Small Vision Systems: Hardware and Implementation

Kurt Konolige
Artificial Intelligence Center, SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025
email: konolige@ai.sri.com

Abstract

Robotic systems are becoming smaller, lower power, and cheaper, enabling their application in areas not previously considered. This is true of vision systems as well. SRI's Small Vision Module (SVM) is a compact, inexpensive realtime device for computing dense stereo range images, which are a fundamental measurement supporting a wide range of computer vision applications. We describe hardware and software issues in the construction of the SVM, and survey implemented systems that use a similar area correlation algorithm on a variety of hardware.

1 Introduction

Realtime stereo analysis, until recently, has been implemented in large custom hardware arrays (Kanade 1996, Matthies 1995). But computational power and algorithmic advances have made it possible to do such analysis on single processors. At the same time, increased density, speed and programmability of floating-point gate arrays (FPGAs) make custom hardware a viable alternative. In this paper, we discuss the implementation of area-based stereo algorithms on microprocessors, and describe in detail one such implementation, the SRI Small Vision Module (SVM), which achieves realtime operation at low power in a small package. We also survey area-based implementations on microprocessors and FPGAs, comparing speed and efficiency.

In a final section, we briefly describe some experiments with the SVM related to mobile robotics and human-computer interaction. These experiments make use of realtime stereo to segment interesting objects from the background.

2 Area-correlation Stereo

Stereo analysis is the process of measuring range to an object based on a comparison of the object projection on two or more images. The fundamental problem in stereo analysis is finding corresponding elements between the images. Once the match is made, the range to the object can be computed using the image geometry.

Matching methods can be characterized as *local* or *global*. Local methods attempt to match small regions of one image to another based on intrinsic features of the region. Global methods supplement local methods by considering physical constraints such as surface continuity or base-of-support. Local methods can be further classified by whether they match discrete *features* among images, or correlate a small *area* patch (Barnard and Fischler 1982). Features are usually chosen to be lighting and viewpoint-independent, e.g., corners are a natural feature to use because the remain corners in almost all projections. Feature-based algorithms compensate for viewpoint changes and camera differences, and can produce rapid, robust matching. But they have the disadvantage of requiring perhaps expensive feature extraction, and yielding only sparse range results.

Area correlation compares small patches among images using correlation. The area size is a compromise, since small areas are more likely to be similar in images with different viewpoints, but larger areas increase the signal-to-noise ratio. In contrast to the feature-based method, area-based correlation produces dense results. Because area methods needn't compute features, and have an extremely regular algorithmic structure, they can have optimized implementations. The SVM and other systems discussed in this paper all use area correlation.

2.1 Area Correlation Method

The typical area correlation method has five steps (see Figure 3):

1. Geometry correction. In this step, distortions in the input images are corrected by warping into a "standard form".
2. Image transform. A local operator transforms each pixel in the grayscale image into a more appropriate form, e.g., normalizes it based on average local intensity.

3. Area correlation. This is the correlation step, where each small area is compared with other areas in its search window.
4. Extrema extraction. The extreme value of the correlation at each pixel is determined, yielding a *disparity image*: each pixel value is the disparity between left and right image patches at the best match.
5. Post-filtering. One or more filters clean up noise in the disparity image result.

In the rest of this section we describe each of these operations in more detail.

2.2 Epipolar Geometry

The fundamental geometry for stereo correlation is the epipolar curve, illustrated in Figure 1. Consider any point *p* in one image; the ray from *p* through the focal point intercepts objects in the scene at different depths. The projection of this ray in the second image is the epipolar curve associated with *p*. The geometric significance of this curve is that, for any object imaged at *p*, the corresponding point in the second image must lie on the epipolar curve. Thus, it suffices to search along this curve when doing stereo matching.

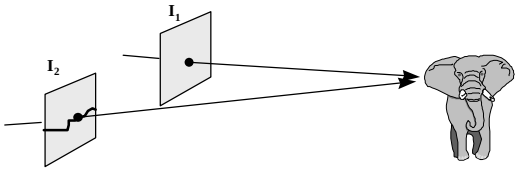


Figure 1 An epipolar curve is the image in I_2 of a ray projected from a point of I_1.

