A Method for Sparse Disparity Densification using Voting Mask Propagation

Jarno Ralli[1], Javier Díaz[1], and Eduardo Ros[1]

jarno@ralli.fi, jdiaz@atc.ugr.es, eros@atc.ugr.es

[1]Departamento de Arquitectura y Tecnología de Computadores*

Abstract

We describe a novel method for propagating disparity values using directional masks and a voting scheme. The driving force of the propagation direction is image gradient, making the process anisotropic, whilst ambiguities between propagated values are resolved using a voting scheme. This kind of anisotropic densification process achieves significant density enhancement at a very low error cost: in some cases erroneous disparities are voted out, resulting not only in a denser but also a more accurate final disparity map. Due to the simplicity of the method it is suitable for embedded implementation and can also be included as part of a system-on-chip (SOC). Therefore it can be of great interest to the sector of the machine vision community that deals with embedded and/or real-time applications.

1 Introduction

Disparity is originally defined as being the horizontal difference of a 3D point being projected on two adjacent imaging devices (e.g. stereo-rig) and if both intrinsic- and extrinsic parameters of the imaging system are known, complete 3D-reconstruction of the scene is possible. However, even if we do not know all the necessary parameters to do a complete 3D-reconstruction, disparity still conveys relative information of the 3D structures of the scene which can also be useful.

Disparity extraction models are based on local or global optimization methods that minimize (or maximize) matching cost of image features between two or more images. Practically all the sparse models have some kind of threshold or other parameter, either implicit or explicit, which affects density and at the same time error in the derived disparity map [18][3]. On the other hand the global methods that minimize energy functions within the whole scene, through local operations, usually derive a disparity map that is typically 100% dense (sometimes detecting occlusions as well). Such global minimization can be done using different approaches such as variational methods [7][1][2][4][6] or graph cuts [9][10]. Typically, depending upon the sparse method

*Escuela Técnica Superior de Ingeniería Informatica y de Telecomunicacíon, Universidad de Granada, Calle Periodista Daniel Saucedo Aranda s/n, E-18071 Granada, Spain

used, as density increases, after a certain limit error also starts to increase concomitantly. Therefore, it is worth calculating a less dense, high-confidence disparity map and afterwards increasing the density by propagating the correct disparity values. In this way we achieve better accuracy vs density trade-off than by directly reducing the reliability threshold and thus increasing density at the expense of higher error. Many global, dense, disparity calculation methods have built-in mechanisms for propagating/diffusing disparity [1][17] but sparse methods usually lack this capacity. To the best of our knowledge there are very few independent propagation methods, apart from interpolation [14] and diffusion [22], that can be applied as a post-processing step. By independent we mean that the propagation method does not depend upon the algorithm used to derive the initial disparity map. This work proposes a new densification method that is able to arrive at denser and more accurate results than the standard one-stage disparity algorithms such as dynamic programming, block- matching and so on that only slightly affects the error rate. Since the scheme is based on very simple operations, it can be considered suitable for efficient implementation. Our method resembles image driven anisotropic diffusion, used for instance by Alvarez et al. [1] in variational disparity calculation, in the sense that the propagation direction is based on the image gradient. Instead of using a set of equations for defining the diffusion model as it is done in [1] our approach (VMP) uses a bank of predefined masks and a voting process to define the local interactions driving the diffusion process.

2 Method

The first step is to calculate a sparse, high-confidence, stereo map. Many feature-based disparity calculation methods match edges present in stereo-images, since these can be considered relatively robust features [8][11]. For the reasons set out in the Introduction we use here a simplified dynamic programming technique based on image edges [14]. The rational for using dynamic programming is that it has been shown to be both computationally efficient [12] and capable of producing highly accurate results [3]. Nevertheless, as mentioned earlier, our densification approach does not depend upon the method used for producing the initial sparse disparity map. The second step is to apply voting mask propagation (VMP) for propagating disparity in the direction where estimations are expected to be similar and to use voting for resolving ambiguities. Local support of the voting process is based on directional masks: for each image position for which disparity is known a mask from a pre-determined bank is chosen, depending upon the underlying image structure (gradient). The properties of the chosen filter define how many votes each of the neighborhood positions will receive.

2.1 Propagation direction

Since without further image analysis we cannot be sure of which object an image pixel for which the disparity is known belongs to, and since we assume that inside objects disparity changes gradually, image gradient is used as a driving force of the propagation direction. We assume that two different objects will almost certainly have two different disparity levels. By propagating in a direction perpendicular to the image gradient we reduce errors since different objects have varying disparities divided by an edge. This assumption of local maximum gradient separating different objects is also the basis of anisotropic diffusion, where diffusion direction is driven by the gradient [22][13]. In this work we concentrate on the case where the disparities for the edges are known and the disparities are propagated in an edge-wise direction. There is, however, no reason why the disparities not residing at the edges could not be propagated as well. The tangent-to-edge direction is approximated by calculating image gradient.

