Ensemble Classifier for Combining Stereo Matching Algorithms

Aristotle Spyropoulos Philippos Mordohai
Stevens Institute of Technology
{ASpyropo, Philippos.Mordohai}@stevens.edu

Abstract

Stereo matching, as many problems in computer vision, has been addressed by a multitude of algorithms, each with its own strengths and weaknesses. Instead of following the conventional approach and trying to tune or enhance one of the algorithms so that it dominates the competition, we resign to the idea that a truly optimal algorithm may not be discovered soon and take a different approach. We present a novel methodology for combining a large number of heterogeneous algorithms that is able to clearly surpass the accuracy of the most accurate algorithms in the set. At the core of our approach is the design of an ensemble classifier trained to decide whether a particular stereo matcher is correct on a certain pixel. In addition to features describing the pixel, our feature vector encodes the agreement and disagreement between the matcher under consideration and all other matchers. This formulation leads to high accuracy in disparity estimation on the KITTI stereo benchmark.

1. Introduction

Several decades of research on stereo matching [32, 11] have led to undeniable progress, but also to the conclusion that different stereo matching algorithms have different strengths and weaknesses and that assigning correct disparities to pixels is a task that has a varying degree of difficulty depending on properties of the scene and the images. For example, early MRF-based stereo approaches [3] and local, winner-take-all (WTA) methods using large aggregation windows work better on smooth surfaces of piece-wise constant disparity at the expense of pixels on thin structures that are often merged on nearby larger surfaces. On the other hand, WTA methods with smaller aggregation windows exhibit the opposite behavior.

In this paper we address binocular stereo matching from a different perspective: instead of trying to improve the accuracy of a single stereo matcher, we present an approach for combining a large number of matchers to obtain one with higher accuracy than any of the constituent matchers. Similar research has been published [27], but the combination is based on handcrafted rules, which are designed according to the intuition of the researchers. We would like our approach to be extensible without manual intervention when the pool of available stereo matchers is modified. This pool may contain a large number of matchers, out of which our algorithm would select an appropriate number of matchers, the *active set*, and then train one classifier per active matcher to determine how disparity values should be automatically assigned to pixels.

Our hypothesis is that a competitive stereo matching system can be constructed by combining multiple local stereo matching methods in a principled way. We pose the problem as multi-class classification in which the available class labels for a pixel are disparity values proposed by different matchers in the active set. In this process, the classifiers are considered *black boxes*, in other words no information from each matcher is used other than the disparity value assigned to each pixel. We describe an implementation of this approach in Section 5 and validate it experimentally in Section 6 using data from the KITTI stereo benchmark [11].

An important aspect of the approach is that we do not treat the matchers in the active set, and the classifiers in the ensemble, as independent. Instead, we capture the agree-

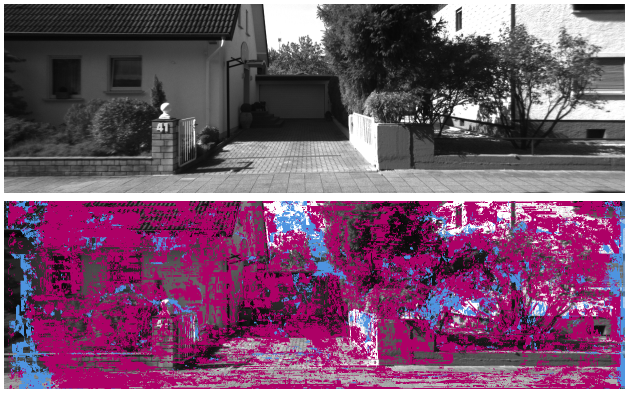


Figure 1: An input frame and a visualization showing pixels colored according to the matcher that was selected for them by the classifiers. Pixels from only two matchers are shown for clarity.

ment and disagreement among diverse stereo matching algorithms. Given a set of m *active matchers*, we train m one-against-all classifiers. Each classifier aims to predict whether the disparity proposed by its *primary matcher* for a particular pixel is correct or not. The feature vectors that are used as inputs to these classifiers include variables that represent whether other stereo matchers agree with the disparity proposed by the primary matcher or not. As a result, our model that jointly considers all matchers is a more accurate representation of the underlying phenomena. This formulation provides powerful information that increases the accuracy of the classifiers. The final disparity value assigned to the pixel is that of the matcher with the highest posterior probability according to the classifiers after calibration [39]. Figure 1 shows an example for one stereo pair. Each pixel is colored according to the matcher with the highest posterior. The disparity of the selected matcher is assigned to the pixel. Our results show that the ensemble reduces the error rate of the most accurate matcher by 38%.