If the images are ideal projections of a pinhole camera, then the epipolar curves will be straight lines. For efficient processing, it is best to align epipolar lines in a regular pattern, usually along the scan lines of the camera. To accomplish this, the images must be in the same plane, with their scan lines and image centers horizontally aligned, and with the same focal length. Stereo images with this geometry are said to be in standard or scan-line form. In practice, it is difficult to achieve this physical geometry precisely, and in some cases vergence of the images is necessary to view close objects. However, if all internal and external camera parameters are known, then it is possible to warp each image so that it appears in standard form. In practice, it is difficult and time-consuming to completely calibrate a stereo setup, and repeated calibration necessitated by vibration or other mechanical disturbances is not practical. A compromise is to position the cameras in approximately the correct geometry, and to calibrate the internal camera parameters. The remaining offset parameters between the cameras can be dealt with by auto-calibration, discussed in Section 3.2.

2.3 Image Transforms and Correlation Measures

Correlation of image areas is disturbed by illumination, perspective, and imaging differences among images. Area correlation methods usually attempt to compensate by correlating not the raw intensity images, but some transform of the intensities. Table 1 lists some of the correlation measures found in the literature. There are three basic types of transforms:

1. Normalized intensities. Each of the intensities in a correlated area is normalized by the average intensity in the area.
2. Laplacian of gaussian (LOG). The laplacian measures directed edge intensities over some area smoothed by the gaussian. Typically the standard deviation of the gaussian is 1-2 pixels.
3. Nonparametric. These transforms are an attempt to deal with the problem of outliers, which tend to overwhelm the correlation measure, especially using a square difference.

Each of these transforms can be applied with different correlation measures. For example, LOG correlations include correspondence of zero crossings and their signs, and two measures of the magnitude: square and absolute difference (also called the *L1 norm*).

The census method computes a bit vector describing the local environment of a pixel, and the correlation measure is the Hamming distance between two vectors. This measure, as with other nonparametric measures, addresses the problem of outliers, which the square difference handles poorly. The L1 norm tends to weight outliers less than the square difference, and is more amenable to implementation on standard microprocessors, which usually do not have the bit-counting hardware necessary for computing Hamming distance efficiently.

There has been relatively little work comparing the quality of results from the different transforms. Partly this results from the difficulty of getting good ground truth test sets. Some recent work (Zabih and Woodfill 1997) comparing census and normalized intensities on synthesized and real images shows that census does better, in that it can

Normalized intensities square difference	Fua 1993
Laplacian of gaussian zero crossings sign square difference absolute difference	Marr and Poggio 1979 Nishihara 1984 Matthies 1995 Kanade 1996
Nonparametric rank census	Zabih and Woodfill 1994

Table 1 Correlation measures for area-based stereo

interpret more of the image correctly, especially at depth discontinuities. From the author's experience, the best quality results appear to come from absolute difference of the LOG, and from census.

Another technique for increasing the signal-to-noise ratio of matches is to use more than two images (Faugeras 1993). This technique can also overcome the problem of viewpoint occlusion, where the matching part of an object does not appear in the other image. The simple technique of adding the correlations between images at the same disparity seems to work well (Kanada 1996). Obviously, the computational expenditure for multiple images is greater than for two.

2.4 Filtering

Dense range images usually contain false matches that must be filtered, although this is less of a problem with multiple-image methods. Table 2 lists some of the post-filters that have been discussed in the literature. The correlation surface shape can be related to the uncertainty in the match. An interest operator gives high confidence to areas that are textured in intensity, since flat areas are subject to ambiguous matches. The left/right check looks for consistency in matching from a fixed left image region to a set of right image regions, and back again from the matched right region to a set of left regions. It is particularly useful at range discontinuities, where directional matching will yield different results.

Finally, a disparity image can be processed to give sub-pixel accuracy, by trying to locate the correlation peak between pixels. This increases the available range resolution without much additional work.

Correlation surface: peak width, peak height, number of peaks	Matthies 1993, Nishihara 1984
Mode filter	
Left/Right check	Fua 1993, Bolles and Woodfill 1993
Interest Operator	Moravec 1984
Interpolation	Nishihara 1984

Table 2 Range filtering operations

3 Algorithm and Implementation

Although area correlation is computationally expensive, good algorithm design coupled with current processors enable video rate performance for small frame sizes. In this section we describe the results of two optimized implementations of the same basic algorithm. The Small Vision Module (SVM), takes advantage of the instruction set and memory design of digital signal processors (DSPs) to achieve high performance at low power. The Small Vision System (SVS) is an implementation of the same algorithm in C on general-purpose microprocessors, and an optimized version for Pentium microprocessors that takes advantage of the MMX instruction set.

