Imperial College London of Science,
Technology and Medicine
Department of Electrical and Electronic Engineering

Context-based Image Acquisition

Jianxiong Liu

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Electrical and Electronic Engineering
of Imperial College London, September 2014
Abstract

The cost of off-chip memory access (in bandwidth, time and energy consumption) has become a major concern in the design of many hardware systems. Due to reasons such as the increasing performance gap between computing engines and memory systems, the process of data acquisition from memory has an increasingly dominant impact to the overall of the system performance. The cost in memory communication bandwidth and time consumption of data acquisition has become more significant, stalling the application and reducing its execution. Energy consumption has also become one of the main concerns to modern hardware systems, especially for embedded applications, and the energy spent on memory accessing has been reported to occupy a large proportion of the overall energy consumption of the system. All these lead to the research topic of reducing the cost of memory access in hardware systems.

Particularly for image processing systems, due to the ever growing size of image data, the task of image data acquisition poses an increasing challenge to the design of the systems. Various researches have addressed this problem of image data acquisition by exploiting the characteristics of memory structures and image processing applications. Some methods approach this problem from software perspective, changing for example the source code of the application so that the off-chip memory access is minimized; other methods approach this problem from hardware perspective, modifying the structure of memories and reorganizing the order of data transmission sequences. This thesis provides an alternative way of dealing with this problem and proposes the framework of “Context-based Image Acquisition” (CbIA) for hardware systems. Instead of accessing from the off-chip memory all image data requested by the application, the proposed framework accesses only fractions of the image and by utilizing image processing algorithms it reconstructs the missing part. This allows the proposed framework to trade computational effort for reduced cost of memory access, and ultimately trade image quality with reduced overall cost of the image acquisition process. On top of this, the proposed framework has the advantage of being independent from both the memory and the image processing application, and therefore can be seamlessly integrated into existing image processing systems.

The thesis elaborates on the proposed framework from both the algorithmic perspective and the hardware architectural perspective. A designed and implemented CbIA architecture is evaluated on reconfigurable hardware, reporting a reduction of up to 88% of communication bandwidth, 68% of time consumption, and 50% of energy consumption of the image acquisition
process, at the expense of reduced image quality (about 33 dB of PSNR on chosen benchmark images). Based on this design, this thesis also investigates on more complex algorithms at simulation level for CbIA procedures, including that for generic images as well as for specific image class. The use of more detailed modelling of images and/or domain-specific knowledge improves the ability of the CbIA procedure to trade image quality for bandwidth reduction (an increase of about 2 dB of PSNR with the same bandwidth used), with additional expenses on the computational cost. Finally, the thesis explores the impact brought by the proposed CbIA framework to the future development of hardware system.
Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.
Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Peter Y.K. Cheung, who has supported my research with his immense knowledge and experience. He has always been a great advisor, mentor, and a dear friend of mine.

I would also like to express my sincere gratitude to my second supervisor Dr. Christos Bouganis. During the four years of my Ph.D research, he has been working closely with me, providing guidance for my research direction as well as attending to the detailed problems. I would like to thank Christos for his patience and help during the publication of my work and the writing of this thesis.

My gratitude also goes to my colleagues working in the Group of Circuits and Systems in the Department of Electrical and Electronic Engineering, Imperial College London.

Last but not least, I would like to thank my family: my sincere gratitude to my parents Changqing Liu and Huimin Liu who have been supporting me all the time and made my Ph.D possible; and many thanks to my beloved wife Ziyu Wang, who has been and will always be my great partner of life.
Contents

Abstract i

Copyright Declaration ii

Acknowledgements iii

1 Introduction 1

1.1 Motivation and objectives 1

1.2 Overview of the thesis 5

1.3 Statement of Originality 6

1.4 Publications 6

2 Background and Related Work 8

2.1 Introduction 8

2.2 Image Data Accessing in Hardware Systems 8

2.3 Scratchpad Memory and Memory Hierarchy 10

2.4 The Optimization of Data Access and Transfer Process 13

2.4.1 Memory Structure and Data Storage 13
CONTENTS

2.4.2 Algorithm Modification and Code Rewriting .................................. 16
2.4.3 Summary of Data Access and Transfer Optimization ......................... 17

2.5 Communication-aware Computing .................................................... 18
2.6 Compression of Image Data ............................................................. 19
2.7 Image Interpolation and Regression .................................................. 21
2.8 Conclusion ......................................................................................... 23

3 Proposed Solution ................................................................................. 24

3.1 The Idea of Context-based Image Acquisition ..................................... 25

3.2 Scenario Setup .................................................................................... 28

3.3 The Baseline of Energy Consumption in Hardware Systems ................... 30

3.3.1 Experiment Setup ........................................................................... 31

3.3.2 Results and Discussions ................................................................. 32

3.4 Analysis of the Proposed Framework ................................................... 37

3.5 Conclusion ......................................................................................... 41

4 Design of a Prototype CbIA Architecture .............................................. 42

4.1 Introduction ......................................................................................... 42

4.2 Progressive Sampling of Images ......................................................... 43

4.3 Design of the Sampling Procedure ....................................................... 46

4.3.1 Scenario Setup ................................................................................. 46

4.3.2 Structure of Memory Systems ......................................................... 46
5.6 Evaluations ................................. 103
  5.6.1 The Balancing Between Variance and Distance Terms ................. 103
  5.6.2 Evaluation of KbAS Algorithms ........................................ 106
5.7 Cost of the Kernel-based Adaptive Sampling Algorithm .................... 107
5.8 Conclusion ........................................ 111

6 Domain Specific Image Acquisition of Face Images ........................................ 112
  6.1 Introduction ........................................ 112
  6.2 Review: the Hallucination of Face Images ...................................... 114
  6.3 Overview of the Domain-specific Point Sampling of Faces ................. 116
  6.4 Design of the Domain Specific Point Sampling of Faces ...................... 118
    6.4.1 Reconstruction by Hallucination ....................................... 118
    6.4.2 Patches vs. Full Image ............................................... 124
    6.4.3 Learning from Database ............................................. 125
    6.4.4 Sampling Order and Validation ...................................... 126
  6.5 Evaluations ........................................ 129
    6.5.1 Experiments without close examples of the testing subject .......... 130
    6.5.2 Experiments with close examples of the testing subject ............ 132
  6.6 The Cost of Domain-specific Sampling ....................................... 133
    6.6.1 Storing and Accessing Learned Prior Knowledge ....................... 134
    6.6.2 The Computational Cost ............................................. 135
  6.7 Conclusion ........................................ 136
7 Conclusion

7.1 Summary ................................. 138

7.2 Analysis of the Proposed CbIA Concept ................................. 141

7.3 Potential of Context-based Image Acquisition ................................. 143

7.4 Future Work ................................ 144

7.4.1 Short Term Plan: Further Investigations and Modifications ............... 144

7.4.2 Long Term Plan: Future Directions of the Work ......................... 146

Bibliography ................................ 147
List of Tables

3.1 Chosen FPGA chips specifications .................................................. 32
3.2 Chosen structured ASIC specifications ............................................. 32
3.3 Memories on Cyclone II development board, Altera [Alt07] .................. 35
3.4 Memories on Stratix IV development board, Altera [Alt10] ................... 35
3.5 Notations used in Eq 3.1 ................................................................. 39
3.6 Notations used in Eq 3.2 ................................................................. 40
4.1 Hardware resource usage of the proposed system, on Stratix IV. The percentage resource usage in the last line shows the percentage of total resource of the corresponding type used on the device .................................................. 63
4.2 Hardware resource usage of the proposed system, on Hardcopy IV. A total of 0.55% of the total HCell resource on device is used ........................................... 63
4.3 Reported max frequencies of the design ............................................. 63
5.1 Flops of example operations. [Min03] ................................................ 109
6.1 The over-fitted data points in Figure 6.5 ............................................ 124
6.2 On-chip memory bits required for the storage of learned prior knowledge. The memory bits are measured in **Mega bits**, and is compared with the total amount of block RAM bits available to Stratix IV EP4SGX530KH40C2 [Alt12]. . . . . 134
List of Figures

1.1 The basic scenario for image acquisition consists of a source memory that contains the target image data to access, and a computing engine that houses the client image processing application that requests for the image data. 1

1.2 The performance gap between processor and memory. The processor line shows the increase in memory requests per second on average, while the memory line shows the increase in DRAM accesses per second. Both serve as a measurement of speed of the device in question. [HP12] 3

1.3 Image accessing with optimization. Evolving from the basic scenario of image acquisition, existing methods work on the development of memory hierarchy and access pattern optimization (highlighted in red). (reviewed in chapter 2) 3

1.4 Scenario setup. The proposed CbIA architecture replaces the conventional image accessing process (Figure 1.3) with a dynamic and progressive sampling procedure (marked in red). 3

2.1 Example of the execution of image processing algorithm as nested loops. [KP01] 9

2.2 Memory hierarchy in computer architecture. The hierarchy can be extended to a wider range of hardware architectures, including custom hardware. 10

2.3 DRAM structure. 14
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4</td>
<td>Optimization of data access pattern and storage can reduce the overall cost in latency, time and energy consumption of data accessing. Note, the minimum time required for each row to be active is omitted for simplicity purpose.</td>
</tr>
<tr>
<td>2.5</td>
<td>Example of code transformation in DTSE. In this particular example, two loops are merged into one to reduce the storage and bandwidth requirement. [CDWD01]</td>
</tr>
<tr>
<td>2.6</td>
<td>The general process of JPEG2000 image compression standard [SCE01].</td>
</tr>
<tr>
<td>2.7</td>
<td>Example of image interpolation and regression problems. (a) Reconstruction of pixels from existing samples on regular grid. (b) Reconstruction of pixels from existing samples on irregular grid. (c) Image denoising corrects pixel values according to regressed signal function. (d) In super-resolution, multiple frames of the video are fused into one high resolution frame and the problem essentially turns into an reconstruction problem on irregularly sampled pixels. [TFM07]</td>
</tr>
<tr>
<td>3.1</td>
<td>The basic scenario for image acquisition consists of a source memory that contains the target image data to access, and a computing engine that houses the client image processing application that requests for the image data.</td>
</tr>
<tr>
<td>3.2</td>
<td>Image accessing with optimization. Evolving from the basic scenario of image acquisition, existing methods work on the development of memory hierarchy and access pattern optimization (highlighted in red). (reviewed in chapter [2])</td>
</tr>
<tr>
<td>3.3</td>
<td>Comparison between conventional image accessing method and the proposed method. (a) the conventional accessing method; (b) the proposed method.</td>
</tr>
<tr>
<td>3.4</td>
<td>Scenario setup. The proposed CbIA architecture replaces the conventional image accessing process (Figure 3.2) with a dynamic and progressive sampling procedure (marked in red).</td>
</tr>
<tr>
<td>3.5</td>
<td>Setup of the baseline energy consumption test on FPGA.</td>
</tr>
<tr>
<td>3.6</td>
<td>Breakdown of energy consumption.</td>
</tr>
</tbody>
</table>
4.1 Example of AFPS-based sampling method. (a) Voronoi diagram on sampled pixels (blue dots); (b) Voronoi vertices are candidates to be sampled in the next iteration; (c) the vertex that has the highest priority score is chosen and sampled; (d) the newly sampled pixel updates existing Voronoi diagram. 45

4.2 Example sampling pattern resulted from AFPS. (a) original image “Camera-man”; (b)-(d) first 1024, 4096, 8192 samples. [ELPZ97] 45

4.3 Scenario setup for the prototype system design. 45

4.4 Linear mapping (a) and block mapping (b) of image data in SDRAM. 48

4.5 Progressive sampling methods: uniform sampling (top); full adaptive sampling (middle); proposed adaptive sampling (bottom). 49

4.6 The continuity of natural signal. Three rows of the image lena are picked and their grayscale values plotted, showing the continuity of pixel values of each object. 51

4.7 The size of macroblocks. Red and green rectangles mark blocks processed by the proposed system. Pixels marked in blue are samples taken during the refining process. The uniform refining of each block requires the size of blocks to be $$(2^n + 1) \times (2^n + 1)$$. 53

4.8 The performance measured in image quality vs. percentage of pixels sampled, tested on the combined LMO and SUN database. The lines show the average performance of the sampling procedures on different images, as well as half the standard deviation of the performance. 54

4.9 Comparison between ground truth image and the reconstruction using pixels sampled at a threshold of 600. 56
4.10 Evaluation of sampling procedure. From left to right, the data points of adaptive refine and full adaptive sampling algorithms in these graphs are results from threshold of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively; the data points of uniform refine algorithm are results from sampling distance of 16, 8, 4, and 2 respectively. 57

4.11 The structure of the proposed CbIA architecture. (a) The proposed design generates pixel addresses for the DRAM interface; (b) a local canvas buffer stores sampled pixels as well as interpolated pixels; (c) the refine unit checks priority scores of each block; (d) the addr translator generates sampling addresses if a block is to be refined; (e) an array of interp units interpolates the missing pixels for blocks that do not need further refining/sampling. 59

4.12 Example of system working mechanism. 60

4.13 Time requirement for sampling process, and complete acquisition process including interpolation. The X axis shows the achieved PSNR given different levels of thr. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively. Reference lines show the time requirement of conventional image accessing method in equivalent clock cycles, assuming the memory data bus is working at x times the prototype system’s frequency. 64

4.14 Breakdown of energy consumption by the proposed system, for sampling process (marked by “s”), and complete process including interpolation (marked by “t”). Reference lines are the energy consumption of accessing the whole target image from SDRAM by conventional method. 67
4.15 The ratio of the energy consumption of the source memory (DDR3-667) to that of the memory access by conventional access method. Reference lines at ratio = 1 shows the energy consumption of the conventional access method. It can be seen from this figure that by trading part of the image quality, the CbIA architecture is able to reduce the energy consumption on the memory side by a significant portion. ................................. 67

4.16 The ratio of total energy consumption of the proposed system (including corresponding energy spent on sampling from DRAM) to that of the memory access by conventional access method. Different DRAM models are used as target memory. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively. ................................. 68

4.17 Case study on JPEG2000: both the ground truth image and the reconstructed image from CbIA system are used for image compression process. This study aims to analyse the impact of reduced image quality brought by CbIA-enabled memory accessing interface. Source memory is Rambus model of DDR3-667. ................ 70

4.18 The quality of compressed image measured in MSE, using both conventional accessing method and the proposed system. DDR3-667 is used as source memory. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively. Because of the additional quality loss introduced by the CbIA procedure, the error of the compression output using CbIA acquired images is higher than that of the conventional image acquisition method (blue reference lines). ................................. 71

4.19 The quality difference of compressed image, using both conventional accessing method and the proposed system. DDR3-667 is used as source memory. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively. ................................. 72
4.20 Pre-fetch in the proposed sampling procedure, assuming a block mapping strat- 

egy same as in Figure 4.4(b). .................................................. 74

4.21 Sampling process evaluation of the system under memory burst mode. (a) achiev-
able PSNR vs Number of samples; (b) SDRAM accessing energy vs. achievable
PSNR. Source memory is Rambus model DDR3-667. ........................................ 75

4.22 Energy consumption evaluation of the system under memory burst mode. Source
memory is Rambus model DDR3-667. .................................................. 75

5.1 The example image patch used in the discussion through this chapter. ............ 81

5.2 The 1D sampling-reconstruction example. The sampling and reconstruction
work on a single row of pixels marked in red in the ground truth image patch.
Two sampling patterns are provided together with the cubic interpolation results
using them. ................................................................. 82

5.3 An example of one iteration during FPS sampling procedure. (a)(b) Based on the
existing sampling pattern (marked in blue dots), Voronoi vertices are identified
(marked in red dots). (c) The one vertex farthest from the sampling pattern is
selected and sampled. (d) The Voronoi vertices are updated with the addition
of the newly sampled pixel. [DL07] ........................................ 84

5.4 An example of the generated sampling pattern using AFPS, on lena 257x257, at
4181 samples. ................................................................. 84

5.5 Effects of applying the steering matrix \( C_\chi = \gamma_\chi \cdot U_{\theta_{\chi_{i}}} \Lambda_\chi \cdot U^T_{\theta_{\chi_{i}}} \); the shape of the
kernel is changed to reflect the local image structure. [LFM07] .......................... 90

5.6 An example of gradient information computed from intermediate reconstructions
of the image, using sampled pixels. Sobel filters are applied along horizontal
\((G_x)\) and vertical \((G_y)\) directions, and the gradient magnitude is computed as
\(\sqrt{G_x^2 + G_y^2}\) ................................................................. 93
5.7 Equivalent kernels (Eq. 5.22) applied at different locations in the image lena. The image is sampled uniformly at the sampling distance of 2.

5.8 Example image patch and priority scores of pixels shown in grayscale, given that all pixels have the same var and dist. Red dots in (b) are locations of already sampled pixels.

5.9 The weights, computed as equivalent kernel values, describe the relationships between pixel pairs.

5.10 Example image patch and priority scores of pixels shown in grayscale, computed as in Eq. 5.28. Red dots in (b) are locations of already sampled pixels. This graph shows that even with a coarse sampling pattern, the priority estimation in Eq. 5.28 is able to roughly identify regions containing high frequency component.

5.11 Example image patch and priority scores of pixels shown in grayscale, computed as in Eq. 5.29. Red dots in (b) are locations of already sampled pixels. Similar to that in Eq. 5.28, the alternative formulation of priority estimation is able to roughly identify regions containing high frequency component.

5.12 Updated priority scores of pixels shown in grayscale, computed as in Eq. 5.29 with more samples retrieved. Red dots in (b) are locations of already sampled pixels. By sampling pixels from high priority regions and updating the priority map accordingly, the sampling procedure iteratively acquires pixels of high estimated significance to the reconstruction process. The sampling is balanced between the two Design Considerations with samples taken from both "flat" regions and regions of high frequency component.

5.13 Example image patch and priority scores of pixels shown in grayscale, computed as in Eq. 5.30. Red dots in (b) are locations of already sampled pixels.

5.14 Examples of variance terms computed as in Eq. 5.31 shown in grayscale. The variance estimation gives a similar result to the distance term estimation.
5.15 Examples of priority map computed by Eq.5.33 which is a combination of data adaptive variance term and distance term, displayed as \(\log(1 + p(x))\) for visual quality. 101

5.16 Examples of priority map computed by Eq.5.39. 103

5.17 The Spearman’s rank correlation coefficient of \(\text{var}(x)\) and \(\text{dist}(x)\) (Eq.5.33), throughout the acquisition of image “lena”. The positive rank correlation coefficient shows that the two terms agree with each other in pixel ranking, in many circumstances. 103

5.18 The slopes of different \(f(l(x))\) choices, shown as \(\frac{\partial f(l(x))}{\partial l(x)}\). 105

5.20 The sampling patterns at 4096 samples, using different distance terms. (a) op1; (b) op3; (c) op5. 106

5.21 Reconstruction examples using KbAS algorithms, compared with grid AFPS. 107

5.22 Sampling/reconstruction results using KbAS algorithms on benchmark images, compared with grid AFPS [DL07]. Five different formulations of KbAS algorithm are evaluated, all producing plausible image quality to b/p ratio. Option 1 in this test, labelled as “op1: rev(1-l(x))”, is the formulation as in Eq.5.30 which is to compute equivalent kernels on sampled pixels. 108

5.23 Breakdown of the cost of reference AFPS algorithm, and selected KbAS algorithms. 110

5.24 Normalized cost of reference AFPS algorithm, and selected KbAS algorithms. 111

6.1 Face hallucination via eigen transformation [WT05]. The projection coefficients are computed using a LR version of the example database. The hallucination is done by mapping these coefficients back to their HR counterparts and guide the reconstruction with the HR example database. 115
6.2 The method proposed by Hu et al. [HLQS11], in which the example HR images are first warped to match the structure of the input LR image. The warped HR examples are then used to learn local pixel structures for the regression of missing pixel values in the LR image.

6.3 Overview of the face-oriented domain-specific sampling-reconstruction process. For a given patch location, e.g. the region marked in the figure, a set of sampling patterns are learned off-line from the training database. Samples retrieved are used for patch reconstruction with learned codebook in the form of eigenspace.

6.4 The problem objective function Eq 6.4 of face hallucination in image progressive sampling, assuming the example space $B$ is the original collection of example faces without transformation.

6.5 Over-fitting of hard constraint solution. In this graph, the reconstruction qualities under various $n$ and $r$ are plotted. a) is the solution using the hard/soft constraints in Liu’s work [LSF07]; b) is the solution using the unified MAP formulation in Eq. 6.10.

6.6 Eigenvectors of an example patch location.

6.7 Number of eigenvectors in different regions, given different sized training database and different threshold $q$.

6.8 Learning for sampling patterns: (a) the variation map of the patch location marked in Figure 6.3; (b) the initial ($S_0$) sampling pattern with only 4 samples, one at each corner (white dots are pixel locations to sample); (c) the priority map computed at each pixel by Eq 6.16; (d) sampling pattern $S_1$ at level 1 iteratively picks pixel locations with highest priority in (c) and update the priority map accordingly; (e) updated priority map after $S_1$.

6.9 Examples of face images of the same testing subject in FERET database.
6.10 Additional examples of the performance comparison at the iteration when 5% and 12% pixels are sampled. Same as the test in Figure 6.11, 500 faces are randomly selected for training, excluding any examples of the testing subject. For each testing face shown in this graph, (a) is the ground truth image; (b)(c) are reconstructions from global grid AFPS and triangulation-based linear interpolation; (d)(e) are reconstruction from the proposed sampling and reconstruction method. 130

6.11 Example reconstructions with different amount of pixels sampled; (b)-(e) are reconstruction examples obtained from global grid AFPS and triangulation-based linear interpolation; (f)-(i) are reconstruction examples obtained from the proposed method with $q = 99.9\%$ and 500 training images in the database, excluding any examples of the testing subject. The locations of sampled sites for these reconstructions are shown as well, below their corresponding reconstructions. 131

6.12 Performance evaluation with $q = 99.9\%$ (left) and $q = 99.5\%$ (right); for the patch location in Figure 6.3, 100, 123 and 134 eigenvectors are preserved for 200, 400 and 600 training examples in the database, respectively. 132

6.13 Impact of including examples of the testing subject (about 5-9 examples per testing subject, depending on the availability of such examples in the original database). 133

6.14 Normalized cost of the proposed domain-specific CbIA procedure. The graph shows the proposed method with 400 training images with and without close examples. While higher energy threshold $q$ leads to better PSNR vs. b/p ratio (figure 6.12), it also leads to more costly computation because the example space is more complex. 136
7.1 The conclusive evaluation of various CbIA procedures involved in the thesis, as well as the reference grid AFPS algorithm. (a) This graph shows the PSNR vs. b/p performance of the various sampling procedures, which shows the ability of the sampling procedures to trade image quality for reduced bandwidth. (b) This graph is the normalized cost of sampling procedures, which abstracts their time and energy consumption.

