Weighted Semi-Global Matching and Center-Symmetric Census Transform for Robust Driver Assistance

Robert Spangenberg, Tobias Langner, and Raúl Rojas

Freie Universität Berlin, Institut für Informatik,
Arnimallee 7, 14195 Berlin, Germany
robert.spangenberg@fu-berlin.de

Abstract. Automotive applications based on stereo vision require robust and fast matching algorithms, which makes semi-global matching (SGM) a popular method in this field. Typically the Census transform is used as a cost function, since it is advantageous for outdoor scenes. We propose an extension based on center-symmetric local binary patterns, which allows better efficiency and higher matching quality. Our second contribution exploits knowledge about the three-dimensional structure of the scene to selectively enforce the smoothness constraints of SGM. It is shown that information about surface normals can be easily integrated by weighing the paths according to the gradient of the disparity. The different approaches are evaluated on the KITTI benchmark, which provides real imagery with LIDAR ground truth. The results indicate improved performance compared to state-of-the-art SGM based algorithms.

Keywords: stereo vision, matching costs, census transform, local binary pattern, semi-global matching.

1 Introduction

In recent years, driver assistance systems based on vision systems have become popular. Typical outdoor scenarios contain large scene depth, many detailed structures and high dynamic illumination, and thus complicate the stereo correspondence problem. Local matching methods usually fail on ambiguous low-texture areas and sharp depth discontinuities, while global methods are too slow for practical applications. Thus, semi-global matching (SGM) has become a popular choice [1–3], providing a good compromise between complexity and robustness. Many research efforts have been put on creating efficient implementations either by taking advantage of specific hardware, e.g FPGAs [4], GPUs [5] and CPUs [6], or by reducing the search space [2]. Geiger [7] achieves the latter for a local method by building a generative model on robust features on a sub-sampled grid, which is used to guide the search in the disparity space and reduce computational complexity. Lately, Hermann [2] used the consistency of the paths to decide where to restrict the search space and how to integrate the paths in order to get more robust matches. Disadvantages of the method are the

R. Wilson et al. (Eds.): CAIP 2013, Part II, LNCS 8048, pp. 34–41, 2013.

inherent priority order for the paths and the need to serialize several parts of the algorithm, which can be run in parallel in the original SGM formulation.

We propose a weighted integration based on the region's normal of the surface each pixel belongs to. This structure could be known a priori, or as in our case is computed approximately beforehand. We tested the preprocessing step of Geiger [7] and a coarse-to-fine step using a scaled down image for SGM.

Another crucial part of stereo matching algorithms is the matching cost function. It has been shown that the Census transform [8] is favorable for outdoor environments with uncontrolled lighting [9] and/or calibration errors [10]. To improve efficiency, a sparse version of it has been proposed [11]. For face recognition, histograms of local binary patterns (LBP) have been extremely popular to provide reliable features. In this context Heikkilä introduced Center-Symmetric LBPs (CS-LBPs, [12]) to gain speed and robustness to illumination. They have been proven to be superior to LBPs and several of its variants in this field [13].

Although having a formulation in parts similar to the Census transform, LBP-based descriptors are too costly for stereo matching. Therefore, we propose to use the idea of CS-LBPs to construct a likewise Census transform. Furthermore, we investigate the effect of weighing the pixels in the distance measure.

The rest of the paper is organized as follows. In Section 2 we briefly describe the Census transform, (CS-)LBPs and introduce the proposed transforms in detail. Section 3 recaptures the basic SGM formulation and presents our modifications. We furthermore explain the creation of the surface model. The experimental results are presented in Section 4. Finally, we conclude the paper in Section 5.

2 Center-Symmetric Census Transform

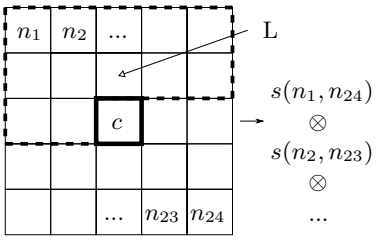
The LBP operator [14] describes each pixel using the relative intensity values of its surrounding neighbors. If the neighbor pixel is of equal or higher intensity, the value is set to one, otherwise to zero. The results for all neighbors are connected in a single number coded as a binary pattern (using the sign function $s(x) = 1$ for $x \geq 0$, $s(x) = 0$ otherwise):

$$LBP_{R,N}(x, y) = \sum_{i=0}^{N-1} s(n_i - n_c)2^i \tag{1}$$

where n_j corresponds to the intensity of a pixel j of N equally spaced pixels on a circle of radius R around (x, y) and c is the index of the center pixel. Intensities of neighbors not lying exactly on a pixel are obtained by bilinear interpolation.

