CBMV: A Coalesced Bidirectional Matching Volume for Disparity Estimation

Konstantinos Batsos kbatsos@stevens.edu
Changjiang Cai ccai1@stevens.edu
Philippos Mordohai mordohai@cs.stevens.edu
Stevens Institute of Technology

Abstract

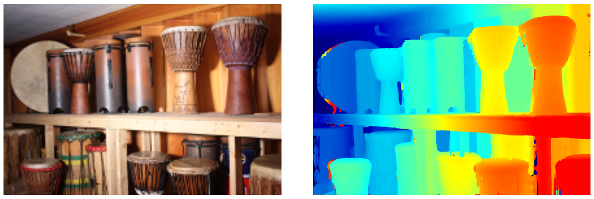
Recently, there has been a paradigm shift in stereo matching with learning-based methods achieving the best results on all popular benchmarks. The success of these methods is due to the availability of training data with ground truth; training learning-based systems on these datasets has allowed them to surpass the accuracy of conventional approaches based on heuristics and assumptions. Many of these assumptions, however, had been validated extensively and hold for the majority of possible inputs. In this paper, we generate a matching volume leveraging both data with ground truth and conventional wisdom. We accomplish this by coalescing diverse evidence from a bidirectional matching process via random forest classifiers. We show that the resulting matching volume estimation method achieves similar accuracy to purely data-driven alternatives on benchmarks and that it generalizes to unseen data much better. In fact, the results we submitted to the KITTI and ETH3D benchmarks were generated using a classifier trained on the Middlebury 2014 dataset.

1. Introduction

The most important recent development in stereo matching is the prevalent use of machine learning techniques that have led to dramatic improvements in accuracy by taking advantage of datasets with ground truth. Methods based on learning are effective because they replace assumptions and hand-crafted rules with data-driven, optimized decision rules and predictions. Classifiers are used to contribute in various stages of the disparity estimation process; several authors have trained classifiers to predict whether two image patches are likely to match [4, 20, 26, 39, 47, 49, 50, 51], while others have used classifiers to replace hand-crafted rules in other stages of the process [8, 27, 29, 30, 37, 38, 41, 42]. A quick inspection of the most active binocular stereo benchmarks [7, 23, 34] reveals that learning the matching function, in particular the work of Žbontar and LeCun [51], has been the primary enabling technology behind the majority of the top-ranked algorithms.

In addition to the above approaches that propose learning-based components integrated into the stereo matching pipeline, there are a few deep learning architectures that allow end-to-end training [14, 16, 21, 25]. While end-to-end training has the advantage that it avoids suboptimal configurations, which often occur when intermediate objectives are optimized separately from final disparity map accuracy, its downside is that these methods tend to over-specialize in the training domain. As anecdotal evidence for this statement we provide the observation that very few results from end-to-end architectures have been submitted to the Middlebury 2014 benchmark [34], which contains images of various resolutions and disparity ranges. Specialization is a desirable property in many applications, such as autonomous driving. In this paper, we aim to create a *general approach that can be effective on a broad range of input imagery.*

In contrast to end-to-end architectures, deep learning methods that learn the matching likelihood of image patches have shown better generalization properties, but they cannot be trained in an end-to-end manner. For example, the MC-CNN method of Žbontar and LeCun [50, 51] uses a Siamese CNN to estimate the matching volume, that contains the matching likelihood/cost for each allowable disparity of every pixel, and conventional steps to optimize the volume and extract the final disparity map. Their pipeline resembles the last three steps of the conventional pipeline according to Scharstein and Szeliski [35]: cost aggregation, disparity optimization and disparity refinement. MC-CNN has been widely adopted as the cost function by a number of authors who have presented state-of-the-art results



Djembe CBMV disparity map

Figure 1. The left view of Djembe stereo dataset [34] along with the disparity map computed by CBMV

[2, 6, 8, 15, 18, 37, 38, 43].

Our goal is similar to MC-CNN, since we also aim to estimate a matching volume that can be used as input to various optimization algorithms enabling them to produce highly accurate disparity maps. Instead of taking an extremely data-driven approach, in which the stereo matcher is only provided with two image patches and a label specifying whether they match, we design our matching volume estimator with an emphasis on *robustness* and *invariance*.

To improve the generalizability of our approach, we design it to be invariant to common variations of the input images. (It may be possible to achieve invariance by applying data augmentation techniques to the training set, but then the designer would have to specify the variations manually.) Most conventional matching functions provide some form of invariance to specific transformations of the input. For instance, normalized cross-correlation (NCC) is invariant to affine intensity transformations, while the census transform [48] is invariant to transformations that preserve the ordering of intensities in the matching window. These matching functions are known to fail quite often, but their failures can be predicted via the use of confidence measures [13, 33]. More importantly, these failures are mitigated by combining a diverse set of matching functions.

In addition to using four matching functions in the current implementation of our approach, we compute two measures of confidence for each matching function and each matching direction: left-to-right and right-to-left. The matching cost between a pixel p_L in the left image and a pixel p_R in the right image is the same regardless of the matching direction. The ambiguity, and thus the confidence, of the correspondence, however, may differ with respect to the matching direction. A disparity assignment, that joins a pixel in the left image with one in the right image, must compete with other possible disparity assignments in both the left and the right epipolar line. Unlike most previous work [10, 27, 42], we measure the degree of competition (ambiguity) in both directions. The resulting matching and confidence data are coalesced by a random forest (RF) classifier [3] that estimates the probability of correctness of each disparity. Hence, we name our method *Coalesced Bidirectional Matching Volume (CBMV)*. Figure 1 shows an example disparity map estimated by our algorithm.

