Confidence Estimation for Superpixel-based Stereo Matching

Rafael Gouveia[1,2] Aristotle Spyropoulos[2] Philippos Mordohai[2]
[1]Universidade Federal de Pernambuco, Brazil [2]Stevens Institute of Technology, USA
rfccg@cin.ufpe.br {ASpyropo, Philippos.Mordohai}@stevens.edu

Abstract

In this paper we propose an approach for estimating the confidence of stereo matches for superpixel-based disparity estimation. To our knowledge, this is the first such method reported in the literature. Starting from a simple superpixel stereo algorithm, we present a representative set of features that can be extracted from the disparity map and the superpixel fitting process. A random forest classifier is then trained on these features to predict whether the disparity assigned to each pixel of a test disparity map is correct or not. We perform experiments on the KITTI stereo benchmark and show that our confidence estimator is very accurate in predicting which disparities are correct and which are not. We also present a post-processing algorithm for improving the accuracy of the disparity maps that exploits the confidence estimates to reject wrong disparity values and achieves significant error reduction.

1. Introduction

Stereo matching methods based on global optimization typically outperform local alternatives in terms of accuracy. This is evident on popular benchmarks such as those hosted by Middlebury College (version 2) [33] and the Karlsruhe Institute of Technology [12]. There is still a need for local stereo algorithms, however, in applications that require real-time processing or operate on high resolution images. Superpixels are an attractive representation between the two extremes because they provide a good trade-off between processing speed and accuracy. Superpixels can serve as domains for regularization and are not heavily biased towards fronto-parallel surfaces of constant disparity. In this paper, we focus on confidence estimation for superpixel-based stereo. Our method is suitable for a fast implementation, but we do not attempt to realize such an implementation here.

While local stereo methods are in general more prone to errors compared to global methods, these errors may be easier to detect. Due to the tight coupling of data-fidelity and smoothness terms in global methods, wrong disparity assignments may be hard to diagnose. On the other hand, matching uncertainty can be assessed more easily in simple matching algorithms. Confidence estimation for stereo matching aims to assign to each pixel a value that reflects the reliability of the disparity that has been assigned to the pixel. If confidence is estimated accurately, incorrect disparities can be identified, discarded and possibly replaced by more accurate ones. These capabilities can be of critical importance in applications that require real-time depth estimation but can tolerate some holes in the depth maps, such as occupancy grid estimation for autonomous navigation or 3D scene modeling using mobile monocular or stereo cameras.

Recent research on confidence estimation [14, 34, 29] integrates multiple sources of information within a discriminative learning framework, instead of relying on a single feature for each pixel [11, 16]. As expected, taking into account multiple features leads to improved performance since different failure modes can be detected, while individual features typically respond to one failure mode. For example, errors in stereo matching due to occlusion or due to repeated structures have different characteristics. What is missing from the literature, however, is confidence estimation for superpixel-based algorithms. This is the objective of the current paper. It should be noted that we view the work of Pfeiffer et al. [30] on stixels as an approach for improving stixel fitting relying on per-pixel confidence, while the work presented here relies on superpixel fitting to improve confidence estimation. We show that significant improvement in confidence estimation accuracy can be achieved via the use of superpixel-specific features.

Processing begins by fitting superpixels to a disparity map estimated using local winner-take-all (WTA) stereo (Section 4). We, then, compute features on pixels and superpixels to form a feature vector for each pixel (Section 5). A random forest (RF) classifier is trained on feature vectors of a training set of stereo pairs to predict the confidence of pixels of the test set (Section 6). We show results on the KITTI stereo benchmark in ranking disparity assignments according to confidence in Section 7 and in improving the input disparity maps using confidence estimates to

guide post-processing in Section 8. Due to the very high accuracy of the classifier, post-processing leads to substantial reduction in the disparity error rate.

Our main contribution is a general and extensible approach to confidence estimation for superpixel-based stereo matching, as well as the superpixel-specific features. The algorithms for initial disparity estimation, superpixel fitting and post-processing are generic and can be replaced by any reasonable technique for performing these computations. We have chosen random forests for classification because they are well suited for learning in inhomogeneous feature spaces due to not requiring a metric in feature space.

2. Related Work

The literature on stereo matching is too voluminous to be reviewed here. In this section, we focus on approaches that estimate the confidence of stereo matches and refer readers to surveys [33, 8] for broader coverage of the stereo literature. The first part of this section covers related work on estimating the confidence of given disparity assignments. To the best of our knowledge, no prior work estimates the confidence of disparities computed using superpixel-based stereo.

