

# Frame-rate Robust Stereo on a PCI Board

John Woodfill  
woodfill@interval.com

Interval Research Corporation

Brian Von Herzen  
brianvon@fpga.com  
Rapid Prototypes, Inc.

Ramin Zabih  
rdz@cs.cornell.edu  
Computer Science Department  
Cornell University

## Abstract

*Many vision applications would benefit from frame-rate, dense stereo. In this paper we present a robust algorithm implemented on a single board for a PC. The algorithm used is census transform correlation, which is well-suited for hardware implementation. Unlike existing frame-rate stereo methods, census transform correlation is robust to outliers, and does not assume that corresponding points have constant brightness. We provide an empirical comparison between our method and other frame-rate stereo algorithms using real data with ground truth. The stereo computation is performed on a reconfigurable computer built from programmable logic on a standard PCI card. The resulting system computes 32 stereo disparities on 320 by 240 pixel images at 46 frames per second. This is the highest performance stereo vision system described to date.*

## 1 Introduction

Many applications would benefit from frame-rate depth maps. Examples range from classic robotic tasks, such as grasping or navigating, to more recent projects like the Intelligent Room at MIT. Stereo is one obvious way to obtain depth maps, but the computational expense of stereo is considerable. Most existing frame-rate stereo systems produce sparse outputs [4, 12]. Dense stereo, however, would have two major advantages. First, individual measurements are inevitably error prone, so a dense output should yield more reliable applications. Second, a dense output can be used for segmentation, an important prerequisite for later processing. Segmentation depends upon accurate results near the borders of objects, which requires an algorithm that robustly withstands outliers.

In this paper we present a robust stereo algorithm and its hardware implementation on a single board for a PC. The algorithm is census transform correlation [15], which is very suitable for hardware implementation. Census transform correlation has several features that distinguish it from the algorithms in stereo

systems such as [7, 9, 11]. It is robust to outliers, and does not assume that corresponding points have constant brightness. The algorithm gives improved results near the borders of objects, as we will document in section 5.

We have implemented census transform correlation on a single PCI board called the PARTS engine [14]. The PARTS engine contains Field Programmable Gate Arrays (FPGAs) and memory, arranged in a rectangular torus. It attains over 3.3 billion RISC-equivalent operations per second. The system computes stereo with 32 disparities on 320 by 240 images at 46 frames per second.

We begin with a discussion of related work. In section 3 we briefly describe census transform correlation and list its most important properties. Section 4 details the architecture of our hardware implementation. Finally, in section 5 we provide an experimental comparison of census transform correlation against other high-performance stereo algorithms, using real data with ground truth.

## 2 Related work

Frame-rate stereo vision has been primarily performed on special-purpose hardware. Nishihara has developed FPGA-based stereo vision systems on custom boards using his Laplacian-of-Gaussian Sign-Correlation algorithm [12]. These systems have tended to compute selectable, sparse depth measurements. An early, dense, special-purpose stereo system was developed by Kayaalp and Eckman on Datacube boards [10]. Using the metric of points  $\times$  disparity-measures per second (PDS), this system achieved 6.7 million PDS. Moll [7] implemented a normalized correlation stereo vision algorithm on a PAM reconfigurable board, described below. Theoretical performance of this system was 7.4 million PDS.

Kanade *et al.* [9] at CMU describe a system believed to be the world's fastest in 1995. This system is composed of five VME boards, including three custom boards built up from discrete components as well as a

C40 DSP-array board, and a real-time OS board. It attains 30 hertz performance on 200 by 200 images using Laplacian-of-Gaussian (LOG) filtering followed by  $L_1$  correlation. It performs rectification, uses a multi-camera technique and performs 30 million PDS. By comparison, the reconfigurable PARTS engine computing census stereo depth as described below currently performs 113 million PDS.

Konolige [11] recently presented the SRI Small Vision Machine, which also implements LOG-filtered  $L_1$  correlation. The hardware used is a 33 MHz DSP. The current SVM computes 16 stereo disparities on 160 by 120 images at 6 frames per second, or 1.8 million PDS.