Center-Symmetric LBPs [12] provide a more compact representation, comparing only center-symmetric pairs of pixels. In addition, an intensity threshold T is introduced:

$$CS\text{-}LBP_{R,N,T}(x, y) = \sum_{i=0}^{(N/2)-1} s(n_i - n_{i+N/2} - T)2^i \tag{2}$$



(a) Construction schema

1	1	2	1	1
1	1	2	1	1
2	2	c	2	2
1	1	2	1	1
1	1	2	1	1

(b) Additional pixel weights along central rows and columns

Fig. 1. Center-Symmetric Census Transform for a 5x5 patch

The Census transform [8] shares the idea of LBPs, but applies it to an image patch of $n \times m$ pixels instead of a circular region (now $s(u, v) = 0$, if $u \leq v$, $s(u, v) = 1$ otherwise):

$$CT_{m,n}(x, y) = \bigotimes_{i=-n'}^{n'} \bigotimes_{j=-m'}^{m'} s(I(x, y), I(x + i, y + j)) \tag{3}$$

with $\otimes$ being a bit-wise concatenation and $n' = \lfloor n/2 \rfloor$, $m' = \lfloor m/2 \rfloor$. The matching cost of two pixels is the Hamming distance of the results of the Census transform for those two. Typical window sizes are 3×3, 5×5 or 9×7 as their results fit into 8, 32 and 64 bit. Real-time implementations often use 5×5 giving the best compromise between speed and quality [6].

We now introduce the Center-Symmetric Census Transform (CS-CT) as

$$CS\text{-}CT_{m,n}(x, y) = \bigotimes_{(i,j) \in L} s(I(x - i, y - j), I(x + i, y + j)) \tag{4}$$

with $L = L_1 \cup L_2$, $L_1 = R_{-n',0} \times R_{-m',0} \setminus \{(0, 0)\}$, $L_2 = R_{1,n'} \times R_{-m',1}$ and $R_{a,b} = \{x \in \mathbb{Z} | a \leq x \leq b\}$. As in CS-LBPs, only center-symmetric pairs of pixels are compared, but over an image patch of $n \times m$ (Figure 1a). Like the Sparse Census transform, CS-CT only needs 31 bits to describe a patch of 9×7 pixels, but takes all pixels into account.

The gained bits may be used to encode a weighted Hamming Distance (Figure 1b) through bit duplication. This fits well to implementations using hardware bit count instructions for the Hamming Distance. Alternatively, weighting can be achieved without additional bits by using lookup tables.

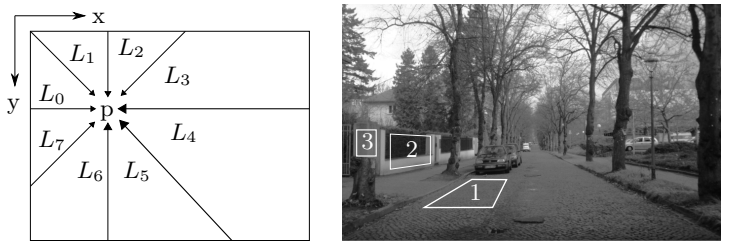
3 Weighted Semi-Global Matching

The SGM method by Hirschmüller[1] seeks to approximate a global MRF regularized cost function by following one dimensional paths L in several directions

$\mathbf{r}$ through the image. According to him it is sufficient to use 8 or 16 paths to cover the structure of the image (Figure 2a). Along each path, the minimum cost is calculated by means of dynamic programming

$$\begin{aligned} L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min(&L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d-1) + P_1 \\ &, L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d+1) + P_1, \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2) \end{aligned} \tag{5}$$

For every pixel $\mathbf{p}$ and disparity d, the cost is calculated as the sum of the matching cost $C(\mathbf{p}, d)$ and the minimum path cost to the previous pixel, with the penalties P_1 and P_2. P_1 penalizes slanted surfaces and P_2 discontinuities.