Egnal et al. [11] published the first comprehensive study of stereo confidence comparing five measures in predicting matching errors on three "single-view" stereo pairs. Early confidence measures include the curvature of the cost curve [11], the ratio of the highest to the second highest matching score for a given pixel [23, 11, 15], interpretations of the cost or matching curve as a probability mass function [24, 32, 25], the number of inflection points of the cost curve [20] and left-right consistency. More sophisticated measures include the Distinctive Similarity Measure (DSM) [39] and the Self-Aware Matching Measure (SAMM) [27] that take into account neighbors of each pixel on its epipolar line. Hu and Mordohai [16] compared 17 confidence measures including most of the above and some that were newly introduced in that paper. A clear conclusion from these studies is that different confidence measures have different strengths and weaknesses. No single measure can diagnose all potential failure modes. Approaches that combine multiple confidence measures have also been reported in the literature [10, 15, 20], but the combination was based on hand-crafted rules. We expect that a formal learning process will lead to better results.

Machine learning techniques have been used to learn characteristics associated with high and low confidence matches. Lew et al. [21] presented an approach for a priori selecting a set of landmarks that are likely to be matched correctly. Kong and Tao [18] used non-parametric techniques to learn the probability of a potential match to belong in three categories: correct, wrong due to foreground over-extension or wrong for other reasons. Later, the same authors present an approach for selecting among a number of stereo matching algorithms [19]. Haeusler and Klette [13] considered several confidence measures, as well as the product of all measures, demonstrating good performance in sparsification, that is the removal of incorrect matches from the disparity map. Sabater et al. [31] introduced an a contrario approach for validating the correctness of stereo matches based on a robust similarity measure. A user-specified acceptable number of false matches determines the density of the final disparity map. Pfeiffer et al. [30] integrated three confidence measures into a mid-level representation (stixels) for 3D reconstruction and showed that Bayesian reasoning outperforms thresholding for sparsification.

Recent work has employed multiple confidence measures as features for discriminative learning. Motten et al. [28] presented a classifier using decision trees implemented on FPGA for selecting among multiple disparity hypotheses generated by trinocular stereo. Haeusler et al. [14] trained a random forest classifier using a number of confidence measures as features to make predictions about the correctness of the outputs of the semi-global matching algorithm. Spyropoulos et al. [34] used a similar classification approach, but also demonstrated that such a classifier can be used to select ground control points, which in turn can help improve the accuracy of the input disparity maps. Park and Yoon [29] also use a number of confidence measures as features in a random forest classifier that predicts the correctness of WTA disparities. Then, classifier predictions are used to modulate the data term of each pixel in SGM-based optimization leading to improvements in accuracy. Our approach is similar to the latter three publications. We are able to show, however, that superpixel-based features lead to much higher accuracy than those based on individual pixels.

In addition to the confidence estimation literature, superpixel or segmentation-based approaches to stereo matching are also relevant to our research. The paper of Birchfield and Tomasi [2] is a milestone for a number of reasons. The most relevant to our work is that it relaxed the fronto-parallel assumption which is typically made in stereo and proposed a practical algorithm for global optimization with plane memberships as labels. This work was extended to non-planar segments by Lin and Tomasi [22]. Segmentation-based approaches [3, 17, 37, 5] have also been very successful on the Middlebury benchmark [33]. Recently, Yamaguchi et al. [38] presented an approach that jointly segments the reference image into superpixels, estimates the disparity and occlusion label of each pixel, and also estimates optical flow if more than two images are available. Multi-view stereo is out of the scope of this paper, but it is worth pointing out recent algorithms [26, 6, 36] that have shown very competitive results.

3. Overview of the Approach

The problem addressed in this paper is confidence estimation for superpixel-based disparity estimation. Given this problem statement, we consider pixel-wise initial disparity estimation as an input to our algorithm. Section 4 briefly describes how this is accomplished using Zero-mean Normalized Cross-Correlation (NCC), but any other matching function could have been used.

Our approach requires a training set of stereo pairs with ground truth, such as those provided by the KITTI Vision Benchmark Suite [12]. The steps during the training phase are:

- Estimate initial disparity maps using a Winner-Take-All (WTA) local stereo algorithm.
- Fitting planes to the superpixels in the initial disparity maps.
- Compute a set of features for each pixel and superpixel based on information from the disparity maps and the superpixel fitting process.
- Train a classifier to predict whether the disparity assigned to a pixel is correct based on the feature vector and the ground truth disparity maps.

During testing, each stereo pair is processed separately. We again fit planes to the superpixels in the test disparity map and estimate the feature vector for each pixel and use the classifier to estimate the confidence of each disparity value. Optionally, we can sparsify the disparity map by rejecting disparities that are likely to be wrong and then replace the rejected disparities with new values that are consistent with the neighborhood of each pixel.

It is worth pointing out that this approach is applicable to any superpixel-based disparity estimation algorithm and that more sophisticated post-processing could have resulted in even larger increases in accuracy.