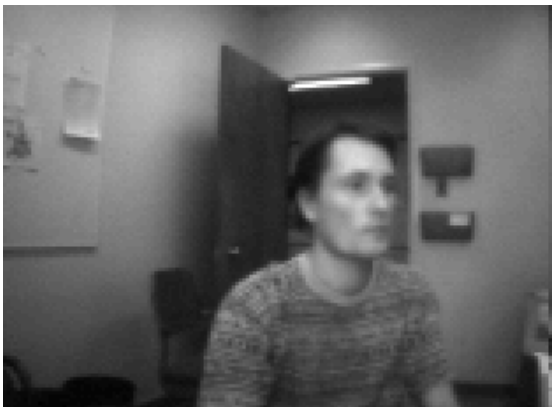
3.1 Algorithm

The algorithm we have implemented (Figure 3) has the following features:

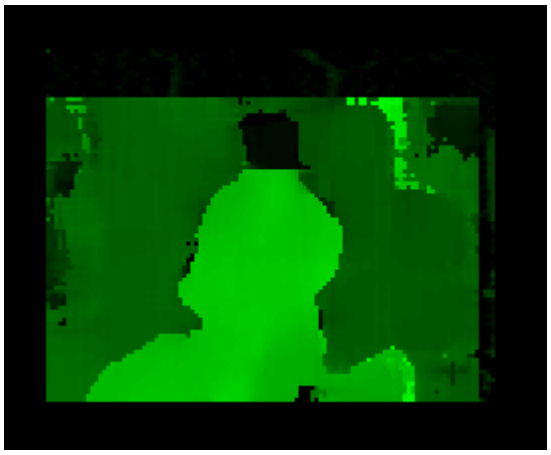
- LOG transform, L1 norm (absolute difference) correlation;
- Variable disparity search, e.g., 16, 24, or 32 pixels;
- Postfiltering with an interest operator and left/right check;
- x4 range interpolation.

The LOG transform and L1 norm were chosen because they give good quality results, and can be optimized on standard instruction sets available on DSPs and microprocessors. The census method is an alternative, but requires fast bit-counting, and is more suitable for custom hardware.

Figure 2(b) shows a typical disparity image produced by the algorithm. Higher disparities (closer objects) are indicated by brighter green (or white, if this paper is printed without color). There are 64 possible levels of disparity; in the figure, the closest disparities are around 40, while the furthest are about 5. Note the significant errors in the upper

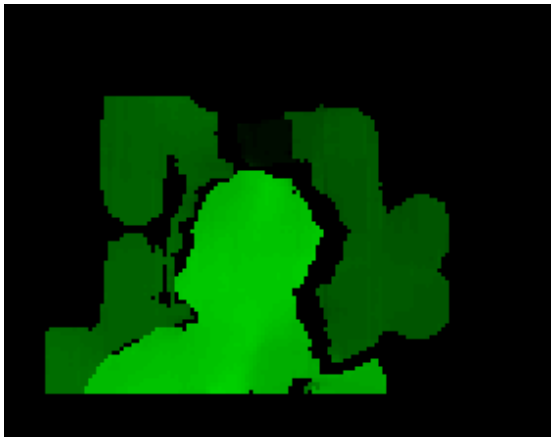


(a) Input grayscale image, one of a stereo pair



(b) Disparity image from area correlation

(c) Texture filter applied



(d) Left/right and texture filter applied

Figure 2 A grayscale input image and the disparity images from the SRI algorithm.

left and right portion of the image, where uniform areas make it hard to estimate the disparity.

In Figure 2(c), the interest operator is applied as a postfilter. Areas with insufficient texture are rejected as low confidence: they appear black in the picture. We've found that the best interest operator is one that measures texture in the direction of epipolar lines (Matthies 1993).

Although the interest operator requires a threshold, it's straightforward to set it based on noise present in the video input. Showing a blank gray area to the imagers produces an interest level related only to the video noise; the threshold is set slightly above that.

There are still errors in portions of the image with disparity discontinuities, such as the side of the subject's head. These errors are caused by overlapping the correlation window on areas with very different disparities. Application of a left/right check can eliminate these errors, as in Figure 2(d). The left/right check can be implemented efficiently by storing enough information when doing the original disparity correlation.