2. Related Work

In this section, we review research on the application of machine learning techniques on stereo and on algorithm selection for stereo and other related problems. Relevant to our research is the literature on confidence estimation [18, 13, 28] and on learning optimization or regularization parameters [41, 37] for stereo. The latter methods aim to learn a small number of global parameters, such as the weights of the data and smoothness terms of an MRF, while our work aims to train classifiers that make a decision per pixel based on local features and context.

In early work, Lew et al. [24] presented an approach for selecting a set of features that form an effective descriptor for stereo matching. Cruz et al. [6] addressed the problem of determining whether a match in edge-based stereo was correct using four features extracted by filtering the images and a perceptron. Sabater et al. [31] introduced an a contrario approach for validating the correctness of stereo matches. A user-specified acceptable number of false matches determines the density of the final disparity map. Zhu et al. [42] fit linear regression models to local image regions. We view this approach as similar to ours since it also assumes that a single model does not work well for all pixels. Recently, Haeusler et al. [14] presented a learning approach that can predict the correctness of the output disparities of the semi-global matching (SGM) stereo algorithm [15]. It uses a number of features computed on the images, disparity maps and matching cost volume and a random forest classifier that is able to reject false matches with high accuracy. Spyropoulos et al. [34] use a random forest to predict match correctness and to select ground control points which provide constraints that improve the overall accuracy. Zbontar and LeCun [40] trained a convolutional neural network (CNN) to predict whether two image patches match or not. The trained CNN generates matching costs which are adaptively aggregated and optimized using SGM.

Approaches using multi-class classification include the following. Kong and Tao [21] used non-parametric techniques to learn the probability of a potential match to belong to three categories: correct, wrong due to foreground overextension or wrong for other reasons. Ladicky et al. [23] address simultaneous semantic segmentation and dense 3D reconstruction from two images, but make strong assumptions about the scene layout and composition in terms of types of possible objects and surfaces. Gehrig and Scharwächter [10] cast optical flow error prediction as multi-class classification where the classes are different ranges of the error.

Related to the above are methods for selecting the best among multiple algorithms for a given task. Kong and Tao [22] selected among 36 matchers which were variations of normalized cross-correlation-based matching in different window sizes and with different locations of the reference pixel in the windows. Motten et al. [27] presented a decision tree classifier implemented on FPGA that selects among multiple disparity hypotheses generated by trinocular stereo. Stenger et al. [35] select trackers either based on maximum confidence or by forming a cascade and deciding whether to accept the current tracker or proceed to the next one, hence avoiding the execution of all trackers a priori. Gao et al. [9] proposed a method for forming an ensemble of trackers that considers the reliability of each tracker individually and also the correlations of pairs of trackers.

The work of Mac Aodha et al. [26] is similar to ours, as it uses a multi-class classifier that selects among four state of the art methods for optical flow estimation. They also make decisions at the pixel level using a random forest. The major differences with our approach are that the complementarity of the algorithms is not considered, the classes are defined with respect to different thresholds on the endpoint error and pixels are used for training only when the top two algorithms disagree. Mac Aodha et al. train one classifier per optical flow estimator to generate a confidence that the estimated optical flow for a particular pixel is below some error threshold. As shown in Section 5, we do not consider the matchers independent but encode whether they agree or not. This provides valuable information that makes our final disparity assignments consistently better than all input matchers, while the combination of [26] is second best on all sequences. In subsequent work, Mac Aodha and Brostow [25] proposed cost sensitive learning for selecting among multiple experts. Incorporating example dependent costs instead of binary correctness labels is an interesting direction for future research.

3. Problem Statement

The objective of our research is to test the hypothesis that a competitive stereo matching system can be constructed by combining multiple stereo matching algorithms in a principled way without requiring domain expertise from the users. The combination is done in two stages. In the first stage the set of *active matchers* is selected from a large pool of matchers. In the second stage, a set of classifiers are trained to select the disparity that is more likely to be correct for a given pixel from a set of disparities proposed by the active matchers. The first stage is necessary for practical purposes since running hundreds of stereo algorithms on every input stereo pair is cumbersome and unnecessary.

In Section 4, we present the algorithms we used to construct the pool of matchers as well as the features we compute on them. The features capture information on the correctness of each matcher for a given pixel.

One-against-all classifiers are trained on feature vectors that combine information from all active matchers according to Section 5. Each classifier corresponds to a *primary matcher* and determines whether the disparity proposed by the primary matcher for a given pixel is correct or not. The novelty of our classification scheme is in the way we encode agreement between different matchers in the feature vector and in that we do not treat the predictions of the matchers as independent. Its effectiveness was tested with a number of experiments on the KITTI benchmark [11] (Section 6).

4. Stereo Matchers and Features

In Section 4.1 we present the stereo matchers that formed the initial pool out of which a small number was selected for the classifiers. We implemented several of these matchers and used publicly available software for the others. In Section 4.2, we present the features that were computed on the disparity maps and used in the classifiers of Section 5.

