USING ATTENTION TO ENHANCE EFFICIENCY IN VIDEO-BASED COMPUTER SYSTEMS

A Dissertation in
Computer Science and Engineering

by

Yang Xiao

© 2014 Yang Xiao

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2014
The dissertation of Yang Xiao was reviewed and approved* by the following:

Vijaykrishnan Narayanan  
Professor of Computer Science and Engineering  
Dissertation Advisor  
Chair of Committee

Yuan Xie  
Professor of Computer Science and Engineering

Mahmut Kandemir  
Professor of Computer Science and Engineering

Brad Wyble  
Assistant Professor of Psychology

Jack Sampson  
Assistant Professor of Computer Science and Engineering

Kevin Irick  
Research Faculty  
Special Member

Lee Coraor  
Associate Professor of Computer Science and Engineering  
Director of Academic Affairs

*Signatures are on file in the Graduate School
Embedded vision systems that analyze complex scenes can bring many benefits to people’s daily lives, ranging from security surveillance, to medical help for visually impaired people, to traffic management systems. The key aspects of these systems are the processing algorithms that analyze image sequences and videos to extract useful information from a noisy background. Traditional image/video processing technologies typically apply the sliding window method to each frame for information processing, but even the potentially useful information occupies a very small portion of the input image. To achieve real-time performance, the processing units are duplicated massively to filter every possible window location of the high-resolution input image in parallel. Such a method not only consumes lots of power, but also limits the algorithm implementation on embedded systems due to the constrained computational resources and limited power supply. Alternatively, many of these algorithms are deployed on high performance servers/desktops with multiple graphics processing units (GPUs) rather than on power efficient portable devices.

As opposed to high performance machine vision platforms, human brains can process the vision tasks at much lower power consumption (approximately 20W) with limited computing resources (neurons in the brain) when exposed to a complex scene. In fact, the literature has shown that people do not perceive every object in a scene equally, but instead they prioritize. Objects with ‘outstanding’ features will be chosen from their surroundings and passed on to further processing stages such as feature extraction or recognition. This biological pre-processing has been identified as the attention stage and has been well-studied in the field of visual neuroscience in the past few decades. Several computation models have been proposed to show how attention within the brain works in a hierarchical way. With the assistance of the attention stage, only the attractive regions of the input image require further processing. This results in a much lower demand of power consumption and computing resources, which makes it possible to build an embedded machine vision system that can understand complex scenes in real time.

In this dissertation, a state-of-the-art bio-inspired attention algorithm called ‘Saliency [17]’ is studied. Its field programmable gate arrays (FPGA) prototype has been implemented to meet the real-time processing requirement. In addition, two extensions of the original attention system are proposed and
evaluated. The first extension is a video-based attention system which integrates two more computing channels, ‘flicker’ and ‘motion,’ for final attention map computing. Experiment results show that the new extended system can achieve 60% power savings on image test cases and 50% savings on video test cases when it is used as an LCD power management. The second extension considers the task influence in the vision mechanism by interpreting task-specific features as bias weights in the attention computing. Accuracy of locating a task item from noisy backgrounds has been evaluated. On average, a 12.7% improvement in accuracy can be achieved when compared to the original system. Furthermore, a comprehensive vision system composed by Saliency [17] and HMAX [54] is proposed and implemented. Its off-chip bandwidth characteristic is analyzed for operating under certain bandwidth caps. As a result, a memory bandwidth aware feedback system is developed to dynamically partition available bandwidth among a set of accelerators at a small expense of the recognition accuracy. Besides the power and performance evaluation, comprehensive user experience tests are conducted to ensure that no obvious image or video quality distortion are introduced by the proposed systems.
# TABLE OF CONTENTS

List of Figures .............................................................................................................. vii

List of Tables ................................................................................................................. ix

Acknowledgements ........................................................................................................ x

Chapter 1 Introduction .................................................................................................. 1

Chapter 2 Saliency Aware LCD Power Saving System .................................................. 6

  2.1 LED Backlight System ......................................................................................... 8
  2.1.1 LED Display Array ....................................................................................... 8
  2.1.2 LED Power Model ....................................................................................... 9
  2.2 Apply Saliency for LCD Power Saving ............................................................... 10
  2.2.1 Saliency Algorithm ..................................................................................... 11
  2.2.2 Saliency Core ............................................................................................. 13
  2.2.3 LED Dimming Strategy ............................................................................... 13
  2.2.4 Image Compensation .................................................................................. 16
  2.3 Experiments Results ......................................................................................... 18
  2.3.1 Image Compensation .................................................................................. 18
  2.3.2 Power Saving Comparison to Prior Work ..................................................... 21

Chapter 3 Extended Bio-inspired LCD Power Management System ............................. 22

  3.1 Saliency Attention Algorithm ............................................................................ 22
  3.2 Extended Bio-Inspired Display System .............................................................. 24
    3.2.1 Extended Saliency Core ........................................................................... 25
    3.2.2 Dimming Core and Compensation Core ..................................................... 28
  3.3 Experiments and Results ................................................................................... 29
    3.3.1 FPGA Resource Utilization ................................................................ ...... 29
    3.3.2 Video Distortion Verification .................................................................... 29
    3.3.3 Power Comparison .................................................................................... 32

Chapter 4 A Task-Oriented Vision System .................................................................... 33

  4.1 Model of Neuromorphic Vision ......................................................................... 35
    4.1.1 TD-Saliency Core ..................................................................................... 35
    4.1.2 SURF Core .............................................................................................. 36
    4.1.3 Comprehensive Learning Particle Swarm Optimizer (CLPSO) Algorithm ...... 37
  4.2 System Design of TDS-SURF .......................................................................... 39
    4.2.1 Architecture of TD-Saliency ..................................................................... 39
    4.2.2 Architecture of SURF ............................................................................... 41
    4.2.3 Evaluation Core ....................................................................................... 42
    4.2.4 System Integration .................................................................................... 43
  4.3 Experiment and Results ..................................................................................... 44
    4.3.1 TD-Saliency Accelerator .......................................................................... 44
    4.3.2 SURF Accelerator ..................................................................................... 45
    4.3.3 TDS-SURF System .................................................................................. 46
Chapter 5 Dynamic Bandwidth Adaptation Using Recognition Accuracy Prediction through Pre-Classification for Embedded Vision Systems ................................................................. 48

5.1 Accelerator Background .......................................................................................... 51
  5.1.1 Saliency Core ..................................................................................................... 52
  5.1.2 HMAX Core .................................................................................................... 52
  5.1.3 Classifier Core .................................................................................................. 53
5.2 Bandwidth Characterization .................................................................................... 53
  5.2.1 SHMAX Bandwidth Characterization ............................................................... 53
  5.2.2 SHMAX Bandwidth Analysis ........................................................................... 54
5.3 Bandwidth Management Policies ............................................................................ 56
  5.3.1 Straightforward Bandwidth Control Policy ($SF$ policy) ..................................... 56
  5.3.2 Bandwidth vs. Accuracy ................................................................................... 56
  5.3.3 Bandwidth Adaptation through Accuracy Prediction ($AP$ policy) ....................... 59
5.4 Experiment and Results ......................................................................................... 60
5.5 Related Work to Break Memory Wall ...................................................................... 63

Chapter 6 Conclusions ............................................................................................... 65

References .................................................................................................................. 68
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Overview of saliency algorithm</td>
</tr>
<tr>
<td>2.1</td>
<td>Different types for backlight panel</td>
</tr>
<tr>
<td>2.2</td>
<td>LED panel example and estimated normalized power</td>
</tr>
<tr>
<td>2.3</td>
<td>System overview of neuromorphic power saving methodology</td>
</tr>
<tr>
<td>2.4</td>
<td>Luminance level distribution based on chosen parameters (a): one salient region at [5, 5] on LED panel, (b): two salient regions at [5, 5] and [9, 4] on LED panel (c): light distribution of luminance of (b) after diffusing</td>
</tr>
<tr>
<td>2.5</td>
<td>Power saving vs. Salient region number</td>
</tr>
<tr>
<td>2.6</td>
<td>Power saving vs. fading range</td>
</tr>
<tr>
<td>2.7</td>
<td>Relationship between emitted luminance from backlight panel with perceived luminance at human eyes</td>
</tr>
<tr>
<td>2.8</td>
<td>Gaussian distribution of image compensation coefficients</td>
</tr>
<tr>
<td>2.9</td>
<td>Moon picture before and after dimming LED regions: [4, 2], [4, 12]</td>
</tr>
<tr>
<td>2.10</td>
<td>Mountain picture before and after dimming LED regions: [8, 2], [8, 3]</td>
</tr>
<tr>
<td>2.11</td>
<td>Beach picture before and after dimming LED regions: [6, 3], [2, 15]</td>
</tr>
<tr>
<td>2.12</td>
<td>Palm tree and island before and after dimming LED regions: [7, 6], [7, 7]</td>
</tr>
<tr>
<td>2.13</td>
<td>Average power saving and the standard deviation among 3 methods</td>
</tr>
<tr>
<td>3.1</td>
<td>Overview of extended Saliency algorithm</td>
</tr>
<tr>
<td>3.2</td>
<td>System overview of extended bio-inspired display system</td>
</tr>
<tr>
<td>3.3</td>
<td>Generic pipeline architecture</td>
</tr>
<tr>
<td>3.4</td>
<td>Pixels’ indexing to center scale by using positions from surround scale</td>
</tr>
<tr>
<td>3.5</td>
<td>Data follow examples when Generic Channel configured as flicker channel (upper) or motion channel (bottom)</td>
</tr>
<tr>
<td>3.6</td>
<td>Video frame examples after compensation</td>
</tr>
<tr>
<td>3.7</td>
<td>Average power saving and standard deviation comparison</td>
</tr>
<tr>
<td>4.1</td>
<td>Different attention location movement based on different task</td>
</tr>
<tr>
<td>4.2</td>
<td>Top down Saliency concept</td>
</tr>
<tr>
<td>4.3</td>
<td>System overview</td>
</tr>
</tbody>
</table>
Figure 4-4. Task weights example ........................................................................................................ 40

Figure 4-5. Overlap results between BU-Saliency and TD-Saliency with different weight threshold (1, 2, and 3) .................................................................................................. 41

Figure 4-6. SURF system overview .................................................................................................... 42

Figure 4-7. Result comparisons between bottom-up saliency and top-down saliency task of top row: red bottle; task of middle row: green round plate; task of bottom row: green rectangle plate .......................................................................................................................... 45

Figure 4-8. Sample results for SURF IP detection ................................................................................. 46

Figure 4-9. Experiment results between BUS-SURF and TDS-SURF ...................................................... 47

Figure 5-1. Bandwidth requirements of HD image processing [50][51][52] ........................................ 49

Figure 5-2. System Overview of SHMAX ............................................................................................. 51

Figure 5-3. Bandwidth of the SHMAX engine ...................................................................................... 54

Figure 5-4. Predicted accuracy using the dot products ........................................................................ 57

Figure 5-5. Accuracy vs. confidence level at varied image scales ......................................................... 59

Figure 5-6. HMAX extended with the bandwidth adaptation using pre-classification ...................... 60

Figure 5-7. Recognition accuracy of original, AP, and SF policies at bandwidth settings of 3.2, 6.4, and 12.8GB/s ......................................................................................................................... 61

Figure 5-8. Bandwidth profile of AP, SF policy, and the baseline when the maximum BW=12.8GB/s .......................................................................................................................... 62

Figure 5-9. Bandwidth profile of AP, SF policy, and the baseline when the maximum BW=6.4GB/s .......................................................................................................................... 63
LIST OF TABLES

Table 2-1. Saliency Resource Utilization on Virtex 6 SX475T FPGA .................................................. 13
Table 2-2. User Experience with the Images Having Saliency Location Shift .............................. 18
Table 3-1. Video Based Saliency Resource Utilization on Virtex 6 SX475T ............................. 29
Table 4-1. CLPSO parameters used for training ............................................................................ 38
Table 4-2. Resource Utilization of TD-Saliency ........................................................................... 44
Table 4-3. Resource Utilization of SURF ...................................................................................... 45
Table 4-4. Test Classes and Test Numbers .................................................................................... 47
Table 5-1. Memory of mainstream mobile devices ....................................................................... 50
Table 5-2. Bandwidth requests of the HMAX at varied scales ....................................................... 55
ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my committee chair, Professor Vijaykrishnan Narayanan, for his patient guidance and support. His enthusiasm and expertise in research tells me how to be a good researcher; his integrity and fairness shows me how to be an honest person. Without his persistent help this dissertation would not have been possible.

I also would like to thank my other committee members and the department of computer science and engineering for the support and help during my five years’ study.

