Dynamic Programming Stereo on Real-World Sequences

Zhifeng Liu and Reinhard Klette

The *.enpeda..* Project, The University of Auckland
Auckland, New Zealand

Abstract. This paper proposes a way to approximate ground truth for real-world stereo sequences, and applies this for evaluating the performance of different variants of dynamic programming stereo analysis. This illustrates a way of performance evaluation, also allowing to derive sequence analysis diagrams. Obtained results differ from those obtained for the discussed algorithms on smaller, or engineered test data. This also shows the value of real-world testing.

Key words: dynamic programming stereo, performance evaluation, stereo analysis, real-world sequences, driver assistance

1 Introduction

Vision-based driver assistance is one of the largest challenges in current applied computer vision. Algorithms have to process real-world stereo sequences (e.g., under all possible weather conditions) in real time. Car crash tests are performed based on very strict international standards; the same is expected soon for tests of vision-based driver assistance modules. This paper deals with real-world stereo sequences.

There are not yet many reference sequences available for comparative performance evaluation. We refer in this paper to Set 1 (provided by Daimler AG) of the *.enpeda..* sequences,[1] as described in [4]. These seven stereo sequences are taken with two Bosch (12-bit, gray-value) night vision cameras. Each sequence contains 250 or 300 frames (640×481), and features different driving environments, including highway, urban road and rural area. Camera calibration is used for geometric rectification, such that image pairs are characterized by standard epipolar geometry as specified in [3].

Intrinsic camera parameters and extrinsic calibration parameters for left and right camera (also in relation to the car) are provided. The vehicle's movement status is also given for each frame. We discuss a way to approximate partial ground truth from these sequences.

[1] `http://www.mi.auckland.ac.nz/EISATS`

2 Methodology

To evaluate the performance of a stereo algorithm, and understand how its parameters affect results, we need a quantitative way to measure the quality of calculated stereo correspondences or motion vectors.

Approximated ground truth. We assume a planar road surface for a selected sequence of stereo frames. These can be short sequences of just (say) 20 stereo frames. Here we illustrate for sequences of the given length of 220 to 300 frames. – We consider the test sequences to be ego-motion compensated [2], which means that the horizon is always parallel with the row direction in the images. We conclude that pixels on the same image row have the same depth value if a projection of the planar road surface.

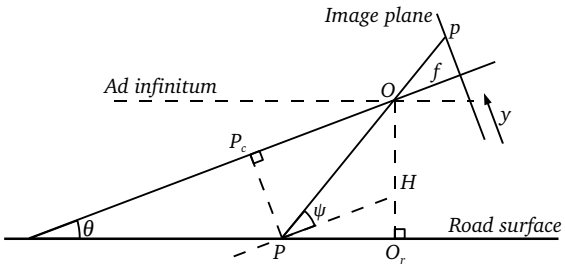

Fig. 1. Projection of a point P of the road surface.

A side-view of the camera setting is shown in Figure 1, where θ is the known tilt angle, P is a road surface point which is projected into $p = (x_p, y_p)$ on the image plane, H is the height of the camera. It follows that

$$Z = d_e(OP_c) = d_e(OP)\cos\psi = \frac{H}{\sin(\theta+\psi)}\cos\psi \tag{1}$$

According to standard stereo projection equations [3], the disparity d can be written as

$$d = \frac{b \cdot f}{Z} = \frac{b \cdot f}{\frac{H}{\sin(\theta+\psi)}\cos\psi} \tag{2}$$

where angle ψ can be calculated as follows, using focal length f and pixel coordinate y_p in the image:

$$\psi = \arctan\left(\frac{(y_p - y_0)s_y}{f}\right) \tag{3}$$

Here, y_0 is the y-coordinate of the principal point, and s_y is the pixel size in y-direction. We can also compute the y-coordinate of a line that projects to infinity

$$y_{inf} = \frac{y_0 - f \cdot \tan\theta}{s_y}$$

This is the upper limit of the road surface, and points on it should have zero disparity (if no objects block the view).

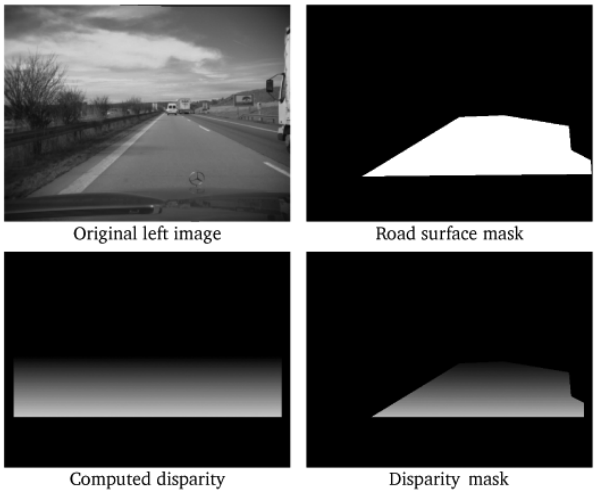

Fig. 2. Generation of a disparity mask: input image, manually generated mask, depth map of a planar road, and resulting disparity mask.

