1} \tau(\mathrm{p};\mathrm{x}_i,\mathrm{y}_i)) \tag{2}$$

In practice, a BRIEF descriptor with 128 bits is sufficient to represent the local structure of a patch, and gaussian blur is applied beforehand to the image patch to make it more robust to noise.

In order to compare the similarity between two patches, the same set of random point pairs is used to compute the BRIEF descriptor of the two patches. The similarity of two BRIEF descriptors are then measured by the Hamming Distance between two binary bit strings. Hamming Distance is a very fast method to measure distance because it can be done with a simple bitwise XOR operation followed by a bit count.

As demonstrated in many image matching applications, BRIEF descriptor is robust to intensity variation across the images under comparison. This is because BRIEF descriptor only compares the relative intensity between two points in the same image. Also, the extracted BRIEF descriptor is robust to image contrast as the gradient magnitude is not encoded. While BRIEF descriptor offers many advantages, it is sensitive when matching two patches in different scale or with different rotation. However, considering the application in matching rectified stereo image pairs, any concern about rotation and scale variation across patches is eliminated.

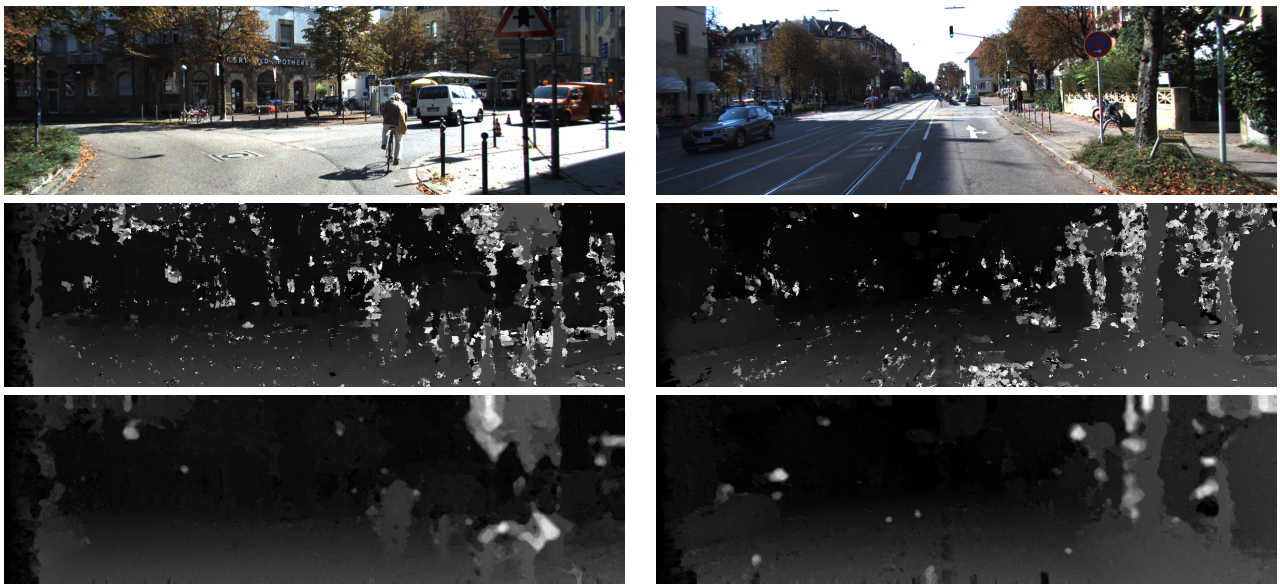

Figure 2. From Top to bottom: Input left image, winner-take-all single-scale depthmap, and winner-take-all multi-scale depthmap.

3.2. BRIEF Stereo

3.2.1 Single-scale Disparity Matching

Assuming the stereo image pair has already been rectified, the matching cost of pixels only needs to be computed along the horizontal axis. To apply it in the stereo matching problem, we use the BRIEF descriptor to characterize the local patches in the left and the right image, and compute the matching cost using the Hamming Distance. Unlike other models such as MC-CNN which require extensive training to learn to extract appropriate features and compute the matching cost, BRIEF feature does not require any training and can be computed very quickly.

