STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LIX**, Number 1, 2014

POST PROCESSING VOTING TECHNIQUES FOR LOCAL STEREO MATCHING

ALINA MIRON

Abstract. In this paper we propose two extensions to the Disparity Voting scheme for local stereo matching algorithms, that improve the quality of the disparity map. These extensions are based on two separate hypothesis on the disparity map: the real disparity value of a pixel is found close to the disparity value that gives the minimum matching cost and the real disparity value of a pixel is not always the one given by the minimum *cost*, but nevertheless is found among one of the minimum matching *costs*. These techniques are compared on a real road scene dataset (KITTI) against the classical Disparity Voting and Winner-Takes-All strategy.

1. Introduction

Stereo vision refers to the extraction of depth information from a scene when viewed by a two camera system (eg. human eyes). When an object is viewed from a great distance, the optical axes of both eyes are parallel, therefore the object's projections, as seen by each eye independently, is similar. On the other hand, when the object is placed near the eyes, the optical axes will converge. The main application of human stereo vision is the perception of depth, while computer stereo vision has applications that vary from 3D reconstruction to image-based rendering or object hypothesis generation.

A task that is learned so easily by the human brain and performed unconsciously has proven to be difficult for computers. In traditional computer stereo vision, two cameras are placed horizontally at a certain distance in order to obtain different views of the scene. The distance between the cameras is called baseline and influences the minimum and maximum perceived depth.

Received by the editors: April 28, 2014.

2010 *Mathematics Subject Classification.* 68T45.

1998 *CR Categories and Descriptors.* I.2.10 [**ARTIFICIAL INTELLIGENCE**]: Vision and Scene Understanding – *3D/stereo scene analysis*; I.4.8 [**IMAGE PROCESSING AND COMPUTER VISION**]: Scene Analysis – *Stereo*.

Key words and phrases. stereo vision, disparity refinement, road scene analysis.

The amount to which a single pixel is displaced in the two images is called disparity and it is inversely proportional to its depth in the scene: closer objects will have greater disparity than background objects.

The field of application that we target is the one of intelligent vehicles, in particular the detection of road obstacles like pedestrians. A robust and accurate disparity map is essential in order to have pertinent information over the location of pedestrians in a scene and the relative distance from the vehicle to those pedestrians.

As presented by [11], most of the stereo matching algorithms rely on four important steps: Cost computation; Cost aggregation; Disparity computation/optimisation and Disparity refinement. Each step is important for the quality of the disparity map, with the cost computation step being crucial as it stands at the basis of the stereo matching algorithms.

There exists several studies where comparison of cost functions is performed, the most extended ones being made by Hirschmuller and Scharstein [2, 3]. While there are different studies that cover the step of Cost Computation, from our knowledge, the step of Disparity computation for local algorithms has been overlooked. In [7], it has been shown that different refinement techniques can greatly improve the accuracy of the obtained disparity map. Unfortunately, most of the disparity refinement techniques rely on computing two disparity maps, one for left and the other one for right image, leading to an increase in computation time. This is followed by techniques as left-right consistency check in order to refine the disparity results. We argue that the disparity map can be improved starting from the initial disparity estimation.

This article is organized as follows: we start by briefly describing the stereo matching algorithms employed in section 2, while in section 3 are presented the two disparity voting techniques proposed. We continue with a description of the results obtained (section 4) and we conclude with a discussion about the impact of the proposed techniques.

2. Stereo Matching Algorithm

Cost function. For the cost function we are going to use *DiffCensus* proposed in [8]. *DiffCensus* has proven to be robust to radiometric distortions, and in the same time provides better results on road scene images than other cost functions. The main advantage of the function is that it does not rely on the value of the pixel intensity but on the difference of intensity between a considered pixel and its neighbourhood (equation 1). This keeps the function as a non-parametric one while incorporating extra information.

$$C_{DIFFCensus}(x,y,d) = \rho(C_{census}(x,y,d),\lambda_{census}) + \rho(C_{DIFF}(x,y,d),\lambda_{DIFF}) \tag{1}$$

where C_{census} is the Census transform cost function proposed by [13], x, y are the coordinates of the considered pixel, d is the disparity value, λ_{census} and λ_{DIFF} are user defined constants, while ρ is defined in eq 2 and C_{DIFF} is defined in eq. 3 .