Figure 2 illustrates the process of generating an approximated disparity map on road surface areas, also using manual input for a conservative outline of the road area in a given image. In the given camera setting (of the seven sequences), there is a yaw angle (0.01 radian) which makes the cameras looking a little bit to the left. This angle can be ignored because it only defines the right camera to be about 3 mm behind the left camera.

See Figure 2 and assume a given pair of corresponding points, with disparity d. By Equation (2) we have that the tilt angle can be written as follows:

$$\theta = \arcsin\left(\frac{H\cos\psi \cdot d}{b \cdot f}\right) - \psi \tag{4}$$

where ψ is as given in Equation (3). Table 1 shows the estimated tilts for the seven sequences.

Table 1. Results of tilt angle estimation for the given seven sequences.

Sequence name	Tilt angle (radian)
1: Construction-Site	0.016
2: Save-Turn	0.013
3: Squirrel	0.021
4: Dancing-Light	0.061
5: Intern-on-Bike	0.062
6: Traffic-Light	0.069
7: Crazy-Turn	0.060

Error metrics. The general approach of stereo evaluation is to compute error statistics based on given ground truth. (Note that any ground truth comes with some measurement error; ground truth is not truth.) We use the same error measurements as on the Middlebury stereo website [6], namely the *root mean squared error* between the disparity map $d(x, y)$ and the ground truth map $d_T(x, y)$, defined as follows:

$$E_R = (\frac{1}{n} \sum |d(x, y) - d_T(x, y)|^2)^{\frac{1}{2}} \tag{5}$$

where n is the total number of pixels, and the percentage of *bad matching pixels*, defined as follows:

$$E_B = \frac{1}{n} \sum (|d(x, y) - d_T(x, y)| > \delta_d) \tag{6}$$

where δ_d is the threshold of disparity tolerance.

Tested approaches. We evaluate dynamic programming stereo, using variations of sources as available on [5]. We run a standard stereo dynamic programming (DP) approach (e.g., see [3]) on the given seven sequences; see Table 2 for evaluation results. Sequence 1 returns smallest RMS errors and bad matching percentages. In contrast, Sequence 6 returns the largest error values out of the seven sequences.

DP is then also modified by using some spatial propagation of disparities (from previous row to the current row, with a weight of 20%) or some temporal propagation of disparities (from the same row in the previous pair of frames, again with a weight of 20%). Furthermore, we run Birchfield-Tomasi (BT, designed to be an improvement of standard stereo DP).

3 Results and Discussion

The experiment on Sequence 7 is only performed on the first 220 frames, instead of the total number of 250, because the road surface is reduced to a very small area after the ego-vehicle makes a large turn to the left.

Table 2. Mean RMS error values (5) and mean bad matching percentages (6) for the standard DP algorithm.

Sequence name	Number of frames	RMS	Bad matches
1: Construction-Site	300	0.020	2.7%
2: Save-Turn	300	0.023	8.5%
3: Squirrel	300	0.023	23.1%
4: Dancing-Light	250	0.068	21.4%
5: Intern-on-Bike	250	0.064	17.5%
6: Traffic-Light	250	0.072	44.8%
7: Crazy-Turn	220	0.056	35.8%

Table 3. Mean RMS error values (5) and mean bad matching percentages (6) for DP with temporal propagation.

Sequence name	Number of frames	RMS	Bad matches
1: Construction-Site	300	0.020	1.9%
2: Save-Turn	300	0.018	3.3%
3: Squirrel	300	0.022	17.8%
4: Dancing-Light	250	0.068	19.2%
5: Intern-on-Bike	250	0.064	16.4%
6: Traffic-Light	250	0.072	45.3%
7: Crazy-Turn	220	0.054	32.9%

The DP algorithm with spatial propagation (DPs) takes 20% of the disparity value from the previous scanline into the final result. In other words, we apply

$$d'_{y,t} = (1 - \lambda_1)d_{y,t} + \lambda_1 d_{y-1,t} \quad \text{where} \quad \lambda_1 = 0.2$$

Table 3 illustrates the DP algorithm with temporal propagation (DPt), which uses

$$d'_{y,t} = (1 - \lambda_2)d_{y,t} + \lambda_2 d_{y,t-1} \quad \text{where} \quad \lambda_2 = 0.2$$

DP with temporal and spatial propagation (DPts) uses

$$d'_{y,t} = (1 - \lambda_1 - \lambda_2)d_{y,t} + \lambda_1 d_{y-1,t} + \lambda_2 d_{y,t-1}$$

where $\lambda_1 = 0.1$ and $\lambda_2 = 0.1$.

