Feasibility Boundary in Dense and Semi-Dense Stereo Matching

Jana Kostlivá and Jan Čech and Radim Šára
Center for Machine Perception, Czech Technical University in Prague, Czech Republic
{kostliva, cechj, sara}@cmp.felk.cvut.cz

Abstract

In stereo literature, there is no standard method for evaluating algorithms for semi-dense stereo matching. Moreover, existing evaluations for dense methods require a fixed parameter setting for the tested algorithms. In this paper, we propose a method that overcomes these drawbacks and still is able to compare algorithms based on a simple numerical value, so that reporting results does not take up much space in a paper. We propose evaluation of stereo algorithms based on Receiver Operating Characteristics (ROC) which captures both errors and sparsity. By comparing ROC curves of all tested algorithms we obtain the Feasibility Boundary, the best possible performance achieved by a set of tested stereo algorithms, which allows stereo algorithm users to select the proper method and parameter setting for a required application.

1. Introduction

Dense stereopsis plays a very important role in the field of computer vision, since its results are usable to many (even very distinct) tasks. In recent years, stereo vision has been re-investigated by many researches, leading to a huge number of algorithms and the need for their comparison and evaluation immediately rose up. Several performance studies [2, 9, 5, 16, 15] showed, that there is not a single winner over the other methods, but some of the algorithms are better in some of the errors and scenes than the others, while on other errors and scenes it is vice versa. Thus, a study showing algorithm potentials and drawbacks, allowing users to select a proper method for their purposes, is desired.

There are two main properties characterising a matching algorithm: its disparity map density and accuracy measured with respect to the ground-truth matching.

One of the most important problems is the setting of algorithm parameters. Typically, each algorithm has several adjustable parameters and some of them basically determine the quality of results. Standard evaluations, such as [2, 15, 9, 5, 16], leave parameter setting on an author and keep this setting fixed over the whole evaluation dataset (which often consists of scenes with different and sometimes even very distinct character). In other words, it assumes uniform behaviour with respect to scenes (i.e. insensitivity to scene character) which not many algorithms fulfil.

In contrast, our goal is to evaluate the algorithms independently on parameter settings. To this end, we propose a kind of ROC analysis of stereo algorithm performance, where we study how the result accuracy and density change with respect to different parameter settings. An attempt towards this goal has been published in [7], where, based on errors defined in [15], ROC curves of the algorithm have been presented. Unlike in [15], we define the errors to be mutually independent and complete and base our decision about the algorithm quality on a well-defined relation "is better". Furthermore, we propose quantitative characteristics of algorithms for numerical comparison. As it is common, the test images are given and fixed. We are preparing a web-site [11] for an automatic evaluation, so that other researchers can easily use this method. Such a collection of evaluation results is also useful for users, who can simply select the most suitable algorithm for a desired application.

In the next section, we introduce our ROC analysis, in Sec. 3, the feasibility boundary. In Sec. 4, we present the experimental dataset. The algorithm evaluation itself is discussed in Sec. 5. Finally, in Sec. 6, we give conclusions.

2. ROC Analysis

To evaluate the stereo algorithm performance, we adopt the Receiver Operating Characteristics (ROC) analysis. The ROC study has been long used in signal theory to show the tradeoff between hit rates and false alarm rates [4]. The original approach has been extended to many distinct fields, such as recognition, object detection, etc [6, 1, 18, 12].

2.1. Error Definitions

We are interested in studying the matching quality, thus will measure two kinds of error statistics: how often an algorithm makes an error and how often an algorithm does not decide when it should. The **Error_rate** is defined as the percentage of incorrect (assigned) correspondences (thus, we do not count a hole as an incorrect correspondence):

$$ER = \frac{incorrect\_correspondences}{all\_pixels}. \tag{1}$$

The **Sparsity_rate** is defined as the percentage of all missing correspondences which are not ruled out[1] by any other incorrect correspondence:

$$SR = \frac{missing\_correspondences}{all\_matchable\_pixels}. \tag{2}$$

We base our evaluation on the error statistics which fulfil the following four principles, adopted from [9]:

1. *Orthogonality*: one error must not influence (or imply) any other error.
2. *Symmetry*: the errors have to be invariant to the selection of the reference image.
3. *Completeness*: the errors are well-defined in all types of scene structure.
4. *Algorithm independence*: the errors do not require completely dense results or one-to-one matchings.

Assuming rectified images, the errors are defined by means of various events in the matching table, which is the set of all possible matches P (per image row), $P = X \times X'$, where X, X' are pixels in the left, and right image rows, respectively. Every ground-truth disparity map (as well as resulting matching or disparity map of a tested algorithm) is directly transformable to matching tables, and thus this definition holds for every scene. This representation has been selected since all errors are easily visible there.

In Fig. 1, we show matching tables of two prototypes of scenes having different kinds of occlusions to demonstrate the error definitions: The table is covered by four regions [9]: the ground-truth G (blue), the jointly or mutually occluded region C (red), the occlusion boundary neighbourhood O (yellow), and the complement $T = P \setminus G \setminus C \setminus O$ (white). The size of the ground truth matching, $|G \setminus O|$, is its length in pixels in the respective image row. The size of the matching table, $D(P)$, is its diagonal length, i.e. the number of pixels in each row.