### 3 Census transform correspondence

Census transform correlation, which we introduced in [15], is an efficient, robust algorithm for visual correspondence. The algorithm relies on the local ordering of intensities. Like [2], it is closely related to non-parametric measures of association such as Kendall's  $\tau$  (see [16] for details). The algorithm has two steps: a transform step, in which we compute a bit string that summarizes local texture; and a correlation step.

The transform step replaces a given pixel  $P$  by the results of intensity comparisons. The comparisons are made between  $I(P)$  and  $I(P')$ , where  $P'$  ranges over pixels in a small neighborhood around  $P$ . If  $I(P) < I(P')$ , a 1 is stored in the bit string, otherwise a 0 is stored. The correlation step of the method computes correspondence based on the transformed images. Individual bit strings are compared using the Hamming distance (count of differing bits). Correlation is performed by summing Hamming distances over a small local region.

Census transform correlation has three key properties. First, it is an outlier tolerant method, and therefore performs well near discontinuities. Second, it does not require that corresponding points have constant brightness. Third, it is an efficient algorithm that is ideally suited for hardware implementation.

#### 3.1 Outlier tolerance

Census transform correlation is outlier tolerant, which is important near discontinuities. To illustrate this, consider a three-by-three region of an image whose intensities are

$$\begin{array}{ccc} 7 & 6 & 9 \\ 5 & 8 & 10 \\ 4 & 11 & A \end{array}$$

for some value  $0 \leq A < 256$ . Consider the effect on various parametric measures, computed at the center of this region, as  $A$  varies over its 256 possible values. Rounding to the nearest integer, the mean

of this region varies from 7 to 35, while the variance ranges from 12 to 6812. These parametric measures exhibit continuous variation over a substantial range as  $A$  changes. Non-parametric methods are more stable, however. For this example, the value of the census transform at the center pixel only changes by 1.

#### 3.2 The constant brightness assumption

It is natural to assume that corresponding points in the left and right images have constant brightness. This assumption is quite common in motion or stereo (e.g., [1, 8]), but it is often violated in practice. For example, Cox *et al.* point out in [5] that most of the images in the JISCT collection [3] violate the constant brightness assumption.

There are several reasons why the constant brightness assumption is invalid. Stereo uses two cameras, and cameras have different internal parameters. The difference between two cameras can be modeled as a linear transformation of intensities  $I = g \cdot I' + b$ , where we will call the multiplier  $g$  the *gain* and the offset  $b$  the *bias*. Bias can be removed by LOG filtering the images [13], although this obliterates some image detail. Other factors also cause corresponding points to have different intensities. For example, there are changes in illumination and viewing angle, which are extremely difficult to model for arbitrary scenes.

Census transform correlation does not make the constant brightness assumption. The census transform uses ordering information among intensities, which is not affected by camera parameters. When objects have sufficient texture, the ordering of intensities should be minimally influenced by changes in illumination or viewing angle.

### 4 Hardware implementation

Census transform correlation is well-suited to a hardware implementation. Most of the operations involve simple arithmetic operations such as comparison, where the quantities involved are small integers. In this section we provide a brief overview of our implementation of census transform correlation on a PCI card. The hardware used is the PARTS engine, which is a collection of FPGA chips arranged on a torus with local memory. More details concerning the hardware implementation can be found in [14].

#### 4.1 The PARTS engine

The PARTS engine combines a homogeneous array of field programmable logic chips (FPGAs) from Xilinx with tightly-coupled local SRAM to maximize memory bandwidth. An FPGA is a programmable integrated circuit that consists of a two-dimensional rectangular array of general-purpose logic elements with wires and switches interconnecting them. The wires

and switches are controlled by static memory residing on the device.

The logic elements usually consist of lookup tables with 4-5 bits of input and one or two bits of output. The lookup tables can be arranged to compute any Boolean function of four or five input variables. The lookup tables are also specified using static memory on the chip. The output of the lookup tables can pass through a synchronizing register before going through the interconnect network to another logic element in the array. All of these static registers are loaded at configuration time with a serial bitstream of data sent from a host computer or stored on a PROM memory chip. By configuring the logic of the FPGA into a pipeline, it is possible to approach the performance of a special-purpose chip using general-purpose hardware.