4.1. Stereo Matchers

We implemented the following local stereo matching techniques for gray-scale images, as in the KITTI data set. In all cases we used square windows ranging from 3×3 to 21×21 increasing their width in steps of 2 pixels. Due to space considerations, we do not provide implementation details here, but refer readers to [16].

SAD: The sum of absolute differences (SAD) of intensity. We denote SAD in 7×7 windows by SAD_7, for example.

SSD: The sum of square differences (SSD) of intensity.

Sobel: The sum of absolute differences of the responses to a vertical edge filter. Similarly to [30], we compute the intensity gradient in the x direction using the Sobel filter. We then treat filter responses as intensities and compute SAD on them. Note that the cost does not include any direct comparison of intensity values.

ZNCC: Zero-mean Normalized Cross-Correlation. Unlike the above matchers, ZNCC produces a score with larger values indicating better matching.

SNCC: The Summed Normalized Cross-Correlation method was proposed by Einecke and Eggert [8]. SNCC is a two-stage process, where in the first stage ZNCC is computed in small matching windows and in the second stage the ZNCC values are aggregated by summation in windows of potentially different size. This approach is more effective than computing ZNCC in large windows because it mitigates an intrinsic drawback of ZNCC: the fact that it locks on edges of high contrast which suppress the rest of the signal in the matching window. By breaking up the large matching window into smaller subwindows, this effect is restricted to a small number of subwindows. Summing the ZNCC scores of the subwindows results in the desired degree of smoothness without suffering as much from errors due to high contrast edges. In this paper, we adopt the practice of [8] and compute ZNCC in 3×3 or 5×5 subwindows. SNCC_{3-11} in our notation denotes 3×3 ZNCC subwindows and 11×11 summation windows.

Census: The census transform has been very effective in stereo matching due to its robustness to illumination variations. We use the Hamming distance between census transform descriptors of corresponding pixels and aggregate the absolute values of these distances in a second stage by adding them. As above, Cen_{3-11} in our notation denotes the census transform computed in 3×3 windows around each pixel and the final cost resulting from distance aggregation in 11×11 windows.

Shiftable windows: To generate correct disparities near discontinuities we applied shiftable windows on the above methods. A shiftable window is one in which the reference pixel that we are trying to match can shift inside the window instead of being fixed to the center. The motivation is that by using several windows that include the pixel we are trying to match, and not just the window centered at that pixel, we increase the probability of including only pixels from one of the surfaces that are near the discontinuity, making matching easier. The shiftable version of SAD_3, for example, is obtained by searching for the minimum cost in 3×3 windows and is denoted by SH-SAD_3.

We also included the following advanced matchers.

MRF: This is an MRF with a data term computed using NCC_5 and edge weights modulated by the intensity difference between adjacent pixels to favor disparity discontinuities aligned with image edges. The optimal disparity assignment is computed using the software of [20].

rSGM: Semi-Global Matching [15] is an efficient approximation for optimizing an MRF. We use the rSGM implementation of Spangenberg et al. [33] which uses the census transform to compute the data term. We denote by

rSGM_5 the version that uses Cen_5 as the matching cost and by rSGM_C the version that uses census in 9×7 center-symmetric windows with horizontal weights.

FCVF: The Fast Cost-Volume Filtering method [30, 17] is based on a bilateral filter that operates on the cost volume to adaptively aggregate costs and produces very competitive WTA disparity maps. Here we use the Matlab implementation provided by the authors. The initial cost is a blend of intensity and gradient differences.

ELAS: Geiger et al. [12] proposed ELAS which detects an initial set of reliable matches and then forms a set of triangles that cover the remaining ambiguous pixels. The implementation of ELAS provided by the authors is included to tackle textureless, planar surfaces.

DAISY: Tola et al. [36] proposed DAISY as a local descriptor for dense wide-baseline matching. We apply the authors' implementation to our narrow-baseline inputs.

Superpixels: We also applied a simple superpixel-based algorithm on all the above matchers as follows: we segmented the reference images into SLIC superpixels [1] and then fitted a plane to each segment using RANSAC on the disparities of the segment. This step doubles the set of matchers and further enhances its diversity, while in many cases it also improves the accuracy compared to the original disparity maps. We denote the new matchers by SUPER-SAD_{11}, for example.

We compute disparity maps using all of the above matchers and then reverse the role of the reference and target image to compute right-to-left disparity maps. Disparity maps from local matchers are filtered to generate their shiftable counterparts, while all disparity maps from local, shiftable and advanced matchers, are post-processed to generate the superpixel versions of their outputs. In total, we generate 122 left-to-right disparity maps for each input stereo pair.

