Performance evaluation of a single-chip digital signal processor based multimedia system using the Abingdon Cross benchmark

Jihong Kim
University of Washington
Department of Computer Science and Engineering
Box 352350
Seattle, Washington 98195

Yongmin Kim, MEMBER SPIE
University of Washington
Department of Electrical Engineering
Box 352500
Seattle, Washington 98195
E-mail: kim@ee.washington.edu

Robert J. Gove
Compression Labs, Inc.
2860 Junction Avenue
San Jose, California 95134-1900

Jeremiah Golston
Texas Instruments
12203 Southwest Freeway
Stafford, Texas 77477

Abstract. Texas Instruments' new single-chip multiprocessor, TMS320C80, is a powerful programmable digital signal processor with many unique features optimized for multimedia, image processing and computer graphics applications. The performance of a TMS320C80-based multimedia system for image processing and computer vision tasks is evaluated using the Abingdon Cross benchmark. The result shows that the TMS320C80-based system outperforms most of 1980s specialized parallel image computers at a fraction of the system cost.

Subject terms: Abingdon Cross; TMS320C80; single-chip multiprocessor; multimedia system; benchmarking.

Paper 29125 received Dec. 21, 1995; revised manuscript received Apr. 17, 1996; accepted for publication Apr. 18, 1996.

1 Introduction

Many tasks in image processing and computer vision applications require a large number of computing cycles. For example, in order to convolve a 512×512 image with a 5×5 kernel, over 13 million multiplications and additions are necessary. In addition, as applications require more real-time tasks (e.g., real-time road traffic monitoring[1]), an even higher level of computing power is necessary. To meet the heavy computing requirements for these applications, parallel processing techniques have been widely used in various systems from the single-chip level to the system architectural level.[2] Coupled with the inherent parallelism available in image processing and computer vision tasks, parallel image computers have been effective in meeting the intense computing requirements of many demanding applications.

While large-scale parallel image computers are still necessary for highly computing-intensive tasks, they are not appropriate for many emerging real-time embedded applications such as an autonomous vehicle operation and telemedicine. Large system size, heavy weight, and large power consumption as well as high system costs make them impractical solutions for such applications. For example, a large $500,000 parallel image computer would not be suitable as an embedded computing engine for a mobile robot. For these types of applications, highly integrated systems based on standard off-the-shelf CMOS/VLSI semiconductor devices such as microprocessor and digital signal processors (DSPs) would be more appropriate if they could meet the specific application requirements.

In this paper, we report on our evaluation of the suitability of one such processor, the Texas Instruments multimedia video processor (MVP) or TMS320C80,[3] for image processing and computer vision tasks using the Abingdon Cross benchmark.[4] The TMS320C80 is a single-chip multiprocessor with many unique features optimized for multimedia, video compression, image and signal processing, and computer graphics. A TMS320C80-based multimedia system, the MediaStation 5000 (MS5000), was used to evaluate the TMS320C80.[5] The Abingdon Cross benchmark was selected from among several image understanding (IU) benchmarks for parallel image computers (e.g., Defense Advanced Research Projects Agency (DARPA) IU benchmarks I and II[6]) because it has been widely implemented in various systems due to its simplicity, thus providing more data for the comparative study. The results from more than 50 systems are available for the Abingdon Cross benchmark, while results from fewer than 10 systems are available for the DARPA IU benchmark.

The rest of the article is organized as follows. The Abingdon Cross benchmark is described in Sec. 2, and the TMS320C80 processor and MS5000 system are explained in Sec. 3. The detailed algorithm and its implementation are described in Sec. 4 while the benchmark results are discussed in Sec. 5.