7.2 The proposed CbIA can be extended from pixel-based access to generic block-based access. This allows the proposed concept of CbIA to be compatible with existing methods such as frame re-compression (e.g. the work of Lee et al. [Lee03, LRL07]). In this figure, sampled pixels by pixel-based CbIA are marked in blue. (a) The current pixel-based CbIA procedures; (b) it is already shown in Chapter 4 that CbIA procedure can adapt to the pre-fetching of memory devices (each sample pixel followed by a burst of pixels, marked in green); (c) the proposed CbIA concept can be extended to generic block-based accessing, which in this example is a $2 \times 2$ block.

7.3 The automated CbIA module generation tool takes into account the task in question, analyse the example images, and generate the CbIA module according to the requirements set by the user.
Chapter 1

Introduction

1.1 Motivation and objectives

With the development of modern technology, the process of image acquisition has become a major concern during the design of image processing systems. It is often seen in state-of-art systems that the bandwidth requirement, time, and energy costs of this acquisition process present a significant impact to the overall cost of the complete image processing system [ZZH+11]. A simplified scenario of image acquisition can be seen in Figure 1.1 where some image processing application is implemented on a computer engine and requests image data from a source memory.

On one hand this increasing impact in cost is due to the increasing resolution of image capturing devices, which leads to larger size of image data. Access and transmission of the image data with ever growing size poses a constant challenge to the design of memory systems [ZZH+11, KP01].

On the other hand, the deviation between the development of computing engines and memory systems leads to the fact that the computational power of devices has become more available than communicational power in hardware systems [RKC13], which makes it beneficial to shift workloads from the memory side to computing engines. This is reflected from two aspects:

Firstly, the improvement of processing speed of computing engines exceeds that of the memory
Chapter 1. Introduction

Figure 1.1: The basic scenario for image acquisition consists of a source memory that contains the target image data to access, and a computing engine that houses the client image processing application that requests for the image data.

access. According to the report of Patterson et al. \cite{PSG05}, the computation performance (floating point operations per second) is increasing by 59% a year whereas communication performance is improving at a much lower rate (DRAM latency improves by 5.5% and bandwidth improves by 23% every year). In particular, the development of processors and memory systems saw an increasing performance gap between the two. The performance in this case measures the speed of the two devices, which is defined to be the average memory requests per second for the processor, and average data accesses per second for memory. Figure 1.2 shows the performance of single processor against the performance increase in time to access main memory. In recent years the design of processors moved towards multi-cores which further increases this performance gap \cite{HP12}. With faster processors, inadequate memory bandwidth and high access latency stall the processors from performing further image processing actions and therefore increases the overall time consumption of image processing operations.

Secondly on top of the time consumption, energy consumption has become a major concern in state-of-art hardware designs, especially embedded systems. Memory system design is limited by the balancing between size, system complexity, manufacturing cost and etc.. These restrictions as well as the lowered energy cost of logic computations has made the energy cost of data accessing more significant in the overall energy cost of the system than ever before. As is reported in the work of Zhou et al. \cite{ZZH11}, in their H.264 video decoding system...
1.1. Motivation and objectives

Figure 1.2: The performance gap between processor and memory. The processor line shows the increase in memory requests per second on average, while the memory line shows the increase in DRAM accesses per second. Both serve as a measurement of speed of the device in question. [HP12]

the decoding process (excluding DRAM access) only spends about 0.36 nJ per pixel decoded, whereas the DRAM accessing process during the decoding costs 1.11 nJ per pixel accessed. In such image processing systems, the memory accessing process has such a significant presence that it dominates the overall energy consumption of the system.

In general, the cost of image acquisition process both in time and energy consumption has become increasingly significant. This leads to various researches and developments in dealing with the cost of the image acquisition process. These efforts include general solutions such as the development of memory hierarchy, and image specific solutions such as access pattern optimization [LZ12], application code rewriting [CDWD01], and etc.. As is shown in Figure 1.3 most of existing works focus on the development of memory hierarchy and access pattern optimization. These works lay down a solid foundation for modern hardware design. A more detailed review of the literature is provided in chapter 2.

With the same goal of reducing the cost of image acquisition process between the computing engine and source memory (Figure 1.1), this work approaches this problem from a novel direction (Figure 1.4). Inspired from the development of image processing techniques, this project proposes the concept of “Context-based Image Acquisition” (CbIA) which serves as a frame-
work of designing an intelligent hardware architecture of image acquisition from a source, often in the form of external memory that holds the target image. Such architecture utilizes image processing techniques to aid the process of image acquisition. It combines sampling of the target image and reconstruction using the sampled fractions, to achieve the goal of exchanging part of the image quality for the reduction of the bandwidth/time/energy cost of this acquisition process, which ultimately leads to the reduction of the overall cost of image processing systems. The illustration of the proposed CbIA framework and its scenario of application is shown in Figure 1.4 with its image acquisition mechanism highlighted in red.

The trade-off between image quality and various costs of image acquisition is the key idea of
the proposed CbIA framework, which provides an alternative approach of designing hardware architectures for image processing systems. Therefore it is the focus of this work to investigate image sampling and reconstruction algorithms and their interaction. Although the proposed CbIA framework is different in design compared with conventional image data acquisition protocols, the scenario setup as is shown in Figure 1.4 aims to be generic. The proposed framework is to be applied in image processing hardware such as in surveillance cameras, video compression/decompression systems, feature extraction and recognition systems etc..

To summarize, the objectives of this work are as follows:

1. This work is to propose the CbIA framework – a framework of intelligent hardware architectures that is capable of dynamically and selectively acquiring image data from a source memory, instead of accessing the full data.

2. The proposed CbIA framework aims to reduce the bandwidth requirement and overall cost of image acquisition process in hardware systems by trading part of the image quality.

3. The designs and discussions in this work are made in an effort to be compatible with existing hardware protocols and common hardware environments.

The work reported in this thesis shows the potential of the proposed CbIA framework. The designed CbIA architecture on Hardcopy IV device (Chapter 4) manages to achieve an overall reduction of: a) 88% of communication bandwidth, b) 68% of time consumption, and c) 50% of energy consumption of the image acquisition process that fetches target image “lena” from an off-chip SDRAM\(^1\) at the expense of reduced image quality (about 33 dB of PSNR) and extra hardware resource used to implement the proposed architecture. The trade-off between image quality and cost of data accessing is extensively discussed in Chapters 5 and 6 where the framework manages to trade more computational power for other performance metrics.

As the performance gap between the computational power and communicational power in hardware systems \(^{[PSG+05, HP12]}\) increases and the need of energy saving keeps growing, the

\(^1\)Performance metrics of the proposed CbIA framework are explained in detail in Chapter 3. Note that the evaluation is application dependent and the performance varies by the target image. Detailed evaluations are provided in corresponding chapters.
reduction of the cost of image data acquisition will always be an important research topic. By introducing the framework of “Context-based Image Acquisition”, this work aims to provide a novel alternative direction specifically for image data, alongside existing research directions such as communication-aware computing [ABDK11 DGHL12 RKC13].

1.2 Overview of the thesis

In this thesis, the concept of Context-based Image Acquisition is proposed. In Chapter 2, the available literature related to the accessing of image data within hardware systems is briefly reviewed, which further motivates the work in this thesis. With the background established, Chapter 3 introduces the CbIA framework which is the key to this thesis. The CbIA framework is explained and discussed on a generic scenario set for custom hardware. A preliminary analysis of the proposed framework is provided, which serves as reference and guideline for the design of the detailed algorithms and architectures in later chapters.

Following the established CbIA framework, Chapters 4, 5, and 6 provide detailed discussion of the design of CbIA architecture/procedures. In particular, a CbIA architecture is designed and evaluated on reconfigurable hardware in Chapter 4, showing the benefit of applying such concept in the design of image processing hardware systems. Chapters 5 and 6 focus on the discussion of the sampling procedure designed for CbIA architecture. A generic sampling procedure, the Kernel-based Adaptive Sampling is proposed in Chapter 5 which achieves higher PSNR vs. b/p performance than state-of-art sampling algorithms. On the other hand, in Chapter 6 the CbIA procedure is discussed within a specific image class, namely human faces. With a specified image class targeted, the Domain-specific CbIA procedure is proposed which explicitly uses learned prior knowledge to improve PSNR vs. b/p performance.

Finally, the conclusions related to this work are presented in Chapter 7 for a clear overview of the achievements made in this thesis.
1.3 Statement of Originality

The main contributions of this thesis are covered in Chapters 3, 4, 5, and 6. Key contributions are summarized as follows.

1. The framework of Context-based Image Acquisition (CbIA) is proposed as an innovative architecture for performing image acquisition in hardware systems.

2. A prototype architecture under the proposed CbIA framework is designed for reconfigurable hardware platforms, which is evaluated and demonstrates the ability of the proposed CbIA framework to reduce the overall cost of image acquisition process.

3. The Kernel-based Adaptive Sampling (KbAS) of image is designed as a novel image sampling algorithm, which has higher image quality vs. bandwidth performance than existing image point sampling algorithms.

4. The Domain-specific CbIA procedure for face images is proposed and designed, which features improved image quality vs. acquisition cost compared with generic image sampling on faces.

I hereby certify that this thesis and the work to which it refers are the results of my own efforts. Any ideas, data, images or text resulting from the work of others (whether published or unpublished) are fully identified as such within the work and attributed to their originator in the text, bibliography or in footnotes. This thesis has not been submitted in whole or in part for any other academic degree or professional qualification.

1.4 Publications

The following publications have been produced during the course of this work.

1. Jianxiong Liu, Christos Bouganis, and Peter YK Cheung. Domain-specific progressive
Sampling of face images. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. [LBC13]


Chapter 2

Background and Related Work

2.1 Introduction

In this chapter, the literature related to the problem of “image acquisition in hardware systems” is reviewed. The review covers existing methods and techniques that have been developed to reduce the cost (bandwidth, time, energy) of the image acquisition process which leads to the novel hardware framework proposed in this thesis.

2.2 Image Data Accessing in Hardware Systems

Because of the large size and high redundancy of natural image data, digital images are often stored in their compressed version. Image/videos compression is also a vital stage of data transmission over the internet due to the reduced bandwidth requirement (a compression rate of 100:1 is common for JPEG standard [SCE01]). However, inside hardware systems that are designed for image processing applications, the processing of image data often requires to work on the decompressed and raw version of the images.

Such intermediate raw image data exists as a 2-dimensional (grayscale) or 3-dimensional (coloured) pixel matrix, and is stored in the main memory of the system. It is intermediate in that such raw
image data is often used as an input of image processing applications and is often discarded after the processing finishes (e.g. Scale Invariant Feature Transform \cite{low99}). Nevertheless, the intermediate image data is an important part to the image processing systems. The processing applications that deal with still images require direct access to these decompressed raw pixels; other processing applications such as some decompression applications themselves (e.g. video decoding) require an intermediate reference frame image to be stored and accessed repeatedly. For image processing systems, the storage of an intermediate raw image and the transmission of it between devices are essential and unavoidable. Due to the increasing size of image data as well as the growing performance gap between computing engine and memory systems, the process of accessing intermediate image data from main memory has become significant in the overall cost of the system \cite{ZZH11}. This is the main motivation behind this project.

Image processing applications have distinct characteristics compared with general digital signal processing. Many of the image processing algorithms operate in a small region of the image at a time. These algorithms are often based on convolution of a filter kernel and a local region of the image data. In terms of software implementation, these algorithms are commonly executed as nested loops. Such nested loops determine the region on the image where the algorithm requires to access and pixels in this region are repeatedly accessed. The innermost loops often process the pixels in a rectangular region. Therefore the memory access for the innermost loops is called “block type access” and the block of pixels accessed is called a “macroblock”. Marcoblocks are the fundamental unit of image processing applications and they are often of fixed size which is determined by the algorithm. An example is shown in Figure 2.1 which is a simplified MPEG-2 video decoding algorithm. The outer loop pair (lines 1-11) performs macroblock processing such as motion vector decoding. In the inner loop pair (lines 5-9), motion compensated $16 \times 16$ pixels are generated as the output of the routine.

In this project, the problem of image acquisition is broken down into the problem of macroblock acquisition because it is the basic unit of image access pattern. In the following sections, a brief review is given about existing methods and techniques of reducing the cost of image acquisition process by exploiting the characteristics of image accessing in hardware systems.
2.3 Scratchpad Memory and Memory Hierarchy

To bridge the performance gap between the computing engine and memory systems, the concept of memory hierarchy is widely used (Figure 2.2). Faster, less energy hungry, but more expensive memories are applied in smaller size closer to computing cores (e.g. Arithmetic Logic Units) to support the immediate access of data. The development and application of memory hierarchy is based on “data locality” or “locality of reference” [HP12]. Buffering the temporally local data into the high level memories that are closer to the computing core achieves an effect of 1) reducing the latency of overall data access, and 2) reducing the overall time and energy cost of data access via data reuse. The “locality” is reflected in various forms such as the temporal relationship between CPU instructions (e.g. CPU instruction cache) or, in the case of image processing, the fact that pixels in the same macroblock are likely to be used at the same time (e.g. data cache) – the latter is closely related to the methods proposed in this work. The accessing and moving of macroblocks is fundamental to the image acquisition process within modern memory hierarchy.

Due to its size, the complete intermediate image data in question is often stored in large memories of the system such as SDRAMs. The accessing of macroblocks is to move the data within the block to a higher level memory that is closer to the computing core (Figure 2.2). For example, in modern hardware platforms such as CPUs and GPUs, multiple levels of cache exist with

```c
Simple_MPEG_Video_Decoding_Routine()
{
    /* A: current frame */
    /* B: previous frame */
    /* C: IDCT result */

    for (2;0:I<SIZE;I+=16){
        for (I=0:10<SIZE;I+=16){
            Motion_vector_Decode(BMVY,BMVX);
            C = IDCT();
            for (Y=0;Y<16;Y++){
                for (X=0;X<16;X++){
                }
            }
        }
    }
}
/* end of routine */
```
Figure 2.2: Memory hierarchy in computer architecture. The hierarchy can be extended to a wider range of hardware architectures, including custom hardware.

L1 cache being the closest to the actual computing core (ALUs). Image processing applications move data from main memory of the system (external SDRAMs) to different levels of the cache hierarchy. Those fractions of data that are likely to be accessed more frequently are often put in higher levels of the hierarchy and vice versa. The GPU in particular, has a large number of computing cores compared with CPUs, as well as higher internal memory bandwidth. It is particularly suitable for applications that perform independent trivial operations in great number, which is indeed the case for many image processing and computer graphics applications.

On the other hand, while more and more computational power is integrated into a dense chip, it is however challenge for memories to achieve a same improvement. In order to maintain sufficient speed and storage capacity, there has been the intuitive method of stacking memories around a computing engine. Wiring multiple memories together to provide a larger combined memory bandwidth and higher level of parallelism has been an effective solution to the memory problem in supercomputers with top level computational power. However the performance gains of these systems due to upwardly scaling processors were not matched by linearly stacked memory \[\text{[hmc14]}\]. To solve the problem of memory wall, Intel and Micron introduced the new design of 3-dimensional memory called “Hybrid Memory Cube” (HMC). This design is to achieve memory stacking internally through a wire-like connection given the name of “Through
Silicon Via™. The HMC is reported to be able to achieve a much improved speed and bandwidth (10x improvement) compared with current DDR3 and DDR4 memories [hmc14]. While the HMC is not to be seen as a development of data management through memory hierarchies, it is an inspiring innovation of memory structure at a single level in the memory hierarchy.

Despite of the development of memory hierarchy within systems/computing packages or the development of complex memory structures such as HMC, it remains untouched that the fundamental element of the image data transmission is the moving of image macroblocks to higher levels of memory hierarchy. It is due to the characteristics of image data accessing explained in section 2.2, local buffers such as scratchpad memory [BSL+02] are often used next to computing cores in dedicated image processing hardware, instead of the widely used cache system in general processing units. The scratchpad memory is similar to L1 cache in that it is the closest to the computing core. Compared with cache, scratchpad memory is employed for simplification of cache control logic and it provides a temporary storage often with Direct Memory Access (DMA) capability [BSL+02]. It is suitable for image data access because the access pattern of image macroblocks is often fixed and time invariant, which does not require the cache control logic in order to be general. In custom hardware designed for image processing applications, other types of local buffers exist that share the same concept of scratchpad memory.

There is also work that constructs memory hierarchy specifically for image processing applications. For example, the work of Fischaber et al. [FWM10] explores ways of changing dataflow graphs of image processing applications, to help deriving suitable memory hierarchy that optimizes data reuse in the application while minimizing the memory usage of the hierarchy. It is reported in the work of Fischaber [FWM10] that by using the derived memory hierarchy which includes several levels of local buffers on-chip, the off-chip memory bandwidth requirement of the motion estimation in video coding can be reduced by a factor of 1000 (from 3.3 GB/s down to 3.7 MB/s).

In general for image processing systems, data reuse in the form of macroblocks is common and can be exploited to reduce the cost of data accessing from off-chip memories. For image processing systems, macroblocks of image data are often seen to be temporarily buffered in a
Chapter 2. Background and Related Work

high-level RAM-type memory to which the computing core has direct access. In this work, the investigation is based on this mechanism but is more focused on how to efficiently move image macroblocks from external source memory to the local buffer. Indeed commercial computing platforms such as CPUs and GPUs have unique working mechanism that match their structure. This work however aims to discuss the fundamental acquisition task of macroblocks from a general point of view without being restricted to any particular computing structure. Therefore reconfigurable hardware platform is the main focus in this work as it provides the freedom of customizing the hardware structure. Nevertheless, with efforts made it is possible to apply this work to CPUs and GPUs to bring innovative working mechanisms in these platforms.

2.4 The Optimization of Data Access and Transfer Process

2.4.1 Memory Structure and Data Storage

While the development of memory hierarchy successfully reduced the cost of data acquisition process via data reuse, there is other research that focuses on the further reduction of the cost. On the side of the source memory, a series of researches were conducted to investigate the schemes of data storage and transfer scheduling. Researchers were inspired by the observation that the costs of accessing and transferring data from source memory to local buffers are not homogeneous. Depending on the type of memory structure as well as the previous accessing status, accessing one pixel from the source memory can have vastly different costs in latency, time and energy consumption. Therefore, accessing a same set of requested pixels from a same piece of memory costs differently depending on how the data is stored, and how the access sequence is arranged \[\text{HP12}\].

As an example, one of the most exploited memory characteristics is the inhomogeneous costs of row/column/bank activations in DRAMs. The dynamic nature of DRAM makes it smaller in size compared with SRAMs. At the same time the accessing of DRAM data is more complicated
2.4. The Optimization of Data Access and Transfer Process

Figure 2.3: DRAM structure.

than that of the SRAM. To access a data in DRAM (Figure 2.3), its containing row has to be activated first. After the command of row activation (Row Access Strobe), bit-lines are pre-charged to mid-voltage and then connected to storage cells to fetch the whole row of data. The latched bits are then enhanced by a sense amplifier which also self-refreshes. After these steps (tRCD: RAS to CAS delay) the row is announced open and data from the particular column can be selected by column address (tCL: column access latency). While a row is open, data from the same row can be accessed without introducing further tRCD overhead. On top of this, state-of-art DRAMs organise the storage into different banks and have complex controlling logic to pipeline data accessing tasks. These features of DRAM all lead to the inhomogeneous costs of data accessing. Optimizations can be performed on data storage and access pattern to exploit data temporal locality and reduce the amount of more costly activities such as row switching.

An example of optimization of data storage/access pattern on the source memory side is shown in Figure 2.4. For the access of a set of data bits \{1, 2, 3\} located in a DRAM as is shown in Figure 2.4(a), if the access pattern is $1 \rightarrow 2 \rightarrow 3$ then two row switching activities will happen.
Figure 2.4: Optimization of data access pattern and storage can reduce the overall cost in latency, time and energy consumption of data accessing. Note, the minimum time required for each row to be active is omitted for simplicity purpose.

By modifying the access pattern to $1 \rightarrow 3 \rightarrow 2$, the same set of data bits are accessed with only one row switching activity involved, which leads to a reduced overall time and energy consumption of the process. In situations where the accessing patterns cannot be modified (i.e. the pattern has to be $1 \rightarrow 2 \rightarrow 3$), moving data bit $\{2\}$ to a corresponding row and column in a different bank can also reduce the overall time required for the whole accessing process as the row switching activities can be partially pipelined with data accessing.

Depending on the actual application, such characteristics are utilized to optimize the storage and access of data in source memory. In the particular case of image data, it is often seen that macro blocks of the image are mapped to memory in such a way that the sequential access of each macro block involves the least amount of row switching activities, therefore costs the least amount of time and energy – an average reduction of 29% of SDRAM commands and 89.2% of row switching activities is reported on testbench videos [KP01]. In the same work a memory interface is also proposed which is in charge of mapping logical accessing addresses to the physical addresses of re-organized memory content. The works of Hettiaratchi et al.
are more general methods which are not limited to image macro blocks but offer an automatic memory content organisation and address mapping scheme, based on given applications.

Apart from the storage of data and access pattern, optimization of transfer scheduling can be performed on the databus of memory systems to exploit the “spatial” locality of data. Here the “spatial locality” of data describes the observation that some data are similar in value to each other or even the same. Therefore if these data are transferred on the data bus in consecutive order, the cost of “self transition” of data bus bit lines can be reduced [LZ12]. This work also utilizes the signal continuity of natural image data and schedules the transfer of bits on bit lines to reduce differences of data transferred on neighbouring bit lines, to reduce the cost of “coupling transition”. Combined with several other techniques (Gray coding, etc.), Li [LZ12] reports a reduction of about 38% of macroblock access energy consumption compared with direct access without scheduling.

There is also research on the modification of memory structure itself, particularly for image processing applications. The work of Perri and Corsonello [PC11] introduces a novel sub-bank single port SRAM-based memory structure to work with some common memory accessing schemes required by image processing applications. It is able to reduce the area occupancy and power consumption (by 25% and 9% respectively) but offers the same data bandwidth and throughput reached by a true dual port SRAM. Targeting the specific application of Bit Plane Coding (BPC) in JPEG2000 [SCE01], the work of Gupta et al. [KGNT05] proposes a 2 sub-bank memory architecture which has simple memory controller and reduced size, resulting in a 77%-79% improvement in memory cost (the cost is defined by the author to be a combination of area, size, working frequency). Further more, there is also research on the development of intelligent memory systems which have integrated processing elements. In the work of Kim et al. [KKL+10], a Visual Image Processing RAM (VIP-RAM) is introduced for object detection tasks. This special memory structure has built-in processing elements that perform the operation of local maximum location search and data consistency management particularly for the Scale Invariant Feature Transform (SIFT) [Low04].
In general all these methods investigate the activities and structures of source memory that contains the data to access from, as well as the requested pattern of addresses the application is likely to issue. Optimizations are performed mainly on the organisation of memory contents and the re-arranging of access patterns and transfer sequences. In the next section, another branch of research is reviewed, which emphasises the rewriting of application codes to achieve the same goal of reducing the cost of memory accessing.