Throughout the paper, we compare our approach to MC-CNN [51]. In order to make the comparison straightforward, we apply their optimization and post-processing pipeline on the CBMV volume. Our experiments show that CBMV generalizes to data from domains that differ substantially from the training domain. We believe that the reason for this is that *our approach learns to reason on relationships in the matching volume without being affected by image appearance,* which it never observes directly.

The contributions of this paper are:

- A novel method for computing the matching volume for stereo that benefits from the combination of multiple matching functions and confidence estimates in both matching directions.
- Competitive results with the fast MC-CNN architecture, which is the most widely adopted deep architecture for patch matching.
- An improved capability to generalize to inputs from unseen domains much better than deep learning approaches, such as MC-CNN.

2. Related Work

For a general survey of binocular stereo methods we refer readers to [35]. In this section, we review learning-based methods in which learning is directly relevant to disparity estimation. We consider methods that learn hyper-parameters of the stereo algorithm [24, 45, 46, 53] out of scope. We classify the methods below according to the primary problem they address: determining disparity correctness, using correctness predictions to improve disparity estimation, learning the matching function, and end-to-end pipelines.

Early research in stereo using machine learning methodology addressed the problem of deciding whether a disparity was correct or not [5, 17] but its short-term impact was limited. This changed recently with publications such as the one by Haeusler et al. [10] who train a random forest to predict the correctness of the output disparities of the SGM algorithm [12] using features computed on the images, disparity maps and matching cost volume. Gouveia et al. [9] extend the confidence estimator of [40] to be applicable to a superpixel-based stereo algorithm. The classifier is able to remove errors from the disparity maps, which are filled in using conventional techniques. Poggi and Mattoccia [31] pose confidence estimation as a regression problem and solve it using a CNN trained on small patches of disparity maps based on the observation that patterns in the disparity map can indicate whether a certain disparity assignment is correct. The same authors [32] improved a number of previous methods by training a CNN to refine confidence maps. The classifier's predictions in all cases [9, 10, 31, 32] are effective in sparsifying the disparity maps by removing potential errors, but do not help in the generation of more accurate disparity estimates.

This shortcoming was addressed by algorithms that inject confidence into the optimization stage. Spyropoulos et al. [40, 42] train a random forest on the cost volume to detect ground control points, the disparities of which are favored during MRF-based disparity optimization. Park and Yoon [27] use the predictions of a random forest to modulate the data term of each pixel in SGM-based optimization. Poggi and Mattoccia [30] present a confidence measure that takes into account multi-scale features and is used to weigh cost aggregation in SGM in order to reduce artifacts. Seki and

Pollefeys [37, 38] present two algorithms for adjusting the regularization parameters of SGM using CNNs trained on stereo pairs.

Matching cost estimation was addressed by Li and Huttenlocher [19] who use a structured support vector machine to learn linear discriminant functions that compute the data and smoothness terms of a Conditional Random Field (CRF) based on discretized values of the matching cost, image gradients and disparity differences among neighboring pixels. Later, Alahari et al. [1] applied convex optimization, using the same node and edge features as [19], to obtain the solution more efficiently. Peris et al. [28] use synthetic data to train a classifier for matching cost aggregation and disparity optimization. Multi-class LDA is applied to learn a mapping from a feature vector that contains neighborhood matching costs at all disparities for a pixel to the disparity that should be assigned to the pixel.

The approach that ignited the recent wave of deep learning based stereo methods was the one of Žbontar and LeCun [50, 51]. MC-CNN comes in two versions depending on the steps that follow a Siamese network that learns a representation of image patches: in the fast architecture, MC-CNN-fst, the representations of the two patches are compared using the cosine similarity measure, while in the accurate architecture, MC-CNN-acrt, patch similarity is the output of several fully connected layers that operate on the concatenated representations. Similarly to our approach, the networks are trained on matching and mismatching pairs of image patches. Žbontar and LeCun also augment the training data by distorting and photometrically modifying the images. MC-CNN generates a matching cost volume that undergoes a number of processing steps, including SGM optimization, to generate disparity maps. Similar Siamese networks followed by the fast or accurate similarity estimation subnetworks have also been proposed by [4, 11, 20, 49], while more recently, other authors have increased the effective receptive field of the networks without loss of resolution [26, 39, 47]. Many of the other top ranked methods have either been inspired by MC-CNN or directly use it to compute the matching cost [2, 6, 8, 15, 18, 37, 38, 43].

Shaked and Wolf [39] rely on deep learning in two stages of the pipeline: cost computation and final disparity map inference from the matching cost volume. Along with GC-Net [14], their global disparity network is the only deep learning approach that operates in the cost volume. The network also estimates confidence using a novel reflective loss function.

Disparity refinement is typically addressed by applying various filters and interpolation techniques on the disparity map [22]. Recent disparity refinement methods [8, 47] have been able to learn to correct mistakes without relying on hand-crafted rules.

The first end-to-end stereo matching system was introduced by Mayer et al. [21]. The proposed architectures, DispNet and DispNetC, go beyond learning how to match small square patches and learn how to estimate disparity maps given a pair of rectified images. Knöbelreiter et al. [16] present a hybrid CNN-CRF model based on a formulation that allows end-to-end training of a 4-connected CRF, which is more effective on stereo matching than fully-connected CRFs. Very recently, Kendall et al. [14] presented an end-to-end pipeline (GC-Net) based on a high-capacity, deep architecture that resembles the conventional pipeline. It included 3D convolutional layers that regress disparity from a cost volume generated by residual blocks that extract patch representations from the images. Compared to DispNetC, the availability of a cost volume allows GC-Net to exploit context and achieve state-of-the-art results. Pang et al. [25] proposed a cascade of two networks that can be trained end-to-end. The first network is similar to DispNetC while the second refines the disparity map.