2 Abingdon Cross Benchmark

The Abingdon Cross benchmark consists of finding the medial axis of a cross in a noisy background.[4,7] It was designed to test many common components of image process-

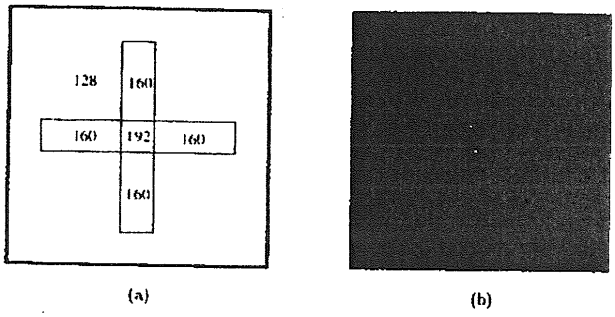


Fig. 1 The Abingdon Cross benchmark: (a) the description of the noise-free cross image and (b) an Abingdon Cross image with random noise.

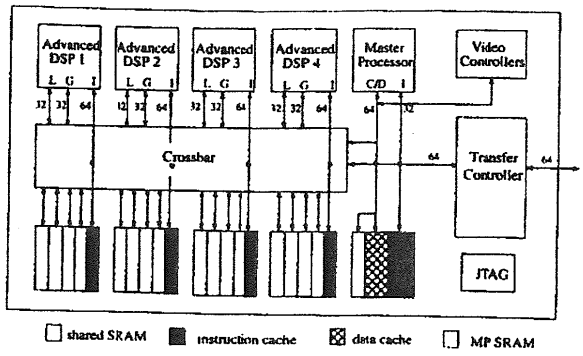


Fig. 2 High-level block diagram of the TMS320C80.

ing systems such as input-output, matrix and vector operations, Boolean operations, filtering, and cellular logic. Each pixel in the test image has an 8-bit depth while the spatial resolution of the test image may be adapted to the architecture being benchmarked. The cross is centered in the middle of the image and consists of a horizontal arm and a vertical arm. All background pixels have a gray value of 128. The pixels of two arms of the cross have a gray value of 160, and the pixels of the intersection of the two arms have a gray value of 192. If the image has an $N \times N$ resolution, the arms of the cross are $3N/4$ pixels long and $N/8$ pixels wide. White Gaussian noise with a mean of zero and a standard deviation of 32 is added to all points. Figure 1(a) shows the noise-free test image while one example image with random noise is shown in Fig. 1(b).

In the Abingdon Cross benchmark, two factors are defined to compare the performance of image computers. The quality factor QF_S of a system S is defined by

$$QF_S = N_S / T_S, \tag{1}$$

where T_S is the total execution time for finding the medial axis of the cross and N_S is the width (or height) of the test image used. The quality factor QF_S is used to assess the computing power of the system S benchmarked. The second factor, the price-performance factor PPF_S, takes into account the system cost C_S (in U.S. dollars) and is defined by

$$PPF_S = N_S^2 / T_S C_S. \tag{2}$$

The PPF_S factor can be interpreted as the amount of processing power achievable per dollar with the system S.

3 TMS320C80 Processor and MS5000 System

In this section, we briefly describe the TMS320C80 (MVP) processor and MS5000 system. (For a detailed description, see Refs. 3 and 5, respectively.)

3.1 *Overview of TMS320C80*

The TMS320C80 can be described as a single-chip, heterogeneous, multiple instruction-stream, multiple data-stream (MIMD) multiprocessor connected via a crossbar to multiple on-chip shared memory modules. It combines a reduced instruction set computer (RISC) processor and four advanced DSPs as well as an intelligent direct memory access (DMA) controller and two video controllers in a single-chip device. It is capable of processing more than 2 billion operations per second (BOPS) with the 2.4 Gbytes/s on-chip data transfer rate. In order to reduce the data transfer overhead with the external memory and devices, a large on-chip memory (25 2-kbyte modules) is provided as well.