(a) 8 paths (b) Different types of surfaces

Fig. 2. Path directions in Semi-Global matching and surface dependent weights

The information from all paths is summed for all pixels and disparities giving the accumulated costs

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d). \tag{6}$$

The disparity for each pixel is now simply chosen by a winner-takes-all strategy on S. In contrast to other dynamic programming solutions, explicit occlusion handling is not possible. So a left-right consistency check is applied, either using the disparities of the right image D_R calculated by the same process or by diagonal search in S [1].

We propose a method called Weighted Semi-Global matching (wSGM) which weighs the cost of each path according to its compliance with the associated surface normal

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} W(\mathbf{r}, \mathbf{p}) L_{\mathbf{r}}(\mathbf{p}, d). \tag{7}$$

Assume we have a plane P which approximates a surface patch. Under central projection the vanishing line of P will coincide with the direction along which disparity values of points on this plane are constant. Hence disparities should be propagated preferably along paths close to this direction. We achieve this by increasing the weight of SGM paths according to the angle between the path and the vanishing line. If we imagine a road scene (Figure 2b), the pixels on the road surface (area 1) should have nearly constant disparity for the horizontal paths

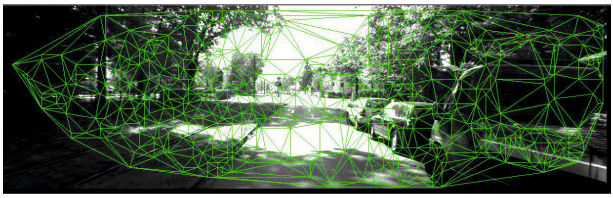


Fig. 3. Support points with Delaunay triangulation (KITTI test set frame 112)

L_0 and L_4. Thus we can safely increase the weight for these paths, whereas vertical structures parallel to the road should benefit from increased weights for the vertical paths (area 2). Frontal-parallel structures should integrate the paths evenly (area 3). However, in many applications surface normals are unknown and as we try to recover the surface by the matching, we encounter a chicken-egg problem.

We tested two different approaches to resolve this. The first one applies SGM on a scaled-down image in a coarse-to-fine fashion, while the second is derived from the method by Geiger [7]. He reduces the search space for stereo matching by creating a generative model based on support points, which are selected from a set of image points sampled on a regular grid and matched for stereo correspondence. Robustly matched points which have sufficient texture, a high uniqueness and are consistent in a left/right-check qualify as support points. They have to be similar to their surrounding support points as well to ensure they are good representatives. The generative model constitutes a Delaunay triangulated mesh with valid disparities which approximates the surface (Figure 3). The weight adaption can be performed for all image points inside the mesh.

4 Experimental Results and Discussion

To evaluate our approach quantitatively we use the KITTI stereo data set [15], providing ground-truth obtained by a laser-scanner. The scenes are rather complex, with large regions of poor contrast, lighting differences among stereo pairs and a large disparity range ($d_{max} = 255$). It is separated in a training and testing data set of around 200 images each (Figure 4). Ground truth is provided freely accessible for the training data set only, results for the testing data set are obtained by an on-line service.

We implemented our own baseline SGM algorithm using a Census window of 9×7 pixels (SGM $CT_{9,7}$). It integrates over 16 paths and uses diagonal search for the left-right check. We use a linear penalty function for the adaption of P_2 depending on the image gradients along the path as in [16] and apply the gravitational constraint [3] to disambiguate regions like sky and improve consistency in vertical regions. Parameters are $P_1 = 7, P_{2_{min}} = 17, \alpha = 0.5, \gamma = 100, P_G = 3$. Results are ahead of the OpenCV implementation (Table 1, parameters equal to KITTI website), which can be attributed to the SAD matching costs. To evaluate the benefit of the two sparse encodings, a 5×5 Census transform was tested