3. Overview of the approach

Before presenting a step-by-step break down of our method, we provide a brief high-level description of the building blocks and the intuition behind each step involved. Figure 2 shows a flowchart of our method. Our objective is to compute a "good" matching volume that captures the support and competition among competing disparity hypotheses and is amenable to global optimization. (See [51] for an analysis of what a good matching volume is.)

The cost computation step combines the matching volumes computed by four basic matchers with confidence volumes extracted from the matching volumes. A Random Forest classifier [3] is trained to coalesce all the input evidence and generate the CBMV. The motivation behind this step is to combine the strengths and mitigate the weaknesses of these basic stereo matchers to generate a robust matching volume for the subsequent optimization steps. Our optimization and post-processing pipeline adopts the steps proposed by Žbontar and LeCun with slight modifications to generate the final disparity maps, allowing a direct comparison of our results with those of MC-CNN.

4. Matching Volume Computation

The unit on which our algorithm operates is the *matching hypothesis*, (x_L, x_R, y), that represents a potential correspondence between pixel $p_L(x_L, y)$ in the left image and a pixel $p_R(x_L - d, y)$ in the right image. Disparity d is always defined as $d = x_L - x_R$ and the matching hypothesis can be written equivalently as (x_L, d, y). In the remainder, we drop y for simplicity since the images are rectified. The range of possible disparities d_{max} is also an input.

To determine whether a matching hypothesis is likely or not, we combine *matching volumes* generated by four basic

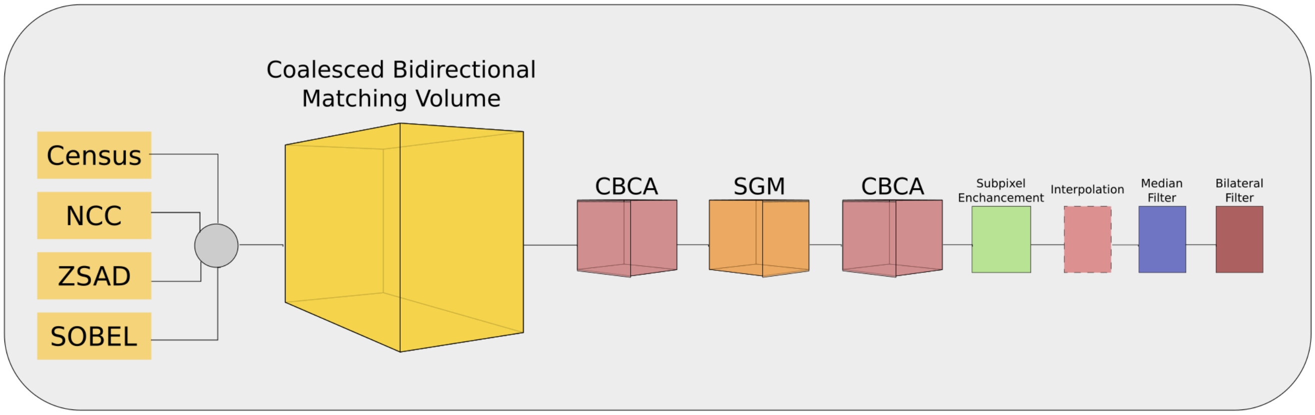


Figure 2. Flowchart of our approach. The matching costs of the four matchers are coalesced with bidirectional confidence features to create the CBMV denoted by the yellow cube. The smaller cubes show processes that operate on the CBMV, while squares show processes that operate on the disparity map after optimization.

block matching algorithms, *NCC*, *CENSUS*, zero-mean SAD on intensities (*ZSAD*) and SAD on the responses of the horizontal Sobel filter (*SOBEL*), with two *confidence volumes* for each matching function and each matching direction.

Matching Volume Representation. The matching volume for a given matching algorithm stores a value for each possible correspondence between a pixel in the left image to a pixel in the right image within a given disparity range. We use the disparity-based representation for the matching volume and write $C_{cen}(x_L, d)$ for the one computed using *CENSUS* for example. Given x_L and d (a matching hypothesis), x_R can be retrieved by $x_R = x_L - d$. Typically, the left image is treated as reference and the right image as matching target. Switching the roles of the images, and negating the disparity range, leads to a new matching volume that can also be obtained by re-ordering the values of the left-right volume without re-computation (see Fig. 3).

We also compute confidence volumes that capture the ambiguity of disparity hypotheses. These are computed bidirectionally since the competitors of a potential correspondence in the left image are not the same as its competitors in the right image. We introduce the following notation: c^L_{min} denotes the minimum observed cost of a matching algorithm for the left pixel of a matching hypothesis and d^L_{min} the corresponding disparity. c^R_{min} and d^R_{min} are their counterparts for the right pixel. c^L_{min} can be obtained by traversing the yellow lines (constant x_L) in Fig. 3 and c^R_{min} by traversing the red lines (constant x_R).

The confidence measures are adapted from [13] so that they can be used to compute the confidence of *all* disparity values of a pixel and not only the one with the minimum cost. For each disparity under consideration and each basic matcher, we extract a feature vector which consists of the following five elements: matching cost C, left and right ratio R^L and R^R, and left and right likelihood L^L and L^R. We use *CENSUS* as an example below, but the process is repeated for all matchers.