4.2. Features

In order to form the feature vectors for the classifiers, we compute for each pixel a few simple features, separately for each matcher, using only the disparity maps. It is important to note here that *matchers are treated as black boxes* that generate a single disparity for each pixel and provide no access to intermediate results. We do this for practical reasons: to avoid storing matching cost volumes for all local matchers, but also because global optimization methods produce neither posterior probabilities for the disparity assigned to a particular pixel nor ranked lists of likely disparities for each pixel. This causes features based on intermediate matching results, such as the ratio of best and second best costs [18], to be unavailable. We also excluded the cost maps of the WTA methods, since such maps are not produced by the global methods. According to Haeusler et al. [14], it is very likely that features based on the cost volume are more effective, but as shown in Section 6, our approach is able to discern the correct disparities relying on consensus and disagreement among different matchers without such features. In this paper, we only used the following features for individual disparities.

Distance from Discontinuity (DD): Pixels near depth discontinuities are likely to be mismatched. Since we do not know the true discontinuities, we use the WTA disparity estimates as a proxy and mark as discontinuous any pixel whose disparity is not equal to the disparities of all of its four neighbors. DD is equal to the horizontal distance from the pixel to the nearest discontinuity.

Left-Right Consistency (LRC): Pixels with inconsistent disparities in the left and right disparity maps computed by the same matcher are likely to be unreliable. We implemented LRC as a binary feature which was set to true when the difference between the disparity d of a pixel (x_L, y) in the left image and the disparity at pixel $(x_L\text{-}d, y)$ in the right image is less than or equal to 1.

Additional features, such as computing a matching cost value for each given disparity using a common local matching function (NCC_5) and responses of filters such as the median on the disparity map and several forms of priors learned on the entire training set, did not appear to be effective in our experiments.

5. Classifier Design

Here, we describe the classifier ensemble we designed to select among multiple candidate disparities for each pixel. We begin with a technique to select which of the over 100 available matchers to use as inputs to the classifiers.

5.1. Matcher Selection

The objective of this stage is to limit the set of matchers for two practical reasons: to maintain the computational cost at manageable levels and also to avoid having to solve a multi-class classification problem with a very large number of classes. Using hundreds of matchers would result in an unnecessarily complex, highly redundant processing pipeline that would be hard to deploy. Here, we propose an algorithm for selecting a small number of matchers, which will be referred to as the *active matchers*. Their disparity maps are the only inputs to feature computation and subsequently to the classifiers.

The obvious requirement for selecting the active matchers is high accuracy. This, however, is not sufficient since our classifiers only select among the disparities proposed by the active matchers. In order to raise the upper bound on the accuracy of the ensemble, we should increase the number of pixels for which the correct disparity is among those

proposed by the matchers. Therefore, including highly accurate matchers with very large overlaps with the current active set is not necessarily beneficial.

We form the set of active matchers as follows: we select the matcher with the highest overall accuracy first and then add the matcher that has the highest accuracy over the pixels on which the first matcher failed. We repeat this procedure for a few rounds, each time selecting the matcher that would result in the largest increase in the number of pixels for which the correct disparity appears at least once. The top matcher (SUPER-rSGM$_5$) has a coverage of 91.9% of all pixels of the validation set (last 97 stereo pairs of KITTI training set). If all 122 matchers were to be considered, the combined coverage would have been 99.55%. The first eight matchers we selected for our experiments, using the above technique, cover 98.57% of all pixels.

5.2. One-against-all Classifiers with Agreement Features

We would like to train a classifier ensemble that can take as input a number of disparity maps, along with the associated feature maps, and select one of the proposed disparities for each pixel. The challenge is that this is *a multi-class classification problem in which the classes are not mutually exclusive* since multiple active matchers can be correct at the same time. Of course, in some cases none of the active matchers may be correct. When this occurs, we cannot determine the correct disparity without post-processing. This is why it is important to maximize the fraction of pixels on which at least one matcher is successful, as in Section 5.1. While we could have used error-correcting output coding [7] to handle non-mutually exclusive classes, this would have resulted in over one hundred classifiers, many of which would suffer from severe lack of training data.

We opt for a one-against-all design for our classifiers, with each class corresponding to the selection of a particular active matcher and the disparity it proposes for the pixel under consideration. Given m matchers in the active set, we train m binary classifiers using features from all matchers as shown below. Each of these one-against-all classifiers is tasked with estimating the likelihood of its *primary matcher* being correct based on features describing its own disparity estimate for a pixel, as well as features from the other active matchers, referred to as *secondary matchers*, for the same pixel. In this setup, the training label is true if the primary matcher is correct regardless of whether other matchers are also correct. The m classifiers assign scores to the proposed disparities and the disparity with the highest score is selected and assigned to the pixel. (Note that the matchers provide no specific information for the majority of possible disparity values for a pixel, since only at most m disparities are selected. Using disparities as labels does not lead to a well formulated problem.)

