AN EXTENDED MOTION-ESTIMATION ARCHITECTURE APPLIED TO SHAPE RECOGNITION

Jason Schlessman[*], *Sankalita Saha*[†], *Wayne Wolf*[*] *and Shuvra S. Bhattacharya*[†]

[*]Princeton University
Department of Electrical Engineering
{jschless,wolf}@princeton.edu

[†]University of Maryland
Department of ECE and UMIACS
{ssaha,ssb}@eng.umd.edu

ABSTRACT

An architecture for shape recognition is presented, with emphasis on low-latency and power efficiency. This architecture is an extension of an existing architecture used for motion estimation. A number of algorithms were mapped to this architecture. Bounds related to power are given per frame for memory access rates. Face detection within CIPR CIF sequences was used as a target application, with feasible frame rates of 30fps attained. Power results for this extended architecture correlate with power consumption of the existing architecture.

1. INTRODUCTION

Shape recognition is a fundamental process associated with a diverse area of research which includes video sequence classification, tracking, and surveillance. The overall goal is to facilitate if not replace the human involvement in these operations. One of the most fundamental requirements of surveillance and security is the determination if a person is within the field of vision provided by a camera. In many cases, this is accomplished via shape recognition. In addition, there are currently many more surveillance cameras deployed in the United States than there are humans monitoring said cameras. Clearly, this makes manifest an implicit need for effective automation of monitoring.

Coupled with the need for recognition automation is a burgeoning interest in mobile surveillance and security systems, however, with the property of mobility comes the necessity for energy-efficient systems. In addition, a large percentage of related application areas are at least time-sensitive if not time-critical (i.e.: the difference between detecting a hostile intruder before potentially fatal action is taken) thus adding a need for real-time considerations such as deadlines and scheduling.

While there is an ample body of work related to implementing face detection at the software level, there is a dearth of hardware architectures using application-specific integrated circuits or processors (ASICs and ASIPs, respectively). As with many image processing operations, software solutions tend to be rather computationally complex[1][2]. Although this may not be an issue on supercomputing systems or even workstations, the interest in mobile systems mentioned previously mandates the utilization of low-power, compact systems. These algorithms typically are deployed on commodity general purpose processors. It is, however, infeasible to expect a mobile system to employ such a processor due to energy, size, and heat constraints. Additionally, specifying architectures is more oriented to our goals as the increase in specificity reduces much of the overhead inherent in generalized hardware.

As a result, we considered architectures for shape recognition. We pursued FPGA deployment, due to portability, availability, and limited cost, with the goal of transitioning to an ASIC or ASIP platform upon successful experimental results[3]. With this in mind, the rest of this paper is organized as follows: Section 2 discusses previous work related to this paper, Section 3 gives a brief overview of shape recognition as well as the algorithms we considered, Section 4 discusses our proposed general architecture, Section 5 provides power analysis for this architecture, while Section 6 details results of experiments. Finally, Section 7 concludes the paper.

2. PREVIOUS WORK

As mentioned, a substantial amount of work exists for solving the problem of recognizing faces from a theoretical standpoint[4][2]. The problem lies in maintaining performance with reduced and limited computational resources. Previously, an FPGA implementation was presented, however, it encompassed a large number of FPGA boards[5]. Other work focused on an ASIC implementation of a face detection algorithm, with no analysis of power consumption[6]. Additionally, work exists involving a new framework for object detection mapped to an embedded general purpose processor[7]. Work prior to this focusing on hardware approaches relied primarily on latency as a performance metric[8].

What is lacking in this body of work is a generalized power analysis of shape recognition architectures, in addition to a cost-effective solution to the desire for mobile detection. It is important to note, however, the similarities between shape recognition and motion estimation architectures, for which architectures have been pursued[9][10]. Also, power analysis has been performed for these [11]. Before providing our architecture and power analysis, we present an overview of shape recognition in the following section.