Matching Cost. This is the raw cost or score of the basic stereo matching algorithm for each disparity under consideration.

$$C_{cen}(x_L, d) = CENSUS(x_L, d) \tag{1}$$

Ratio. The ratio of the minimum cost $c_{cen,min}$ over the cost of the candidate disparity $C_{cen}(x_L, d)$ assigns high confidence to disparities with close to minimum cost.

$$R^L_{cen}(x_L, d) = \frac{c^L_{cen,min}}{C_{cen}(x_L, d)} \tag{2}$$

This is computed by finding the minimum cost over x_L along the yellow lines in Fig. 3.

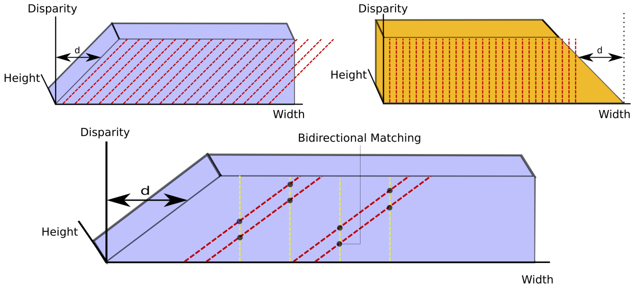


Figure 3. Top: The left and right matching volumes. The dashed red lines correspond to the matching costs for a given pixel of the right image. A volume can be generated from its counterpart by shifting its equal-disparity slices by d. Bottom: For a given element of the matching volume, the ratio and likelihood features are computed along the yellow and red lines corresponding to the right and left epipolar lines respectively. Black dots denote a few intersections of left and right epipolar lines on the matching volume. Each intersection is a matching hypothesis linking a pixel in the left image with a pixel in the right image.

Similarly, for the right-to-left matching direction:

$$R^R_{cen}(x_L, d) = \frac{c^R_{cen,min}}{C_{cen}(x_L, d)} \qquad (3)$$

This is computed by finding the minimum cost over x_R along the red lines in Fig. 3.

Likelihood. Similar to AML in [13], we convert the cost curve for a given pixel to a probability density function to generate a confidence measure for a given disparity hypothesis.

$$L^L_{cen}(x_L, d) = \frac{e^{-\frac{(C_{cen}(x_L,d)-c^L_{cen,min})^2}{2\sigma^2}}}{\sum_i e^{-\frac{(C_{cen}(x_L,d_i)-c^L_{cen,min})^2}{2\sigma^2}}} \qquad (4)$$

where σ is a hyper-parameter that depends on the matching algorithm. To obtain $L^R_{cen}(x_L, d)$ the summation in the denominator must be over $C_{cen}(x_L + d_i, d_i)$ so that all terms match to the same pixel in the right image.

Training. We train a Random Forest (RF) classifier to predict the matching likelihood of two image patches. The classifier learns whether disparity candidates for each pixel are likely to be correct based on the costs, agreements and disagreements of the matchers and the degree of ambiguity along the left and right epipolar lines captured by the confidence measures.

Each disparity hypothesis is represented by 20 cost and confidence values (5 per matching function). Since at most only one hypothesis is correct per pixel, our dataset is imbalanced. To counter this, during training we keep all correct correspondences and sample twice as many incorrect correspondences, while Žbontar and LeCun [51] keep a $1:1$ ratio. We keep all exact correspondences as positive samples, while they also consider correspondences that are off by one disparity level as correct. Then we randomly pick two disparity values, one in the lower range $[0 \dots d_{gt} - 1)$ and one in the upper range $(d_{gt} + 1 \dots d_{max}]$, where d_{gt} is the ground truth disparity, and label them as negative samples. The RF learns to make predictions on the correctness of each disparity assignment that links a pair of pixels.

During testing, the RF is applied on the entire matching and confidence volumes to produce the Coalesced Bidirectional Matching Volume, which is the input to the optimization steps described below.

Note that the right CBMV can then be obtained by shifting the left CBMV. The shift is valid under a mild assumption that the left and right confidence values are generated from the same distributions. If this is the case, swapping the left and right confidence features should not affect the classifier's prediction. That is, $[C\ L^L\ R^L\ L^R\ R^R]$ and $[C\ L^R\ R^R\ L^L\ R^L]$ should be equivalent as feature vectors, yielding equal predictions from the classifier.

5. Optimization and Post-processing

The next step of our algorithm is optimization and post-processing to generate the final disparity map from the coalesced volume. Since we need both the left and right disparity maps to apply consistency constraints, we generate the right CBVM by shifting the left one as shown at the top of Fig. 3. The following steps are applied to the two volumes separately to produce the two disparity maps.

We use the pipeline of Mei et al. [22], which was also adopted by MC-CNN [51]. Its steps can be seen on the right side of Fig. 2. We only provide a high level description of the components since they are not novel. There are steps that operate in volumes, namely cross based cost aggregation (CBCA)[52]and Semi-Global Matching (SGM) [12], and further steps that are applied on 2D disparity maps, namely sub-pixel refinement via parabolic fitting, left-right consistency test and interpolation to fill in invalidated pixels, followed by median and bilateral filtering. Following MC-CNN we apply CBCA before and after SGM. While we keep the structure of the pipeline, we tuned the values of its parameters via cross-validation per dataset to obtain high accuracy. More details are included in the supplement.

6. Experimental Results

We evaluate our algorithm on the 2014 version of the Middlebury Stereo Evaluation dataset [34], the 2012 and 2015 versions of the KITTI stereo benchmark [7, 23] and the ETH3D stereo benchmark [36].