2.4.2 Algorithm Modification and Code Rewriting

The data-transfer and storage bottleneck was discussed as early as in the work of Catthoor and Nachtergaele [CDKO00, NCK01] and it has attracted much research interest. Unlike the methods of data access/storage optimization over the source memory, there are also a large family of methods developed to solve this problem from the application side. Based on the structure of general purposed processing unit and cache system, code rewriting has been studied as a way to reduce the cost of memory accessing. Without touching the source memory from which the data is read, source-to-source code rewriting is done by compilers to transform and optimize the application code in such a way that less temporal storage (e.g. cache) and fewer memory accesses are needed to execute the code. Such methods of automatic code rewriting can be designed to be platform independent [CDKO00, DCDM00] which is an advantage for practical use, although the resulting optimized code can produce further improvement if passed through a platform-dependent stage. Benini and De Micheli gave a good overview of the related approaches of system-level transformations which are focused on reducing power consumption [BM00].

Among the vast amount of compiler related methods, the concept of Data Transfer and Storage Exploration (DTSE) proposed by Catthoor et al. [CDWD01] is a good example that also has a focus on multimedia data. Similar to other source-to-source code transformation techniques, DTSE focus on the optimization of nested loops – which is a featured pattern in many image processing applications – in the source code and it aims to enhance the temporal and spatial locality of data throughout the application which results in the improvement of cache
2.4. The Optimization of Data Access and Transfer Process

Figure 2.5: Example of code transformation in DTSE. In this particular example, two loops are merged into one to reduce the storage and bandwidth requirement. [CDWD01]

performance. Indeed the work of Catthoor [CDWD01] uses cavity detection as an illustration example. In the framework of DTSE, the source code of the application is analysed and global optimization is performed. The optimization involves data-flow, loop, and data-reuse-related transformations. A simple example is given in Figure 2.5 where two loops are merged into a single loop to reduce the storage and bandwidth requirement. The full package of DTSE can optimize the source code to effectively reduce the amount of memory accesses as well as the size of source temporal storage needed. The code transformation methods in DTSE also expose the inherent parallelism of algorithms which leads to faster execution of the application. The work of Catthoor [CDWD01] reports a much reduced accesses (about 90% reduction) to the original image during cavity detection, via DTSE loop transformation and the use of line/pixel buffers to temporarily hold accessed pixels.

In general, it can be seen from the literature of code transformation that the nested loops in application codes have very distinct characteristics that can be exploited to reduce the overall cost of memory accessing. The emphasis on increasing data temporal and spatial locality is essential to code transformation methods such as DTSE. This family of techniques offer a set of tools to deal with the cost of memory accessing from the application side.

2.4.3 Summary of Data Access and Transfer Optimization

The above sections provide a brief review of the memory structure and data access/transfer optimization methods that aim to reduce the overall cost of memory accessing. These tech-
niques all have their own advantages. The hardware centric methods that focus on the data transmission and storage scheme are highly platform dependent but most of them can be universally applicable to different applications. The various innovations in memory structure produce optimized memories particularly for image processing applications, offering a wider range of choices for system design. The software centric methods such as code rewriting are application dependent, but can be platform independent and be performed off-line. However, most of these methods still treat image data as a pure matrix of numbers without exploiting its contextual information. On top of that, no matter how the access patterns are optimized by these methods, all pixels of the image are still accessed without exception.

In this project, the proposal of *Context-base Image Acquisition* is to exploit the contextual meaning of image data in the process of image acquisition from source memory, to achieve a similar goal of reducing overall cost of image acquisition process. Specifically designed to deal with image data, the framework of the proposed CbIA system is to be platform-independent as well as application-independent. The investigation and research in this project aims to alter the conventional way of memory access, enabling contextual selective access of image data instead of accessing all pixels.

### 2.5 Communication-aware Computing

While not introduced for image data accessing, the concept of “Communication-aware Computing” or “Communication-avoiding Computing” is related to this work in that a trade-off is made between computational effort and communicational effort [RKCT13, Hoe10]. It is observed that the computational performance of computing engines (not just general purpose processors) improves at a much higher speed than communication performance of memories [PSG+05]. Therefore a set of innovative algorithms is introduced whose core idea is to extensively use the available computational power of the computing engine for redundant computation, in order to reduce the communication with off-chip memories.

This series of methods are often focused on the solving of linear algebra problems such as
matrix multiplication. Particularly for matrices of extreme aspect ratios (e.g. tall-skinny matrices with many rows but few columns), operations such as QR decomposition can be communication intensive. In the work of Demmel et. al [ABDK11], “Communication-avoiding QR” (CAQR) is introduced for GPUs to perform redundant operations with already fetched data in order to reduce the communication with off-chip memory and improve the overall speed of the QR decomposition. There have been other works that share a similar concept [DGHL12, RKCI3, SCF01, Tol95, CKDI3]. Particularly in the work of Rafique et al. [RKCI3], the idea is applied on the reconfigurable platform of FPGA. By explicitly sharing data across computation kernels and exploiting the on-chip memory, it achieves 1x to 4.2x speedup of Lanczos method over FPGA-based standard implementation.

The trade-off between computation and communication effort in communication-aware computing is via performing redundant computation and exploiting on-chip memory. It is suitable for linear algebra problems but is application dependent, which requires to modify the implemented algorithm/application. While it provides motivations and inspirations this concept is not specifically designed for image data. Moreover, communication-aware computing methods are mostly focused on the speedup of algorithms without considering the energy consumption of the overall process. In this work, the proposal is made to specifically exploit the contextual information of image data and to cover the trade of image quality for a reduced overall cost in bandwidth, time, and energy consumption of the image acquisition process.

2.6 Compression of Image Data

The compression of image data is a well studied topic and compression techniques are fundamental to modern digital image storage and transferring. The various image compression standards, e.g. JPEG 2000 [SCE01], are widely used on digital images to reduce the requirement of storage capacity and transmission bandwidth.

Image compression standards often involve forward transform, quantization, and entropy coding stages (Figure 2.6). Through these stages, the original raw image is analysed and re-arranged
into a bit stream that has the most entropy encoded in the starting part of it. At the receiver end, the coded bit stream is decoded and backward transformed to recover the original raw image. The compression/decompression operations are often performed on rectangular image tiles independently, effectively breaking the problem down to a smaller size. The tiling of the image also allows the system to focus preserving quality on Region of Interests (ROIs).

While the compression standards can achieve satisfying image quality vs. compression ratio performance, the process does require the compression and decompression stages to be implemented and executed on both the data source and the receiver end respectively. In image processing systems, compression of intermediate image data may be too costly to run as well as to implement. On top of that, some image processing applications require to access random regions of the image. For example, to perform motion compensation the process of video decoding might ask for irregularly aligned macroblocks from the reference frame, pointed to by stored motion vector (H.264/AVC video coding standard). In such case, compression of the intermediate image as a whole or even tile-based image compression introduce yet further cost to the random accessing of macroblocks.

Based on the fact that image compression algorithms are too costly and are not beneficial to be applied to intermediate images in hardware systems, there have been studies that try to focus on the light-weight version of image compression algorithms. Among these various solutions, the work of Weinberger et al. is essentially a low
2.6. Compression of Image Data

A complexity version of image compression algorithm similar to JPEG and JPEG2000. However, it is still not considered to be beneficial in dealing with intermediate image data [YLY08]. Specifically for reducing the storage and cost of transferring of image frames during video encoding/decoding applications, the concept of Frame Re-compression is introduced. An example is in the work of Lee et al. [Lee03, LRL07] where hardware friendly low complexity compression is proposed to temporarily re-compress the decoded image frames. Such re-compression is only performed on a relatively small sequence of pixels (8 pixels or 4x4 pixels in the referenced works) and the compression algorithm (e.g., Modified Hadamard transform) is fast and low cost in energy. Such low complexity re-compression techniques are still able to make trade-offs between image quality and storage/bandwidth requirement. At the same time they preserve some random accessibility of the data, and when optimized they can be of low cost to run on hardware systems [Lee03, LRL07]. In detail, the work of Lee [LRL07] reports to introduce an average PSNR degradation of 1.03 dB (average of around 37 dB) compared with the original H.264 encoding without recompression, while a 50% compression rate is achieved.

Nevertheless, these compression techniques all require an additional compression stage to be implemented on the data source which requires adding processing elements next to the source memory. It is the proposal of this project to design a hardware architecture that is based on blind (no pre-processing on the data source) point sampling (with no entropy encoding at the end), which is independent of both the source memory and the computing engine for the sake of compatibility with existing devices. It is worth noting that, the proposed framework is compatible with re-compression techniques if each 8-pixel re-compressed segment is considered as a sampling unit instead of individual pixels. In other words, the proposed framework only requires the source memory to have a certain level of random accessibility in order to perform the sampling procedure, but it does not require to add additional processing capabilities next to the source memory, nor does it require to modify the structure of the memory.
2.7 Image Interpolation and Regression

The core idea of this project is to reduce the overall cost of image acquisition process in hardware systems via point sampling from the source memory. This results in the fact that only fractions of the original image data is acquired by direct sampling and the acquisition process is incomplete until the rest of the data is filled in by estimation algorithms. Given a set of irregularly (non-uniform sampling) sampled pixels from the original image, essentially the proposed framework has to interpolate or regress for the missing pixels.

Given a set of known/sampled data points, interpolation methods estimate the underlying signal function(s) which passes through the given data points. The commonly seen interpolation methods include linear interpolation, polynomial interpolation, and spline interpolation [B⁺06]. Regression analysis achieves a similar goal in terms of filling in missing pixel value, but the problem setting of regression does not require the estimated signal function(s) to pass through the known data points. Both methods are widely used not only in the literature of image restoration, which is closely related to this work, but also in the area of image denoising [TFM07], resolution enhancing [Fat07 FREM04], etc.. In Figure 2.7 some example image interpolation and regression problems are shown. Particularly, the problem setting in Figure 2.7(b) is closely related to this project, where fractions of data from the target image are acquired and scatter on irregular grid. From these scattered pixels, the values of missing pixels are estimated.

Most interpolation/regression methods that are used in image restoration are applicable to generic natural images due to the implicit assumption of signal continuity, or local smoothness of natural data [B⁺06]. Basic algorithms such as Bilinear interpolation for 2D images apply this implicit prior knowledge across the whole image. More advanced techniques introduce data adaptivity and endeavour to identify the locations where such prior knowledge breaks [LO01 Fat07]. Different from pure image restoration problems, the proposed framework of CbIA architecture incorporates both sampling and restoration/reconstruction problems. Therefore inspirations can be acquired from advanced restoration algorithms to design for both the sampling and reconstruction algorithm pairs, which jointly achieve a better reconstruction
2.7. Image Interpolation and Regression

Figure 2.7: Example of image interpolation and regression problems. (a) Reconstruction of pixels from existing samples on regular grid. (b) Reconstruction of pixels from existing samples on irregular grid. (c) Image denoising corrects pixel values according to regressed signal function. (d) In super-resolution, multiple frames of the video are fused into one high resolution frame and the problem essentially turns into an reconstruction problem on irregularly sampled pixels. [TFM07]
quality of the image. More on this topic is discussed in chapter 5.

Finally, there are also techniques that make explicit use of learned prior knowledge from a set of known examples when estimating for missing information of a piece of image data. [FJP02, WT05, HFL10]. One example of this concept is in the example-base image super-resolution. Detailed discussions of the literature and the use of this concept in this project are provided in chapter 6.

The above image interpolation and regression methods are introduced and discussed in the field of image processing. In this project, these methods serve as inspirations to the design of novel hardware architectures for image acquisition.

2.8 Conclusion

Various methods and techniques have been briefly reviewed which mitigate the cost of image acquisition, or to a larger extent, data acquisition from data source (in the form of source memory). As mentioned in each section of this chapter, these methods have their distinct advantages, as well as specific application scenarios. Image compression is widely used for image transmission on network but it is costly to perform for intermediate image data in hardware systems. Methods that work on the source memory side (e.g. pixel scheduling in the work of Li [LZ12]) exploit memory structure or characteristics of the data to achieve faster and lower energy consuming memory access. While being application dependent, these methods often require additional processing elements to be added next to the source memory, or a modification of the memory structure. Methods that work on the computing engine side (e.g. DTSE code rewriting [CDWD01]) do not require hardware modification but are often application dependent. Moreover, existing methods in hardware literature optimize for image processing applications only by the characteristics of the algorithms instead of exploiting the contextual information contained in the image data, i.e. image data is treated as generic matrix of numbers.

Inspired by the development of image processing techniques, this thesis proposes a new approach
that explicitly uses the contextual information of image data. In this work a novel framework of hardware architecture, given the name of “Context-based Image Acquisition” (CbIA), is introduced which involves the design of a stand-alone image acquisition function block that allows a loss of image quality in exchange for reduced cost of image acquisition process. It is required to be independent of both source memory and the client application to achieve universal compatibility to existing systems. It is also required to be able to reduce the overall cost in bandwidth, acquisition time, and energy consumption via sampling and reconstruction of the target image data. In the next chapter, details of the proposed CbIA framework are explained and discussed.
Chapter 3

Proposed Solution

In this project, the idea of “Context-based Image Acquisition” (CbIA) is introduced to reduce the cost of image accessing process within hardware systems. This idea originates from observation of the biological fact that human brain deals with input visual data in a dynamic way [OF97]. Image data, or any natural signal data to a larger extent, is well structured and spatially/temporally correlated in many cases. These featured characteristics naturally promote the thinking of understanding and therefore intelligently processing the data, instead of treating the data as plain sequence of numbers. This idea serves as the motivation of modern image processing researches. In this project however, it is used to enable an alternative thinking about hardware architecture design.

In this chapter, the proposed framework of Context-based Image Acquisition is explained. A generic scenario is set to accommodate the proposed hardware framework (section 3.2). The energy cost of image accessing process in hardware systems is discussed in section 3.3. Finally in section 3.4 a high level analysis of the proposed framework is provided to gauge the potential of trade-offs enabled by the system, which also serves as a guideline to the evaluation of system performance in the rest of the thesis.
3.1 The Idea of Context-based Image Acquisition

Image accessing within hardware system can be simplified to the scenario shown in Figure 3.1 (repeated from Chapter 1), where a client image processing application is implemented on a computing engine and it requests for access of image data from a memory external to the housing computing engine. The effort spent on this accessing process is considered to be of increasingly higher impact to the overall performance of the whole system. Such effort/cost includes but is not limited to:

1. **Acquisition time**: the interval from the time when the acquisition order is issued to the time when the requested image data is acquired from the source memory. This is directly related to how much time it is going to take for the complete image processing task to finish.

2. **Energy consumption**: the overall energy required to perform the acquisition of target image data, including the energy consumption of both the source memory and the computing engine.

3. **Bandwidth**: the total amount of data (measured in bits) accessed from the source memory and transmitted to the computing engine in a fixed period of time. The bandwidth requirement here measures how frequently the source memory is occupied during the image acquisition process. Lower bandwidth requirement means the source memory can be freed to serve other potential hardware entities.

These efforts are becoming more dominant to the performance of an image processing system due to the increasing resolution and size of image data, as well as the advancing in hardware technology scale. Reducing these effort has attracted plenty of research interests, and is indeed the main objective of this project. In the rest of the thesis, these metrics are collectively denoted as “the cost of image acquisition”.

Various research has approached this problem of accessing cost from different perspectives, aiming to reduce the effort of the image data acquisition process (reviewed in chapter 2). As
Chapter 3. Proposed Solution

Figure 3.1: The basic scenario for image acquisition consists of a source memory that contains the target image data to access, and a computing engine that houses the client image processing application that requests for the image data.

Figure 3.2: Image accessing with optimization. Evolving from the basic scenario of image acquisition, existing methods work on the development of memory hierarchy and access pattern optimization (highlighted in red). (reviewed in chapter 2)

is shown in Figure 3.2 existing methods of reducing accessing effort involve the introduction of memory hierarchy and optimization of accessing patterns [KP01, LZ12]. Particularly in image processing systems, local buffering memory such as scratchpad memory is often used to bridge the speed difference between computing engine and source memory. This leads to a reduction of overall image acquisition cost and latency from the external source memory. Application dependent optimizations of accessing patterns are also employed to reorganize the data sequence, which helps in reducing the accessing and transmission cost of the image data. Most of these techniques treat image data as sequences of numbers that are to be acquired from external memory.
3.1. The Idea of Context-based Image Acquisition

Different from conventional approaches, the proposed “Context-based Image Acquisition” framework approaches the problem from a different perspective. The core idea of CbIA framework is to allow the hardware computing engine to dynamically and adaptively acquire a target piece of image data from a source memory. Instead of accessing every bit of stored data from the source, CbIA architecture is designed to be able to select and sample particular fractions of data and fill in the missing parts by combining the sampled data with a certain level\(^1\) of prior knowledge learned from natural images. The end result of this alternative accessing process is a reconstructed approximation to the original image, which is stored in the local scratchpad memory and used by a client application. By reducing the number of times of memory accessing actions, the proposed CbIA framework aims to reduce the overall cost of image acquisition process.

Without losing generality, Figure 3.3 explains the difference between the conventional accessing process and the proposed process. As is shown in Figure 3.3(a), the conventional image accessing process\(^2\) often sees the pixels of the target ground truth image being accessed in a sequential order. The memory accessing interface of the computing engine requests for data by providing

---

\(^1\)The different levels of involvement of prior knowledge is elaborated through the rest of the thesis.

\(^2\)Referring to chapter 2 for the discussion of conventional image accessing methods.
addresses to the memory. Each pixel is accessed in this way and transmitted from the source memory back to the local image buffering memory, as is described in section 2.2. The image data is not ready and cannot be used before time point $t_0$. At the end of the process, the ground truth is completely moved to the local image buffering memory.

The altered accessing process sees the CbIA architecture maintains a modelling of the target image and from the model, it requests for pixels that are considered of most “significance” (Figure 3.3(b)). The “significance” is defined to be the estimated potential of a pixel to improve the quality of the image reconstruction with samples at the end of the process, which will be used as an approximation to the ground truth image. Starting from some initial coarse sampling pattern, the proposed CbIA architecture identifies candidate pixel locations that are most significant and asks for these pixels from the source memory. After acquiring these pixels, the modelling of image statistics is updated accordingly, providing another set of candidate pixel locations of high estimated significance. The architecture iteratively performs these steps to progressively acquire more pixels from the source memory and refine the internal prediction of the image. At the point ($t_{IS}$ in Figure 3.3(b)) when the architecture determines that enough samples have been acquired (to be elaborated later in the thesis), the sampling process stops and the source memory is freed to be used by other computing engines if there is any. However, the image is not ready at this point. The rest of the missing pixels are filled in by reconstruction methods using existing samples. At the end of the reconstruction process ($t_{IA}$ in Figure 3.3(b)), an approximation of the ground truth is completed within the local image buffering memory, serving as the substitute of the ground truth.

This altered accessing process is dynamic in that, it can be tuned according to available resources in the current environment, such as bandwidth, energy consumption, and time consumption etc. By allowing a reduction in the quality of the acquired image, both $t_{IS}$ and $t_{IA}$ can be changed by demand unlike $t_0$ which is always fixed.

It is worth noting that while the conventional image accessing process requires to occupy the source memory throughout the process, the proposed method releases the source memory early on. Therefore in the rest of the thesis, the following two terms are defined:
3.2 Scenario Setup

1. **Image sampling process** refers to the process of accessing actual pixels from the source memory, i.e. the time period upto \( t_{IS} \). This is directly related to the bandwidth requirement of the image acquisition process.

2. **Image acquisition process** refers to the complete process starting from the client application requiring to access an image data from source memory, till the end of reconstruction when the approximation of the ground truth is ready and returned to the client application. (from start to \( t_{IA} \))

From the perspective of the client application the data is not ready until the end of the complete image acquisition process. Although that makes the time of the image acquisition process a major performance metric, a short image sampling process still carries benefits as it can reduce the bandwidth requirement of the memory. Between \( t_{IS} \) and \( t_{IA} \) the source memory is freed and is available to be used by other entities that potentially exist alongside the computing engine in question. This brings flexibility to the design and execution of a system in larger scale. Therefore, the bandwidth requirement is also part of the main metrics that this project addresses for.

### 3.2 Scenario Setup

There are various hardware computing engines that are commonly used in modern times. Starting from the custom hardware platforms (ASIC and FPGA, etc.), to the commercially popular general purpose computers, these computing engines all have their own structure and individual features. There are also a large collection of different designs of memory chips that all have distinguishing accessing protocols, making the collection of different hardware system designs even richer.

The idea behind CbIA is to introduce such a framework of hardware architecture which does not conflict with existing hardware protocols while providing the functionality described in the previous section. It endeavours to have universal compatibility with existing hardware systems.
and therefore is easy to use and of low implementation cost. In this project however, the custom hardware is targeted as the research platform and the following scenario is set on which the problem is discussed (Figure 3.4).

Figure 3.4: Scenario setup. The proposed CbIA architecture replaces the conventional image accessing process (Figure 3.2) with a dynamic and progressive sampling procedure (marked in red).

This scenario sticks with the core mechanism of digital image data access within hardware systems, despite the possible varying designs of either end of the transmission. It is based on the very fundamental addressing-accessing mechanism of memory accessing protocols. It does not conflict with common custom hardware structures (e.g. ASIC, FPGA) and allows for a rich selection of source memories to act as the holder of the target image data.

Within the computing engine, an implementation of a client image processing application is installed which issues the order to fetch the target image data. The data required is stored in the source memory in the form of pixel matrix. This source memory must have a certain degree of random accessibility: data content in this memory should be able to be accessed by addressing. A local buffer memory is embedded between the client application and the source memory, as is the case for most image processing systems (refer to section 2.2).

The CbIA architecture is responsible for the image acquisition process. According to the request from the client application to access a region of the target image, the architecture computes and generates an optimized sequence of addresses for the pixels to be accessed. The architecture works on the local buffer memory, storing sampled pixels and filling in the missing parts by
reconstruction algorithms. At the end of the process, an approximation of the ground truth exists in the local buffer, ready to be used by the client application.

The designed CbIA architecture can serve as a reference for the application of CbIA procedure in general purpose processors such as CPUs and GPUs. While the concept of progressive acquisition of macroblocks is the same across different hardware platforms, the application of CbIA to CPU/GPU however is not straight forward and poses different challenges. The discussion of the subject is not within the scope of this thesis, and is part of the future extension of this project.