4. Superpixel Fitting

Processing for a stereo pair begins by computing a matching score for all possible disparities for each pixel. The steps described in this section are performed on all stereo pairs regardless of whether they are in the training or test set. We use Zero-mean Normalized Cross-Correlation (NCC) as the similarity function, but our results are applicable to any reasonable choice for such a function. The outputs of the NCC computations can be thought of as external inputs to our algorithm. We define the NCC for assigning a disparity d to a pixel (i, j) of the reference (left) image as

$$NCC(i,j,d) = \frac{\sum_{p\in W}(I_L(i_p,j_p)-\mu_{_L})(I_R(i_p-d,j_p)-\mu_{_R})}{\sigma_{_L}\sigma_{_R}}, \quad (1)$$

where I_L and I_R are the two images of the stereo pair, W is the matching window centered on (i, j), $\mu_{_L}$ and $\mu_{_R}$ are the means and $\sigma_{_L}$ and $\sigma_{_R}$ are the standard deviations of all intensities in the square window in the left and right image, respectively. We assign to each pixel the disparity with the maximum NCC value to produce a disparity map D_{NCC}. We also reverse the role of the reference and target image and compute the right-to-left disparity map, which is used for consistency tests in the remainder.

Separately, we segment the images into superpixels following the SLIC algorithm of Achanta et al. [1]. SLIC is a fast adaptation of k-means for image oversegmentation in which the pixel dissimilarity metric depends on color and image coordinate distances. The relative weights of the two distances can be adjusted to obtain more or less compact superpixels. In our experiments, we used superpixels comprising approximately 400 pixels each. Sensitivity to this parameter, however, is low.

Given D_{NCC}, we fit a 3D plane (in disparity space) to each superpixel using RANSAC. We follow the standard RANSAC procedure and generate hypotheses by picking minimal samples of three points, computing the plane equation and counting the number of inliers to the hypothesized plane. Unlike general RANSAC-based plane fitting, we assume that a single plane exists for each superpixel and that this plane should have the majority of the underlying pixels as inliers. We experimented with adapting the threshold for each superpixel according to the noise level in the underlying disparities, but a fixed threshold leads to indistinguishable accuracy and we opted for the simpler implementation. The final plane for a superpixel is estimated using least squares on all inliers of the best hypothesis. We obtain a new disparity map D_{SP} by replacing all disparities with those generated by intersecting the ray of each pixel with the estimated plane for its superpixel.

5. The Feature Vector

In this section, we present the features that are computed for each pixel in order to determine the correctness of the disparity. These features are aggregated in a feature vector per pixel and the resulting feature vectors are used to train a random forest classifier. During testing, feature vectors of pixels from the test dataset are presented to the classifier that predicts whether their disparity is correct or not. We follow the KITTI benchmark protocol throughout and consider a disparity correct if it is within three levels from the ground truth.

Our classifier relies on eight features, the first four of which are computed strictly at the pixel level and four more that leverage superpixel information. Superpixel level features are copied to the feature vectors of all pixels in a superpixel.

Matching Score (C_{NCC}) We use the NCC value of the selected disparity as the first feature based on the WTA assumption that high NCC values correspond to high likelihood of correct matching.

Left-Right Difference (C_{LRD}) This confidence measure [16] favors a large margin between the two largest NCC maxima for pixel (i, j) in the left image and also consistency of the maximum NCC scores between the left-to-right and right-to-left disparity maps.

$$C_{\text{LRD}}(i,j) = \frac{c_2(i,j) - c_1(i,j)}{|c_1(i,j) - min_{D'}\{c_{RL}(i - D(i,j), j, D')\}|} \quad (2)$$

where $c_1(i, j)$ is the maximum NCC value for pixel (i, j), $c_2(i, j)$ is the second largest NCC value, c_{RL} is the right-to-left correlation volume, $D(i, j)$ is the disparity assigned to pixel (i, j) in the NCC disparity map D_{NCC} for the left image and D' sweeps over the set of competing disparities for the correspond pixel $(i - D(i, j), j)$ in the right-to-left correlation volume. The intuition is that truly corresponding pixels should result in similar NCC values and thus a small denominator. LRD can be small for two reasons: if the margin is small, or if the margin $c_2 - c_1$ is large, but the pixel has been mismatched causing the denominator to be large.

Naive Peak Ratio (C_{PKRN}) This captures low confidence due to ambiguity by comparing the largest and second largest NCC values. c_1 and c_2 are defined as above. Note that c_2 is not required to be a local maximum [16].

$$C_{\text{PKRN}}(i,j) = \frac{c_2(i,j)}{c_1(i,j)} \quad (3)$$

Distance from the Image Border (C_{DB}) C_{DB} measures the distance in pixels from the nearest image border [34]. It is based on the assumption that pixels near the borders are likely to be outside the field of view of the other camera and that causes mismatches. We use a single feature for all four image borders following [34].