In the ROC evaluation, we use three kinds of errors (computed for each image row independently) defined as follows: Let $Q \subset P$ be a matching obtained from a tested algorithm, then

Mismatches: are the assigned correspondences with incorrect disparity. Correspondences with disparity differing from the ground-truth disparity by more than one are considered as mismatches:

$$MI = |\, Q \cap T|. \tag{3}$$

The allowed difference of ±1 in disparity is included in G.

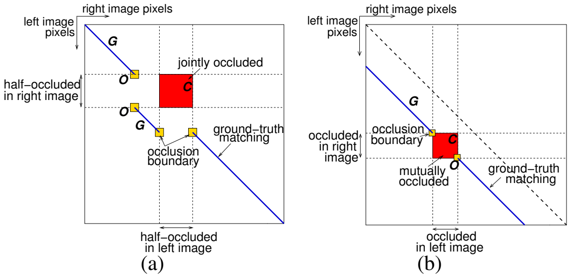


Figure 1. Matching table error definitions–two prototypes of scenes with different occlusions: half-occlusion (left), mutual-occlusion (right).

False negatives: are the unassigned correspondences at positions where the correct match exists. I.e. they are missing correspondences (holes). For the sake of error orthogonality we define it as a set of unmatched ground-truth correspondences which are not induced by a mismatch:

$$FN = |\{p \in (G \setminus Q \setminus O), X(p) \cap Q = \emptyset\}|, \tag{4}$$

where $X(p)$ represents the occlusion model: For $p = (i, j)$, $X(p) = \{(k, l) \mid (k = i) \text{ or } (l = j), (k, l) \neq (i, j)\}$.

False positives: are the assigned correspondences in occluded areas, i.e. areas where no correspondences exist, thus they are incorrect. Due to error symmetry, it is in fact only within the region C, where no ground-truth matching exists:

$$FP = |\, Q \cap C|. \tag{5}$$

Now we can define the ROC statistics precisely: The **Error_rate** of a matching is the percentage of the sum of all *false positives* and *mismatches*:

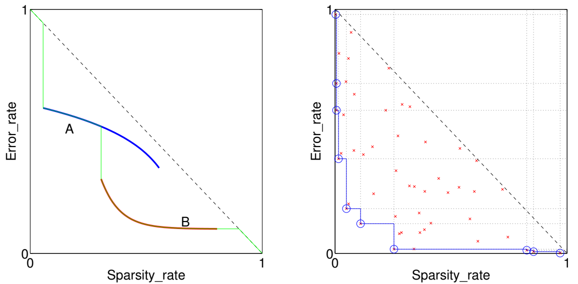
$$ER = \frac{1}{N \cdot D(P)} \sum_{r=1}^{N} \big(MI(r) + FP(r) \big), \tag{6}$$

where $r = 1, ..., N$ represents rectified image rows and $D(P)$ is the same for all the rows. The **Sparsity_rate** of a matching is defined as the percentage of all *false negatives*:

$$SR = \frac{\sum_{r=1}^{N} FN(r)}{\sum_{r=1}^{N} |\, G(r) \setminus O(r)|}. \tag{7}$$

In our analysis, we study the dependence between ER and SR. On the ROC plots, the ER is plotted on the vertical axis, while the SR on the horizontal axis. Let us now discuss the ROC space: The lower left corner, point $(0, 0)$, corresponds to an ideal algorithm, which is fully-dense and 100% correct. Point $(1, 1)$ represents the situation that in the matching, all the pairs are wrong and jointly it is completely empty, which cannot happen and thus this point is unreachable. Point $(0, 1)$ corresponds to fully-dense results which are totally incorrect, while point $(1, 0)$ to completely empty results and thus of no errors. The diagonal

[1] A correspondence is ruled out if uniqueness constraint is violated given all other correspondences.



(a) ROC space (b) ROC curve

Figure 2. (a) ROC space with two ROC curves of algorithms A and B. Dashed black line represents reachable boundary. The green curve is SFB of algorithms A and B (Sec. 3). (b) ROC curve construction from all ROC points produced by a single algorithm. Blue circled points are selected to represent the ROC curve (blue).

line $y = 1 - x$ (dashed black line in Fig. 2) represents the worst-case boundary, which we call *zero-algorithm*.

An important property of ROC plots is that they measure the matching accuracy with respect to the matching density, which are in contradiction. Hence, it gives the possibility to a user to select the most suitable algorithm: E.g. *View prediction* requires low sparsity rate, while a few local errors do not affect result quality, while *3D scene reconstruction* requires low error rate, while lower density is acceptable.

2.2. ROC curve

One parameter setting of an algorithm gives a pair (SR, ER), hence, a single point in the ROC space, a *ROC point*. A different setting gives (generally) different (SR, ER) pair. Since the parameters are given discretely, one in fact only samples the ROC space. It is the responsibility of an author to select parameter quantisation which gives results as close to the best ROC results as possible.

