

# Parallel framework for dense disparity map estimation using Hamming distance

Victor Gonzalez-Huitron<sup>1</sup> · Volodymyr Ponomaryov<sup>1</sup> · Eduardo Ramos-Diaz<sup>2</sup> · Sergiy Sadovnychiy<sup>3</sup>

Received: 1 October 2016 / Revised: 28 May 2017 / Accepted: 17 July 2017 / Published online: 4 August 2017  
© Springer-Verlag London Ltd. 2017

**Abstract** A novel framework for sparse and dense disparity estimation was designed, and the proposed framework has been implemented in CPU and GPU for a parallel processing capability. The Census transform is applied in the first stage, and then, the Hamming distance is later used as similarity measure in the stereo matching stage followed by a matching consistency check. Next, a disparity refinement is performed on the sparse disparity map via weighted median filtering and color  $K$ -means segmentation, in addition to clustered median filtering to obtain the dense disparity map. The results are compared with state-of-the-art frameworks, demonstrating this process to be competitive and robust. The quality criteria used are structural similarity index measure and percentage of bad pixels ( $B$ ) for objective results and subjective perception via human visual system demonstrating better performance in maintaining fine features in disparity maps. The comparisons include processing times and running environments, to place each process into context.

**Keywords** Stereo matching · Disparity estimation · Parallel computing

## 1 Introduction

Estimation of disparity maps (DM) has been a heavily researched topic for more than a decade. One of the main reasons for this rising interest is the wide range of applications that are being conceived and designed in the medical imaging [1–3] and computer vision fields [4,5,9]. As a consequence, a variety of solutions and variations of former proposals have been presented with the aim of designing a robust, fast and reliable framework for specific needs or requirements.

We can classify the existing frameworks into two groups: those focused on speed and those focused on quality. The frameworks in the first group mainly use local search for the disparity estimation and fast smoothing stages, avoiding iterative methods. As a consequence, they fail to meet some quality criteria, specifically in more challenging scenarios. These methods are designed to provide a fast DM, when the error threshold is larger than the average and the resolutions and disparity range are limited. The quality-focused methods provide acceptable results in most scenarios, but the processing time is inconvenient due to limited compatibility with specified hardware or extremely iterative methods.

Due to the continuous advances in image processing algorithms, along with the use of GPU computation and development of multicore CPUs, it is possible to design and implement more exhaustive and abstract methods, achieving better results and shortening distances between frameworks, such as the ones aimed at real-time processing [7,8,10,12] and the ones aimed at obtaining the best quality criteria scores [6,11,13,14]. Actually, it is possible to choose a trustworthy and robust framework for a given task, but a general solution is still in development. During this work, an effort has been made to reduce the distance between speed and quality, by implementing the same process in CPU and GPU parallel architectures, with only minor modifications into a structure

✉ Victor Gonzalez-Huitron  
alx3416@hotmail.com

<sup>1</sup> ESIME Culhuacan, Instituto Politecnico Nacional, Av. Santa Anna 1000, Mexico City, Mexico

<sup>2</sup> Universidad Autonoma de la Ciudad de Mexico, Prolongacion San Isidro 151, Mexico City, Mexico

<sup>3</sup> Instituto Mexicano del Petroleo, Lazaro Cardenas 152, Mexico City, Mexico

of framework applied but maintaining the same results, thus reducing the processing time.

In this paper, there are three main contributions. Firstly, a Census transform GPU implementation, allowing to represent every pixel value in the form of a binary vector, is estimated from a few values in their neighborhood pixels. Secondly, this work presents a GPU implementation in the stereo matching stage with Hamming distance as similarity measure; this allows the proposed framework to estimate a sparse disparity map with large size datasets. And finally, a clustered median filtering (CMF) is applied for refinement disparity and occlusion handling, which, in conjunction with a clustering algorithm, as K-means, is capable to reconstruct an acceptable dense disparity map.

The remainder of the paper is organized as follows. In Sect. 2, some related works are described. In Sect. 3, the novel framework for the parallel dense disparity estimation is presented and the hardware implementations detailed; in Sect. 4, the quality criteria, datasets, and experimental results are presented; and in Sect. 5, the conclusions are drawn.

## 2 Related works

There exists several methods for obtaining disparity maps. Some of these frameworks will be compared with our approach. Among the comparison frameworks will be some works submitted in the Middlebury v3 table, available for two sizes (quarter and half size). Additionally, sparse and dense results are compared to this dataset.

Binary stereo matching (BSM) [17] is a sparse-only method, consisting of binary-based cost computation and aggregation. The cost volume is constructed through bitwise operations on a series of binary strings. Next, this approach is combined with a traditional winner-takes-all strategy. The method for computing the stereo matching cost with a convolutional neural network (IDR) [18] consists of a two-pass aggregation with a Census-gradient cost metric, followed by iterative cost penalization and disparity re-selection to encourage local smoothness of disparities. The disparities are chosen as the sum of truncated absolute differences (SAD).