$$\rho(c,\lambda) = 1 - exp(-\frac{c}{\lambda}) \tag{2}$$

$$C_{DIFF}(x,y,d) = |\overline{DIFF}(x,y) - \overline{DIFF}(x,y-d)| \tag{3}$$

where $n \times m$ is the same support window that is used to compute the CT.

$$\overline{DIFF}(u,v) = \frac{DIFF(u,v)}{CensusSize} \tag{4}$$

where *CensusSize* is the size of the bit string given by the support window $n \times m$ and *step*.

$$DIFF(u,v) = \sum_{\substack{i=\overline{1:step:n} \\ j=\overline{1:step:m}}} (|I(u,v) - I(u+i,v+j)|), \tag{5}$$

where $I(u,v)$ is the intensity of the pixel located at coordinates (u,v).

Cost aggregation. Zhang et al. [14] proposed an efficient technique based on cross-zone aggregation for computing a pixel aggregation region, that takes into consideration color and euclidean distances.

The idea behind is to construct a *cross* region for each pixel. For this, it is necessary to find only four pixels, corresponding to the end of the four arms: up, down, left and right (figure 1.a). Then, in order to construct a region of various shapes, for each pixel that lies on the vertical arm, the horizontal arm will give the region boundaries for the specific row.

In order to choose an arm endpoint p_e for a given pixel $\mathbf{p}$, two rules are applied that pose limitations on color similarity and maximum arm length:

- $D_c(p_e,p) < \tau$. τ is a user-defined threshold value, while the color difference is defined to be $D_c(p_e,p) = max_{i \in R,G,B}|I_i(p_e) - I_i(p)|$.
- $D_s(p_e,p) < L$. L is a user-defined threshold value and represents a maximum length in pixels. $D_s(p_e,p)$ is a spacial distance given by $|p_e - p|$.

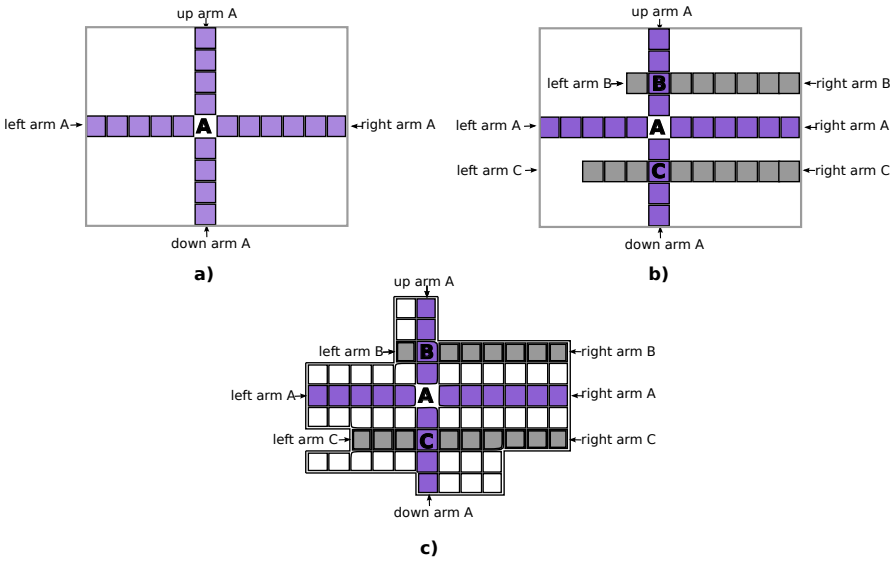


FIGURE 1. Cross region construction: **a)** For each pixel four arms are chosen based on some color and distance restrictions; **b)**,**c)** The cross region of a pixel is constructed by taking for each pixel situated on the vertical arm, its horizontal arm limits.

After having the cross region for each pixel, the next step is to compute the cost in the defined region. For this, the cost aggregation is computed in two steps. First the horizontal matching cost is computed and stored, secondly the final cost is obtained by aggregating the intermediate results vertically. The two steps can be efficiently computed using $1D$ integral images.