We first evaluate the performance of BRIEF descriptor in stereo matching by applying the winner-take-all approach. In our implementation, we sample the BRIEF features within a 15x15 local window, and set the length of the bitstring k_d equal to 128, and the maximum disparity d_{max} equal to 255 for the KITTI dataset. In our implementation, we compute the BRIEF feature map sequentially in CPU. However, it can easily be computed in parallel in GPU using the frame buffer since it only involves shift operation and pixelwise comparisons.

For all possible disparity d ($0 \leq d \leq 255$), the matching cost for a pixel located at position p can be computed as:

$$C(p,d) = D_{\mathcal{H}}(B^L(p), B^R(p-d)). \tag{3}$$

where $D_{\mathcal{H}}$ denotes the hamming distance operation, and B^L, B^R denote the BRIEF feature map of the left and right image respectively.

Figure 2 shows the computed depthmap using the winner-take-all approach. The depthmap is quite accurate in most areas especially for high-textured regions. However, it is also very noisy, and the estimated disparities are incorrect for occluded regions and regions with homogeneous color.

3.2.2 Multi-scale Disparity Matching

From single-scale results, we can see that computing the matching cost for all possible disparity values is not only time-consuming but also leads to large disparity errors in occluded regions. Erroneous results may also occur in regions where the building/object structure is repeated or texture is monotone (i.e. the BRIEF feature is not effective in such regions). To overcome these limitations, we extend the single-scale approach to a multi-scale approach.

We resize the input stereo pair into 4 different resolutions using Gaussian pyramid. The scale difference between two consecutive layer is 2. At each resolution, we iteratively extract BRIEF feature maps, compute the matching cost and generate a depthmap. Note that the patch size for computing the BRIEF feature across different resolution is identical. Thus, the BRIEF feature maps are extracted at multiple scales. At the end of each depthmap computation, we apply post-processing to refine the depthmap before passing it to the next layer. Post-processing methods will be discussed in the later section. The overall process of multi-scale BRIEF model is shown in Figure 3.

The BRIEF disparity computation in each layer is defined as follows:

$$D_l(p) = \arg\min_d C(p,d), \tag{4}$$

$$d = \begin{cases} [0, d_{max}/n] & if \quad l = 0 \\ [2D_{l-1}(p) - \sigma, 2D_{l-1}(p) + \sigma] & otherwise \end{cases}$$

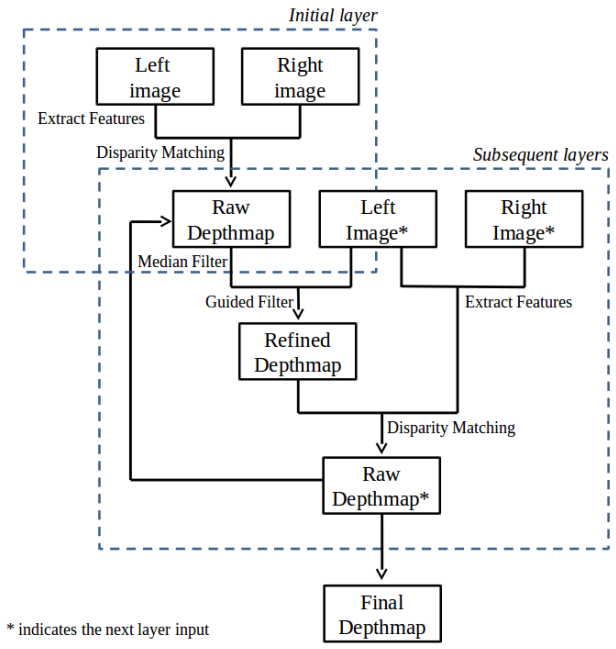

Figure 3. Architecture of Multi-scale BRIEF method

where d_{max} is the maximum possible disparity, l is index of layer, n is number of layers, and σ is a hyperparameter. Thus, except for the lowest resolution layer, *e.g.* $l = 0$, we only need to search the disparity range from $2D_{l-1}(p) - \sigma$ to $2D_{l-1}(p) + \sigma$. In our implementation, we set $\sigma = 4$.