The last but also the most important, I would like to thank my beloved fiancée, Huichu Liu, and my parents, Qing Yang and Bangyuan Xiao, for their endless love, instructions, and support.
Chapter 1

Introduction

Modern embedded vision systems are capable of analyzing image sequences and videos to extract the useful information in a complex scene. They can be applied to many aspects in our daily life [1][2][3][4][5]. For example, embedded cameras have been introduced into modern automobiles to improve driving safety, and they may further enable the development of self-driving cars. New high-tech glasses (e.g. Google Glass [6]) have been mounted with cameras to provide an augmented reality experience for their users. Robots wired with intelligent vision systems can perceive the surrounding circumstances for movement or direction decisions.

Traditional image or video analyzing approaches [7][8][9][10][11] use a sliding window method to scan over the whole image for face recognition or object detection. To achieve real-time processing, multiple processing units are duplicated in the system to operate at different window locations for parallel computing. Because the resolutions of input images are small, typically around 640×480 or 480×360, these computing-intensive algorithms can be mapped to embedded vision platforms. As a result, visions systems are relatively large and consume a hefty amount of power compared to the capturing camera used at the front end of the whole system.

Accompanying the development in technology, digital cameras are capable of capturing very high quality images and videos ranging from 2 megapixels/frame to 10 megapixels/frame [12]. Such dramatic increases in image size and resolution result in a huge demand of the computing resource and power supply for finishing image/video-processing tasks in real time. Achieving such demands with high-definition image sequence and video is still possible on embedded platforms. However, due to the constrained system size and power resources, the input image has to be sub-sampled to a proper size before any object detections or feature extractions can happen. A lot of details in original images will be lost due to this additional pre-processing step. Alternatively, high performance servers and desktops mounted with GPUs are adopted as back ends to do all the vision processing tasks. In this case, mobility and lightness are
sacrificed for the sake of real-time performance and processing quality. It should be noted that because the useful information normally occupies a very small portion of the input image, this method has wasted lots of computing power and resources.

On the contrary, the human brain can process the scene and understand the task more efficiently than the machine vision platforms. This happens with a much lower power consumption of approximately 20W when exposed to a complex scene and with limited computing resources (neurons in the brain). Research studies of human visual systems over the past few decades have shown that there are two fundamental paths existing in human visual systems: one is the ‘Where’ path, which guides the attention; the other is the ‘What’ path, which provides the recognition [13][14][15][16]. These two paths in the brain collaborate with each other to complete our vision tasks. When exposed to a complex scene, people do not perceive every object in a scene equally, but instead they prioritize, and the attention path will pre-filter every potential object in the scene. Objects with ‘outstanding’ features will be chosen from their surroundings and passed to further processing stages in the ‘What path’. If it is possible to build an embedded machine vision system which operates like our vision attention mechanism, the further processing steps such as object recognition or feature extraction are only required on the salient regions of the input image. This method can not only save a lot of computing resources, but also retain the details of the identified objects.

Several neuromorphic visual models have been developed because of the recent breakthroughs in understanding the path of the primate’s visual system. As a pre-processing stage to understanding complex scenes, attention path has been well studied in [17][18]. The work in [17] has described the computation models in detail to show how attention works when people are exposed to different scenes. The attention model used in my Ph.D. work is depicted in Figure 1-1, which is based on the work of Itti, Koch, and Niebur [17].

As shown in Figure 1-1, the model is divided into three channels and four stages. Multi-scale image features are first generated from the input image, which include intensity, orientation and color. Then these image features are combined into a single topographical saliency map, from which the outstanding regions are selected for further processing in a decreasing order of saliency values. The whole
saliency system adopts a computation-efficient manner to mimic how a human brain breaks down a complex-scene-understanding problem by selecting regions of interest based on their saliency values.

![Diagram of saliency algorithm]

**Figure 1-1. Overview of saliency algorithm**

Although the proposed models can precisely mimic the human brain to perform the attention prediction, they still cannot match the human brain in terms of the overall energy efficiency. In addition, those models exhibit long processing times on modern general-purpose platforms, and their performance is even worse when running on resource-limited embedded systems. To achieve better power efficiency and real-time performance, the assistance of application-specific hardware acceleration is a necessity.

There are currently three main candidates for alternative computing platforms to fulfill special application tasks: GPU, application specific integrated circuit (ASIC), and FPGA. The GPU is composed of hundreds of processing cores [19], and its main advantage is its capacity to provide high throughput by
exploiting data-level parallelism. At its early stage, GPU was used primarily for 3D rendering. With the growing programmability, it has become widely used for accelerating computational intensive workloads such as vision processing and financial modeling. The main drawback of GPU computing is its huge power consumption. ASIC, on the other hand, is a customized design targeted for a particular use. Therefore, ASIC design is capable of delivering the best performance while consuming the least power. However, the long design cycles and high non-recurring engineering costs are the two problems that need to be addressed in a typical ASIC design. These disadvantages make it an unattractive option for hardware accelerating in research areas [20]. FPGA is a programmable design that can be configured after manufacturing based on the designer’s needs. It offers a lot of freedom for the designer to explore both the data parallelism and the architecture parallelism in hardware. Besides that, it consumes much less power than a GPU does. Moreover, the design duration from logic design to bit-stream downloading is much shorter than an ASIC design. All these advantages make FPGA an excellent candidate for algorithm prototyping and algorithm-architecture co-design.

It seems like building a machine vision system with an attention model based on FPGA is very promising, but there are still many challenges remaining. First of all, a thorough understanding of the algorithm is the foundation of a successful prototype. Code re-writing of the algorithm model is preferred for both understanding the details and creating a golden model for testing purposes. The second challenge is creating custom (intellectual property) IP cores using hardware design language. The design and validation efforts of a FPGA design are roughly the same as needed by an ASIC design. However, there are several tools [21][22][23] on the market aiming to ease the design complexity by abstract hardware designs to high-level languages. The code’s conciseness and efficiency of complex logic designs are still unsatisfying. Additionally, simply mapping the algorithm to the register-transfer-level design usually fails to achieve the best system performance. As a result, a comprehensive analysis of the system performance is required to find the perfect balance of algorithm mapping and hardware optimization. The last, but most important, aspect is to find proper application markets for the proposed hardware platform. This dissertation addresses the abovementioned challenges accordingly and its contributions are provided below.
• Studying how an attention model guides human attention when people are exposed to a complex scene, and what kind of low features are contributed to the saliency map in our vision system.

• Prototyping the attention model with state-of-the-art FPGA technology and providing one example of daily life applications as biological-inspired (Liquid Crystal Display) LCD power management.

• Extending the prototype system with consideration of impacts of the frame-to-frame difference and the object movements in the attention map computation and analyzing the corresponding user experience.

• Exploring the task influence on the human attention mechanism when different tasks are introduced; Integrating the task information as task weights into the original system to form a task-guided hardware prototype.

• Analyzing the off-chip bandwidth of a comprehensive vision system, examining the tradeoff between the recognition accuracy and the bandwidth, and proposing a self-adapted bandwidth aware feedback system.

This dissertation is organized as follows. The ‘Bottom-up’ mechanism of saliency algorithm is described in Chapter 2, showing the details of its hard implementation, its performance analysis and power consumption. A corresponding application example, bio-inspired power management for LCD display, is proposed. Chapter 3 extends the ‘Bottom-up’ saliency to the video-based attention computing. In the extended model, two additional channels, ‘flicker’ and ‘motion’, are added in order to take into account the influence of frame-to-frame differences and moving objects in the final attention map. Chapter 4 further extends the ‘Bottom-up’ saliency to ‘Top-down’ saliency by integrating the task information into the system. Highlighted here will be the improvement of the task object locating of the proposed attention system. In Chapter 5, we perform the off-chip bandwidth analysis of a vision system composed by saliency and HMAX. A corresponding bandwidth-aware feedback system is proposed to partition the available bandwidth among a set of accelerators at the expense of accuracy for operating under certain bandwidth caps. Chapter 6 concludes this dissertation and discusses the potential research opportunities.
Chapter 2

Saliency Aware LCD Power Saving System

Modern multimedia entertainment applications are pervasive in various digital visualization platforms including personal computers, notebooks, smartphones and large screen flat panel televisions. The common among these devices is the integration of Thin-Film Transistor (TFT) LCDs for delivering rich and engaging user interfaces. In an LCD panel, the liquid crystal elements themselves do not emit light so that a separate light source is required. Traditional transmissive LCD devices use a backlight panel as the lighting source. Currently two classes of lighting sources are employed as the backlight components. The fluorescent lamp class consists of the Cold Cathode Fluorescent Lamp (CCFL) and the Hot Cathode Fluorescent Lamp. The newer class of backlighting technology is based on the light-emitting diodes (LEDs). This latter technology offers many advantages in terms of greater dynamic contrast, wider color gamut, and higher power efficiency.

Among all the display components, the backlighting system is the dominant power consumer regardless of the particular lighting technology or class. As a result, many display power saving approaches actively dim the backlight. In [24], a dynamic luminance scaling technique was applied, which modulated the luminance of the CCFL backlight source according to the current image being displayed on the LCD panel. This technique also adaptively applied the brightness compensation and image enhancement based on the class of the images being displayed. The authors reported 20% to 80% power savings achieved by the proposed system. In [25], an X-Y channel dimming strategy was proposed, in which the LCD panel was partitioned into an X-Y grid of LED zones. Within each zone, the luminance level was determined as the maximum luminance level of all pixels in the zone. For each zone, the resultant luminance was set to the minimum zone luminance across the row and the column driver of that given zone. By using this method, the number of LED drivers was linear to the sum of the block numbers along rows and columns. Besides dimming, image compensation technique was also applied to each pixel to retain the image quality as perceived by a human observer.
In addition to the active backlight dimming techniques, passive methods have been proposed for LCD power reduction. In [26], low power sensors were used to detect the user intent and behavior for energy management of laptops. The authors of [27] introduced a camera based power saving strategy that utilized a camera system to track a user’s attention and accordingly controlled the pixel luminance. The screen was brightened when the system determined that the user was paying attention, and dimmed otherwise. This approach was primarily useful for single user laptop and smartphone displays rather than LCD monitors and television displays due to the complexity of tracking multiple viewers’ eye-gaze from arbitrary distances.

Dimming the LCD at the granularity of backlight zones was presented in [28] and [29]. In [28] the luminance level was set as the ratio between the average grayscale value of all pixels and the maximum pixel value representable (i.e., 255 in an 8-bit representation). Such method, however, did not apply subsequent image compensation. The authors of [29] proposed a three steps object recognition scheme for local backlight dimming on LED panels. The first step was the object recognition, in which the largest object based on color similarity was located and its majority color was used as a bias for subsequent dimming. Secondly, local dimming was applied based on the majority color. The backlight was dimmed on the constraint that the value of the major color did not exceed the representable limit of a single color channel. Finally, image enhancement was applied to remove significantly visible artifacts on the image.

In this chapter, a visual saliency mechanism is studied for scaling the backlight of LCD displays with 2D LED backlight arrays. Moreover, an adaptive image compensation technique is also applied for preserving quality of viewer experience. Although this approach has similarities to [29], it is fundamentally different in methodology. The key distinction is that the proposed approach here utilizes a neuromorphic method which localizes regions of attention based on a comprehensive set of image features rather than simply based on color variances.
2.1 LED Backlight System

2.1.1 LED Display Array

The LED technology has become the dominant display technology compared to the LCD technology due to its higher dimming ratios, and lower operating voltages. Given these benefits, the LED display technology is considered in this work.

![LED Display Array](image)

Left: backlight panel with Edge LED
Right: back light panel with LED array

**Figure 2-1. Different types for backlight panel**

According to the backlight organization, the LED display technologies fall into two categories [30]. The first category, edge-lit panels, consists of panels that organize the LED backlights along the edges of the panel. The second category consists of panels that organize the backlights as a 2D array across the entire panel. The LEDs of the first category are installed behind the edges of the LCD panel: one, two, or all four possible installations. A light guide is used for spreading light evenly across the LCD panel. Although edge-lit displays can be made more power efficient and ultra-thin, they are limited in applicable dimming strategies because they can only be dimmed at the panel level. This problem is solved by the 2D LED array panel inherently. On the LED panel, a dense array of LEDs is distributed across the area. To simplify the control of pixels, the LED array is divided into zones. As a result, dimming the luminance at the granularity of zones makes the 2D array LED panel a good candidate for applying region-of-interests algorithms such as ‘Saliency’. An example of a LED backlight panel is shown in
In this case, the LED panel is divided into \( M \times N \) zones, where \( M=8, N=8 \). Each zone has a chain of LEDs controlled by the same driver circuit. Hence, 576 LEDs in total are embedded with each zone containing 9 LEDs. In this chapter, all the experiments are conducted based on an LED panel model with 128 zones \((M=8, N=16)\).