3. SHAPE RECOGNITION ALGORITHMS

In general, shape recognition algorithms belong to one of two classifications: feature-based and template matching. Feature-based algorithms recognize shapes by considering geometrical information available within an image. For example, in face detection,

[*]This work was supported by NSF grant #CCF-0324869
[†]This work was supported by NSF grant #0325119

feature-based algorithms attempt to derive positional information of facial attributes such as eyes and mouth. With template matching, the input image is compared with a collection of reference shape images. This comparison effectively consists of a correlation operation giving a match "score" between a particular mask m and a correspondingly-sized region of the image i, that is:

$$Score(x, y) = \sum_{s} \sum_{t} i(s, t) * m(x + s, y + t) \qquad (1)$$

where s and t cover the region for which the image and mask overlap. A running tally is kept of the location and mask for which highest correlation occurred. In the case of allowing for detecting multiple instances of shapes, a threshold is employed such that coordinates having scores exceeding said threshold are retained. Since our focus is on low-power and low-latency architectures, approaches leading to a lower degree of complexity are of paramount importance. As template-based algorithms tend to be of lower complexity with higher accuracy, we considered their utility[12].

We note that shape recognition composed of template-based matching with a relatively simple operation such as correlation is in many ways similar to that of block-matching motion estimation. As a result, we pursue algorithms associated with this technique, specifically those providing favorable results in previous work[11]. These are Full Search, One-dimensional Full Search, Three-Step Search, and Modified Log Search. All algorithms operate based on the metric of image size (W x H), mask size (M x M), and the number of considered overlays of the mask to the image, the last of which varies between algorithms.

Full search performs correlation between every possible overlay of the mask to image. As expected, it is exhaustively accurate, however, it is also computationally complex. The one-dimensional full search algorithm attempts to reduce this complexity by reducing the number of candidate templates. The best match on a given row is determined, followed by a calculation of the best match on the column of the previously found best match. The three-step procedure chooses a window of templates for correlation, which is then shifted to the center of the point with highest correlation score, and repeated until an optimal location is found. Modified log search is similar to three-step search, adding more search points and centerpoint criteria for searching.

The primary distinction between each of these algorithms is the number of areas searched for a given image. Also, aside from the first two algorithms, there is data dependency inherent in these procedures, which reduces potential for exploiting task granularity. This will also likely mandate additional control hardware. In the next section, we address these distinctions and present our proposed architecture.

4. PROPOSED ARCHITECTURE

The architecture proposed attempts to address the aforementioned demands for mobile shape recognition, and is presented in Figure 1. This architecture is an extension of that proposed by Yang et al. [11]. The architecture consists of two main memories for frame and mask storage, down and upsampling hardware, address generation hardware, and a number of processing elements (PEs).

Due to the use of FPGA targets, we model the entire system on-chip. That is, we do not consider off-chip memory. Because of our stated interest in ASIC extensions, however, we focus on appropriate designs. Therefore, we model two large memories to hold an entire frame and to hold all masks which are to be processed. As will be discussed in the following section, our design metrics are based on memory accesses, data flow rate and number of operations performed. This architecture provides the potential for reducing the number of operations performed via its downsampling hardware. The number of operations decreases linearly with an increase in sampling rate, however, this is at the expense of recognition accuracy.

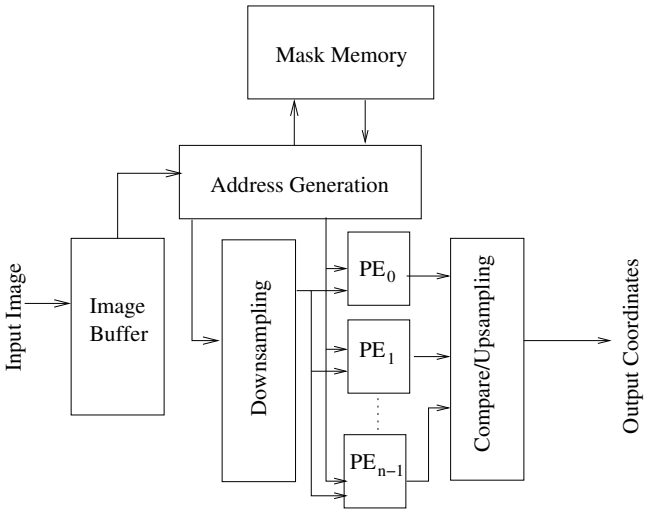

Fig. 1. Proposed general shape recognition architecture

In the interest of exploiting granularity, multiple PEs are utilized. The number of PEs is determined upon fitting the other hardware onto the target deployment with only one PE. A block diagram of a PE is shown in Figure 2. The PE contains buffers for an individual sub-image and mask. These buffers recognize the restrictive aspects of storing several copies of an image on-chip, as well as the entire mask. The contents of these buffers are fed to a processor array to perform correlation. This array consists of a number of smaller processing elements which perform multiplication and addition, with an additional row of adders after the final multiplication row. The use of this systolic array helps to provide for efficient data flow as well as consistent resource utility, while also reducing excess control and interconnect due to its autonomous nature.