In efficient large-scale stereo matching (ELAS) [19], a prior disparity image is calculated by matching a set of reliable support points and triangulating between them. A maximum a posterior approach refines the disparities, and disparity segments below a size of 50 pixels are removed. A two-stage correlation method for stereoscopic depth estimation (SNCC) [20] is a block matching stereo approach with a summed normalized cross-correlation (SNCC) measure. Then, the standard post-processed is applied, including error island removal (region growing), hole-filling and median filtering. Real-time correlation-based stereo vision with reduced border errors (Cens5), Hirschmiller et al. [21] is

based on a correlation with five, partly overlapping windows on Census-transformed images using the Hamming distance as the matching cost. MAP disparity estimation using hidden Markov trees (TMAP) [22] is a message passing on minimum spanning trees to acquire the maximum a posteriori disparity estimates. Hirschmiller [23] is the OpenCV's "semi-global block matching" method (SGBM2). The matching cost is the sum of absolute differences over small windows. Aggregation is performed by dynamic programming along paths in only 5 of 8 directions. The iterative guided filter (IGF) [27] is a stereo matching algorithm based on edge preserving filter at cost aggregation (SAD and Census transform) and image segmentation at disparity refinement stage, and for morphological processing of stereoscopic images (MPSV) [28], the computation of the sparse disparity maps is achieved by means of a 3D diffusion of the costs contained in the disparity space volume. The watershed segmentations of the left and right views control the diffusion process, and valid measurements are obtained by cross-checking.

Frameworks such as [18–21] are focused on real-time applications and are based on discriminating disparity estimations on some areas, often considering similarity measures with low computational cost, based on sets of previously determined rules. Such frameworks are more prone to errors with light, noise or exposure changes. Thus, the disparity refinement and smoothing stage are necessary for increasing a disparity map quality. Another group of frameworks [17,22] rely on the stereo matching stage with full computational cost, implementing exhaustive convolutions and discriminating in later stages the incorrect disparities. Some instances of these frameworks are capable of real-time applications based on parallelization [18], specifically with GPU computation. In such cases, there is a continuous development of GPUs with more computational capability. In this instance, we opted to design a general framework and then work toward a better adaptation in both, CPU and GPU. Thus, we proposed two implementations, one focused onto smaller datasets (CPU) and one for large-scale data (GPU). We used MATLAB to design and prototype the proposed framework. Then, an optimization was performed via migration to C++, toward the creation of MEX files (MATLAB executable files) in the case of CPU, with the C++ linear algebra library Armadillo and multiple thread processing capabilities using the application program interface (API) OpenMP. Also a CUDA migration for the GPU implementation is presented.

## 3 Proposed method

The proposed method consists of the followings stages: a color-space conversion from RGB to CIEL\*a\*b is applied to the stereo pair, and then, in the L-channel, the Census transform is applied to the stereo dataset, following a stereo