Left Right Consistency (C_{LRC}) A common technique for verifying a disparity assignment is to test whether the left-to-right and the right-to-left disparity map contain consistent disparities for the pixel in question. Instead of defining LRC as a binary feature, we set it equal to the absolute value of the difference between the left and right disparity maps. By not converting C_{LRC} to a binary feature, we allow the classifier to separate the pixels internally into subclasses of pixels that are very likely to have correct disparities, somewhat likely to have correct disparities, etc.

$$C_{\text{LRC}}(i,j) = |D_{SP}(i,j) - D_{SP,RL}(i - D(i,j), j)| \quad (4)$$

where D_{SP} is the left-to-right superpixel-based disparity map and $D_{SP,RL}$ is the right-to-left superpixel disparity map. The latter is computed by segmenting the right-to-left image and fitting planes to its disparity map. The intuition is that if disparities in an image region are reliable, the planes fitted on the left and right disparity map should assign similar disparities to corresponding pixels despite differences in the segmentation of the two images. Note that *consistency between the segmentations of the left and right image is not required since C_{LRC} is computed per pixel.*

Inlier Ratio (C_{IN}) This feature measures the fraction of inliers among the pixels of a superpixel during plane fitting. The intuition behind C_{IN} is that, if the best fitting plane to a superpixel is supported by a small fraction of the disparities generated by NCC, disparity estimation is likely noisy for that superpixel. Therefore many of its constituent pixels may have wrong disparity assignments.

Slant (C_{SL}) Binocular stereo is biased towards estimating fronto-parallel planes more precisely due to reduced perspective distortion compared to slanted planes. We capture this using the cosine of the angle between the normal of the fitted plane and the optical axis of the camera.

Neighborhood Consistency (C_{NC}) C_{NC} measures how consistent the plane of a superpixel is with those of neighboring superpixels. We first define a measure of similarity between two neighboring superpixels S_i and S_j with normals $\vec{n}_i$ and $\vec{n}_j$ and mean disparities μ_i and μ_j, respectively. We define the similarity between S_i and S_j as

$$s_{ij} = \frac{cos(|\vec{n}_i \cdot \vec{n}_j|)}{max\{|\mu_i - \mu_j|, 1\}} \quad (5)$$

The $max()$ operation in the denominator is a safeguard against division by zero or by a small number. It effectively treats all differences of mean disparity between 0 and 1 equally, since they indicate smooth continuation between the two superpixels.

Having defined a similarity measure for two superpixels, we need to combine the similarities between the current superpixel and all its neighboring superpixels to generate the neighbor consistency feature C_{NC}. This is accomplished by taking the weighted average of the pairwise similarities between the current superpixel and its neighbors. The similarities are weighted by the length of the boundary between each pair of superpixels

$$C_{NC} = \frac{\sum_{S_j \in N_i} s_{ij} b_{ij}}{B_i} \quad (6)$$

where N_i is the neighborhood of S_i, b_{ij} is the length of the boundary between S_i and S_j in pixels and B_i is the total length of S_i's boundary in pixels. We use four-connected neighborhoods throughout these computations.

6. Confidence Estimation

We have selected a random forest [7, 9] as our classifier. Random forest classifiers are ensembles of classification trees that have gained popularity due to their high accuracy and ability to generalize. They are well suited for inhomogeneous feature spaces, such as ours because, unlike a Support Vector Machine (SVM) for example, they do not require a distance metric in feature space. During training we generate decisions trees that partition the feature space separating the training data according to their labels, which are correct and incorrect disparities, in our case.

We begin by splitting the data into a training and a test set. The latter is never observed by the classifier during training. Occluded pixels are not used for training. After the feature vectors have been computed, image neighborhoods and superpixel membership information is no longer necessary. The training set can be viewed as a collection of pixels with assigned disparities, feature vectors and correct/incorrect labels coming from all stereo pairs. *The label of a pixel is correct if its estimated disparity in D_{SP} is within the specified tolerance from the ground truth*, otherwise the pixel is labeled as incorrect. During training, a new training set is created for each tree by bootstrapping from the original training set. Each node performs randomly generated tests on random subsets of the full attribute set. The attribute and threshold value that best separate the input samples are selected and the data are divided to the node's children, which are subdivided recursively.

Once the forest has been trained, the pixels of the test set with their assigned disparities and feature vectors are presented to each trained tree in the forest. The current pixel is run down each tree and decisions are made at every node based on the optimal splits computed during training. This process continues until a terminal node is reached and a decision is made about the current pixel's class label. The predictions of all trees for a pixel are averaged and the average is the confidence (prediction score) for that pixel.

7. Experimental Results

In this section, we present quantitative and qualitative results using the publicly available stereo dataset from the KITTI Vision Benchmark Suite [12]. This dataset contains stereo pairs taken by a binocular camera rig mounted on a vehicle driving in an urban environment. The resolution of the images is approximately 1241×376, and valid disparities range from 0 to 256. We use only the training set as it allows us to freely experiment and measure the accuracy of different algorithms and variations. In accordance with the protocol of the dataset, disparity assignments are considered correct if they are within three levels from the ground truth, which has been acquired by a LIDAR sensor. Approximately a third of the pixels have ground truth disparity values associated with them. We report results on non-occluded pixels throughout.