In order to solve the problem of matching texture-less regions, we compute a confidence map to evaluate the reliability of matching bit-strings. In particular, for a k_d dimensional bit-string, it is the most confident when the patch is highly structured. This can be evaluated by counting total number of set bits, *e.g.* $\sum \tau = 1$, that is close to $k_d/2$. The confidence map $Conf(p)$ at position p is computed as:

$$Conf(p) = 1 - \frac{(|s - k_d/2|)^2}{d_{max}} \tag{5}$$

where s is the total number of set bits in $B_l(p)$. Figure 4 shows examples of our confidence map. If the confidence map at the current layer is higher than the previous layer, we expand the disparity search range. That is, for a confidence map $Conf_l(p)$ at position p in layer l,

$$\sigma = 2 * \sigma \quad if(Conf_l(p) >= Conf_{l-1}(p)). \tag{6}$$

By increasing the search range for confident regions, we have a higher chance in matching the correct disparity for small scale objects which has low confidence at low resolution but high confidence at high resolution. In contrast, large homogeneous area can be matched correctly at lower resolution, and a smaller search region for inconfident regions would allow the reliable matching at lower resolution to be propagated to higher resolution.

Figure 2 shows comparisons between the single scale approach and the multi-scale approach. As discussed, the disparity in regions around the occlusions, and texture-less regions (center of the road) are inaccurate. However, such error estimation is greatly improved with the multi-scale method. Note that the results presented in Figure 2 does not include any post-processing to be described in the next sub-section.

3.3. Post-Processing

Using post-processing to refine the initial disparity map is a common pipeline to boost the qualitative performance of stereo matching algorithms. Among common post-processing methods, the MRF based methods, such as belief propagation or graph cut can produce quite decent results, even with very simple SAD/SSD measurements in the cost volume. However, these optimization based methods are computationally expensive. Since developing a fast stereo matching algorithm is the primary goal of our paper, we discuss our method for depth map refinement that only utilizes tools which are well-known to be fast.

For each level of our results in the multi-scale method, we apply 5×5 median filter to remove specks of noise from our estimation. As discussed in [23], the secret of success in the multi-scale method for optical flow estimation is intermediate median filtering before upsampling. Note that the median filter or weighted median filter has a very efficient implementation as described in [30].

In the upsampling process, instead of using bicubic upsampling, we found that using guided filter [8] with the input image at the next level as the guided image to upsample the disparity map can better preserve the disparity discontinuities. Similar to median filter, the guided filter has a very fast implementation as described in [7].

In our implementation, we have also performed the left-right consistency check. The left-right consistency check is an effective method to compute occluded regions. In each layer, we compute the left and right disparity map, and identify pixels where the left disparity value do not agree with the corresponding pixel in the right disparity map. Using the left-right consistency check, we can identify pixels that require hole filling. Similar to the upsampling process, the hole filling process can be implemented using guided filter with very fast implementation [15].

We notice that sampling BRIEF features within a 15×15 local window may not capture very well the structure of thin objects. To overcome this limitation, in the last layer of our multi-scale method, we additionally sample BRIEF features within 7×7 local window. Note that although we sample the BRIEF features in a smaller local window, the sampled pattern are identical to the one used in the 15×15 local win-