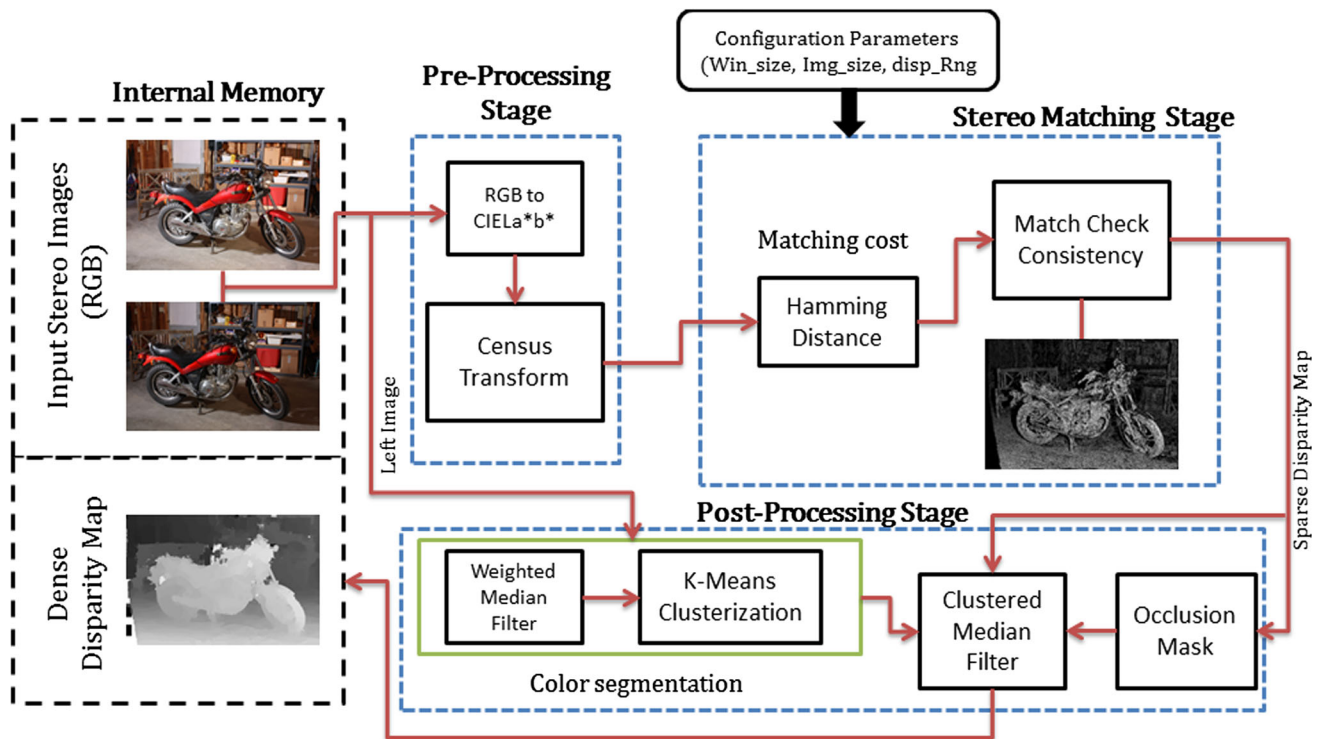


Fig. 1 Diagram for proposed method

matching performed under the Hamming distance measure (matching cost). Also, a match-check consistency is applied for occlusion labeling in order to obtain the sparse disparity map. In the next stage, a post-processing stage is performed on the estimated DM. A weighted median filter is applied to smooth objects and surfaces in the  $a^*b^*$  channels, reducing the number of iterations required for a color segmentation via  $k$ -means. On the other hand, an occlusion mask is obtained from the sparse disparity map. Finally, a clustered median filter is employed, where the ROI is determined by the occlusion mask and the color segmentation, estimating and smoothing surfaces for a dense disparity map; the block diagram of the method is presented in Fig. 1.

### 3.1 Pre-processing stage

The stereo pair is converted to CIELa\*b\* color-space, where the L-channels are employed for the stereo matching stage and the channels  $a^*b^*$  help to smooth the sparse disparity map along the color segmentation section in the post-processing stage. Next, the Census transform is applied. The Census transform can be described as a local binary pattern processing method, in which every pixel is related to a binary value of length  $N$ , where  $N$  is the number of local elements [15]. The transform is obtained using the Heaviside function of the neighborhood (1). The function determines a number, which is represented as a binary vector suitable for Hamming distance computation.

$$H(g(i+d_1, j+d_2) - g(i, j))$$

$$H(z) = \begin{cases} 0 \rightarrow z < 0 \\ 1 \rightarrow z \geq 1, \end{cases} \tag{1}$$

where the Census transform is applied only in horizontal, vertical and diagonal axes across each local submatrix, thus reducing the length of the binary value for each pixel

### 3.2 Stereo matching stage

For the stereo matching stage, the Hamming distance (2) was selected, which is the number of coefficients in which two binary vectors differ from each other [16].

$$HD(x, y) = \sum_i |x_i - y_i|, \tag{2}$$

The stereo matching process is done obtaining the Hamming distance between every binary vector along the left and right Census-transformed images, where Hamming distance is the number of coefficients, in which two binary vectors differ from each other to estimate the left disparity map, the right image is shifted along the  $X$ -axis until the disparity range is covered, and the winner-takes-all (WTA) approach is performed to select a disparity value for each pixel. This procedure is performed twice, one time for each image in the stereo pair, according to the resolution of the stereo pair. We propose that the 480p width resolution or lower should

be estimated with a minimum kernel size of  $9 \times 9$ . Two DM are obtained and are compared to evaluate consistency and occlusion labeling via the stereo match consistency process; the result is a sparse DM.

### 3.3 Post-processing stage

In the post-processing stage, the weighted median filter is applied to the left image, in order to speed up the  $k$ -means clustering algorithm. The weighted median filter (WMF) is an operator that replaces the current pixel with the weighted median of neighboring pixels within a local window [24]. This method is edge preserving, which is a major task in stereo matching smoothing techniques. The color segmentation data are added as a parameter for the clustered median filter, along with the sparse DM and the occlusion mask. The occlusion mask is a binary dataset containing 0s for every occlusion or mismatched disparity value and 1s for every valid disparity in the sparse DM. After application of the CMF algorithm, the dense DM is finally obtained.

The WMF is very similar to bilateral filtering, where for each local window, Gaussian weights are computed, and the median is subsequently estimated based on those Gaussian weights:

$$O[i, j] * g[i, j] = \sum_{i=1}^n \sum_{j=1}^m O[i, j] \text{sort}(g[x_w - i, y_w - j]), \quad (3)$$

where  $g$  is the image to be processed and  $O$  is a kernel suitable for order filtering. The sort process is performed to establish the weighted median value, where weights are obtained using the Gaussian difference:

$$w = e^{-\left(\frac{f(x,y)-f(\xi)}{\sigma_r}\right)^2}, \quad (4)$$

where  $f(\xi)$  is the local window from the reference image (the borders in this image will be preserved),  $f(x, y)$  is a local window from the sparse DM defined at the  $x, y$  position as center, and  $\sigma_r$  controls the weight between two pixels. For handling occlusions, a binary mask is defined according to the occlusions in the sparse DM. This approach enables the local avoidance of all the labeled values in the computation of each local weighted median, permitting reduction in the areas with occlusions while smoothing the surfaces, obtaining the dense disparity map at the end of the process.

