

The Future of High Performance Machine Vision

Introduction

Recently several of Alacron's customers from the inspection and semiconductor industries have asked us to integrate multiple fast large CCD arrays or the new fast high frame rate CMOS sensors with data rates in the 500 to 1000 MB/sec range into a real-time systems. This trend is opening a two tiered approach to machine vision. There is the "Native", i.e. Pentium based, computing with a "basic", i.e. non-accelerated frame grabber. The other avenue is to accelerate the process prior to PC transfer as represented by accelerated frame grabbers or cameras. The native approach is usually is preferred by customers because it is able to provide an environment, which is:

- Easy to program with optimized native libraries from multiple vendors,
- Fast enough for real-time processing,
- Cheaper to deploy.

This approach is feasible as noted by this author in the supplement to the May 2002 Vision System Design Magazine, for the lower end of the frame grabber market, which will probably migrate to USB2.0 or IEEE 1394, i.e. Firewire, because of adequate performance, widespread availability, and low or no cost motherboard options. Hence these interfaces achieve reasons 1 to 3 above for data rates within the 0-40MB/sec range, which is adequate for a large portion of the machine vision market. This also is within the realistic throughput for a single or dual Intel based Pentium solution.

What is the alternative if a customer would like to do more intensive processing or use a significantly increased sensor data rates, which for newer multiple CCD and CMOS sensors is approaching or exceeding 1GB/sec? The "native" approach to this problem is to buy a SMD Pentium box with an adequate throughput bus i.e. 32 or 64-bit PCI bus, and an adequate basic frame grabber. While this may seem to be the direct solution, it may not be the cheapest, fastest, nor easiest to deploy in the high performance machine vision environment.

Scalability of Native Processing

To examine the feasible of scalability of native processing we need to examine the two different memory schemes that are commercially available, i.e. cluster versus shared memory. The cluster (shared private distributed memory approach model) is that every processor has local memory in which to operate. A stack of PCs can be linked by 100Mbit or Gigabit Ethernet exemplifies this model or for an embedded example is Coreco's Mamba Series. The performance and unit cost is generally linear for some reasonable number of units, i.e. less than 10. The other shared memory scheme is inherent in the commercially available server and workstation units, which come with a support, chip that shares memory among four to eight processors. These units often are not linear but super linear with cost and with the number of processors their performance is sub-linear. To see this effect, Alacron used the Intel Graphics Suite, to benchmark the scalability of two and 4 processor shared memory architectures. The extrapolation to eight processors is straightforward since the eight-processor solution is no faster than the cluster of two four shared memory units. The Intel Fusion Chipset supplies this architecture. For the shared two-processor model we obtained a performance increment of 1.6 units, i.e. the time to perform two threads of the Intel library was 1.7 times the uni-processor model. For the four-processor configuration running four threads the result was 5/8 times four the uni-processor time. This leads to a

scalability factor of 90% for the two-processor model, i.e.. 2 processors have the throughput of 1.7, and 60% for the four-processor model i.e. 4 processors has the throughput of 2.6 processors.

Microprocessor and FPGA Comparisons

In order to understand which approach native or other has advantages we studied, the relative performance of near-future microprocessor and FPGA offerings, the cost of implementation, and the power consumption relative to throughput. From this data we then can examine the implications for the future of high performance machine vision. In the above section we established that the scalability of the shared memory native solution is approximately 85% for two and %65 for four processors for the SMD approach. The cluster or Shared private memory approach is usually linear or nearly so with processing units because:

- No inter-processor contention for a common piece of hardware
- Splitting of the I/O streams does not unduly burden a processor with I/O that it does not need. For example a 1GByte/sec I/O stream split over 8 processors results in a 125 MB/sec load to each processor on a cluster or SPDM machine where all the processors will see the 1GByte/second load in the SMP solution.

Solution Comparisons

In order to measure relative performance for imaging we selected a suite of routines to give a performance ratio for imaging. We are comparing the near future state-of-the-art processors from Intel, Philips, and Motorola and FPGAs (field programmable gate arrays) from Xilinx, which represent the various solutions vendors are using to handle high data rate or compute intensive applications. The table below is consists of examples of processors and FPGA which are new or will be soon introduced, and their performance of imaging , cost and power relative to the benchmark Intel P4. The table indicates speed that is a larger number means faster, relative to the P4.