2.2 Bank of masks

A bank of masks is designed using a 2D multivariate Gaussian distribution which is rotated in order to generate masks corresponding to different propagation directions. The basic mask, corresponding to orientation 0° (i.e. the horizontal axis), is calculated as per (1).

$$z = G(i,\, j,\, \mu_i = \mu_j = 0,\, \Sigma), \qquad i = -N \ldots N,\, j = -P \ldots P \tag{1}$$

where $G(\cdot)$ denotes multivariate Gaussian, (i, j) are the coordinates of the mask, $(\mu_i = \mu_j = 0)$ are the mean, Σ is the covariance, z is the number of votes that each position receives and N and P define the mask size. Fig. 1 shows a voting mask corresponding to several different orientations (rotations).

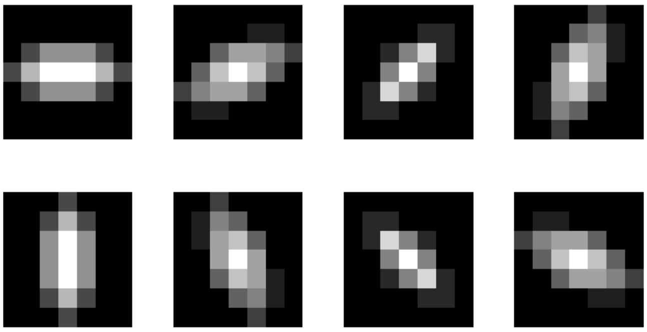


Figure 1: A bank of 7x7 voting masks corresponding to different orientations. Intensity codifies the number of votes each position receives.

The use of Gaussian distribution is motivated by the fact that it reflects the probabilistic nature of our approach: the underlying image structure drives the propagation direction and thus reflects our belief on how the disparity is distributed. Other authors have used similar approaches for image denoising [5]. Furthermore Gaussian multivariate distribution allows a smooth transition from isotropic to anisotropic cases, depending on the certainty of the image structure, which can be used in more elaborated schemes by futher analyzing the image. Besides a Gaussian distribution can be implemented as a separable convolution thus making it efficient computationally.

2.3 Choosing the mask

Once the orientations of the edge tangents have been approximated, propagation is carried out for each disparity value using the mask whose orientation best matches the tangent of the edge. The most closely corresponding mask's centre is placed on top of the disparity value of interest and each pixel within mask size receives as many votes for the disparity value as defined by the mask. This is shown in the (2).

$$V_{x+i,y+j}(D_{x,y}) = G_{x,y,\Delta}(x+i,\, y+j), \qquad i = -N \ldots N,\, j = -P \ldots P \tag{2}$$

where $V_{x,y}$ indicates votes received by position (x, y) for disparity, D, $G_{x,y,\Delta}(x+i, y+j)$ denotes how the Gaussian voting mask, chosen as per gradient Δ, with a size of $(2N+1, 2P+1)$, placed at (x, y) votes for each mask position. As a final step, after the disparities have been propagated for each of the original disparity values, each pixel position assumes the disparity that receives most votes, as defined in (3).

$$D_{x,y} = MAX_V\Big((V_{x,y}, D_{x,y})\Big) \tag{3}$$

where $D_{x,y}$ indicates the final chosen disparity value for a position (x, y) and $MAX_V\Big((V, D)\Big)$ returns the disparity value that has received the most votes for a set of vote-disparity tuples (V, D). Due to the spatial support of the voting mechanism erroneous values are in certain cases effectively voted out: if within a certain neighborhood there are more correct values than erroneous values, the erroneous values receive fewer votes and are discarded. Interestingly enough this effect allows the densification process to arrive at a denser but at the same time more accurate disparity map than the original.

2.4 Pseudocode

For the sake of clarity, below we have included a pseudocode of the propagation process.

Algorithm 1 Pseudocode Describing the Voting Process.

```
INPUT: I (image driving the voting process)
INPUT: D (disparity map)
INPUT: M (bank of masks with mask size (N, P))
STORAGE: V (a matrix for storing the votes)

//OBTAIN COORDINATES FOR DISPARITIES
(x, y) = coords( notEmpty(D) )

for $i = 1 \rightarrow numel(x)$ do
    //CHOOSE THE CLOSEST MASK TO GRADIENT NORMAL
    ΔI = calculateImageGradient( I( x(i), y(i) ) )
    mask = chooseMask( M, ΔI )
    //VOTE USING THE CHOSEN MASK
    V = vote( x(i), y(i), mask, D( x(i), y(i)), N, P )
end for
(x, y) = coords( notEmpty(V) )

//CHOOSE WINNERS
for $i = 1 \rightarrow numel(x)$ do
    Out( x(i), y(i) ) = MAX( V( x(i), y(i) ) )
end for
```

The actual voting process can be seen as superimposition of the chosen mask on top of the disparity value to be propagated where each neighboring pixel (defined by the mask) receives as many votes for the disparity as defined by the weight of the mask as each position. For each pixel position the number of votes for each disparity needs to be stored so that a winner can be chosen accordingly.

