Algorithmic Considerations for Real-Time Stereo Vision Applications

Kristian Ambrosch, Christian Zinner, and Wilfried Kubinger
Austrian Research Centers GmbH
A-1220 Vienna, Austria
{kristian.ambrosch, christian.zinner, wilfried.kubinger}@arcs.ac.at

Abstract

Real-time stereo vision is a very resource intensive application, requiring a high computational performance. Therefore, we analyze the well known Census Transform not only for an increase in accuracy, but also for a reduction in complexity. We propose a novel approach, using the Modified Census Transform on the intensity as well as the gradient images, that can be efficiently combined with a sparse computation. Our evaluation of this approach on the images of the Middlebury stereo ranking shows that it allows scaling the algorithm's complexity down by a factor of 5.8, while still being more accurate than the original transform.

1 Introduction

Stereo vision is a popular technique for maintaining three-dimensional images in robotic applications [11, 12, 16]. It is flexible, small in size and since it is entirely passive, it does not affect its neighborhood. Stereo vision uses two cameras side by side, measuring the displacement of the image objects caused by the cameras' different viewpoints. This displacement is called disparity and directly correlated to the distance of the corresponding objects.

Great efforts have been made in improving the quality of the resulting disparity map, leading to highly accurate algorithms such as graph cuts, belief propagation, and segmentation [8, 9, 19]. However, the algorithm most frequently used in real-time stereo vision applications is the Sum of Absolute Differences (SAD) [1] on gray scale images [6, 5, 17]. Recently, the more robust Census Transform [20] has been used [7, 10, 18], but at the costs of higher computational effort.

The lack of state of the art stereo vision algorithms in real-time applications is due to the fact that stereo vision algorithms are optimized to gain a high accuracy, while the computational complexity is usually neglected. Then again, most real-time applications simply implement algorithms with a manageable complexity, focusing on the optimization of the implementation. However, only little effort has been made to optimize the algorithms themselves to reduce their computational complexity.

Thus, we analyze the Census Transform and propose a novel version of the algorithm that not only exhibits an improved accuracy, but also a highly reduced computational complexity. Furthermore, we discuss how the algorithm's complexity can be efficiently scaled to fit systems with high as well as low computational performance.

2 The Census Transform

The Census Transform [20] is an area-based stereo matching algorithm with high robustness to illumination variations [2]. The algorithm first transforms the images, before the matching costs of each disparity level are calculated. Therefore, it uses a comparison function ξ, which is used to compare the center pixel value i_1 of a block of pixels N, with the other pixels' intensity values i_2.

$$\xi(i_1, i_2) = \begin{cases} 1 & | \quad i_1 > i_2 \\ 0 & | \quad i_1 \leq i_2 \end{cases} \tag{1}$$

Its result, 1 if the center pixel is larger, and otherwise 0, is then concatenated ($\bigotimes$) to a bit-vector.

Thus, the transformation function T_{Census} is defined as

$$T_{Census}(I, x, y, s_t) = \bigotimes_{[n,m]\epsilon N} \xi[I(x, y), I(n, m)] \tag{2}$$

where, I is either the primary or the secondary intensity image delivered by the stereo cameras and s_t is the size of the transformed block N.

For the calculation of the matching costs of each disparity level d, the cost function C_{Census} is defined as the Hamming distance over the bit-vectors.

$$C_{Census}(t1_{x,y}, t2_{x+d,y}) = hdist(t1_{x,y}, t2_{x+d,y}) \tag{3}$$

For a higher accuracy of the algorithm, the calculation of the matching costs can be followed by a further aggregation [13].

3 Intensity and Gradient-Based Census Transform

For the Census Transform a higher accuracy can be achieved by increasing the block size. However, an increased block size also smoothes the images, resulting in noticeable image blur for extensively large block sizes. To increase the accuracy of the Census Transform in a different way than increasing the block size, it is necessary to extend the processed information. For extending the processed information, we integrated the gradient value and its direction into the transform. However, the computation of the gradient value and direction must have a low complexity and fit signal-processor- as well as hardware-based implementations. While the absolute gradient values in x and y

directions can be computed easily using the Sobel operator, the calculation of the direction's angle is known to be computationally rather expensive. Thus, we simply expanded the Census Transform to be processed over the intensity image and the absolute value of the gradient in x and y direction. This way, we incorporate information about the direction of the gradient, without having to calculate its exact angle.