### 2.1.2 LED Power Model

With an appropriate luminance distribution model, the power consumption can be derived based on the luminance of the backlight panel. To compute the power saving, the power model from [35] is adopted, which uses the relationship between the luminance and power consumption as shown in Equation (2-1).

\[
p = (0.00347b^2 + 3.2194b - 10.576)\text{mW}
\]  

\[(2-1)\]

Here \( b \) is the average luminance in the zone and the total power of the panel is the sum across all the LED zones. The model has been calibrated so that the power is approximately 100W for a typical 40 inch LED-TV [36] in the case that all the backlight LEDs illuminate at full level. Furthermore, the corresponding power estimations of luminance levels from 100% to 50% are computed, as shown in the right side of the figure.
**Figure 2-2.** This result reveals the opportunities of achieving substantial power saving by using the zone dimming technique.

**Figure 2-3.** System overview of neuromorphic power saving methodology

### 2.2 Apply Saliency for LCD Power Saving

The overview of the saliency adaptive LED panel control system is depicted in **Figure 2-3**. The system is partitioned into two paths. The first path, highlighted in red, includes the saliency accelerator which processes the entire frame to identify the most salient regions and corresponding LED zones. It modulates the luminance level in the zones that overlap with the salient regions. The second path, highlighted in blue, performs pixel compensation within the saliency modulated regions. Among these three processing modules (saliency, dimming, and compensation), the saliency block is the most computing intensive part, which occupies 98% of the total operation time. Simply using the software implementation fails to achieve...
30 frames per second (fps) processing speed. Therefore, the saliency algorithm is implemented in a custom accelerator as shown in the upper part of Figure 2-3. The dimming module, bottom part in Figure 2-3, is responsible for computing the distance between the LED zones (total 128 zones) and their closest salient LED zone and assigning the luminance level accordingly. This will be explained in Section 2.2.3. The compensation module performs image compensation. It takes in 32 pixels’ positions at one time, computes their distances to the saliency location in the image, picks up a proper Gaussian weight from memory, and modifies the color value via multiplying the Gaussian weight. Details of this operation will be explained in Section 2.2.4.

### 2.2.1 Saliency Algorithm

As introduced in Chapter I, the saliency model [13] is an attention model that mimics the human’s attention pathway. There are three independent information channels in the system: intensity, orientation, and color. In each channel, the input image goes through linear filtering stage, center surround differencing stage, and across scale aggregation stage to generate a conspicuity map. The final saliency map is computed as the average across all three conspicuity maps.

The system diagram of the bottom-up saliency algorithm containing three major computing steps is shown in Figure 1-1. In the first step, the original input image is low-pass filtered and sub-sampled according to the step sizes in \{0...8\} to create a nine scales image pyramid for supporting scale invariance described in [13]. From the image pyramid, three types of conspicuity pyramids, color, intensity, and orientation, are generated scale by scale (from Scale 0 to Scale 8). The color conspicuity pyramid, including the Red-Green, Green-Red, Blue-Yellow, and Yellow-Blue, is generated by applying a pixel-wise color opponent processing between the respective color channels of the original image pyramid. The intensity pyramids is created by averaging the sum of Red, Green, and Blue channel values for each pixel on each scale of image pyramid. The last pyramid, orientation type, is acquired by projecting oriented Gabor filters on each scale of the image pyramid at certain degrees. In my case, the Gabor filters are used for detecting the angles of \[0^\circ, 45^\circ, 90^\circ, \text{and} 135^\circ\].
In the second step, each conspicuity pyramid is subsequently passed through an operation called ‘center-surround difference’ to compute feature map. The purpose of this operation is to compute the differences between the feature responses at the finer scales \( \alpha = \{2, 3, 4\} \) and the corresponding feature responses in the coarser scales \( \beta = \alpha + \gamma, \gamma = \{3, 4\} \). This across-scale difference operator is referred as \( \Theta \). Passing those conspicuity pyramids will generate the intensity feature map \( I(\alpha, \beta) = |I(\alpha) \Theta I(\beta)| \), the color feature maps \( RG(\alpha, \beta) = |(R(\alpha) - G(\beta)) \Theta (G(\beta) - R(\alpha))| \), \( BY(\alpha, \beta) = |(B(\alpha) - Y(\beta)) \Theta (Y(\beta) - B(\alpha))| \), and the orientation feature map \( O(\alpha, \beta, \theta) = |O(\alpha, \theta) - O(\beta, \theta)| \) respectively. Here, a feature map normalization operator denoted by \( \mathcal{N}(\cdot) \) is applied to each of the resulting feature maps. This operation promotes the maps that have strong value peaks compared to the mean value and suppresses those contain many similar peaks.

The third step is a sum-up procedure across scales; this operation is called ‘across-scale addition’, donated as \( \oplus \). It will reduce the feature maps at all scales to one single center scale (4th scale in the case of a pyramid with 9 scales). Three of them, aka ‘conspicuity channels’, are composed at scale \( \sigma = 4 \) from the feature maps via across-scale addition. The detail is shown in the following three equations:

\[
\bar{I} = \bigoplus_{\alpha = 2}^{4} \bigoplus_{\beta = \alpha + 3}^{4+\alpha} \mathcal{N}(I(\alpha, \beta)) \tag{2-2}
\]

\[
\bar{C} = \bigoplus_{\alpha = 2}^{4} \bigoplus_{\beta = \alpha + 3}^{4+\alpha} \left[ \mathcal{N}(RG(\alpha, \beta)) + \mathcal{N}(BY(\alpha, \beta)) \right] \tag{2-3}
\]

\[
\bar{O} = \sum_{\theta \in \{0, 45, 90, 135\}} \mathcal{N} \left( \bigoplus_{\alpha = 2}^{4} \bigoplus_{\beta = \alpha + 3}^{4+\alpha} \mathcal{N}(O(\alpha, \beta, \theta)) \right) \tag{2-4}
\]

With the three separate conspicuity channels, the final saliency map is obtained from Equation (2-5)

\[
S = \frac{1}{3} \left( \mathcal{N}(\bar{I}) + \mathcal{N}(\bar{C}) + \mathcal{N}(\bar{O}) \right) \tag{2-5}
\]
2.2.2 Saliency Core

A prototype of the saliency accelerator is implemented on Virtex6 SX475T FPGA with its resource utilization shown in Table 2-1. It takes in a typical high definition image, finds the saliency regions, and identifies the corresponding LED zone on the panel. Operating at 100MHz allows it to process 31 fps even with the consideration of the image data fetching and the saliency result storing. The power consumption, estimated by Xilinx Xpower Analyze, is around 9.1W. Double accelerators are used to increase the frame rate to 60 fps. Correspondingly, the power consumption increases to 18.2W which is 18.2% of the power consumption of the LED panel. It is should be noticed that the current power consumption percentage is reported from an FPGA prototype and thus represents an upper bound of the power consumption that can be achieved in ASIC form. According to [34], the dynamic power consumption ratio from FPGA to ASIC is approximately 14 times. So roughly the accelerator’s power would only occupy a negligible portion, 1.3%, of the total LCD power.

Table 2-1. Saliency Resource Utilization on Virtex 6 SX475T FPGA

<table>
<thead>
<tr>
<th>SLICE REGS</th>
<th>SLICE LUTS</th>
<th>BRAM36S</th>
<th>BRAM18S</th>
<th>DSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saliency Core</td>
<td>159,394 (26%)</td>
<td>173,811 (58%)</td>
<td>589 (55%)</td>
<td>404 (18%)</td>
</tr>
</tbody>
</table>

2.2.3 LED Dimming Strategy

The saliency map is used to set the luminance of the corresponding LED zones. The LED zones corresponding to the “N” most salient regions in the image are set to a new luminance level, $L_{\text{initial}}$, which is determined by scaling the original luminance by a pre-determined fraction. This fraction is set based on required power savings. Other LED zones within a defined fading range of these salient LED zones receive a luminance value that scales down linearly as a function of their distance from the salient LED zones. A scaling factor $\gamma$ is used for this linear scaling. This technique results in gradual dimming around the salient
zones on the back panel. The LED zones beyond the defined fading range are not dimmed any further and are assigned the same luminance as those zones at the boundary of the fading distance. So the luminance of a LED zone can be represented by Equation (2-6).

\[ L_{\text{zone}} = L_{\text{initial}} - \gamma \times \text{dist}_{\text{ToCenter}}, \forall \text{dist}_{\text{ToCenter}} \in [0, \text{dist}_{\text{fading}}] \]  

\[ (2-6) \]

**Figure 2-4.** Luminance level distribution based on chosen parameters

(a): one salient region at [5, 5] on LED panel, (b): two salient regions at [5, 5] and [9, 4] on LED panel

(c): light distribution of luminance of (b) after diffusing

Figure 2-4(a) (Figure 2-4(b)) illustrates the luminance setting achieved when with parameter settings of \( L_{\text{initial}} = 80\% \), \( \gamma = 10\% \), and \( \text{dist}_{\text{fading}} = 3 \) for a single (two) salient LED zone(s) located at grid point [5,5] (grid points [4,9] and [5,5]) . The small boxes in these images depict the LED zones, and the shades are used to depict the luminance levels. The impacts of the three parameters, \( L_{\text{initial}} \), \( \gamma \), and \( \text{dist}_{\text{fading}} \) on the power savings potential are identified based on the experiments. Figure 2-5 shows the influence of different \( L_{\text{initial}} \) and \( \gamma \) values for a given \( \text{dist}_{\text{fading}} \) of 3. A higher \( L_{\text{initial}} \) value or an increasing number of salient regions, which also means more salient LED zones, in the image reduces the potential for power saving. The effectiveness of a larger value of \( \gamma \) is observed when the number of salient regions is small. Since a region receives the luminance corresponding to its closest salient region, the
impact of a large value of $\gamma$ diminishes with a larger number of salient regions. A similar behavior is noticed for larger fading distances in Figure 2-6 when number of salient regions increases.

Based on the power saving tradeoffs, the experiment results in the rest of the paper are generated by choosing $L_{\text{initial}}$, $\gamma$, and $\text{dist}_{\text{fading}}$ with the values of 80%, 10% and 3 respectively with two salient regions.

**Figure 2-5. Power saving vs. Salient region number**

**Figure 2-6. Power saving vs. fading range**
2.2.4 Image Compensation

The luminance of images received by the human eyes mainly depends on three factors: the emitted luminance of the backlight, the transmittance of the liquid crystal in the LCD panel, and the image intensity \( Y \). The perceived luminance by eyes is the product of them as shown in Equation (2-7).

\[
L_{\text{perceived}} = L_{\text{emitted}} \times \rho \times Y
\]  
\( \text{(2-7)} \)

When the backlight is dimmed, the other two factors, transmittance and image intensity should be correspondingly modified to retain the perceived luminance. Much effort has been dedicated to understanding this tradeoff between luminance and perceived image quality. In [31], Gatti proposes to change the transmittance to compensate for the reduced luminance. In [32], a piecewise smoothing function based on the pixel luminance of input image is used for pixel compensation. In [33], the luminance compensation is done via the image RGB value. Inspired by [33], the perceived luminance is maintained by modifying the RGB value. In the settings, the salient LED zones are dimmed to 80% of the initial
luminance. Hence, the image compensation coefficient is set to 1.2 for the salient zones. As a result the new perceived luminance is 96% of the original image after dimming. For the neighbor LED zones around the salient zone, the compensation degree of a neighbor zone is controlled based on its distance to its closest salient LED zone similar to the settings used in the luminance dimming strategy. Consequently, the image compensation coefficients are set to 1.2, 1.3, 1.4, and 1.5 from the center to boundary. The new perceived luminance after compensation is: 96%, 91%, 90%, and 85%. Figure 2-8(a) illustrates the compensation coefficients of the neighbor LED zones around the salient zone with distances of {1, 2, and 3} zone(s) away.