We propose a novel way to benefit from agreements among the active matchers in a one-against-all scheme. This is accomplished by introducing *agreement features* in the feature vector of each classifier. These are binary variables indicating whether each secondary matcher agrees with the primary matcher within a given tolerance. We experimented with different values of the tolerance and settled on 3 which matches the evaluation protocol [11]. Differences with other small values of tolerance are small. In the context of [19], this formulation does not assume independence among the matchers and reasons on joint probabilities. If we were to assume independence, Kittler et al. [19] suggest combining the classifiers via majority voting since only decision outcomes are available to us. We compare majority voting with our formulation in Section 6.

The feature vector for each classifier then consists of the following:

- *agreement features* a_i for each secondary matcher. a_i is equal to 1, if the primary matcher agrees with matcher i, and -1, otherwise. There are m-1 agreement features.
- *individual-matcher features* for the primary and secondary matchers. There are $f \cdot m$ such features that are computed from the disparity maps as in Section 4.2. represent In this paper we use two features ($f = 2$): DD and LRC.
- *product features* which are the products of the agreement features with the corresponding individual-matcher features. As a result, feature values corresponding to secondary matchers in disagreement are negated. These features capture whether different matchers strongly or weakly agree or disagree with the primary matcher. There are $f \cdot m$ such features.
- a *total support feature (TS)* defined as $TS = \Sigma a_i$, for $a_i > 0$. TS encodes the support the primary matcher has received from the secondary matchers, i.e. it is the sum of all positive a_i.

Due to the inhomogeneity of our feature vector, we choose a random forest [4, 5] as the classifier since it does not require a metric in feature space. We trained random forests comprising 50 trees since increasing the number of trees did not improve performance. All parameters were selected by cross validation. RF averages the predictions of the trees to assign a score between 0 and 1 to each test pattern, a disparity proposed by an active matcher here. The closer the score is to 1, the more confident are we that the disparity of the primary *classifier* is correct. We train m RFs on all pixels with ground truth in the training set. During testing, each pixel of the test set is presented to the m RFs, with a different feature vector and potentially different disparity for each one depending on the primary matcher for that RF. Each RF assigns a score to the proposed disparity and the maximum score is selected.

While the scores computed by the RFs are supposed to correspond to posterior probabilities of the disparities being correct, in practice they do not [29, 38]. This is mostly due to the classifiers being optimized for classification accuracy and not posterior estimation, and to inaccurate approximations such as independence assumptions. Uncalibrated scores are still useful for ranking disparities for a single matcher but they are suboptimal for performing comparisons across different classifiers. To rectify this situation we perform *classifier calibration* using the pair-adjacent violators (PAV) algorithm [39]. PAV finds a step-wise constant non-decreasing function that optimally maps the raw classifier scores to posterior probabilities in the mean square sense. This leads to an improvement in the final disparity selection step.

6. Experimental Validation

We perform experiments on the binocular KITTI data set [11], which comprises 194 training and 195 test gray-scale stereo pairs captured from a vehicle. The ground truth is semi-dense covering approximately 30% of all pixels concentrated in the lower part of the images and it is only available for the training set. We further divide the training images into a *training set* that contains the first 97 stereo pairs (or about 13 million non-occluded pixels) and a *validation set* containing the last 97 stereo pairs of the data with publicly available ground truth. This allows us to validate our algorithms while complying with the submission policy of the KITTI benchmark. According to the evaluation protocol of the benchmark, we consider a disparity correct if it is within three levels from the ground truth.

Applying the procedure of Section 5.1 to the 122 matchers produces the sequence listed in Table 1 along with their error rates on non-occluded pixels of the validation set. The first eight matchers are shown. Note that these are not the top individually performing matchers, other than the first one, but their correct matches are complementary. A perfect ensemble of $m = 8$ RFs would be able to achieve a minimum error rate of 1.43% on all pixels by optimally combining these matchers. Our classifiers will attempt to reach this lower bound by selecting among the disparities proposed by the eight matchers with error rates ranging from 8.06% to 54.78%. Our method must achieve an error rate below 8.06% to be useful, otherwise the best strategy would be to abandon the ensemble and use the best individual matcher. As our experiments show, the ensemble was indeed able to achieve a lower error rate than that of the best matcher (5.03% vs. 8.06%).

In the first set of experiments we train different random forest classifiers to observe the effects of each feature type on accuracy using the top *six* matchers from Table 1. Table 2 shows the error rate on all, as well as on non-occluded only pixels, of the validation set (stereo images 97-193).