The address generation hardware in this architecture is used to reduce memory accesses by performing overlap detection. For many of these algorithms, some overlap may occur between sub-images examined. Therefore, rather than reloading an entire sub-image for each mask, the data is reused whenever possible. For example, in the case of full search shape recognition, consider the image as consisting of rows with elements of size equal to that of the masks. Then, at the beginning of each such row, an entire element must be loaded into the PEs image buffer. For the remainder of iterations within that row, only the successive column must be loaded into the image buffer, due to overlap. Employing the generators in this case significantly reduces the number of memory accesses, thereby reducing power.

Upon completion of correlation, the determined score is then compared both with a threshold for multiple shape recognition and the current highest score. This is accomplished by the compare/upsampling hardware. For appropriate scores, their values as

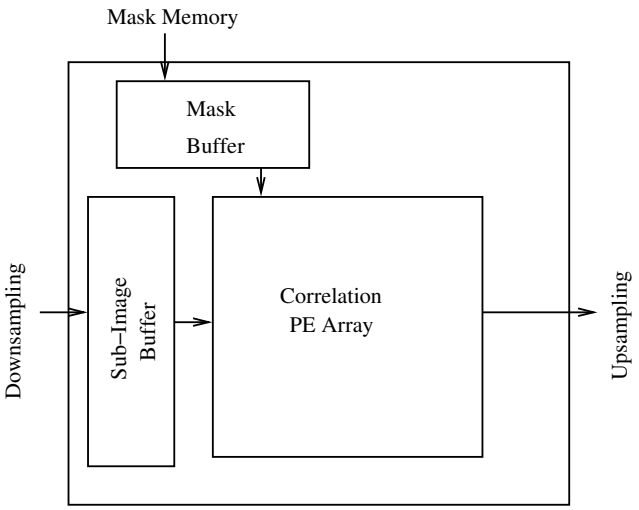

Fig. 2. Proposed processing element

well as the center point of their location on the image are saved locally, with the points upsampled and output as needed.

5. POWER ANALYSIS

For a mapping of each of the aforementioned algorithms onto the proposed architecture, high-level power analysis was performed by Yang et al. We apply these derivations to our extended architecture, while adding necessary extensions as needed. Our overall goal is insight into the effectiveness of this extended architecture as well as those algorithm mappings offering the best compromise of power and performance. We focused on interconnect power, buffer power, and processing element power as a sum of relevant power consumption. In the interest of design exploration, the factors which are altered by algorithm selection were focused upon. The first of these is the number of memory accesses. Transferring data from memories to buffers is expensive, both in terms of power and latency. This becomes an even larger problem for ASICs and ASIPs, as we expect the memories to be off-chip in these cases. By nature of their design, some algorithms will require fewer accesses than others, yielding a lower amount of power dissipation. Next, the size of the correlation units within the PEs will also affect power, as well as the related value of the number of operations performed by these units. In the interest of having insight into general performance of algorithms mapped to the architecture, we consider bounds on these metrics. For each algorithm, we determined bounds on the input and output data rates for the image and mask buffers. The mask input and output rates are identical, since each mask must be processed. Hence, the rates for the mask buffer are defined as:

$$f_{m\_buf\_in} = f_{m\_buf\_out} \cdot \#Sub\_Img \cdot N^2 \cdot \#Masks \quad (2)$$

The image buffer input rate is characterized by the amount of image data required for processing a given sub-image. This is affected by the algorithm employed, and potential for data reuse due to overlap. The previously mentioned scenario for full search applies here. In this case, the image buffer must input at the following rate:

$$f_{i\_buf\_in} = f_{ps}(\frac{W}{M} \cdot M^2 + (\frac{W}{M-1} \cdot M)) \quad (3)$$

Finally, the image buffer output rate is defined as:

$$f_{i\_buf\_out} = f_{ps} \cdot \#Sub\_Img \cdot \#Overlays \cdot \#Masks \quad (4)$$

The number of overlays to consider is the only variable in this equation which is dependent upon algorithm selection. Bounds on this quantity are presented in Table 5, where M represents the number of overlays in a sub-image.