## 4.2 Census transform correlation on the PARTS board

Census transform correlation is well suited to the PARTS engine in that the algorithm is uniform and involves many simple, local operations. Mapping the census algorithm onto the PARTS engine involves evaluating the memory, the computation, and the inter-chip communication requirements. The goal of the design is to pipeline the whole stereo process. As each pair of pixels enters the system, the pipeline is stepped one cycle and a single disparity result is emitted. Thus two census transforms and 32 correlations are performed simultaneously as each pair of pixels enters. Communicating census vectors and correlation scores between FPGAs entails wide datapaths, while performing the census transform and census correlation require considerable memory bandwidth. Inter-chip communication is performed over two clock cycles to double the data width. Similarly, external SRAM is accessed and updated twice for each pair of pixels that enters the system.

At present, the PARTS engine is clocked with the 33Mhz PCI clock. If suitable data were available, the system could produce 16.5 million disparity results per second. This would correspond to 225 frames per second for 320 by 240 images, or for 32 disparities, 528 million PDS. Currently, live video sources are sent over the PCI bus with images coming from video frame-grabbers. In the future, digital cameras will send video data directly through the connectors on the top of the board. The disparity algorithm on the PARTS engine produces real-time video in which brightness corresponds to proximity of scene elements to the video cameras.

## 5 Comparative analysis

In this section we compare our system with other frame-rate stereo implementations, both in terms of throughput and accuracy. The throughput comparison uses millions of pixels-disparities per second. We also attempt to provide a comparison against other hardware architectures by estimating the amount of computation performed by our system in terms of RISC operations. The output comparison is done using real imagery with ground truth.

### 5.1 Comparative throughput

Figure 1 summarizes the performance of a number of extant frame-rate stereo systems. Note that some systems perform cross-checking, which detects inconsistency by comparing left-right and right-left matches. We have listed both the actual performance of our FPGA-based stereo system, running with our current I/O capability, and the maximum performance attainable ignoring these I/O limitations.

In order to characterize the amount of computation performed by the system, we use the notion of RISC-equivalent instructions, i.e., the number of operations that a RISC machine would perform in order to produce the same result. Loop overhead is ignored in this analysis. For a pair of images, a census transform must be performed at each pixel in each image, followed by a search over a fixed disparity window at each pixel. The census transform involves comparing the center pixel with a number of other pixels surrounding it. The disparity search for each pixel involves pairing the transformed census pixel for one image, and each of the transformed pixels for the other image and computing the minimum summed Hamming distance over the pairs. Our current implementation performs two 22-bit census transforms and a 32-disparity search at 46 frames per second, which is approximately 3.3 billion RISC equivalent operations per second.

### 5.2 Comparative accuracy on ground truth

We obtained an image pair from the University of Tsukuba Multiview Image Database for which approximate ground truth is known at every pixel. The image and the ground truth are shown in figure 3, along with the results from stereo methods with frame-rate implementations. Note that the background of the image is a large area which according to the ground truth lies at a single disparity. Within this area there are points of varying depth (for example, gaps above the books on the shelves).

Having ground truth allows a statistical analysis of algorithm performance. We have calculated the

System	Image size	Disparities	Cross-check	Frame rate	Merit
Kayaalp and Eckman '88	512 × 512	128	No	.2	6.7
Faugeras <i>et al.</i> '93	256 × 256	32	Yes	3.5	7.5
Kanade <i>et al.</i> '95	200 × 200	25	No	30	30
Konolige '97	160 × 120	16	Yes	6	1.8
Woodfill <i>et al.</i> '97, actual	320 × 240	32	Yes	46	113
Woodfill <i>et al.</i> '97, maximum	320 × 240	32	Yes	225	528

Figure 1: Comparative performance of hardware stereo systems. Figure of merit is millions of pixel-disparities per second.

number of correct answers that are obtained by various methods. To handle discretization errors in the ground truth, we declare an answer to be correct at a pixel if it lies within  $\pm 1$  disparity of the ground truth. We have also displayed the points in the image with incorrect results for the various algorithms.