3 Experiments and Results

We benchmarked the method using well known stereo-images from the Middlebury database[1]. In order to study how the VMP densification process behaves when dealing with different initial densities and/or errors, we have used two different methods for generating the initial disparity maps provided to the VMP model. The different methods used were dynamic programming (DP) [14] and a phase-based method [18][16]. Furthermore we have used different thresholds and interleave factors for DP in order to generate initial disparity maps with different densities and errors. Computational complexity of the our method was approximated by comparing it with execution times of the DP method. We also introduce a sample application that clearly benefits from a more dense disparity map as input. In the experiments, size of the propagation masks was 7x7 pixels. Density is given in terms of a ratio between the number of pixels for which disparity has been defined and the total number of pixels in the image. Overall accuracy is measured as the percentage of correct pixels (±1 disparity level) calculated against the ground-truth values.

3.1 Results

Fig. 2 shows the original stereo-pair images and results calculated directly using DP and densified by VMP.

[1] `http://vision.middlebury.edu/stereo/`

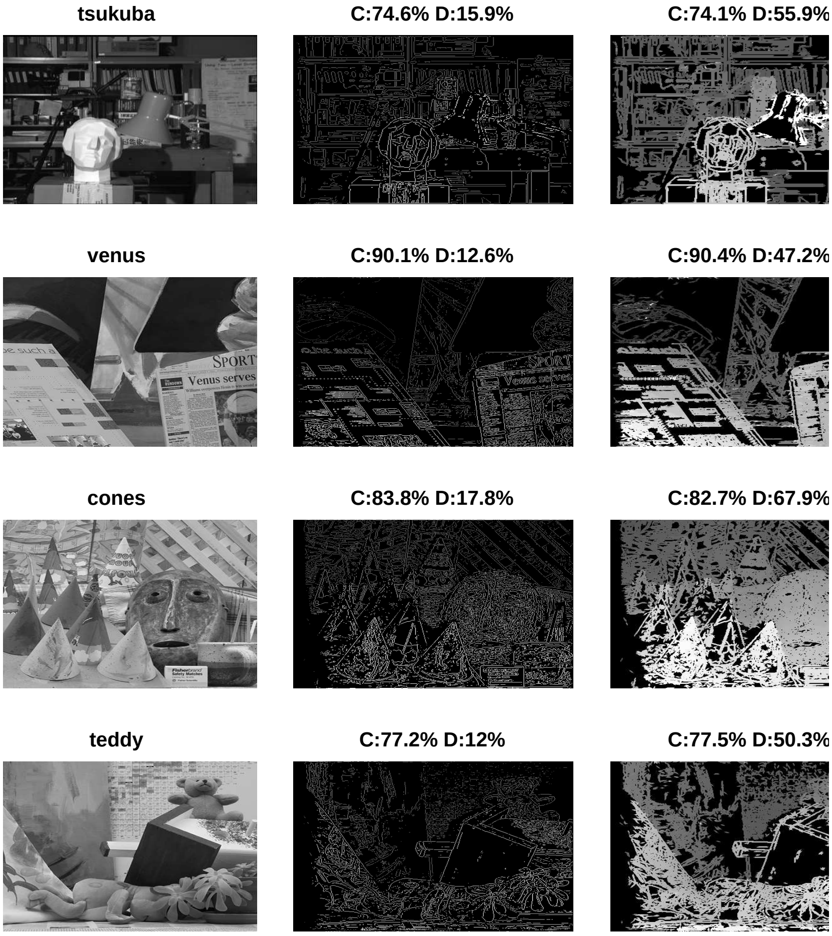
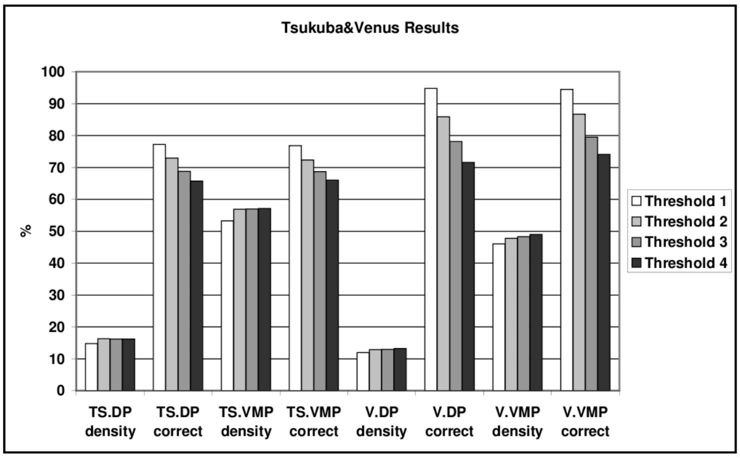
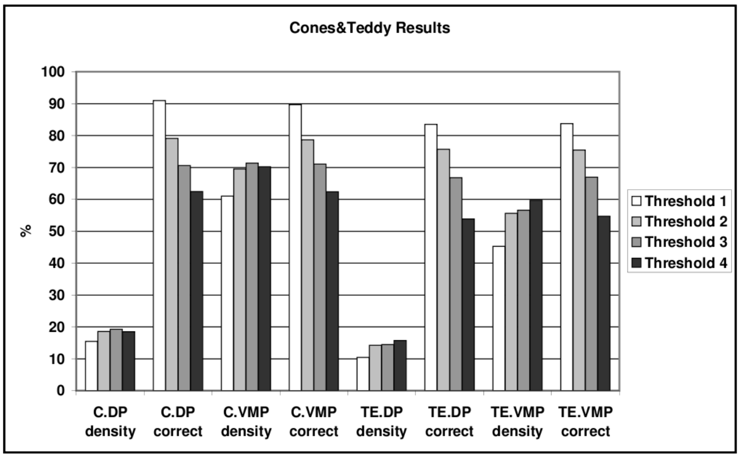