Figure 2 shows a high-level block diagram of the major functional blocks of the TMS320C80. The master processor (MP) is a general-purpose RISC processor with an integral IEEE 754 compatible floating-point unit. In a typical operation mode, the MP serves as the main supervisor and distributor of tasks within the TMS320C80. Also, the MP is the preferred processor for performing high-precision floating-point operations. Four advanced DSPs (ADSPs 1 to 4) have a highly parallel architecture optimized for multimedia, video and image compression, image and signal processing, and computer graphics. Each ADSP is capable of performing up to 15 RISC-equivalent operations in a single clock cycle via a long instruction word (64 bits) mechanism and has many powerful features not found in conventional DSPs. They include:

- single-cycle accesses to on-chip memory, allowing two 32-bit data transfers per processor in every cycle concurrently with data operations
- a three-operand 32-bit arithmetic and logical unit (ALU) that can be optionally split into two 16-bit or four 8-bit units
- multiple flags (mf) register, which captures the multiple status results (flags) from split ALU operations, and an expander that takes 1, 2, or 4 bits in the mf register and replicates them 32, 16, or 8 times
- a barrel rotator that can prerotate an input to the ALU by 0 to 31 bits
- three zero-overhead hardware loop controllers that allow three levels of nested loops to be controlled with no associated overhead
- dedicated adders for address generation that can also be used for arithmetic operations
- conditional ALU and data transfer operations that substitute for many compare-and-branch operations.

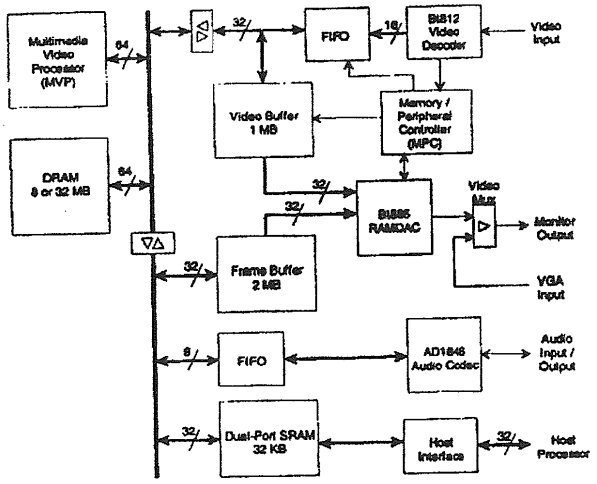


Fig. 3 Overview of the MS5000 system.

The transfer controller (TC) is a dedicated memory controller with sophisticated data transfer logic. The TC is responsible for servicing the data transfer requests and cache misses of the five internal processors. The 50-kbyte on-chip memory consists of instruction caches, data caches, data RAMs, and parameter RAMs. An instruction cache is assigned to the MP and each ADSP, but the data cache is available only to the MP. For the ADSPs, the data RAMs serve as the on-chip local storage area. While the cache memory is automatically serviced by the TC to read from and write to the external memory, the data RAMs need explicit management and transfer requests by the programmer in software. The processors and on-chip shared-memory modules are fully interconnected through the high-performance crossbar switch network which operates at the instruction clock rate. The video controllers (VC) provide supports for programmable video timing to control both capture and display.

3.2 *Overview of MS5000*

The MS5000 system based on the TMS320C80 is a highly integrated desktop multimedia system implemented on a single personal computer (PC) plug-in board. It transforms a PC into a programmable high-performance multimedia and image workstation, off-loading the host computer from compute-intensive tasks using a TMS320C80, custom programmable logic devices and other supporting chips. It supports real-time Motion Picture Experts Group (MPEG) compression and decompression[8] and still-picture processing [e.g., Joint Photographic Experts Group (JPEG)[9]] as well as image processing (e.g., convolution, Fourier transform, contrast enhancement and geometric transformation) and 2-D/3-D computer graphics. The MS5000 has several built-in special I/O features such as a video digitizer and a stereo audio coder and decoder for multimedia applications as well. In implementing the Abingdon Cross benchmark, these special I/O peripherals were not utilized.