We divide the training set of the KITTI stereo benchmark so that we use the first 97 stereo pairs as training data and the following 97 stereo pairs as test data. We use the terms "training set" and "test set" to refer to this split of the benchmark throughout the paper. The other parameters of the algorithm were set to the following values for all experiments reported in this section: the window size for the initial NCC matching was 9×9; the S parameter that controls the size of the SLIC superpixels was set to 20, resulting in superpixels comprising approximately 400 pixels each; the SLIC regularization parameter was set to 1000 favoring compact superpixels; and 80 iterations of RANSAC are performed to fit a plane to each superpixel. The NCC window size is a reasonable choice, but it has not been optimized to achieve high accuracy. We used the implementation of the SLIC algorithm provided by VLFEAT [35].

After superpixel fitting, approximately 27% of the pixels are assigned disparities that differ by more than three levels from their initial values. 16.5% of pixels with available ground truth disparity change status: 10.6% that were previously wrong are assigned correct disparities, while 5.9% that were correct are assigned incorrect disparities.

We train two random forest classifiers: one using all features that will be referred to as RF-8 throughout and one using the first five features of Section 5. The latter will be referred to as RF-5 and serves as a baseline for evaluating the contribution of the paper over [14, 34, 29]. The parameters of the two random forests were chosen on a separate validation set. Both converged to the same settings: 100 trees with a minimum leaf size of 500 pixels. Both operate on the superpixel-based disparity maps for fairness, since the WTA disparity maps are noisier. We estimate feature importance by measuring the increase in prediction error on a validation set if the values of that feature were permuted [7]. For RF-8, the most important feature is LRC, followed by slant and neighborhood consistency, while the rest of the features form a third cluster in terms of importance.

Table 1 shows the confusion matrices of the two classifiers. A disparity is considered correct if the classifier's prediction for it is above 0.5. Leveraging superpixel-based features, RF-8 reaches 94.9% in classification accuracy, 97.5% in classifying pixels with correct disparity and 85% in classifying pixels with incorrect disparity. The same statistics for RF-5 are 91.9%, 93.8% and 84.7% respectively.

We also evaluate confidence methods according to their ability to rank disparity assignments from most confident to least confident using receiver operating characteristic (ROC) curves of error rate as a function of disparity map density as in [16]. We rank disparities in decreasing order of confidence to produce quasi-dense disparity maps of increasing density by selecting pixels according to rank. That

is, we produce a disparity map of 5% density and measure its error rate, then a disparity map of 10% density, etc. (The error rates are computed using only pixels that have been selected, not the total number of image pixels.) The area under the curve (AUC) quantifies the ability of a confidence measure to rank correct disparities ahead of wrong ones. Better confidence measures result in lower AUC values.

Figure 1 shows ROC curves for the three confidence measures on four stereo pairs of the test set. The three measures are: Left Right Consistency (LRC), Inlier Ratio (IN) and the output of RF-8. We use the superpixel-based disparity maps as inputs for this experiment. LRC and IN can be directly used as confidence methods by sorting pixels in decreasing order of LRC or increasing order of IN, respectively. In the case of the latter, all pixels of a superpixel are simultaneously selected for inclusion in the quasi-dense disparity maps since their confidences are equal.

Table 2 summarizes the performance of individual features and the two classifiers according to AUC on the superpixel-based disparity maps D_{SP}. The AUC values are averaged over the 97 stereo pairs of the test set. The last row of Table 2 shows the optimal AUC which can be achieved by selecting all correct disparities before starting to fill the quasi-dense disparity maps with the remaining wrong ones. As shown in [16], the optimal AUC is given by

$$AUC_{opt} = \int_{1-\varepsilon}^{1} \frac{d_m - (1-\varepsilon)}{d_m} dd_m = \varepsilon + (1-\varepsilon) ln(1-\varepsilon) \tag{7}$$

where ε is the error rate and d_m is the density of the disparity map over which we integrate. The confidence measure generated by RF-8 is clearly superior to the best performing individual features that are used as inputs to the classifier, as well as to that generated by RF-5. Individual features or a classifier without the use of superpixel-based features fall short in discriminating correct from wrong disparities. Figure 2 shows the individual AUC values for all stereo pairs obtained by individual features and the classifiers.