All output pairs under the varying parameter settings determine the ROC curve of an algorithm (two exemplary ROC curves are shown in Fig. 2(a)): we define it as the lower hull of all its ROC points.[2] First, we define relation between them: We say that ROC point $u = (SR(u), ER(u))$ is *better* than point $v = (SR(v), ER(v))$, which we write $u < v$, if it is more accurate and denser:

$$u < v \Leftrightarrow \{u \neq v \,\&\, SR(u) \leq SR(v) \,\&\, ER(u) \leq ER(v)\}. \quad (8)$$

For the ROC curve, points for which there is no better point are selected. Hence, the ROC curve is formed by the following set of points:

$$R = \{u \in P \colon \nexists v \in P,\, v < u\}, \quad (9)$$

where P is a set of all ROC points. Points which are not in R are worse (in ER or SR, or even in both statistics), and thus they are in fact redundant. The ROC curve constructed from the ROC points is shown in Fig. 2(b): red crosses represent all ROC points, those marked by blue circles have been selected to R, dotted black lines originating in each of these points show regions of ROC points dominated by the ROC curve points.

Points R defined in (9) form the ROC curve. Only in these points we know the exact position of the curve. Thus, we define the ROC curve as a curve connecting points R by a piecewise constant line, alternately with respect to SR and ER (shown as blue solid line in Fig. 2(b)). Above this curve all points are worse than those in R (i.e. it is a worst case boundary between ROC curve points). Since the ROC curve is directly determined by points R, we write it as $C(R)$.

2.3. Algorithm Characteristics

For quantitative comparison, we define two numerical characteristics, derived from algorithm's ROC curve. The definition requires the ROC curve representation as a function: only piecewise constant parts of the ROC curve with respect to one of the axes are selected for *ROC function*.

Efficiency The efficiency is an integral characteristic which expresses the algorithm qualities along the entire ROC space. The efficiency E is defined as

$$E(A) = 2 \int_0^1 \big(Z(x) - A(x)\big) dx, \quad (10)$$

where $Z(x) = 1 - x$ represents the ROC function of zero-algorithm (shown in Fig. 3 as a diagonal line). The $A(x)$ is the ROC function of the measured algorithm, Fig. 3(a). The meaning of this characteristic is the measure of improvement compared to the worst case. The range of the efficiency is $E \in [0, 1]$, where $E = 0$ is for the zero-algorithm which produces the worst possible results, and $E = 1$ is for the ideal algorithm which is error free and fully dense, i.e. has a single point $(0, 0)$ forming the ROC curve.

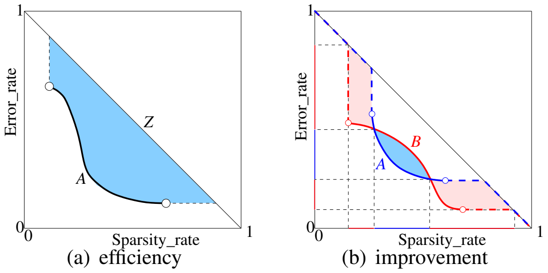
The efficiency characterises over-all behaviour of the algorithm. But, it does not mean that if the efficiency of algorithm A is higher than that of algorithm B, the algorithm A is better. The reason is that in some region of ROC space algorithm B can be better than A, i.e. the $B(x) < A(x)$ for some $x \in [0, 1]$.

Improvement The improvement is a comparative characteristic which measures how much one algorithm is better than the other in areas where it is better. We define the improvement $I(A|B)$ of algorithm A over B, as:

$$I(A|B) = 2 \int_{\{x : A(x) < B(x)\}} \big(B(x) - A(x)\big)\, dx, \quad (11)$$

where $A(x)$ is the ROC curve of the measured algorithm, $B(x)$ is the ROC curve of a reference algorithm, Fig. 3(b). The set, where the algorithm is better than the other, we call algorithm's *dominant interval*. Dominant intervals of each algorithm are marked on the axes by their respective colour

[2] More precisely, it is the lower hull of the ROC points of the algorithm unified with the ROC points for the zero-algorithm.



(a) efficiency (b) improvement

Figure 3. Definitions of: (a) $E(A)$, and (b) $I(A|B)$.

in Fig. 3(b). If the measured algorithm is nowhere better, the dominant interval is empty, and its improvement is zero. Note that $I(A|B) \in [0,1]$, $I(A|Z) = E(A)$, and in general, $I(A|B) \neq I(B|A)$.

3. Stereo Feasibility Boundary

Each algorithm produces its own ROC curve. For algorithm's developers, the ROC curve shape and its comparison with other algorithm curves is very significant and useful. However, for stereo users it is important to know what can be done by existing algorithms. This we call the *Stereo Feasibility Boundary* and define it as the ROC curve of all ROC points of all the (already examined) algorithms altogether:

$$\mathit{SFB} = \{u \in \bigcup_{i=1}^{n} R_i : \nexists\, v \in \bigcup_{i=1}^{n} R_i,\, v < u\}, \qquad (12)$$

where R_i are ROC curve points of all n examined algorithms. In Fig. 2(a), we show the $C(\mathit{SFB})$ as a green curve.

Some of the algorithms need not have any representative point in SFB. It is the situation when those algorithms are better neither in density nor in accuracy comparing to other approaches. Each of the SFB points is associated with the algorithm name, and jointly with its parameter setting as well as the corresponding disparity map.