Figure 3 shows the block diagram of the MS5000 system. Based on the single bus architecture, the TMS320C80 is the core computing engine of the system while the memory and peripheral controller (MPC) works as both an arbitrator of data transfer requests among different system components and a controller for various peripheral devices. The incoming video signals are digitized by the video decoder, stored into the video buffer, processed in the TMS320C80, and sent to the random access memory digital-to-analog converter (RAMDAC) for display. The dual-port static random access memory (SRAM) and host interface logic provide a bidirectional communication path between the host PC and MS5000.

4 Algorithm and Implementation

In order to find the medial axis of the cross from the Abingdon Cross test image, three tasks are typically performed: thresholding, noise removal, and skeletonization. However, various implementations reported in the Abingdon Cross benchmark surveys[4,7] were quite different in detailed organization of these tasks. One of the main differences was in how to remove white Gaussian noise from the test image. Many different techniques have been used for noise removal, such as an iterative morphological filtering, median filtering, and convolution. Additional variations include the specific skeletonization algorithm selected, the order of thresholding and noise removal tasks, and the use of erosion before the skeletonization step.

In our algorithm, the image was first thresholded, and noise was removed by two separate 1-D binary median filterings with a large window. Then, erosion operations shrank the arms of the cross quickly before the skeletonization algorithm was applied to the image. The step-by-step sequence of our algorithm is as follows:

Step 1: thresholding of the image at gray value 140
Step 2: noise removal in the binary image by 1×21 binary median filtering followed by 19×1 binary median filtering
Step 3: packing of 8 bits/pixel image to 1 bit/pixel bitmap image for optimized implementation of skeletonization steps
Step 4: 4 iterations of binary dilation with a 3×3 square structuring element
Step 5: 35 iterations of binary erosion with a 3×3 square structuring element
Step 6: 6 iterations of thinning
Step 7: unpacking of 1 bit/pixel bitmap image to 8 bits/pixel image.

Step 3 (packing to the bitmap image) and 7 (unpacking to 8-bit image) are not necessary to find the skeleton itself, but they make the implementation of the skeletonization step more efficient than it would be without taking these steps, as discussed later.

Figure 4 shows the intermediate results and final output of our algorithm implemented on the MS5000 system. A 512×512 Abingdon Cross test image was used for benchmarking. As shown in Fig. 4(a), there is still some noise left after the thresholded image is filtered using 1×21 vertical and 19×1 horizontal binary median filters. This noise could be easily removed by another phase of median filtering, if necessary. In our algorithm, we skipped this step because the remaining noise in the background could be completely eliminated by subsequent iterations of binary erosion operations. Figure 4(b) shows the intermediate result after 35

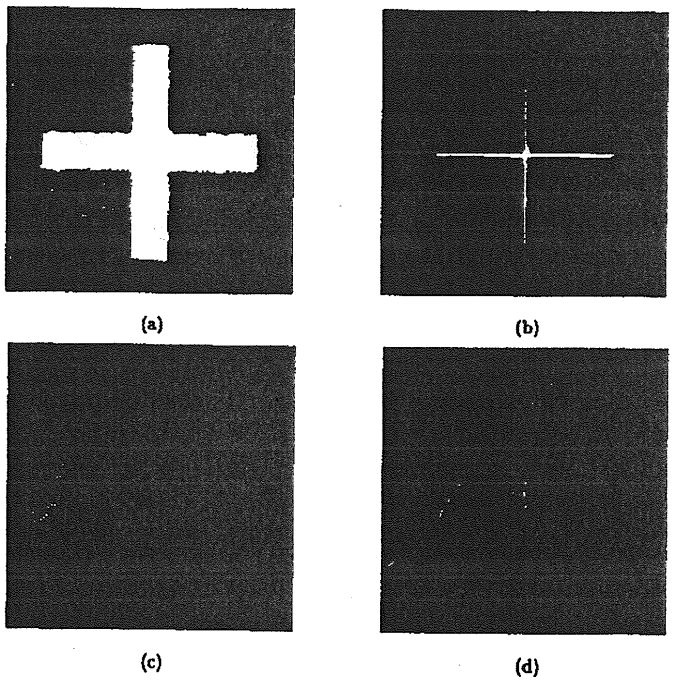


Fig. 4 The Abingdon Cross implementation on the MS5000 system: (a) the output image after thresholding and separate 2 1-D median filtering, (b) the output image after partial dilation and erosion, (c) the final output image after thinning, and (d) the final result superimposed on the original cross image.

iterations of binary erosion with a 3×3 square structuring element were performed. The image shows an almost complete skeleton. The final skeleton image was computed using a thinning algorithm,[10] and Fig. 4(c) shows the skeleton computed. Figure 4(d) shows the final output superimposed on the original test image.