Figure 2: Results for the test images: the left-hand column contains left-hand images of the original stereo-pairs, the middle column shows disparity maps calculated by dynamic programming and the right-hand column disparity maps densified using VMP. C denotes the percentage of correct disparities (±1 disparity level) and D, density.

Fig. 3 demonstrates the results for four different initial maps with different densities. The initial maps are calculated using different thresholds for occlusion detection with the effect of increasing density at the expense of accuracy. Thus it can be observed that after certain reasonable limit, in order to obtain even more dense map, the error starts to increase. In such a case it is better to calculate more reliable initial map and then densify.



(a) Tsukuba&Venus results



(b) Cones&Teddy results

Figure 3: Densification results for several initial densities obtained using different thresholds for dynamic programming. DP refers to results calculated by directly using dynamic programming and VMP refers to results densified from corresponding DP results using voting mask propagation. (a) TS refers to Tsukuba and V refers to Venus images (b) C refers to Cones and TE refers to Teddy images

Fig. 4 shows the results for the Venus case only. It can be clearly seen that as the cost for occlusions gets higher (threshold from one to four) density of the resulting disparity map increases slightly at the expense of accuracy. On the other hand as the error increases the voting mechanism starts to vote out some of the erroneous values thus increasing the accuracy.

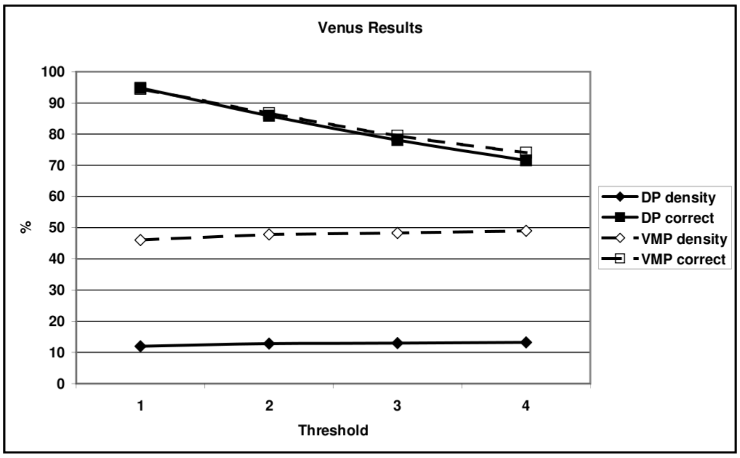


Figure 4: Densification results for several initial densities obtained using different thresholds in dynamic programming. DP refers to dynamic programming and VMP to voting mask propagation. Density refers to the density of the obtained disparity map and correct refers to the percentage of correct disparities (±1 disparity level).

Fig. 5 displays densification results using an initial disparity map calculated by a phase-based method [16]. Density and error of the resulting densified map is similar to the rest of the results even though the starting density with this technique is significantly higher than in the previous experiments based on dynamic programming.