Even if the computation of the gradient value using the Sobel operator incorporates some basic image smoothing [4], it still results in a gain in image noise. Furthermore, image edges in the intensity image result in a saturation of the pixels in the gradient images. The Census Transform is not able to cope with image blocks, where the center pixel is saturated, because in this specific case the output bit-vector is always at its maximum value, regardless of the texture in the block. Thus, incorporating the gradient images using the original Census Transform does not lead do an increase in accuracy.

However, Froeba and Ernst proposed the Modified Census Transform (MCT) [3], which uses the mean value over the whole pixel block instead of the center pixel value. The original purpose of this was to maintain an additional bit for the center pixel value in the bit-vector for face detection applications. Though we were not able to measure an improved accuracy by this additional bit-value for the application of stereo vision on intensity images, it has an additional advantage for the gradient images: It delivers reliable results even for blocks with a saturated center pixel.

Thus, we defined our approach as the MCT over the intensity as well as the gradient images in x (I_x) and y (I_y) direction (I/G$_{xy}$ MCT) as shown in equation 4.

$$T_{I/G_{xy}MCT}(I,x,y,s_t) = \bigotimes_{[n,m]\epsilon N} \xi[\overline{I(x,y)}, I(n,m)] \bigotimes_{[n,m]\epsilon N} \xi[\overline{I_x(x,y)}, I_x(n,m)] \bigotimes_{[n,m]\epsilon N} \xi[\overline{I_y(x,y)}, I_y(n,m)] \quad (4)$$

The calculation of the matching costs is performed on the enlarged bit-vector in the same fashion as described in equation 3.

4 Sparse Computation

Incorporating the gradient in x and y direction not only extends the information processed by the algorithm, it also extends its complexity by a factor of 3. To reduce the complexity of the algorithm we chose to reduce the image resolution of the pixel blocks, processed by the I/G$_{xy}$ MCT. The common approach for a reduction in resolution would be using gauss pyramids [4], which low-pass filter the images and therefore avoid aliasing effects, caused by the reduced sampling rate. However, the filtering of high frequencies also reduces accuracy of the matching algorithm, since they contain essential information for the exact localization of the matched image objects. Thus, we are simply under-sampling the images and therefore intentionally approving the appearance of aliasing effects during the matching procedure.

The kind of mask used for the under-sampling is not just a choice of resulting accuracy, it is also highly depending on the processing platform. While hardware-based implementations, using FPGAs or ASICs, might be very flexible in the choice of mask, processor-based systems can have a considerable advantage, if they do not have to process all image lines in the block and therefore can optimize their memory access.

Thus, we present four different kinds of masks used for the under-sampling: Sequentially picking every n^{th} pixel, a raster mask picking every $\frac{n^{th}}{2}$ pixel in horizontal and vertical direction, as well as picking either every n^{th} line or column, and call n the sparse factor. Figure 1 depicts these mask variations for a sparse factor $n = 2$ for all masks but the raster mask, where the minimum $n = 4$ is presented.

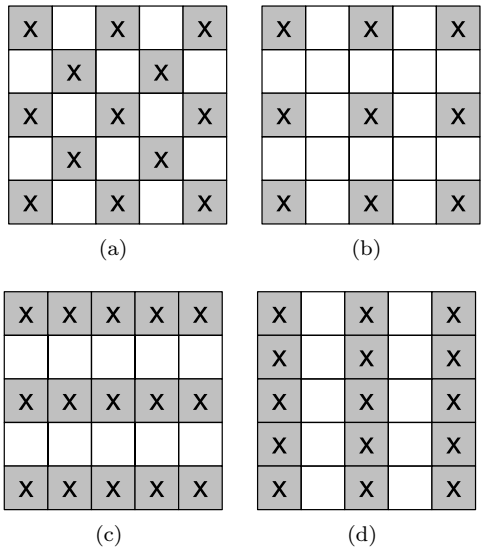


Figure 1: Mask variations for sparse computation: (a) sequential, $n = 2$; (b) raster, $n = 4$; (c) lines, $n = 2$; (d) columns, $n = 2$.