In our implementation of steps 1 to 7, the MP functions as a task distributor and arbitrator among four ADSPs, passing tasks for each step with appropriate parameters to each ADSP and synchronizing their executions for the next step. Since the ADSPs are responsible for all the compute-intensive processing, attaining the maximum performance of the ADSPs is important. Because of the substantial performance difference (by at least an order of magnitude[11]) between the programs coded in high-level language and hand-optimized programs in the ADSPs, the time-critical portion (e.g., tight-loop bodies) of steps 1 to 7 was programmed in ADSP assembly language, directly utilizing various performance-enhancing features of the ADSPs to speed up execution. For example, consider the thresholding operation that produces a binary output image from an 8-bit gray-scale input image. If an input pixel is greater than or equal to the threshold value, the corresponding output pixel is set to the user-selected nonzero output pixel value. Otherwise, it is set to zero. In the first instruction, the 32-bit ALU is split into four 8-bit units, and the differences of 4 input pixels and a threshold value are calculated simultaneously in each ADSP. The carry bits from this operation are saved in the mf register. The carry bit is set if the input pixel is larger than or equal to the threshold value. In the next instruction, the saved four least significant bits of the mf register are replicated eight times using the expander, and four bitwise logical AND operations with the user-specified output pixel value are performed. If the carry bit was 1, the bitwise logical AND operation is performed between the user-specified output value and 0xFF. If the carry bit was 0, the bitwise logical AND operation is performed between the user-specified output value and $0x00$. Therefore, thresholding can be performed in two cycles for 4 pixels (0.5 cycles per pixel) on each ADSP. With four ADSPs running concurrently, 8 pixels can be thresholded in every cycle. Similar techniques are used as well to optimize the implementation of median filtering, packing, and unpacking.

The packing operation compresses the noise-removed image into the bitmap for the fast implementation of steps 4 to 6. By using the bitmap, steps 4 to 6 can process multiple pixels simultaneously. In addition, the data transfer time between external memory and on-chip memory is reduced to one-eighth of the time necessary for transferring 8-bit data. Because steps 4 to 6 are the most compute-intensive parts of the Abingdon Cross benchmark, the benefits of using the packed image far outweigh the extra processing time necessary for packing and unpacking operations.

Binary dilation, binary erosion, and thinning operations used in steps 4 to 6 are all neighborhood operations based on a 3×3 window and their implementations are optimized similarly using the three-input ALU and barrel rotator features of the ADSP. For example, in binary erosion of an image K with a 3×3 structuring element S, if the origin of the structuring element is in the middle of the square, say, at coordinate (1, 1), the output pixel $E[i,j]$ could be computed as

Table 1 Summary of the execution time of the Abingdon Cross implementation on the MS5000.

Operation	Execution Time (ms)	Frequency	Total Execution Time (ms)
Thresholding	3.34	1	3.34
1×21 median filtering	5.38	1	5.38
19×1 median filtering	3.67	1	3.67
packing to bit image	1.96	1	1.96
3×3 binary dilation	1.14	4	4.56
3×3 binary erosion	1.14	35	39.90
thinning	12.78	6	76.68
unpacking to byte image	2.16	1	2.16

Total execution time = T_{MS5000} = 137.65 ms.