In practice, the combination of an interest operator and left/right check has proven to be the most effective at eliminating bad matches. Correlation surface checks, in our experience, do not add to the quality of the range image, and can be computationally expensive.

We have been careful in designing the algorithm to consider implementation issues, especially storage efficiency. Buffering of intermediate results, a critical part of the algorithm, is kept to a minimum by calculating incremental results for each new scanline, and by recalculating rather than storing some partial results. We have been especially careful about storage of correlation results, and require only space of size (number of

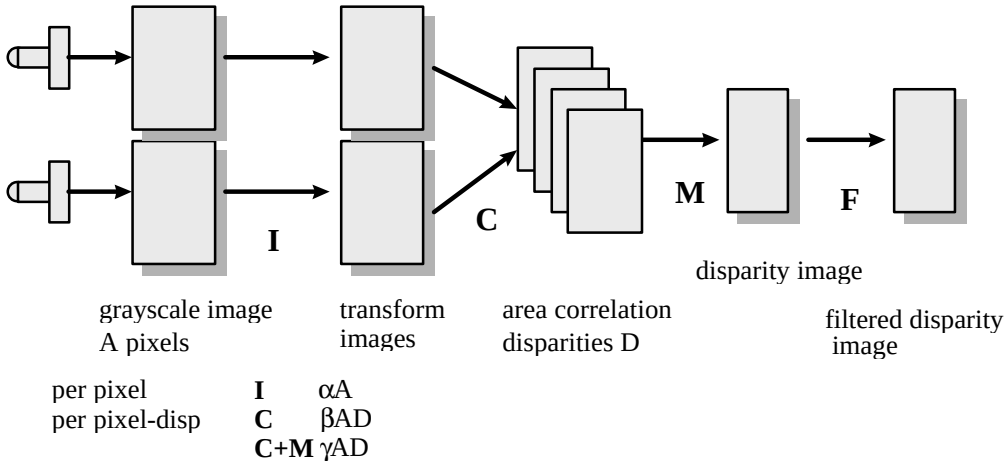


Figure 3 Area correlation algorithm diagram. *A* is the area of an image, and *D* is the number of disparities searched over. Image warping and post-filtering costs are not considered here. The cost of the algorithm is divided into per pixel and per pixel-disparity contributions.

disparities x line width). Storage efficiency is critical for DSP implementations, since DSPs have limited onboard high-speed memory. It also helps to make microprocessor implementations faster, since storage buffers can be contained in high speed cache.

3.2 Calibration

As described in Section 2.2, the standard form for stereo processing assumes that the two images are from pinhole cameras of the same focal length, and are co-planar, with scanlines and focal centers aligned horizontally. The difficulty of acquiring and maintaining this alignment has led us to consider strategies for automatic calibration. We assume that the internal camera parameters are already known, so that lens distortion can be corrected (Devernay 1995). We also assume that the cameras are in approximately the correct position, although there may be small errors in orientation. To a first approximation, orientation errors can be considered as generating horizontal, vertical, and rotational offsets of one image to another. Additionally, vergence may make the epipolar lines non-horizontal.

To calibrate these external offsets, we consider the appearance of disparity images under calibrated and noncalibrated conditions (Figure 4). The difference between these two is visually obvious, and we have found two image measure that correlate well with calibrated images. The first measure is simply the sum of all left/right consistency matches in the area correlation algorithm. If the input images are not calibrated, there will be few valid stereo matches, and the left/right check will reject many of these.