The Middlebury dataset consists of a training set of 15 stereo pairs, 13 additional stereo pairs, all with publicly available ground truth, and a test set of 15 stereo pairs, the ground truth for which has not been released. Compared to previous versions of the benchmark, this version is more challenging because most stereo pairs have imperfect rectification, except those with a suffix 'P' in their filename, while several others contain images taken under different exposure or lighting, denoted by 'E' and 'L' respectively [34]. The image resolution varies between 1.5 and 5.9 megapixels with an average of 5.2 megapixels and the disparity range varies between 256 and 800. As most authors, we use half-resolution images. The ranking in the new tables is determined by weighted averages of the selected metric.

The KITTI datasets consist of approximately 200 training and 200 testing stereo pairs each. The ground truth is semi-dense covering roughly 30% of all pixels and is concentrated in the lower part of the images. The ground truth of the test sets has been withheld. An important difference between the two versions of the benchmark is that cars have been manually annotated in the 2015 version and have dense ground truth, including their windshields. The latter are explicitly deleted from the ground truth of the 2012 benchmark.

The ETH3D stereo dataset consists of 27 training and 20

	Out-Noc				Out-all			
Method	bad .5	**bad** 1.0	bad 2.0	rms	bad .5	bad 1.0	bad 2.0	rms
CBMV	**13.69%**	**5.35%**	**1.56%**	**0.71**	**14.52%**	**5.97%**	**1.97%**	**0.86**
SGM_ROB	19.52%	10.08%	4.07%	1.89	20.33%	10.77%	4.67%	2.11
MeshStereo	22.27%	11.52%	5.78%	1.21	22.95%	11.94%	6.09%	1.29
SPS-STEREO	55.62%	15.04%	3.08%	1.07	56.02%	15.83%	3.67%	1.22
SGM-STEREO	54.67%	15.62%	4.39%	1.83	55.54%	17.25%	6.27%	2.67
ELAS	33.66%	16.72%	8.05%	1.89	34.78%	17.99%	9.07%	1.52

Table 1. Results of our method (CBMV) on ETH3D two-view benchmark. Our method outperforms all other methods by a large margin. All our submissions, including on ETH3D, use the same model, trained on the Middlebury 2014 dataset. The methods are sorted based on the main validation metric: bad 1.0 out-noc.

testing stereo pairs. The ground truth is dense and generated by a Faro Focus X 330 laser scanner. The ground truth for the training set is publicly available,while for the test set, it has not been released.

Experimental Setup. To compute the initial matching volumes on the Middlebury data using the four block matching algorithms, we set the width of the matching windows to: 3×3 for *NCC*, 5×5 for *ZSAD*, 11×11 for *CENSUS* and 5×5 for *SOBEL*. The σ parameter in Eq. (4) was set to 0.02 for *NCC*, 100 for *ZSAD* and *SOBEL* and 8 for *CENSUS*. Parameters for the KITTI data are similar and are included as supplementary material.

On the Middlebury 2014 dataset, we use three-fold cross-validation to train our RF classifier. We split the training set into three sets of five images. Two of these sets of five and the set of 13 additional images, which are available with the dataset but are not evaluated, are used for training during each fold of the cross-validation process, while the remaining five images are used for testing. Before the final testing phase on the Middlebury test set, we train our classifier on all 28 training stereo pairs. Due to the availability of more data, we use two-fold cross validation on the KITTI datasets. We did not train on the ETH3D dataset.

We tune the optimization parameters (see Section 5) separately for each dataset, as in [51]. A complete hyper-parameter configuration is provided in the supplement.

	bad-2.0	bad-2.0	avgerror	rms-error
	nonocc	all	all	all
Middlebury 2014 test set				
MC-CNN-acrt	**8.08**%	**19.1**%	17.9	55.0
CBMV(ours)	11.1%	20.5%	**14.4**	**46.9**
MC-CNN-fst	9.47%	20.6%	19.3	55.7
Middlebury 2014 training set				
MC-CNN-acrt	**10.1**%	**19.7**%	11.8	36.6
CBMV(ours)	11.7%	20.3%	**11.5**	**34.9**
MC-CNN-fst	11.7%	21.5%	12.8	37.5

Table 2. Results of our method (CBMV) on the Middlebury 2014 test and training sets, compared with the results of MC-CNN-acrt and MC-CNN-fst using four different metrics.

Results. On the Middlebury data, the final disparity maps computed by CBMV have an average error rate of 11.1% and 11.7% out of non-occluded pixels on the testing and training set, respectively, with the error tolerance set to 2.0 disparity levels. These results show that our method can produce competitive results with the MC-CNN-fst architecture. However, our method outperforms MC-CNN-fst on both training and testing sets when we consider all pixels and shows competitive results to the MC-CNN-acrt architecture. Moreover our method ranks higher than both MC-CNN-fst and MC-CNN-acrt with respect to average and RMS error. Table 2 shows a comparison of our method with MC-CNN-acrt and MC-CNN-fst. Figure 5 shows disparity maps generated by our method with corresponding error maps on the Middlebury 2014, KITTI 2012 and KITTI 2015 datasets.

