

Cost Modeling for Early Image Processing Applications

William H. Robinson and D. Scott Wills

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250
{whrobins, scott.wills}@ece.gatech.edu

Abstract

This paper describes a method to evaluate image applications mapped to a focal plane processing array with a single pixel per processing element. An application is specified and compiled to a common instruction set architecture. The application is then simulated to create a dynamic workload. A design space of possible execution machines is compiled using predicted performances and costs for computation, communication, and storage operations. The application workload is evaluated against the design space of execution machines based on a weighted metric of cost and performance.

1 INTRODUCTION

Inexpensive imaging chips are being incorporated into a wide range of portable products such as video and still cameras, laptops, portable data assistants (PDAs), and even children's toys. Other lower cost, higher volume applications include surveillance, user authentication, and variations of object recognition. Table 1 presents requirements for data bandwidth and processing throughput for selected image sensor arrays. The data bandwidth assumes 8 bits per pixel at 30 frames per second. The processing throughput assumes 100 operations per pixel for a given image application. Image processing workloads will soon exceed the computational and storage capabilities of current microprocessor and digital signal processor (DSP) architectures. This demand requires a different paradigm where image processing occurs near the sensor array.

Table 1: Data bandwidth and processing throughput for selected sensor arrays

Sensor Array	Image Resolution (pixels)	Data Bandwidth (Mbits/sec)	Processing Throughput (Gops/sec)
CMOS Image Sensor (Photobit) [1]	512 x 384	47.2	0.6
VGA (Olympus C-800L) [2]	640 x 480	73.7	0.9
XGA (Olympus C-1400L) [2]	1024 x 768	188.7	2.4
Advanced CMOS Image Sensor [3]	2000 x 2000	960	12

Parallel processing in the 2-D focal plane can provide increased computational performance on image applications by utilizing the natural spatial locality of image data. Processing an image in the focal plane moves the computation closer to the data acquisition to reduce the storage requirements of the system. The functionality of data acquisition, analog-to-digital conversion, image processing, and memory is integrated into a single processing element (PE). This tile element is replicated in a monolithic 2-D array to form a focal plane processor. The granularity of pixel mapping affects the performance of a particular application. If a single pixel is assigned to each processing element, applications requiring only neighboring pixel values utilize the natural spatial locality effectively for computation. However, some applications require significant communication and storage, and would not be effective with a single pixel per processing element. Currently, there is no formal classification of early image algorithms that benefit from focal plane architectures with a single pixel per processing element.

This paper discusses a method to evaluate early image processing applications mapped to a focal plane architecture with a 1:1 pixel to processing element (PPE) ratio. Section 2 provides the background for applications and architectures used by early image processing. Section 3 describes the research methodology for evaluating image applications and presents a brief case study. Finally, Section 4 concludes the discussion on image application evaluation and presents directions for future development.

2 EARLY IMAGE PROCESSING

An imaging system includes data acquisition and data processing. Image data enters the system where processing techniques convert the data into a useful format. Two functions of image processing are enhancement and analysis. Image enhancement improves the quality of the image data. Image analysis provides interpretation of image data. Each algorithm within a category requires different computation, communication, and storage costs per pixel of data.

2.1 Characterization of Applications

Image processing applications are categorized into (1) point operations, (2) local operations, and (3) global operations [4]. A point operation such as thresholding occurs at the individual pixel level. A local operation such as filtering requires knowledge of an individual pixel and its immediate neighbors. A global operation such as histogramming uses all pixel data from the image. These operations form the basis of applications found in imaging systems. Figure 1 illustrates the processing sequence required for a typical imaging system where (a) shows the processing tasks, (b) shows examples, and (c) shows the amount of data [5]. The variable b represents the number of bits per pixel, and the variable N represents the image resolution on a side. The first three stages consist of local operations while the fourth stage is a global feature measuring operation [6]. Some processing algorithms exploit the similar data types found in the first three stages to both reduce the volume of data and provide image enhancement. Feature extraction implements algorithms to provide image analysis. Focal plane systems merge these tasks into an integrated component to utilize the spatial locality naturally found in image data.