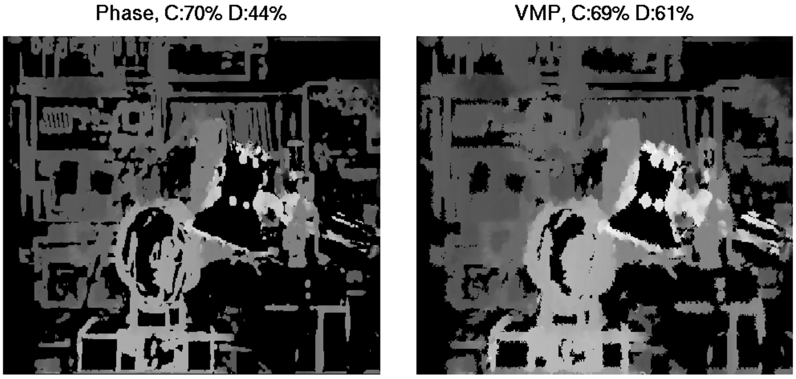
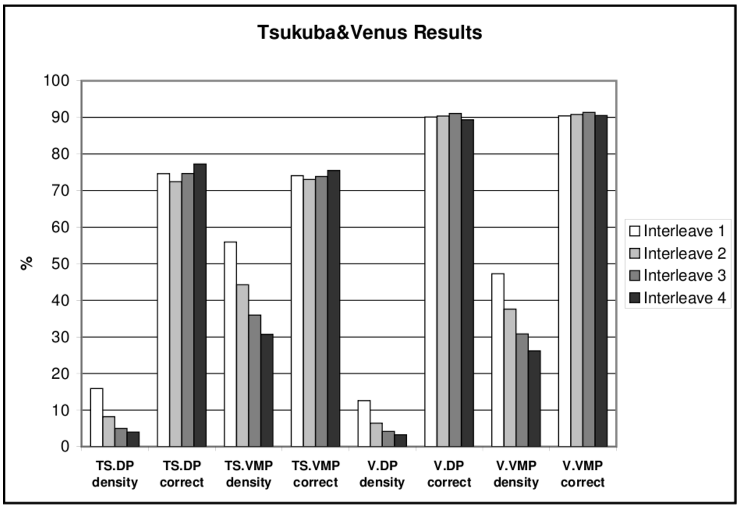
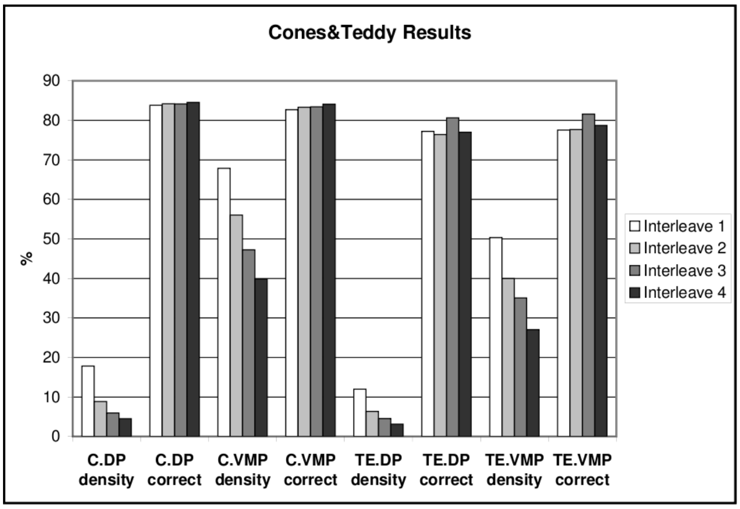


Figure 5: Phase refers to phase-based method and VMP to voting mask propagation. C denotes the percentage of correct disparities (±1 disparity level) and D, density.

Fig.6 and fig.7 demonstrate the results of different initial disparities upon the densified disparity and the obtained densification factor. Different initial density disparities were obtained using DP with different vertical line interleaves (interleave 1 = disparity calculated for all the vertical lines, interleave 2 = disparity calculated for every second line etc.). This experiment demonstrates both robustness of the VMP method in relation to initial density and what kind of densification factors can be expected. As the density of the input map increases the densification factor decreases which is due to overlapping of the propagation filters.



(a) Tsukuba&Cones results



(b) Cones&Teddy results

Figure 6: Densification results for several initial densities obtained using different interleaves for dynamic programming. DP refers to results calculated by directly using dynamic programming and VMP refers to results densified from corresponding DP results using voting mask propagation. (a) TS refers to Tsukuba and V refers to Venus images (b) C refers to Cones and TE refers to Teddy images.