The goal of the  $k$ -means algorithm that is applied in clustering stage (Fig. 1) is to minimize the objective function  $J$  that measures the closeness between data points and cluster centers:

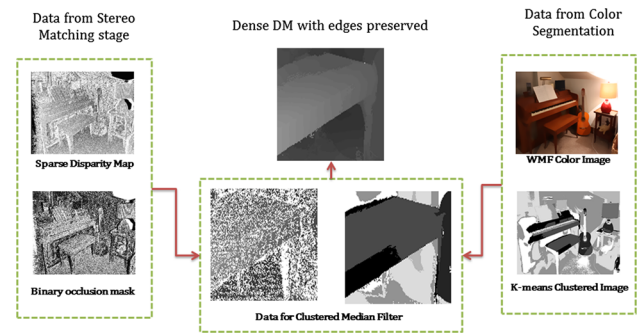


Fig. 2 Diagram for post-processing stage

$$\min_{c_j} (J) = \sum_{j=1}^K \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2, \quad (5)$$

When  $K$ -means is applied after the WMF, the number of iterations required to converge to the solution for clustering is reduced, speeding up the color segmentation process and avoiding scattered points that are present after only the  $k$ -means segmentation, due to noise and textures in color images. Once the color segmentation is completed, the data, along with the binary mask occlusion and the sparse DM, are processed with the CMF. The block diagram for this process is presented in Fig. 2. In the CMF, disparity values are median filtered with a main condition. The median value is determined for every pixel, but only using data belonging to the same color cluster. Data of a different color cluster are discarded, and the median value obtained is utilized to occlusion handling. Finally, a padding array is fixed for interpolation of the borders of the image, in order to obtain a full DM with the same resolution as the stereo pair images.

### 3.4 Hardware implementations

The proposed method was implemented on a CPU Intel i7-3770k 3.4GHz with 16 Gb RAM. This implementation aims for a fast and efficient process with low processing times. With this goal in mind, the image resolutions and disparity ranges are limited to a quarter size. The process was implemented on the IDE MATLAB R2015a, where the main stages were programmed using the Armadillo C++ linear algebra library and the API OpenMP for multithread compatibility. The operating system was Windows 7 (64 bit), and the compiler for the MEX files was TDM-GCC with optimization level O3. The GPU implementation was created using a NVIDIA Quadro K2000D with 384 CUDA cores and a memory interface of 2Gb GDDR5.

For the quarter-size simulation results, a multithread similarity measure is proposed, via open multiprocessing (OpenMP). OpenMP provides an easy method for threading applications based on shared memory architecture.



The parallelization is achieved by adding directives in C/C++ programs. These directives support the distribution of autonomous tasks over the available processor cores.

For the GPU implementation, the data are sent to the GPU, and the Hamming distance is performed, where the block size is fixed at  $32 \times 32$  threads, and the grid size is estimated along the image dimensions. The data are gathered and returned to the CPU once the stereo matching process is complete. The send and gather operations are performed only once for a sparse DM estimation. The implementation was performed via MATLAB R2015a, and the CUDA algorithm was compiled into a CUDA kernel for MATLAB usage.

Let explain novel framework in detail. The proposed approach employs a customized Census transform and Hamming distance as similarity measure in the stereo matching stage, where a WTA (winner-takes-all) approach is used to determine the best possible disparity. The post-processing stage uses clustered median filtering. Here, the weighted median filtering is applied to the original left image. Next, a  $K$ -means color segmentation is applied in the CIELa\*b\* color-space. Next, the clustering data and the occlusion mask data are utilized in the clustered median filter to create a dense disparity map.

## 4 Experimental results

### 4.1 Data

To evaluate the performance of the proposed method, the most recent training set to date was taken from the Middlebury Stereo Vision Web site [26]. The results of the proposed method are compared to the results from other methods, as tested on the same dataset, to demonstrate the robustness of our approach to different image content and properties.

### 4.2 Quality criteria

Two quantitative metrics, similarity structural index measure (SSIM) and percentage of bad matching pixels ( $B$ ), are used. The results of these metrics allow us to justify the efficiency of the proposed framework and to compare the processing times using the same metrics. To compute the selected metrics, the ground truths (GT) obtained from the Middlebury Stereo Vision Web site for each stereo pair and the DM estimate are employed.

