Temporally Coherent Disparity Maps Using CRFs with Fast 4D Filtering

Siavash Arjomand Bigdeli
Institute of Computer Science
University of Bern
bigdeli@iam.unibe.ch

Gregor Budweiser
3D Impact Media AG
gregor.budweiser
@3dimpactmedia.com

Matthias Zwicker
Institute of Computer Science
University of Bern
zwicker@iam.unibe.ch

Abstract

State of the art methods for disparity estimation achieve good results for single stereo frames, but temporal coherence in stereo videos is often neglected. In this paper we present a method to compute temporally coherent disparity maps. We define an energy over whole stereo sequences and optimize their Conditional Random Field (CRF) distributions using mean-field approximation. We introduce novel terms for smoothness and consistency between the left and right views, and perform CRF optimization by fast, iterative spatio-temporal filtering with linear complexity in the total number of pixels. Our results rank among the state of the art while having significantly less flickering artifacts in stereo sequences.

1. Introduction

While some disparity estimation methods leverage information over several frames of stereo video sequences, most do not attempt to produce temporally coherent disparity maps. In applications like video production for 3D displays, however, temporally coherent disparity maps are crucial. While human observers are more forgiving about incorrect disparities, they easily notice flickering artifacts due to temporally incoherent disparity maps.

We address these challenges by proposing a technique that produces temporally coherent disparity maps over stereo videos. We formulate an energy minimization problem consisting of unary, smoothness, and consistency terms, which we solve using the mean-field approximation of a densely connected CRF. Our contributions are: 1) a new smoothness term that leverages both left and right images to distinguish between image edges due to disparity discontinuities, and edges due to surface texture; 2) a novel consistency term to obtain a joint left-and-right disparity estimation problem; 3) a temporal smoothness term to achieve temporally coherent disparity maps over stereo video sequences (Figure 1). Our algorithm has linear complexity in terms of image resolution and number of frames, and our GPU implementation requires only a few seconds per frame. Our method ranks among the state of the art in the KITTI benchmark [3].

2. Related work

Disparity estimation is mostly defined as a discrete labeling problem. Aggregation-based methods [10] share the cost of each assignment with neighboring pixels to reduce noise. They are efficient, but unable to reason about more complex assignment configurations. Optimization-based methods try to find the best assignment of disparities by minimizing an energy function. Semi Global Matching (SGM) [5] is a fast and effective approach that enforces local smoothness over many directional scan-lines using dynamic programming. While SGM is able to find a semi-global establishment of disparity labels, it is unable to capture the local structure due to the simple energy function.

On the other hand, filter-based mean-field approximation [6] supports very fast optimization over a fully-connected CRF. Many methods use a multi-scale approach to increase robustness to local minima [17]. We use the SGM method to initialize our CRF-based optimization, which further incorporates other complex terms.

Some methods use several stereo frames and attempt to ensure temporal coherence. Slanted plane StereoFlow [15] uses two consecutive frames to improve results. Vogel et al. [13] use a piece-wise rigid model to include consistencies in the temporal dimension. Unlike these methods we do not enforce segmentation nor local planarity on our disparity maps. In addition, our method has linear complexity with respect to the number of frames, which allows us to compute the disparity maps of the whole sequence in a single optimization.

Disparity flicker artifacts have been previously addressed [11, 9]. Richardt et al. [11] assumed that the pixel's disparity persist in time and aggregated the costs between temporally consecutive pixels. Min et al. [9] filtered noisy disparity maps between different frames. In addition to end-to-end disparity error, we propose a quantitative measure to better evaluate the flicker artifacts in disparity sequences and compare with previous works.