Figure 3 shows a comparison between DP and its variants, for all the frames of Sequence 1. Result show that spatial propagation causes more errors than the standard DP algorithm. Of course, the road surface is represented as a slanted plane whose disparity map changes smoothly from 0 (at infinity) to about 50.

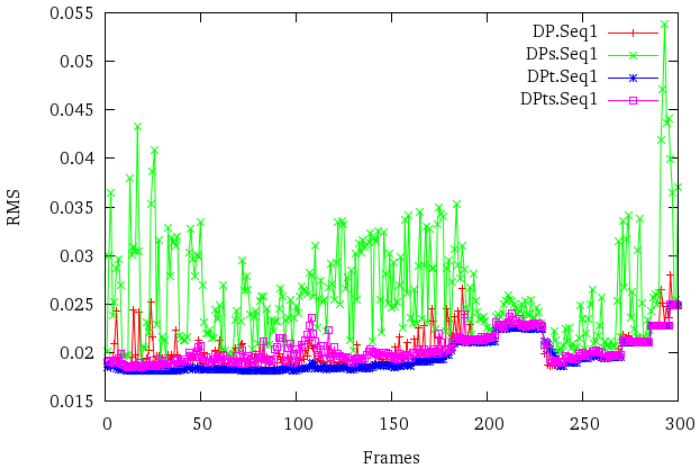

Fig. 3. Comparing RMS error (5) results between DP and its variants.

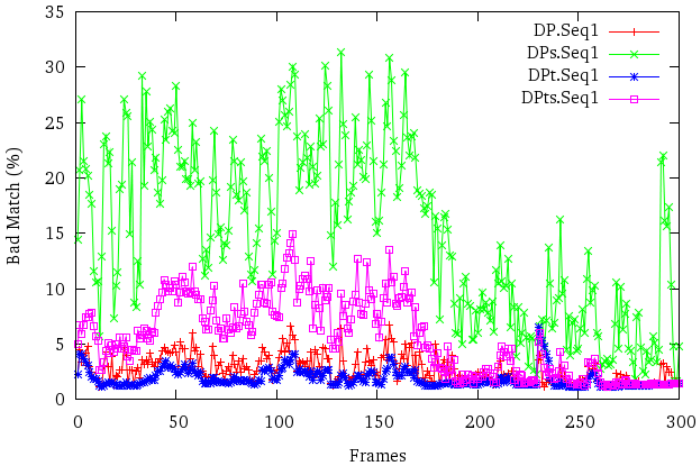

Fig. 4. Percentages of bad matches (6) for DP and its variants.

This particular geometry violates the assumption of spatial propagation. (Spatial propagation might be still of interest within object regions.)

Time propagation shows (for all seven sequences) an obvious improvement by keeping the RMS error about at the local minimum of the standard DP. Of course, driving on a plane means that disparity values should remain constant, and any deviation from this may be used to detect a change, such as a 'bumpy' road. DPts, the combined propagation method, shows a similar outcome as DP without any propagation.

A comparison with respect to the second quality metric (percentage of bad matches) is shown in Figure 4. Again, temporal propagation does have a positive effect, and spatial propagation is worsening results. (Note that this evaluation is only restricted to the road surface area.)

Now we discuss the Birchfield-Tomasi algorithm (BT). Table 4 shows evaluation results of the BT algorithm (with an occlusion penalty of 25 and a reward parameter of 5). Compared with DP techniques, the disparity maps and the qual-

Table 4. Mean RMS error values (5) and mean bad matching percentages (6) for the BT algorithm.

Sequence name	Number of frames	RMS	Bad matches
1: Construction-Site	300	0.09	61%
2: Save-Turn	300	0.11	97%
3: Squirrel	300	0.11	81%
4: Dancing-Light	250	0.13	99%
5: Intern-on-Bike	250	0.12	95%
6: Traffic-Light	250	0.14	100%
7: Crazy-Turn	220	0.11	99%