The SSIM index is based on the fact that natural images are highly structured; also, it is known that SSIM is more consistent with human perception than other metrics, such as PSNR or MAE. The SSIM contains three similarity metrics: luminance ( $l$ ), contrast ( $c$ ) and structure ( $s$ ). The SSIM metric values [25] are defined as:

$$\text{SSIM}(x, y) = [l(x, y)] \cdot [c(x, y)] \cdot [s(x, y)], \quad (6)$$

The  $B$  values are calculated as shown:

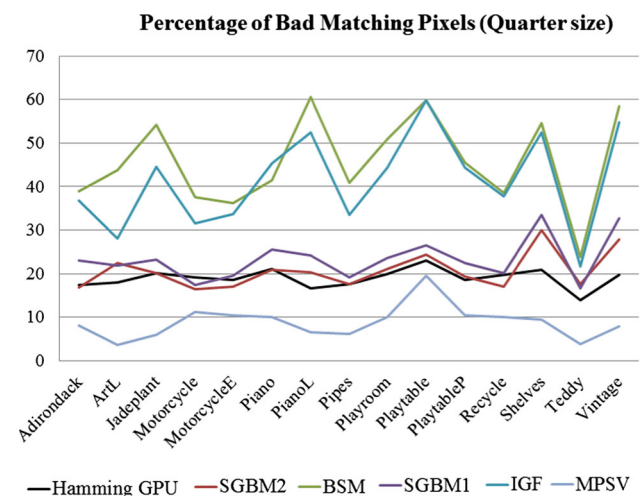
$$B = \frac{1}{N} \sum_{(x,y)} (|DM_I(x, y) - DM_{GT}(x, y)| > \delta_d), \quad (7)$$

where  $N$  is the total number of pixels in an image or frame,  $DM_I$  is the estimated disparity, and  $DM_{GT}$  is the GT.  $\delta$  is the error threshold difference for each pixel evaluated. There are two results for  $B$ : sparse and dense. Sparse is the DM estimated without any smothering stage, with occlusions marked. Dense map is the post-processed DM. Sparse results are shown only when these ones are available for other proposals.

### 4.3 Evaluation results

The DM results for the Middlebury datasets are presented in this section. Our proposed method is first compared in quarter size with other DM sparse schemes. Next, the half-size sparse frameworks are compared during evaluation. In this comparison, only  $B$  measure and processing times are presented. Finally, the half-size dense DM is presented for the GPU implementations. For these results, the SSIM measure is presented for every framework.

The  $B$  measure results show a robust performance despite different resolutions, image sizes, exposures and light conditions. For a better pixel-by-pixel comparison, a low  $\delta$  threshold value was chosen ( $\delta = 0.5$  for quarter size, and  $\delta = 1$  for half size); this allows for the understanding of the improvement made to the DM estimation in the sparse stage only. Our approach is able to maintain quality over the number of pixels estimated in this stage, in



**Fig. 3**  $B$  results for sparse disparity map estimation for the quarter-size dataset

order to achieve acceptable  $B$  and to reduce the computational cost for an exhaustive approach at the stereo matching stage for quarter-size dataset. In Fig. 3, our designed proposal can be distinguished by the better performance at the stereo stage, particularly in different light conditions (ArtL, PianoL), which is a downside for most of the state-of-the-art methods. Bricola et al. [28] showed better results, but the processing time is almost 100 times greater than our proposal for the same dataset size and sparse DM estimation.

In the half-size dataset, the best result is achieved with our proposal using Hamming distance for sparse DM estimation, as shown in Fig. 4, where the scenarios with different light conditions (ArtL, PianoL), exposure (MotorcycleE) or misalignment (Playable) are better estimated employing the

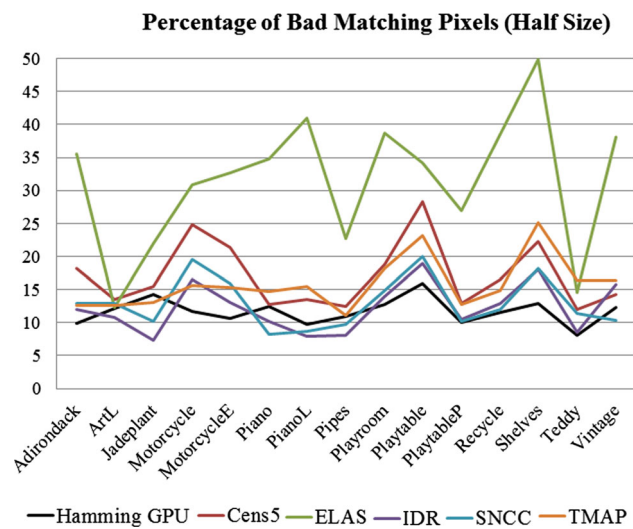


Fig. 4  $B$  results for sparse disparity map estimation for the half-size dataset

proposed method. Hamming distance as similarity measure is often overlooked in fast implementations because of the amount of processing time required, specifically with the Census transform and the binary vector length, where it is possible to short the binary values performing the Census operator only in selected directions instead of using a full kernel. However, the use of GPU implementation and customized kernel for Census transform can compensate this drawback. Subjective results are shown in Fig. 5 where the comparison with state-of-the-art methods can be viewed, demonstrating reduced number of mismatched disparities (black areas); this is a crucial task to obtain better results in the smoothing stage, allowing to reconstruct finer details and edges. Non-occluded pixels visible mask, provided by Middlebury dataset, shows pixels that are visible in both views of the stereo pair marked as white pixels, then occluded and invalid pixels are marked as gray pixels, and mismatched disparities related to black pixels are marked also in Fig. 5.