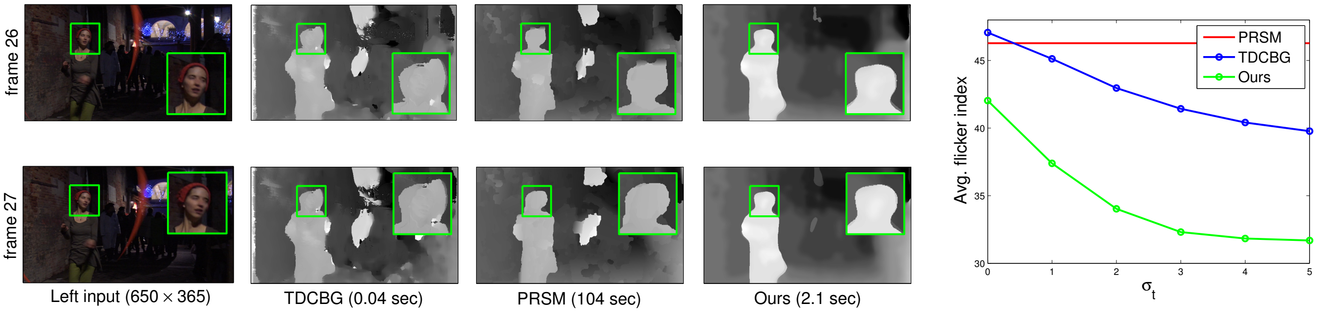

Figure 1. Our optimization includes the temporal dimension to achieve temporally coherent disparity maps in linear time. Here we compare disparity maps from TDCBG [11] using 8 frames, PRSM [13] using three consecutive frames, and our method using 21 frames. On the right we show the average disparity flicker index in this sequence. Our algorithm and TDCBG [11] allow controlling temporal smoothness using a temporal support parameter σ_t. Sequence courtesy of MEDIA LEADER Srl (www.medialeadersrl.com).

3. Proposed Method

We first define our energy terms, fast spatio-temporal energy minimization, initialization, and post processing, followed by a description of our GPU implementation.

We define random variables x_i^L for the disparity values of pixels i in the disparity field X^L of the left image, and similarly x_i^R in X^R for the right image. Our joint energy function over X^L and X^R includes unary (per-pixel), smoothness, and consistency terms. We omit the left and right superscripts unless necessary.

Unary Term. We denote the cost of assigning disparity d to pixel i in the left image L by the unary term $\phi_u^L(x_i = d)$. We compute this term based on edge differences and Census transform distances similar to Yamaguchi et al. [15].

Disparity-Dependent Smoothness Term. The goal of the smoothness term is to encourage pairs of pixels that are close in some sense (defined more precisely below), to get similar disparity assignments. We define the smoothness term $\phi_s^L(x_i = d_i, x_j = d_j)$ for a pair of assignments $x_i = d_i$ and $x_j = d_j$ in the left image as a function of both the pixel locations i, j and the disparity assignments d_i, d_j (similarly for the right image). We express this term as a sum of weights $W^L(P)$ over all paths P that connect the points $\langle i, d_i\rangle$ and $\langle j, d_j\rangle$ in the joint pixel-disparity space,

$$\phi_s^L(x_i = d_i, x_j = d_j) = -\left(\sum_{P \in \mathcal{P}(i,d_i,j,d_j)} W^L(P)\right),$$

where $\mathcal{P}(i, d_i, j, d_j)$ is the set of all paths between $\langle i, d_i\rangle$ and $\langle j, d_j\rangle$ in the joint space of pixel locations and disparity hypotheses, and each path $P = \{\langle k, d\rangle\}$ is a sequence of (4-connected) pixels k paired with a disparity hypothesis d. We define the weight kernel W based on three length functions of the path, its length $l_s(P)$ in the image, its length $l_d(P)$ in the disparity label space, and a length δ^L (discussed below) that takes into account potential disparity discontinuities along the path. Specifically, the weight kernel is

$$W^L(P) = \exp\left\{-\left\|\frac{\delta^L(P)}{\sigma_r} + \frac{l_s(P)}{\sigma_s} + \frac{l_d(P)}{\sigma_d}\right\|_2^2\right\},$$

where σ_r, σ_s, and σ_d control the kernel support for the three length terms. Applying a Gaussian to the weighted sum of the three distances ensures that $W^L(P)$ decreases when the two pixels are separated by a large distance, and it increases when they are close. This choice of weight will later allow us to efficiently compute the smoothness energy.