Although in this thesis, discussions are focused on accessing image data from source memory chips, this source may as well be of other types such as cameras or other image capturing sensors. The proposed method requires the data source to have a certain degree of random accessibility in order to function, as is the requirement for memory chips. The proposed architecture is expected to be applicable in any situation where the accessing of data is relatively costly compared with computational effort, i.e. where a dynamic trade between accessing effort and computational effort is desired.

This scenario serves as a guideline for the design of CbIA architectures so that the design and discussion of the idea are applicable to any hardware environment that agrees with the scenario.

3.3 The Baseline of Energy Consumption in Hardware Systems

Apart from the adaptivity to situations, to allow a loss of image quality also opens other beneficial trade-offs. As is part of the motivations driving this project, due to a) the performance gap between memory and processor, and b) the increasingly dominating presence of the data accessing effort, it is generally considered beneficial to reduce the work of data accessing by performing more local computations to compensate for the loss.

While the bandwidth requirement and acquisition time are major concerns here, the benefit
and impact of reducing $t_{IS}$ and $t_{IA}$ is obvious. However, the change to the energy consumption of the image acquisition process is more difficult to gauge. Therefore a baseline is drawn in this section about energy consumption in state-of-art hardware environment. An experiment is conducted to provide numbers about energy consumption of basic operations in practical custom hardware systems. In later chapters where advanced algorithms are explained, these statistics will serve both as a reference and as a foundation of predictions into the future development.

### 3.3.1 Experiment Setup

To establish a baseline about energy consumption in hardware systems, the basic operations “ADD” (fixed-point addition) and “MULT” (fixed-point multiplication) are run on a chosen FPGA chip (detailed in the following sections) respectively. These two operations are fundamental components of digital computation and therefore their energy consumption can serve as a reference for more complex operations. This experiment sets up a testing platform on reconfigurable hardware where these two basic operations run and their energy consumption is recorded. The small system is synthesised and placed & routed on FPGA, and is tested by a testbench that generates random inputs. Signal activities during the test are recorded to generate a report of estimated energy consumptions of the target operation.

The use of FPGA as testing platform is to benefit from its customizable structure. On FPGA the test can be focused on the very core of the implementation of target operations, with minimum interference from surrounding hardware elements. Compared with ASIC, FPGA implementations are easier and faster to design due to the reconfigurable nature of FPGA chips.

This experiment makes use of the Quartus II software from Altera for synthesising and place & route. Input and output data are all registered inside the design (marked by red dashed rectangle in Figure 3.5). The synthesised design is called by the testbench in Modelsim to run the test (> 10,000 operations). Modelsim dumps signal activities within the design to a .vcd
3.3. The Baseline of Energy Consumption in Hardware Systems

Figure 3.5: Setup of the baseline energy consumption test on FPGA.

The .vcd file is then analysed by Quartus II which provides a detailed report of estimated energy consumption of the operations.

These reported values are compared to energy consumption of accessing one pixel from various types of source memories (excluding the energy of transferring the data). Listed in this section are statistics of memories that exist on Cyclone II and Stratix IV development boards, designed by Altera. The large amount of memory designs and varying working conditions make it difficult to make a generalised and conclusive estimation of energy consumption of data accessing. Therefore the comparisons and discussions in this section are only to provide a basic reference to the modern hardware system.

### 3.3.2 Results and Discussions

Two Altera FPGAs are chosen as representatives of two different technology scales (Table 3.1). Additionally, to provide data regarding potential ASIC environment, Altera Hardcopy IV “structured ASIC” is also chosen, which has its characteristics listed in Table 3.2.

Running on these platforms, the implemented operations (ADD and MULT) have their corresponding energy consumption recorded in Figure 3.6. The test is run on three different I/O
The Cost of Computation

Compared with other categories of energy consumption the logic effort of the actual computation, labelled as “Logic” in the plots, is insignificant. While the other types of energy consumption are more determined by the structure of the housing chip, the logic effort is decided mainly by the technology scale of the chip and therefore is a reliable reflection of the actual computational cost of the operations.

The other types of energy consumption (namely for routing and temporal storage) are spent for supporting the core operations. These supportive costs are important and non-negligible parts of the total energy cost of running modern hardware systems. The exact energy costs of these works are dependant on the size and complexity of the design, as well as the degree of optimization of the design.

The Cost of Maintaining Clock Signals

It can be seen that maintaining the clock signal – routing clock signals to different parts of the chip – consumes the dominant amount of energy in this experiment. This is the case through different settings of I/O width, types of chips, and types of operations. The amount of effort to maintain clock signals on a chip is determined largely by the area and wire placement of

<table>
<thead>
<tr>
<th>FPGA type</th>
<th>Model ID</th>
<th>Technology</th>
<th>Equivalent LEs</th>
<th>On-chip multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclone II</td>
<td>EP2C20F484C6</td>
<td>90 nm</td>
<td>18,752</td>
<td>18x18 embedded multiplier</td>
</tr>
<tr>
<td>Stratix IV</td>
<td>EP4SGX530KH40C2</td>
<td>40 nm</td>
<td>531,200</td>
<td>18x18 DSP multiplier</td>
</tr>
</tbody>
</table>

Table 3.1: Chosen FPGA chips specifications.

<table>
<thead>
<tr>
<th>FPGA type</th>
<th>Model ID</th>
<th>Technology</th>
<th>Equivalent LEs</th>
<th>On-chip multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardcopy IV</td>
<td>HC4GX35FF1517</td>
<td>40 nm</td>
<td>9,003,878</td>
<td>18x18 DSP multiplier</td>
</tr>
</tbody>
</table>

Table 3.2: Chosen structured ASIC specifications.

width settings, namely 4-bit, 8-bit, and 16-bit. The bar plots in Figure 3.6 show the breakdown of energy consumption of target operations on chosen FPGA chips and structured-ASIC. Discussion on the results is given in the following sections.
3.3. The Baseline of Energy Consumption in Hardware Systems

Figure 3.6: Breakdown of energy consumption.
Chapter 3. Proposed Solution

the implementation. Given that in this experiment, only a simple ADD or MULT operation is implemented on an otherwise powerful chip, the dominating presence of the effort to maintain clock signal is reasonable. It is also worth noting that Stratix IV is a more advanced chip than Cyclone II, offering more LEs and more sophisticated on-chip subsystems such as DSP blocks. However it does come with drawbacks as the routing of clock signals through Stratix IV costs more energy even when the technology scale is smaller. On the other hand, the logic and routing effort on Stratix IV costs less than that of the Cyclone II chip thanks to the advanced technology scale. Therefore while it is interesting to see the whole picture of energy consumption breakdown, it is beneficial to ignore the cost of clock maintenance when discussing the performance gap between processors and memories.

The Margin Between Computational Cost and Cost of Accessing Data from Memory

The purpose of this experiment is to draw a baseline reference of the performance gap in terms of energy consumption, between processors and memories. To make the comparison, memory chips that exist on the development boards are referenced as representatives of common memory chip choices that accompany the processors of their corresponding technology scale.

Table 3.3 and Table 3.4 list the characteristics of memory chips on the development boards designed for Cyclone II and Stratix IV respectively, by Altera.

<table>
<thead>
<tr>
<th>Memory type</th>
<th>DDR[3]</th>
<th>Flash</th>
<th>SSRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface frequency (MHz)</td>
<td>200</td>
<td>10</td>
<td>250</td>
</tr>
<tr>
<td>Data bus frequency (MHz)</td>
<td>400</td>
<td>10</td>
<td>250</td>
</tr>
<tr>
<td>VDD (V)</td>
<td>2.5</td>
<td>3</td>
<td>3.3</td>
</tr>
<tr>
<td>IDD (mA)</td>
<td>200</td>
<td>30</td>
<td>350</td>
</tr>
<tr>
<td>Interface width</td>
<td>8-bit</td>
<td>8-bit</td>
<td>18-bit</td>
</tr>
<tr>
<td>Energy/access (nJ)</td>
<td>1.25</td>
<td>9</td>
<td>4.62</td>
</tr>
<tr>
<td>Normalised energy/access (to 8-bit)</td>
<td>1.25</td>
<td>9</td>
<td>4.11</td>
</tr>
</tbody>
</table>

Table 3.3: Memories on Cyclone II development board, Altera [Alt07].

3Works in burst read mode. IDD4R is used as working current
4Works in burst read mode. IDD4R is used as working current
3.3. The Baseline of Energy Consumption in Hardware Systems

<table>
<thead>
<tr>
<th>Memory type</th>
<th>DDR3</th>
<th>Flash</th>
<th>QDR II SRAM</th>
<th>SSRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface frequency (MHz)</td>
<td>667</td>
<td>52</td>
<td>400</td>
<td>250</td>
</tr>
<tr>
<td>Data bus frequency (MHz)</td>
<td>1333</td>
<td>52</td>
<td>800</td>
<td>250</td>
</tr>
<tr>
<td>VDD (V)</td>
<td>1.5</td>
<td>1.8</td>
<td>1.8</td>
<td>2.5</td>
</tr>
<tr>
<td>IDD (mA)</td>
<td>200</td>
<td>21</td>
<td>690</td>
<td>450</td>
</tr>
<tr>
<td>Interface width</td>
<td>8-bit</td>
<td>16-bit</td>
<td>8-bit</td>
<td>18-bit</td>
</tr>
<tr>
<td>Normalised energy (nJ)/access (to 8-bit)</td>
<td>0.225</td>
<td>0.365</td>
<td>1.55</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Table 3.4: Memories on Stratix IV development board, Altera [Alt10].

Image data is often in the format of 2-dimensional (grayscale) or 3-dimensional (colour image) data matrix. In this experiment and the discussions in the rest of the thesis, grayscale images are used with the grayscale value of each pixel represented by an 8-bit integer, ranging from 0 to 255. Therefore despite the varying I/O width configurations of the above mentioned memories, their energy consumption of accessing is normalized to have an equivalent of 8-bit I/O. In practice, these memories are optimized to have different I/O widths and therefore it is infeasible to simply normalize their energy consumption by data width. Such normalization only serves to provide a general performance measurement of the various types of memory, without being limited to a specific application.

The last line of the two tables above shows the normalized energy consumption of accessing one 8-bit pixel from a given memory (excluding the energy of transferring the data). It can be seen that the energy consumption of accessing one pixel is significantly larger than the logic effort of performing one 8-bit ADD or MULT operation. For example, the DDR3 memory on Stratix IV development board is of the lowest energy/access value equal to 0.225 nJ per access. The costs of energy to perform an 8-bit ADD and MULT computation on Stratix IV are 0.0045 nJ and 0.0092 nJ respectively (omitting clock maintenance). This means that by accessing one fewer pixel from the DDR3 memory in question, the 40 nm logic elements on Stratix IV can afford to perform about 50 ADD or 25 MULT operations. This is the margin of energy that the proposed system can make trade-offs to achieve an overall reduction of energy consumption of the complete image acquisition process.

It is worth emphasising again that the experiment conducted in this section simplifies the
otherwise complex architectures of hardware systems in order to provide a basic reference line without losing generality. In practice, there are other sources factoring into the energy consumption of data accessing process, such as the energy cost of data transmission between hardware blocks. It is the purpose of this section of discussion to show that there is a margin to make trade-offs. This brings potential to the proposed framework to reduce the total energy consumption of image acquisition process.

**Towards ASIC Implementation**

The structured-ASIC provides an estimation of system performance as close as possible to the actual ASIC design. When fully optimized and ripped off of unnecessary surrounding circuits, the implementation of operations on ASIC will have a much lower energy consumption than that on FPGAs. From the plots in Figure 3.6 it can be seen that the energy consumption of the operations is significantly less than that on both FPGAs, even though Hardcopy IV is of the same 40 nm technology as Stratix IV is.

Although it can be applied as an IP core for designing reconfigurable systems, the proposed idea of CbIA architecture is ideally implemented on ASIC as a customized hardware block. The difference of cost (both in time and energy) between FPGA and ASIC offers an even increased margin of trade-offs than that discussed above. For example, the energy cost of performing an 8-bit ADD and MULT operations are 0.0009 nJ and 0.0035 nJ respectively. Compared with Stratix IV, which is of the same 40 nm technology, Hardcopy IV performs logic computations at an even lower cost: by accessing one fewer pixel from the same DDR3 memory on Stratix IV development board, about 250 ADD or 64 MULT operations can be performed.

### 3.4 Analysis of the Proposed Framework

The essence of the proposed idea of context-based image accessing is to reduce the number of times of memory accessing, to trade image quality for a potentially faster and/or less energy consuming image acquisition process.
The trade-off is enabled by the use of explicit or implicit understanding of natural image data. The sampling stage of the context-based accessing is to de-correlate the otherwise spatially correlated images. The reconstruction stage after sampling is to fill in missing parts by inserting correlation back to the incomplete image. The explicit or inexplicit prior knowledge learned about image data plays parts in both stages. It can be seen that to solve the same problem, the fundamental concept of CbIA-enabled memory framework takes an opposite approach to existing optimization methods for memory access patterns. While the existing optimization methods focus on arranging stored data or accessing sequence to be more spatially correlated, context-based accessing aims to de-correlate the sequence and reduce the length of it. The concept of de-correlating sampling sequence is also against the design of the modern memory mechanism of pre-fetching, which is to reduce the effort of addressing by operating the memory in burst mode. More detailed discussions about the compatibility of the proposed framework are given in later chapters.

For the Context-based Image Acquisition, the focus of the trade-off is to reduce the overall cost in bandwidth, time, and energy of acquiring an image from a source memory. However the implementation of the controlling mechanism and the reconstruction required by the nature of the proposed idea both introduce cost overhead. In this section the metrics of the system are listed and discussed, providing a guideline for more detailed research about context-based image acquisition.

**Loss of Image Quality**

The reconstructed image by CbIA architecture serves as an approximation to the ground truth which is stored in the source memory. This introduces quality loss of the image and it will be carried over to client applications. Making efficient trade with image quality is essential to Context-based Image Acquisition. Therefore the design of the CbIA architecture is to achieve the goal of accessing as few pixels as possible while producing an approximation of as high quality as possible.
Chapter 3. Proposed Solution

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{pat}$</td>
<td>pixel access time</td>
</tr>
<tr>
<td>$t_{pao}$</td>
<td>pixel access overhead</td>
</tr>
<tr>
<td>$t_{clo}$</td>
<td>control logic overhead</td>
</tr>
</tbody>
</table>

Table 3.5: Notations used in Eq 3.1

**Reduced Bandwidth Requirement**

By sampling only fractions of data from the target image stored in source memory, the CbIA procedure reduces the amount of times of memory accessing. This is directly related to the required bandwidth of memory databus. Throughout the discussion of CbIA architecture and its data accessing procedure, the image quality vs. bandwidth (measured in bit per pixel) is an important metric. A lowered bandwidth occupation is beneficial to the overall system in that it bridges the potential speed difference between memory and computing engine. Moreover, it also means that the memory is not occupied during the whole image acquisition process and therefore has the ability to serve other hardware entities that potentially exist in the whole system.

**Time Cost of Image Acquisition**

Reducing the total time of image acquisition is beneficial to image processing systems. As is explained in section 3.1, the altered image accessing process of CbIA divides the total image acquisition process into image sampling process and image reconstruction process. The time cost $t_{IS}$ of the image sampling process is comprised of:

$$t_{IS} = t_{pat} + t_{pao} + t_{clo}$$

where $t_{pat}$ is the time needed to retrieve the requested pixels from the memory; $t_{pao}$ is the time overhead for navigating through the memory, such as for activating new pages in DRAMs; $t_{clo}$ is the overhead of executing control mechanisms that cannot be hidden within the accessing process.
The CbIA-based method can reduce the pixel access time $t_{pat}$. But because of the de-correlative nature of the access sequence issued by CbIA-based method, the access procedure might require to switch accessing locations in the source memory more frequently than conventional accessing method does. This may introduce switching overhead $t_{pao}$ in some memories, with DRAMs being an example where row switching results in energy overhead. To dynamically and adaptively decide accessing locations during the sampling process is to complicate the address generating process. While the conventional address generator in a memory accessing interface is simply an integer counter, it is more complex in CbIA architecture and therefore introduces additional overhead time $t_{clo}$. With careful design, this additional work can be paralleled with the actual memory accessing process to some extent. But in general, this overhead limits how complex the algorithms can be if to be applied in CbIA-based architecture.

The total acquisition time $t_{IA}$ of the image data comprises also the additional time overhead from image reconstruction process:

$$t_{IA} = t_{IS} + t_{ro}; \ t_{ro} = \text{reconstruction overhead}$$

Similar to the control mechanism, the reconstruction process can be designed to have a certain level of execution overlapping with the sampling process. However in the worst case scenario, every missing pixel will need to be reconstructed from existing samples. Therefore for the CbIA architecture to reduce the total time of image acquisition, the average time cost of reconstructing for one pixel has to be smaller than the average time cost of retrieving that pixel directly from the memory. This limits the complexity of the reconstruction algorithms.

**Energy Cost of Image Acquisition**

The equation of energy consumption for CbIA process is:

$$E_{total} = E_{pae} + E_{pao} + E_{clo} + E_{ro} + (E_{trans}) \quad (3.2)$$

where the entry $E_{trans}$ is the energy cost of data transmission between source memory and
Notation | Definition
--- | ---
$E_{pae}$ | pixel access energy consumption
$E_{pao}$ | pixel access overhead
$E_{clo}$ | control logic overhead
$E_{ro}$ | reconstruction energy overhead
$E_{trans}$ | cost of data transmission

Table 3.6: Notations used in Eq 3.2

computing engine.

As is discussed in section 3.3 the margin of energy consumption exists between computational effort and memory accessing process. This is inspiring to the idea of CbIA because it is the margin where the energy overhead of control mechanism and reconstruction process can be hidden. Moreover, reducing memory accessing times not only reduces $E_{pae}$ but also the data transmission cost $E_{trans}$ proportionally. Although in the simplified scenario setup of this project the cost of data transmission is omitted, it is still an important part in practice and is an advantage brought by CbIA-based method over conventional methods. In general, to reduce the overall energy consumption of the image acquisition process, the average energy spent to reconstruct for a missing pixel has to be smaller than that of accessing one pixel from the memory.

### 3.5 Conclusion

In this chapter, the concept of Context-based Image Acquisition is proposed to reduce the cost of image data accessing from memory. The framework of a CbIA-based hardware architecture is explained in a generic scenario targeting custom hardware. The potential of this proposed framework is briefly investigated based on the understanding of general hardware systems, as well as a set of preliminary experiments on reconfigurable hardware platforms. Based on the general guidelines and references provided in this chapter, the design and implementation of CbIA architecture is discussed and evaluated in the remainder of this thesis.
Chapter 4

Design of a Prototype CbIA Architecture

4.1 Introduction

Based on the proposed concept of Context-based Image Acquisition, a hardware CbIA architecture is designed and implemented on custom hardware. The design of the CbIA architecture is to investigate the validity and potential of the idea in practical hardware systems of modern technology scale. The designed architecture aims to achieve a reduced cost of image acquisition (bandwidth, time, and energy consumption as is defined in the previous chapter) by the adaptive sampling and reconstruction procedures.

The design of the CbIA architecture follows the framework of Context-based Image Acquisition introduced in Section 3.1 and is based on the scenario set in Section 3.2. The core of the architecture is image point sampling which helps the architecture to achieve the progressive data acquisition process described in Section 3.1. A review of related image point sampling algorithms is provided in Section 4.2.

Among various custom hardware platforms, FPGA is chosen as the evaluation platform due to its customizable structure and relatively low design cycle. Design tools provided by major
FPGA manufacturers such as Altera and Xilinx are well developed, offering a well supported design flow and reliable evaluation data. In the ideal situation of ASIC implementation of the proposed system, overhead cost (Section 3.4) introduced by the complication of the accessing process can be minimized. In this project, structured-ASIC chips from Altera are used to estimate the ASIC implementation performance because of its low design cost and easy compatibility with FPGA design.

In this chapter, the design of the CbIA architecture is described. The prototype system makes use of basic models of natural images. It sticks to the bare bone of the proposed concept without sophisticated algorithms, to establish a practical and demonstrative model of the proposed idea. This architecture is evaluated from various aspects discussed in section 3.4, showing its capability of making trade-offs between image quality and cost metrics of image acquisition.

The main contributions of this chapter includes:

1. A hardware architecture of the proposed CbIA framework is designed and implemented on FPGA and structured-ASIC devices.

2. A set of evaluations are conducted on the implemented architecture, demonstrating the potential of the proposed framework in reducing the cost (bandwidth, time, and energy consumption) of image acquisition process in practical hardware environment.

4.2 Progressive Sampling of Images

The proposed CbIA architecture is based on the image point sampling procedure to achieve the progressive acquisition of image data. In this section, the closely related research field of Progressive Image Transmission is briefly reviewed.

*Progressive Image Transmission* (PIT) is a family of methods that aims to make efficient use of the available communication bandwidth to transmit large image data [Tzo86]. The sampling procedure of PIT can stop at any time during the transmission and an approximation to the
4.2. Progressive Sampling of Images

ground truth image can be reconstructed which serves as a substitute of the original image. Algorithms designed for PIT are able to rearrange the order of transmission so that “significant data” is transmitted first. The “significance” is application dependent and it is most commonly defined as the potential of bringing a high improvement to the quality of reconstructed image.

PIT methods are classified into spatial domain and transform domain. The spatial domain methods order the image pixels so that the most significant information (such as the most significant bit of each pixel [CSC99]) is transmitted first in the bit stream. Some of the methods [RSM07, DDI06] identify the most significant information in the form of pixels and requires the receiver to approximate the ground truth image by interpolation or regression using the pixels received. The transform domain methods on the other hand, transform the original image from spatial domain to frequency domain and transmit significant frequency coefficients accordingly. Discrete Wavelet Transform (DWT) is a popular tool of analysing images and is included by many compression standards, such as the JPEG2000 [SCE01].

Most techniques in this family rely on the pre-processing of ground truth image to better organize the image data and therefore achieve a better quality to bandwidth ratio. The techniques of image progressive sampling however, specifically target the scenario where such pre-processing is not available and blind point sampling is the only option. In these situations, image progressive sampling relies on stochastic method that iteratively samples pixels while refining the underlying model. As established early in Adaptive Farthest Point Strategy (AFPS) by Eldar et al. [ELPZ97], the pattern of stochastic point sampling should be a) random to prevent aliasing, and b) farthest from current sampling pattern to increase effectiveness of the sampling process. The ability of AFPS-based sampling process to organize pixels according to their significance was demonstrated in the work of Eldar et al. [ELPZ97, DL07, DDI06].