The dense disparity map results for the stereo pairs that presents different light conditions, exposure or misalignment are presented in Table 1, where the SSIM,  $B$  and processing times with megapixel ratio are listed. For quantitative comparison,  $B$  is evaluated only in discontinuity and non-occluded regions, to show effectiveness of the proposal in border regions. For the dense disparity map estimation in the half-size dataset (Fig. 6), the weighted median filtering was applied with a  $15 \times 15$  kernel, and  $\sigma_r = 12$ . The Gaussian weights were obtained from the L-channel of the left image, and a binary mask that is based on as the occlusion labels for each image. The results have shown that our proposal is competitive among other better frameworks, despite the simple but efficient design of novel framework. The SSIM value was obtained using an  $11 \times 11$  Gaussian kernel and defining the dynamic range as the disparity range for every dense

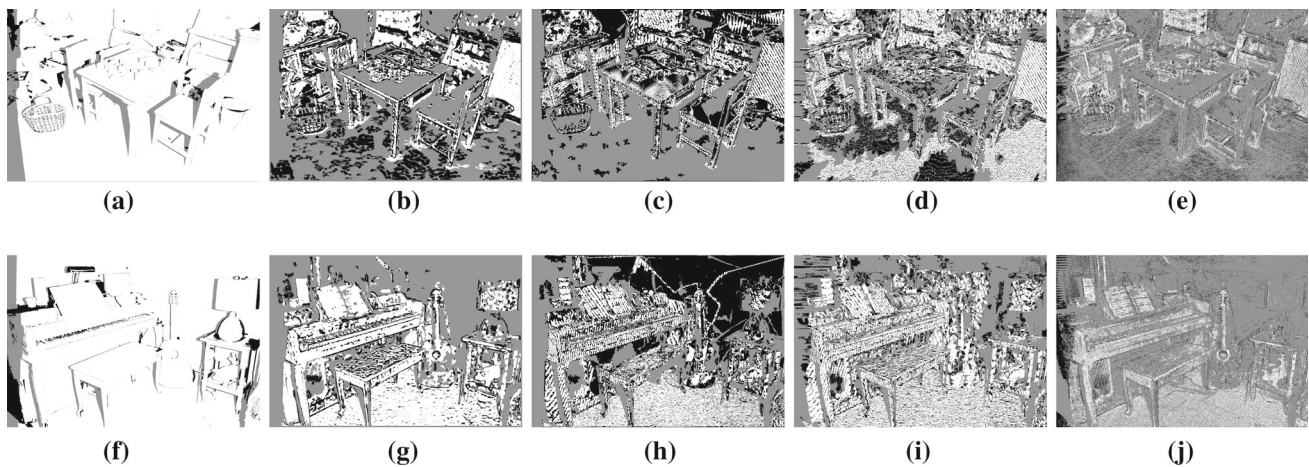


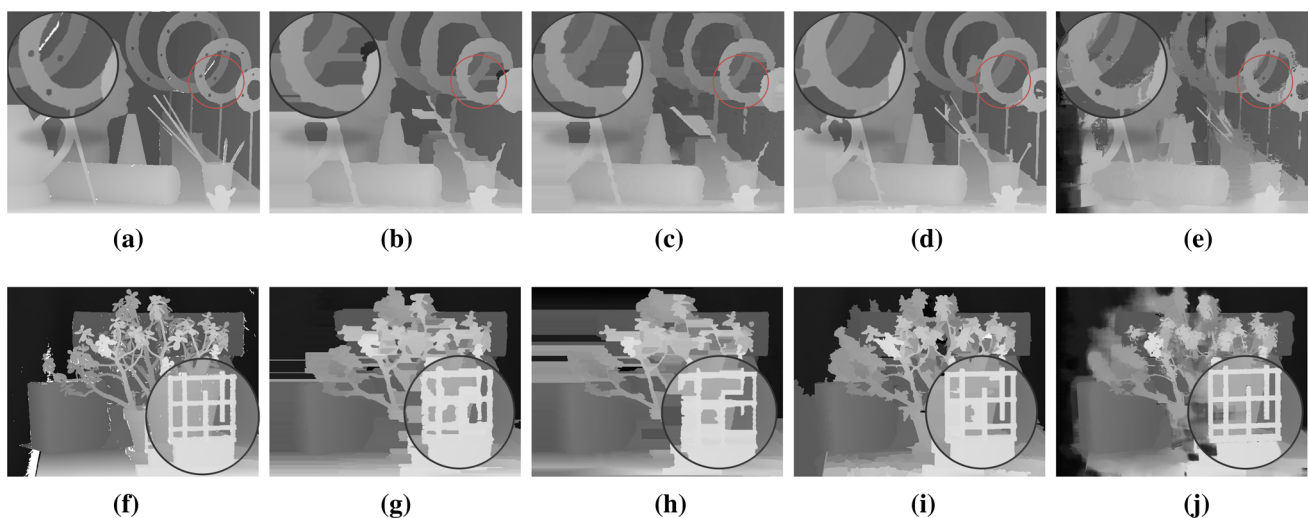
Fig. 5 Error image results for the sparse disparity map estimation for the half-size dataset from Middlebury dataset [26], Playable stereo pair: a non-occluded pixels visible mask, b Cens5, c ELAS, d TMAP and

e Hamming measure. PianoL stereo pair: f non-occluded pixels visible mask, g Cens5, h ELAS, i TMAP and j Hamming measure