Matcher	Error Rate	Novelty
SUPER-rSGM$_5$	8.06 %	11,558,271 px
MRF	10.82 %	416,122 px
FCVF	22.31 %	147,758 px
DAISY	11.14 %	90,369 px
SH-ZNCC$_{21}$	29.01 %	41,837 px
SH-SOB$_{21}$	44.05 %	23,989 px
SH-SSD$_5$	54.78 %	18,159 px
ELAS	20.70 %	16,119 px

Table 1: The first eight selected matchers, their error rates over non-occluded pixels and their contributions in terms of previously unobserved pixels with correct disparity

The table reports which feature types were used for training as well as the total number of features in the feature vector. Even ensemble A that uses only the agreement features surpasses the accuracy of SUPER-rSGM$_5$. Comparing classifier ensemble B to C or D and J to K on Table 2, we see that our proposed agreement variables are more effective than TS, which collects votes in favor of a given disparity ignoring the identities of supporting matchers. However, removing both the agreement and TS features – as in ensemble M – shows a significant drop in accuracy in comparison to ensemble N that has both features included.

After selecting the best ensemble, N, we expanded the number of matchers to eight. Calibrating any of the ensembles leads to an error reduction in the order of 0.3%. Table 3 shows the error rates on all, as well as the non-occluded pixels, for experiments with varying number of matchers with and without post-processing. In Table 3 we also show results by the best individual matcher, SUPER-rSGM$_5$, for comparison. Two additional methods are included: the *Median* method selects the median of the disparities proposed by the eight matchers, and the *Majority Voting* method which selects the disparity that is recommended by the most matchers. In all three cases the error rate is significantly higher than that of our best method's, $N8$-C, before post-processing.

$N8$-CP, the best-performing ensemble based on the results on Table 3, uses the full feature vector (a_i, DD, $a_i \cdot DD$, LRC, $a_i \cdot LRC$ and TS) on eight matchers with their classifiers calibrated. We selected this ensemble to generate the final results. Figure 2 shows a visualization of the top performing ensemble on one of the input stereo pairs. When, for example, FCVF is selected for a pixel, then that pixel is colored orange in Fig. 2(c). The figure shows that different matchers are better suited for different pixel types, depending on their degree of smoothness and local illumination conditions among other factors. More importantly it shows that our classifiers are able to determine that. Note that there is variability in how often matchers are selected from stereo pair to stereo pair.

Post-processing was applied to generate the final dis-

Ensemble	a_i	DD	a_iDD	LRC	a_iLRC	TS	# of Features	Out-Noc	Out-All
A	✓						5	6.81 %	8.69 %
B	✓	✓					11	6.54 %	8.45 %
C		✓				✓	7	6.82 %	8.70 %
D	✓	✓				✓	17	6.94 %	8.70 %
E	✓		✓				11	6.49 %	8.38 %
F	✓	✓	✓				17	6.65 %	8.52 %
G	✓			✓			11	6.73 %	8.64 %
H	✓				✓		11	6.71 %	8.63 %
J	✓			✓	✓		17	6.58 %	8.50 %
K	✓			✓	✓	✓	18	6.68 %	8.59 %
L	✓		✓	✓	✓		23	6.42 %	8.33 %
M		✓	✓	✓	✓		24	6.69 %	8.56 %
N	✓	✓	✓	✓	✓	✓	30	**6.42** %	**8.32** %

Table 2: Error rates for classifier ensembles using six matchers and various feature combinations.

parity maps. We first determine which pixels require correction. For each pixel, we use the calibrated prediction score of the winning classifier as a measure of confidence. We then reject the disparities for all pixels that fall below a certain threshold and replace them similarly to [2]. For each pixel without a disparity value, we look for the nearest matched pixel to its left, since occluding surfaces are to the right in the left image, and copy its disparity. If there is no such pixel to the left, as is the case near the left border of the image, we search to the right. Finally, all disparities, existing and filled in, are iteratively filtered with a 3×13 median filter. Using a validation set, we selected 0.64 as the threshold on the prediction score. The top ensemble, $N8$-C, achieved an error rate of 5.82% for non-occluded pixels. After post-processing, the error rate was reduced further by another 0.79% to 5.03% (or lower by 3.03% compared to the best matcher).

Ensemble	Matchers	Calibrated	Out-Noc	Out-All
SUPER-rSGM$_5$	8	-	8.06 %	10.17 %
Median	8	-	8.63 %	10.64 %
Majority Voting	8	-	10.24 %	12.14 %
N	6	No	6.42 %	8.32 %
N8	8	No	6.21 %	8.21 %
N6-C	6	Yes	6.15 %	8.02 %
N8-C	8	Yes	**5.82** %	7.68 %
N6-CP	6	Yes	5.36 %	6.87 %
N8-CP	8	Yes	**5.03 %**	6.48 %

Table 3: In the first three rows we show, for comparison purposes, error rates of the best matcher, SUPER-rSGM$_5$, as well as results for the *Median* and *Majority Voting* methods. The next four rows show error rates for ensembles using the same features as ensemble N from Table 2, but with a varying number of matchers. Suffix C represents experiments with classifier calibration. Our best ensemble, $N8$-C, is superior to all top three rows. The last two rows with suffix P represent experiments where post-processing has been applied to further reduce the error rate.