The defined numerical characteristics of Sec. 2.3 can be evaluated also for the stereo feasibility boundary. The efficiency describes how far the nowadays stereo methods are, which will be true when state-of-the-art algorithms are tested. Therefore, we are preparing an on-line evaluation tool. Improvement is however even more interesting: An evaluated algorithm can measure its improvement over the current SFB. This is an important characteristic of the algorithm, since it measures how much the algorithm improves over the best algorithms. The dominant interval then determines where this occurs in ROC space.

3.1. Worst, Best, and Mean SFBs

The ROC curve and also Stereo Feasibility Boundary are defined over results on one scene (which guarantees a fair comparison). However, it is interesting to study the algorithm's performance, over all the scenes. To this end, we define three more SFBs:

The *Worst SFB* is defined as the worst case over SFBs of all m scenes:

$$W = \{u \in \bigcup_{s=1}^{m} \mathit{SFB}_s : \nexists\, v \in \bigcup_{s=1}^{m} \mathit{SFB}_s, u < v\}. \qquad (13)$$

The W then represents a kind of $(\min, \max)$ (*pessimistic*) strategy, where over the best for each scene (SFB) we select those minimising the risk (the worst), i.e. it is very unlikely we get worse results than these.

The *Best SFB*, on contrary, is defined as the best case over SFBs of all m scenes:

$$B = \{u \in \bigcup_{s=1}^{m} \mathit{SFB}_s : \nexists\, v \in \bigcup_{s=1}^{m} \mathit{SFB}_s, v < u\}. \qquad (14)$$

The B represents the (*optimistic*) strategy with maximal risk, i.e. it is almost sure, the results we will get will be of worse performance.

Since it is not possible to evaluate algorithms on all scenes which may exist, we define *Mean SFB* to measure algorithm's expected performance, based on the following points:

$$\bar{P}(\theta) = \sum_{s=1}^{m} w_s \cdot P_s(\theta), \qquad \sum_{s=1}^{m} w_s = 1, \qquad (15)$$

where $P_s(\theta)$ is ROC point for parameter setting θ on scene s, and w_s is the weight (probability) of scene s, giving the scene representativity. Points $\bar{P}$ are used in the same way as in (9) to obtain ROC curve $C(\bar{R}_i)$ for each algorithm i. Hence, the *Mean SFB* is:

$$M = \{u \in \bigcup_{i=1}^{n} \bar{R}_i : \nexists\, v \in \bigcup_{i=1}^{n} \bar{R}_i,\, v < u\}. \qquad (16)$$

We believe that this statistic is useful mainly for applying stereo algorithms on new unknown scenes since it gives the expectation of algorithm's behaviour.

4. Experimental Data

For our ROC analysis, we use a wide range of stereo scenes with ground-truths (shown in Fig. 4): Tsukuba, Venus, Teddy, Cones, Stripes, and Slits. Ground-truth disparity maps are colour coded: warmer the colour higher the disparity, gray is occluded, black excluded.

The first four scenes are from Middlebury dataset [15, 14] (courtesy of D. Scharstein), which is a well known dataset and thus we show only the ground-truths. To enhance the set for more complex occlusions, we add Stripes and Slits scenes [9]. Both of them are based on artificial scenes with varying texture contrast over the scene. Stripes scene consists of five thin textured stripes in front of a slanted textured plane, with half-occlusions, Fig. 1(a). Slits scene consists of two parallel textured planes, the front one contains narrow slits, in which cameras see the background plane. However, each of the cameras captures in these areas different part of the background and thus the background

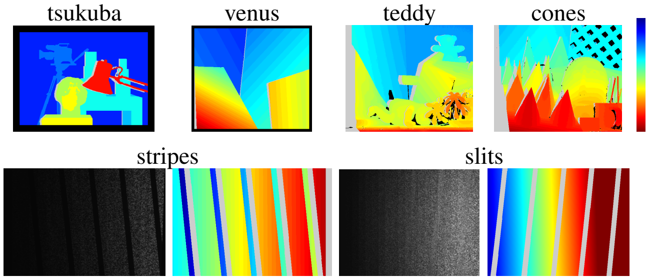


Figure 4. Ground-truths of selected testing scenes. Top row: Middlebury dataset, bottom row: Stripes and Slits scenes. Disparities are colour coded: higher the disparity, warmer the colour (as shows the rightmost bar), occlusions are gray, black is excluded.

is not binocularly visible. These regions we call *mutually occluded* (shown in Fig. 1(b), as red area).

Each scene has prescribed (known and fixed) disparity search range, which has been constructed from ground-truth disparity range $[d_1, d_2]$ as $[d_1 - \frac{d_2-d_1}{2}, d_2 + \frac{d_2-d_1}{2}]$. A wider range is used intentionally, since in many applications it need not be defined precisely or even to be known.

We have selected testing scenes to cover distinct types of configurations: well-textured together with un-textured regions, various planes, slanted as well as curved surfaces, different types of occlusions, etc. Too simple scenes (e.g. a constant disparity well-textured plane) or too tricky (unrealistic) scenes are excluded on purpose, since they are not representative to contribute to (16).