<i>Parameter</i>	<i>300 Mhz TriMedia1500</i>	<i>1000 Mhz MPC8540</i>	<i>1300 Mhz MPC7455</i>	<i>3000 Xilinx V2</i>	<i>3000 Mhz P4</i>
ImageAdd	0.26	0.90	0.77	29.00	1.00
Sobel8	0.28	0.43	2.01	1.30	1.00
3x3 Conv	0.51	0.96	2.30	1.30	1.00
11x11 conv	1.37	2.31	12.18	41.83	1.00
3x3 erode gray	0.90	0.58	1.22	0.69	1.00
Hist8 (32)	2.10	2.24	1.89	1.08	1.00
2D FFT	1.19	0.72	2.61	21.29	1.00
Lut8	0.86	1.11	0.95	0.53	1.00
Cost	\$60	\$100	\$300	\$600	\$350
Power	3W	7W	20W	4W	60W
Average Performance	0.94	1.16	2.99	8.50	1.00
Relative Cost	\$64	\$86	\$100	\$70	\$350
Relative Power	3.2W	6W	6.7W	0.5W	60W

This table demonstrates the relative advantages and disadvantages of each approach to the method one uses to solve machine vision problems. The conclusions are:

- **Performance**

As one can see, the FPGA approach has a significant performance advantage.

- **Implementation Cost.**

This table shows that an embedded or FPGA solution is the most cost effective if more than one or two P4s are needed to handle the data flow. It should also be noted that the sensor or frame grabber I/O is limited by the latest PC bus architecture, i.e. PCI-X, i.e. 64 bit X133 MHz = 1.2 GB/sec. Also to handle this throughput most native solutions are simply incapable of sufficient throughput to handle these data rates at the high end, even in a Cluster or SMD native environment because of the prohibitive cost of real-time processing

- **Power Budget**

The power consumed by processors is also important, especially if one considers the placement into a camera as the optimal solution to high-speed imaging. The camera environment provides a significant advantage for processing since the selection of frame grabber and processing platform is solved in the camera environment. Also significant data reduction can take place prior to downloading the camera, which significantly decreases integration and frame grabber costs.

Conclusion

The native solution is both feasible and desired for data rates and camera applications that can be performed on one or two P4s, which is cost effective for data rates in the 40 MB/sec or somewhat greater range. However, when data rates or real-time processing become intensive, i.e. 80 MB/sec or greater, then an embedded or FPGA solution offers more cost effective, efficient and no more difficult development than the native solution. If one wants to produce smart or embedded processor cameras, then the only feasible choice is the embedded processor with the FPGA solution being optimal since most preprocessing of images requires only limited repertoire of fixed point processing which gives the FPGA solution the distinct advantage. Thus the FPGA approach:

- In the really high speed applications may be the only workable solution
- Is superior for some operations that are not well suited to processors because of the limited amount of cache, or register space. An FPGA implementation of LUTs is a good example. A more specialized routine is to perform an 8x8 DCT or 2DFFT in a single clock. No processor has implemented that kind of special purpose instruction.
- Is more economical or financially feasible since a few FPGAs can do some very hard application that might take many general purpose processors.
- Are very flexible and can change on the fly.

If more complex or significant floating point processing is needed then the use of embedded or FPGA or a mix in either the camera or frame grabber yields significant performance, cost and power advantages over the native approach.

With the recent introduction of hybrid FPGA with processors, e.g. Xilinx Vertex II Pro, this combination may allow significant simplification of the choices for a manufacturer who can modify the mix of cells and processors as need in either the camera or frame grabber to meet the needs of the customers particular application. Thus individual customization using the hybrid solution may be the preferred approach in the not too distant future when the hybrid FPGA/processor solution becomes more widely available and cost effective.



71 Spit Brook Road, Suite 200
Nashua, NH 03060

Tel: 603.891.2750 Fax: 603.891.2745

Web: www.alacron.com E-mail: sales@alacron.com

© Alacron, Inc. All Rights Reserved.