In the framework of AFPS, the sampling of an image starts from a random coarse sampling pattern. Based on already sampled pixels, the next sampling locations are determined by candidates’ priority scores that are computed from the local estimated statistics in the neigh-
Figure 4.1: Example of AFPS-based sampling method. (a) Voronoi diagram on sampled pixels (blue dots); (b) Voronoi vertices are candidates to be sampled in the next iteration; (c) the vertex that has the highest priority score is chosen and sampled; (d) the newly sampled pixel updates existing Voronoi diagram.

The priority of a pixel $i$ in the sampled set is defined as

$$f(x_i) = \min_k \|x_i - x_k\|_2 \cdot \max_{k \neq i}(B_{min}(x_k, x_i)) \quad (4.1)$$

where $B_{min}$ is an estimation of local minimum bandwidth and $x_i$ is the coordinate vector of pixel $i$. The pixels $s_k$ are neighbouring pixels to $i$. In the work of Eldar [ELPZ97] the neighbourhood is defined to be the vertices of the Delaunay triangle that contains $i$ (Figure 4.1 and 4.2). Pixels sampled are used by interpolation algorithms to reconstruct an approximation to the ground truth image.

Under this formulation, the priority combines the Euclidean distance of coordinates and the estimated local variance to jointly estimate how significant a candidate pixel is. Pixels of high priority score are considered to be statistically significant as they are estimated to be of most potential variance in their values. Sampling pixels of significance is likely to reduce the reconstruction error in its local neighbourhood. In this way, the system is able to progressively and adaptively acquire pixels that can bring the most potential improvement to the quality of the reconstructed image, resulting in an efficient use of the available bandwidth. This work exploits the above concept within the remits of designing a hardware system in order to minimise the cost of the image acquisition process.
4.3 Design of the Sampling Procedure

4.3.1 Scenario Setup

For the design of the CbIA architecture, in this chapter the scenario in Figure 4.3 is set. The scenario remains true to the generic scenario set for the general Context-based Image Acquisition (section 3.2). To allow for a practical design, the target scenario in this chapter is more detailed than the generic scenario. Among various types of memories, popular SDRAMs are used as source memory which contains the target image to acquire. The architecture is implemented and evaluated on FPGA and structured ASIC devices because of their reconfigurability and short design cycle. The designed CbIA architecture makes use of the logic and storage resources on these reconfigurable hardware. In this scenario, the task of the CbIA architecture is to adaptively and dynamically acquire the target image from the external SDRAM, and eventually prepare for an approximation of the ground truth within on-chip memory for the
Figure 4.3: Scenario setup for the prototype system design.

4.3.2 Structure of Memory Systems

The family of Random Access Memories (RAMs) includes two major types of design: static RAM and dynamic RAM. Both of them are widely used in modern hardware systems. Static RAMs use more transistors than DRAMs to store a same amount of data. However because DRAMs are required to periodically refresh their content the controlling mechanism of DRAMs are more complex. The dynamic nature of data storage of DRAMs also introduces additional workloads of pre-charging bitlines and activating new lines, which increase the access time of data when random accessed.

State-of-art DRAMs mitigates the drawbacks of row switching overhead by arranging spatially or temporally correlated data within a single row/page of the memory. Assuming that highly correlated data is more likely to be requested in a subsequent order, this arrangement of storage is able to reduce the overall row switching activities and therefore reduce both the average access time and power consumption. As is discussed in section 3.4 the concept of Context-based Image Acquisition is to de-correlate data access sequence. While it makes little difference in the situation of SRAMs, it is however enlarging the drawbacks of DRAM designs.

Additionally, pre-fetching is an important feature of state-of-art memories. For one address
4.3. Design of the Sampling Procedure

received, a pre-fetching enabled memory returns the data of the requested address and also a
burst of data of the subsequent addresses, without waiting for further addressing command.
This not only bridges the speed difference between memory access interface and memory data
bus, it also reduces overall power consumption (less addressing activities). The use of pre-
fetching is also based on the assumption of the high correlation of stored data, and therefore is
not a feature that Context-based Image Acquisition can easily benefit from.

The design scenario set for the discussion in this chapter assumes the source memory in the
form of SDRAM (Figure 4.3), which is one of the popular commercial memory types that has
complex mechanism/structure, including the concerns described above. This configuration of
task is not only to make the prototype system of practical use, but also to make the discussion
in this chapter more complete.

4.3.3 Sampling Procedure

As is briefly reviewed in Section 2.2, image processing algorithms often access image data in
nested loops [KP01]. This process repeatedly asks for pixels from a local area, and each pixel
is used multiple times within a period of time when the algorithm is processing around its
neighbourhood. The size of the local area is determined by the innermost loop pair [KP01].
Because of this featured “block-type access”, hardware systems that are designed specifically for
image processing applications often use high speed high bandwidth local buffers to temporarily
store local regions of the target image. The buffered area is normally of size of a “macroblock”,
i.e. the block of region defined by the innermost loop pair. This buffering reduces the time and
energy cost of repetitively accessing data from source memory. The buffering of the block-type
image fractions also gave rise to various data organizing strategies developed to store image
data in SDRAMs [Lee03, YLY08] (Section 2.4). In this chapter, the discussion is focused on
the linear and block mapping strategy of image data as they are the most popular methods
(Figure 4.4). For the linear mapping strategy, pixels of an image are stored row by row within
the source memory, each row of the image corresponds to a single page of the SDRAM. The
block mapping strategy on the other hand stores each macroblock of the image in a single page
of SDRAM to minimize the number of row switching activity.

Context-based Image Acquisition does not change the way conventional memory hierarchy works but instead introduces adaptivity and intelligence to the memory accessing protocols. In the scenario defined in this chapter, the design of the CbIA architecture aims to adaptively and dynamically move macroblocks from the source SDRAM to the local buffer, which is the on-chip BRAMs/registers. For memory chips, data (pixels) is accessed by addressing and this basic memory accessing protocol is compatible with the concept of image progressive sampling. Therefore the design of CbIA architecture starts with basic point based image progressive sampling methods. (More advanced modelling and explicit use of more complex prior knowledge are discussed in later chapters.)

A possible design of image sampling is to uniformly refine the image. Starting from a relatively coarse sampling distance\(^1\), during each iteration the sampling process reduces the sampling distance by a factor of 2 and samples all missing pixels belonging to this sampling distance.

---

\(^1\)In this thesis, the "sampling distance" of image uniform sampling refers to how dense the sampling is. A sampling distance of 4 means the image is sampled every 4 pixels in both horizontal and vertical direction.
The process stops when available bandwidth is depleted and a reconstruction of image can be retrieved from the sampled pixels. Although this scheme requires only the minimum amount of controlling logic, it does not have any data adaptability, therefore the pixels sampled are not always statistically significant as is defined in the literature of PIT. This leads to the inability of the system to make efficient trade between bandwidth and image quality. Moreover, the step size in the number of samples between each sampling distance is fixed and is large (6.25% to 25% to 100%), rendering such strategy of limited use in practice. Such fixed step size limits the flexibility of the sampling procedure to adapt to current hardware environment, i.e. the procedure cannot stop at any time by request.

With the practical hardware scenario in mind, this work takes a different approach to uniform sampling and adopts the estimation of priority scores of candidate pixels. The use of variants of priority scores holds key to many progressive point sampling techniques. Most sampling procedures start with a coarse sampling pattern of the target image, and iteratively identify and sample the pixels of most significance to the improvement of the reconstruction quality. Although defined differently, variants of priority scores share a similar base concept. The priority score from Eq.4.1 is extended in this work to a more general form that describes such concept:

\[ f(x_i) = d_{x_i,P} \times v_i \quad P: \text{sampled pixel locations} \]  

(4.2)

where \( x_i \) is the coordinate vector of a candidate unsampled pixel, and the distance term \( d_{x_i,P} \) measures the likelihood of determining pixel \( x_i \) with existing samples. This distance therefore includes, but is not limited to Euclidean distance of pixel coordinates. The variance term \( v_i \) is the estimated variance of the distribution of pixel \( x_i \). An instance of progressive sampling using priority scores (denoted as Full Adaptive Sampling in this article) is shown in Figure 4.5. This adaptive sampling procedure works on regular grid and the priority of pixels is determined by the priority score of its containing image block:

\[ f(b_i) = \text{area}(b_i) \times [\max_j(p(v_j)) - \min_j(p(v_j))] \]  

(4.3)

where \( \text{area}(b_i) \) is the area of the block \( b_i \) and \( p(v_j) \) is the pixel value of one of the four vertices.
Figure 4.5: Progressive sampling methods: uniform sampling (top); full adaptive sampling (middle); proposed adaptive sampling (bottom).
of \( b_i \) at location \( \{ v_j = (x_j, y_j) | j = 1, 2, 3, 4; v_j \in b_i \} \). The area of the block \( b_i \) is measured by the total number of pixels contained in this block. In every iteration, the block of the highest score is refined to a finer resolution and the process keeps running until a user defined quality requirement is met or there is no more pixel to sample from. At each step of sampling 5 more pixels, this sampling procedure samples the pixels that are considered by the procedure to be the most significant. Therefore compared with uniform sampling, this sampling procedure has a much finer step size of sampling, being more responsive to the current hardware environment.

However it can be seen that such adaptive sampling is to some extent against the structure of existing hardware systems and DRAMs in concept. While the design of modern hardware systems emphasises the use of data locality to reduce the cost of accessing, full adaptive sampling utilizes the data locality in a different way. It decouples data transmitted in a stream in an attempt to achieve maximum entropy gain with a limited bandwidth. The design of such sampling procedure is based on the assumption that switching sampling location has no significant cost, which is not true for DRAM memory systems. Therefore this work proposes the Adaptive Refine procedure which adopts a modified process based on Full Adaptive Sampling but is more suitable for DRAM accessing (Figure 4.5 (c)).

The proposed Adaptive Refine procedure follows the same steps as the Full Adaptive Sampling but adaptively refines every block belonging to the current sampling distance, that has a priority score higher than a given threshold, instead of refining only the block that has the highest priority score at the moment. In practice, the threshold can increase gradually as well to adapt to currently available bandwidth. Although this Adaptive Refine process cannot guarantee a best sampling pattern in between different threshold levels, it still produces the same sampling pattern as full adaptive sampling does when each threshold is met. The Adaptive Refine procedure uses the threshold parameter to control the length of buffering candidate sampling addresses, which allows the architecture to better arrange the actual order of accessing these candidate samples from the DRAM. It also allows for deeper pipelining of the memory accessing process and the address generating process, while maintaining the data adaptivity of Full Adaptive Sampling.
4.3.4 Prior Knowledge of Point Sampling

The proposed concept of Context-based Image Acquisition is based on the understanding of the contextual meaning of natural image data. The fundamental assumption about natural images is that such data is not randomly generated and pixel values of an image are correlated. By learning from such correlation, the system is able to re-arrange data acquisition to make better use of otherwise limited bandwidth. This prior knowledge, or \textit{a priori}, also enables the reconstruction of the image using part of the original image data.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{lena.png}
\caption{The continuity of natural signal. Three rows of the image \textit{lena} are picked and their grayscale values plotted, showing the continuity of pixel values of each object.}
\end{figure}

For the design of the CbIA architecture, the proposed sampling procedure shares the same basic \textit{a priori} of natural images, which is signal continuity. As is depicted in Figure 4.6, natural objects and effects are assumed to have continuously changing properties and therefore when a picture of the scene is taken, the neighbouring pixels are likely to have similar values or form continuous signals. The boundaries between objects often creates significant change of pixel values and thus produces high frequency components. Image reconstruction algorithms such as image regression and image super resolution techniques are mostly compensating for the un-captured high frequency components. The reconstruction algorithm used in this chapter (interpolation) is also based on the assumption of signal continuity.
A significant difference of the scenario in the context of CbIA methods, compared with regression or super resolution, is that the sampling and the reconstruction process are dynamically interacting with each other. The sampling method designed for CbIA caters for the need of the selected reconstruction algorithm, accessing the most “significant” piece of information for the reconstruction process. In turn, the already sampled pixels produce information to update the modelling of the unknown target image and therefore guide future sampling process. This dynamic understanding-accessing iteration allows for the sampling and reconstruction processes to compensate for the weakness of each other, to jointly achieve a higher image quality at the end of the acquisition process. Therefore, the design of sampling-reconstruction algorithm pair is key to the design of CbIA image acquisition procedure. Extended discussions of the design of this algorithm pair and the use of more complex prior knowledge are provided in Chapter 5 and 6.

4.3.5 Evaluation of the Sampling Procedure

The various sampling procedures discussed above, including the proposed Adaptive Refine procedure, are evaluated via simulation in this section. Target images that used for this evaluation and the evaluations in the rest of this chapter are all of size 527x527. Each image is broke down to 31x31 non-overlapping macroblocks which are then processed by the various sampling procedures.

This test is to evaluate from a high level abstraction the impact these procedures bring to the problem of image progressive acquisition in hardware systems. The source memory that holds the target image is simulated by power models to give an estimation of the energy/time consumption of the memory accessing process. For this purpose, various 1Gb DDR3s are simulated by the power model from Rambus [Vog10] as the target SDRAM, and the test is also carried out on two smaller sized SDRAM memories modelled by the CACTI tool designed by HP [TMAJ08]. The SDRAMs simulated by Rambus model and CACTI model are of 55 nm and 45 nm technology respectively, while both are of 8 bit I/O and have a burst length of 8.

The image quality is measured by its Peak Signal to Noise Ratio (PSNR) in this thesis. There
are extensive discussions about the quality of images and it is often application dependent. Nevertheless, Mean Squared Error (MSE) and PSNR are often accepted as measurements of the distortion of a processed image. Although full images are shown as evaluation results, the sampling procedure for CbIA only works on macroblocks of these images. The acquired result of these benchmark images are displayed only as a reference of visual quality for the PSNR numbers. For image processing systems, macroblocks of size of \(2^n \times 2^n\) are often used due to the nature of digital systems. For the proposed method however, in order for the sampling procedure to uniformly refine image blocks, the size of the blocks has to be 3x3, 5x5, 9x9, 17x17, and etc. (Figure 4.7(a)). There are various approaches to mitigate the effect of the irregular block size. For example, when consecutive non-overlapping 4x4 macroblocks (Figure 4.7(b)) are requested by the client application, the proposed acquisition procedure can fetch 5x5 blocks that overlap by 1 row/column to minimize the overhead of computations on the last row/column. In the worst case scenario if a single macroblock of size \(2^n\) at random location is requested, the proposed acquisition procedure fetches a \((2^n+1) \times (2^n+1)\) macroblock without interpolating pixels on the last row/column to minimize the overhead cost. In the evaluation of the various sampling procedures and the evaluation of implemented CbIA architectures later in this chapter, the experiments treat each macroblock as an independent task and their size is set to be 17x17, providing performance measurements of the proposed method in ideal situation. In later chapters, the sampling algorithm is expanded to refining on irregular grid.

Figure 4.7: The size of macroblocks. Red and green rectangles mark blocks processed by the proposed system. Pixels marked in blue are samples taken during the refining process. The uniform refining of each block requires the size of blocks to be \((2^n+1) \times (2^n+1)\).
The proposed Adaptive Refine is evaluated against three reference methods for its ability to trade image quality (PSNR) for reduced bandwidth, access time, and SDRAM access energy. The references are as described in section 4.3.3: conventional accessing pattern that reads every pixel from SDRAM, uniform refine, and Full Adaptive Sampling on a regular grid.

Firstly, the core performance metric of image progressive sampling procedures is studied, which is the image quality vs. number of samples ratio. This test is performed on two datasets that contain a large variety of generic natural scenes, which are LabelMe Outdoor (LMO) \cite{LYT09} and SUN datasets \cite{XHE+10} respectively. The LMO dataset and SUN dataset jointly cover a total of 12254 outdoor and indoor images that represent natural scenes, and therefore are used in this evaluation and are considered to be representative for the generic images as a whole. Figure 4.8 shows the achieved PSNR of the reconstructed image as a function of the percentage of the pixels sampled. In this graph, performance lines of Full Adaptive Sampling and Adaptive Refine coincide because they provide the same sampling pattern at each threshold level and their difference lie in the flexibility of the procedure to optimize the actual order of address sequence sent to the memory.

It can be seen that only four data points exist for the reference uniform sampling procedure, corresponding to the sampling distances of 16, 8, 4, and 2 respectively from left to right. On the other hand, Adaptive Refine procedure is able to take small steps and progressively acquire more pixels to refine the sampling pattern, hence the populated data points through the graph. This is indeed one of the advantages brought by the Adaptive Refine method. When implemented in the proposed CbIA architecture, it allows the architecture to adapt to the changing environment around the computing engine, whereas fixed step method like the uniform sampling cannot.

In this test, both the Adaptive Refine and the reference uniform sampling method start with a uniformly sampled pattern of sampling distance 16. The Adaptive refine method is restricted to not go beyond sampling distance of 2 because any performance beyond this point is difficult to present and is also of little significance to the discussion of image sampling algorithms. There-

\footnote{Note that the conventional method (accessing all pixels) has infinite PSNR and therefore is not plotted in this graph.}
Figure 4.8: The performance measured in image quality vs. percentage of pixels sampled, tested on the combined LMO and SUN database. The lines show the average performance of the sampling procedures on different images, as well as half the standard deviation of the performance.

Before the performance of both methods meet at the points where 1.38% (sampling distance 16) and 28.03% (sampling distance 2) are sampled. The graph shows that the proposed Adaptive Refine, as original Full Adaptive Sampling does, improves the ratio of image PSNR vs. number of pixels sampled in between the starting and ending point (i.e. the performance curve is raised up). It uses limited bandwidth more efficiently than the uniform sampling method does, and it fills the large gaps between data points from uniform refine method. This makes the Adaptive Refine procedure of more practical use than uniform refine method.

Moreover, Figure 4.8 shows that the performance of sampling procedures is application dependent. At the same time it also shows a general trend of the image quality vs. percentage of pixels sampled ratio across different images. In order to further examine the problem in detail, in the rest of this chapter the discussions and evaluations will focus on several selected benchmark images, namely “lena”, “barbara”, and “boat”.
4.3. Design of the Sampling Procedure

Targeting these representative benchmark images, tests are carried out to analyse the upper limit of the performance of the proposed system in reducing bandwidth, time and energy consumption of the image acquisition process, temporarily ignoring the cost overhead introduced by the proposed architecture itself. For this test both linear and block mapping strategies are used. For linear mapping each row of the image is stored in a single page within the DRAM, whereas for block mapping each macroblock is stored in a single page such that the row switching activities are reduced while reading a macroblock.

Figure 4.9(a) shows the target image that needs to be acquired, where Figure 4.9(b) shows an instance of the reconstructed image from the proposed adaptive refine sampling method where the threshold was set to 600. Figure 4.10(a) shows the performance of the sampling procedures working on “lena”, which if compared with the previous graph of Figure 4.8 is representative and close to the average performance of the sampling procedures working on the full image dataset. Again, both Full Adaptive sampling and the proposed Adaptive Refine procedure outperform uniform sampling and these two procedures achieve the same performance in this department (bandwidth). Moreover, in this evaluation the way memory contents are mapped (linear or block) makes no difference to this performance metric.

The graphs in Figure 4.10(b)(c) are the corresponding SDRAM access energy and access time at each achieved PSNR level. Under the assumption of no cost overhead of operating the proposed CbIA architecture, these two metrics represent the overall cost in energy and time of the image acquisition process. Therefore these two graphs show an upper limit of the performance of the proposed CbIA architecture as well as the margin of trade-offs. Additionally, SDRAM access time in Figure 4.10(c) also represents the reduction in memory bandwidth requirement. Both graphs show that the introduced cost overhead (SDRAM access energy and time) from progressive sampling methods is more obvious on linear mapped memory content, but the proposed Adaptive Refine allows the system to more flexibly organise the accessing pattern, resulting in a much lowered cost overhead than Full Adaptive Sampling (blue lines vs. green lines). In the case of block mapped memory content such overhead is minimized, and therefore a greater reduction of access energy and access time can be seen. Nevertheless, for both mapping strategies an overall reduction in SDRAM occupation time and access energy can be seen with
Figure 4.9: Comparison between ground truth image and the reconstruction using pixels sampled at a threshold of 600.
4.3. Design of the Sampling Procedure

Figure 4.10: Evaluation of sampling procedure. From left to right, the data points of adaptive refine and full adaptive sampling algorithms in these graphs are results from threshold of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively; the data points of uniform refine algorithm are results from sampling distance of 16, 8, 4, and 2 respectively.
PSNR up to 35 dB.

As is discussed in section 3.4, apart from the cost overhead of SDRAM access, the implementation and execution of the proposed system also inevitably introduce overhead. Any access time or energy saved from the SDRAM side has to be compared with the cost of implementing the method. Results in Figure 4.10 shows an overview of the trade-off margin offered by the proposed sampling procedure. It shows the ability of the CbIA procedure to trade image quality for reduced number of access times. To reduce the overall time and energy cost of the image acquisition process, the assumptions explained in section 3.4 have to be met:

1. For the CbIA-based memory interface to reduce the total time of image acquisition, the average time cost of reconstructing for one pixel has to be smaller than the average time cost of retrieving that pixel directly from the memory;

2. To reduce the overall energy consumption of the image acquisition process, the average energy spent to reconstruct for an unsampled pixel has to be smaller than that of accessing one pixel from the memory.

This is discussed in the following sections where a proposed CbIA architecture based on Adaptive Refine procedure is implemented and evaluated on reconfigurable hardware platforms.

### 4.4 Hardware Structure of the Proposed Architecture

The proposed CbIA architecture of image acquisition is implemented on reconfigurable hardware devices. The implemented CbIA architecture is based on the Adaptive Refine procedure discussed above. It is responsible for progressively generating addresses of pixels to sample from a source memory, reconstructing an approximation of the original image data using sampled pixels, and finally store the reconstruction in a local buffer for the potential client application to use.
4.4. Hardware Structure of the Proposed Architecture

Figure 4.11 shows a block diagram of the system. In general the proposed design generates pixel addresses for the DRAM interface (Figure 4.11(a)) to access the requested macroblock from the original image. The CbIA procedure operates on a local buffer (Figure 4.11(b)) that is prepared for buffering the macroblock of the target image. Sampled pixels are filled into this buffer and based on these samples, the system decides where to sample next. Starting from a coarse uniform sampling pattern, the system checks priority scores of existing blocks (Figure 4.11(c)) and refines their resolution accordingly (Figure 4.11(d)). After all blocks of the current sampling distance are processed, the system moves onto next resolution level. When the sampling process achieves a given quality threshold it stops and the remaining missing pixels in the buffer are filled by Bilinear interpolation (Figure 4.11(e)). During the process, the target macroblock is divided into a number of smaller blocks depending on the sampling statistics. The proposed design characterises each block by its resolution level and anchor, which is the coordinates of its upper left pixel. In detail, the system consists of three major units: refine_unit, addr_translator, and interp_unit. The connection and block diagram of these units are shown in Figure 4.11. The process is described in Algorithm 1.