	RF-5		RF-8	
GT	Correct	Incorrect	Correct	Incorrect
Correct	74.30%	4.93%	77.23%	2.00%
Incorrect	3.17%	17.59%	3.12%	17.64%

Table 1. Confusion matrices for the two random forests on predicting correct and incorrect disparity assignments on all non-occluded pixels with ground truth over the 97 stereo pairs. Superpixel disparity maps were used and RF predictions were thresholded at 0.5 to make decisions. The error rate averaged over all pixels is 20.76%. Rows correspond to the ground truth labels, columns 2-3 to the predicted labels by RF-5 and columns 4-5 to the predicted labels by RF-8. Statistics here are computed at the pixel level ignoring the disparity map each pixel belongs to. *The overall accuracy of the RF-8 classifier is 94.9%.*

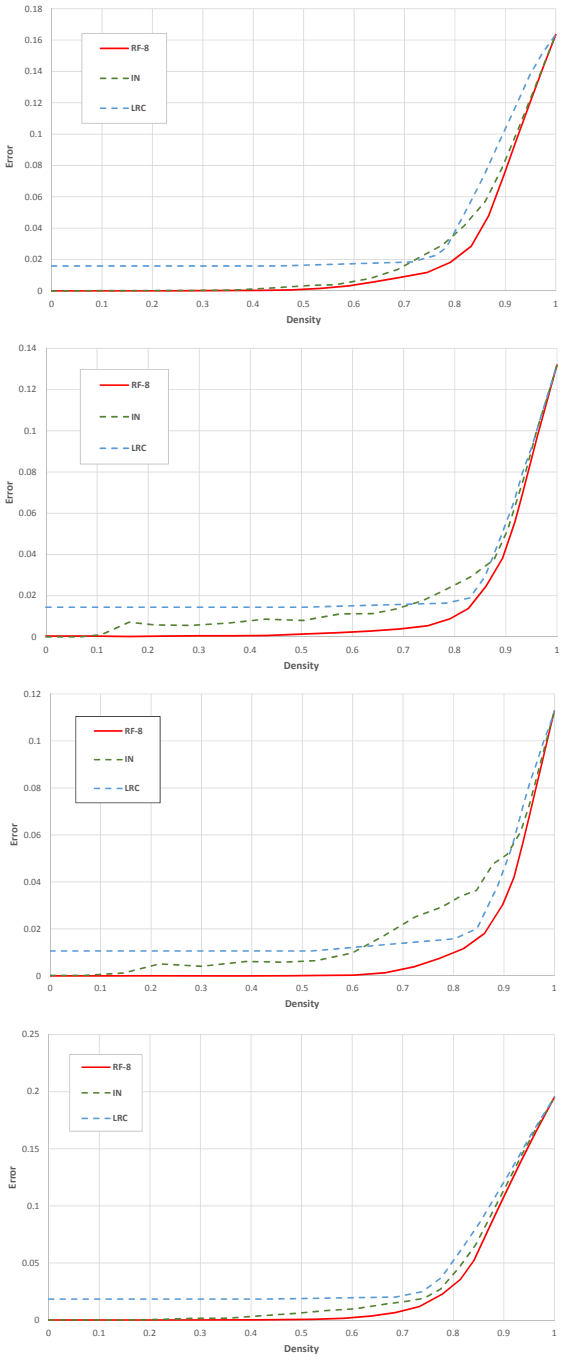


Figure 1. ROC curves for LRC (blue), IN (green) and the RF-8 classifier (red) for frames 110, 113, 134 and 136 (from top to bottom) of the test set. The x-axis is disparity map density and the y-axis is the error rate. In all cases, the RF-8 classifier achieves the minimum area under the curve (AUC). All curves reach the same point at full density since they all select all pixels, and all errors, to achieve full density.

8. Post-processing

As an application of the proposed confidence measure, we present a post-processing technique that significantly