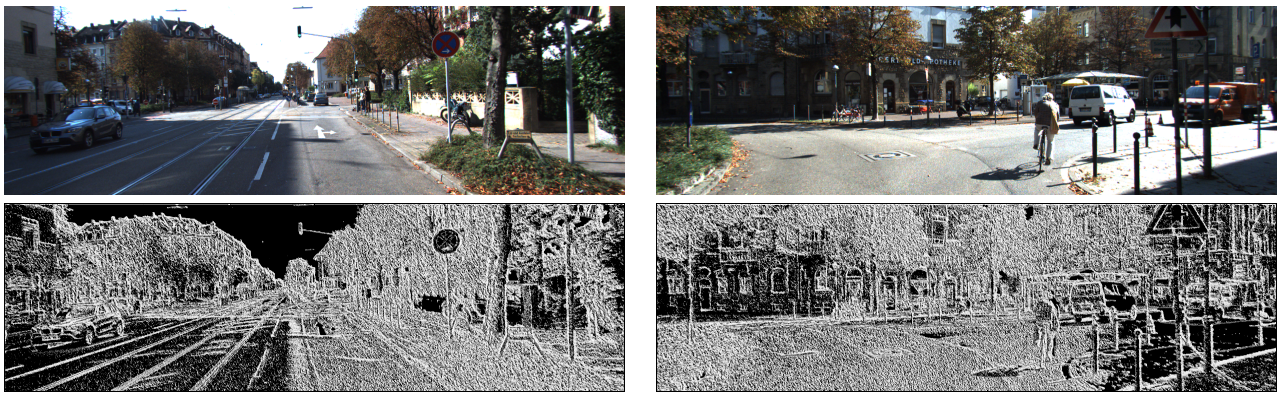

Figure 4. Confidence map at the last layer. Our confidence map is accurate in identifying texture-less regions, and its computation is also very fast.

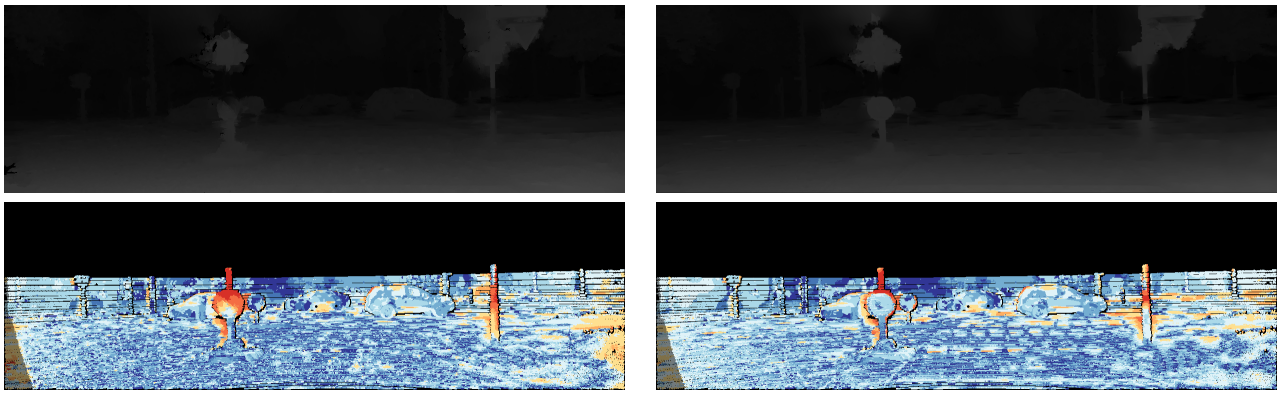

Figure 5. Depthmap results without (left) and with (right) the small window BRIEF feature sampling at the last layer.

Time	Single-scale	Multi-scale
Feature map Extraction	4.5 s	2.22 s
Disparity Computation	30 s	1.2 s
Post-Processing	1.5 s	0.3 s
Total	36 s	3.72 s

Table 1. Comparison of Runtime

Error	D1-bg	D1-fg	D1-all
All / All	7.04 %	18.72 %	8.99 %
All / Est	7.04 %	18.71 %	8.98 %
Noc / All	6.50 %	17.49 %	8.31 %
Noc / Est	6.50 %	17.49 %	8.31 %

Table 2. KITTI 2015 Test set Results

dow. This way, we can compare the confidence map across levels, and we only update the disparity of small objects only when it has higher confidence than the confidence in the previous level. Figure 5 compares our results with and without the small window BRIEF feature sampling. With the additional small window sampling of BRIEF feature, the thin structures are better estimated.

4. Experimental Results

We test the performance of our method on the KITTI 2015 stereo dataset. The dataset contains 200 training scenes and 200 test scenes. Each scene has a pair of stereo image with resolution 375×1242, and the disparity level ranges from 0 to 255. To provide a quantitative evaluation, the dataset contains ground truth disparity map captured by a laserscanner. Because the ground truth disparity maps are captured at different locations from the stereo camera, the ground truth disparity maps contain considerable amount of holes, and the evaluation is only carried out on the valid pixels of the ground truth disparity maps.