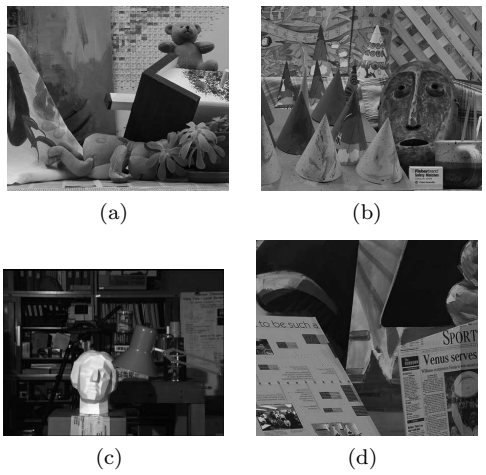
5 Experimental Evaluation

5.1 Test Configuration

For the experimental evaluation of our algorithm we used the gray-scaled Teddy, Cones, Venus and Tsukuba images from the Middlebury dataset [13, 14] as depicted in figure 2, measuring the disparity maps' average correct matches within a deviation of 0.5 pixels.

For the algorithms, the calculation of the matching costs was followed by a 3×3 aggregation, before the best match was selected using the Winner Takes All (WTA) algorithm. Furthermore, we refined the results with sub-pixel resolution by parabola fitting [15], always using the same sub-pixel resolution as given by the Middlebury dataset. However, since our analysis focuses on the accuracy of the stereo matching algorithm depending on its complexity, we avoided further post-processing steps, even if this would further improve the results. Thus, they are not competitive to the ones of the Middlebury stereo evaluation ranking.

For the comparison of the algorithms' complexity, we focused at the number of Hamming distance bit comparisons, i.e., the complexity of the matching cost computation. While the images are transformed only once for the whole computation, the computation of the matching costs has to be performed d times, where d is the number of different disparity levels computed. Thus, the computational complexity of the Sobel operator and the additional transforms are negligible, when compared to the matching cost computation.



(a) (b) (c) (d)

Figure 2: Middlebury dataset: (a) Teddy; (b) Cones; (c) Tsukuba; (d) Venus.

5.2 Results and Comparison

Figure 3 depicts the correct matches vs. the number of Hamming distance bit comparisons to be performed per transformed block for the Census Transform using block sizes reaching from 3×3 to 19×19 and the I/G$_{xy}$ MCT with constant block size 11×11. A horizontal line at 72.19% correct matches outlines the Census Transform's accuracy at block size 11×11. Here, the I/G$_{xy}$ MCT uses a rastered sparsing with sparse factors reaching from 1 to 6. As can be seen in the diagram, the sparse factor for the I/G$_{xy}$ MCT has a far smaller influence on the algorithm's accuracy than the reduction in block size for the Census Transform. Thus, it is possible to sparse the I/G$_{xy}$ MCT until its complexity is far less than the Census Transform's and still be more accurate. In this specific case, the I/G$_{xy}$ MCT can be sparsed until it requires just 27 Hamming distance bits per 11×11 block, and with 73.4% correct matches be 1.4% more accurate than the original Census Transform, having 121 calculations at the very same block size.

To reveal the impact of the different kinds of masks for the sparse computation, we present their results for the I/G$_{xy}$ MCT using block size 11×11 in figure 4. Again, we outlined the Census Transform's accuracy at the same block size by an additional horizontal line. While the raster mask's accuracy shows a logarithmic curve, the sequential mask shows a fall-off at sparse factors that are close to the dimension of the block, i.e., 11. At these sparse factors, the sequential mask results in evaluating nearly one column only, while factors that are not close to the block dimension are better distributed over the block, leading to more accurate results.

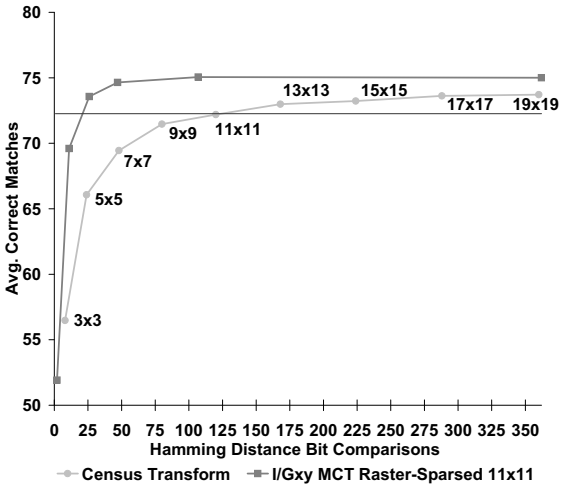


Figure 3: Accuracy vs. complexity for Census Transform block sizes 3×3 to 19×19 and the sparsed I/G$_{xy}$ MCT at block size 11×11.

