STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVI**, Number 3, 2011

STEREOMATCHING USING RADIOMETRIC INVARIANT MEASURES

ALINA MIRON[1,2], SAMIA AINOUZ[1], ALEXANDRINA ROGOZAN[1], ABDELAZIZ BENSRHAIR[1], AND HORIA F. POP[2]

Abstract. Stereo Matching can provide important information about the environment for many autonomous systems. Usually one of the demands for these systems is real-time or almost real-time running. This is why we studied how to adapt two stereo matching measures to this demand, by providing an implementation on a GPU . For stereo matching algorithm a simple local approach based on window aggregation was chosen, and as pixels measures were used two radiometric insensitive measures: Census transform and Daisy features. Also, in response to the on going debate of whether to use color information in stereo matching, experiments were performed using different color spaces for both of the radiometric insensitive measures considered.

1. Introduction

Stereo Vision is a technique for obtaining 3D information from a given scene by using a pair of cameras. There exist many applications for this technique especially in the field of automated systems. For example in the field of robotics or autonomous driving systems, stereo vision can be used to extract relative position of objects or to separate occluding image components, such as one person in front of another one.

In this paper the focus is on the stereo matching step of the stereo vision flow. An important challenge for finding reliable correspondences in a pair of stereo images of a scene is the fact that images could present radiometric distortions and great baseline distance (if the cameras have a significant distance of each other). The stereo matching algorithms could be divided in local and global approaches. Local approaches rely mostly on a support window (typically square), which can be implemented efficiently using a sliding window technique [7]. In the global approaches typically the problem is expressed as

Received by the editors: April 10, 2011.

2010 *Mathematics Subject Classification.* 68T45.

1998 *CR Categories and Descriptors.* I.4.8 [**Computing Methodologies**]: Image Processing and Computer Vision – *Scene Analysis*.

Key words and phrases. stereo vision, census transform, daisy features, GPU.

r

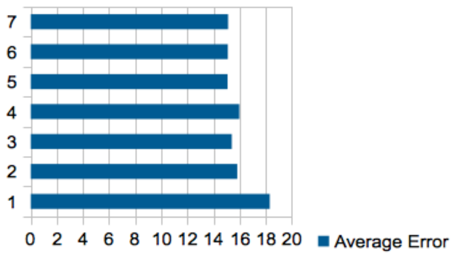


FIGURE 1. Average Error for different masks configuration. (1) Dense mask, (2) Star, (3) Check 2, (4) Check with fortified core, (5) Check 3, (6) Check 4, (7) Check 5.

a energy function that must be optimized. This can be done using dynamic programming [2], graph-cuts [8] or belief-propagation [5].

As is often the case for real time applications, there must be found a trade-off between computational cost and the disparity map precision. That is why in practice local methods for stereo matching are used. For these, there exist different pixels descriptors. The simpler would be the color intensity, and probably the most complex are the local region descriptors (like SIFT). For our experiments we chose to use the following pixels descriptors: census transform [11] and daisy features [10].

The rest of the abstract is organized as follows: Section 2 presents the results of the implementation on GPU and experiments performed with different color spaces, and in Section 3 are presented the conclusion and further work.

2. Experiments

2.1. **GPU computation.**

2.1.1. *Census Transform.* The first algorithm which would be suitable for a GPU implementation is the Census Transform. This is because the values computed for each pixel are independent, thus the algorithm is highly parallelizable. If on the CPU implementation the algorithm runs in 0.5 sec, in the GPU implementation the algorithm runs almost 35 times faster, in 0.015 sec.[1] Like we previously stated , for the census transform a check like census mask was used where it is taken into consideration each 3^{rd} pixel at each 3^{rd} row.