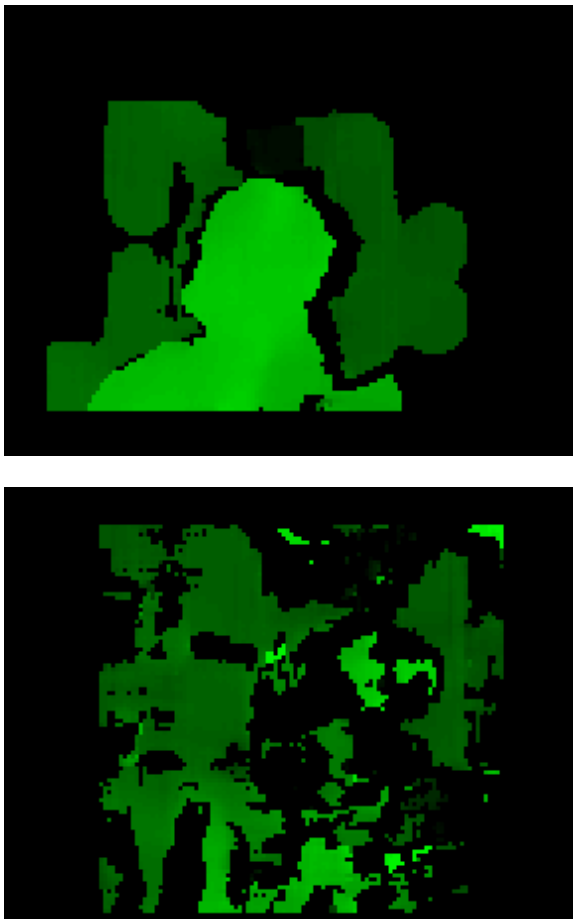


Figure 4 Calibrated and uncalibrated disparity images. The stereo input to the bottom image was offset 2 vertical pixels from the top.

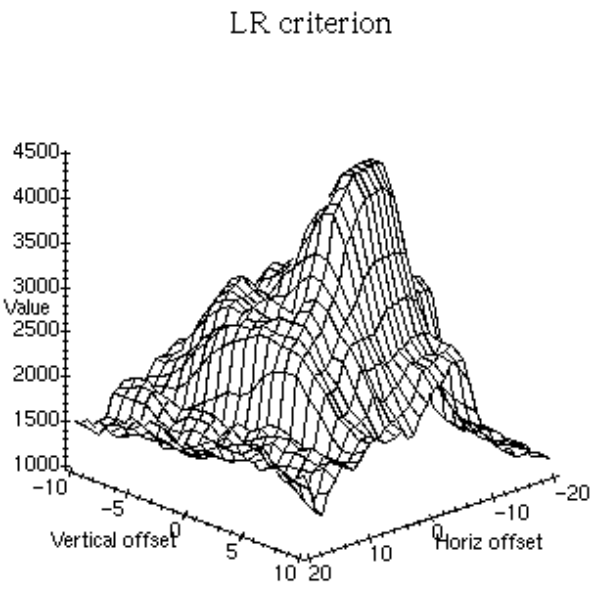


Figure 5 Number of left/right matches as a function of horizontal and vertical offset.

The second measure looks at the smoothness of the disparity image; it is the number of disparity values at the mode for a small area, summed over the entire image.

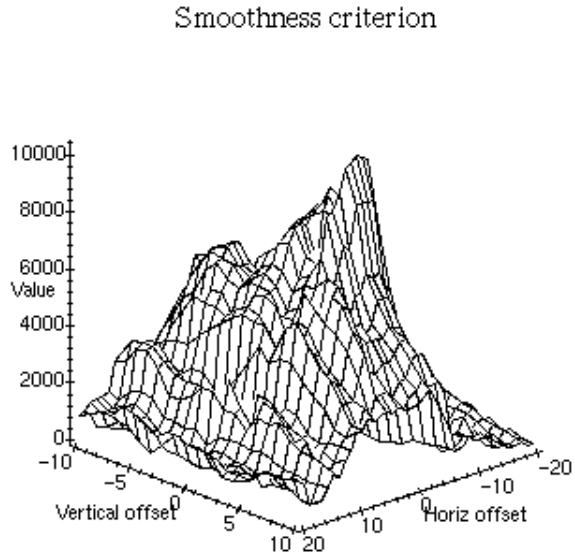


Figure 6 Smoothness of disparity image as a function of horizontal and vertical offset.

To use these measures, we first capture a stereo pair of a scene with objects at different distances, and compute disparity images at different offsets. Plots of the measures against vertical and horizontal offsets are shown in Figure 5 and Figure 6. Note that there is a peak at the calibrated offsets. The peak is very sharp in the vertical offset direction, because even small changes in vertical offset destroy the horizontal alignment of the images. The horizontal offset is more forgiving, since different offsets still give enough leeway to capture the depths of all objects in the image.

Unfortunately, the surfaces show that a hill-climbing algorithm is not sufficient to find the best offsets, since there are local maxima away from the peak. Instead, we can use a hierarchical search strategy where we first search at a coarse resolution, then use a fine resolution at a few of the highest points from the first search. This method takes on the order of 100 disparity image calculations, which is a few seconds at video rates.