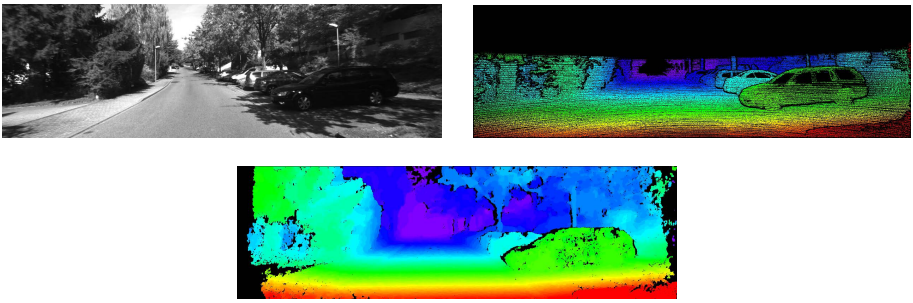


Fig. 4. KITTI training data example, left image, ground truth and baseline SGM results (top to bottom)

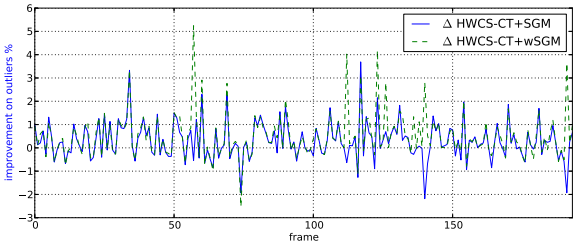
Table 1. Comparing different variants with the KITTI training data set: baseline algorithms, modifications to the Census transform and weighted SGM (Out-Noc: outliers non-occluded pixels, Out-All: outliers all pixels)

	2px		3px		
Method	**Out-Noc**	**Out-All**	**Out-Noc**	**Out-All**	**Density**
OpenCV SGM	11.40 %	12.92 %	8.39 %	9.81 %	85.50 %
SGM $CT_{5,5}$	10.90 %	12.28 %	7.34 %	8.54 %	88.74 %
SGM $CT_{9,7}$	9.39 %	10.80 %	6.23 %	7.44 %	91.53 %
SGM $Sparse\text{-}CT_{9,7}$	9.70 %	11.15 %	6.61 %	7.87 %	91.49 %
SGM $CS\text{-}CT_{9,7}$	9.59 %	11.06 %	6.51 %	7.76 %	91.97 %
SGM $WCS\text{-}CT_{9,7}$	9.12 %	10.47 %	6.03 %	7.17 %	92.12 %
SGM $HWCS\text{-}CT_{9,7}$	9.09 %	10.44 %	6.05 %	7.20 %	92.18 %
wSGM $WCS\text{-}CT_{9,7}$	8.99 %	10.35 %	5.90 %	7.04 %	91.99 %
wSGM $HWCS\text{-}CT_{9,7}$	8.89 %	10.25 %	5.89 %	7.04 %	92.17 %

as well (SGM $CT_{5,5}$, adapted parameters, tuned to be optimal). Both sparse encodings expose a gain in matching quality, CS-CT being slightly better than Sparse-CT.

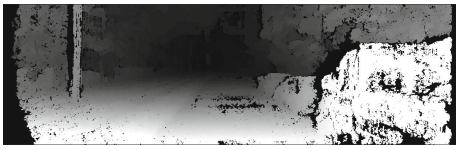
For the weighted CS-CT, we tested two variants, one with only additional horizontal weights (HWCS) and a fully weighted one (WCS), each weighted region being 3 pixels wide. Both variants perform better than the classic Census transform, providing a higher matching density and reduced outliers.

wSGM was tested with those two as well and gives additional improvements (weight factor 3 for preferred paths). Only the results using the generative model are reported. The scaled down SGM version did not provide any improvements. Limiting the weight adaption to the vertical and horizontal paths gave a slightly better result. In urban scenarios, the other surface types are not that prominent and harder to estimate with a sub-sampling approach due to their smaller size. Looking at the changes in detail (Figure 5a), one can see that the weighted Census transform reduces the number of outliers in general, whereas wSGM leads to large improvements at specific frames (Figure 5c). The poor contrast



(a) Improvements per frame to SGM $CT_{9,7}$ on KITTI training set (% outlier pixels $> 2px$)