The size of the census window was chosen in an empirical way to be 15. This is because a window size of 15 will give in a check like configuration a total

[1]All the experiments were performed on laptop having a CPU with 2.4 GHz Intel Core 2 Duo, and a GPU NVIDIA GeForce 320M with VRAM 256MB

TABLE 1. Comparison between computation time between CPU and GPU per major operations for Daisy features

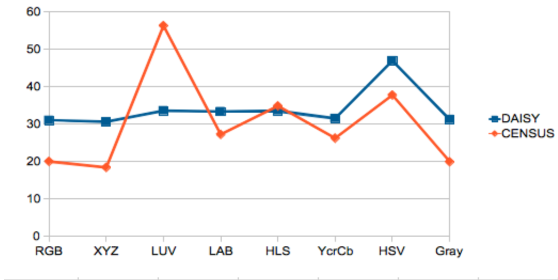
Operation	CPU time (sec)	GPU time (sec)
Image loading from disk	0.01	0.01
Copy Image from CPU RAM to GPU RAM	-	0.045
Daisy Set Parameters	0.000150	0.000184
Daisy Initialize Descriptor	0.36	0.08
Compute Daisy features	0.532	0.22
TOTAL time	0.9	0.35

of 25 relevant pixels. This bit string of 25 elements can be represented as a simple number of integer datatype. In figure 1 are shown comparative average errors obtained for different configuration masks. As it can be seen keeping the same number of relevant pixels but using a check like configuration better results are obtained. The dataset used to run this experiment was formed from 4 pairs of stereo images taken from the Middlebury database [9]. The distance between two bit strings is a simple Hamming distance that can be implemented efficiently in the form of a bitwise exclusive or.

2.1.2. *Daisy Features.* According to the authors of [10] for an image of 800×600 the computation time for Daisy features is around 3.8sec. In our implementation of Daisy features the running time for an image of 450×375 is around 1sec. This is still not fast enough for a real time application and that is why we decided to try a GPU implementation of these features. At the time of writing of this paper there didn't exist a GPU implementation for daisy features. For the GPU implementation we started implementing all the basic operations. One of the basic operations that is used intensively in the daisy algorithm is a 1D gaussian convolution of the image. The convolution time in our CPU implementation is about $4.6 * 10^{-3}$. After the GPU implementation the new time of the convolution operation is about $5 * 10^{-5}$, therefore the GPU implementation of the convolution runs almost a hundred times faster than the CPU counterpart.

In table 1 can be saw a comparison between the computation time of the CPU and GPU implementations for Daisy features. Even if the computation of Daisy features is still not fast enough for a real-time running, with more optimization (more use of the shared memory on GPU, loading the image directly in the GPU RAM) the running time could be decreased even more to reach near-real time levels.

FIGURE 2. Mean Error across different color spaces



2.2. **Color Information for radiometric invariant measures.** In stereo matching, there exist different opinions in what concerns the use or not of color information. There exist different studies that prove the performance of color-based stereo in comparison with grey-scale matching [3], [7] . According to [6] by using color information instead of gray values the improvement obtained is around 25%. In another study [4], the authors evaluate different matching measures that are insensitive to radiometric distortions and they stated that color information didn't help in stereo matching. Other study [1] claims the fact that color information is less robust with the radiometric insensitive measures, leading to a degradation, so they state the fact that color should not be used in stereo matching. This is why we tried to reproduce the experiments made by [1] with census measure, but with two important differences: the first one is the fact that there is no energy minimization function used (because this could interfere with the actual results obtained), and the second difference is that the results are validated on another radiometric invariant measure, that provided by the daisy features.

In figure 2 can be saw the results of stereo matching experiments using Census measure and Daisy features. There were used for the experiments eight different color spaces: *RGB, XYZ, LUV, LAB, HLS, YCrCb, HSV, Grayscale.* The first remark for this figure is that this is not intended to be a comparison between Census transform and Daisy features, because they address different problems: Daisy can be used for wide baseline stereo matching while Census is used for standard stereo matching. This is why for the set of images from Middlebury database used (four images with small baseline) the Census measure obtains better results than Daisy. Instead of making a comparison between the two measures we are more interested on the color space trend comparison. For the Census transform the results obtained when using for pixel color comparison a simple sum over the channels are: the smallest mean error was obtained when using the XYZ color space: *18.4%*, followed by RGB and Grayscale with *19.9%*. The interesting fact is that the same order is preserved when using as measure Daisy features: for XYX color space the error is *30.58%*, followed by RGB with *30.96%* and Gray with *31.21%*. Therefore, color didn't lead to a degradation of results when using in combination with radiometric invariant features, but in fact it improved the results.