As comparison for the local stereo matching algorithm based on cross-zone aggregation, we are also going to use a global stereo matching algorithm based on Graph Cuts. As described by Kolmogorov and Zabih [4], a graph cut is a partition of a graph with two distinguished terminals called source (s) and sink (t) into two sets V^s and V^t, such that $s \in V^s$ and $t \in V^t$. The cost of the cut is represented by the sum of the edges' weights between the two partitions. Finding the minimum cut (the cut of minimum costs among all possible cuts), and implicitly the minimum cost, can be resolved by computing a maximum flow between terminals. In practice, the global energy minimisation technique using graph cuts has been shown to be effective with the condition of having an appropriate cost function. The same *DiffCensus* cost function was used also for the graph cuts algorithm.

Disparity computation. As reference point we are going to consider two strategies for choosing the disparity: a classical one, Winner-Takes-All (WTA) and disparity voting.

The simplest strategy of choosing a disparity for a given pixel is the WTA strategy, i.e. finding the point that will minimize the matching cost (see equation 6). Unfortunately, this does not always give the best results. This is due to the fact that even if a certain disparity has the minimum matching cost, this does not make it necessarily the right disparity (stereo matching is an NP-complete problem).

$$d_p = min_{d_{min} \leq d \leq d_{max}} \sum_{q \in N_p} c(q, q-d) \tag{6}$$

where

- d_p is the final disparity assigned to pixel p
- d_{min} and d_{max} is the minimum possible disparity, respectively maximum.
- N_p represents the neighbourhood of pixel p that is taken as aggregation area
- $c(q, q-d)$ represents a cost between the pixel q in the left image and the corresponding pixel at disparity d in the right image

Contrary to WTA, a simple strategy is the disparity voting strategy proposed in [6] and reused in [7, 14]. The aggregation area will usually originate from the same scene patch. Therefore, the pixels in an aggregation area should share similar disparities. For every pixel p, having a disparity estimate d_p cumputed with WTA, a histogram h_p of disparities is build as showed by equation 7:

$$h_p(d) = \sum_{q \in U(p)} \delta(d_q, d) \tag{7}$$

where $U(p)$ represents the set of all aggregation areas that contain the pixel p, and the function δ is defined as follows:

$$\delta(d_a, d_b) = \begin{cases} 1 & \text{if } d_a = d_b \\ 0 & \text{otherwise} \end{cases}$$

$$d_p^* = argmax(h_p(d)) \tag{8}$$

where $d \in [0, d_{max}]$.

3. Proposed Voting Strategies

We extend the voting strategy proposed by [6] using two different assumptions:

- The real disparity of a pixel is found close to the disparity corresponding to the minimum matching cost (see figure 2.a))
- The real disparity of a pixel is given by one of the disparities that have a matching cost close to the minimum matching costs (see figure 2.b)).

Different from Zhang et al. [14], we propose an extension for the voting algorithm. Due to the fact that different but close disparities have similar matching costs, the surface of inclined objects will not appear very smooth. This corresponds to the first presented assumption. Our proposal is for the voting scheme to not only consider the disparity d_p obtained with WTA, but also the disparities in the interval $[d_p - v, d_p + v]$, as presented in equation 9. We will further refer to the strategy given by this assumption as *VotingInInterval*.

$$h_p(d) = \sum_{d \in [d-v, d+v]} \sum_{q \in U(p)} \delta(d_q, d) \tag{9}$$

In certain situations the minimum cost might not give the real disparity value. Moreover, there exist cases where the same minimum cost is shared by multiple disparity values. Therefore we can use the second assumption in order to model the voting strategy: the voting scheme will not only consider the disparity d_p obtained with WTA, but also the disparities that have close matching cost of d_p, as presented in equation 10. We will further refer to the strategy given by this assumption as *VotingMinCosts*.

$$h_p(d) = \sum_{d \in M_{\overline{1,v}}} \sum_{q \in U(p)} \delta(d_q, d) \tag{10}$$

where M represents a sorted set of disparity using as criterion the matching cost. M_1 corresponds to the disparity that has the minimum matching cost, therefore d_p.