![Figure 2-8(a)](image)

**Figure 2-8. Gaussian distribution of image compensation coefficients**

As shown on Figure 2-8(a), many sharp changes can be observed at the edges between two LED zones. To solve this problem, a finer level compensation is used. An inverse-Gaussian function is used for this purpose: the pixel at the center of the salient zone gets the least compensation since it has highest luminance level; the pixel beyond the three-zone away will have the same largest compensation level since they get dimmed most. The finer granularity compensated image is showed in Figure 2-8(b) with a much smoother transition.
2.3 Experiments Results

2.3.1 Image Compensation

For quantitative metrics, the saliency maps are plotted for all the images, where the most salient points remain unchanged for 95% of the images, while 4% of the images have only a small translational shift in most salient points without significant distortion. A small fraction (~1%) shifts the saliency to a new location. Even for these images, 15 users are picked to perform a visual inspection. Every user was asked to pick one of the three choices ‘different, dimmed, and same’ for 20 image pairs (original and compensated). As indicated by the test result, the visual perception was not significantly distorted (see Table 2-2). For the other images with no saliency shift there was not an observable visual distortion also. Several images, Figure 2-9, Figure 2-10, Figure 2-11, Figure 2-12, are randomly chosen to provide a qualitative assessment of image distortion. The image sets are shown here and the sub-pictures (a), (b), (c), and (d) represent the original image, the saliency maps, the dimmed image, and the compensated image respectively. With the help of color compensation, no significant difference can be noticed between the (a) and (d).

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Different</th>
<th>Dimmed</th>
<th>Same</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.33%</td>
<td>41.33%</td>
<td>54.33%</td>
</tr>
</tbody>
</table>

Table 2-2. User Experience with the Images Having Saliency Location Shift
Figure 2-9. Moon picture before and after dimming LED regions: [4, 2], [4, 12]

Figure 2-10. Mountain picture before and after dimming LED regions: [8, 2], [8, 3]
Figure 2-11. Beach picture before and after dimming LED regions: [6, 3], [2, 15]

Figure 2-12. Palm tree and island before and after dimming LED regions: [7, 6], [7, 7]
2.3.2 Power Saving Comparison to Prior Work

Besides computing the power from the proposed model, the power consumptions are also estimated based on the methods presented in [28], [29]. In Figure 2-9, the power savings the proposed system has achieved is 64.96% compared to 23.60%, and 39.62% for [28] and [29] respectively. While for the mountain scenario in Figure 2-10 the power saving is 67.45% with the proposed method, 46.31% with [28], and 51.58% with [29]. Beach picture, Figure 2-11, gives power saving at 66.65%, 21.24%, and 39.86% respectively. The last picture, Figure 2-12, shows the similar trend with power saving at 65.22%, 34.44%, and 44.37% respectively. Figure 2-13 displays the comparison of average power saving among three methods using 3000 HD test images. We can clearly notice not only does the saliency method can achieve the higher saving ratio but also it gives the most stable power saving with 1.26% standard deviation. In general, the proposed model provides a reasonable power saving on different image types.

![Power Saving Comparison](image_url)

Figure 2-13. Average power saving and the standard deviation among 3 methods
Chapter 3

Extended Bio-inspired LCD Power Management System

Chapter 2 has described the saliency algorithm, its hardware implementation, and relative application as LCD power management. In this chapter, the saliency core is extended with two additional processing channels, ‘motion’ and ‘flicker’, to account the impact of the differences of the continuous frames and object moving directions in saliency map computing when it is applied to videos. A new re-configurable FPGA image/video processing macro is also proposed for the purpose of easing the hardware design complexity. At end of this chapter, the extended system is evaluated and compared with the original system for the user experience and the system’s power saving.

3.1 Saliency Attention Algorithm

The extended saliency algorithm used in this chapter originates from the work in [13] and [18]. In addition to the three channels (intensity, orientation and color) of the bottom-up saliency model explained in [37], two more channels, flicker and motion, are introduced here. The system hierarchy, shown in Figure 3-1, is composed of five channels, in which each channel consists of four stages. The extended saliency algorithm with the two additional channels is explained in the following.

In the first step, the original image (in intensity and chrominance format) is filtered by a low-pass filter at every channel and further sub-sampled based on the scale sizes from 0 to 8. As a result, within one channel, an image pyramid containing nine image scales will be created based on the channel type. The flicker channel in this stage goes through a pre-processing step which computes the absolute value of luminance difference for each pixel between the previous and current frames. The orientation channel and the motion channel use a post-processing step to extract the features of interest (edge orientation/motion direction) from their image pyramids.
Second, the feature pyramids are computed from the image pyramids at each channel by applying center-surround differencing (CSD) operation and normalization. In the CSD operation, the difference is computed pixel by pixel between center scales \( c \in \{2, 3, 4\} \) and corresponding surround scales \( s = c + \delta \) with \( \delta \in \{3, 4\} \). This cross-scale differencing is denoted by \( \Theta \) and the differencing result of each channel is computed as shown in Equation (3-1) ~ (3-6):

\[
I(c, s) = |I(c) \Theta I(s)|
\]

\[
RG(c, s) = |RG(c) \Theta RG(s)|
\]

\[
BY(c, s) = |BY(c) \Theta BY(s)|
\]

\[
F(c, s) = |F(c) \Theta F(s)|
\]

\[
O(c, s, \theta) = |O(c, \theta) \Theta O(s, \theta)|, \theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}
\]

\[
M(c, s, D) = |M(c, D) \Theta M(s, D)|, D \in \{\uparrow, \downarrow, \leftarrow, \rightarrow\}
\]
Third, the feature pyramids are converted into conspicuity maps at channel granularity and they are listed in Equation (3-7) ~ (3-11). A normalization step, denoted as $\bar{N}(\cdot)$, happens first and it consists of two steps: (1) finding the global maximum $M$ of the input map and the average of all other local maxima $\mu$; (2) multiplying the input maps by $(M-\mu)^2/M$. This operation promotes the maps that have strong value peaks compared to the mean value and suppresses the maps contains many similar peaks. Then the center-surround addition (CSA, denoted as $\bigoplus$), similar as CSD, will be applied to the feature pyramids with centers $c \in \{2, 3, 4\}$ and corresponding surrounds $s = c + \delta$ with $\delta \in \{3, 4\}$. All the feature maps in each channel will be resized to the feature map size at Scale 4 (the medium scale of Scale $\{0..8\}$) to generate the conspicuity map.

$$\bar{T} = \bigoplus_{c=2}^{3} \bigoplus_{s=c+3}^{c+4} \bar{N}(I(c, s))$$  \hspace{1cm} (3-7)

$$\bar{C} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} (\bar{N}(RG(c, s)) + \bar{N}(BY(c, s)))$$  \hspace{1cm} (3-8)

$$\bar{F} = \bigoplus_{c=2}^{3} \bigoplus_{s=c+3}^{c+4} \bar{N}(F(c, s))$$  \hspace{1cm} (3-9)

$$\bar{O} = \sum_{\theta \in \{0, 45, 90, 135\}} \bar{N}(\bigoplus_{c=2}^{3} \bigoplus_{s=c+3}^{c+4} \bar{N}(O(c, s, \theta)))$$  \hspace{1cm} (3-10)

$$\bar{M} = \sum_{D \in \{\uparrow, \downarrow, \leftarrow, \rightarrow\}} \bar{N}(\bigoplus_{c=2}^{3} \bigoplus_{s=c+3}^{c+4} \bar{N}(M(c, s, D)))$$  \hspace{1cm} (3-11)

The last step is computing the saliency map. In this step, all the five conspicuity maps from color, intensity, orientation, flicker, and motion channel will be normalized and averaged into a saliency map.

$$S = \frac{1}{5} (\bar{N} (\bar{T}) + \bar{N} (\bar{C}) + \bar{N} (\bar{O}) + \bar{N} (\bar{F}) + \bar{N} (\bar{M}))$$  \hspace{1cm} (3-12)

### 3.2 Extended Bio-Inspired Display System

The entire saliency adaptive LED panel control system is depicted in Figure 3-2. As illustrated, there are three major components. The Saliency Core is responsible for computing the salient positions from the input image, and it has five channels as indicated in Section 3.1: intensity, color, orientation,
flicker, and motion. Here a *Generic-Pipeline Channel Architecture* is proposed for all the channels and it will be explained in detail later. Another two components are *Dimming Core* and *Compensation Core*. As the name suggests, the *Dimming Core* will dim zones of LED panel based on the position of salient LED zones passed from the *Saliency Core*. The *Compensation Core* takes care of color compensation of each pixel and sends the resultant image to the LCD panel.

![Figure 3-2. System overview of extended bio-inspired display system](image)

Blue path: pixel data flow; Red path: saliency control flow

### 3.2.1 Extended Saliency Core

The *Saliency Core* takes in the input pixel stream and generates a saliency map based on biological response from low level features. The saliency map and its salient regions will be mapped to an 8×16 array based on the zone partition of the current LED backlight panel. The positions of most salient zones will be passed to the *Dimming Core* and the *Compensation Core*.

From left to right in **Figure 3-2**, the *Image Pre-processing* module is the first module the pixel stream will go through. It decomposes the pixel stream into intensity and chrominance streams for later usage.

The *Generic-Pipeline Channel Architecture*, used as channel processing component, is the second part of the *Saliency Core*. There are six micro function blocks in this pipeline as shown in **Figure 3-3**. Their details are explained from left to right in the following part.
The first part is frame differencing block. It buffers the pixel stream of the current frame and in the meantime creates the pixel difference with the reference to the previous frame.

The second is the pyramid generation block. A pyramid needs 9 scale instants, Scale 0~8 from the input image with sizes ranging from 256×256 to 1×1, by down-sampling every other pixel from one scale to another scale. In this design it was wisely implemented through one dimension down-sampling block. When an image comes in as a stream of pixels, it will pass the X-direction processing first to down-sample the input in the row dimension and then the Y-direction processing in column dimension. The output streams of Y down-sampling blocks are saved into memory for later use. In total, 9 X-Y processing pairs and small buffers are used in this block.

The third block is the Gabor filter block and it is used for edge detection. In this saliency algorithm, four orientations {0°, 45°, 90°, 135°} are needed; therefore four dedicated filters are used.

Motion filter block is the fourth block in the generic channel. It is a post-processing step after the image pyramid is generated. In this step, Reichardt motion detector is implemented to estimate the motion energy at four directions {↑, ↓, ←, →}.

The fifth one does the ‘center-surround differencing & normalization’ as mentioned in Section 3.1. When both the center scale and the surround scale are ready from the pyramid generation block, the computation will kick in. Because the center scale is smaller than the surround scale, resizing it to the same size of the surround scale is required before computing. Instead of wasting addition storage and logic to do the resizing, the pixel position indexes of the surround scale is used to locate the corresponding pixel in the center scale for differencing computing. The dimension ratio.
between the center and surrounding scale is fixed because it can be known in advance which two scales are chosen for CSD computing. For example, when \( c=2, s=4 \), the ratio is 1:4. The x and y positions of surround scale can be left shifted 2-bit to index the pixel of center scale. An example is shown in Figure 3-4. Normalization is done in a stream pattern. Global maximum and location maxima are computed via comparing logic with intermediate buffers. The final normalization coefficient \( (M-\mu)^2/M \) is available after the stream of differencing results between two scales is finished.

- The sixth part is ‘center-surround addition & normalization’. The normalization processing does the same job as described in last block. Because the CSA operation needs to merge all the center scales and surround scales to the medium scale with fixed size (Scale 4 with size 16×16), the resizing has to be done for both center and surround scales when they have different sizes. For the smaller size scale, the same method as shown in Figure 3-4 is used for indexing. For the larger size scale, the X-Y down-sampling pairs are used. After resizing, all the scales are added via an adder-tree to generate a conspicuity map.

\[
\begin{align*}
\text{C}=2 & & \text{S}=4 \\
\begin{array}{c}
\text{Red/green pixel with position [1, 1]/[6, 4] at surround scale is translated to [0, 0]/[1, 1] in center layer}
\end{array}
\end{align*}
\]

Figure 3-4. Pixels’ indexing to center scale by using positions from surround scale

- The controller part at bottom is responsible for the pipeline configuration. Before the computing, configuration packages are sent to the controller. The controller sends the control signals to each
computation module respectively based on the information in the package and sets the dataflow of the pipeline channel. Two examples are shown in Figure 3-5. In flicker channel, upper part of Figure 3-5, the frame differencing block is enabled and its output is set as data input of pyramid generation block. Gabor Filter and Reichardt Filter are by-passed. While in the motion channel, bottom part of Figure 3-5, the frame differencing block is disabled and Reichardt Filter is used for motion detection.

The *Addition and Averaging Core* in the *Saliency Core* will compute the final saliency map from all the conspicuity maps of the enabled channels.

The *Ranking Core* will find the most salient regions from the map and map them to the corresponding LED zones on the backlight panel.