$$E[i,j]=\overline{\bigvee_{0\le k,l\le 2} K[i+k-1,j+l-1]\oplus S[k,l]}, \quad (3)$$

where $\oplus$, $\vee$ and $\bar{x}$ represent EXCLUSIVE-OR, OR and NOT logical operations. In our implementation, each word (or register) contains 32 pixels or bits. Since the leftmost and rightmost pixels do not contain their corresponding neighbors, 30 pixels are processed simultaneously. In order to form a 30-pixel processing unit, it is necessary to merge two 32-bit words and extract appropriate 30 pixels from the merged words. With the three-input ALU and barrel rotator, the EXCLUSIVE-OR and OR operations can be performed in one instruction, resulting in 9 cycles per 30 pixels for a complete 3×3 binary erosion operation on each ADSP. Owing to the processing overhead in merging and extracting input and output pixels into 30-pixel units, the overall implementation took 21 cycles to erode 30 pixels on each ADSP.

Because of the TC's autonomous data transfer capability, the ADSPs can focus on performing the operations necessary for each step without wasting computing cycles on bringing the raw image data on-chip and storing the processed results off-chip. The ADSP submits a transfer request for the next image block in advance while processing the current on-chip image block. In this way, our implementation overlaps the I/O and processing time as much as possible, effectively avoiding the data movement overhead for the ADSP.

5 Results

Table 1 shows the execution time of steps 1 to 7 implemented on the MS5000 with the TMS320C80 running at 50 MHz. The total execution time was 137.65 ms, of which skeletonization (steps 4 to 6) took about 88%. Packing and unpacking operations introduced about 3% of processing overhead, but they reduced the execution time of steps 4 to 6 significantly. The quality factor QF_{MS5000} of the MS5000 is computed to be 3720, and the price-performance factor PPF_{MS5000} is 381, with the end-user's system cost of $5,000.* If the higher clock frequency versions of the TMS320C80 become available in the near future, the total execution time would be reduced further, because skeletonization steps are all compute-bound operations and their performance would be directly proportional to the clock speed.

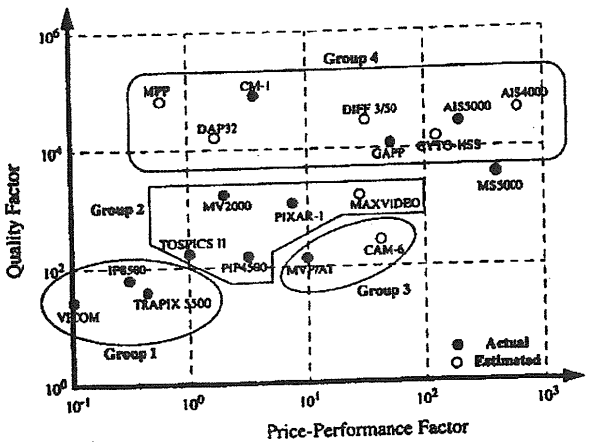


Fig. 5 The results of the Abingdon Cross benchmark with various commercial systems. Solid circles are results measured using actual systems and open circles are results estimated using actual systems.

In Fig. 5, the quality factor and price-performance factor of the MS5000 are compared with the results of other systems reported in Ref. 4. [It should be noted that there is a gap of roughly two technological generations between the MS5000 system and the systems described in Ref. 4. If the benchmark results were available for more recent versions of these systems, they would provide more interesting comparisons with the MS5000 system. However, data are not available (to the best of our knowledge) for them. Furthermore, considering that our main purpose in this study was not to claim that the MS5000 outperforms the newer generation of these systems, but to understand the relative performance of the MS5000 compared with the various systems that existed in the past, these old data are still useful.] For the objective comparison, we did not include hypothetically estimated results for systems that were not actually built. When interpreting the graph in Fig. 5, it should be noted that the QF and PPF coordinates for a system are not fixed and change with time. As commonly found in the revised versions of the same system, if the performance and price of the system improve or a more efficient algorithm is implemented, the coordinate would move upward and/or to the right. In addition, since the Abingdon Cross benchmark did not specify the exact method for finding the medial axis of the cross and did not clearly constrain *a priori* information that could be used to solve the problem, it would be more appropriate to interpret the QF and PPF coordinates as representing a region instead of a point. These shortcomings of the Abingdon Cross benchmark make the objective performance comparisons more difficult as well.†

*As explained in Sec. 3, our implementation did not utilize the built-in special I/O features available on the MS5000 for multimedia applications. The system cost of the MS5000 can be further lowered if these peripherals are not included in the MS5000.