4. Experiments

There exist several challenging databases for testing the stereo matching algorithms, from simulated road scenes like *Van Synthetic stereo* [12] and EISATS [9], to real road scenes with some degree of ground truth like KITTI [1], Make3D Stereo [10] or Ladicky[5]. Moreover one of the most well know benchmark for the stereo matching algorithms is the Middlebury[11] dataset. From all these datasets, the most used datasets are Middlebury, that contains

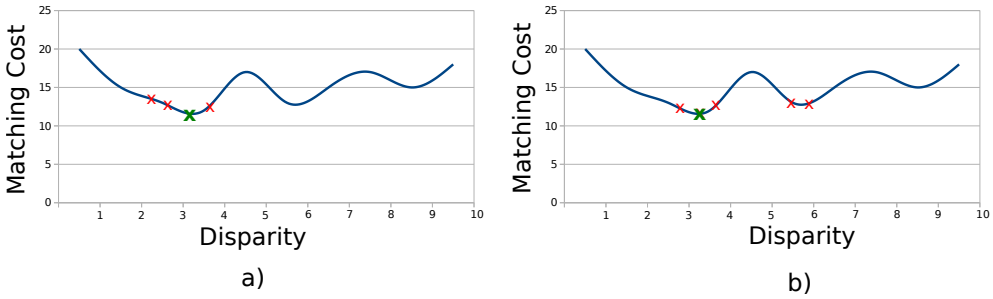


FIGURE 2. Proposed voting strategies: a) the first one uses voting strategy in an interval of disparities around the minimum matching cost (showed in green) b) while the second one uses a voting strategy for the disparities that give matching cost close to the minimum matching cost

images taken indoors in controlled light conditions, and KITTI that has real road scenes images.

In what follows, we use the KITTI stereo images for all the numerical experiments. KITTI dataset is divided into 194 images in the training set for which the ground truth images is provided, and 195 images in the testing set for which an evaluation server should be used in order to obtain the results. The following experiments are performed only on the 194 images in the training set[1] All the cost functions in this paper are evaluated by the average percentage of erroneous pixels in all zones, occlusions included, and computed at 3 pixels error threshold.

In table 1 is presented a comparison of obtained error rates on the KITTI dataset using cross-zone aggregation, the cost C_{DIFFCF}, and five strategies for deciding the final disparity: WTA, the voting method proposed by Zhang et al. [14], a global method Graph Cuts [4] and the two proposed voting strategies. For the proposed voting strategies the parameter v was empirically chosen to be *two* for VotingInInterval strategy and *six* for VotingMinCosts strategy. It can be observed that by simply adding the votes to a disparity interval rather than just one disparity values the error rate decreases with 2.2%. Adding the votes not just for the minimum matching cost, but rather for all the disparity values that give a matching cost close to the minimum one, also improves the results, but not as much as the voting interval strategy.

[1]At the moment of performing the tests, only one submission in 72 hours was allowed on the evaluation server.

Disparity Decision Strategy	**Error Rate**
Winner Takes-All	15.05%
Voting Zhang et al. [14]	12.70%
Graph Cuts	13.05%
Proposed VotingInInterval (v=2)	**10.50**%
Proposed VotingMinCosts (v=6)	11.05%

TABLE 1. Comparison of different strategy methods for choosing the disparity

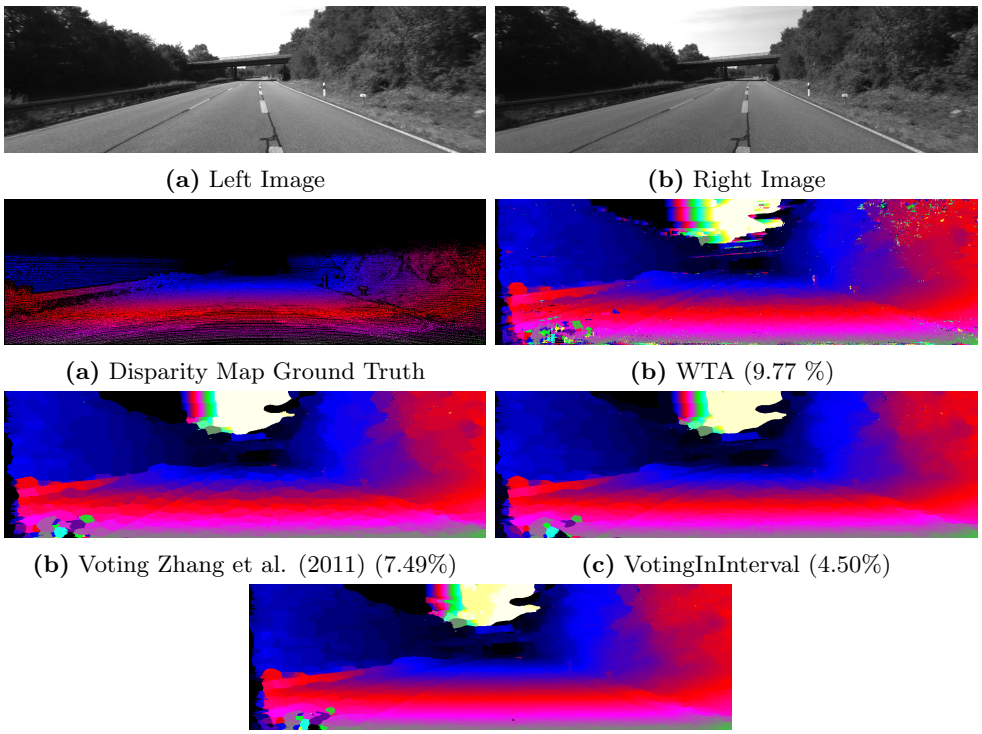
In figure 3 is presented a visualisation of the disparity map produced by the compared algorithms: WTA, Voting Zhang et al. (2011) , and the two proposed Voting strategies (VotingInInterval and VotingMinCosts). From the presented images, it can be observed that the VotingMinCosts produces the smoothness disparity map, but the overall better results are obtained by the VotingInInterval strategy.