![Diagram](image.png)

**Figure 3-5.** Data follow examples when Generic Channel configured as flicker channel (upper) or motion channel (bottom)

### 3.2.2 Dimming Core and Compensation Core

The *Dimming Core* and *Compensation Core* used in extended system are similar to the cores used in last Chapter. For their detail, please refer to Section 2.2.3 and Section 2.2.4.
3.3 Experiments and Results

3.3.1 FPGA Resource Utilization

For the prototyping purpose, an extended saliency core with one generic channel is implemented and mapped to Virtex 6 FPGA. The saliency computation is done iteratively at channel granularity. Configuration packages are sent in to the core before a particular channel computation starts. The throughput of the core is 30 fps. The resource utilization is shown in Table 3-1.

The resource utilization of a saliency system with all five channels operating in parallel is also estimated. A rough linear projection shows that such a system can be easily mapped to a DINI-Group DNV7F2B board with a 5× performance improvement.

The main component in the Compensation Core is a Gaussian memory which stores all the coefficients used for color compensation. It can be easily mapped on Virtex 6/7 FPGA via using block rams.

| Table 3-1. Video Based Saliency Resource Utilization on Virtex 6 SX475T |
|---|---|---|---|---|
| | Slice Regs | Slice LUTs | BRAM36S | RRAM18S | DSPs |
| Saliency Pipeline | 167258 (28%) | 184,734 (62%) | 589 (55%) | 399 (18%) | 807 (40%) |

3.3.2 Video Distortion Verification

When buying a TV, people consider two kinds of image quality: one is the single-static-image quality; another one is the image quality in videos. Because the original saliency system is verified in last chapter for its capability of retaining the image quality, in this chapter, tests are conducted to evaluate the video quality on the extended saliency core.

All the system settings remain the same as for the image test. While evaluating my model on videos, a ‘shimmer’ effect can be noticed when a salient object was in motion. This is because, for a same
salient object, if the Euclidean distance of itself in the previous frame and the current frame are larger than 2, which means a large distance jump of the salient zone in the backlight panel, people begin to notice the luminance zone changes in the compensated video. Due to this reason salient zones in previous frame are only allowed to shift to their neighbors, i.e. one LED zone away vertically, horizontally or diagonally, in current frame. With this constraint, a smooth salient region shift for a slow moving object can be achieved. No particular actions are done for fast moving objects because they will be out of the camera view within several frames.

However, this method does not eliminate all shimmers that occur in videos. Let us consider a very simple scenario: a yellow bottle is kept on a grey-color table with another red bottle on its right side with some distance (out of the camera view when the camera is focused on the yellow bottle). The recording is started with the camera facing the yellow bottle. When the camera turns its head slowly towards the red bottle, a new salient region will be introduced because low level features of the red bottle are outstanding from the background. Since the new salient region in the current frame is too far away from the old salient region (yellow bottle) in the previous frame, the new salient region will move towards to the red bottle within several frames because of the movement constraint introduced. As a result, in the final compensated video, people will notice a luminance trace from the yellow bottle to the red bottle. To solve this problem, rather than having step size $\gamma=10\%$, the step size is changed to $5\%$ to reduce the luminance difference between the salient zone and it surrounding zones. Correspondingly the compensation coefficients are adjusted to 1.2, 1.25, 1.3, and 1.35. In user perception test, two random video pairs, each containing the original and the compensated videos side by side, are shown to 10 students. No one noticed the shimmer phenomenon. Furthermore, a non-pre-knowledge test is also conducted. In this experiment, both left and right side of each video pair are randomly picked from the original/the compensated videos with the same contents. Four possible types of combination, original/original, compensated/compensated, original/compensated, and compensated/original, will be generated. The test subject was asked to examine the video quality difference between the left and right side without knowing the combination type in advance. 20 video pairs of each combination type are generated, grouped together, and then shown to the test subject in a shuffled order. The ‘difference rate’ is used to show how many video pairs are felt different
for one test case. For example, in the original/original case with 20 test video pairs, 10% means 2 video pairs are felt different. The difference rates of those four combination types are 5%, 10%, 20%, and 25% respectively. The result shows the video compensation on the extended saliency core can do a decent job to preserve the video quality.

Figure 3-6. Video frame examples after compensation
Left: A girl holding red bottle; Right: Bottles with different color. Sub-pictures (a), (b), (c), and (d) represents the original frame, compensated frame, dimmed frame, and salient maps respectively.

Figure 3-7. Average power saving and standard deviation comparison
3.3.3 Power Comparison

LCD power consumption of the extended system is re-evaluated on both image and video displaying cases. In addition, the power consumption is also evaluated via methods presented in [28] and [29] for comparison. 3000 high-definition (HD) images and 29 videos are tested with each method and the result is depicted in Figure 3-7. In the static image testing, a higher saving ratio can be noticed on the extended saliency method with a very low standard variation. In the video test, the proposed method does not achieve the highest power saving for the reason that a smaller step size is chosen to preserve the video quality. While in [28], the luminance level of LED zones is simply decided based on the pixels’ grayscale value ratio and no image or video quality compensation is considered. Last but not least, the extended saliency method gives the lowest standard deviation of power saving under both image and video tests. This means the saliency method can always provide concrete power saving regardless of what it is showing.
Chapter 4

A Task-Oriented Vision System

Recent breakthroughs in understanding the visual path of the human visual cortex have inspired a number of computer vision algorithms that can improve the efficiency of the image/video processing. With deeper understanding of those algorithms, biologically inspired vision systems have been the focus of intense research effort to emulate the high energy-efficiency, performance and robustness of mammalian vision systems. Previous neuromorphic vision systems have focused on either the recognition paths [38] or attention paths [39], or required significant hardware resources to do both in real time. Kestur et al. [40] used several FPGAs to embody a system that mimics both the attention and recognition paths of mammalian vision. The whole system employs the Saliency [13] and HMAX [54] algorithms developed by Itti and Mutch, respectively. In [41], a network-on-chip based parallel processor is demonstrated for vision tasks and it requires an ARM10 core and eight single-instruction multiple-data (SIMD) clusters to cooperate in order to fulfill the real-time processing requirements of bio-inspired object recognition.

A very important factor in the human vision system is the task influence. However, it has been paid little attention in previous system designs. Figure 4-1 illustrates the idea of the experiment conducted by Yarbus [42] to demonstrate the different patterns of eye fixation (attention) changes when human test subjects are asked to perform different tasks in the same scene. Figure 4-1(a) here is the original picture. When no task is assigned to the test subjects, human visual attentions are randomly located at different places in the scene as Figure 4-1(b). When tasks are assigned, the subject’s attention is focused on certain locations of image that are relevant to the given task. For example, when the subject is asked to describe the girl, attention is mainly focused on the girl rather than the food on the table in the room as depicted in Figure 4-1(c). In contrast, Figure 4-1(d) shows the case when the task is to identify the food on the table. In this chapter, the original saliency algorithm is further extended, which focuses on integrating the task influence into the system to have a better performance of locating assigned task objects.
In this chapter, a visual attention and recognition system is proposed and it ties the task influence to the digital human visual system in a straightforward way. It is also shown that by integrating the knowledge of a given task into the system, the system can pay more attention to a specified task than a system that purely depends on the biological response of low level features of a scene.

Figure 4-1. Different attention location movement based on different task
4.1 Model of Neuromorphic Vision

The neuromorphic vision system proposed is composed by three major parts, and they are described in detail here.

4.1.1 TD-Saliency Core

The saliency-based attention model originates from the work of Itti, Koch, and Niebur [13] and its prototype is shown in Chapter 2. The model mimics the early stage visual pathway in a human brain, e.g. the LGN, V1 and V2 brain regions. During processing, the low level features of color, intensity, and orientation are gathered together via a bottom-up path to predict the attention location in a scene. Borji, in [43], takes this one step further by integrating the task information into the system. As shown in Figure 4-2, the system itself contains five steps to reach the final saliency map.

First, the original image is decomposed into three feature channels: intensity, orientation, and color. The intensity channel contains the image’s intensity array; the color channel contains RG and YB arrays; the orientation channel contains four arrays generated via Gabor filtering at degrees of [0°, 45°, 90°, 135°]. Low-pass filtering and subsampling are applied to three channels’ arrays for generating intensity, orientation, and color image pyramids (9 scales per pyramid). Second, the center surround inhibition is then applied to every pixel of at each scale of the image pyramids to form the feature pyramids. Third, within each feature pyramids every scale is multiplied with its task weight, and then all the scales in each channel are summed up, and normalized to create a single feature map. Fourth, the feature map of each channel is then multiplied with their task weights, summed and normalized again to create three individual conspicuity maps. Finally, all the conspicuity maps are biased by the task weights and averaged to form the final saliency map.
4.1.2 SURF Core

SURF, developed by Bay et al [44], is a novel detector and descriptor with scale and rotation robustness. It allows the same objects to be recognized even when they are observed under significantly different lighting and pose conditions. The algorithm uses the approximated 2nd order Gaussian response
for detecting interest points in an image via utilizing the integral images and the Fast Hessian filters. Here, the algorithm is decomposed into the two stages with detailed explanation below.

1. Interest Point Detection Stage:
   - First, the integral image of the input image is computed.
   - Second, three octaves, each containing 4 scale responses, are created via Fast Hessian matrix filters with size of \{9, 15, 21, 27\}, \{15, 27, 39, 51\} and \{27, 51, 75, 99\}.
   - Third, every pixel of the two middle scales in every octave (for example, scale of 15, 21 in \{9, 15, 21, 27\}), is evaluated with its 3×3×3 surrounding cube in 3D dimension for the local maximum.
   - Fourth, the located local maximums are compared with the Hessian threshold, the points with their value above the threshold is considered as interest points (IPs) of the input image. At the same time, the scale and the image space information will also be stored for the processing in next stage.

2. Interest Point Description Stage:
   - First, the dominant orientation of every IP is located by using Haar wavelet filtering.
   - Second, for each IP, the descriptor is computed based on the IP’s dominate orientation via Haar wavelet response. The descriptor can be a 64-value/128-value vector based on whether the extended feature is required. Those descriptors will be sent to classifiers/matchers for recognition processing.

4.1.3 Comprehensive Learning Particle Swarm Optimizer (CLPSO) Algorithm

To realize task-related saliency processing, it is critical to find the proper weights associated with a task. As described in Section 4.1.1, the image pyramids generated in the second stage need to go through three stages of task processing before reaching the final saliency map. Since there are nine scales in the pyramids, 9 weights are used to combine them together into the feature map. For generating the conspicuity map, 7 feature maps (1 in intensity channel, 2 in color channel, and 4 in orientation channel) need 7
weights. At the final stage, 3 more weights are used to bias the three channels to reach the final top-down saliency map. In total, there are 17 weights used to represent the task information. It should be noticed that, in the third and fourth stage, a normalization operation is also applied. The reason for doing this is to promote the response maps that only have sparse strong feature responses while suppressing other maps with large amounts of peak response. It takes three steps to do the normalization: (1) clamp all the values in the map to a range $[0, \text{Max}]$; (2) calculate the average value $\text{Avg}$ of all the local maxima except the one with Max value; (3) multiply all the values of the map by $(\text{Max} - \text{Avg})^2$. Because the normalization operation here is not a linear process, a linear regression method is not applicable. More complex search algorithms like particle swarm optimizations, genetic algorithms, or simulated annealing algorithms should be used to find a good solution. To get the proper task weights, CLPSO algorithm in [45] is used for the task weights searching. The distance between the most salient pixel and task’s geometric center is used as a measure of how good the task weights are. Theoretically, this distance can be 0 which means the salient pixel is just at the task center. The parameters used for CLPSO are listed in Table 4-1.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max generation</td>
<td>30</td>
</tr>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Dimensions</td>
<td>19</td>
</tr>
<tr>
<td>Variable range [min, max]</td>
<td>[0, 6]</td>
</tr>
<tr>
<td>Acceleration constants $C_1, C_2$</td>
<td>1.49445, 1.49445</td>
</tr>
</tbody>
</table>
4.2 System Design of TDS-SURF

The proposed system is called TDS-SURF which stands for Top-Down Saliency+SURF. The system overview is provided in Figure 4-3 and it is composed of (1) a top-down saliency accelerator that identifies the task-related regions of interest (ROIs); (2) three SURF accelerators that extract descriptors from the ROIs; and (3) an evaluation module responsible for Euclidean distance computing and task class labeling.