Our method is implemented in CPU without any parallel processing. We tested it on a Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz machine and it takes approximately 3.72 seconds for our method to compute a single depthmap image. When including the left-right consistency check, the running time is doubled because we need to estimate both the left and the right disparity map simultaneously. Table 1 provides a detailed analysis of our runtime. The runtime comparison between our single-scale and multi-scale

method is also provided. In our current implementation, the BRIEF feature map computation is the bottleneck of our multi-scale method. However, as discussed in previous section, the computation time of feature map extraction can be significantly accelerated using GPU/multi-core parallel implementation.

Our quantitative results on the KITTI 2015 dataset are provided in Table 2. The "D1-bg/fg" represents the percentage of stereo disparity outliers in background/foreground regions, "All" represents all ground truth pixels, and "Noc" represents non-occluded regions of ground truth pixels. Our method estimates a dense disparity map, and has 0.5% lower error rate when excluding occluded regions.

Figure 6 and Figure 7 show some of our results and error images for qualitative evaluation. The blue areas in the error maps represent correct regions with error values less than 3 pixels. When looking at the result images, we can see that there are still errors in detecting thin objects and occluded regions. However, considering the autonomous driving application, fast runtime is far more important than the minor errors shown in our examples. It is worth noting that the amount of accurate area of our disparity maps is already over 85%, and such accuracy is already sufficient for real-world applications.

5. Conclusion

In this paper, we have presented a fast and robust method to estimate disparity map using BRIEF features and the multi-scale approach. The BRIEF feature map extraction and matching cost computation can be easily implemented with simple bit-wise operations. Because of its simplicity, we believe it can be accelerated drastically when implemented with parallel computation. The BRIEF descriptor uses a small number of bits to effectively represent an image patch, and our multi-scale approach saves further computation time by reducing the number of disparity values to be computed. Furthermore, our method does not require any training beforehand, and therefore can be applied in any disparity map computation tasks even when the dataset is not available. These advantages are useful in many industrial applications, including but not limited to autonomous vehicles.

References

[1] A. Alexandre, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2012. 1

[2] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision(ECCV)*, pages 778–792, 2010. 1, 2

[3] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang. A deep visual correspondence embedding model for stereo matching costs. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 2

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *Int. J. Rob. Res.*, 32(11):1231–1237, 2013. 1

[5] F. Guney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2015. 1, 2

[6] R. Haeusler, R. Nair, and D. Kondermann. Ensemble learning for confidence measures in stereo vision. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2013. 1, 2

[7] K. He and J. Sun. Fast guided filter. *CoRR*, abs/1505.00996, 2015. 2, 4

[8] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(6):1397–1409, 2013. 4

[9] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2), Feb. 2008. 2

[10] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2007. 2

[11] J. C. Kim, K. M. Lee, B. T. Choi, and S. U. Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2005. 2

[12] V. Komolgorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2002. 2

[13] K.Yoon and I.S.Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:492–504, 2006. 2

[14] S. Leutenegger, M. Chli, and R. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 1

[15] Y. Li, D. Min, M. N. Do, and J. Lu. Fast guided global interpolation for depth and motion. In *European Conference on Computer Vision(ECCV)*, pages 717–733, 2016. 4

[16] W. Luo, A. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2016. 1, 2

[17] N. Mayer, E. Ilg, P. Husser, 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 *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*. 1

[18] M.Gong and Y.H.Yang. Fast stereo matching using reliability based dynamic programming and consistency constraints. In