Once horizontal and vertical offsets are calibrated, the measures can be used to compensate for vergence and rotation, using hill-climbing and perhaps iterating the offset calculation over a small space.

3.3 Small Vision Module

The Small Vision Module is a hardware and firmware implementation of the SRI area correlation algorithm. The hardware consists of two CMOS 320x240 grayscale imagers and lenses, low-power A/D converters, a digital signal processor and a small flash memory for program storage. All of these components are commercially available. The SVM is packaged on a single circuit board measuring 2" x 3" (Figure 7). Communication with a host PC for display and control is through the parallel port. During operation, the DSP and imaging system consume approximately 600mW.

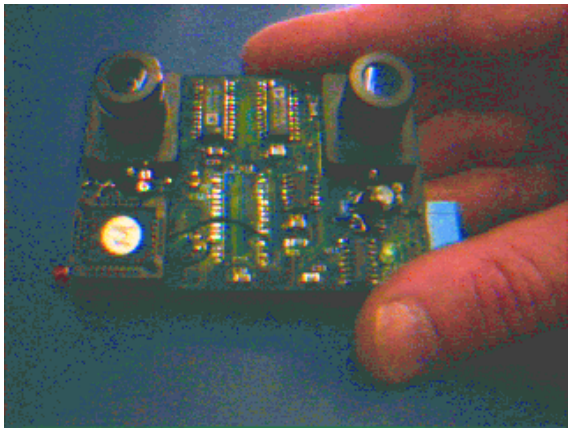


Figure 7 The SRI Small Vision Module

In its current implementation, the SVM uses an ADSP 2181 running at 33 MHz. Figure 3 shows the general algorithm, with the data structures

produced at each stage. One of the surprising aspects of the SVM is that the entire algorithm, including buffering for two 160x120 images, fits into the DSP's internal memory of 80 Kbytes.

Based on instruction set timings, the SVM can perform 8 fps on 160x120 images; in practice, communication overhead with the host reduces this to a maximum of 6 fps, enough for realtime operation of some range applications. Currently some 15 SVMs have been distributed to research labs for evaluation.

We have designed a second-generation SVM, the SVM II, using a new DSP from Texas Instruments (TMS320C60x), running at 200 MHz. In simulation, the SVM II has a performance improvement of x30 over the SVM. Performance of these devices relative to other implementations of area correlation is given in the next subsection.

One area of improvement for the SVM is in the imaging system. Current CMOS imagers are still an order of magnitude noisier and less sensitive than corresponding CCDs. On the other hand, they offer a degree of on-chip integration that cannot be achieved with CCDs: automatic exposure control, clock and control signals, and even A/D conversion. Future generations of CMOS imagers will narrow the gap in video performance with CCDs, and yield better stereo results. We have recently started using new imagers from Omnivision Technologies (Kempainen 1997), which achieve S/N ratios of 46 dB even with automatic gain, and which have sensitivities down to 1 lux.

3.4 Small Vision System

For development and experimentation on standard hardware, we have implemented the SRI area correlation algorithm in C on standard microprocessors. Further optimization of the algorithm is possible by using the Single-Instruction, Multiple Data (SIMD) instructions available on the Pentium (MMX) and several other microprocessors, which enable a parallel computation of the transform and correlation operations. For the SVM algorithms, these instructions increase speed by at least a factor of 4 over comparable μPs without SIMD.

4 Benchmark results

We survey some implementations of area correlation algorithms, comparing them to the SVM hardware, as well as software implementations of the SRI algorithm on general-purpose microprocessors. Figure 3 diagrams a generic area correlation algorithm, showing the operations and costs associated with each (Fua 1993). For concreteness, we take a benchmark problem to consist of input 320x240 images, with a search window of 32 disparities. The performance of an implementation is measured as a frame rate on the benchmark problem.

In implementations with multiple processors or distributed computation, algorithm operations can be performed in a pipelined fashion to increase throughput. The largest cost is in the correlation and extrama-finding operations, which are proportional to the number of pixels times the number of disparities. Note that the area correlation window size is not a factor, since algorithms can take advantage of redundancy in computing area sums, using a sliding sums method (Fua 1993).