Where the sequential and raster masks lead to the same algorithm complexity the accuracy is also nearly equal. However, the sequential mask allows for a better scaling of the algorithm's complexity. This way, it is possible to reduce the I/G$_{xy}$ MCT's complexity down to 21 Hamming distance bit comparisons, which is an reduction factor of 5.8 compared to the Census Transform's 121 calculations. At this complexity, the I/G$_{xy}$ MCT delivers 72.8% correct matches, which is still a bit more than the Census Transform at this block size.

Even if a reduced complexity is not required for the desired application, the I/G$_{xy}$ MCT increases the accuracy to 75.1%, reducing the incorrect matches by 10.3% at the same complexity, when using a sequential mask with a sparse factor of 3.

For systems with low performance, as can be found in low cost applications, it will be noticeable that the I/G$_{xy}$ MCT has an accuracy of 67.8% even when sequentially sparsed by a factor of 41. Here, only 9 bits are required for the Hamming distance, the same number as for the Census Transform with block size 3×3. The column and line sparse masks show a significantly decreased accuracy when compared to the raster and sequential masks. However, especially the line sparse mask might be an interesting approach for systems that have to focus on an optimized memory access and therefore keep the number of lines required for the computation as low as possible.

6 Conclusions

The sparse computation of the I/G$_{xy}$ MCT not only allows for an increase in accuracy at the same computational complexity. It also allows to scale the complexity of the stereo matching algorithm with a minimum loss in quality to fit systems with high as well as low computational performance.

For the reduction of complexity we proposed the sequential and raster masks as the most promising approaches, while the column and line masks will be of interest only for implementations with specific requirements.

Using our novel approach, the performance of a Cen-

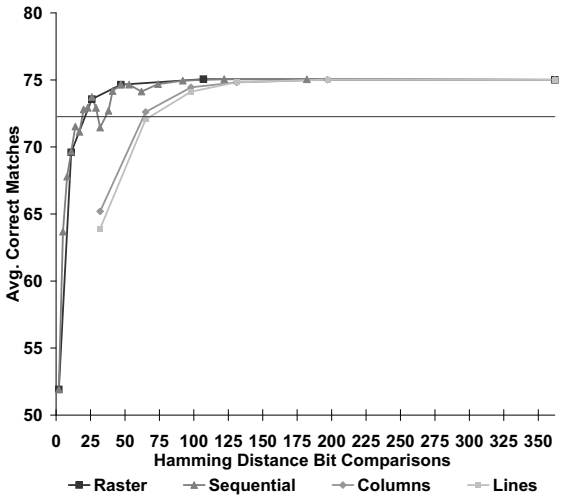


Figure 4: Accuracy vs. complexity for the sparsing masks of the I/G$_{xy}$ MCT.

sus Transform based stereo matching system could be increased 5.8 times, while still having a slightly improved accuracy. This demonstrates the necessity for real-time stereo matching applications to optimize not only the implementation of the algorithm, but also the algorithm itself.

7 Future Work

The real-time implementation of the proposed algorithm on a DSP-based as well as an FPGA-based system is ongoing work. For the DSP-based system we will focus on using a raster mask for the sparse computation. However, for the FPGA-based system the sequential mask will be the more promising approach.

8 Acknowledgements

The research leading to these results has received funding from the European Community's Sixth Framework Programme (FP6/2003-2006) under grant agreement n° FP6-2006-IST-6-045350 (robots@home).

References

[1] Jasmine Banks, Mohammed Bennamoun, and Peter Corke. Non-parametric techniques for fast and robust stereo matching. In *Proceedings of the IEEE Conference on Speech and Image Technologies for Computing and Telecommunications*, 1997.