![Figure 4-3. System overview](image)

4.2.1 Architecture of TD-Saliency

As discussed in Section 4.1.1 and 4.1.3, the TD-saliency module incorporates the task object information to compute the attention locations on the input image for further detail processing. There are 17 weights to control the low level feature responses for generating the final saliency map which represents the visual attention of a corresponding scene. Based on the object’s properties, different task objects are represented as different 17×1 weight vectors with elements ranging from 0 to 6. Two weight vector examples are shown in the Figure 4-4. The red dots represent the task weights for recognizing a red apple, while the green dots represent the weights for finding a green bottle. The weights that closely representing the target have higher weights, like the weight of the R/G channel of feature map for the red apple, and the weight of the 90° channel at the feature map for the green bottle.
Inspired by the suppression theory provided by Tsotsos in [46], a further analysis is performed to see how the suppression affects the task-related attention. Eight different task are tested in this experiment. 10 images purely containing the task object are used for training and 50+ extra images containing the task object and other distractors are used for testing. Suppression is gradually applied to the components whose weight value is below a weight threshold pre-set to 1, 2, and 3 respectively. In this experiment, both the task objects and the computed saliency regions in the test images are annotated by boxes (saliency boxes are placed at the center of saliency regions). Two types of information are collected for each task. First, for the first three salient regions, how many saliency boxes overlap with the task boxes are collected. Second, the corresponding ratio of the total overlap area to the area of the task box is computed. The reason for collecting the second type of information is to make sure that the overlapping happens at the main body of the task rather than at the corner. The average result across all the tasks is shown in Figure 4-5. The stacked bar diagram, to the left Y-axis, summarizes the first type information across the bottom-up saliency to the top-down saliency with different weight thresholds. A significant improvement can be seen when the top-down saliency is used for the attention computing. However, the benefit from suppressing the components by using a higher task threshold does not seem significant. The coverage line, to the right Y-axis, shows the second type of information. It can be noticed that when the weight threshold is set to 3, the...
overlap number of the 3rd salient region and total coverage begin to decline. This is because the system is over-suppressed so that a distractor with one/several same low level feature(s) will be wrongly predicted as the assigned task. For example, if the system is tuned too much to focus on the R/G channel to find a red apple it may take a distractor, like a red rectangle box as its task because of the same color feature. Based on these results, the TD-saliency is adopted without considering the task weights suppression.

![Figure 4-5](image)

Figure 4-5. Overlap results between BU-Saliency and TD-Saliency with different weight threshold (1, 2, and 3)

4.2.2 Architecture of SURF

The SURF overview is shown in Figure 4-6. The upper part stands for the interest point detection stage and it contains the integral image (II) module, Fast Hessian (HS) module, and localization (LOC) module. The interest point description stage is represented in the lower part of the graph, and its components are the orientation assign (OA) module and feature extraction (EX) module. The runtime configurations of SURF are listed below:

- Input image size: the accelerator can support different image with size of 128×128, 256×256, 512×512, 640×480, 800×600, and 1024×1024.
- Hessian threshold: a tuning knob that can determine working effort of the IP detection. A large threshold will loose the system and reduce the total number of IP found in the input image; the system will operate in an opposite way with a small threshold.

- Extended-SURF: the accelerator will generate 128-value descriptors when this setting is turned on, and will generate 64-value descriptors otherwise.

In the system, the SURF core is set as non-extended with the input image size of 256×256. The Hessian threshold is chosen from [0.001, 0.002, 0.003 and 0.004] based on the system needs.

4.2.3 Evaluation Core

The Evaluation core is the last stage of the system, and it contains three brute-force matchers implemented in software. Within each matcher, every given task object is represented as a set of descriptors from its own IPs computed by SURF. The matching begins when the descriptor extraction is done for all the IPs of the incoming ROI. For every IP detected in the ROI, the Euclidean distances are computed with all the IPs of the current-checking task class. The task class’s IP with the shortest distance is considered as the matching point to the ROI’s IP. After finding matching points for all the IPs of the input ROI, sorting is applied to those IPs based on the distance to their matching points. The first 50 IPs with the shortest distances are picked and their values are averaged to stand for the closeness between the input ROI and the

![Figure 4-6.SURF system overview](image)
current-checking task. This operation goes through all the task classes in the database, and the ROI will be recognized as the task object whose average distance is the shortest. Since the ROIs of the first three saliency regions from the TD-Saliency are analyzed, the assigned task will be considered as found in the scene when any of the three matchers has a positive response.

4.2.4 System Integration

Without reliable pre-processing stages, recognition systems need to go through the whole image exhaustively to find the right feature vector/descriptor. This is not only computationally intensive but also very power hungry. Therefore, as shown in Figure 4-3, a two level system is proposed. The first stage is the saliency core that pre-processes the input image to locate the attention locations. For each attention location a ROI window with the size of 256×256 will be used to extract the information from the original picture. ROIs will be further processed by the SURF cores and the Evaluation core to decide whether there is a task object in the image and where that object is. The baseline system, BUS-SURF (Bottom-Up Saliency+SURF), contains 1 BU saliency core and 3 SURF cores and is shown in the Figure 4-3 with all the black boundary components. The reason to have 3 SURF cores is because although the BU-saliency can’t always locate the given object in the most salient region, at most situations it can locate it in the first three salient regions. Passing the ROIs of the first three salient regions can prevent us from missing the task object when it is not treated as the first in saliency ranking.

The dedicated components of TDS-SURF are annotated in red in Figure 4-3. The major differences are: (1) for each special task, the task weights trained by CLPSO can be loaded to the system for the saliency map computing; (2) The SURF can be configured online with different Hessian thresholds.
4.3 Experiment and Results

The entire system is developed on the Dinigroup DNV6F6-PCIe multi-FPGA platform [47]. It contains six computing FPGAs, and each one is a Xilinx Virtex6 SX475T device operating at 100MHz. The board is mounted on the mother-board via a PCIe x8 link for the host-accelerator communication. The results of TD-Saliency, SURF, and the whole system are evaluated individually in this section.

4.3.1 TD-Saliency Accelerator

The saliency accelerator can be configured to the bottom-up mode or top-down mode based on the user’s need. When the task-oriented property is required for the saliency processing, the red-annotated logic components in Figure 4-3 will be turned on. The resource utilization of this accelerator on Xilinx Virtex 6 SX475T is provided in Table 4-2:

<table>
<thead>
<tr>
<th>Table 4-2. Resource Utilization of TD-Saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slice REGs</td>
</tr>
<tr>
<td>26%</td>
</tr>
</tbody>
</table>

Figure 4-7 compares the results of BU and TD-Saliency. In the pictures, three different tasks are shown: red bottle on the top, green round plate in the middle, and green rectangle plate at the bottom. In each task, two pairs of result are displayed. In the top row, Figure 4-7(a) and (c) are the outputs of the original bottom-up saliency method, while Figure 4-7(b) and (d) are the outputs of the top-down saliency. The index within the yellow circle tells the ranking of the salient regions. With the original saliency core, the salient regions located at the red bottle have the lowest ranking. When the task information is considered, the top-down system gives a significant improvement: double coverages in Figure 4-7(b) and 1st saliency ranking at Figure 4-7(d). The same trend can be also seen in the middle row and bottom row where the task is not considered as a salient region in the bottom-up saliency (Figure 4-7 (e) and (k)) but located as the salient region in the top-down saliency (Figure 4-7(f) and (l)).
4.3.2 SURF Accelerator

The SURF accelerator is developed on Xilinx Vertex 6 SX475T with its resource utilization shown in Table 4-3. The implementation of SURF is verified with the model from OpenSURF [48], and no significant interest point mismatch can be found in my hardware’s result. The Figure 4-8 shows two sample tests of the SURF system. In each test, the interest points of the object are marked out by small circles in both sides of the test image pair, and the matching pairs are connected by color lines. It can be seen that the proposed system effectively locates interest points of a given object in different poses.

<table>
<thead>
<tr>
<th>Table 4-3. Resource Utilization of SURF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slice REGs</td>
</tr>
<tr>
<td>20%</td>
</tr>
</tbody>
</table>
4.3.3 TDS-SURF System

As talked before in Section 4.2.1, the TD-Saliency has a high possibility to locate the task at the 1st saliency box or the 2nd saliency box. Based on this observation, different operating efforts are assigned to three SURF accelerators based on the saliency ranking for their input ROIs. Hessian thresholds of SURFs of the 1st, 2nd, 3rd ROI are set to \{0.0001, 0.0002, and 0.0003\} or \{0.0002, 0.0003, and 0.0004\} respectively. This is done since the 2nd and 3rd ROIs have low probabilities of locating the task object compared to the 1st ROI and as a result the less computing effort is spent on them.

Six tasks are listed in Table 4-4 for the system testing. During training, the task weights of the task object are updated in the weights library. Then the test images containing both task objects and distractors are sent to TDS-SURF for processing. Three ROIs are extracted and sent to the SURF for the descriptor generation. The evaluation core collects information that determines whether or not the task object is in the scene. In the experiment, the false positive case is not considered since the task object is
always in the test images. The average accuracy rate and the average number of computed interest points are shown in Figure 4-9. Compared to the BUS-SURF’s 59.39% accuracy rate, the accuracy rates of task locating on the two TDS-SURF’s configurations are 64.98% and 66.96%, respectively. Furthermore, around 25% computation time is saved when Hessian thresholds of 0.0002, 0.0003, and 0.0004 are used in the SURF implementation. This is due to the fact that 25.7% fewer IPs need to be analyzed in the system.

Table 4-4. Test Classes and Test Numbers

<table>
<thead>
<tr>
<th>Class</th>
<th>Test number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red bottle</td>
<td>54</td>
</tr>
<tr>
<td>Red round plate</td>
<td>51</td>
</tr>
<tr>
<td>Green round plate</td>
<td>50</td>
</tr>
<tr>
<td>Blue round plate</td>
<td>42</td>
</tr>
<tr>
<td>Green rectangle plate</td>
<td>55</td>
</tr>
<tr>
<td>Yellow bottle</td>
<td>51</td>
</tr>
</tbody>
</table>

Figure 4-9. Experiment results between BUS-SURF and TDS-SURF
Chapter 5

Dynamic Bandwidth Adaptation Using Recognition Accuracy Prediction through Pre-Classification for Embedded Vision Systems

At last, I want to talk about an architecture optimization based on the bandwidth characteristic of the vision system I build during my PhD study. Although it is an application specific system solution, we should still notice that with a better understanding of the system it is possible to resolve a problem internally rather than seeking helps of new advanced technologies from the outside.

In current decades, the rapid evolution of CMOS technology has resulted in mobile devices such as smartphones and tablets becoming more ubiquitous and essential in modern life. Consequently applications that are once targeted for powerful laptops and desktops are expected to function equally well on mobile devices. Emerging applications such as ultra-high definition video streaming, real-time augmented reality [48], and advanced biometric recognition will be standards on mobile devices in the near future. Domain specific accelerators become indispensable in satisfying the performance and quality requirements of such applications executing on energy constrained mobile devices because of their high energy efficiency and throughput.

A key design challenge is determining how to cater the diverse bandwidth requirements of the data hungry accelerators. Table 5-1 summarizes the memory size and the peak bandwidth of the mainstream mobile devices. 12.8GB/s is the typical peak bandwidth supported by contemporary mobile devices. The bandwidth requirements of streaming videos at different image quality standards at 60 frames per second are shown in Figure 5-1. Notice that the 8K imagery begins to saturate the LPDDR2 bandwidth. When the 3D rendering is required, the available bandwidth is significantly exceeded. Moreover, the problem is compounded as newer devices enable users to run several bandwidth demanding applications concurrently. Bandwidth exhaustion ultimately results in severely diminished user experiences and application usability.

Several approaches focus on increasing the overall memory bandwidth at the I/O interface by using advanced banking or physical integration techniques. For example, borrowed from the desktop and server domains, one approach is to arrange several memory chips into banks to provide several individual
memory channels. However, this technology is hard to be applied on mobile systems because the footprint and cost constraints of mobile platforms limit the integration of many discrete memory devices.

![Figure 5-1.Bandwidth requirements of HD image processing][50][51][52]

Rather than increasing the off-chip memory bandwidth, the techniques proposed in this work optimizes the bandwidth demands of each accelerator. This work uses Saliency [13] and HMAX [54] algorithms as an example to demonstrate the benefits of dynamic adaptation of accelerator parameters to manage a limited shared bandwidth. In general the tunable parameter is accelerator specific and in this case the tunable parameter is the number of scales which need to be processed. This work proposes an adaptive system that solves the bandwidth allocation problem from the accelerator’s perspective. The major new observations and contributions are as follows.

1. The observation that the recognition accuracy of the system can be traded off with the bandwidth requirements through reducing the total number of image pyramid scales to be processed.

2. The observation that the recognition accuracy of the system can be predicted through pre-classification by using the smallest image pyramid scales.
3. A feedback control mechanism that tunes the number of image pyramid scales such that
the recognition accuracy of the system adapts to the available bandwidth.