We submitted results on the KITTI test data to the KITTI evaluation page using the best performing ensemble, $N8$-CP, which was trained on the complete set of 194 images. Table 4 contains the automatically generated results. At the 3-pixel error level, the error rate of our submitted results was 5.34%. Figure 3 shows two examples from the test set.

7. Conclusions

We have presented a novel approach for stereo matching that combines the strengths of a diverse set of stereo matchers in a supervised learning framework. A set of active matchers is selected from a potentially very large initial pool and random forest classifiers are trained to select the disparity that is most likely to be correct for each pixel of the left image. The classifiers, even with a small number of features compared to [14, 34], are *always able to surpass the accuracy of the best matcher in the active set*. Our method combines eight matchers and achieves an error rate before post-processing (5.82%) that is significantly smaller than that of the best individual matcher (8.06%). Using the classifier predictions to guide post-processing further reduces the error to 5.03%. We consider these as strong indications that our approach is effective in this task and by extension in our overarching goal: to develop a methodology that is able

Error	**Out-Noc**	**Out-All**	**Avg-Noc**	**Avg-All**
2 pixels	9.17 %	10.89 %	1.5 px	2.0 px
3 pixels	5.34 %	6.91 %	1.5 px	2.0 px
4 pixels	3.92 %	5.30 %	1.5 px	2.0 px
5 pixels	3.20 %	4.43 %	1.5 px	2.0 px

Table 4: Error rates on non-occluded and all pixels of the test set generated by the KITTI website for various error thresholds. The last two columns are independent of threshold.

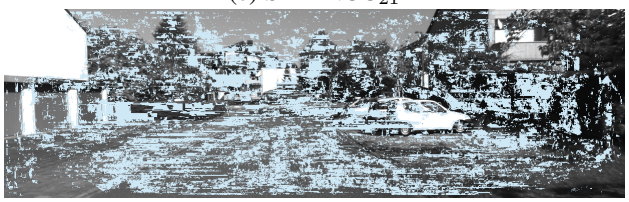
(a) SUPER-rSGM$_5$

(b) MRF

(c) FCVF

(d) DAISY

(e) SH-ZNCC$_{21}$

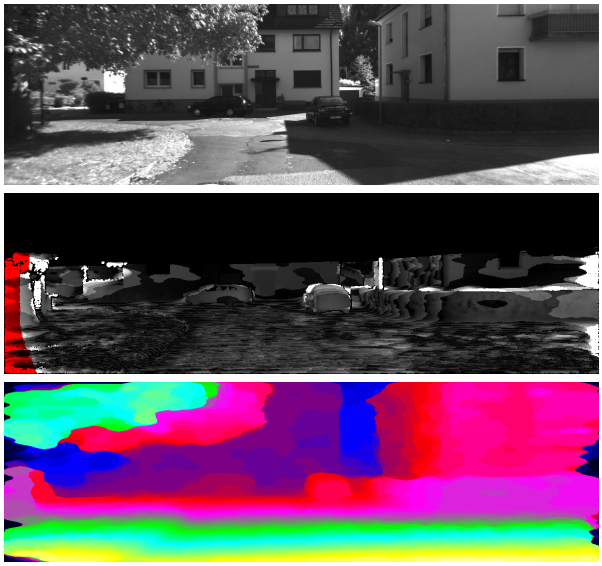


(f) SH-SOB$_{21}$

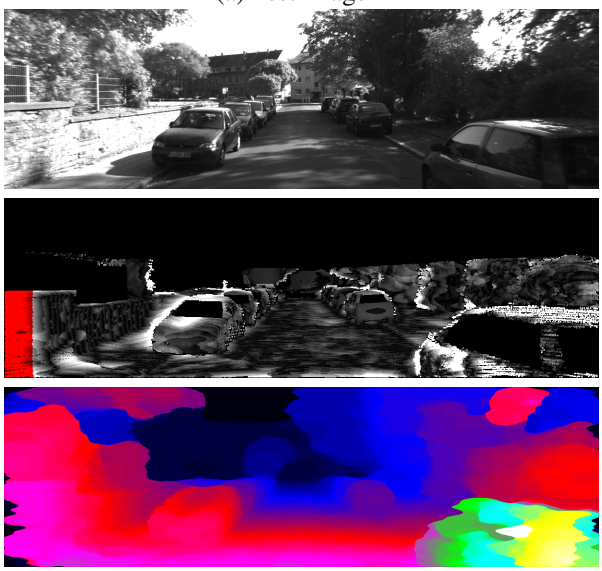
Figure 2: Each subfigure highlights the pixels that selected the corresponding disparity of each matcher. The top six (out of eight) matchers are shown. SUPER-rSGM$_5$ and SH-SOB$_{21}$ dominated the disparity selection for this stereo pair.

to benefit from the different strengths of different matchers.