[2] Boguslaw Cyganek. Comparison of Non parametric Transformations and Bit Vector Matching for Stereo Correlation. *Lecture Notes in Computer Science*, 3322, 2004.

[3] Bernhard Froeba and Andreas Ernst. Face detection with the modified census transform. In *Proceedings of the Sixth IEEE Conference on Automatic Face and Gesture Recognition*, 2004.

[4] R. C. Gonzalez and R. E. Woods. *Digital Image Processing, Second Edition.* Pearson Education International, 2002.

[5] Yunde Jia, Mingxiang Li, Luping An, and Xiaoxun Zhang. Autonomous navigation of a miniature mobile robot using real-time trinocular stereo machine. In *Proceedings of the IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, 2003.

[6] Ali E. Kayaalp and James L. Eckman. Near real-time stereo range detection using a pipeline architecture. *IEEE Transactions on Systems, Man and Cybernetics*, 20:1461–1469, 1990.

[7] Bahador Khaleghi, Siddhant Ahuja, and Jonathan Wu. An Improved Real Time Miniaturized Embedded Stereo Vision System (MESVS-II). In *Proceedings of the 2008 Conference on Computer Vision and Pattern Recoginition Workshops*, 2008.

[8] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure. In *Proceedings of the 18th International Conference on Pattern Recognition*, 2006.

[9] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the International Conference on Computer Vision*, 2001.

[10] Michael Kuhn, Stephan Moser, Oliver Isler, Frank K. Gurkaynak, Andreas Burg, Norbert Felber, Hubert Kaeslin, and Wolfgang Fichtner. Efficient ASIC implementation of a real time depth mapping stereo vision system. In *Proceedings of the 46th IEEE International Midwest Symposium on Circuits and Systems*, 2004.

[11] Larry Matthies, Mark Maimone, Andrew Johnson, Yang Cheng, Reg Willson, Carlos Villalpando, Steve Goldberg, Andres Huertas, Andrew Stein, and Anelia Angelova. Computer Vision on Mars. *International Journal of Computer Vision*, 75(1):67–92, 2007.

[12] David Meger, Per-Erik Forsseen, Kevin Lai, Scott Helmer, Sancho McCann, Tristram Southey, Matthew Baumann, James J. Little, and David G. Lowe. Curious george: An attentive semantic robot. *Robotics and Autonomous Systems*, 56(6):503–511, 2008.

[13] Daniel Scharstein and Richard Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1–3):7–42, 2002.

[14] Daniel Scharstein and Richard Szeliski. High-Accuracy Stereo Depth Maps Using Structured Light. In *Proceedings of the 2003 Conference on Computer Vision and Pattern Recoginition*, 2003.

[15] Masao Shimizu and Masatoshi Okutomi. Precise Subpixel Estimation on Area-Based Matching. In *Proceedings of the eight IEEE International Conference on Computer Vision*, 2003.

[16] Willian Travis, Robert Daily, David M. Bevly, Kevin Knoedler, Reinhold Behringer, Hannes Hemetsberger, Juergen Kogler, Wilfried Kubinger, and Alefs Bram. SciAutonics Auburn Engineering's Low Cost High Speed ATV for the 2005 DARPA Grand Challenge. *Journal of Field Robotics*, 23(8):579–597, 2006.

[17] Gooitzen van der Wal, Mike Hansen, and Mike Piacentino. The Acadia vision processor. In *Proceedings of the Fifth IEEE International Workshop on Computer Architectures for Machine Perception*, 2000.

[18] John Iseling Woodfill, Gaile Gordon, Dave Jurasek, Terrance Brown, and Ron Buck. The Tyzx DeepSea G2 Vision System, A Taskable, Embedded Stereo Camera. In *Proceedings of 2006 Conference on Computer Vision and Pattern Recoginition Workshops*, 2006.

[19] Qingxiong Yang, Liang Wang, Ruigang Yang, Henrik Stewenius, and David Nister. Stereo Matching with Color-Weighted Correlation, Hierachical Belief Propagation and Occlusion Handling. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recoginition*, 2006.

[20] Ramin Zabih and John Iseling Woodfill. Non- parametric Local Transforms for Computing Visual Correspondence. In *Proceedings of the 3rd European Conference on Computer Vision*, 1994.