It should be noticed that the proposed work is to meet the real time processing with limited
bandwidth. When the bandwidth is a limiting factor to real time processing, the proposed work can meet
the real time processing requests with a small amount of recognition accuracy loss through dynamic
bandwidth adaptation. When real time processing is not a concern, the example visual Saliency and HMAX
recognition system can achieve the highest recognition accuracy with a longer execution time.

Table 5-1. Memory of mainstream mobile devices

<table>
<thead>
<tr>
<th>Product</th>
<th>Memory BW</th>
<th>Memory type</th>
<th>Memory Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone 5</td>
<td>8.53GB/s</td>
<td>LPDDR2</td>
<td>1GB</td>
</tr>
<tr>
<td>iPad 4</td>
<td>12.8GB/s</td>
<td>LPDDR2</td>
<td>1GB</td>
</tr>
<tr>
<td>Galaxy Tab210.1</td>
<td>12.8GB/s</td>
<td>LPDDR2</td>
<td>1GB</td>
</tr>
<tr>
<td>Kindle fire HD</td>
<td>12.8GB/s</td>
<td>LPDDR2</td>
<td>1GB</td>
</tr>
<tr>
<td>Nexus 4</td>
<td>12.8GB/s</td>
<td>LPDDR2</td>
<td>2GB</td>
</tr>
<tr>
<td>Nexus 10</td>
<td>12.8GB/s</td>
<td>LPDDR2</td>
<td>2GB</td>
</tr>
<tr>
<td>Surface PRO</td>
<td>25.6 GB/s</td>
<td>DDR3 (Dual Channel)</td>
<td>4GB</td>
</tr>
<tr>
<td>Surface RT</td>
<td>6.4GB/s</td>
<td>DDR3L</td>
<td>2GB</td>
</tr>
</tbody>
</table>
5.1 Accelerator Background

Figure 5-2 illustrates the attention and recognition system which is composed of (1) a saliency accelerator that identifies Regions of Interest (ROIs); (2) a HMAX accelerator that extracts visual features from the ROIs; and (3) a software classifier that uses the features to classify the ROI into a known object class. The object detection and recognition pipeline is collectively named SHMAX (saliency + HMAX).

The rest of this paragraph will talk about the detail of each block individually.
5.1.1 Saliency Core

Saliency [13] is an attention model that mimics the behavior of the human vision pathway when perceiving visual complex environment. The model assigns a saliency score to each image region by incorporating three independent information channels: intensity, orientation, and color. Each channel consists of three processing stages: linear filtering, center surround differencing, and across scale aggregation. Each channel generates a conspicuity map from which the final saliency map is computed as the average across all the maps.

The saliency accelerator used in this work extends the design of Kestur [53] with an additional step that ranks the top $N$ salient regions. Each salient ROI is a fixed size patch of 256×256 intensity pixels extracted from the original image. The implementation used in this study fixes the maximum number ‘$N$’ of ROIs at ten. The ROIs are simultaneously supplied to HMAX for feature extraction.

5.1.2 HMAX Core

HMAX [54] is a biologically inspired feature extractor that mimics the response of V1 cortical cells found in the mammalian visual pathway. The HMAX model consists of five stages that extract features from a 256×256 intensity patch. The first stage is a pre-processing in which the ROI is converted to a twelve-scale image pyramid. In the subsequent S1 stage each of the twelve images in the pyramid is convolved with a Gabor filter bank comprised of 12 equally spaced orientations. Next, a local invariance operation is followed in the C1 layer to find the maximum Gabor responses within a local window across two adjacent scales (thus eleven scales to the S2). The S2 stage is the computational hotspot in which the output (11 scales intermediate result) from the C1 stage is correlated with 5120 prototype patches (obtained through training) stored in the off-chip memory. The C2 stage performs a global invariance operation on the intermediate features obtained from the S2 stage. The result of the C2 stage is a feature vector from which the classification can be performed. This work extends the HMAX design in [38] with the proposed dynamic bandwidth adaptation policy that will be explained in Section 5.3.
5.1.3 Classifier Core

In the final stage of classification, the extracted feature vector is compared with trained features. A software implementation of a linear classifier is used for the SHMAX engine. The HMAX accelerator extracts a feature vector from an unclassified ROI and forwards it to the classifier for recognition. The classifier computes a dot product between the observed feature vector and the feature vectors learned offline in a training process. The same numbers of dot products are computed (each for one class) as the number of the trained classes in the database. The class exhibiting the largest dot product is the winning class and becomes the label of the ROI under observation.

5.2 Bandwidth Characterization

5.2.1 SHMAX Bandwidth Characterization

The off-chip bandwidth of the SHMAX engine includes the bandwidth for loading images to the saliency core and the bandwidth of loading prototype patches to the HMAX’s S2 stage for intermediate feature computation. The bandwidth of the saliency core is almost constant at 0.1GB/s because of the stream processing architecture of the saliency core. The bandwidth requirements of the HMAX at S2 stage are shown in Figure 5-3 when operating at 30 fps with at most 10 possible ROIs in each image. The bandwidth requirements of S2 exhibit a wide changing range from ~5GB/s (one ROI) to ~50GB/s (ten ROIs) with an average bandwidth at ~26GB/s (red line in Figure 5-3) for high definition input streams. Note that the bandwidth requirement is higher than the bandwidth supported by contemporary mobile devices.
5.2.2 SHMAX Bandwidth Analysis

A further investigation of the SHMAX system reveals the two main factors responsible for the changing bandwidth requests. The first one is the number of ROIs which is dependent on the input image. The saliency accelerator locates the ROIs based on the image content. Therefore different images will have different numbers of ROIs. The current SHMAX implementation caps the maximum number of ROI at ten.

The second source of bandwidth variation is attributed to the S2 stage of HAMX which continuously reads the prototype patches from the off-chip memory. In the design presented in [38], the C1 layer passes an image pyramid composed of eleven scales at twelve orientations to the S2 for processing. For every scale, each of the “associated” patches is buffered from off-chip memory for the correlation operation. The qualifier “associated” here refers to the patches that can be used for correlation with the current image scale. The load sizes for each scale are shown in Table 5-2.

![Figure 5-3. Bandwidth of the SHMAX engine](image)

Table 5-2 shows that the total data size to be loaded from memory is 162.8MB when processing all scales. Assuming 30 fps for a real time processing system, the bandwidth requirement for processing a single ROI is $30 \times 162.8 = 4.9 \text{ GB/s}$. The required data size decreases significantly when higher scales are
skipped at the expense of lower accuracy. For example, if processing is restricted to scales 8–11 the data size is reduced to 14.38MB, which results in a bandwidth request of 0.43 GB/s. However, accompanying the large reduction in bandwidth requirements is a loss in recognition accuracy as the intermediate features extracted from larger scales are excluded. The following sections detail a mechanism to reduce the bandwidth request with a trivial loss of accuracy.

**Table 5-2. Bandwidth requests of the HMAX at varied scales**

<table>
<thead>
<tr>
<th>Scale #</th>
<th>Scale size</th>
<th>Patch size (associated)</th>
<th>Load size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47×47</td>
<td>16, 12, 8, 4</td>
<td>22.52</td>
</tr>
<tr>
<td>2</td>
<td>39×39</td>
<td>16, 12, 8, 4</td>
<td>22.52</td>
</tr>
<tr>
<td>3</td>
<td>33×33</td>
<td>16, 12, 8, 4</td>
<td>22.52</td>
</tr>
<tr>
<td>4</td>
<td>27×27</td>
<td>16, 12, 8, 4</td>
<td>22.52</td>
</tr>
<tr>
<td>5</td>
<td>21×21</td>
<td>16, 12, 8, 4</td>
<td>22.52</td>
</tr>
<tr>
<td>6</td>
<td>17×17</td>
<td>16, 12, 8, 4</td>
<td>22.52</td>
</tr>
<tr>
<td>7</td>
<td>15×15</td>
<td>12, 8, 4</td>
<td>13.29</td>
</tr>
<tr>
<td>8</td>
<td>11×11</td>
<td>8, 4</td>
<td>5.75</td>
</tr>
<tr>
<td>9</td>
<td>9×9</td>
<td>8, 4</td>
<td>5.75</td>
</tr>
<tr>
<td>10</td>
<td>7×7</td>
<td>4</td>
<td>1.44</td>
</tr>
<tr>
<td>11</td>
<td>5×5</td>
<td>4</td>
<td>1.44</td>
</tr>
</tbody>
</table>
5.3 Bandwidth Management Policies

5.3.1 Straightforward Bandwidth Control Policy (SF policy)

Based on the bandwidth analysis of SHMAX, a bandwidth management policy is first proposed as follows: for a fixed bandwidth, the policy equally partitions the bandwidth to all the ROIs to be processed. This simple policy greedily finds the maximum input bandwidth for all the ROIs equally. For example, when input bandwidth is set to 6.5GB/s and three ROIs are found, the largest bandwidth that can be assigned to each ROI is 2.18GB/s.

This SF policy does not discriminate the ROIs and tries to achieve the highest accuracy obtainable for all ROIs given the available bandwidth. Although this method allows all the ROIs to be processed at the same time, many computing resource are wasted on the low ranking ROIs because it is less possible to locate objects at those regions when compared to the high ranking ROIs [55].

5.3.2 Bandwidth vs. Accuracy

1) Saliency ranking and recognition accuracy expectation

After saliency detection, all extracted ROIs are sent to the pool of the HMAX accelerators for concurrent processing. Collectively the HMAX stage supports processing at most ten ROIs in parallel. The behavior of the mammalian vision system suggests that at a given time higher recognition effort is expended to a small percent of the entire visual field: namely the fovea. Accordingly, the accuracy requirement of a ROI can be assigned based on its saliency ranking and the distance from the fovea.

In our test, a total of 40 object classes are trained for recognition. The highest recognition accuracy obtainable by using HMAX on 256×256 8-bit grayscale image patches is roughly 80% [54]. This value is also set as the accuracy expectation for the first salient ROI. The accuracy requirement of lower ranking ROIs is reduced by 10% until reaching the minimum accuracy expectation of 50%. In summary, the top three ROIs have higher accuracy expectations, 80%, 70%, and 60% than other ROIs with the accuracy
expectation of 50%. These accuracy expectations can be tuned to trade off bandwidth demands and recognition accuracies.

![Figure 5-4. Predicted accuracy using the dot products](image)

2) Accuracy predictions through pre-classification

A linear classifier using the weighted regularized least square method is adopted in the final classification stage. The ROI’s feature vector is classified as one of the forty object classes that has the largest response when the dot product is calculated. As a result, there are a total of 40 dot product responses for each ROI.

Figure 5-4 shows the results obtained from the Caltech 101 dataset. Eight computation profiles are designated with each having a range of image scales that will be processed. The smallest profile consists of scales 8–11 meaning that only the lowest four scales (from scale 8 to scale 11) of the image are used for pre-classification, and the largest profile consists of all scales 1–11.

From the experimental results, the dot products are classified into three groups, one positive (bar ‘one pos’), all negative (bar ‘all neg’), and multiple positive values (bar ‘multi pos’). It is observed that among the 40 classes in the database, the recognition accuracy is the highest when there is only one
positive value in the dot products for all the scale profiles, which accounts for 41.7% of the total test cases. Even when only four scales are used (scales 8–11) in the S2 computation, the accuracy rate has already reached 81%. With more scales, the accuracy becomes higher. Therefore, when there is only one positive dot product among the 40 classes, the recognition accuracy can be easily predicted no matter how many scales are used.

In the second case when the dot products are all negative, representing 57.2% of all test cases, the recognition accuracy increases as additional scales are used for recognition. The final case occurs when there are several positive values in the dot products. This case represents only 1.1% of all test cases. Furthermore, there are no such occurrences in the three largest scales (scales 3–11, 2–11, and 1–11). Thus, this work focuses on solving the bandwidth problem on first two cases.

3) Confidence Level

To further aid the detection, a new detection criterion called the Confidence Level (CL) is introduced as Equation (5-1). The CL is computed at the classification stage to determine how an all-negative-value dot product affects the recognition accuracy. It evaluates the degree of distinction between the first to the second largest dot product values.

\[
\text{ConfLevel} = \frac{\text{Value}_{\text{top1}}}{\text{Value}_{\text{top2}}} \quad (5-1)
\]

The CL is divided into five intervals as shown in Figure 5-5. The lower value of the CL means the more distinction between the Top 1 and the Top 2 classes because all the values in the dot product are negative. Two observations are made from Figure 5-5. First, for a given number of scales, the recognition accuracy decreases as the CL increases. Second, the recognition accuracy of each interval increases when more scales are involved in the S2 computation. Given a configuration of the S2, it is possible to predict the recognition accuracy based on the CL when the dot products are all-negative.
5.3.3 Bandwidth Adaptation through Accuracy Prediction (AP policy)

The SP bandwidth management policy discussed in Subsection 5.3.1 maintains fairness among all the ROIs. This sharing policy may result in wasting bandwidth to the ROIs whose potential recognition accuracy has already met the requirements. A new bandwidth adaptation policy is proposed by using the predicted accuracy rate from the dot products.