Table 3 gives performance measurements for various implemented area correlation systems.[1] The columns list the type of correlation, the processor(s), and the operation coefficients, where available. Finally, as an overall measure of performance, the estimated results on the benchmark problem are given. From these results, it's possible to draw some general conclusions about the performance of the various types of processors: general-purpose microprocessors (μPs), digital signal processors (DSPs), and custom hardware (typically field-programmable gate arrays, or FPGAs).

First, it makes little difference which of the three correlation methods is chosen: all of the correlation methods can be implemented on all types of hardware. The census operator, because of its heavy reliance on bit-counting, is more suited to FPGA hardware, since DSPs and microprocessor generally don't have bit-counting instructions. But, algorithm design can make a big difference: both the Point Grey and SRI systems use the same correlation measure, but the latter has double the performance (considering Point Grey uses three images).

Second, the highest level of performance is provided by FPGAs and DSPs. General-purpose μPs, even when running at comparable clock rates, are at least three times slower on the benchmark problem. The best performance of μPs comes from using the Single-Instruction, Multiple Data (SIMD)

[1] These figures are the author's best estimate based on published reports and his own experiments.

System	Method	Processor(s)	α β γ	Bench fps	Remarks
Matthies JPL 1995	LOG square diff	Datacube MV200, 68040, Sparc		0.5	
Nishihara Teleos 1995	LOG sign Hamming	Pentium 166 MHz		0.5	
Woodfill Stanford 1995	Census Hamming	Sparc 10-51 50 MHz		1.0	
Faugeras INRIA 1993	Normalized correlation	PeRLe-1 FPGA array		2.5	
SVM+ SRI 1997	LOG abs diff	2 x ADSP 218x 50MHz	800 160 200	2.5	
CYCLOPS Point Grey 1997	LOG abs diff	Pentium II 233 MHz		3.0	3 images est. for P II
SVM algorithm SRI 1997	LOG abs diff	Pentium II 233 MHz	70 20 33	12	
Kanade CMU 1996	LOG abs diff	Custom hardware, TMS320C40 array	66 33 33	15	5 images α, χ pipelined
Dunn CSIRO 1997	Census Hamming	Occam FPGA array	15	20	
SVM II SRI 1997	LOG abs diff	TMS320C60x 200 MHz	50 10 12.5	30+	
Woodfill Interval 1997	Census Hamming	PARTS FPGA array		30+	Est. >100 fps with upgrade

Table 3 Performance of some implementations of area correlation stereo.

instructions, which enable a parallel computation of the transform and correlation operations. For the SRI algorithms, these instructions increase speed by a factor of 4 over comparable μPs without SIMD (e.g., the R10000).

Despite performance limitations, μPs offer the most flexible development environment. DSPs, while somewhat less flexible than μPs, can be programmed in C, with critical inner loops in assembly. At this point they probably offer the best compromise between performance and programmability. FPGAs, because they can be configured to pipeline successive operations and parallelize individual operations, offer the best performance. However, programming FPGAs can be difficult, and any changes in the algorithm can take significant development time (Woodfill and Von Herzen 1997).

4.1 Machine cycles and efficiency

A more critical examination of the same algorithm running on different processors reveals the tradeoffs in hardware and software design, and helps to quantify the efficiency of a hardware implementation. This measure is especially critical in some areas such as space missions or portable applications. To compare power ratings, selected processors of each type were rated based on how much power they would take to achieve the benchmark.[2] The results are in Table 4, which also details the number of instructions used per pixel per disparity in the correlation calculation. All of these processors ran the same SVM algorithm; the MMX and FPGA array implementations were optimized for parallel computation.

Without the SIMD MMX instructions, μPs do poorly. The Pentium MMX is fairly efficient, since it processes four to eight pixels at once. Since there are two MMX pipelines, theoretically it can process 2 instructions at once, which would make it almost as fast as the TMS320C6x DSP on a MIPS basis. However, because of cache miss limitations, the MMX instructions are issued at below their theoretic maximum rate. In DSPs, the programmer has control over fast cache memory, and can buffer critical information in this area as needed. The

[2] These power ratings are the author's estimates rather than measured results, and are based on processor data sheets and published results. Since processor characteristics change quickly, these results may not be valid in the future.