The overall accuracy of the different methods lies within a very narrow range. However, for many tasks, such as recognition or robotic grasping, it is particularly important to obtain good results near discontinuities. Our method offers improved performance at such points. The advantage of census transform correlation is evident in the imagery, and is also supported by statistics. For example, the incorrect region at the border of the statue is much thinner in figure 3(d) than in (f) or (h). Similar effects occur near the borders of the other objects.

We have analyzed the performance of various methods on those pixels which (according to the ground truth) lie near a discontinuity. The results of this analysis are shown in figure 2. Unfortunately, census transform correlation performs poorly in the low-textured areas to the left of the camera, and in the upper right, making its overall performance roughly comparable to the other methods.

All three algorithms search a 16 disparity window. The census algorithm uses a 22 bit census transform and  $7 \times 7$  correlation. We obtained the parameters used by other hardware systems both by reading the literature and by consulting with the authors. The LOG filtered  $L_1$  results are generated using  $\sigma = 1.29$  with a correlation window of  $11 \times 11$ . A  $15 \times 15$  window is used for normalized correlation.

## 6 Conclusions

We have described a robust, frame-rate stereo vision system implemented on a single, general-purpose, reconfigurable board. We compared the accuracy of the algorithm with other frame-rate stereo algorithms using ground truth. Similarly, we have compared the

actual performance of the system with other frame-rate stereo systems.

We hope that frame-rate stereo vision will become an important component of vision systems. In [6] the frame-rate system has been successfully used as part of a system that performs person tracking in crowded and/or unknown environments using multi-modal integration. The system combines stereo, color, and face detection modules into a single robust system.

## Acknowledgements

We are grateful to Dr. Y. Ohta and Dr. Y. Nakamura for supplying the ground truth imagery from the University of Tsukuba Multiview Image Database. We also thank P. Rander and L. Moll for providing details about the CMU and INRIA frame-rate stereo systems.

## References

- [1] S. Barnard. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3(1):17–32, 1989.
- [2] Dinkar N. Bhat and Shree K. Nayar. Ordinal measures for visual correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 351–357, 1996.
- [3] R. Bolles, H. Baker, and M. Hannah. The JISCT stereo evaluation. In *DARPA Image Understanding Workshop*, pages 263–274, 1993.
- [4] David Coombs and Christopher Brown. Real-time smooth pursuit tracking for a moving binocular head. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1992.
- [5] Ingemar Cox, Sebastien Roy, and Sunita Hingorani. Dynamic histogram warping of image pairs for constant image brightness. In *IEEE International Conference on Image Processing*, 1995. Extended version available as an NEC technical report.