**Table 1** Average  $B$  for discontinuity regions

$B$	Prop. GPU	Cens5	ELAS	TMAP	SNCC
ArtL	31.21	32.05	28.39	29.91	30.02
MotorcycleE	38.75	37.43	48.86	33.09	35.09
PianoL	48.33	47.59	63.86	45.82	42.14
Playtable	57.02	65.30	67.01	56.64	61.94
Average $B$	43.82	45.59	52.03	47.95	50.21
SSIM	Prop. GPU	Cens5	ELAS	TMAP	SNCC
ArtL	0.8351	0.8382	0.8268	0.8837	0.8464
MotorcycleE	0.8411	0.8328	0.8385	0.8592	0.8455
PianoL	0.7497	0.8058	0.7853	0.8000	0.8253
Playtable	0.7362	0.6924	0.7482	0.8184	0.6343
Average SSIM	0.7905	0.7923	0.7997	0.8403	0.7879
Average time/Mp	31.59	1.08	0.48	1264	0.78

SSIM and processing times per megapixel results for dense disparity map estimation and processing times for the half-size dataset ( $\delta = 2$ )



**Fig. 6** Experimental results for the dense disparity map estimation for the half-size dataset from Middlebury dataset [26], ArtL stereo pair: **a** ground truth, **b** Cens5, **c** ELAS, **d** TMAP and **e** Hamming measure. Jadeplant stereo pair: **f** ground truth, **g** Cens5, **h** ELAS, **i** TMAP and **j** Hamming measure

DM. Subjective results for dense disparity map estimations can be seen in Fig. 6, where the CMF demonstrates better performance in maintaining object features. One can see this observing small details and edges in the zoomed image where finer details are reconstructed with visibly better perception. Time processing values per megapixel demonstrate higher values, but they are significantly better, even compared to iterative frameworks, which run longer.

## 5 Conclusions

In this paper, a comparison between sparse and dense DM estimation frameworks was presented, where the principal

contributions are: Firstly, the proposed technique uses a customized Census transform scheme with GPU implementation. Secondly, we use Hamming measure in the stereo matching stage implemented in GPU; the results have shown that this appears to demonstrate better performance for the sparse DM estimation than state-of-the-art methods especially in reconstructing finer details and edges. Finally, DM estimation using a novel clustered median filtering was designed and parallelized on their CPU implementation via OpenMP. Experimental results have shown that Hamming distance implemented on GPU is effective in estimation of sparse DMs presenting higher quality than state-of-the-art methods (especially in reconstructing finer details and edges) and the dense DMs obtained using clustered median filter



which are in both terms, objective and subjective, adequate for a depth reconstruction of stereo pair images, maintaining a parallel design to be more competitive in processing times.

**Acknowledgements** The authors would like to thank Instituto Politécnico Nacional, Consejo Nacional de Ciencia y Tecnología (Project 220347) and Secretaría de Ciencia, Tecnología e Innovación del D.F. (Mexico) for their support in accomplishing this work.