Algorithm 1 The working mechanism of the proposed CbIA architecture

Require: An initial uniform sampling pattern; a given priority threshold $thr$; identities of initial blocks stored in FIFO_A.
Ensure: the acquired image macroblock

while sampling distance $> 1$ do
  while FIFO_A is not empty do
    refine_unit fetches a block stored in FIFO_A and check its priority score (Eq 4.3)
    if priority score $> thr$ then
      Stores the block in FIFO_C
      addr_translator fetches blocks from FIFO_C and generates the addresses of pixels to sample from these blocks
      Each block refined by addr_translator is broken down to four sub-blocks
      Newly generated sub-blocks are stored in FIFO_B
    else
      Stores the block in FIFO_D
      interp_units fetch blocks from FIFO_D and interpolate for the missing pixels
      The interpolation results are stored into the local canvas buffer
    end if
  end while
  FIFO_A and FIFO_B swap place
  Sampling distance divided by 2
end while
return the acquired image macroblock
Chapter 4. Design of a Prototype CbIA Architecture

An example is given in Figure 4.12. In stage 1 (Figure 4.12(a)), the anchor of block (1,1) is passed to *refine unit* to check for the priority of this 5x5 block (sampling distance of 4). It is classified as priority score $> \text{threshold}$ and therefore the anchor is passed to *FIFO}_C. This is then translated to the addresses of pixels to sample next. The newly generated blocks have their anchors stored in *FIFO}_B. In stage 2 (Figure 4.12(b)), the process is repeated but with *FIFO}_A and *FIFO}_B exchanging their roles. Blocks with anchors stored in *FIFO}_B are checked by *refine unit*. The blocks in this stage are all of size 3x3 (sampling distance of 2). This time only the block (3,3) is determined to still have priority score $\text{threshold}$. Therefore the rest of the three blocks ((1,1) (1,3) (3,1)) are passed to *FIFO}_D for interpolation process, while block (3,3) is translated into sampling addresses to further refine this block. The process keeps going until no more blocks need to be refined. The end result is the reconstructed approximation of the ground truth consisting of both sampled pixels and interpolated pixels.

4.5 Evaluation of the Designed CbIA Architecture

In this section, the designed CbIA architecture is evaluated on reconfigurable hardware platform for its impact to the overall cost of image acquisition process. Taking one step further than the simulation in section 4.3.5, the evaluation in this section takes into consideration the cost of running both the source memory and the computing engine.

The designed CbIA architecture was synthesised and placed and routed on Stratix IV FPGA (Table 3.1) and Hardcopy IV structured ASIC (Table 3.2). The SDRAM (source memory) response and the rest of the SDRAM accessing interface are both simulated with Modelsim testbench instead of being implemented. The generated SDRAM accessing addresses are passed to SDRAM power models designed by Rambus and HP, which in turn report the corresponding SDRAM energy consumption of the input access pattern. Again same as in section 4.3.5, various 1Gb DDR3s are simulated by the power model from Rambus [Vog10] as the target SDRAM, and the test is also carried out on two smaller sized SDRAM memories modelled by the CACTI tool designed by HP [TMAJ08]. The SDRAMs simulated by Rambus model and CACTI model
Figure 4.11: The structure of the proposed CbIA architecture. (a) The proposed design generates pixel addresses for the DRAM interface; (b) a local canvas buffer stores sampled pixels as well as interpolated pixels; (c) the refine_unit checks priority scores of each block; (d) the addr_translator generates sampling addresses if a block is to be refined; (e) an array of interp_units interpolates the missing pixels for blocks that do not need further refining/sampling.
Figure 4.12: Example of system working mechanism.
are of 55 nm and 45 nm technology respectively, while both are of 8 bit I/O and have a burst length of 8. Block mapped image content is used in the following test as it is the most popular storing strategy used in image processing hardware systems.

The synthesised architecture performs sampling and reconstruction of the selected benchmark images which are: “lena”, “barbara”, and “boat”. All of them are of size 527x527 and transformed to grayscale image with each pixel represented by a 8-bit value. The system works on non-overlapping macroblocks of size 17x17 on the target image. For each test, the architecture progresses the threshold \( \text{thr} \) score\(^3\) from 1800 (worst quality) to 150 (best quality), gradually increasing the number of samples.

In this set of evaluations, the proposed CbIA architecture is assumed to work at the same frequency as the memory data bus, i.e. it has the ability to random access a single 8-bit pixel from the memory without pre-fetching/burst-reading consecutive pixels. However, for the baseline performance (the conventional image acquisition method) the target image patch is read into the local buffer in burst mode with burst length of 8. Therefore the evaluation provided here is a lower bound performance of the proposed CbIA architecture.

In the following sections, the proposed architecture is compared with conventional address generator in SDRAM access interface, for the difference in image acquisition time and overall energy consumption of the acquisition process. It is worth noting that the cost in bandwidth is covered in section 4.3.5 and therefore is not repeated in this section. Besides the evaluation of the performance of the CbIA architecture, another evaluation is provided on the impact of the proposed CbIA procedure to an image compression application. Finally the mapping from FPGA implementation to ASIC implementation is discussed.

\(^3\)The priority threshold set here is for research purpose as this range of priority scores roughly covers the performance of the proposed design sampling 5% to 20% of total pixels.
Chapter 4. Design of a Prototype CbIA Architecture

<table>
<thead>
<tr>
<th></th>
<th>Combinational ALUTs</th>
<th>Logic registers</th>
<th>Block RAM usage</th>
<th>DSP block 18-bit elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>refine_unit</td>
<td>147</td>
<td>101</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>interp_unit (x3)</td>
<td>919</td>
<td>474</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>addr_translator</td>
<td>94</td>
<td>72</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FIFOs</td>
<td>971</td>
<td>365</td>
<td>1751</td>
<td>9</td>
</tr>
<tr>
<td>control</td>
<td>83</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>2214</strong></td>
<td><strong>1027</strong></td>
<td><strong>1751</strong></td>
<td><strong>36</strong></td>
</tr>
<tr>
<td><strong>Total (%)</strong></td>
<td><strong>0.61%</strong></td>
<td><strong>0.008%</strong></td>
<td><strong>0.7%</strong></td>
<td><strong>3.52%</strong></td>
</tr>
</tbody>
</table>

Table 4.1: Hardware resource usage of the proposed system, on Stratix IV. The percentage resource usage in the last line shows the percentage of total resource of the corresponding type used on the device.

<table>
<thead>
<tr>
<th></th>
<th>Hcells</th>
<th>Block RAM bits</th>
<th>DSP block 18-bit elements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>total</strong></td>
<td>49693(0.55%)</td>
<td>1751</td>
<td>36</td>
</tr>
</tbody>
</table>

Table 4.2: Hardware resource usage of the proposed system, on Hardcopy IV. A total of 0.55% of the total HCell resource on device is used.

4.5.1 Evaluation of the Proposed Architecture on Reconfigurable Platforms

Hardware resource usage

The conventional address generator in DRAM interface often acts as a simple counter with minimum implementation cost. Therefore, the hardware resource required to implement it is omitted in this evaluation. Table 4.1 reports the added cost of hardware resources for implementing the prototype system on Stratix IV FPGA. The table shows that a significant proportion of the hardware resources, including all the block RAM bits and the majority of the ALUTS/registers, are used to maintain the intermediate data structure. The array of interp_units also uses a major proportion of ALUTs and registers as they are the most computational intensive parts of the system.

Table 4.2 reports the added cost of hardware resources for implementing the prototype system on Hardcopy IV structured-ASIC.
4.5. Evaluation of the Designed CbIA Architecture

Acquisition time

The estimated maximum frequency of the prototype system is given in Table 4.3.

<table>
<thead>
<tr>
<th>Model ID</th>
<th>f\text{max at slow 900mV 85\textdegree C}</th>
<th>f\text{max at fast 900mV 0\textdegree C}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stratix IV EP4SGX530KH40C2</td>
<td>200 MHz</td>
<td>357 MHz</td>
</tr>
<tr>
<td>Hardcopy IV HC4GX35FF1517</td>
<td>327 MHz</td>
<td>593 MHz</td>
</tr>
</tbody>
</table>

Table 4.3: Reported max frequencies of the design.

The SDRAM access time (in clock cycles) as well as the total image acquisition time spent including interpolation are reported in Figure 4.13. This evaluation test assumes that the prototype system works under a full random accessing situation, i.e. for every address provided by the prototype system, the SDRAM returns only the pixel value of that address. The reference lines in the plots show the image acquisition time required by conventional image accessing process, transformed to equivalent clock cycles.

It can be seen that the achievable PSNR differs with test subject. The image “barbara” has more complex local structures than the other two images and therefore the PSNR of its reconstruction is relatively lower, even when a same priority score threshold is met. In general, due to the reduced number of sampled pixels, the proposed system has a much lower SDRAM occupation time (black lines) than that of the conventional access method (blue reference lines). This results in a much reduced bandwidth requirement of SDRAM and when needed, it frees the SDRAM early on to be accessed by other potential processing units in a large system. On the other hand, a significant amount of time is spent on interpolating the image. Nevertheless, the total image acquisition time is reduced in most cases in this test. In this particular test three interp_units are used, but more of this module can be added to accelerate the process at the expense of more hardware resources. This is because blocks that need interpolation are recorded in FIFO_D (Figure 4.11) and the interpolation task can be completed by multiple interp_units in parallel.

Referring to the discussion in section 3.4, the designed CbIA architecture demonstrates that the proposed framework is capable of reducing the total image acquisition time by employing on-chip computational resources to compensate for the selective and lossy sampling process.
Figure 4.13: Time requirement for sampling process, and complete acquisition process including interpolation. The X axis shows the achieved PSNR given different levels of thr. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively. Reference lines show the time requirement of conventional image accessing method in equivalent clock cycles, assuming the memory data bus is working at x times the prototype system’s frequency.
The time cost overhead of computation, including that for controlling the sampling process and for interpolation, is mitigated by the advanced computation capability of the computing engine as well as by design methods such as parallelism in the interpolation process.

**Energy consumption**

By reducing the number of pixels accessed from the source memory, the proposed CbIA system has the potential to reduce the overall energy consumption of image acquisition process. However as is expected in section 3.4, the proposed CbIA procedure complicates the conventional address generating process and it employs extra computational effort for the reconstruction of the target image. By evaluating the energy consumption of the implemented CbIA architecture on reconfigurable hardware, this section aims to demonstrate the actual impact of the CbIA procedure.

To evaluate the energy consumption, the core dynamic energy consumed by the proposed system is analysed by Quartus PowerPlay analyser, as the cost of executing the CbIA procedure on top of the conventional address generator in SDRAM access interface. The implemented architecture is run on both Stratix IV and Hardcopy IV platforms to demonstrate the energy consumption of the proposed image acquisition procedure, as well as the difference in energy consumption brought by different choices of housing platform. Again, in this test the conventional address generator in SDRAM access interface is assumed to have a zero cost in the energy consumption department as well. Due to the simplicity of the conventional address generator, this assumption is made to make the evaluation more clear. Any energy consumed by the prototype system is considered to be overhead cost on top of the conventional address generator.

Firstly, Figure 4.15 shows the resultant energy consumption of accessing the source memory. All measurements are normalized by the required SDRAM energy consumption of acquiring the target image by conventional accessing method. Therefore the reference line at ratio of “1” shows the total SDRAM energy consumption of the conventional accessing method. It can be seen from this figure that the CbIA procedure is able to reduce the memory energy consumption
by a significant amount. For example, when working on the image “lena”, the memory energy consumption is reduced by 80% when maintaining 33 dB of the image PSNR.

Next the energy consumption of the implemented architecture itself is evaluated. Figure 4.14 shows the breakdown of energy consumption of the sampling and interpolation process spent purely by the proposed system, demonstrating the amount of energy overhead introduced by the proposed CbIA system. It can be seen that among the energy consumed by the proposed system, the reconstruction process (interpolation in this case) is also much more dominating than that of the sampling control mechanism. In the case of Stratix IV implementation (Figure 4.14(a)), the energy consumption overhead of running the proposed CbIA system has a significant presence compared with that of the overall energy consumption of conventional image acquisition process. On the other hand, the energy overhead has a much less impact in the case of Hardcopy IV implementation (Figure 4.14(b)), leading to a much higher potential for the system to reduce the overall energy consumption of the image acquisition process.

Finally, the overall energy consumption of the image acquisition process is reported in Figure 4.16. For the overall energy consumption in this figure, SDRAM energy cost is added to the energy cost of the CbIA system. Again, all measurements are normalized by the required SDRAM energy consumption of acquiring the target image by conventional accessing method. Therefore a ratio lower than 1 indicates that the proposed CbIA system is able to save energy for the image acquisition process.

With the presence of energy overhead introduced by the proposed system implemented on Stratix IV, a reduction of overall energy consumption can still be seen when the threshold is above about 600 if DDR3s are targeted. For general purpose SDRAMs simulated by CACTI, a reduction of energy consumption can be seen across most threshold levels. In the case of “lena”, a reduction of up to 30% can be seen while maintaining a PSNR above 30 dB. On the other hand, the test on Hardcopy IV shows a significant reduction of overall energy consumption across all threshold levels and different memory types. In the case of “lena”, a energy saving of up to 75% can be seen while maintaining a PSNR above 30 dB.

This set of evaluations demonstrates the potential of the proposed CbIA framework in efficiently
Figure 4.14: Breakdown of energy consumption by the proposed system, for sampling process (marked by “s”), and complete process including interpolation (marked by “t”). Reference lines are the energy consumption of accessing the whole target image from SDRAM by conventional method.
Figure 4.15: The ratio of the energy consumption of the source memory (DDR3-667) to that of the memory access by conventional access method. Reference lines at ratio = 1 shows the energy consumption of the conventional access method. It can be seen from this figure that by trading part of the image quality, the CbIA architecture is able to reduce the energy consumption on the memory side by a significant portion.
Figure 4.16: The ratio of total energy consumption of the proposed system (including corresponding energy spent on sampling from DRAM) to that of the memory access by conventional access method. Different DRAM models are used as target memory. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively.
trading image quality for reduced energy consumption of image acquisition process.

4.5.2 Case study on JPEG2000

The proposed CbIA procedure can be applied to various image processing systems to reduce the cost of image acquisition process. The quality of the acquired image is indeed part of the trade-off that enables the CbIA procedure. Therefore it is particularly beneficial to be used in applications that remove image redundancy to some extent, such as in surveillance cameras and image recognition systems.

In this section, the proposed procedure is evaluated under a practical application scenario in order to assess its impact under a real-life problem. The selected application is the JPEG2000 image compression and it is chosen due to its wide usage. The compression unit accesses image macroblocks read either in the conventional access method, or by the proposed CbIA acquisition procedure. The image quality of the compression output using both image acquisition methods are compared with each other (Figure 4.17) to demonstrate the impact of quality reduction resulting from Context-based Image Acquisition process.

Figure 4.17 shows the corresponding quality of compression outputs, using the two image
4.5. Evaluation of the Designed CbIA Architecture

acquisition methods. In this test, the image quality is shown in Mean Squared Error (MSE) instead of PSNR, in order to be able to represent the situation where compression ratio is 1, i.e. no compression is performed. In this situation the output image has 0 MSE but its PSNR is $+\infty$ which cannot be plotted on the graph. From this figure, it can be seen that due to the sampling nature of the CbIA procedure, the reconstruction error during the CbIA process is carried over to the client application which in this case is the image compression unit. This additional error increases the quality loss of the compression output, in addition to the compression distortion.

To further examine the impact on the compression quality, Figure 4.19 shows the differences of MSE between compression output using ground truth image and that using the image acquired by the CbIA procedure. The black curve is the MSE difference when no compression is used and it is in fact the same as in Figure 4.16 but presented in MSE. When the acquired image is processed by the compression unit, it can be seen that the image quality difference keeps decreasing as the compression rate increases. This shows that some loss of image quality due to progressive sampling is absorbed by the process of compression.

In general, if the client tends to remove image redundancy as the proposed system does then the impact of quality loss due to applying the proposed system is reduced, and therefore the system can achieve an even larger gain in bandwidth and image acquisition time/energy. Although the “image redundancy” is difficult to define as a concept, in general it describes the fact that natural image data is not random signal but one with spatial/temporal correlation. This is the same as the basis of the proposed Context-based Image Acquisition concept.

Another example of such redundancy in image processing tasks is the face recognition via sparse representation [WYG+09], in which the author shows that a downsampled version (from 192x168 to 12x10, which contains a significant removal of high resolution details) of a face image is enough of a feature to compute for its sparse representation and be used for face recognition. In such applications, the proposed CbIA framework is particularly beneficial.
Figure 4.18: The quality of compressed image measured in MSE, using both conventional accessing method and the proposed system. DDR3-667 is used as source memory. Data points from left to right represent thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively. Because of the additional quality loss introduced by the CbIA procedure, the error of the compression output using CbIA acquired images is higher than that of the conventional image acquisition method (blue reference lines).
4.5. Evaluation of the Designed CbIA Architecture

Figure 4.19: The quality difference of compressed image, using both conventional accessing method and the proposed system. DDR3-667 is used as source memory. Data points from left to right represents thr of 1800, 1300, 900, 600, 400, 300, 200, and 150 respectively.
4.5.3 Targeting an ASIC Implementation

The evaluation of the CbIA architecture on FPGA and structured-ASIC suggests its potential in saving both bandwidth and time/energy of the image acquisition process. It is desirable to be implemented as ASIC architecture, replacing conventional memory accessing interface, to provide an alternative image accessing methods when bandwidth or acquisition time/energy is of a major concern. According to the work of Kuon [KR07], the ASIC implementation of a same design has an average of 4.6x decrease in path delay, and an average of 14x decrease in dynamic power consumption running the same test vector. A projection of the proposed FPGA implementation to ASIC by these factors sees the proposed design able to reduce both image acquisition time and energy. It will meet the clock frequency of DDR3-800 but for faster models it still requires more interp_unit to accelerate the interpolation process, in order to be capable of reducing total image acquisition time. On the other hand, the energy consumption of the architecture will be reduced greatly, making the proposed architecture promising in saving energy by large margin.

4.6 Performance Under Burst Mode

Modern memory designs allow for pre-fetching (burst mode) capability. By receiving one address, the memory returns a consecutive sequence of data starting at the provided address. When the storage of the content is well optimized for the client application, this functionality can bridge the clock frequency difference between the control bus and data bus of the memory. It also reduces the addressing effort of the memory interface and therefore save energy.

The block mapped image storage (Figure 4.4) is an example of the benefit of utilizing the pre-fetching ability of SDRAMs. Because of the fact that image processing applications often require to access a local region of the target image, the block mapping strategy can minimize the number of row switching activities in SDRAMs where such actions are costly (both in time and energy) compared to the rest of actions. This leads to a reduction of the overall cost of memory accessing.
The discussion provided in above sections assumes for a complete random access situation where the proposed CbIA memory interface has full ability to random access individual pixels from the source memory. It is assumed that the memory works under full random access (no DDR feature) and that the CbIA architecture runs at a frequency equal to or higher than that of the memory data bus. In many situations it is not beneficial to meet the requirement of random accessing, or to ignore the pre-fetching ability of memories. It is likely that due to reasons such as 1) working frequency difference and 2) data bus width higher than required for an individual pixel, the source memory may return a burst of pixels each time it is provided with a sampling address by the propose system.

Even though the proposed context-based image acquisition does not directly benefit from the concept of pre-fetching, burst reading more pixels than addressed for is design-wise compatible to the proposed acquisition procedure and the CbIA architecture can take advantage of it. As is shown in Figure 4.20(a), in ideal situation five individual samples are requested (marked in green) when the 9x9 block is called to be refined by CbIA procedure. In the situation where such random accessing of individual pixels is not viable and memory pre-fetch is effective, a consecutive sequence of pixels are accessed (Figure 4.20(b)).

Figure 4.20: Pre-fetch in the proposed sampling procedure, assuming a block mapping strategy same as in Figure 4.4(b).

In its essence, the idea of Context-based Image Acquisition is to identify and acquire only pixels of most statistical significance, and rely on reconstruction algorithms to build an approximation of the ground truth. Under the basic assumption of signal continuity, the spatially continuous
sampling patterns in Figure 4.20(b) are considered sub-optimal and the potential of the CbIA procedure is not achieved. However, even though the burst read pixel sequence in Figure 4.20(b) may not be considered to be optimal by the sampling procedure, they are nevertheless ground truth information that add to the quality of the final reconstruction result and they reduce the workload of the reconstruction process.

A full evaluation of the proposed system working under memory burst mode is done and results are shown in Figure 4.21 and 4.22. In this evaluation, the proposed CbIA system is not able to access individual pixels but the memory returns a burst of pixels every time it receives a sampling address.

It can be seen from Figure 4.21 that the longer the burst is, the less informative the sampling patterns are, i.e. a same amount of samples leads to a lower reconstruction quality. However, when the overall energy consumption of the image acquisition process is considered it can be seen that the impact of these extra samples is two-fold (Figure 4.22). At lower achievable PSNR levels, burst reading more pixels than addressed for is a compromise that increases overall energy consumption; at high achievable PSNR levels where the system refines gradually smaller regions, the energy spent on running the system outweighs the reconstruction quality improvement it brings, so much so that directly burst reading the rest of pixels within the small region in question actually costs less in energy consumption.

In general, the proposed CbIA system is compatible with memory burst mode design-wise. To a larger extent, the proposed framework of context-based image acquisition is compatible with any memory mechanism that allows for a certain level of random access ability. The samples does not have to be individual pixels but instead small regions on the target image. This also means that the proposed framework is compatible with existing techniques such as image re-compression as is described in section 2.6 where compression happens within local sequences of pixels.
4.7 Conclusion

In this chapter, a design of CbIA architecture is given which is based on the concept of Context-based Image Acquisition framework. The implementation and evaluation of the proposed design is on the reconfigurable hardware platform of FPGA and structured-ASIC. Evaluation results show the potential of the prototype system in reducing the overall bandwidth, time, and energy cost of the image acquisition process, compared with conventional method.

Following the analysis in section 3.4, the designed CbIA architecture demonstrates the behaviour of the proposed CbIA procedure.

1. By reducing the number of times of memory accessing, the overall cost of image acquisition process can be reduced at a cost of lowered image quality in a practical hardware environment.

2. A suitable progressive sampling algorithm/mechanism such as the proposed Adaptive Refine algorithm allows the CbIA architecture to dynamically adjust to the available bandwidth, time, and energy resources. The CbIA architecture always refines the sampling
pattern in a way that each memory access brings in a pixel considered by the sampling procedure to be most statistically significant.