5. Evaluation/Results

For our evaluation, we have chosen five algorithms, representatives of different approaches, with available implementations. They can be roughly divided into two groups, based on prior models: (1) a strong continuity model, (2) a weak continuity model. The first group is represented by: MAP matching via graph cuts (GC) [8], MAP matching via dynamic programming (DP) [3], and MAP matching via belief propagation (BP) [17]. The second group is represented by: Confidently-Stable Matching (CSM) [13], and Stratified Dense Matching (SDM) [10]. We have selected algorithms whose implementations are publicly available, which allows experimental reproducibility.

Each of the algorithms has its own adjustable parameters, and we let an author to define himself/herself adequate range for the parameters together with the step of parameter change. For this study, the fundamental parameters for the tested algorithms have been spanned in the following intervals:

- GC: λ = 0:30:180, $penalty_0 = p$ = 0:20:140.
- BP: $opt\_smoothness = s$ = 0:25:50, $opt\_grad\_thresh = t$ = 0:2:8, $opt\_grad\_penalty = p$ = 0:4.
- CSM: α = {0:5:20,50:50:150}, β = {0,0.02,0.03,0.05,0.07,0.1,0.3,0.5,0.7}.
- SDM: α = {0:5:20,50:50:150}, β = {0,0.02,0.03,0.05,0.07,0.1,0.3,0.5,0.7}.
- DP: $penalty = p$ = {0:20:140, 150:50:1000}.

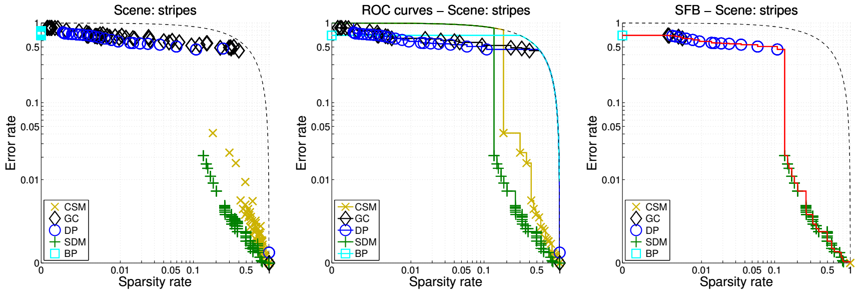
For each algorithm, the errors are computed under all parameter combinations (e.g. BP under 75 different settings).

We present the results as plots (Figs. 5-6). All the plots are shown in scales modified by $\log(x + c)$ function, where $c = 0.001$, in both axes to allow a detailed study. The diagonal line $y = 1 - x$ of reachable area is shown as a curve (dashed black) due to this modification. For visual inspection, we show selected disparity maps in Fig. 8. They correspond to ROC points which are the best of each algorithm with respect to SFB of each scene, if there are more of them, the middle one is reported.

In Fig. 5, we show results on Stripes scene. Each algorithm has a different colour and marker (for description see legend). Fig. 5(a) shows all the ROC points $(\mathit{SR}, \mathit{ER})$ of all tested algorithms resulting from all parameter settings. Algorithms with a strong continuity prior model (DP, GC, and BP) give results with higher ER, mainly due to that they fail if the prior model overweights the data (cf. Fig. 8). The density of GC and DP is varying (up to completely empty results) because both the approaches have incorporated occlusion model, unlike BP. GC and DP perform comparably (and even more, the DP is slightly better) which is mainly due to the fact that the planparalelity model of GC is violated in this scene. Algorithms with only weak model (CSM and SDM) give more accurate (about an order of magnitude) results, which are sparse however. The improved matching feature modelling in SDM decreased the ER about 2× with the same density compared to CSM.

In Fig. 5(b), we show the ROC curves. Each algorithm has its own, in BP it is only a single point since all its ROC points have the same SR. In Fig. 5(c), the SFB is shown: It shows that SDM is mostly better than CSM (as it has been visible already in previous plots) and thus CSM has only one point on the SFB, but with $\mathit{SR} = 1$. It also clearly shows that DP is better than GC in most of the range.

In Fig. 6, we show results on the other scenes (the figure is best viewed zoomed-in in the electronic version). The Slits scene shows that although the GC and DP have occlusion model incorporated, their model corresponds to half-occlusion only and thus they are not able to identify mutually occluded region (causing false positives), cf. Fig. 8. Tsukuba is the only scene, where GC reached the SFB. On this kind of scene, i.e. of narrow disparity range of only 10 pixels, nearly frontoparallel objects of almost constant disparity, GC is very good (its model holds) and thus it is for such scenes a suitable algorithm. Unlike in GC, for SDM and CSM, this scene is the most difficult one among all the tested. The last three scenes (Venus, Teddy, and Cones) show a common behaviour: GC is about 5× worse than DP on average and thus has no representative at any scene SFB.



(a) ROC points (b) ROC curves (c) SFB

Figure 5. ROC statistics on Stripes scene. The plots are best viewed in the electronic version.