The key ingredient in the definition of $W^L(P)$ is the length $\delta^L(P)$, which we design to become large when the path crosses depth discontinuities. Crucially, we consider color information from both (left and right) views to compute the path length $\delta^L(P)$ such that it depends on the disparities along the path P. For each disparity on the path, we compute a pixel-wise difference of the two views where one is shifted by that disparity. At pixels where the disparity happens to be the correct one, this will cancel image edges due to surface textures, indicating that these edges are not disparity discontinuities. If the disparity is wrong, image edges typically do not cancel. We use this intuition to define a disparity discontinuity indicator for pixel k and disparity d as $\min(|L_k - L_{k-1}|, |L_k - R_{k+d}|)$, where taking the minimum makes sure we do not introduce any spurious discontinuities. The path length $\delta^L(P)$ is now simply the sum of these disparity discontinuity indicators along the path,

$$\delta^L(P) = \sum_{\langle k,d\rangle \in P} \min(|L_k - L_{k-1}|, |L_k - R_{k+d}|),$$

where L and R denote the left and right color images. This distance will be small if the pixel colors along the path have correspondences in the other image under their disparities, even if the image itself has large color dissimilarities along that path.

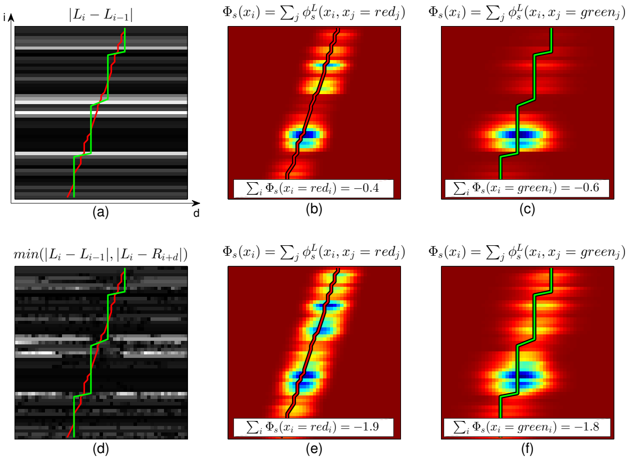

Figure 2. Visualization of the smoothness energy of a slice of the joint disparity-pixel space. (a) shows discontinuities given by pixel differences and (d) our proposed indicator δ^L function per-pixel. (a,d) show ground truth disparities in red, and some estimated disparities consisting of fronto-parallel segments in green. The smoothness energy for the red and green disparity assignments are shown using the conventional (b, c) and our approach (e, f).

We visualize our approach in Figure 2. We show slices of the joint disparity-pixel space (d, i), where disparities d are along the horizontal axis, and the vertical axis corresponds to one vertical column of pixels i. The data is from a continuous, slanted surface patch that is highly textured (ground region in Figure 3, top left). Figure 2a shows conventional disparity discontinuity indicators given by pixel differences $|L_i - L_{i-1}|$, and Figure 2d are our proposed indicators $\min(|L_i - L_{i-1}|, |L_i - R_{i+d}|)$. Figure 2a,d show the ground truth disparities in red, and some estimated disparities consisting of fronto-parallel segments in green. In Figures 2b,c,e,f we visualize the smoothness energy for the red and green disparity assignments using the conventional and our approach. That is, each point (d, i) in these figures shows the sum $\sum_j \phi_s^L(x_i = d, x_j = \Delta_j)$ where the Δ contain either the ground truth (red) or estimated (green) disparities. We also indicate the total smoothness energy $\sum_{i,j} \phi_s^L(x_i = \Delta_i, x_j = \Delta_j)$. This shows that in the conventional approach some pixels have high smoothness energies even with the ground truth disparity assignment, and the total smoothness energy of the piecewise fronto-parallel disparities (green) is actually lower than the ground truth here. With our approach, we obtain low smoothness energies at all pixels, and the ground truth has lower energy than the piecewise fronto-parallel assignments.