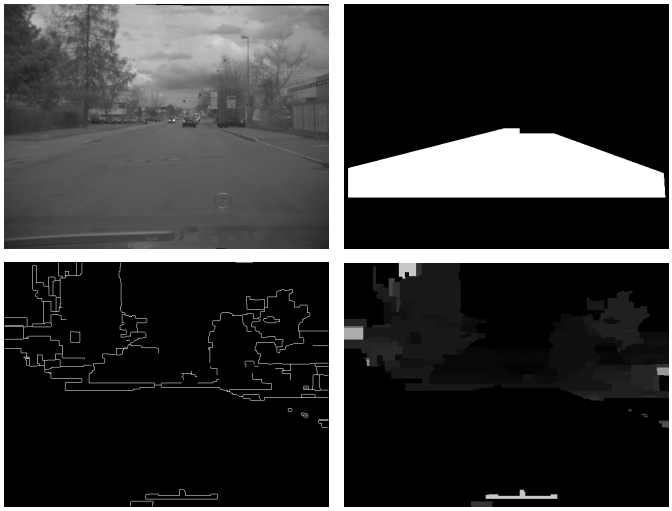

Fig. 5. The performance of the BT algorithm depends on depth discontinuities. Upper left: left image of a stereo input pair. Upper right: road mask. Lower left: depth discontinuity image. Lower right: calculated disparity map.

ity metrics indicate bad results for BT; disparity values are typically incorrect on the road surface.

This bad performance may be due to the following two reasons. First, the BT algorithm is developed on the concept of the existence of depth discontinuities.

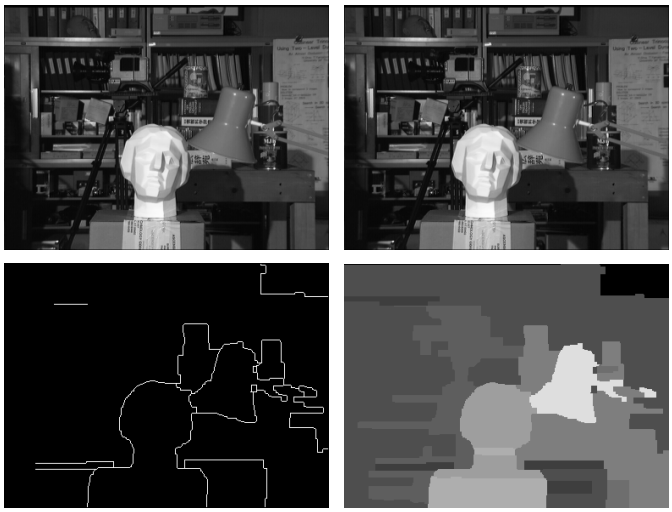

Fig. 6. Upper row: a stereo input pair of the Tsukuba sequence. Lower left: depth discontinuity image. Lower right: calculated disparity map using BT.

However, depth discontinuities may not exist in many real world situations, such as on the road. For example, Figure 5 shows that there is no edge detected close to the car.

Second, the BT algorithm uses a disparity propagation method to fill in untextured areas, both in horizontal and vertical directions. However, within the road surface area, the true disparities only change very smoothly in vertical direction.

We also run the BT algorithm (as implemented) on Middlebury stereo data, see Figure 6 for the Tsukuba test sequences. The same problem, as widely visible in the road scenes, occurs in the untextured area in the upper right corner. Except of this minor image region, BT appears here to be of advantage in general.

4 Conclusions

The difficulty for the evaluation of stereo techniques on real-world sequences is the lack of ground truth. This problem is partially solved by approximating the 3D geometry of the road.

The paper illustrated the use of these on-road estimates for evaluating the performance of variants of dynamic programming stereo on real-world sequences.

Further approximate ground truth (such as estimated poses of simple objects, such as rectangular faces in the scene) might be accumulated, to go, step by step, towards a 3D modeling of the actually recorded real scene. Of course, some objects or features are not of interest with respect to applications such as driver assistance or traffic monitoring.

The order of the algorithms' performance is clearly inconsistent to that reported on the Middlebury stereo or optical flow website. This difference shows the necessity for establishing performance evaluation methods on (various) real-world sequences ('Computer Vision beyond Middlebury' - without neglecting the very positive influence these engineered test examples had and have; but it is certainly critical if overdoing one particular way of evaluation).

References

1. *.enpeda..* Image Sequence Analysis Test Site, `http://www.citr.auckland.ac.nz/6D/`
2. Franke, U., Gehrig, S., Badino, H., Rabe, C.: Towards optimal stereo analysis of image sequences. In Proc. *Robot Vision*, LNCS 4931, pages 43–58 (2008)
3. Klette, R., Schlüns, K., Koschan, A.: *Computer Vision.* Springer, Singapore, 1998.
4. Liu, Z., Klette, R.: Performance evaluation of stereo and motion analysis on rectified image sequences. Technical report, Computer Science Department, The University of Auckland (2007).
5. Open Source Computer Vision Library, `http://www.intel.com/research/mrl/research/opencv/`
6. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, **47**:7–42, 2002.