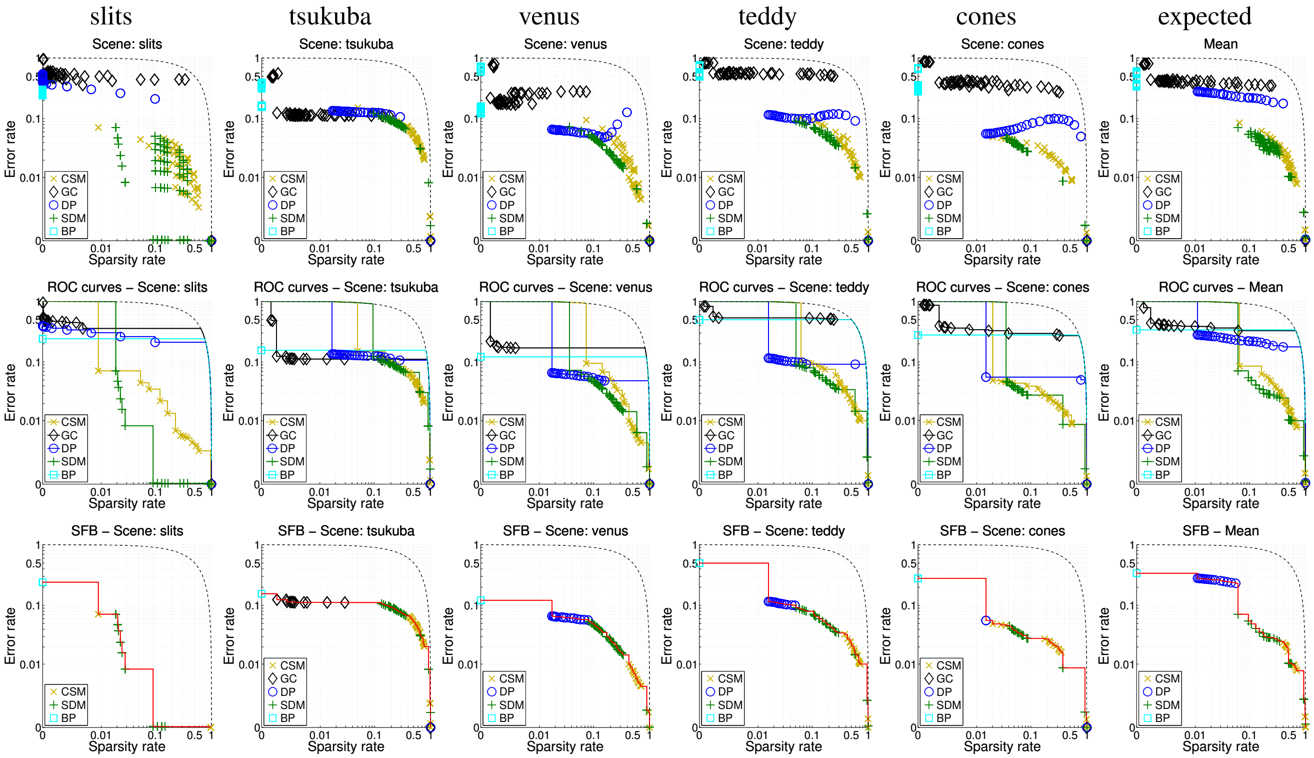


Figure 6. ROC statistics on other tested scenes, and expected.

The SFBs are of the same character consisting of BP, DP, SDM, and CSM points.

The rightmost column of Fig. 6 shows plots with expected performance, defined in (15), scene weights were set equally ($w_s = \frac{1}{m}$). They show interesting properties: ROC points of DP are monotonous and do not exhibit any worsening, unlike in individual scenes. GC has no point in M which shows its sensitivity to both scene character (and prior model) and parameter setting. SDM and CSM alternate on M which shows their comparable performance.

For easier comparison, in Fig. 7, we show SFBs of all scenes: each scene has its own colour, the algorithm markers are unchanged, and the W and B boundaries are plotted as red solid lines, while the M boundary as a red dashed line. Consequently, we can directly compare scene difficulty with respect to the tested algorithms. The worst handled scene is Stripes (for GC, DP, and BP), since thin objects at foreground together with low data regions make this scene rather difficult for them. Second scene is Tsukuba (for CSM and SDM) due to poor textured regions and repetitive patterns. However, Tsukuba is also handled the best (for GC) since it well fulfils its prior model. The Slits scene is handled the best (for SDM), even in regions of poor texture, cf. Fig. 8. To conclude: Stripes are handled the worst, Slits the best and Tsukuba the worst and simultaneously the best.

In Tab. 1, we show algorithm's efficiency $E(A)$ on all scenes: for each scene, the best is shown in bold, the worst in italic. The last column shows expected E computed for $\bar{R}$ defined in Sec. 3.1. This could be considered as overall algorithm ranking. The table confirms conclusions from the

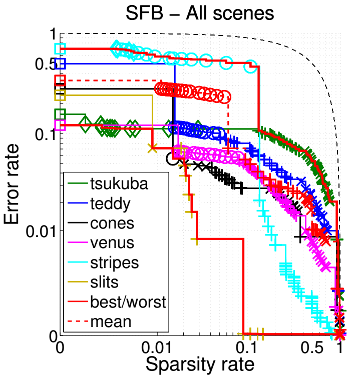


Figure 7. All scene SFBs together with B, W, M.

plots: SDM and CSM have the best efficiency on all scenes except Tsukuba. GC has the best efficiency on Tsukuba. DP is consistently better in all scenes than BP and except Tsukuba also than GC.