Processor	Number of Instructions	β	Bench Power
SGI R10000 200 MHz	24	100	280W
Sparc Ultra 200 MHz	24	90	180W
FPGA array Dunn	NA	16	40W
Pentium II MMX 233 MHz	5 (20 / 4)	20	25W
ADSP 218x 50 MHz	8	160	10W
TMSC6x 200 MHz	2 VLIW, 8FU	10	4W

Table 4 Processor power ratings for the benchmark problem.

general caching scheme of μPs means that the processor will idle while waiting for cache fills. With tuning of the code, and some attention to caching issues, it may be possible to double the performance of the MMX implementation. Since most of the major μP manufacturers plan versions of their chips with SIMD instructions, μPs are a viable platform for realtime stereo, except in low-power applications.

It is somewhat surprising that the DSPs are more efficient than FPGAs, since the latter have a speed advantage based on their highly-parallel implementation of the stereo algorithm. Because of their fixed functional units, however, DSP designs can be optimized for high performance at low power. The number of instructions for a standard design such as the ADSP 21xx is 1/3 that of a non-SIMD μP. The TMS320C6x achieves its low instruction count by having parallel functional units (in contrast to the parallel data concept of SIMD), achieving 12 operations per cycle in the correlation loop.

5 Applications

Practical realtime stereo has been available recently, and applications are just being developed. Kanade's group has implemented a Z-keying method, where a subject is placed in correct alignment with a virtual world (Kanade 1996). At SRI, we are experimenting with several applications involving mobile robotics and human interaction. In collaboration with Gaetan Marti and Nicola Chauvin of the Swiss Federal Institute of Technology in Lausanne, we have developed a *contour method* for fast segmentation of range images based on 3D areas. This method can be used, for example, to distinguish terrain height in front of a moving vehicle. The important part of the method is that results are computed within the range image, rather than in 3D space, making it suitable for realtime implementation.

A second application is the detection of people within the camera's range, in collaboration with Chris Eveland of the University of Rochester (Eveland 1997). Here the solution is to segment the range image by first learning a background range image, and then using statistical image comparison methods to distinguish new objects that appear in the field of view. A simple shape analysis is used to find the head and shoulder area of the person. The interesting part of this application is that the camera can pan and tilt to track the subject.

More details of these applications can be found at http://www.ai.sri.com/~konolige/svm..

References

Barnard, S. T. and M. A. Fischler (1982). Computational stereo. *Computing Surveys* **14**(4), 553-572.

Bolles, R. and J. Woodfill (1993). Spatiotemporal consistency checking of passive range data. *Proc ISRR93*.

Devernay, F. and O. Faugeras (1995). Automatic calibration and removal of distortion from scenes of structured environments. *Proc SPIE95*.

Dunn, P. and P. Corke (1997). Real-time stereopsis using FPGAs. *FPL97*, London.

Eveland, C., K. Konolige and R. C. Bolles (1997). Background Modeling for Segmentation of Video-Rate Stereo Sequences. Submitted to *CVPR98*.

Faugeras, O. et al. (1993). Real time correlation-based stereo: algorithm, implementations and applications. *INRIA Technical Report 2013*.

Fua, P. (1993). A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications* **6**(1).

Kanada, T. et al. (1996). A stereo machine for video rate dense depth mapping and its new applications. *Proc. CVPR96*.

Kempainen, S. (1997). CMOS image sensors: eclipsing CCDs in visual information? *EDN*, October 9, 1997.

Marr, D. and T. Poggio (1979). A computational theory of human stereo vision. *Proc. Royal Society* B-204, 301-328.

Matthies, L. (1993). Stereo vision for planetary rovers: stochastic modeling to near realtime implementation. *IJCV* **8**(1), 71-91.

Matthies, L., A. Kelly, and T. Litwin (1995). Obstacle detection for unmanned ground vehicles: a progress report. *Proc. ISRR.*

Moravec, H. P. (1979). Visual mapping by a robot rover. *Proc. IJCAI*, Tokyo, 598-600.

Nishihara, H. K. (1984). Practical real-time imaging stereo matcher. *Optical Engineering* **23**(5), 536-545.

Point Grey Research (1997) *http://www.ptgrey.com*.

Woodfill, J. and B. Von Herzen (1997). Real-time stereo vision on the PARTS reconfigurable computer. *IEEE Workshop on FPGAs for Custom Computing Machines*.

Zabih, R. and J. Woodfill (1994). Non-parametric local transforms for computing visual correspondence. *Proc. ECCV94*.

Zabih, R. and J. Woodfill (1997). A non-parametric approach to visual correspondence. Submitted to *IEEE PAMI*.