Higher Order Local Consistency Term. Each disparity assignment indicates that the corresponding pixel appears with a shift (disparity) in the other image, therefore we expect that the disparity in the other view would agree with this assignment. We design the consistency energy to be low if the disparity assignments in two corresponding pixels in the left and right image agree. As a key idea, we compute this term over pixel neighborhoods, instead of individual pixels, to be more robust to per-pixel errors. We first introduce a binary consistency factor $\nu = [|x_j^L - x_{j+x_j^L}^R| \leq 1]$, which is one when two corresponding pixels x_j^L and $x_{j+x_j^L}^R$ (according to the disparity assignment in the left image) agree on their disparities up to a threshold of one disparity level, and zero otherwise. We allow for a difference of one disparity level to compensate for sub-pixel disparities and self occlusions. We now define the consistency energy as

$$\phi_c^L(x_i^L = d_i, x_j^L = d_j) = -\left(\sum_{P \in \mathcal{P}(i, d_i, j, d_j)} W^L(P) \right) \nu,$$

where we sum over all paths between joint pixel-disparity assignments x_i^L and x_j^L and use the same path weight $W^L(P)$ as for the smoothness term. Intuitively, given an assignment x_i^L, our consistency energy is low if many assignments x_j^L that are close to x_i^L in the left image, have consistent assignments $x_{j+x_j^L}^R$ in the right image. Since we cannot confirm consistency in the case of occlusions, we ignore them here and treat them later when finalizing the disparity map.

Temporal Extension. A main advantage of our filter-based CRF optimization is that we can easily extend it to the temporal domain, and simultaneously optimize disparity assignments over all frames of a stereo video sequence. We define the smoothness and consistency energies (ϕ_c , ϕ_s) as before, but now with weight kernels W over paths in the joint spatio-temporal and disparity domain,

$$W^L(P) = \exp\left\{ - \left\| \frac{\delta^L(P)}{\sigma_r} + \frac{l_s(P)}{\sigma_s} + \frac{l_t(P)}{\sigma_t} + \frac{l_d(P)}{\sigma_d} \right\|_2^2 \right\},$$

where $l_t(P)$ is the length of the path in time, and σ_t determines the kernel width along time. Our assumption here is that the disparities persist over a short time defined by σ_t. As a key idea, we define the temporal dimension by following flow vectors of a precomputed flow field over the video sequence. Specifically we use the flow by Lang et al. [7], and refer the reader to their paper for more details.

Energy Minimization via Mean-Field Approximation. We define the global energy function E as a sum of the unary, smoothness, and consistency terms, all evaluated on

Algorithm 1 Optimization of the left and right disparity maps using mean-field

```
initialize $Q^L, Q^R$ with SGM
**loop** $\#iterations$
    1. $\tilde{Q}_i(d) \leftarrow \lambda Q_i^L(d) +$
       $\gamma \sum_{l, d-1 \leq l \leq d+1} Q_i^L(d) Q_{i+d}^R(l)$
    2. $\hat{Q}_i(d) \leftarrow \sum_{j,l} [- \sum_{P \in \mathcal{P}_{(i,d_i,j,d_j)}} W^L(P) \tilde{Q}_j(l)]$
    3. $Q_i(d) \leftarrow \exp \left\{ -\phi_u^L(x_i = d) - \hat{Q}_i(d) \right\}$
    4. $Q_i^L(d) \leftarrow Q_i(d) / \sum_l Q_i(l)$
    5. switch L and R
**end loop**
```

both left and right images,

$$\begin{aligned} E(X^L, X^R | L, R) = & \sum_i \left\{ \phi_u^L(x_i) + \phi_u^R(x_i) \right\} \\ & + \lambda \sum_{i,j} \left\{ \phi_s^L(x_i, x_j) + \phi_s^R(x_i, x_j) \right\} \\ & + \gamma \sum_{i,j} \left\{ \phi_c^L(x_i, x_j) + \phi_c^R(x_i, x_j) \right\}, \end{aligned}$$

with parameters λ and γ to control the influence of the smoothness and consistency terms relative to the unary term.

We minimize the energy function by following the filter-based mean-field approximation [6] (Algorithm 1). In each step we update the probability $Q_i^L(x_i = d)$ of assigning disparity d to variable x_i. We compute the per-pixel consistencies by multiplying the two probabilities and adding to the current distribution values (line 1). Next we compute the expected value of the smoothness and consistency terms (line 2) using a single, fast filtering operation over the accumulated values $\tilde{Q}_i$. A single filtering step is possible since we have the same weights W defined in ϕ_s and ϕ_c. The iteration ends by completing the update (lines 3, 4) and switching the target distribution (line 5).