## References

1. Penza, V., Ortiz, J., Mattos, L.S., Forgione, A., De Momi, E.: Dense soft tissue 3D reconstruction refined with super-pixel segmentation for robotic abdominal surgery. *Int. J. Comput. Assist. Radiol. Surg.* **11**, 197–206 (2016)
2. Wang, C., Palomar, R., Cheikh, F.A.: Stereo video analysis for instrument tracking in image-guided surgery. In: 2014 5th European Workshop on Visual Information Processing (EUVIP 2014), Paris (2014)
3. Balicki, M., Sznitman, R., Meisner, E., Taylor, R., Hager, G.: Vision-based proximity detection in retinal surgery. *IEEE Trans. Biomed. Eng.* **59**, 2291–2301 (2012)
4. Ozgunalp, U., Ai, X., Dahnoun, N.: Stereo vision-based road estimation assisted by efficient planar patch calculation. *Signal, Image Video Process.* **10**, 1127–1134 (2016)
5. Orfanidis, G., Tefas, A., Nikolaidis, N., Pitas, I.: Signal processing: image communication facial image clustering in stereoscopic videos using double spectral analysis. *Signal Process. Image Commun.* **33**, 86–105 (2015)
6. Huitron, V.G., Ponomaryov, V.: Robust approach for disparity map estimation based on multilevel decomposition. *IEEE Lat. Am. Trans.* **14**, 2968–2973 (2016)
7. El Jaafari, I., El Ansari, M., Koutti, L.: Fast edge-based stereo matching approach for road applications. *Signal Image Video Process.* **11**, 267–274 (2017). doi:[10.1007/s11760-016-0932-3](https://doi.org/10.1007/s11760-016-0932-3)
8. Ramos-Diaz, E., Kravchenko, V., Ponomaryov, V.: Efficient 2D to 3D video conversion implemented on DSP. *EURASIP J. Adv. Signal Process.* **2011**, 110 (2011)
9. Gonzalez-Huitron, V., Ramos-Diaz, E., Kravchenko, V., Ponomaryov, V.: 2D to 3D conversion based on disparity map estimation. In: *Lecture Notes in Computer Science*, vol. 8827, pp. 982–989 (2014)
10. Kim, C.G.: Parallel SAD for fast dense disparity map using a shared memory programming. In: Park, H.J.J., Jong, Barolli, L., Xhafa, F., Jeong, H.-Y. (eds.) *Information technology convergence: security, robotics, automations and communication*, pp. 1055–1060. Springer, Netherlands (2013)
11. Yang, Q., Ji, P., Li, D., Yao, S., Zhang, M.: Fast stereo matching using adaptive guided filtering. *Image Vis. Comput.* **32**, 202–211 (2014)
12. Zhao, Y., Taubin, G.: Real-time stereo on GPGPU using progressive multi-resolution adaptive windows. *Image Vis. Comput.* **29**, 420–432 (2011)
13. Ramos-Diaz, E., Gonzalez-Huitron, V., Ponomaryov, V.I., Hernandez-Fragoso, A.: 2D to 3D conversion implemented in different hardware. In: *Real-Time Image and Video Processing 2015* (2015)
14. Makkithaya, U.R.K., Karunakar, A.K.: Anchor-diagonal-based shape adaptive local support region for efficient stereo matching. *Signal Image Video Process.* **9**, 893–901 (2013)
15. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: *Proceedings of European Conference on Computer Vision*, pp. 151–158 (1994)
16. Humenberger, M., Zinner, C., Weber, M., Kubinger, W., Vincze, M.: A fast stereo matching algorithm suitable for embedded real-time systems. *Comput. Vis. Image Underst.* **114**, 1180–1202 (2010)
17. Zhang, K., Li, J., Li, Y., Hu, W., Sun, L., Yang, S.: Binary stereo matching. In: 2012 21st International Conference on Pattern Recognition (ICPR), Tsukuba, pp. 356–359 (2012)
18. Kowalczyk, J., Psota, E.T., Prez, L.C.: Real-time Temporal Stereo Matching using Iterative Adaptive Support Weights. In: 2013 IEEE International Conference on Electro/Information Technology (EIT), Rapid City, SD (2013)
19. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: *Asian Conference on Computer Vision (ACCV)* (2010)
20. Einecke, N., Eggert, J.: A two-stage correlation method for stereoscopic depth estimation. In: 2010 International Conference on Digital Image Computing: Techniques and Applications, pp. 227–234 (2010)
21. Hirschmiller, H., Innocent, P.R., Garibaldi, J.: Real-time correlation-based stereo vision with reduced border errors. *Int. J. Comput. Vis.* **47**, 229–246 (2002)
22. Psota, E.T., Kowalczyk, J., Mittek, M., Perez, L.C.: MAP disparity estimation using hidden Markov trees. In: 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, Santiago, pp. 2219–2227 (2015)
23. Hirschmiller, H.: Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**, 328–341 (2008)
24. Zhang, Q., Xu, L., Jia, J.: 100+ times faster weighted median filter (WMF). In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2830–2837 (2014)
25. Wang, Z., Bovik, A.C., Sheikh, H.R., Member, S., Simoncelli, E.P., Member, S.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, 600–612 (2004)
26. Scharstein, D., Hirschmiller, H., Kitajima, Y., Krathwohl, G., Nesić, N., Wang, X., Westling, P.: High-resolution stereo datasets with subpixel-accurate ground truth. In: *German Conference on Pattern Recognition (GCPR 2014)*, Munster (2014)
27. Hamzah, R.A., Ibrahim, H., Hassan, A.H.A.: Stereo matching algorithm based on per pixel difference adjustment, iterative guided filter and graph segmentation. *J. Vis. Commun. Image Represent.* **42**, 145–160 (2017). doi:[10.1016/j.jvcir.2016.11.016](https://doi.org/10.1016/j.jvcir.2016.11.016)
28. Bricola, J.-C., Bilodeau, M., Beucher, S.: A top-down approach to the estimation of depth maps driven by morphological segmentations. In: Benediktsson, J.A., Chanussot, J., Najman, L., Talbot, H. (eds.) *Mathematical Morphology and Its Applications to Signal and Image Processing: 12th International Symposium, ISMM 2015, Reykjavik, May 27–29, 2015. Proceedings*, pp. 122–133. Springer, Cham (2015)