Compared to the work of Mac Aodha et al. [26] on optical flow, we claim that our formulation using the one-against-all classifiers is more effective in capturing the agreement and disagreement between different matchers by not assuming that they are independent. Evidence for this can be seen by comparing ensembles M and N in Table 2.



(a) Test image 4



(b) Test image 11

Figure 3: Left image, error and disparity maps for KITTI test images 4 and 11

Acknowledgements This research has been supported in part by the National Science Foundation award #1217797 and #1527294.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11):2274–2282, 2012. 4

[2] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo: Stereo matching with slanted support windows. In *BMVC*, 2011. 7

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001. 1

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

[5] A. Criminisi and J. Shotton. *Decision forests for computer vision and medical image analysis*. Springer, 2013. 5

[6] 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. 2

[7] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995. 5

[8] N. Einecke and J. Eggert. A two-stage correlation method for stereoscopic depth estimation. In *DICTA*, 2010. 3

[9] Y. Gao, R. Ji, L. Zhang, and A. Hauptmann. Symbiotic tracker ensemble towards a unified tracking framework. *IEEE Transaction on Circuits and Systems for Video Technology*, 24(7):1122–1131, 2014. 2

[10] S. K. Gehrig and T. Scharwachter. A real-time multi-cue framework for determining optical flow confidence. In *ICCV Workshops*, pages 1978–1985, 2011. 2

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1, 3, 5, 6

[12] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *ACCV*, pages I: 25–38, 2010. 4

[13] R. Haeusler and R. Klette. Analysis of KITTI data for stereo analysis with stereo confidence measures. In *ECCV Workshops*, pages II: 158–167, 2012. 2

[14] R. Haeusler, R. Nair, and D. Kondermann. Ensemble learning for confidence measures in stereo vision. In *CVPR*, 2013. 2, 4, 7

[15] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *PAMI*, 30(2):328–341, 2008. 2, 3

[16] H. Hirschmüller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *PAMI*, 31(9):1582–1599, 2009. 3

[17] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *PAMI*, 35(2):504 – 511, 2013. 4

[18] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. *PAMI*, 34(11):2121–2133, 2012. 2, 4

[19] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *PAMI*, 20(3):226–239, 1998. 5

[20] N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic MRFs. In *CVPR*, 2007. 3

[21] D. Kong and H. Tao. A method for learning matching errors for stereo computation. In *BMVC*, 2004. 2

[22] D. Kong and H. Tao. Stereo matching via learning multiple experts behaviors. In *BMVC*, 2006. 2

[23] L. Ladicky, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *IJCV*, 100:122–133, 2012. 2

[24] M. Lew, T. Huang, and K. Wong. Learning and feature selection in stereo matching. *PAMI*, 16(9):869 –881, 1994. 2

[25] O. Mac Aodha and G. J. Brostow. Revisiting Example Dependent Cost-Sensitive Learning with Decision Trees. In *ICCV*, 2013. 2

[26] O. Mac Aodha, A. Humayun, M. Pollefeys, and G. J. Brostow. Learning a confidence measure for optical flow. *PAMI*, 35(5):1107–1120, 2012. 2, 8

[27] A. Motten, L. Claesen, and Y. Pan. Trinocular disparity processor using a hierarchic classification structure. In *IEEE/IFIP International Conference on VLSI and System-on-Chip*, 2012. 1, 2

[28] D. Pfeiffer, S. Gehrig, and N. Schneider. Exploiting the power of stereo confidences. In *CVPR*, pages 297–304, 2013. 2

[29] F. Provost and P. Domingos. Well-trained PETs: Improving probability estimation trees. Technical report, CDER Working Paper #00-04-IS, Stern School of Business, New York University, 2000. 6

[30] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, 2011. 3, 4

[31] N. Sabater, A. Almansa, and J. Morel. Meaningful matches in stereovision. *PAMI*, 34(5):930–942, 2012. 2

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

[33] R. Spangenberg, T. Langner, S. Adfeldt, and R. Rojas. Large scale semi-global matching on the cpu. In *IEEE Intelligent Vehicles Symposium*, pages 195–201, 2014. 3

[34] 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. 2, 7

[35] B. Stenger, T. Woodley, and R. Cipolla. Learning to track with multiple observers. In *CVPR*, pages 2647–2654, 2009. 2

[36] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *PAMI*, 32(5):815–830, 2010. 4

[37] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun. Continuous Markov random fields for robust stereo estimation. In *ECCV*, pages V: 45–58, 2012. 2

[38] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, pages 609–616, 2001. 6

[39] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 694–699, 2002. 2, 6

[40] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, 2015. 2

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

[42] S. Zhu, L. Zhang, and H. Jin. A locally linear regression model for boundary preserving regularization in stereo matching. In *ECCV*, 2012. 2