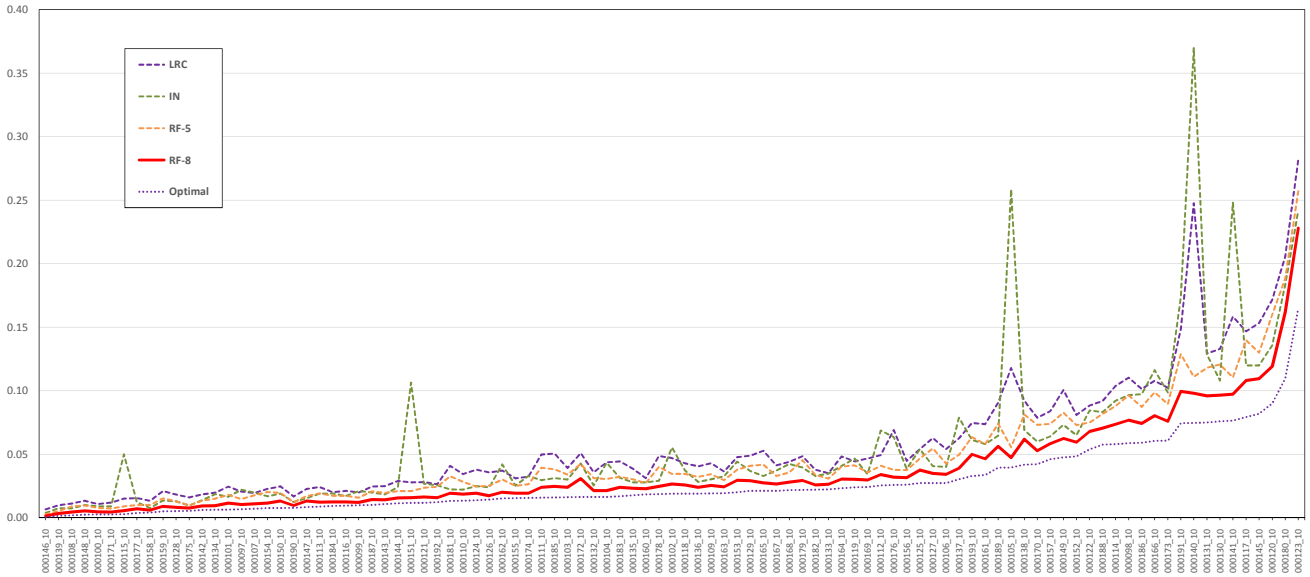


Figure 2. AUC values for all stereo pairs in the test set obtained according to LRC, IN, RF-5 and RF-8. The disparity maps have been sorted according to optimal AUC (dotted curve) to aid visualization. The RF-8 predictions (solid red curve) are more accurate than all other methods *on every single input*.

Confidence	Average AUC
LRC	0.0583
IN	0.0554
RF-5	0.0474
RF-8	0.0370
Optimal	0.0277

Table 2. Average AUC according to the most effective individual features (LRC, IN) and the two RFs. The optimal AUC is also shown.

reduces the errors in the disparity maps. Post-processing entails *sparsification* that removes wrong disparities from the input disparity map and *densification* that fills in the removed disparities, followed by filtering to remove any remaining noise.

In the sparsification step, we learn a threshold on confidence that is used to reject unreliable disparity values. As before, the test set remains sequestered during this step. The threshold is learned so that it results in the highest accuracy after densification. In other words, it is the threshold that achieves the best trade-off between error suppression and preserving enough disparity values to guide the next step. The challenge lies in that the disparity maps have different error rates and therefore each of them requires more or less aggressive post-processing. Disparity maps that are already accurate, for example, may suffer a loss in accuracy if too many of the correct disparities are rejected and replaced by different values. In this section we compare two techniques for sparsification:

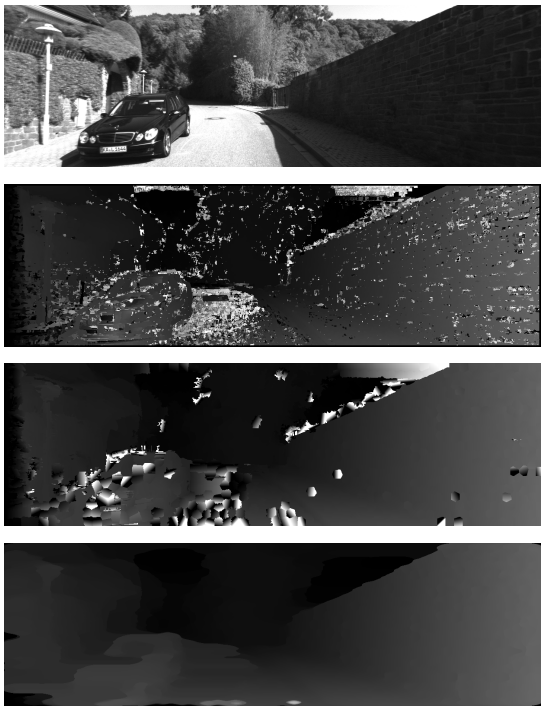
- *Fixed sparsification* in which we reject a constant fraction of the disparities in each disparity map. For the results below we reject 20% of the input disparities.
- *Adaptive sparsification* in which we learn a threshold on the RF prediction and reject all disparity values with confidence below that threshold. This results in different degrees of sparsification for each disparity map. For the results below, we use 0.67 as the threshold.

Both threshold values are learned on parts of the training set keeping the test set completely isolated.

After sparsification, we obtain a new dense disparity map by propagating disparity values as in [4]. For each pixel without a final 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 for pixels 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. We have found that 50 iterations result in lower error rates according to the KITTI evaluation protocol, at the expense of rather smooth-looking disparity maps. Table 3 shows the average error rate of the WTA disparity maps using NCC, the superpixel-based disparity maps, as well as those generated by post-processing. Following the specification of [12], a disparity is considered wrong if it is off by more than three pixels from the ground truth.

The conclusions from Table 3 are:

- The improvement due to post-processing of the superpixel-based disparity maps is significantly larger than the improvement due to fitting superpixels. We attribute this to the effectiveness of our confidence es-



(a) Input image and disparity maps for frame 126

(b) Input image and disparity maps for frame 172

Figure 3. From top to bottom for each example: left input image, WTA disparity map, superpixel disparity map, post-processed disparity map using the adaptive threshold technique.

timator to reject most of the wrong disparities, which are then replaced by more accurate ones.

- Adaptive sparsification reduces the error rate by 11.7% compared to 5.9% achieved by fixed sparsification. This is a remarkable difference.