We compute the path weights W efficiently using the Domain Transform Filter [2]. We use interpolated convolution by iteratively applying a moving sum (box filter) in the transformed domain. The joint image and disparity space leads to 3D filtering, and our temporal extension to 4D filtering over two spatial, the temporal, and the disparity dimensions in line 2 of Algorithm 1. In the temporal dimension we filter along the precomputed flow vectors similar as Lang et al. [7]. We obtained our best results by iterating over passes along spatio-temporal directions and filter in the disparity domain at the end. We refer to the original publication [2] for more details about filtering.

Initialization. For initializing Algorithm 1 we leverage semi-global matching (SGM) [5] with penalties $P_1 = 4$, $P_2 = 64$ in four directions. Instead of the MAP results of SGM, we rather use the obtained energies to initialize our distribution $Q_i(d)$. For a better initialization, we run the first two iterations of the optimization using a large kernel support ($\sigma_s = 7$, $\sigma_r = 100$, $\sigma_d = 2$).

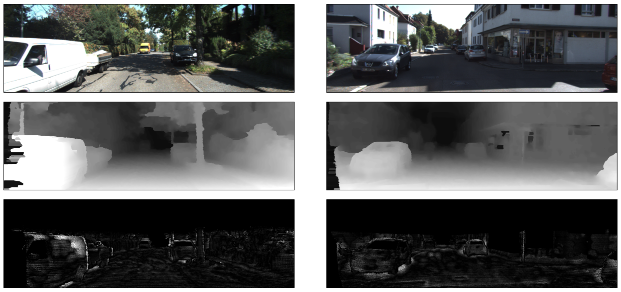

Figure 3. Examples results from the KITTI dataset. First row left image, middle row our final disparity and last row shows the errors clamped to 5.

Final Disparity Map. We compute final disparities by finding the one with the minimum energy $-\log(Q_i(d))$ from Algorithm 1. For accuracy below the level of the disparity discretization we fit a quadratic to the three disparity costs centered at the minimum. We remove spikes by applying a 5×5 median filter. We fill occluded regions by checking for left-right consistency to find pixels with disparity differences higher than a threshold, and replacing disparities marked as occluded with the last non-occluded disparity in the left direction for the left view (similarly for the right view).

Implementation. The CPU version of the proposed pipeline supports 256 or more disparity hypotheses. We also implemented a GPU version for the whole pipeline that takes advantage of parallelism in the optimization at the pixel level. We ran our experiments on an Nvidia Titan Black graphics card with 6GB memory on board. We allocate memory for a batch of left and right images, including the disparity hypothesis layers requiring $2 \times Width \times Height \times Frames \times Disparities$ floating point values. Because of the limited GPU memory we are currently restricted to batches of 14 frames at a resolution of 960×540 and 32 disparity layers. Note that we evaluate the unary term at a finer discretization of disparity steps, typically at one pixel steps. We then store the minimum for each of the 32 layers. At the end of the optimization the disparity is computed and finalized as described above, and by fitting the quadratic to the 32 layers we achieve finer levels of disparity. After the disparities of a batch of frames are computed, we move forward by seven frames and compute the disparities for the next batch. We finally interpolate the disparity values of the overlapping frames in consecutive batches for smoother transitions.

Method	% >3px	% >4px	% >5px	Time
Displets [4]	2.47	1.94	1.67	265 s
MC-CNN [16]	2.61	2.04	1.75	100 s
PRSM [13] *	2.78	2.15	1.74	300 s
SPS-StFl [15] *	2.83	2.24	1.90	35 s
VC-SF [12] *	3.05	2.35	1.92	300 s
OSF [8] *	3.28	2.59	2.16	50 min
CoR [1]	3.30	2.59	2.16	6 s
Ours	**3.32**	**2.45**	**1.96**	**60** s
SPS-St [15]	3.39	2.72	2.33	2 s
PCBP-SS [14]	3.40	2.62	2.18	5 min

Prior knowledge Planarity *: Flow

Table 1. The top 10 methods in KITTI benchmark.