Generalization. Being able to evaluate a method on the available benchmarks is very important since a it allows fair, comprehensive comparisons with other methods. However, benchmarks cannot capture all the difficulties involved when deploying a method on the field. To evaluate the transferability of our method, we use our trained RFs on the three different datasets, Middlebury 2014, KITTI 2012 and KITTI 2015, and test on the corresponding unseen training sets. As an example, we use the RFs from KITTI 2012 and KITTI 2015 to test on the Middlebury 2014 training set. The same approach is used for the downloaded MC-CNN-fst models, which serve as baselines. Since the optimization and post-processing stage is an integral part of both methods, for fairness and to avoid inconclusive results due to hyper-parameter tuning, we keep the hyper parameter values that worked best when the method is applied to the same dataset. More precisely when we use the RF trained on KITTI 2012 to test on the Middlebury 2014 training set, we use the Middlebury 2014 best hyper-parameters during testing. We were unable to run MC-CNN-acrt KITTI models on the Middlebury dataset due to the limited amount of global memory on the GPU, but we include results based on the numbers reported in [51]. It is worth noting that MC-CNN-acrt is significantly worse that MC-CNN-fst in this particular experiment, which shows that MC-CNN-acrt specializes even more to the particular training dataset to achieve higher ac-

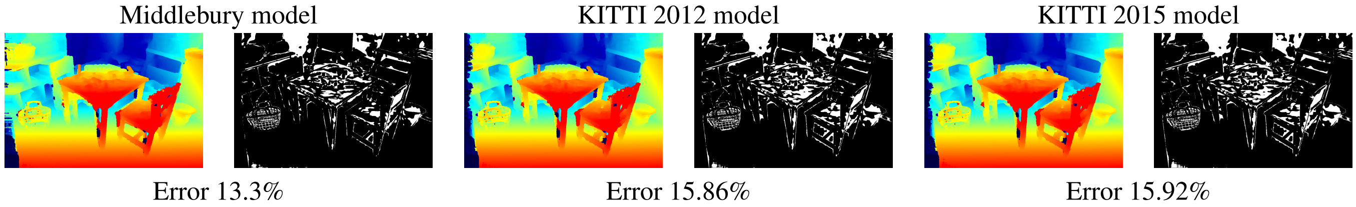


Figure 4. Generalization examples of CBMV on the Playtable data from Middlebury. Our method is robust when tested in different domains, especially compared to MC-CNN. The corresponding error rates at a 2-disparity level tolerance for MC-CNN-fst are 18.0% , 41.43% and 38.67% respectively.

		Test set								
		KITTI 2012 (Out-Noc)			KITTI 2015 (Out-All)			Middlebury (bad 2.0)		
		MC-ac	MC-fst	CBMV	MC-ac	MC-fst	CBMV	MC-ac	MC-fst	CBMV
Training set	KITTI 2012	0%	0%	0%	23.07%	13.28%	-0.41%	40.20%	33.58%	7.00%
	KITTI 2015	63.98%	17.54%	3.02%	0%	0%	0%	79.39%	41.62%	7.69%
	Middlebury	17.62%	10.51%	-4.62%	38.15%	18.79%	-2.09%	0%	0%	0%

Table 3. Quantitative generalization results for CBMV, MC-CNN-fst and MC-CNN-acrt. This table shows relative increases in error rate when the training set is different than the test set. For example, the rightmost column means that the CBMV error rate increases by 7% when trained on KITTI 2012 and tested on Middlebury, compared to training and testing on Middlebury. For MC-CNN-acrt we use data from Table 10 of [51].

curacy.

Table 3 shows that our method adapts much better to new unseen environments. Figure 4 shows results on a particular example where MC-CNN-fst has a dramatic drop in accuracy. In Table 3 we can see that our Middlebury RF outperforms the RFs trained on KITTI 2012 and 2015. We believe that this behavior can be attributed to that fact that Middlebury 2014 dataset contains much more diverse scenes, thus the classifier can generalize better.

We submitted results to the KITTI 2012, KITTI 2015 and ETH3D test sets using the RF trained on the Middlebury 2014 dataset and optimization and post-processing parameters tuned on each target dataset. The error rates of our disparity maps are 3.56% and 4.73%, on non-occluded and on all pixels on KITTI 2012, and 4.58% and 5.06% on non-occluded and on all pixels on KITTI 2015. Table 1 shows a comparison of our method with other methods on the ETH3D benchmark. Our method outperforms every other method by a large margin. We are not aware of any other submission that was not trained on data from KITTI or ETH3D itself. Please visit the KITTI and ETH3D websites for comparisons with other methods.

Runtime. At first glance our method seems very expensive both in computation and space requirements. This is partially true. Computing the four initial cost volumes is very efficient and can be done in parallel. In our implementation the total time spend computing the four initial costs is approximately 2.5 seconds for a KITTI stereo pair. The feature extraction process takes about 7 seconds. To lower the space requirements, features can be extracted in batches of image rows. The bottleneck of our method is the random forest classifier which takes approximately 162 seconds. A better implementation of the random forest where training is done on the CPU and inference is performed on the GPU is feasible but out of the scope of the current paper. Moreover, due to the robustness of our model, an ASIC RF implementation is possible and would enable very high frame rates. Most optimization and post-processing steps have to be executed for both the left and right disparity map, but they only take a few seconds. The total runtime of our method on KITTI is 250 seconds. Our complete implementation including the trained RF model is available at `https://github.com/kbatsos/CBMV`.

7. Conclusions

We have proposed a novel approach for estimating a bidirectional matching volume by coalescing matching and confidence data generated by applying conventional matching functions on rectified stereo pairs. We have evaluated the accuracy and the generalizability of this approach quantitatively and qualitatively.

Comparing the results of CBMV with those of MC-CNN on the 2014 Middlebury benchmark, we observe that CBMV is superior with respect to average and RMS disparity errors when all pixels are considered. Considering other error metrics on both non-occluded and all pixels, the ordering of the two MC-CNN architectures and CBMV fluctuates. It would be fair to say that MC-CNN-acrt is the most accurate overall, with the other two methods being essentially tied.