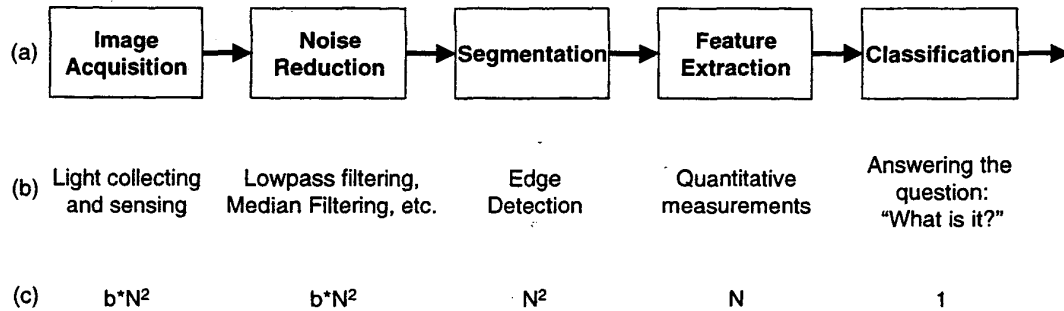


Figure 1: Typical image processing sequence (a) processing tasks (b) examples and (c) amount of data

2.2 Implementation of Processing Architecture

Most portable imaging systems use CCD or CMOS detector arrays to acquire an image. Then the pixel values are sequentially digitized and loaded into a frame buffer for processing, typically incorporating instruction level parallelism. Data parallel processing on the focal plane offers superior performance and efficiency [3]. A block diagram of an integrated smart pixel tile [7] is shown in Figure 2. This tile element is replicated to form a focal plane array imager with integrated analog-to-digital conversion and SIMD processing. The most significant issue is fill factor (the percentage of the tile that collects light) since in single-level VLSI, the detector, analog-to-digital converter (ADC), digital processing, and storage compete for silicon area. Area tradeoffs among components can influence system performance [8].

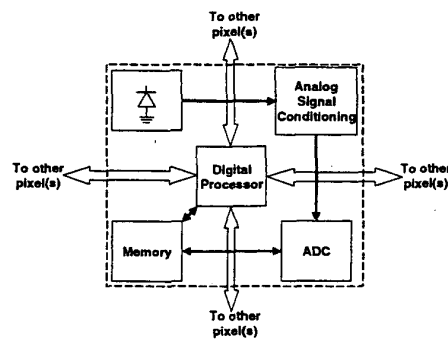


Figure 2: Programmable digital pixel

3 EXPERIMENT

Early image processing includes common filtering applications as well as image transform applications. Current research is developing a quantitative cost model to compare algorithms mapped to a focal plane architecture that uses a 1:1 pixel to processing element (PPE) ratio. Selected image applications are evaluated for effectiveness in the single pixel per processing element domain.

3.1 Methodology

Several steps are required to evaluate a selected algorithm for effectiveness with the granularity of a single pixel per processing element. An algorithm is implemented using an instruction set architecture (ISA) that corresponds to the available functional units within the digital pixel. An application is then simulated to provide a dynamic workload. The instructions are categorized into computation, communication, and storage as a percentage of the overall dynamic instruction count. Each category has an associated cost metric in terms of power, area, and performance. The category percentages are combined with category weights to provide a quantitative evaluation of an algorithm mapped into the single pixel per processing element domain. Table 2 summarizes potential applications for the single pixel per processing element domain.

Table 2: Summary of potential early image processing applications

Applications	Description
Image Transform	
Discrete Cosine Transform	<ul style="list-style-type: none"> Exploits the spatial redundancy inherent in image data, and it is a fundamental component of image compression standards.
Image Rotation	<ul style="list-style-type: none"> Two-step rotation operator to rotate the image in the focal plane by any specified angle.
Image Enhancement	
Convolution	<ul style="list-style-type: none"> Performs different filtering operations, such as shadowing, smoothing, and edge-detection.
Magnification	<ul style="list-style-type: none"> Performed in digital zooms for cameras and camcorders to enlarge by a given factor some portion of the image.
Median Filtering	<ul style="list-style-type: none"> Removes binary noise from an image while preserving spatial resolution.
Image Analysis	
Morphological Processing	<ul style="list-style-type: none"> Performs feature extraction and segmentation of binary images.