3. Running the CbIA architecture over the conventional memory access interface introduces overhead cost both in time and energy consumption. On the chosen evaluation platform in this thesis, the overhead cost has an impact on the CbIA system’s ability of reducing overall cost of image acquisition process. In some cases the overhead of the architecture even leads to a higher overall energy consumption than conventional method.

In an ideal situation where the overhead of the CbIA architecture is ignorable, more complex sampling and reconstruction algorithms can be employed to optimize the ratio of achievable PSNR vs. memory access cost, leading to a minimized overall cost of the image acquisition process under certain image quality requirement. However in practice, the complexity of the sampling and reconstruction limits the performance gain of the CbIA architecture depending on the actual hardware environment. Nevertheless, with the designed and implemented CbIA architecture in this chapter establishing a solid ground for the research, in the next chapter the discussion is focused on the optimization of sampling procedure under the ideal situation. This is to explore the potential upper-bound of the performance of CbIA procedure.
Chapter 5

Kernel-based Adaptive Image Sampling

5.1 Introduction

In the previous chapter, a hardware architecture of Context-based Image Acquisition is designed and evaluated on reconfigurable hardware. The design is based on the concept of progressive image sampling using stochastic models. The evaluation on reconfigurable hardware platforms shows the potential of the proposed CbIA architecture in reducing the bandwidth requirement, time and energy consumption of the image acquisition process.

In this chapter, the discussion is focused on the trade-off between image quality and the number of times of memory accessing (measured in $\text{b} / \text{p}^1$ in this chapter). This trade-off is essential to the general Context-based Image Acquisition framework and is directly related to the bandwidth requirement to the source memory. As long as the prerequisites discussed in section 3.4 are met\(^2\), this trade-off has a dominant impact on the overall performance of the proposed architecture in reducing image acquisition cost.

This chapter provides an extended discussion on point sampling strategies focusing on the higher quality vs. b/p performance. The Adaptive Refine sampling algorithm used in the proposed CbIA architecture is limited to regular grids for simple data structure management. Complex

---

\(^1\) Average bits per pixel, can also be denoted by “bpp”.

\(^2\) The prerequisites serve as a general guideline and are dependent on the actual situation of application.
Chapter 5. Kernel-based Adaptive Image Sampling

sampling algorithms such as the grid AFPS [DL07] are able to work on irregular grids which brings more freedom to the sampling locations, at the cost of high computational cost.

To further achieve a better quality vs. b/p ratio, the work explained in this chapter proposes a collection of more complex models of a natural image to the sampling procedure. The proposed methods are given a shared name of Kernel-based Adaptive Sampling (KbAS) as they are a series of stochastic progressive sampling methods based on the construction of equivalent kernels. The proposed methods are able to model natural images in a detailed manner and achieve a better quality vs. number of samples ratio by identifying statistically significant pixels at each sampling iteration. However, these models are more computationally intense than the Adaptive Refine algorithm explained in the previous chapter, which will be discussed in section 5.7.

The design and discussion of the KbAS algorithms is to achieve a better image quality vs. b/p ratio which is directly related to the memory bandwidth requirement during image acquisition process. Moreover, it is to enable a bold speculation into how the implementation cost overhead is going to hamper the usefulness of the proposed framework, and to estimate for the situation under which the proposed framework is going to be beneficial in reducing the time and energy consumption of the image acquisition process. The remainder of this chapter is organized as follows: in section 5.2 the point sampling problem is discussed and studied, offering some insight into the design considerations about point sampling algorithms; in section 5.3 a generalized point sampling framework is proposed to guide the design of the actual sampling algorithms; section 5.4 briefly reviews the work of Takeda et al. [TFM07] on kernel regression of image data, which serves as a foundation of the proposed KbAS methods; section 5.5 provides a detailed discussion about applying the concept of kernel construction in the design of KbAS algorithms; section 5.6 provides evaluation results of the proposed algorithm in comparison with previous sampling algorithms; in section 5.7 the cost of implementing the proposed method on hardware systems is analysed; finally in section 5.8 conclusions of the chapter are given.

The main contributions of this chapter include:

1. A generalized framework of stochastic point sampling algorithm is proposed, which is an
extension of previous works in this field. The proposed generalization of point sampling algorithms focuses on the estimation of pixel priorities from the statistical relationship between the candidate pixel and existing sampling pattern.

2. Kernel-based Adaptive Sampling (KbAS) algorithms are proposed following the generalized framework. These algorithms make use of the technique of kernel construction, and are universally applicable to all natural images. They offer a better image quality vs. b/p ratio than state-of-art point sampling algorithms.

5.2 Revisiting the Point Sampling Problem

In the framework of Context-based Image Acquisition, the requested image is accessed pixel by pixel from the source memory. These acquired pixels are used to reconstruct an approximation of the original image for the client application to use. The accessed pixels in this way should be considered by the system to be of most significance in terms of improving the reconstruction quality. In its essence, the CbIA sampling procedure is the same as blind point sampling of image data.

Blind point sampling is the process of identifying and acquiring pixels of significance from a target image without any explicit prior knowledge about it. This kind of sampling process aims to progressively learn about the underlying structure of the ground truth, using only the limited amount of information gathered from already sampled pixels. Although the performance (image quality vs. b/p) of this kind of process is often inferior to those that can pre-process the target image, it does offer the benefit of being independent from the source. In situations where the ground truth is not readily available and the sampling itself is expensive, the advantage of blind point sampling can be very appealing. In the context of accessing image data from source memory in hardware systems, the method of blind point sampling has several merits. Firstly it does not require to pre-process the target image. Pre-processing can be costly to implement on the source memory side as most memory systems have only a minimum amount of computational ability. Therefore point sampling makes the proposed CbIA architecture
universally compatible with most existing memory systems. Secondly, point sampling does not alter the conventional memory accessing protocol. The proposed architecture requests for pixels by providing addresses as it does in the conventional way, and the only change is the patterns of addressing. This also makes the proposed architecture applicable to most existing hardware systems. Moreover, the use of conventional memory accessing protocol does not introduce extra implementation overhead, which leads to a potentially easy implementation.

Given these benefits of point sampling methods, the remainder of this chapter will focus on the discussion of the design towards an ideal image point sampling procedure, without the restriction of the practical capability of existing hardware platforms. Figure 5.1 shows an example image patch that is to be accessed by point sampling. This example is used in the discussion through this chapter.

The design of a point sampling algorithm is highly dependent on the choice of reconstruction method. The system samples pixels only to improve the quality of the final reconstruction. Therefore the sampling procedure serves the need of the reconstruction process. In other words, the design of blind point sampling is about the design of sampling-reconstruction algorithm pair.

Given a set of spatially scattered samples, the most commonly used method of reconstruction is
5.2. Revisiting the Point Sampling Problem

Figure 5.2: The 1D sampling-reconstruction example. The sampling and reconstruction work on a single row of pixels marked in red in the ground truth image patch. Two sampling patterns are provided together with the cubic interpolation results using them.

2D interpolation/regression \[B^+06\]. This chapter therefore discusses the sampling algorithms targeting linear interpolation (on non-uniformly sampled grid) for the purpose of generality. Based on the grand assumption of signal continuity of natural images, interpolation and regression algorithms estimate the value of a missing pixel by computing a weighted sum of neighbouring sampled pixels. Depending on the actual modelling of different functions, this “weighted sum” can have different formats \[B^+06\]. Nevertheless, because of the assumption of signal continuity acting as \textit{a priori}, the reconstruction results from these algorithms tends to be smooth between consecutive samples. As is shown in Figure 5.2, different sampling patterns result in distinct qualities. Both with 11 pixels sampled out of a total of 71 pixels, Sampling Pattern 1 complements the reconstruction method by positioning these samples at places where the ground truth would most likely break the signal continuity assumption. Sampling Pattern 2 on the other hand failed to make the full use of the samples. The reconstruction results show a difference in PSNR as well as in visual quality. This example is a demonstration of how the sampling and reconstruction algorithm interact. The reconstruction method makes up for the missing pixels that the sampling process could not acquire, while the sampling process provides more samples at locations where the reconstruction is not accurate enough.

With the understanding of linear interpolation as reconstruction method, it is obvious that its
underlying assumption of signal continuity breaks at locations with high frequency component, i.e. sudden pixel value changes within a relatively small region. Therefore to complement for the linear interpolation that is used as reconstruction method in this chapter, the following considerations/goals are set to guide the strategy of sampling:

**Design Considerations:**

1. Essentially, the task is to estimate for the underlying structure of the ground truth image as accurate as possible, i.e. correctly identifying locations where the assumption of signal continuity fails. Given that pixels in natural images are not random and the collection of pixels carry meanings as a whole, it is possible to estimate the existence of regions of high frequency components. Focusing the sampling power within these regions is likely to improve the reconstruction quality the most. (For example, in the local region around point #20 in Figure 5.2)

2. Although regions with high frequency components are important, those flat regions are not to be ignored. Firstly, the seemingly “flat” region can be part of a textured area that are yet to be identified as with high frequency components due to the lack of samples. Secondly, when the sample around complex spatial structures such as edges are dense, a new sample will only refine the reconstruction of a relatively small amount of unsampled pixels. On the other hand, a new sample in “flat” regions where sampling is coarse might actually result in a larger improvement to the overall quality of the image because it impacts the reconstruction of more unsampled pixels³.

A balance between the two considerations should be made in order to achieve a good performance. However, it is non-trivial to find an optimal balance by analytic method. The strategies discussed in the rest of the chapter are therefore all numerical and progressive, relying on the refining of statistics to approximate using sampled pixels.

³From another perspective of view, although an unsampled pixel in high frequency regions might have a larger potential variance of its value distribution than that in flat regions, the reconstruction algorithm might also have more gathered information in estimating its value because the sampling is already dense in this region.
Both the estimation of underlying image structure and the balancing between sampling considerations pose challenges. A good sampling-reconstruction algorithm pair leads to a better trade-off between image quality and memory accessing effort, which is measured by b/p in this chapter. Therefore this chapter is dedicated to the discussion and design towards an ideal point sampling procedure for interpolation/regression methods.

5.3 Generalized Framework of Stochastic Point Sampling

The Farthest Point Strategy (FPS) by Eldar et al. [ELPZ97] is a well established method for blind point sampling of image data. The principle behind it is that in order to achieve a better reconstruction quality, pixels that are geometrically farthest from existing sampling pattern should be sampled with highest priority. This is because the interpolation of the pixel values at these locations are considered to be most inaccurate. Given that within a requested image patch $I$, the collections of already sampled pixels and unsampled pixels are $P_g$ and $Q_g$ respectively, then the “farthest point” $x_i$ which has the highest priority of sampling is:

$$x_i = \max_{x_i \in Q_g} (\min_{x_j \in P_g} (dist(x_i, x_j)))$$ (5.1)

where $dist(x_i, x_j)$ is the Euclidean distance between the coordinates of pixel $x_i$ and $x_j$. On top of the definition of the farthest point sampling priority, randomness is introduced in order to minimize the effect of aliasing. This randomness is ensured by a random initial sampling pattern from which the system starts the actual FPS sampling procedure. The actual implementation of FPS strategy involves iteratively computing and updating Voronoi vertices, using already sampled pixels (Figure 5.3). The original FPS method is non-adaptive to the actual data it is used to sample. The resulting sampling pattern is a random but uniformly distributed one in the end. It can be applied to all types of signals and serves as a starting point of later point sampling strategies.

The adaptive version of FPS, the AFPS, is also introduced by Eldar [ELPZ97] as a complement to the non-adaptive FPS to work on natural images. A weighted distance function is introduced
with the weights computed from local statistics gathered from previously sampled pixel values. Eldar et al. provided an example of possible weight formulation based on the estimation of local bandwidth:

\[
p(x_i) = \min_{x_j \in P}(dist(x_i, x_j)) \times \max_{x_j, x_k \in P}(B_{\min}(x_j, x_k))
\]  
(5.2)

where \(p(x_i)\) is the estimated priority score of the candidate unsampled pixel \(x_i\), and \(P\) is the collection of sampled pixels in the local neighbourhood of \(x_i\). The estimated bandwidth \(B_{\min}\) is:

\[
B_{\min}(x_j, x_k) = \frac{\arcsin(\frac{2I(x_j) - M}{M}) - \arcsin(\frac{2I(x_k) - M}{M})}{2\pi \times dist(x_j, x_k)}
\]  
(5.3)

where \(M\) is the bound on the amplitude of the pixel and \(I(x_i)\) is the pixel value of \(x_i\). An example of the generated sampling pattern is shown in Figure 5.4.

Later in the work of Devir et al. [DL07], the concept of AFPS is applied to range sampling and is extended to work on regular grid, called Adaptive Grid Algorithm. The priority score is also given a series of new possible forms, based on the estimation of local pixel variance:

\[
p(x_i) = \min_{x_j \in P}(dist(x_i, x_j)) \times log(1 + \hat{\sigma}^2)
\]  
(5.4)

where the variance \(\hat{\sigma}^2\) of the candidate pixel is estimated by the weight variance of sampled pixels in the local neighbourhood. Several weighting schemes are discussed in the work of Devir [DL07] and empirical conclusions are given based on the given testing dataset.
The AFPS-based sampling framework establishes a solid foundation for later applications of point sampling algorithms. Based on this, in this project it is proposed to extend the framework of AFPS to a generalized format:

$$p(x_i) = var(x_i) \otimes dist(x_i, P)$$ (5.5)

In this generalized format, the priority score $p(x_i)$ of a candidate unsampled pixel $x_i$ is computed by a combination of two terms:

1. The variance term $var(x_i)$ measures the potential bound of the amplitude of an unsampled pixel $x_i$. This variance is estimated from the sampled pixel values in the local neighbourhood to reflect the image structure complexity of the local region. From an alternative point of view, it can also be constructed to measure the reduction of the collective neighbourhood variance when the candidate pixel is sampled, which also reflects the image structure complexity of the local region of $x_i$.

2. The distance term $dist(x_i, P)$ measures how related an unsampled pixel $x_i$ is to the current
sampling pattern $P$, given the chosen reconstruction method. This term therefore includes but is not limited to Euclidean distance of pixel coordinates. The construction of this distance term requires to find a statistical relationship or dependency between unsampled pixels and existing samples.

3. The operator $\otimes$ that combines the two terms can have different forms. It can be a product $(a \ast b^n)$, weighted sum ($w_1a + w_2b$), or in other forms depending on the situation. The design of this operator is to weight the impact of the two terms.

Referring to the two Design Considerations in section 5.2, it can be seen that the variance term reflects the first consideration which is to measure the spatial complexity of the underlying image data; the distance term on the other hand, is a mixed embodiment of both considerations. The design of the distance term as well as the way of combining the two terms also involves balancing between the two considerations.

The aim of this generalized framework of priority score estimation is to provide a systematic way of designing the actual point sampling strategy for a given application. As is stated in the original work of Eldar et al. [ELPZ97], choosing an appropriate estimation function may depend on the specific application, since it reflects some a priori knowledge about the target image. This generalized framework of priority score encourages the designer to think from the perspective of constructing both terms using already sampled pixels, and to find a balanced way of combining the two terms. Note that in the original AFPS or the later developed grid AFPS, the distance between candidate $x_i$ and the existing sampling pattern is computed purely by Euclidean distance of coordinates and therefore is non-adaptive to the actual data. However in the proposed framework, either or both terms can be constructed based on the sampled data and hence be adaptive to the actual data.

5.4 Review: Kernel Regression on Image Data

As is explained in previous sections, the key to the designing of point sampling algorithms is to estimate for the underlying image structure using the limited amount of sampled information.
5.4. Review: Kernel Regression on Image Data

It is to identify regions with potentially high frequency components, as well as to establish the statistical relationship between sampled and unsampled pixels. This gathered knowledge is then used in the computation of the variance and distance terms which lead to the estimation of priority scores of candidate unsampled pixels.

The main idea behind the KbAS strategy proposed in this chapter, is indeed to use the kernel-based method to extract information from existing samples. In the work of Takeda et al. [TFM07], they provide a detailed and complete discussion on the use of kernel-based method in image regression tasks. The derivation of their kernel-based regression algorithms is rigorous and systematic. The derived framework of kernel regression turns out to be able to contain some popular regression techniques, such as the Nadaraya-Watson estimator (NWE) and the Bilateral Filter.

Inspired by the use of kernel-based method for image regression, this work applies such methods to extract information from the samples to guide the future sampling process instead of just for regression. The intuition behind is the fact that the point sampling algorithm to be designed is ultimately used to serve the reconstruction of the image. If the kernel-based methods are able to estimate and describe the statistics of pixels for reconstruction purpose, then this information might as well be useful to the sampling algorithm.

In this section, the framework of kernel regression proposed by Takeda et al. [TFM07] is briefly reviewed, which serves as a foundation on which the KbAS method is explained in next section.

**Kernel Regression**

In general, kernel regression explained in the work of Takeda [TFM07] is a non-parametric method relying on the actual data to determine the structure of the model. The construction of kernels provides a way to build for the implicit model, or regression function. In the case of 1D signal, if the measured data can be represented in the following form:

\[ y_i = z(x_i) + \varepsilon_i, \ x_i \in P \]  \hspace{1cm} (5.6)
where \( z(x_i) \) is the regression function and \( \varepsilon_i \) is independent and identically distributed zeros mean noise values. Again, \( P \) is the collection of sampled points. Although the regression function \( z(x_i) \) is unknown, with the assumption of local continuity of natural signal, a local expansion can be made at each sampled point \( x_i \). If the signal is locally smooth to an order of \( N \), then we can have the \( N \)-term Taylor series at point \( x_i \) with \( x \) being a point close to it:

\[
\begin{align*}
z(x_i) &\approx z(x) + z'(x)(x_i - x) + \frac{1}{2!}z''(x)(x_i - x)^2 + \ldots + \frac{1}{N!}z^{(N)}(x)(x_i - x)^N \\
&= \beta_0 + \beta_1(x_i - x) + \beta_2(x_i - x)^2 + \ldots + \beta_N(x_i - x)^N
\end{align*}
\]  

(5.7)

In this way, a relationship is established between an unsampled point \( x \) and a sampled point \( x_i \) which are close to each other. And to regress for the value of \( z(x) \), the problem becomes to estimate for the set of parameters \( \{\beta_n\}_{n=0}^N \) from which \( z(x) = \beta_0 \). It is natural that a set of local samples close to point \( x \) can be used to estimate for \( \{\beta_n\}_{n=0}^N \) which is indeed derived from local expansions. Therefore given a set of sampled points \( x_i \in P \) in the local neighbourhood of \( x \), to estimate for its value \( z(x) \) is to solve the following optimization problem:

\[
\min_{\{\beta_n\}} \sum_{x_i \in P} [y_i - \beta_0 - \beta_1(x_i - x) - \beta_2(x_i - x)^2 - \ldots - \beta_N(x_i - x)^N]^2 \frac{1}{h^N} K_h(x_i - x) 
\]  

(5.8)

Referring to the work of Takeda [TFM07], the above steps can be extended to multi-dimensional data. If the data measurement model of Eq.5.6 is changed to 2D image data with pixel coordinates represented by 2 dimensional vector \( \mathbf{x} \):

\[
y_i = z(x_i) + \varepsilon_i, \quad x_i \in P
\]  

(5.9)

then based on the assumption of local continuity of the signal, to estimate for the value at point \( \mathbf{x} \) is to solve the following optimization problem:

\[
\min_{\{\beta_n\}} \sum_{x_i \in P} [y_i - \beta_0 - \beta_1(x_i - x) - \beta_2(x_i - x)^2 - \ldots - \beta_N(x_i - x)^N]^2 K_{H}(x_i - \mathbf{x}) 
\]  

(5.10)
in which:
\[
K_H(t) = \frac{1}{\det(H)} K(H^{-1}t)
\]

and \( H \) is a smoothing matrix:
\[
H = \begin{bmatrix}
h & 0 \\
0 & h
\end{bmatrix}
\]

To reflect the embedded image structure in the local neighbourhood, the smoothing matrix \( H \) is modified into a full matrix \( H_{\text{steering}}^{x_i} \):
\[
H_{\text{steering}}^{x_i} = h_i C_{x_i}^{-\frac{1}{2}}
\]

where \( h_i \) is the smoothing parameter at \( x_i \) and \( C_i \) is a symmetrical covariance matrix reflecting local gradient information of pixel values centred at \( x_i \). In detail, the covariance matrix \( C_x \) at a point \( x_i \) is computed by:
\[
C_{x_i} = \gamma_{x_i} U_{\theta_{x_i}} \Lambda_{x_i} U_{\theta_{x_i}}^T
\]

\[
U_{\theta_{x_i}} = \begin{bmatrix} 
\cos \theta_{x_i} & \sin \theta_{x_i} \\
-\sin \theta_{x_i} & \cos \theta_{x_i}
\end{bmatrix}
\]

\[
\Lambda_{x_i} = \begin{bmatrix} 
\sigma_{x_i} & 0 \\
0 & \sigma_{x_i}^{-1}
\end{bmatrix}
\]

The parameter set \((\sigma_{x_i}, \theta_{x_i}, \gamma_{x_i})\) is computed from singular value decomposition of the matrix of local gradients. If \( z_{x_1}(\cdot) \) and \( z_{x_2}(\cdot) \) are first derivatives of the grayscale value along \( x_1 \) and \( x_2 \) directions respectively, then the decomposition of the gradient matrix in the neighbourhood is:
\[
\begin{bmatrix}
\vdots & \vdots \\
z_{x_1}(x_j) & z_{x_2}(x_j) \\
\vdots & \vdots
\end{bmatrix} = U_x S_x V_{x_i}^T, \quad x_j \in P \cup Q
\]

where \( P_{x_i} \) and \( Q_{x_i} \) are the collections of sampled and unsampled pixels in the local neighbour-
Figure 5.5: Effects of applying the steering matrix $C_{x_i} = \gamma_{x_i} U_{\theta_{x_i}} \Lambda_{x_i} U^T_{\theta_{x_i}}$; the shape of the kernel is changed to reflect the local image structure. [TFM07]

The rotation angle $\theta_{x_i}$ is computed from $v_2 = \begin{bmatrix} \nu_1, \nu_2 \end{bmatrix}$, the second column of the orthogonal matrix $V_{x_i}$.

$$\theta_{x_i} = \arctan \left( \frac{\nu_1}{\nu_2} \right)$$ (5.18)

The elongation parameter is the ratio of the energy in the two dominant directions, indicated by the two diagonal elements of $S_{x_i}$: $\sigma_{x_i} = s_1/s_2$. The scaling parameter $\gamma_{x_i}$ is determined by the geometric mean of the energy normalized by the number of pixels $M$ in the neighbourhood:

$$\gamma_{x_i} = \sqrt{s_1 s_2 / M}.$$ The impact of applying the rotation, elongation, and scaling parameters is shown in Figure 5.5.