The advantage of our method lies in its generalizability. According to Tonioni et al. [44], end-to-end deep architectures [21] tend to specialize on their training domain. MC-CNN is better suited for previously unseen domains, but as we have shown in Table 3, our method generalizes

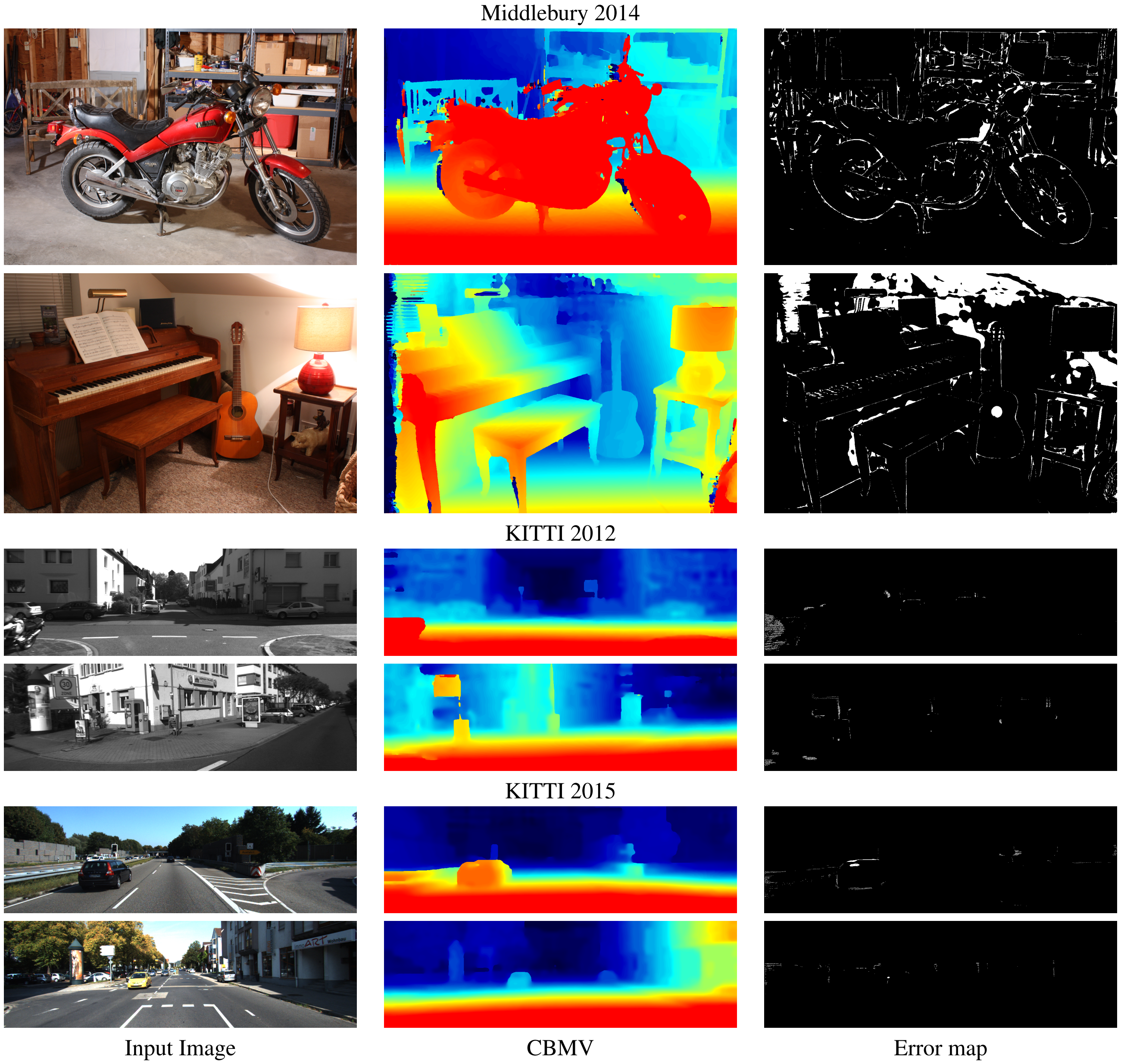


Figure 5. Results of our method (CBMV) on the three datasets: Middlebury 2014, KITTI 2012 and KITTI 2015. Note that the KITTI disparity maps were generated using an RF trained on Middlebury 2014.

much better. Training on the Middlebury data resulted in even higher accuracy on the KITTI benchmark than training on the target dataset itself. Transferring RFs in the other direction led to a small loss of accuracy. *The strength of our approach is that, due to its design that avoids learning directly from image appearance, trained classifiers can be applied in domains without ground truth data.* This is a critical feature for being able to apply a method on images taken in the field (not necessarily in real time).

In our future work we plan to investigate ways of combining the generalization properties of our approach with the advantages of end-to-end deep learning architectures.

Acknowledgments. Research reported in this publication was supported by the National Science Foundation under Awards IIS-1527294 and IIS-1637761. We are grateful to Enrique Dunn for constructive discussions, especially on the name of our approach.

References

[1] K. Alahari, C. Russell, and P. Torr. Efficient piecewise learning for conditional random fields. In *CVPR*, pages 895–901, 2010.

[2] J. T. Barron and B. Poole. The fast bilateral solver. In *ECCV*, 2016.

[3] L. Breiman. Random forests. *Machine Learning Journal*, 45:5–32, 2001.

[4] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang. A deep visual correspondence embedding model for stereo matching costs. In *ICCV*, pages 972–980, 2015.

[5] J. Cruz, G. Pajares, J. Aranda, and J. Vindel. Stereo matching technique based on the perceptron criterion function. *Pattern Recognition Letters*, 16(9):933 – 944, 1995.