3.2 Example

This section describes initial research towards evaluating selected image applications. Algorithms for convolution and median filtering, described by Smith [9], were mapped into a 1:1 PPE focal plane architecture. Table 3 provides the dynamic workload percentages for convolution and median filtering. Operations have been categorized into computation, communication, and storage. Although both applications used a (3 x 3) filtering mask, the relative percentage of computation is greater in the median filtering application.

Table 3: Workload factors for image applications

Application	Computation	Communication	Storage
Convolution	71.70%	22.64%	5.66%
Median filtering	95.35%	3.72%	0.93%

Let us assume that the power consumed per cycle is normalized to a computation instruction. Let us also assume that a communication or storage operation requires 3x the power of a computation instruction. We could then calculate a normalized power consumption value for each application. The average power can be derived using the actual power consumed by a computation instruction. This information could be used to determine whether an application makes efficient use of battery life in a portable device.

4 SUMMARY

The demand for handling increasingly larger images at real-time frame rates will continue to saturate the available computation, communication, and storage capabilities of conventional imaging systems. Processing on the focal plane addresses the potential architectural constraints by exploiting data-parallel processing naturally found in image applications. A highly parallel focal plane system can be implemented by integrating data acquisition and data processing. A system-on-a-chip, created by a tiled monolithic array of integrated pixels, can utilize the anticipated technological improvements in fabrication. Applications can be mapped to a granularity of a single pixel per processing element. However, it cannot be assumed that all applications can be effectively translated into this fine-grain domain. Some applications may not be feasible because they require significant communication and storage costs. Others may not provide sufficient computational performance for the selected application.

This paper presents ongoing research to quantitatively evaluate early image processing applications mapped to a 1:1 pixel per processing element ratio. A methodology for investigation was described. Selected applications were simulated to provide dynamic workloads for computation, communication, and storage. Future work involves the development of categorical weights for power, area, and performance. Other applications, such as DCT and FFT image transformations, can be mapped and simulated using the 1:1 PPE instruction set architecture. With additional instructional histograms, a set of applications can be classified as efficient within the single pixel per processing element domain.

5 REFERENCES

- [1] E. R. Fossum, "Digital camera system on a chip," *IEEE Micro*, vol. 18, pp. 8-15, May-June 1998.
- [2] S. Kawamura, "Capturing images with digital still cameras," *IEEE Micro*, vol. 18, pp. 14-19, Nov-Dec 1998.

- [3] E.-S. Eid and E. R. Fossum, "Real-time focal-plane array image processor," presented at Automated Inspection and High-Speed Vision Architectures III, Philadelphia, PA, USA, November 1989.
- [4] R. S. Bajwa, R. M. Owens, and M. J. Irwin, "Image processing with the MGAP: a cost effective solution," presented at International Parallel Processing Symposium, Newport, CA, USA, April 1993.
- [5] R. Forchheimer and A. Astrom, "Near-sensor image processing: a new paradigm," *IEEE Transactions on Image Processing*, vol. 3, pp. 736-746, November 1994.
- [6] A. Astrom, J.-E. Eklund, and R. Forchheimer, "Near-sensor image processing - theory and practice," presented at Advanced Focal Plane Arrays and Electronic Cameras, Berlin, Germany, October 1996.
- [7] A. El Gamal, D. Yang, and B. Fowler, "Pixel level processing - why, what, and how?" presented at Sensors, Cameras, and Applications for Digital Photography, San Jose, CA, USA, January 1999.
- [8] W. H. Robinson, G. E. Triplett, and D. S. Wills, "Area and Performance Modeling for Smart Pixels," computer engineering technical report #2000-9471, Georgia Institute of Technology, October 2000.
- [9] M. J. T. Smith and A. Docef, *A Study Guide for Digital Image Processing*. Riverdale, GA: Scientific Publishers, Inc., 1997.