Qualitative results are shown in Fig. 3 for two stereo pairs of the test set. The error rates for the three disparity maps for stereo pair 126 are: 21.6% for the WTA disparity map, 16.9% for the superpixel-based one and 8.4% after post-processing. The same rates for frame 172 are: 25.3%, 18.1% and 9%, respectively. The loss of sharpness at surface boundaries reduces the visual quality of the post-processed disparity maps, but also reduces their error rates.

Algorithm	Average Error Rate
WTA (NCC)	0.259
Superpixel	0.212
Post-proc. fixed	0.153
Post-proc. adaptive	0.095

Table 3. Error rates for WTA, superpixel-based and post-processed disparity maps. The second column shows the average error rate per disparity map on the non-occluded pixels of the 97 test stereo pairs.

9. Conclusions

We have presented a general approach for estimating the confidence of disparity assignments in superpixel-based stereo matching. It relies on a classifier that combines features derived from individual pixels, as well as superpixels, to predict the confidence of each pixel in a disparity map. In addition, the superpixel-based features are novel and should be considered as contributions of the paper. On a large benchmark with ground truth, we have shown that our new confidence measure is superior to baseline techniques according to the AUC criterion (Table 2).

We have also shown an application that benefits from accurate confidence estimation. By being able to identify and reject the least reliable disparities, we can post-process the disparity maps to significantly improve their accuracy.

This work, which is the first that addresses confidence estimation for stereo matching using superpixels, opens many directions for future research. We plan to investigate the use of multiple overlapping segmentations as in [5] and will attempt to fit superpixels considering both intensity and disparity as in [38].

Acknowledgements The first author was a visiting student at Stevens Institute of Technology supported by a scholarship from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil. This research has also been supported in part by the National Science Foundation awards #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. 3

[2] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, pages 489–495, 1999. 2

[3] M. Bleyer and M. Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):128–150, 2005. 2

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

[5] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR*, pages 1570–1577, 2010. 2, 8

[6] A. Bodis-Szomoru, H. Riemenschneider, and L. Van Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *CVPR*, pages 469–476, 2014. 2

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

[8] M. Brown, D. Burschka, and G. Hager. Advances in computational stereo. *PAMI*, 25(8):993–1008, 2003. 2

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

[10] C. Dima and S. Lacroix. Using multiple disparity hypotheses for improved indoor stereo. In *ICRA*, volume 4, pages 3347–3353, 2002. 2

[11] G. Egnal, M. Mintz, and R. Wildes. A stereo confidence metric using single view imagery with comparison to five alternative approaches. *Image and Vision Computing*, 22(12):943–957, 2004. 1, 2

[12] 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, 7

[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. 1, 2, 5

[15] H. Hirschmüller, P. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *IJCV*, 47(1-3):229–246, 2002. 2

[16] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. *PAMI*, 34(11):2121–2133, 2012. 1, 2, 4, 5, 6

[17] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, pages III:15–18, 2006. 2

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

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

[20] S. Lefebvre, S. Ambellouis, and F. Cabestaing. A colour correlation-based stereo matching using 1d windows. In *IEEE Conf. on Signal-Image Technologies and Internet-Based System*, pages 702–710, 2007. 2

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

[22] M. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *CVPR*, pages I: 710–717, 2003. 2

[23] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2

[24] L. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *SPIE*, 1570:187–200, 1991. 2

[25] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nistér, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *ICCV*, 2007. 2

[26] B. Micušık and J. Košecká. Multi-view superpixel stereo in man-made environments. *IJCV*, 89(1):106–119, 2010. 2

[27] P. Mordohai. The self-aware matching measure for stereo. In *ICCV*, 2009. 2

[28] 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. 2

[29] M.-G. Park and K.-J. Yoon. Leveraging stereo matching with learning-based confidence measures. In *CVPR*, pages 101–109, 2015. 1, 2, 5

[30] D. Pfeiffer, S. Gehrig, and N. Schneider. Exploiting the power of stereo confidences. In *CVPR*, pages 297–304, 2013. 1, 2

[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. Stereo matching with nonlinear diffusion. *IJCV*, 28(2):155–174, 1998. 2

[33] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002. 1, 2

[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. 1, 2, 4, 5

[35] A. Vedaldi and B. Fulkerson. VLFeat Library. http://www.vlfeat.org/, 2014. 5

[36] C. Vogel, S. Roth, and K. Schindler. View-consistent 3d scene flow estimation over multiple frames. In *ECCV*, pages 263–278, 2014. 2

[37] Z.-F. Wang and Z.-G. Zheng. A region based stereo matching algorithm using cooperative optimization. In *CVPR*, 2008. 2

[38] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, 2014. 2, 8

[39] K. Yoon and I. Kweon. Distinctive similarity measure for stereo matching under point ambiguity. *CVIU*, 112(2):173–183, 2008. 2