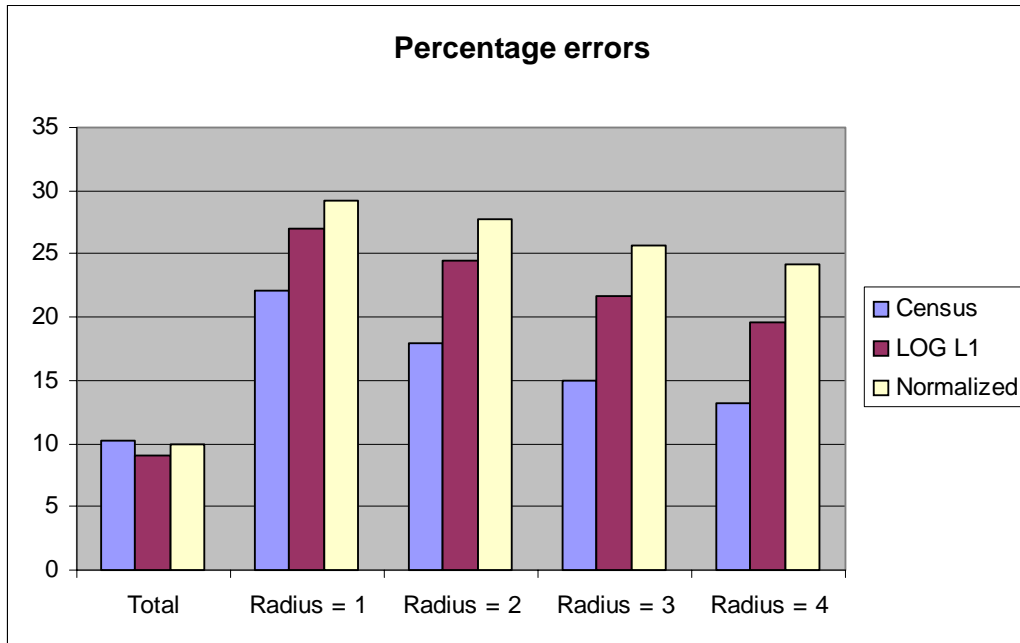


Figure 2: Algorithm performance on pixels within a given radius of a discontinuity.

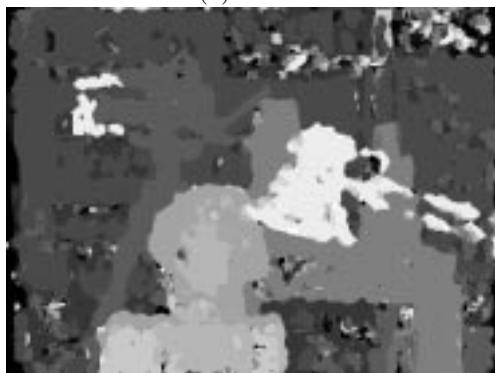
- [6] T. Darrell, G. Gordon, J. Woodfill, H. Baker, and M. Harville. Robust, real-time people tracking in open environments using integrated stereo, color, and face detection. In *ICCV Workshop on Visual Surveillance*, 1998.
- [7] O. Faugeras, B. Hotz, H. Mathieu, T. Vieville, Z Zhang, P. Fua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy. Real-time correlation-based stereo: algorithm, implementations and applications. Technical Report 2013, INRIA, August 1993.
- [8] Berthold Horn and Brian Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- [9] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 196–202, 1996.
- [10] A. Kayaalp and J. Eckman. A pipeline architecture for near real-time stereo range detection. In *Proceedings of SPIE Mobile Robotics III*, pages 279–286, 1988.
- [11] Kurt Konolige. Small vision systems: Hardware and implementation. In *International Symposium on Robotics Research*, October 1997.
- [12] H. K. Nishihara. Real-time stereo- and motion-based figure-ground discrimination and tracking using LOG sign-correlation. In *Proceedings of the Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers*, pages 95–100, 1993.
- [13] Masatoshi Okutomi and Takeo Kanade. A multiple baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.
- [14] John Woodfill and Brian von Herzen. Real-time stereo vision on the PARTS reconfigurable computer. In *IEEE Symposium on Custom Computing Machines*, 1997.
- [15] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *3rd European Conference on Computer Vision*, pages 151–158, 1994.
- [16] Ramin Zabih and John Woodfill. A non-parametric approach to visual correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997. Submitted for review. Available from [www.cs.cornell.edu/home/rdz](http://www.cs.cornell.edu/home/rdz).



(a) Scene



(b) Ground truth



(c) Census transform correlation



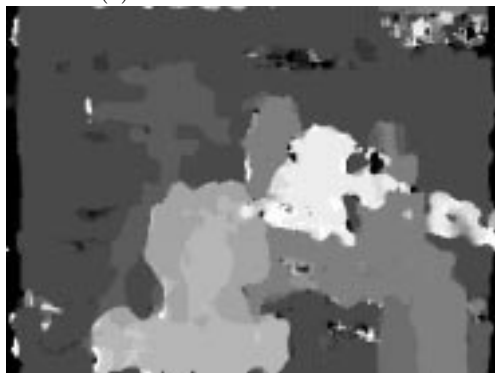
(d) Census errors



(e) Normalized correlation



(f) Normalized errors



(g) LOG-filtered  $L_1$



(h) LOG errors

Figure 3: Ground truth results