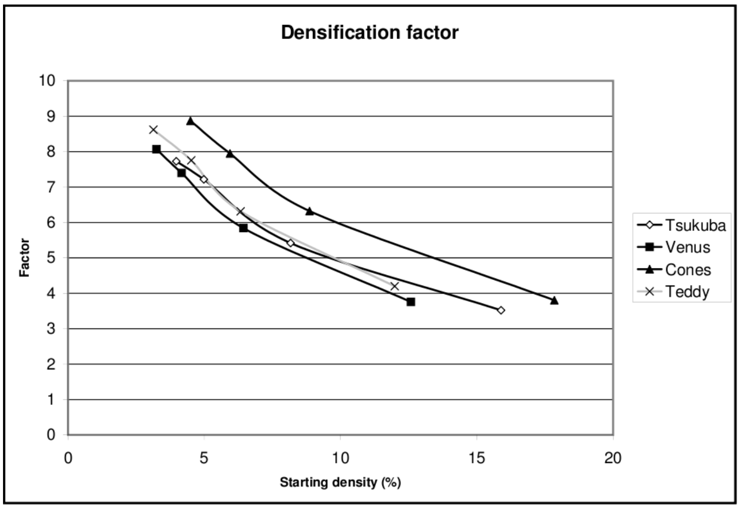


Figure 7: Densification factor results for different initial disparities. Y-axis shows the density of the initial map while X-axis displays the obtained densification factor.

Fig. 8 displays the computational times of both the dynamic programming and the voting mask propagation methods, implemented in Matlab. The four different cases correspond to the four different thresholds already seen in fig. 3. This kind of a comparison is approximate since it depends on implementation issues but however it does give a valuable hint of the relative computational complexities.

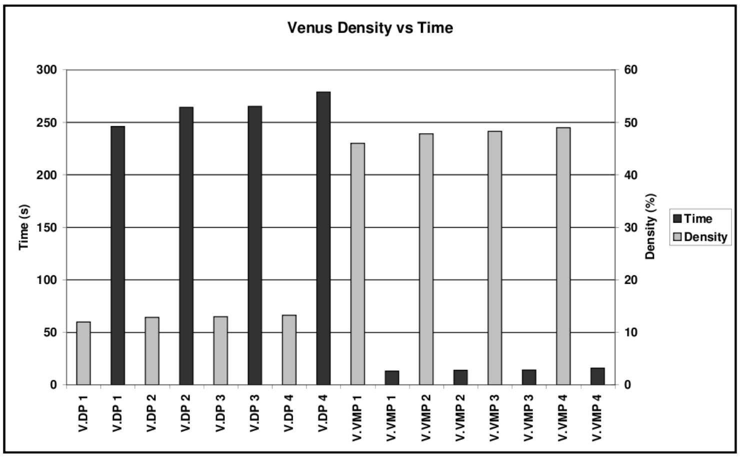


Figure 8: Computation times for both the dynamic programming and the voting mask propagation methods. DP refers to dynamic programming and VMP to voting mask propagation.

In applications where less resolution is needed, further densification can be achieved by down-scaling. Fig. 9 shows the results of down-scaling disparity maps by a factor of two, using a 2x2 median filter and discarding those pixels that do not have a disparity estimation. In order to calculate the percentage of correct disparities, each of the downscaled disparity maps was upscaled to the same size as the ground-truth.