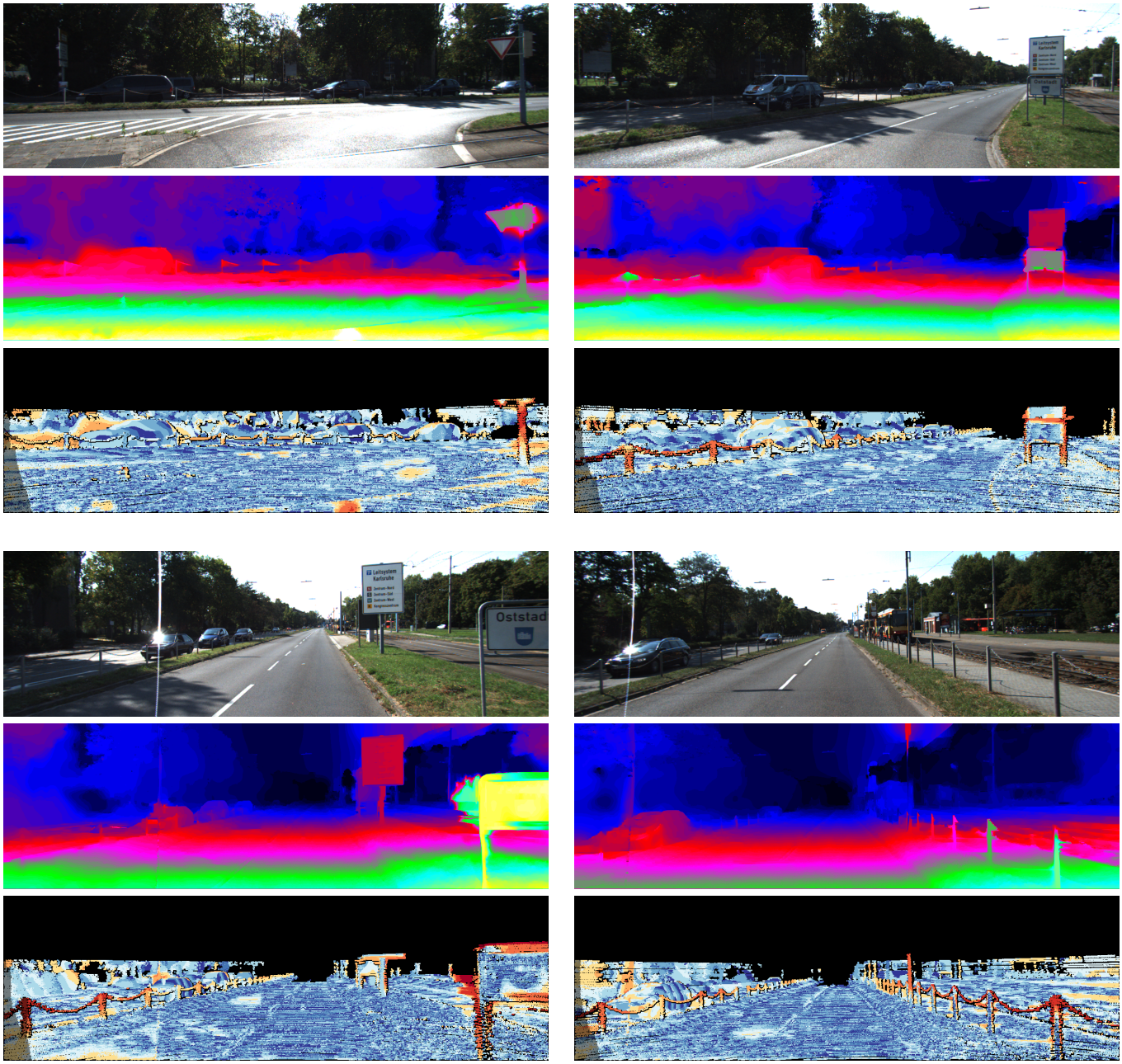

Figure 6. KITTI 2015 Stereo Test Result. From top to bottom, it shows left input image, disparity map, and error map respectively.

Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2003. 2

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 1

[20] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *Int. J. Computer Vision*, 28(2):155–174, 1998. 2

[21] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, 47:7–42, 2002. 1

[22] A. Seki and M. Pollefeys. A deep visual correspondence embedding model for stereo matching costs. In *British Machine Vision Conference (BMVC)*, 2016. 1, 2

[23] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2010. 4

[24] J. Sun, N. Zheng, and H. Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(7):787–800, 2003. 2