The improvement $I(A|B)$ we demonstrate using Mean ROC curves $C(\bar{R})$ since it gives performance expectation. Detailed results on each scene independently, together with its dominant intervals are given in [11]. Tab. 2 shows the results: all algorithms are somewhere better than the others and elsewhere worse, confirming previous conclusions that there is not a single winner.

6. Conclusions

In this paper, we have presented ROC-based evaluation of stereo algorithm performance allowing to study algorithms over a wide range of different parameter settings. ROC curve of each algorithm shows its best performance, Stereo feasibility boundary shows the best performance over the tested algorithms. Consequently, it is easy to see which algorithms are worth testing and under which parameter settings, which is useful also for stereo algorithm users.

For demonstration of our method, we have selected five distinct algorithms which have available implementations: GC, DP, BP, SDM, and CSM. The evaluation showed that if high density is required, MAP methods (GC, DP, BP) should be used; if, on contrary, low errors, methods based on stability principle (SDM, CSM) should be applied. We conclude that the main problem of MAP methods is in prior model definition: if the scene slightly violates the model, the performance is significantly decreased, thus a prior model of higher order is required (as it has been recently recognised in community).

It might seem that some of the tested algorithms (GC or DP) do not have parameters controling disparity map density directly. But these algorithms do it indirectly by assigning the label "occluded" even if there is in fact no other assigned correspondence that would imply such label at a given position. This happens when the energy for this label is lower than the energy for a disparity. Such mechanism is a valid way to generate semi-dense maps. Therefore, we consider these methods as semi-dense. If an algorithm is truly a dense method (BP in our case) then its ROC curve is induced by just a single point with the best achieved ER, which gives correct conclusions using the proposed method since the quality measures E, I works in this case as well.[3]

Table 1. Efficiency $E(A)$.

	Tsukuba	Teddy	Cones	Venus	Stripes	Slits	Expected
CSM	0.784	0.814	**0.927**	0.817	0.654	**0.962**	0.822
GC	**0.789**	*0.236*	*0.510*	*0.687*	0.243	*0.412*	0.445
DP	0.760	0.795	0.865	0.870	0.269	0.609	0.634
SDM	0.714	**0.842**	0.901	**0.896**	**0.743**	0.961	**0.841**
BP	*0.712*	0.253	0.519	0.772	*0.090*	0.575	*0.437*

Table 2. Expected Improvement $I(A|B)$.

B A	CSM	GC	DP	SDM	BP
CSM	–	0.454	0.266	0.002	0.467
GC	0.078	–	0.013	0.073	0.013
DP	0.078	0.201	–	0.073	0.211
SDM	0.022	0.470	0.280	–	0.483
BP	0.083	0.005	0.015	0.079	–

We are aware that one should not directly compare algorithms that use a different occlusion model (uniqueness, ordering). Instead, one should create a boundary for each type of model. In our study, we did mix all algorithms together, not to overload the paper. The goal of computational stereovision is to obtain algorithms that work under realistic conditions, after all, and comparing all algorithms together is important for studying where we are in stereovision.

There are open questions: We have selected a wide range of test scenes having available ground-truths. Our selection is biased towards scenes with planar objects, however, for more complex objects it is difficult to compute the ground-truth. It might be necessary to revise the selection and representativity of scenes suitable for such analysis, which we leave for open discussion. Algorithm running-time is also an important characteristic, mainly for algorithm's users. Thus, time-based evaluation would be interesting enhancement of performance evaluation.

We are preparing on-line test-bed for an automatic evaluation [11] and encourage other researchers to contribute with their algorithms to create a wider study.

Acknowledgements. This work has been supported by the Czech Academy of Sciences under project 1ET101210406, by the EC projects FP6-IST-027113 eTRIMS and MRTN-CT-2004-005439 VISIONTRAIN, and by the STINT Foundation under project Dur IG2003-2 062.

[3] In case when an author wants to see a more complex ROC curve it is not difficult to make a dense algorithm semi-dense by adding a post-processing thresholding based on image similarity and/or contrast.