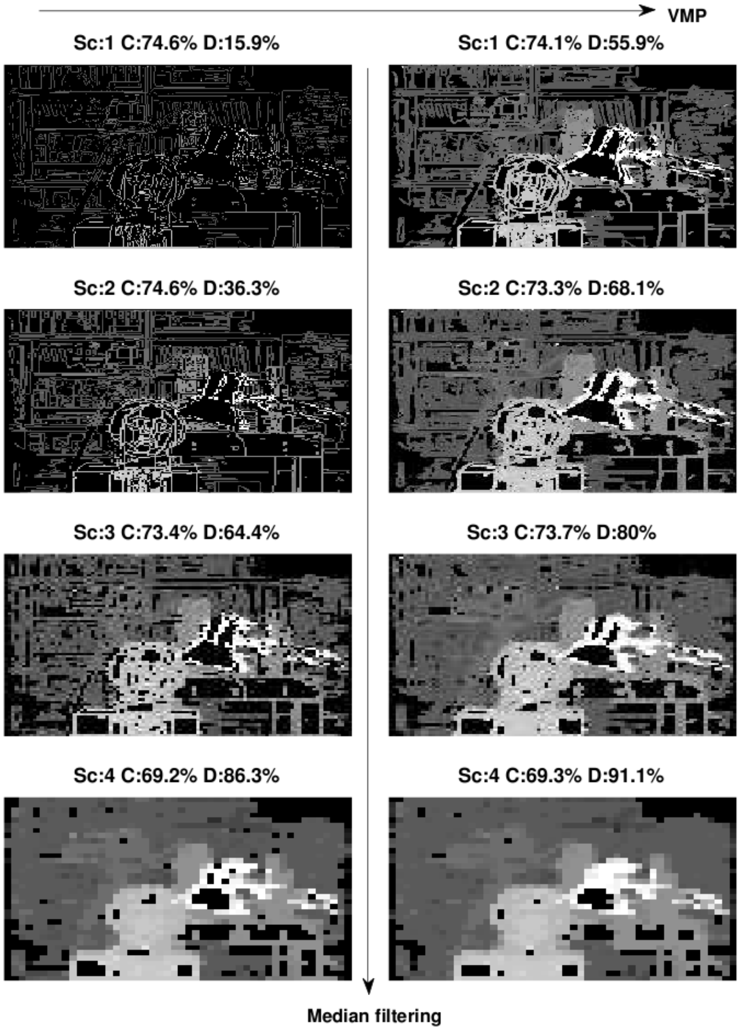


Figure 9: Down-scaling results using a median filter: the left-hand column contains results of down-scaling Tsukuba disparity calculated using dynamic programming, whilst that on the right shows the results of applying VMP on the first scale and then down-scaling. Sc denotes the scale, while C and D denote percentage of correct disparities (±1disparity level) and density.

3.2 Sample application

We present here a specific application which benefits from a denser sparse-disparity map: fusion of sparse- and dense-disparity stereo [15] for disambiguation of dense disparity estimations. In the framework of DRIVSCO [15] Ralli et al. have demonstrated the benefits of using symbolic, reliable sparse-disparity to disambiguate unclear cases in dense-disparity calculation. We tested the fusion scheme, with different sparse-disparity densities, upon a hardware simulation of a phase-based [18][16] disparity calculation method with and without fusion. The hardware simulation approximates a realtime FPGA implementation [20][21] of the phase-based method. Due to limited on-chip computational resources the implementation requires a trade-off between accuracy and efficiency. In such a scheme where external approximations are available, these can be used to guide the dense method and thus the accuracy can be restored as shown in fig. **??**.

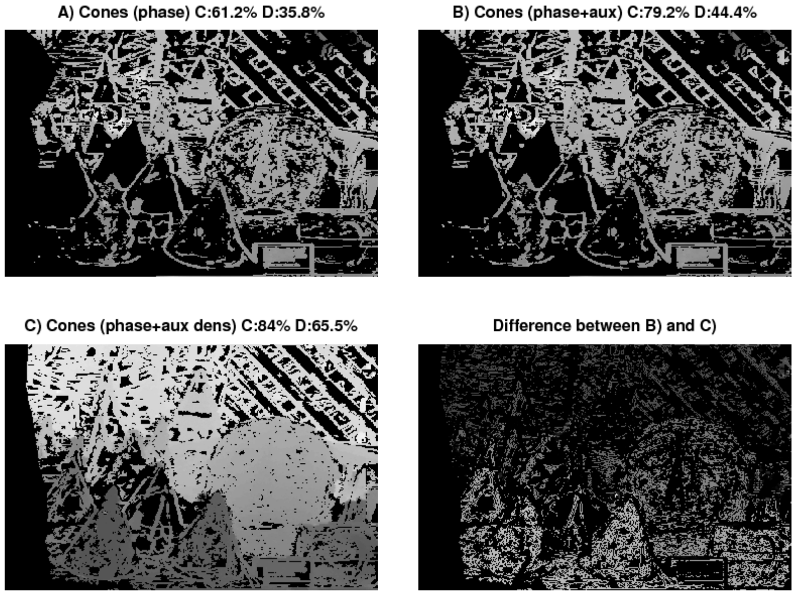

A) Cones (phase) C:61.2% D:35.8%
B) Cones (phase+aux) C:79.2% D:44.4%
C) Cones (phase+aux dens) C:84% D:65.5%
Difference between B) and C)

Figure 10: Dense-disparity calculation results: (a) without fusion (b) fusion using sparse-disparity with 15% density (c) fusion using sparse-disparity with 60% density and (d) the difference between (b) and (c). C denotes the percentage of correct disparities and D, density.

As can be seen in fig. 10, the fusion process benefits clearly from a denser sparse-disparity map used to guide the dense disparity calculation method.

4 Conclusion

We have shown that our novel method of propagating sparse disparity information based on directional masks and a voting scheme is capable of significantly increasing density with a very minor increase in overall error, thus considerably enhancing the initial sparse disparity map. Further densification can be achieved by down-scaling, by active interpolation [23][24][19] or by diffusion [22]. Even though in this study we have used VMP for propagating binocular visual information based on monocular cues, VMP can also be used for propagating other visual cues, such as optical flow and others. The simplicity of the method facilitates its efficient implementation.