The accuracy prediction (AP policy) based bandwidth adaptation works as follows. All the qualified ROIs found in the image are sent to the pool of HMAX accelerators and processed in the S2 stage with four image scales (scales 8~11) for the least bandwidth requirements. After the dot products are computed through the pre-classification, the recognition accuracy of each ROI is predicted via its dot product. For the ROIs whose predicted accuracies have met the requirements, no additional scales are used. For other ROIs, when the accuracy expectation is not met, the unused closest scale to the current profile is scheduled for the S2 stage (scale 7 in the 1st iteration). At the same time, the extra bandwidth consumption is compared with the remaining bandwidth to prevent bandwidth overdraft. The accuracy is predicted and...
re-evaluated each time a larger scale is assigned to the ROI for S2 processing. The AP policy stops increasing the scales when any of the following three conditions is reached: 1) there is no available bandwidth; 2) all the predicted recognition accuracies are above the expectations; 3) all the image scales are used.

The saliency accelerator of the SHMAX hardware system remains unchanged; however modifications are applied between the HMAX and Classification stages to include the AP bandwidth adaptation policy. Figure 5-6 shows the modification to the system. A feedback loop is created to relay the decision to perform the additional scale processing in the S2 stage. In this figure, the S2 module has already computed scales 5-11 and is waiting for the decision from the AP process.

![Figure 5-6.HMAX extended with the bandwidth adaptation using pre-classification](image)

### 5.4 Experiment and Results

High definition 1920×1080 image is used in all experiments as it is representative of the camera capability of current mobile devices. It should be noted that the images used for the experiments are different to show the diverse behaviors of the input. Three different input bandwidth settings (3.2, 6.4, and 12.8 GB/s) are applied to the SHMAX core. The maximum/minimum accuracy requirement for ROIs is 80%/50% with a step size of 10%. The maximum permitted number of ROIs that can be processed ranges
from 5 to 10 (5 to 7 for 3.2GB/s since at most 7 ROIs can be processed with the minimum computing profile).

Figure 5-7 shows the recognition accuracy of the original design, the dynamic adaptation policy (bar AP) and the equal bandwidth allocation policy (bar SP) at maximum bandwidths of 3.2, 6.4, and 12.8 GB/s respectively. Several observations are made from the results.

First, when the maximum bandwidth is 12.8GB/s, the proposed AP bandwidth adaptation policy can achieve recognition accuracy at around 75% (red bar) when the maximum ROIs are 5, 6, or 7. When the maximum number of ROI is 8 or 9, the recognition accuracy is reduced to 73%, and the accuracy is 71% when the permitted number of ROI is 10. The AP policy achieves 75% accuracy with an average bandwidth of 7.3GB/s (computed from Figure 5-8) while the baseline achieves 79% accuracy with an average bandwidth of 21.8GB/s. That means the proposed bandwidth adaptation reduces the bandwidth consumption by \((21.8-7.3)/21.8 = 66.5\%\) with a 4% recognition accuracy reduction when the number of ROIs is 7.

Figure 5-7. Recognition accuracy of original, AP, and SF policies at bandwidth settings of 3.2, 6.4, and 12.8GB/s
Second, when the maximum bandwidth is reduced to 6.4GB/s, the recognition accuracies are 72%, 71%, 71%, 70% and 69% when the maximum possible number of ROIs are 5, 6, 7, 8, 9, 10, respectively. The average bandwidth used by the proposed AP is 5.4GB/s while the baseline consumes an average bandwidth of 26.9GB/s. The bandwidth is reduced by \((26.9-5.4)/26.9 = 80\%\).

Third, the SHMAX can always work below the maximum bandwidth with both AP and SF policies. Higher accuracy can be achieved by the AP at all three bandwidth settings because the bandwidth allocated to the ROIs can be dynamically determined. Compared to SF policy, AP policy assigns more resources to the ROIs that fail meeting the accuracy expectation. The AP consumes an average of 7.3GB/s with an accuracy of 75% while the SP uses all the 12GB/s with an accuracy of 72% when the number of ROI is five.

![Bandwidth profile of AP, SF policy, and the baseline when the maximum BW=12.8GB/s](image)

**Figure 5-8.** Bandwidth profile of AP, SF policy, and the baseline when the maximum BW=12.8GB/s
5.5 Related Work to Break Memory Wall

Package on Package (POP) is a popular method to solve the memory bandwidth problem and is employed in the iPhone 4S and iPhone 5 [49]. In this solution, multiple memory dies are stacked to form a memory package. And this package is then stacked with a logic package via the memory's ball grid pin array on the PCB board. As a result, relatively high logic to memory bandwidth (still limited by inter-package connections) can be achieved. However, many defects such as open solder connections, signal shorts, and mask misalignment can occur in the POP assembly process. Moreover, even for a well staked package, detrimental thermal issues can arise if heat generated by any of the stacked dies is not sufficiently dissipated [56].
Recently a 3D IC integration standard called ‘Wide-IO memory’ has been proposed by JEDEC [57] to solve the memory wall problem. In a wide-IO memory, two or more memory dies are stacked together. The Through Silicon Vias (TSV) are ‘drilled’ through all the dies to provide vertical connectivity for the layer-to-layer communication. Therefore, if the 3D integration technology is mature enough, a larger number of TSVs can exist in a stacked memory system to provide very high memory bandwidth. In [58] a Samsung Wide-IO memory device with four 128-bit channels provided by TSVs is described. The peak bandwidth is 12.8GB/s when operating at 200MHz. Indeed, the achievable bandwidth can be as high as ~100GB/s. However, critical issues in 3D integration including TSV cost, TSV yield, heat management, and 3D testing impede the massive production of Wide-IO memory devices [59][60].

In summary, ongoing research efforts largely focus on increasing aggregate memory bandwidth through physical integration techniques. Less effort, however, has been spent on understanding the accelerators’ bandwidth characteristics. This chapter proposes a new direction to solve the limited bandwidth problem imposed by mobile devices. By leveraging the system bandwidth characteristics, the new system can operate in real-time under only 25% of the original needed bandwidth.
Chapter 6

Conclusions

Driven by the growing market for applications that assist people who are visually impaired, enhance driving safety, and enrich the shopping experience, the development of embedded vision systems is highly beneficial to our daily lives. Traditional image/video technologies process such images or videos via brutal-force searching in order to locate the important information, which makes the algorithm mapping on embedded systems very challenging due to limited computing power and short battery life. Visual neuroscience studies have found that human brains detect the salient regions with ‘interesting’ features in a decreasing order of their saliency degrees, which has inspired the design of embedded systems for detail-preserved, real-time visual processing using the ‘biological’ attention model. However, because of their long processing time and large power consumption when running on general-purpose platform in software, the assistance of application specific hardware acceleration is required to achieve better power efficiency and real-time performance.

This dissertation studies the state-of-the-art biological attention algorithm called ‘Saliency’. Based on this algorithm, an FPGA prototype system has been implemented to mimic the primate’s vision system and meets the real-time processing requirement. The performance and power efficiency achieved by using this attention model are also evaluated and demonstrated.

In Chapter 2, the biological attention model ‘Saliency’ is studied and implemented on an FPGA platform. It is also used as a bio-inspired power management for LCD display systems. The proposed system actively dims the LED zones on the lighting panel based on the salient regions found on the high definition image sequences and videos. Correspondingly, the image compensation is applied to the saliency modulated regions for retaining the original image quality. The user experience is validated by showing 20 random image pairs (original, compensated) to every test subject of a group of 15. Experiment results show no noticeable image quality distortion between the original image and the compensated one generated by our system. Finally, around 3000 high definition images are used to evaluate the power saving potential of
the proposed bio-inspired LCD power management. Experimental results show a stable power saving potential of 65% with 1.26% standard deviation.

Chapter 3 extends the proposed attention prototype to video based applications. Instead of three channels, two more channels, ‘flicker’ and ‘motion’, are introduced to the system to include the impacts from the differences between the continuous frames and the moving objects in videos. Furthermore, an FPGA accelerating macro is proposed. It contains a set of basic image processing primers for easing the design effort of video-based systems. The purpose of this generic-channel-architecture is to provide a flexible design space and ease the design complexity when developing customized attention accelerators. Like the image quality validation test conducted in Chapter 1, a video quality validation is also conducted with ten different test subjects. Additionally, a non-pre-knowledge test is conducted by asking the test subject to examine the video quality difference in shuffled video pairs without knowing the pair type in advance. Both tests show that no significant changes are introduced to the original videos by the video-based system. On average, around 51% power savings can be achieved in the whole system when the extended prototype is applied to LCD display systems.

The other extension of the original attention system is explained in Chapter 4. Different than previous vision accelerators that focus on accelerating the computationally intense portions of algorithms, the extended model in this paragraph pays attention to the task influence in the human vision mechanism. The task influence is interpreted as task weights to bias the computing of the final saliency map. Based on this task-oriented system, a two-level vision system containing TD-Saliency and SURF is implemented. The experiment results show that the new system can achieve at most a 12.75% accuracy improvement and save 25% in terms of computational work in locating the task-related objects.

A whole vision system composed of Saliency and HMAX is proposed in Chapter 5. Due to the massive growth of mobile devices, more mobile applications are in demand to enrich the user experience. Computation intensive algorithms that working on high resolution videos require more application specific accelerators on mobile platforms for delivering real-time performance. How to guarantee the off-chip memory bandwidth required by each accelerator becomes a key challenge in these accelerator-rich mobile platforms. Rather than seeking the assistance of new technologies, this work investigates the off-chip
bandwidth characteristics of the built-in system. A corresponding bandwidth aware feedback system is proposed so that the available bandwidth is dynamically partitioned among a set of accelerators. Although there is a 4% recognition accuracy lost due to the bandwidth optimization, the whole vision system can offer smooth real-time processing by only using 25% of the original bandwidth demand.

Future work can proceed in two directions. First, it would be crucial to develop sophisticated vision theories and algorithms for accuracy and performance improvement when different tasks are introduced in a more complex environment, given that tasks have a very important influence on vision attention systems. Although this dissertation has proposed a simple method to interpret the task influence, our brain operates in a more complicated and more accurate way to represent the task information. Thus, a sophisticated model is critical to bring more insights to how the human brain relates to and chooses the low-level vision features for the assigned task. Second, it is important to develop an application specific architecture optimization for biological and machine vision systems. This dissertation has mainly discussed the comprehensive vision system for object location and recognition. Based on its off-chip bandwidth characteristic, a dynamic off-chip bandwidth adaption method has been applied to maintain the system operation under the constrained off-chip memory bandwidth. In the future, a smart vision system can have different accelerators on the same die, such that it is capable of fulfilling various vision tasks at the same time. This type of heterogeneous system will impose new design challenges onto hardware and computer architecture domains. How to wisely use the on-chip bandwidth to feed vision specific accelerators on the same die while fulfilling the operation power/performance requirements would be another direction for research.
References


[19] What’s the difference between a cpu and a gpu?


[20] Introduction to ASIC Design

http://www.ece.ncsu.edu/asic/2012/docs/Module1.htm

[21] Xilinx System Generator and HDL Coder.


[22] FPGA-based hardware acceleration of C/C++ based applications – Part 3.

http://www.eetimes.com/design/programmable-logic/4015127/FPGA-based-hardware-acceleration-of-C-C-based-applications--Part-3?pageNumber=0

[23] Catapult C Synthesis


[36] SONY 40" LED EX640 Internet TV Specification


VITA

Yang Xiao

Yang Xiao currently is a senior R&D engineer at Synopsys. He received his B. Engineer in Automation in 2005 from Huazhong University of Science and Technology. He got his Master degree in Electrical Engineering in 2008 from North Carolina State University. He did his Ph.D. under the supervision of Prof. Vijaykrishnan Narayanan and passed his defense in 2014. His research interests are reconfigurable neuromorphic design, domain specific acceleration, and vision embedded system. During his Ph.D. study, he participated several important research projects such as the DARPA Neovision II Project, the Future Stores Project (with CMU and Intel), and the NSF Expedition Project in Computing.

Publications


Talks/Demos

- Demonstration: “Systems for Recognizing and Enhancing Consumer Experiences in Future Retail Environment”, University Collaboration Office Showcase and Research@Intel Day 10th Anniversary, Jun 2012