4. Results and Conclusions

KITTI Stereo Evaluation. We tested our CPU implementation without the temporal extension on the KITTI [3] dataset. We fixed parameters $\sigma_s = 4$, $\sigma_r = 6$, $\sigma_d = 4$, $\lambda = 10^9$, $\gamma = 50\lambda$, which we found by exhaustive search, and observed convergence after four iterations. Figure 3 illustrates our qualitative results from two scenes of the KITTI training dataset. Table 1 summarizes the quantitative performance of our method on the KITTI test dataset. Our method obtains an averge error of 3.32% for error threshold 3 and we currently rank number 8 on the list. Our CPU implementation compares to the rest in simplicity and scalability, and still obtains state of the art results.

Stereo sequences. To measure the temporal coherence we compared the flicker index (IESNA standard) of the final disparity maps. This index is computed in a temporal window of five frames as the ratio of the time-averaged disparities and the disparities above that average, which indicates how much disparities deviate from their average value in a temporal window. In Figure 1 we compare the average flicker index of our GPU implementation with Richardt et al. [11] and Vogel et al. [13]. The plot on the right shows that we can significantly reduce the flicker index by enlarging the temporal smoothness kernel σ_t. In Table 2 we report the average computation times and flicker indices over five video sequences with resolutions from 417×360 to 960×540. Our GPU implementation requires less than three seconds per frame, and with $\sigma_t = 5$ it produces significantly less temporal artifacts.

Method	Time	Flicker
SGM	1.89s	39.48
SPSS-St [15]	1.62s	47.95
PRSM [13]	130.24s	45.98
TDCBG [11]	0.06s	35.21
Ours	**2.57s**	**25.44**

Table 2. Flicker index.

Conclusions. We have presented a robust method to compute disparity maps of stereo sequences in a single optimization. The optimization is solved efficiently using 4D filtering in pixel-disparity space. The proposed method ranks amongst the state of the art in challenging tests (KITTI) and produces less flicker artifacts in stereo videos.

Acknowledgments. This research was supported by the Swiss Commission for Technology and Innovation (CTI) under project nr. 15592.1 PFES-ES.

References

[1] A. Chakrabarti, Y. Xiong, S. J. Gortler, and T. Zickler. Low-level vision by consensus in a spatial hierarchy of regions. *CoRR*, abs/1411.4894, 2014. 5

[2] E. S. Gastal and M. M. Oliveira. Domain transform for edge-aware image and video processing. In *ACM Transactions on Graphics (TOG)*, volume 30, page 69. ACM, 2011. 4

[3] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The Kitti vision benchmark suite. In *Proc. CVPR*, pages 3354–3361, 2012. 1, 5

[4] F. Güney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proc. CVPR*, June 2015. 5

[5] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. PAMI*, 30(2):328–341, 2008. 1, 4

[6] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proc. NIPS*, pages 109–117, 2011. 1, 4

[7] M. Lang, O. Wang, T. Aydin, A. Smolic, and M. H. Gross. Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph.*, 31(4):34, 2012. 3, 4

[8] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proc. CVPR*, 2015. 5

[9] D. Min, J. Lu, and M. N. Do. Depth video enhancement based on weighted mode filtering. *IEEE Trans. Imag. Proc.*, 21(3):1176–1190, 2012. 1

[10] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Proc. CVPR*, pages 3017–3024, 2011. 1

[11] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. A. Dodgson. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In *Proc. ECCV*, pages 510–523, 2010. 1, 2, 5

[12] C. Vogel, S. Roth, and K. Schindler. View-consistent 3d scene flow estimation over multiple frames. In *Proc. ECCV*, pages 263–278. Springer, 2014. 5

[13] C. Vogel, K. Schindler, and S. Roth. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, pages 1–28, 2015. 1, 2, 5

[14] K. Yamaguchi, D. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. In *Proc. CVPR*, pages 1862–1869, 2013. 5

[15] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *Proc. ECCV*, pages 756–771. Springer, 2014. 1, 2, 5

[16] J. Žbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. *arXiv preprint arXiv:1409.4326*, 2014. 5

[17] K. Zhang, Y. Fang, D. Min, L. Sun, S. Yang, S. Yan, and Q. Tian. Cross-scale cost aggregation for stereo matching. In *Proc. CVPR*, pages 1590–1597, 2014. 1