(b) Frame 112: SGM $CT_{9,7}$



(c) Frame 112: wSGM $HWCS\text{-}CT_{9,7}$

Fig. 5. KITTI training set: changes in detail

Table 2. Evaluation on KITTI test set: error threshold 3px

Rank	Method	Out-Noc	Out-All	Avg-Noc	Avg-All	Density	Runtime
1	PCBP-SS	3.49 %	4.79 %	0.8 px	1.0 px	100.00 %	5 min
2	StereoSLIC	3.99 %	5.17 %	0.9 px	1.0 px	99.89 %	2.3 s
3	PR-Sf+E	4.09 %	4.95 %	0.9 px	1.0 px	100.00 %	200 s
4	PCBP	4.13 %	5.45 %	0.9 px	1.2 px	100.00 %	5 min
5	PR-Sceneflow	4.46 %	5.32 %	1.0 px	1.1 px	100.00 %	150 s
6	wSGM	5.03 %	6.24 %	1.3 px	1.6 px	97.03 %	6 s
7	ATGV	5.05 %	6.91 %	1.0 px	1.6 px	100.00 %	6 min
8	iSGM	5.16 %	7.19 %	1.2 px	2.1 px	94.70 %	8 s
9	AABM	5.50 %	6.60 %	1.1 px	1.3 px	100.00 %	0.43 s
10	SGM	5.83 %	7.08 %	1.2 px	1.3 px	85.80 %	3.7 s

in the example frame leads to mis-propagations on the road surface with SGM. wSGM increases the weights for the horizontal paths and is able to recover the real surface.

The results of wSGM + WCS on the KITTI test set (Table 2, with additional interpolation) show a comparable performance to iSGM and a significant improvement to the baseline SGM method. Its runtime is similar or better to the closest competitors (C++ implementation, no SSE/multi-threading). Optimizations for speed should offer gains as in [6], enabling real-time performance.

5 Conclusion

We presented a new variant of the Census transform providing higher efficiency and quality. The robustness of SGM was improved by introducing surface

normal based weights in the path integration step. For both we could show better performance on the KITTI stereo dataset. The estimation of the correct surface normals seems to be the crux of wSGM. Calculating a better approximation using stereo reconstructions from previous frames and optical flow looks promising. Further future work includes the integration of symbolic map knowledge.

References

1. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. 30(2), 328–341 (2008)
2. Hermann, S., Klette, R.: Iterative semi-global matching for robust driver assistance systems. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part III. LNCS, vol. 7726, pp. 465–478. Springer, Heidelberg (2013)
3. Gehrig, S.K., Franke, U.: Improving stereo sub-pixel accuracy for long range stereo. In: ICCV, pp. 1–7. IEEE (2007)
4. Gehrig, S.K., Eberli, F., Meyer, T.: A real-time low-power stereo vision engine using semi-global matching. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 134–143. Springer, Heidelberg (2009)
5. Ernst, I., Hirschmüller, H.: Mutual information based semi-global stereo matching on the GPU. In: Bebis, G., et al. (eds.) ISVC 2008, Part I. LNCS, vol. 5358, pp. 228–239. Springer, Heidelberg (2008)
6. Gehrig, S.K., Rabe, C.: Real-Time Semi-Global Matching on the CPU. In: CVPR Workshops, San Francisco, CA, USA, pp. 85–92 (June 2010)
7. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 25–38. Springer, Heidelberg (2011)
8. Cremers, D., Kohlberger, T., Schnörr, C.: Nonlinear shape statistics in mumford-shah based segmentation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part II. LNCS, vol. 2351, pp. 93–108. Springer, Heidelberg (2002)
9. Hirschmüller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. IEEE Trans. Pattern Anal. Mach. Intell. 31(9), 1582–1599 (2009)
10. Hirschmüller, H., Gehrig, S.K.: Stereo matching in the presence of sub-pixel calibration errors. In: CVPR, pp. 437–444. IEEE (2009)
11. Zinner, C., Humenberger, M., Ambrosch, K., Kubinger, W.: An optimized software-based implementation of a census-based stereo matching algorithm. In: Bebis, G., et al. (eds.) ISVC 2008, Part I. LNCS, vol. 5358, pp. 216–227. Springer, Heidelberg (2008)
12. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with local binary patterns. Pattern Recogn. 42(3), 425–436 (2009)
13. Meena, K., Suruliandi, A.: Local binary patterns and its variants for face recognition. In: 2011 International Conference on Recent Trends in Information Technology (ICRTIT), pp. 782–786 (June 2011)
14. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 971–987 (July)
15. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR, pp. 3354–3361. IEEE (2012)
16. Banz, C., Pirsch, P., Blume, H.: Evaluation of penalty functions for semi-global matching cost aggregation. ISPRS XXXIX-B3, 1–6 (2012)