[6] S. Drouyer, S. Beucher, M. Bilodeau, M. Moreaud, and L. Sorbier. Sparse Stereo Disparity Map Densification using Hierarchical Image Segmentation. In *International Symposium on Mathematical Morphology*, pages 172–184, 2017.

[7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, pages 1229–1235, 2013.

[8] S. Gidaris and N. Komodakis. Detect, replace, refine: Deep structured prediction for pixel wise labeling. In *CVPR*, 2017.

[9] R. Gouveia, A. Spyropoulos, and P. Mordohai. Confidence estimation for superpixel-based stereo matching. In *International Conference on 3D Vision (3DV)*, pages 180–188. IEEE, 2015.

[10] R. Haeusler, R. Nair, and D. Kondermann. Ensemble learning for confidence measures in stereo vision. In *CVPR*, 2013.

[11] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*, pages 3279–3286, 2015.

[12] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *PAMI*, 30(2):328–341, 2008.

[13] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. *PAMI*, 34(11):2121–2133, 2012.

[14] 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. In *ICCV*, 2017.

[15] K.-R. Kim and C.-S. Kim. Adaptive smoothness constraints for efficient stereo matching using texture and edge information. In *International Conference on Image Processing (ICIP)*, pages 3429–3433, 2016.

[16] P. Knöbelreiter, C. Reinbacher, A. Shekhovtsov, and T. Pock. End-to-end training of hybrid cnn-crf models for stereo. In *CVPR*, 2017.

[17] D. Kong and H. Tao. A method for learning matching errors for stereo computation. In *BMVC*, 2004.

[18] L. Li, X. Yu, S. Zhang, X. Zhao, and L. Zhang. 3d cost aggregation with multiple minimum spanning trees for stereo matching. *Applied Optics*, 56(12):3411–3420, 2017.

[19] Y. Li and D. Huttenlocher. Learning for stereo vision using the structured support vector machine. In *CVPR*, 2008.

[20] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *CVPR*, 2016.

[21] 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 *CVPR*, 2016.

[22] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *ICCV Workshops*, pages 467–474, 2011.

[23] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[24] C. Pal, J. Weinman, L. Tran, and D. Scharstein. On learning conditional random fields for stereo: Exploring model structures and approximate inference. *IJCV*, 99(3):319–337, 2012.

[25] 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*, 2017.

[26] H. Park and K. M. Lee. Look wider to match image patches with convolutional neural networks. *IEEE Signal Processing Letters*, 24(12):1788–1792, 2017.

[27] M.-G. Park and K.-J. Yoon. Leveraging stereo matching with learning-based confidence measures. In *CVPR*, pages 101–109, 2015.

[28] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui. Towards a simulation driven stereo vision system. In *ICPR*, pages 1038–1042, 2012.

[29] M. Poggi and S. Mattoccia. Deep stereo fusion: combining multiple disparity hypotheses with deep-learning. In *International Conference on 3D Vision (3DV)*, 2016.

[30] M. Poggi and S. Mattoccia. Learning a general-purpose confidence measure based on o (1) features and a smarter aggregation strategy for semi global matching. In *International Conference on 3D Vision (3DV)*, 2016.

[31] M. Poggi and S. Mattoccia. Learning from scratch a confidence measure. In *BMVC*, 2016.

[32] M. Poggi and S. Mattoccia. Learning to predict stereo reliability enforcing local consistency of confidence maps. In *CVPR*, 2017.

[33] M. Poggi, F. Tosi, and S. Mattoccia. Quantitative evaluation of confidence measures in a machine learning world. In *ICCV*, 2017.

[34] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, 2014.

[35] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[36] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] A. Seki and M. Pollefeys. Patch based confidence prediction for dense disparity map. In *BMVC*, 2016.

[38] A. Seki and M. Pollefeys. Sgm-nets: Semi-global matching with neural networks. In *CVPR*, 2017.

[39] A. Shaked and L. Wolf. Improved stereo matching with constant highway networks and reflective confidence learning. In *CVPR*, 2017.

[40] A. Spyropoulos, N. Komodakis, and P. Mordohai. Learning to detect ground control points for improving the accuracy of stereo matching. In *CVPR*, pages 1621–1628, 2014.

[41] A. Spyropoulos and P. Mordohai. Ensemble classifier for combining stereo matching algorithms. In *International Conference on 3D Vision (3DV)*, pages 73–81. IEEE, 2015.

[42] A. Spyropoulos and P. Mordohai. Correctness prediction, accuracy improvement and generalization of stereo matching using supervised learning. *IJCV*, 118(3):300–318, 2016.

[43] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura. Continuous 3d label stereo matching using local expansion moves. *PAMI*, 2017.

[44] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano. Unsupervised adaptation for deep stereo. In *ICCV*, 2017.

[45] H. Trinh and D. McAllester. Unsupervised learning of stereo vision with monocular depth cues. In *BMVC*, 2009.

[46] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun. Continuous markov random fields for robust stereo estimation. In *ECCV*, pages V: 45–58, 2012.

[47] X. Ye, J. Li, H. Wang, H. Huang, and X. Zhang. Efficient stereo matching leveraging deep local and context information. *IEEE Access*, 5:18745–18755, 2017.

[48] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, pages B:151–158, 1994.

[49] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015.

[50] J. Žbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, 2015.

[51] J. Žbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *The Journal of Machine Learning Research*, 17(65):1–32, 2016.

[52] K. Zhang, J. Lu, and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(7):1073–1079, 2009.

[53] L. Zhang and S. Seitz. Estimating optimal parameters for mrf stereo from a single image pair. *PAMI*, 29(2):331–342, 2007.