The presented results were obtained by using a simple sum over the channels. This fact may not do justice to all the color spaces and that is why an experiment was performed with optimized weights of the three channels using genetic algorithm (there were not optimized the weights for RGB color space). The results improved for almost all the color spaces but the best results were still obtained by XYZ color space. Also because the initial database of testing images was small, we performed the same experiments on the TNO/MARS PRESCAN [2] but the results remained consistent.

3. Conclusion

The power of parallel computation given by the GPUs can be used to reach real-time or near real-time running for some stereo matching algorithms. Here, there were presented two radiometric insensitive measures that are suitable for a GPU implementation. Even if in numerical error the Census transform performs better than Daisy features for the dataset of images used, the latter could represent a solution for a general approach for both standard and wide base line stereo matching. In what concerns the use of color for radiometric insensitive measures, based on our results we can conclude that color information taken as a simple sum over channels from XYZ color space performs better than Grayscale.Some could argue that the improvement brought by the color information is not significant enough to compensate a greater running time needed for the transformation in that color space. But stereo vision does not represent the final goal: color information may also be useful for further object detection. As future work we plan to integrate a way for fast image segmentation because this can provide important information in low texture areas (that appear widely for example in a traffic situation).

References

1. M. Bleyer and S. Chambon, *Does Color Really Help in Dense Stereo Matching?*, Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), vol. 2010, 2010.
2. M. Bleyer and M. Gelautz, *Simple but effective tree structures for dynamic programming-based stereo matching*, International Conference on Computer Vision Theory and Applications (VISAPP), Citeseer, 2008.

[2]http://staff.science.uva.nl/ wvdmark/StereoVisionMarsPrescan/

3. S. Chambon and A. Crouzil, *Colour correlation-based matching*, International Journal of Robotics and Automation **20** (2005), no. 2, 78–85.
4. H. Hirschmuller and D. Scharstein, *Evaluation of cost functions for stereo matching*, Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.
5. A. Klaus, M. Sormann, and K. Karner, *Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure*, Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol. 3, IEEE, 2006, pp. 15–18.
6. R. Klette, A. Koschan, K. Schluns, and V. Rodehorst, *Surface reconstruction based on visual information*, TR95/6, July (1995).
7. K. Muhlmann, D. Maier, J. Hesser, and R. Manner, *Calculating dense disparity maps from color stereo images, an efficient implementation*, International Journal of Computer Vision **47** (2002), no. 1, 79–88.
8. N. Papadakis and V. Caselles, *Multi-label depth estimation for graph cuts stereo problems*, Journal of Mathematical Imaging and Vision (2010), 1–13.
9. D. Scharstein and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*, International journal of computer vision **47** (2002), no. 1, 7–42.
10. E. Tola, V. Lepetit, and P. Fua, *DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo*, IEEE transactions on pattern analysis and machine intelligence (2009), 815–830.
11. R. Zabih and J. Woodfill, *Non-parametric local transforms for computing visual correspondence*, Computer VisionECCV'94 (1994), 151–158.

(1) Laboratoire d'Informatique, de Traitement de l'Information et des Systemes , INSA Rouen, Avenue de l'Universite, 76800, Saint-Etienne-du-Rouvray, France, web:www.litislab.eu,

(2) Babeş-Bolyai University, Faculty of Mathematics and Computer Science,, 1, Mihail Kogalniceanu St., 400084, Cluj-Napoca, Romania, web:www.cs.ubbcluj.ro,

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

E-mail address: `samia.ainouz@insa-rouen.fr`

E-mail address: `alexandrina.rogozan@insa-rouen.fr`

E-mail address: `abdelaziz.bensrhair@insa-rouen.fr`

E-mail address: `hfpop@cs.ubbcluj.ro`