†The original intent of defining an image processing problem as a benchmark instead of a set of specific operations was to give programmers enough flexibility so that the performance-enhancing features of an image computer could be fully utilized in computing the medial axis.

Preston[4] defined four groups of systems as shown in Fig. 5. Systems in group 1, so-called video-rate processors, have the lowest quality factors and worst price-performance factors. The group 2 systems consist of specialized (multimodule) image computers with architectures optimized for imaging applications. Board-level systems intended as plug-in peripherals to PCs belong to group 3, and massively parallel image computers form group 4.

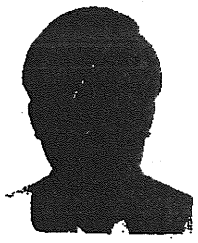
The MS5000 belongs to the group 3 category. However, the MS5000 is at least an order of magnitude faster (in terms of the quality factor) than systems in group 3, in spite of the fact that it was mainly designed to meet the real-time requirements of multimedia applications and did not include any dedicated special-function hardware circuitries for image processing or computer vision tasks. Further, it is faster than the group 2 systems while its integration level is much higher than that of those systems. The MS5000 cannot match the quality factors of systems in group 4, but this was expected because of the significant difference in the degree of parallelism between the MS5000 and group 4 systems. For example, the AIS5000[12] used a linear array of 512 processors while the MS5000 has one chip (TMS320C80). From the price-performance perspective, the MS5000 is one of the best systems and is behind only the AIS4000 in group 4. Unlike the group 4 systems, however, the MS5000 provides practical solutions for many demanding imaging applications because of its low system cost and one PC board implementation.

6 Conclusions

The performance of the MS5000 multimedia system based on the Texas Instruments TMS320C80 DSP chip was evaluated for image processing and computer vision tasks using the Abingdon Cross benchmark. Unlike many specialized parallel image computers tested by this benchmark, the MS5000 is based on a programmable single-chip multiprocessor (TMS320C80) and provides a highly integrated, low-cost solution for compute-intensive imaging applications. With the heavy use of many unique powerful features of the TMS320C80, we were able to achieve efficient implementation, although more optimizations on the algorithm and its implementation would further improve the performance. The measured result of the MS5000 system shows that the MS5000 outperforms most of 1980s specialized parallel image computers at a fraction of system cost, making the TMS320C80 a practical solution for many emerging real-time embedded imaging applications.

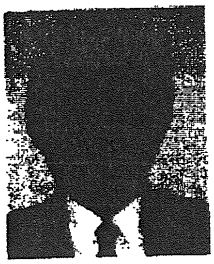
References

1. A. T. Ali and E. L. Dagless, "A parallel processing model for real-time computer vision-aided road traffic monitoring," *Parallel Processing Lett.* **2**, 257–264 (1992).
2. V. P. Kumar (Ed.), *Parallel Architectures and Algorithms for Image Understanding*, Academic Press, Boston (1991).
3. K. Guttag, R. J. Gove, and J. R. Van Aken, "A single-chip multiprocessor for multi-media: the MVP," *IEEE Computer Graphics Appl.* **12**(6), 53–64 (1992).
4. K. Preston, "The Abingdon Cross benchmark survey," *IEEE Computer* **22**(7), 9–18 (1989).
5. W. Lee, Y. Kim, R. J. Gove, and C. J. Read, "MediaStation 5000: Integrating video and audio," *IEEE MultiMedia* **1**(2), 50–61 (1994).
6. C. Weems, E. Riseman, A. Hanson, and A. Rosenfeld, "The DARPA image understanding benchmark for parallel computers," *J. Parallel Distributed Computing* **11**, 1–24 (1991).
7. K. Preston, "Benchmark results: the Abingdon Cross," in *Evaluation of Multicomputers for Image Processing*, L. Uhr, K. Preston, S. Levialdi, and M. Duff, Eds., pp. 23–54, Academic Press, Orlando, FL (1986).
8. D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM* **34**(4), 46–58 (1991).
9. G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM* **34**(4), 30–44 (1991).
10. T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM* **27**(3), 236–239 (1984).
11. J. Kim, "Towards more efficient domain-specific image computing," Ph.D. diss., Dept. of Computer Science and Engineering, Univ. of Washington, Seattle, WA (1995).
12. S. S. Wilson, "One-dimensional SIMD architectures-The AIS-5000," in *Multicomputer Vision*, S. Levialdi, Ed., pp. 131–149, Academic Press, San Diego, CA (1988).