5 Acknowledgements

This work was supported by the EU research project DRIVSCO (IST-016276-2) and the Spanish Grants DINAM-VISION (DPI2007-61683), RECVIS (TIN2008-06893-

C03-02), P06-TIC-02007 and TIC-3873. The authors thank A.L. Tate for revising their English text.

References

[1] L. Alvarez, R. Deriche, J. Sánchez, and J. Weickert. Dense disparity map estimation respecting image discontinuities: A pde and scale-space based approach. *Journal of Visual Communication and Image Representation*, 13(1):3–21, 2002.

[2] L. Alvarez, J. Weickert, and J. Sánchez. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1): 41–56, 2004.

[3] M.Z. Brown, D. Burschka, and G.D. Hager. Advances in computational stereo. *Pattern analysis and machine intelligence*, 25(8):993–1008, 2003.

[4] Thomas Brox. *From Pixels to Regions: Partial Differential Equations in Image Analysis*. PhD thesis, Saarland University, 2005.

[5] Dongwook Cho and Tien D. Bui. Multivariate statistical modeling for image denoising using wavelet transforms. *Signal Processing: Image Communication*, 20(1):77–89, 2005.

[6] Olivier Faugeras and Renaud Keriven. Level set methods and the stereo problem. In *Proc. of First International Conference on Scale-Space Theory in Computer Vision*, pages 272–283, 1997.

[7] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17: 185–203, 1981.

[8] X. Jiangjian and S. Mubrak. Two-frame wide baseline matching. In *Proceedings of ICCV'03,2003*, pages 603–609, 2003.

[9] V. Kolmogorov. *Graph Based Algorithms for Scene Reconstruction from Two or More Views*. PhD thesis, Cornell University, 2003.

[10] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts. *Pattern Analysis and Machine Intelligence*, 26:147–159, 2004.

[11] N. Krüger and M. Felsberg. An explicit and compact coding of geometric and structural information applied to stereo matching. *Pattern Recognition Letters*, 25(8):849–863, 2004.

[12] G. Minglun and Y. Yee-hong. Real-time stereo matching using orthogonal reliability-based dynamic programming. *IEEE Transactions on Image Processing*, 16(3):879–884, 2007.

[13] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12: 629–639, 1990.

[14] J. Ralli, F. Pelayo, and J. Díaz. Increasing efficiency in disparity calculation. In *BVAI2007*, volume 4729, pages 298–307, 2007.

[15] J. Ralli, J. Díaz, S. Kalkan, N. Krüger, and E. Ros. Disparity disambiguation by fusion of signal- and symbolic-level information. *Machine Vision and Applications*, 2010. URL `http://dx.doi.org/10.1007/s00138-010-0266-z`.

[16] S.P. Sabatini, G. Gastaldi, F. Solari, J. Diaz, E. Ros, K. Pauwels, M.M. Van Hulle, N. Pugeault, and N. Krüger. Compact and accurate early vision processing in the harmonic space. In *Proceedings of VISAPP (1)*, pages 213–220, 2007.

[17] D. Scharstein and R. Szeliski. Stereo matching with non-linear diffusion. *International journal of computer vision*, 28(2):155–174, 1998.

[18] F. Solari, S.P. Sabatini, and G.M. Bisio. Fast technique for phase-based disparity estimation with no explicit calculation of phase. *Electronics Letters*, 37(23): 1382–1383, 2001.

[19] Hou Chun Ting and Hsueh Min Hang. Edge preserving interpolation of digital image using fuzzy inference. *Journal of Visual Communication and Image Representation*, 8(4):338–355, 1997.

[20] M. Tomasi, M. Vanegas, F. Barranco, J. Díaz, and E. Ros. High-performance optical-flow architecture based on a multi-scale, multi-orientation phase-based model. *IEEE Trans. on Circuits and Systems for Video Technology*, 20(12):1797–1807, 2010.

[21] M. Tomasi, M. Vanegas, F. Barranco, J. Díaz, and E. Ros. Massive parallel-hardware architecture for multiscale stereo, optical flow and image-structure computation. *IEEE Trans. on Circuits and Systems for Video Technology*, 22(2):282–294, 2012. ISSN 1051-8215. URL `http://dx.doi.org/10.1109/TCSVT.2011.2162260`.

[22] J. Weickert. *Anisotropic Diffusion in Image Processing*. PhD thesis, Universität Kaiserslautern, 1998.

[23] H. Yoo. Closed-form least-squares technique for adaptive linear image interpolation. *Electronics Letters*, 43(4):210–212, 2007.

[24] L. Zhang and W. Xiaolin. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Transactions on Image Processing*, 15: 2226–2238, 2006.