5. Conclusions

In this paper we have presented two new Voting strategies for the step of Disparity computation for local Stereo Matching algorithms. These are based on two rather simple assumptions about the disparity map. The proposed strategies have proven that they are capable to improve the disparity map results. In comparison with other techniques of disparity refinement, because the decision is taken at the moment of disparity computation, both strategies come with a very low computation cost. As future work, the two strategies could be combined in order to take advantage of both voting mechanisms. Also, these strategies can be used in combination with any stereo matching algorithms. Thus, it would be interesting a comparison across multiple stereo vision algorithms in order to see if the performance gain remains always the same.

References

[1] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR)*, Providence, USA, June 2012.



(a) Left Image **(b)** Right Image

(a) Disparity Map Ground Truth **(b)** WTA (9.77 %)

(b) Voting Zhang et al. (2011) (7.49%) **(c)** VotingInInterval (4.50%)

(d) VotingMinCosts (6.9%)

FIGURE 3. Different Disparity maps produced by different Voting Strategies for the same image

[2] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[3] Heiko Hirschmuller and Daniel Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1582–1599, 2009.

[4] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *IEEE International Conference on Computer Vision*, volume 2, pages 508–515, 2001.

[5] Lubor Ladickỳ, Paul Sturgess, Chris Russell, Sunando Sengupta, Yalin Bastanlar, William Clocksin, and Philip HS Torr. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*, pages 1–12, 2012.

[6] Jiangbo Lu, Gauthier Lafruit, and Francky Catthoor. Anisotropic local high-confidence voting for accurate stereo correspondence. In *Proc. SPIE-IS&T Electronic Imaging*, volume 6812, pages 605822–1, 2008.

[7] Xing Mei, Xun Sun, Mingcai Zhou, Shaohui Jiao, Haitao Wang, and Xiaopeng Zhang. On building an accurate stereo matching system on graphics hardware. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 467–474, 2011.

[8] Alina Miron, Samia Ainouz, Alexandrina Rogozan, and Abdelaziz Bensrhair. A robust cost function for stereo matching of road scenes. *Pattern Recognition Letters*, 38:70–77, 2014.

[9] Sandino Morales and Reinhard Klette. Ground truth evaluation of stereo algorithms for real world applications. In *Computer Vision–ACCV 2010 Workshops*, pages 152–162. Springer, 2011.

[10] Ashutosh Saxena, Jamie Schulte, and Andrew Y Ng. Depth estimation using monocular and stereo cues. IJCAI, 2007.

[11] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1):7–42, 2002.

[12] Wannes Van Der Mark and Dariu M Gavrila. Real-time dense stereo for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):38–50, 2006.

[13] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. *ECCV*, pages 151–158, 1994.

[14] Ke Zhang, Jiangbo Lu, and Gauthier Lafruit. Cross-based local stereo matching using orthogonal integral images. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(7):1073–1079, 2009.

Insa Rouen, Avenue de l'Université, 76800 Saint-Étienne-du-Rouvray, France; Universitatea Babeş-Bolyai, Facultatea de Matematică şi Informatică, Str. Mihail Kogălniceanu, nr. 1, 400084 Cluj-Napoca

E-mail address: alina.miron@insa-rouen.fr