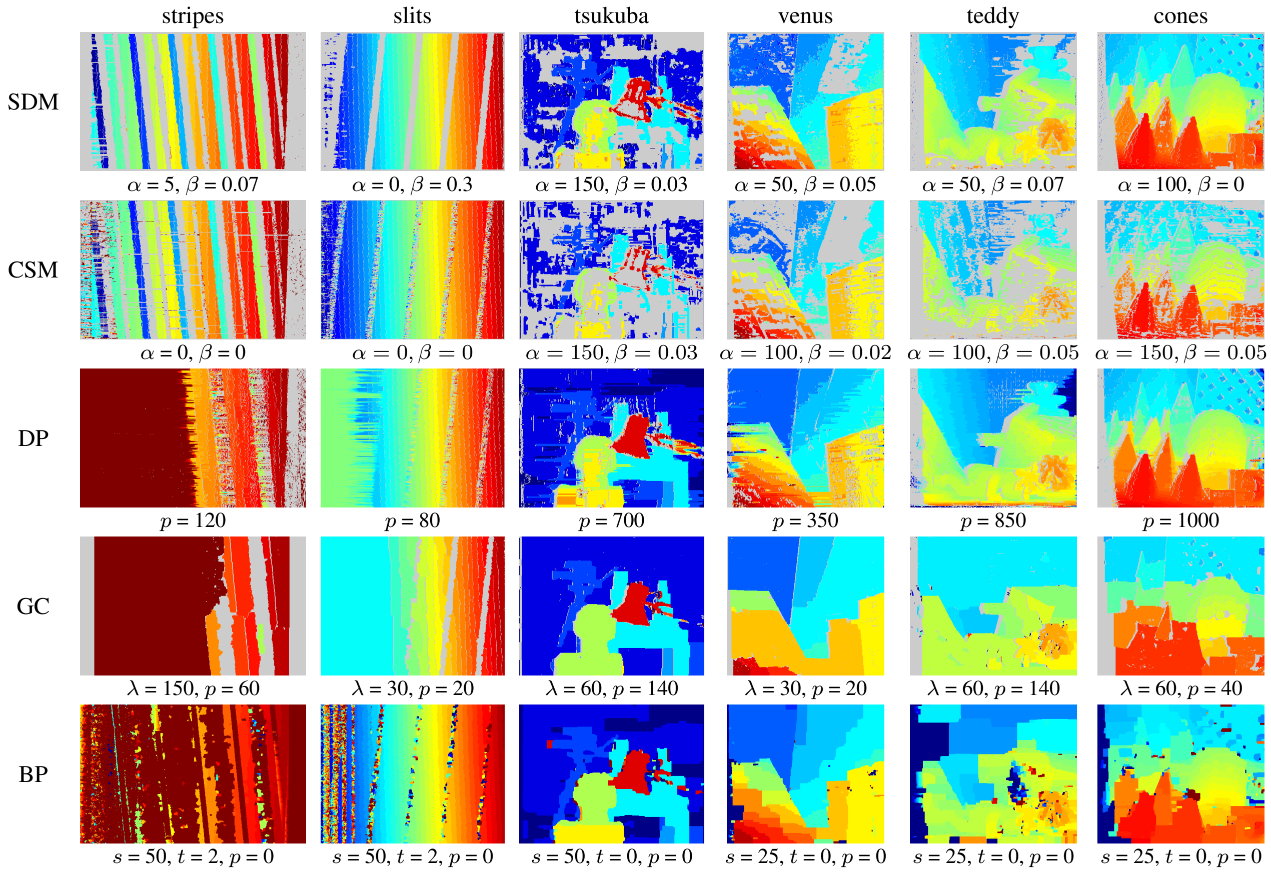


Figure 8. Disparity maps of the best ROC points with respect to each scene SFB.

References

[1] H. Blockeel and J. Struyf. Deriving biased classifiers for better ROC performance. *Informatica*, 26(1):77–84, 2002.

[2] R. C. Bolles, H. H. Baker, and M. J. Hannah. The JISCT stereo evaluation. In *DARPA Image Understanding Workshop*, pp. 263–274, 1993.

[3] I. J. Cox, S. L. Higorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *CVIU*, 63(3):542–567, 1996.

[4] J. P. Egan. *Signal Detection Theory and ROC Analysis*. Cognition and Perception. Academic Press, New York, 1975.

[5] G. Egnal and R. P. Wildes. Detecting binocular half-occlusions: Empirical comparison of five approaches. *IEEE Trans PAMI*, 24(8):1127–1133, Aug. 2002.

[6] T. Fawcett. Using rule sets to maximize ROC performance. In *ICDM*, pp. 131–138, 2001.

[7] M. Gong and Y.-H. Yang. Fast unambiguous stereo matching using reliability-based dynamic programming. *IEEE Trans PAMI*, 27(6):998–1003, 2005.

[8] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *ICCV*, pp. 508–515, 2001.

[9] J. Kostková, J. Čech, and R. Šára. Dense stereomatching algorithm performance for view prediction and structure reconstruction. In *SCIA*, pp. 101–107, 2003.

[10] J. Kostková and R. Šára. Stratified dense matching for stereopsis in complex scenes. In *BMVC*, pp. 339–348, 2003.

[11] J. Kostlivá, J. Čech, and R. Šára. ROC based evaluation of stereo algorithms. RR CTU–CMP–2007–08, Center for Machine Perception, 2007. http://cmp.felk.cvut.cz/ ˜stereo.

[12] M. Maloof. On machine learning, roc analysis, and statistical tests of significance. In *ICPR*, pp. 204–207, 2002.

[13] R. Šára. Finding the largest unambiguous component of stereo matching. In *ECCV*, pp. 900–914, 2002.

[14] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, pp. 195–202, 2003.

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

[16] R. Szeliski et al. A comparative study of energy minimization methods for Markov random fields. In *ECCV*, pp. 16–29, 2006.

[17] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *ICCV*, pp. 900–907, 2003.

[18] G. I. Webb and K. M. Ting. On the application of ROC analysis to predict classification performance under varying class distributions. *Machine Learning*, 58(1):25–32, 2005.