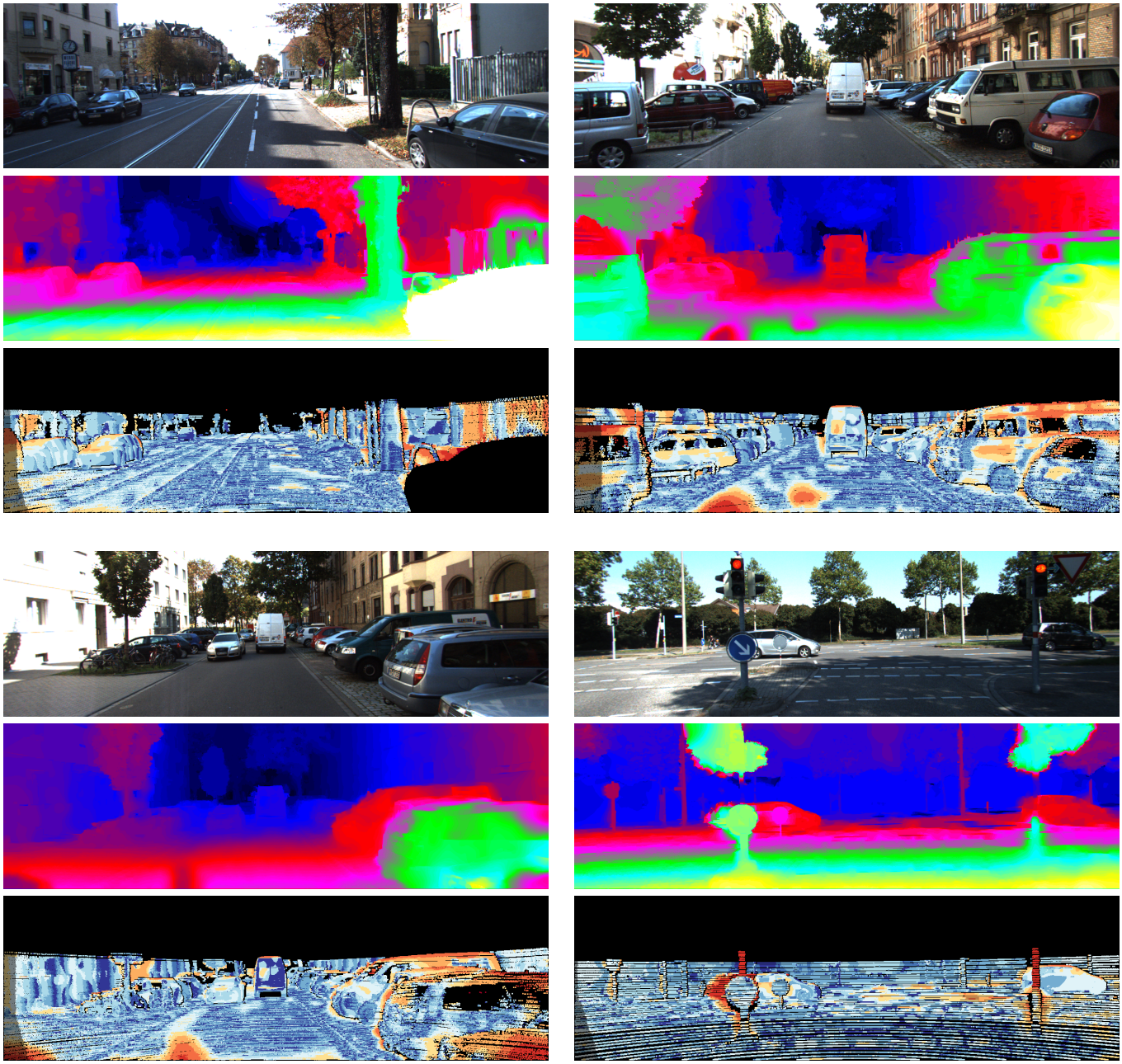

Figure 7. KITTI 2015 Stereo Test Result. From top to bottom, it shows left input image, disparity map, and error map respectively.

[25] F. Tombari, S. Mattoccia, L. D. Stefano, and E. Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2008. 2

[26] L. Xu and J. Jia. Stereo matching: An outlier confidence approach. In *European Conference on Computer Vision(ECCV)*, 2008. 2

[27] Q. Yang. A non-local cost aggregation method for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*, 2012. 2

[28] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31:492–504, 2009. 2

[29] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. 2016. 1, 2

[30] Q. Zhang, L. Xu, and J. Jia. 100+ times faster weighted median filter (wmf). In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recogni-*

tion(CVPR), 2014. 4