Alg	#Overlays
FS	$(M)^2$
1D	$3M-1$
3S	$1+8log_2(M/2)$
ML	$1+6log_2(M/2)$

Table 1. Data transfer rate variables

6. EXPERIMENTAL RESULTS

We considered face detection in sequences of CIPR CIF format as a target application mapped to the proposed architecture. These mandate image buffers of size 360x288. The PE buffers were sized such that the frames were composed of four sub-images. Thirty distinct masks composed the mask set, with scaling provided by the sampling hardware. For each algorithm, power was acquired using these parameters and the equations discussed in the previous section. We assume that the main components of power consumption are due to interconnect, processing elements, and memory accesses. Other power sources, including the operation power for address generation, are considered negligible, and not included in our calculations. The power for a single frame of CIF video is shown in Figure 3. As expected, the full search algorithm consumes a markedly greater amount of power than the other algorithms. For this reason, the results were normalized with respect to the full search values. It should be noted that these results correlate with previously published results of Yang et al. for motion estimation.

7. CONCLUSIONS

This work pursued architectures for hardware implementations of shape recognition algorithms. An extension to an existing architecture for a power-efficient block-matching motion estimation was developed. Power consumption, frame rate, and accuracy were design criteria, with emphasis on power minimization through reduced memory accesses and lower hardware complexity. Extensions to derived bounds were used to determine system power consumption for a face detection application operating with a frame rate of 30fps. We feel this architecture provides means to lower power with its provision for novel data addressing and multi-resolution capability.

Future goals include adaptation of this architecture for performing associated operations, building upon our existing extension. Since technology scaling puts increased importance on leakage power, an analysis of leakage will be pursued, with focus on

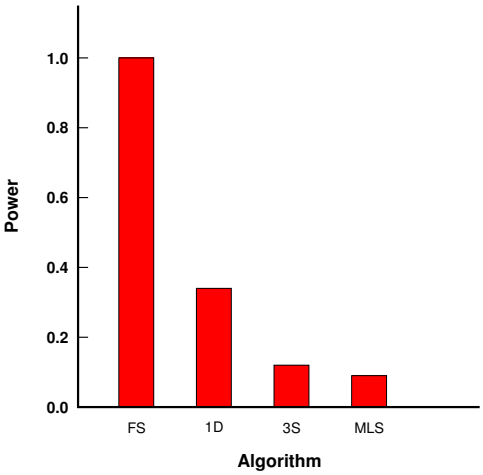

Fig. 3. CIF power consumption

the tradeoff between buffer sizing, rotational compensation via a lookup table of Gaussian functions, and the number of PEs. Finally, reliability studies for this architecture would be helpful, specifically for real-time scheduling considerations.

8. REFERENCES

[1] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1988.

[2] M-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, January 2002.

[3] W. Wolf, *FPGA-Based System Design*, Prentice Hall, 2004.

[4] P. Suetens, P. Fua, and A. J. Hanson, "Computational strategies for object recognition," *ACM Computing Surveys*, vol. 24, 1992.

[5] R. McCready, "Real-time face detection on a configurable hardware system," in *FPL '00: Proceedings of the The Roadmap to Reconfigurable Computing, 10th International Workshop on Field-Programmable Logic and Applications*, 2000, pp. 157–162.

[6] T. Theocharides, G. Link, N. Vijaykrishnan, M. J. Irwin, and W. Wolf, "Embedded hardware face detection," in *Proceedings of the 17th International Conference on VLSI Design*, 2004, pp. 133–138.

[7] P. Viola and M. Jones, "Robust real-time object detection," in *Proceedings of IEEE workshop on Statistical and Computational Theories of Vision*, 2001.

[8] B. S. Farroha and R. G. Deshmukh, "A novel approach to design a massively parallel application specific architecture for image recognition systems," in *Southeastcon '95 Proceedings*, 1995, pp. 293–299.

[9] K-M. Yang, M-T. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 1317–1325, October 1989.

[10] S. Dutta and W. Wolf, "A flexible parallel architecture adapted to block-matching motion-estimation algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 74–86, February 1996.

[11] S-Q. Yang, W. Wolf, and V. Narayanan, "Power modeling of motion estimation VLSI architectures," in *Proceedings of the 5th Workshop on Media and Streaming Processors*, 2003.

[12] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1042–1052, october 1993.