Jihong Kim received a BS in computer science and statistics from the Seoul National University, and MS and PhD degrees in computer science and engineering from the University of Washington. He is a member of the technical staff at Texas Instruments' Corporate Research laboratories, Dallas, Texas. His research interests include image computing, multimedia systems, algorithm design and software tools, performance analysis, real-time systems, and simulation. He is a member of the Institute of Electrical and Electronics Engineers Computer Society and the Association for Computing Machinery.

Yongmin Kim received his BS in electronics engineering from the Seoul National University, and MSEE and PhD from the University of Wisconsin, in 1975, 1979, and 1982, respectively. From 1982 to 1986, he was assistant professor of electrical engineering at the University of Washington, Seattle. From 1986 to 1990, he was associate professor of electrical engineering. Currently, he is professor of electrical engineering, and adjunct professor of bioengineering, radiology, and computer science and engineering. Dr. Kim's current research interests include multimedia, image processing and computer graphics, medical imaging, telemedicine, picture archiving and communications systems, and modeling and simulation. He has more than 240 research publications and many patents, and he is the editor of eight conference proceedings. Dr. Kim and his research group have made many inventions in imaging and multimedia systems, transferring the invented technologies to industry with ten licenses, and helped commercialization of these technologies. He has been a member (chairman during 1993–1994) of the steering committee of the *IEEE Transactions on Medical Imaging* since 1990, and is a member of the editorial board of several journals. He is the director of both the Image Computing Systems Laboratory and the University of Washington Image Computing Library Consortium. Dr. Kim was awarded the 1988 Early Career Achievement Award of the IEEE/EMBS for his contributions to medical imaging. He is a fellow of the American Institute for Medical and Biological Engineering, a fellow of the Institute of Electrical and Electronics Engineers, and an ABET program evaluator for computer engineering.



Robert J. Gove received a BSEE from the University of Washington and an MSEE and PhDEE from Southern Methodist University. He is executive director of technology at Compression Labs, Inc. (CLI) and is leading the development of next- generation room video conferencing products, including video and audio compression technologies. Prior to joining CLI in late 1994 and during his 18-year career in Texas Instruments' Corporate R&D Groups, he served as technology development manager and system architect/principal investigator for several imaging technology programs, including digital micromirror displays (DMD), DSPs, CCDs, infrared systems and various video/vision/image processing algorithms and architectures. He was the chief imaging architect developing the multimedia video processor (MVP or TMS320C80) and holds the primary architecture patents for the chip. He has been a member of Institute of Electrical and Electronics Engineers, SPIE and SID.

Jeremiah Golston received BSEE and MSEE degrees from the University of Missouri-Rolla. Currently, he is the TMS320C8x video applications manager and a member of Group Technical Staff at Texas Instruments. He was one of the original C80 applications engineers and has several patents related to the C80's advanced DSP architecture and image processing algorithms. Recently, he has been focusing on video compression and videoconferencing. He implemented real-time full-duplex H.261 video compression on the C80 and now leads the development of the C8x H.320 and H.324 videoconferencing software libraries.