The end result of building the steering kernel is the data adaptive version of regression function:

$$\hat{\tilde{z}}(x) = \sum_{x_i \in P} W_{\text{steer}}(x) y_i$$ (5.19)

This estimator (Eq.5.19) remains a weighted average of sampled pixels within a local neighbourhood of pixel $x$ which is the pixel to reconstruct. The weights are now determined not only by Euclidean distance between pixel pairs, but also the local image structure estimated.

---

4There are a few minor difference between the parameter formulation in this thesis than those in the original work [TFM07]. This is for reducing the computational effort and it does not have a significant impact to the performance of the design.
from values of already sampled pixels. Such weights describe the relationship between pixel pairs and Eq. 5.19 is the core of the framework of image kernel regression.

5.5 Kernel-based Adaptive Sampling

In this section, the Kernel-based Adaptive Sampling (KbAS) is introduced and explained. It is based on the generalized formulation of point sampling problem (Eq. 5.5, section 5.3), and it makes use of kernel construction methods similar to that used in kernel regression algorithms (section 5.4).

5.5.1 Describe Pair-wise Relationship Using Kernels

In this section, the application of kernel construction to describe pixel pair-wise relationship in image sampling problems is explained. It can be seen that the application of kernels in the kernel regression algorithm is based solely on the assumption of local signal continuity (Eq. 5.7). This is the reason that the framework of kernel regression is able to contain some other popular interpolation/regression techniques. The use of kernels in the proposed KbAS algorithms also aims to stay true to this very basic and universal prior knowledge.

Notice although the original kernel regression algorithms are derived at sampled pixels \( \{x_i | x_i \in P_g\} \), the same process can be applied to unsampled pixels as well. In other words, the method of equivalent kernel construction:

1. describes the relationship between any pixel pair. It can be between an unsampled pixel and a sampled pixel as is the case in kernel regression. It can also be between two unsampled pixels or two sampled pixels. (Observation 1)

2. is applicable to local regions of image data that contains some sampling pattern dense enough (explained later in this section) to provide a satisfying amount of information about underlying image structure. (Observation 2)
Following Observation 1, a general form of relationship description can be derived, based on the derivation of kernel construction introduced in kernel regression literature. Regarding a sampled/unsampled pixel \( x \), Taylor expansion can be done at all pixel locations \( \{x_n | x_n \in P \cup Q\} \) in its local neighbourhood. Here \( P \) and \( Q \) are defined as before, being the collection of sampled pixels and unsampled pixels in \( x \)’s neighbourhood respectively. With Taylor expansion performed in the neighbourhood, the estimated pixel value of \( x \) can be computed from the following objective function:

\[
\min_{\{\beta_n\}} \sum_{x_i \in P \cup Q} \left[ y_i - \beta_0 - \beta_1(x_i - x) - \beta_2(x_i - x)^2 - \cdots - \beta_N(x_i - x)^N \right]^2 K_H(x_i - x) \quad (5.20)
\]

Following the similar steering kernel construction introduced in Takeda’s work [TFM07], to reflect the local gradient information the smoothing matrix is modified to be a steering smoothing matrix \( H_{x_i}^{steering} \):

\[
H_{x_i}^{steering} = h_i C_{x_i}^{-\frac{1}{2}} \quad (5.21)
\]

The accurate formulation of \( C_{x_i} \) relies on the accurate estimation of underlying gradient information. The gradient information can be extracted via various filters based on a rough reconstruction of the image using existing samples. The accuracy requirement poses a challenge to applying the idea of kernel construction to the image sampling problem, where the initial sampling pattern can be coarse. An example is shown in Figure 5.6 where the target image is sampled by two different sampling patterns. Gradient information is extracted by applying Sobel filters along horizontal and vertical directions. The gradients along the two directions are used in Eq.5.17 to compute the steering parameters.

It is obvious that a dense sampling pattern (pattern 2 in Figure 5.6) results in a more accurate estimation of the ground truth gradients, hence the observation 2 above. However it is worth noting that, while the construction of kernels relies on a “dense enough” sampling pattern in the local area, this is always an approximation. It is especially true at the beginning of the point sampling procedure when the sampling pattern is coarse (pattern 1).
With the steering matrix \( C_x \), computed and taking order \( N = 0 \), the end result is that the pixel value of \( x \) can be described as a weighted sum of the values of all pixels in its neighbourhood as in Eq.5.22:

\[
\tilde{z}(x) = \sum_{x_i \in P \cup Q} W_{x_i}(x)y_i \\
= \sum_{x_i \in P \cup Q} \frac{K_{H_{steer}}(x_i - x)}{\sum_{x_j \in P \cup Q} K_{H_{steer}}(x_j - x)} \cdot y_i 
\]

(5.22)

If Gaussian kernel is used as base kernel, then the steering kernel function is:

\[
K_{H_{steer}}(x_i - x) = \sqrt{\text{det}(C_{x_i}) \frac{2\pi h_i^2}{h_i^2}} \exp \left\{ -\frac{(x_i - x)^T C_{x_i} (x_i - x)}{2h_i^2} \right\} 
\]

(5.23)

The equivalent kernel computed using Eq.5.22 at different locations in the image reflects its local structures. Some examples are given in Figure 5.7. It can be seen that if the pixel \( x \) in question is surrounded by complex spatial structures such as edges, the majority of the weights in its neighbourhood are on few pixels. If \( x \) is on an edge, the significant weights in its neighbourhood can be found along the edge structure. These show that the construction of equivalent kernel is indeed to identify the optimal size and shape of the neighbourhood of \( x \) which is still subject to the assumption of signal continuity. On the other hand if \( x \) is on flat surfaces, the weights are spread out to more neighbouring pixels and there is no stand-out high weights. It shows an averaging effect and a general agreement to signal continuity in these areas.

To summarize, in the description Eq.5.22 of the pixel value of \( x \), the weight \( W_{x_i}(x) \) (equivalent kernel value) is computed using the gradient information centred on the neighbouring pixel \( x_i \) and is normalized\(^5\) in the neighbourhood of \( x \). Again, the final descriptor Eq.5.22 is applicable to any pixel \( x \) and its local neighbourhood. Although for KbAS the estimation of pixel values is not the task, the descriptor Eq.5.22 establishes a pair-wise relationship between any neighbouring pixel pair \((x, x_i)\) in the form of equivalent kernel value \( W_{x_i}(x) \). Note that while the derivation

\(^5\)Despite of the actual order \( N \) used in the kernel construction, the equivalent kernels all have a similar normalization effect in the neighbourhood.
Figure 5.6: An example of gradient information computed from intermediate reconstructions of the image, using sampled pixels. Sobel filters are applied along horizontal ($G_x$) and vertical ($G_y$) directions, and the gradient magnitude is computed as $\sqrt{G_x^2 + G_y^2}$. 
Figure 5.7: Equivalent kernels (Eq. 5.22) applied at different locations in the image lena. The image is sampled uniformly at the sampling distance of 2.
of the equivalent kernel \( W \) is based on the assumption of Taylor expansion in an infinitely small neighbourhood, in practice it is not possible to achieve and therefore \( W_{x_{i}}(x) \neq W_{x}(x_{i}) \). However approximation is made here to assume that in the small neighbourhood that contains both \( x \) and \( x_{i} \) the Taylor expansion holds and \( W_{x_{i}}(x) \approx W_{x}(x_{i}) \), either being valid to be used to describe the pair-wise relationship between the two pixels in question.

Such description of pair-wise relationships between pixels is non-parametric and data adaptive. It is accurate in the sense that it provides a detailed analysis of the image structure using existing samples in the local neighbourhood. Moreover, as emphasised at the beginning of this section, this description using kernels is universally applicable to natural images because the only \textit{a priori} behind is signal continuity. Therefore the proposed KbAS algorithms are centred around the use of such kernel information in the form of \( W_{x_{i}}(x) \) to formulate for the point sampling problem.

### 5.5.2 Basic Formulation of KbAS Algorithm

With the relationship between pixels described in the form of equivalent kernel \( W_{x_{i}}(x) \), in this section we discuss the formulation of KbAS point sampling problems.

Referring to the generalized form of point sampling problem introduced earlier (Eq.5.5), if we consider every unsampled pixel of equal \textit{variance}\(^6\) and same \textit{distance} from existing sampling pattern then they will all have the same priority score as shown in Figure 5.8. In this figure the example image patch is sampled by a uniform pattern, marked as red dots, and the priority scores of the unsampled pixels are all the same. This example in Figure 5.8 is the “canvas” that KbAS algorithms works on. The aim of KbAS algorithm is to compute the priority score for each unsampled pixel and form an informative priority map which shows the system where to sample in the next iteration.

The formulation of the priority scores – determining \( \text{var} \) and \( \text{dist} \) – using the information provided in the form of equivalent kernels, is to reflect the two \textbf{Design Considerations} explained

---

\(^6\)The term “variance” and “distance” here refer to the variance term and distance term described in Eq.5.5
Figure 5.8: Example image patch and priority scores of pixels shown in grayscale, given that all pixels have the same var and dist. Red dots in (b) are locations of already sampled pixels.

above. Based on this, the following strategy is proposed:

**Strategy 1**: Pixels on whom the reconstruction is considered to be less accurate should be sampled with high priority.

The distance term dist directly measures how determined an unsampled pixel is, given the reconstruction algorithm and surrounding samples. Notice that in the neighbourhood of an unsampled pixel $x$ as shown in Figure 5.9, if we have five already sampled pixels $\{(x_i|i = 1, 2, ..., 5)\}$ then the relationship between $x$ and these samples can be written as:

$$
\hat{z}(x) = W_{x_1}(x)y_1 + W_{x_2}(x)y_2 + W_{x_3}(x)y_3 + W_{x_4}(x)y_4 + W_{x_5}(x)y_5
$$

$$
= \frac{1}{C}(0.043 \cdot y_1 + 0.003 \cdot y_2 + 0.002 \cdot y_3 + 0.0008 \cdot y_4 + 0.035 \cdot y_5)
$$

(5.24)

where $C$ is a normalization parameter. According to this example, the estimated pixel value of $x$ is closely related to the value of $x_1$ and $x_5$ and is only loosely related to the rest three samples. This suggests that $x_1$ and $x_5$ dominate the estimation of the value of pixel $x$. Therefore even without the other three samples, the estimation will stay roughly the same. Moreover, between the two dominant pixels $x_1$ and $x_5$ the estimation is of an averaging effect: both pixels are considered to be of the same likelihood to determine the value of $x$ and therefore an average is computed and used as estimation.

To utilize this described relationship, it is defined in the framework of KbAS that the highest
Figure 5.9: The weights, computed as equivalent kernel values, describe the relationships between pixel pairs.

weight among weights of the sampled pixels in the neighborhood measures how related the pixel \( x \) is with current sampling pattern. On top of that, determining a pixel by itself always returns the ground truth. Hence the likelihood of determining \( x \) with its neighborhood is normalized by the equivalent kernel value on \( x \) itself:

\[
l(x) = \max_{x_n \in \mathcal{P}} \frac{W_{x_n}(x)}{W_x(x)}
\]  

(5.25)

With this likelihood being higher, the pixel \( x \) is closer to the existing sampling pattern and therefore would have a lower distance term:

\[
dist(x) \propto \frac{1}{l(x)}
\]  

(5.26)

In this thesis, the dynamic range of the grayscale value of each pixel is between 0 to 255. Without adding in extra \textit{a priori}, the grayscale value of each unsampled pixel can be assumed to have a same distribution over \([0, 255]\) and thus having the same variance term:

\[
var(x) = \sigma_0^2
\]  

(5.27)
5.5. Kernel-based Adaptive Sampling

Figure 5.10: Example image patch and priority scores of pixels shown in grayscale, computed as in Eq.5.28. Red dots in (b) are locations of already sampled pixels. This graph shows that even with a coarse sampling pattern, the priority estimation in Eq.5.28 is able to roughly identify regions containing high frequency component.

Combining $\text{dist}(x)$ and $\text{var}(x)$, we can have an example of the formulation of priority score:

$$p(x) = \sigma_0^2 \ast \left(1 - \frac{\max_{x_n \in P} W_{x_n}(x)}{W_x(x)}\right)$$

(5.28)

In this case since the variance term is a constant, the distance term can be of other formats as long as it reflects Eq.5.26. For example, an alternative could be:

$$p(x) = \sigma_0^2 \ast \log(1 + \left(\frac{\max_{x_n \in P} W_{x_n}(x)}{W_x(x)}\right)^{-1})$$

(5.29)

When applied to the example image patch, the uniform priority map in Figure 5.8 turns to be informative and able to identify pixels to be sampled in the next iteration (Figure 5.10 and 5.11). It can be seen that the priority map now shows high priorities roughly around edged areas in the original image patch. Even with a coarse sampling pattern to provide gradient information, the priority map is able to resemble the underlying structure. While identifying these complex structures in the image, the algorithm also puts high priority to pixels that are far away from existing samples.

By finding local maximums in the priority map, pixels with highest priority scores can be

---

7In this case where the variance term is a constant, Eq.5.28 and Eq.5.29 produces the same sampling result, despite the priority maps having different visuals.
Chapter 5. Kernel-based Adaptive Image Sampling

Figure 5.11: Example image patch and priority scores of pixels shown in grayscale, computed as in Eq. 5.29. Red dots in (b) are locations of already sampled pixels. Similar to that in Eq. 5.28, the alternative formulation of priority estimation is able to roughly identify regions containing high frequency component.

identified. With another iteration of sampling of these high priority candidates, the priority map is to be updated as well as the new samples bring in more accurate gradient information and they themselves change the distance between other unsampled pixels and existing sampling pattern. Examples are shown in Figure 5.12. It can be seen that after a number of iterations of sampling, the relative priority of pixels in “flat” regions becomes higher and these pixels are also sampled. These reflects the ability of the algorithm in balancing between the two Design Considerations.

By iteratively sampling pixels and updating the priority map, the algorithm refines the estimation of underlying image structure via kernel construction. The full evaluation of the performance is given in section 5.6.

It is worth noting that with the formulation of problem in Eq. 5.28 or Eq. 5.29, the design flow can also be reversed. Since the objective is focused on finding the “distance” between an unsampled pixel $x$ and its surrounding samples, this distance term can be estimated from the equivalent kernels centred on the sampled pixels instead. An equivalent kernel can also be constructed on $x_n$, one of the sampled pixels in the neighbourhood of $x$. In this equivalent kernel, the value of $W_x(x_n)$ describes the relationship between $x_n$ and $x$, but it is computed using gradient information centred on $x$. This can be interpreted as the sample $x_n$ projecting its influence to its neighbouring pixels, to stabilize their reconstruction results. With all neighbouring samples
Figure 5.12: Updated priority scores of pixels shown in grayscale, computed as in Eq. 5.29 with more samples retrieved. Red dots in (b) are locations of already sampled pixels. By sampling pixels from high priority regions and updating the priority map accordingly, the sampling procedure iteratively acquires pixels of high estimated significance to the reconstruction process. The sampling is balanced between the two Design Considerations with samples taken from both “flat” regions and regions of high frequency component.

of \( x \) projecting their influence, the priority score of \( x \) can be:

\[
p(x) = \sigma_0^2 \cdot \min_{x_n \in P} \left( 1 - \frac{W_x(x_n)}{W_{x_n}(x_n)} \right)
\] (5.30)

An example of using this reverse KbAS algorithm is shown in Figure 5.13. In the ideal situation where the size of the neighbourhood is small enough, the weights \( W_x(x_n) \) is the same as \( W_{x_n}(x) \) since the gradient information centred on \( x \) and \( x_n \) is considered to be the same. Therefore Eq. 5.30 is equivalent to Eq. 5.28. The full evaluation of this reverse KbAS algorithm is also given in section 5.6.

5.5.3 The Addition of the Variance Term

While the distance term measures how determined a pixel is given the existing samples, the variance term measures the potential range of its magnitude. In previous discussions, the variance term of each candidate unsampled pixel is assumed to be a constant \( \sigma_0 \). It can also be estimated by the weighted variance of the collection of its neighbouring samples:

\[
var(x) = \frac{N}{N - 1} \cdot \frac{\sum_{x_n \in P} W_{x_n}(x) \cdot (I(x_n) - \mu)^2}{\sum_{x_n \in P} W_{x_n}(x)}
\] (5.31)
Chapter 5. Kernel-based Adaptive Image Sampling

Figure 5.13: Example image patch and priority scores of pixels shown in grayscale, computed as in Eq.5.30. Red dots in (b) are locations of already sampled pixels.

Figure 5.14: Examples of variance terms computed as in Eq.5.31, shown in grayscale. The variance estimation gives a similar result to the distance term estimation.

where \( N \) is the number of samples in the neighbourhood of \( x \). Examples of the computed weighted variance are shown in Figure 5.14.

Combining the variance term in Eq.5.31 with the distance term proposed in the previous section is to balance the contribution of the two terms to the priority score. Some of the possible combinations are:

\[
p(x) = \begin{cases} 
var(x) \cdot (1 - l(x))^n \\
var(x) \cdot \log(1 + \frac{1}{l(x)^n}) \\
var(x) \cdot \frac{1}{l(x)^n}
\end{cases} \quad (5.32)
\]

The format of the distance term as well as the operator that combines the two terms are application dependent. A discussion about this combination is given in section 5.6.
5.5. Kernel-based Adaptive Sampling

Figure 5.15: Examples of priority map computed by Eq.5.33 which is a combination of data adaptive variance term and distance term, displayed as \( \log(1 + p(x)) \) for visual quality.

Examples of the priority map are shown, computed using the following equation:

\[
p(x) = \text{var}(x) \ast \frac{1}{l(x)}
\]  

\[(5.33)\]

5.5.4 Variance Term by Kernel Regressor

In my previous work [LBC14b], an example solution of KbAS sampling algorithm is proposed. In this section, it is explained in the context of the general KbAS algorithm framework.

While the priority scores designed above all follow **Strategy 1**, the problem can be treated in an alternative approach:

**Strategy 2**: Those pixels – which when sampled can bring the most improvement to the collective stability of its neighbouring unsampled pixels during reconstruction process – should be sampled with high priority.

For a candidate unsampled pixel \( x \), if we look at another unsampled pixel \( x_i \) in its neighbourhood then it can be described as in Eq.5.22

\[
\hat{z}(x_i) = \sum_{x_j \in P \cup Q} W_{x_j}(x_i) y_j
\]  

\[(5.34)\]
With \( x_i \) in the neighbourhood of \( x \), \( x \) is in the neighbourhood of \( x_i \) as well:

\[
\hat{z}(x_i) = W_x(x_i)y_x + \sum_{x_j \in P \cup Q, x_j \neq x} W_{x_j}(x_i)y_j
\]  

(5.35)

If it is assumed that the candidate unsampled pixel \( x \) in question is independent with the rest of pixels, then sampling \( x \) in the next iteration will eliminate the distribution variance of the value \( y_x \) which in turn reduces the distribution variance of the estimation \( \hat{z}(x_i) \). If we approximately compute the change of variance of \( \hat{z}(x_i) \) in the following way (assuming every candidate \( x \) has a same constant variance \( \sigma_0^2 \)):

\[
\Delta(\sigma_{x_i}^2)| \text{with } (x) \text{ sampled} = W_x(x_i)\sigma_0^2
\]  

(5.36)

then sampling the candidate pixel \( x \) is going to bring a collective reduction of reconstruction variance of:

\[
\sum_{x_i \in Q} \Delta(\sigma_{x_i}^2)| \text{with } (x) \text{ sampled} = \sum_{x_i \in Q} W_x(x_i)\sigma_0^2
\]  

(5.37)

which is the variance term \( \text{var}(x) \). However, each candidate pixel is determined to some degree by its neighbouring samples, therefore:

\[
\sigma_x^2 = \sigma_0^2 \ast \text{dist}(x)
\]  

(5.38)

where \( \text{dist}(x) \) is as defined in Eq.5.26. As is the case for previous discussions, there are various forms the distance term can take. One valid form is as in Eq.5.28:

\[
p(x) = \text{var}(x) \ast \text{dist}(x) = (\sum_{x_i \in Q} W_x(x_i)) \ast \sigma_0^2(1 - l(x))
\]  

(5.39)

Applying this priority score estimation to the sampling pattern has a weighted averaging effect over the priority map produced by Eq.5.28 (Figure 5.16).
In my previous work [LBC14b], the distance term takes logarithmic form:

\[ p(x) = \text{var}(x) \times \text{dist}(x) = \left( \sum_{x_i \in Q} W_x(x_i) \right) \times \sigma_0^2 \log(1 + \frac{1}{l(x)}) \] (5.40)

5.5.5 Summary of KbAS Algorithm Design

In this section, KbAS algorithm designs are discussed based on the general form of point sampling algorithm (Eq.5.5) and the use of gradient information extracted by equivalent kernels. Although the discussed methods have different formulations of priority score, they all reflect the Design Considerations by constructing the variance and distance terms. In the following section, these proposed KbAS algorithms are evaluated on benchmark images.

5.6 Evaluations

5.6.1 The Balancing Between Variance and Distance Terms

In the context of KbAS framework introduced above, the addition of variance term (section 5.5.3) is to assist the distance term to determine the priority score for each unsampled pixel by providing another priority metric. In many cases the two terms agree with each other (Figure 5.16: Examples of priority map computed by Eq5.39).
Figure 5.17: The Spearman’s rank correlation coefficient of \( \text{var}(x) \) and \( \text{dist}(x) \) (Eq. 5.33), throughout the acquisition of image “lena”. The positive rank correlation coefficient shows that the two terms agree with each other in pixel ranking, in many circumstances.

such as the case for most pixels in the example shown in Figure 5.11 and 5.14. However in some situations when either of the terms struggles to provide a distinctive information to rank the pixels in priority, the other term may offer extra information.

The aim of designing for combining the two terms is to find a balance between their contribution in determining the priority score. As is mentioned above, this is application dependent. However in this section, discussions about this design are made targeting the collection of benchmark images which are considered to be examples of natural images.

There are various possibilities for the forms that the distance term can take as is shown in Eq. 5.32 and there are also different ways to combine the two terms (design for the operator \( \otimes \) in Eq. 5.5). In this thesis we focus on the discussion of the priority score computed as a product of the two terms:

\[
p(x) = \text{var}(x) \ast \text{dist}(x)
\]  

(5.41)

The estimation of priority scores is to give candidate pixels a ranking. Between two candidate
unsampled pixels $x_1$ and $x_2$, their relative priorities are determined by the following ratio:

$$r(x_1, x_2) = \frac{\text{var}(x_1) * \text{dist}(x_1)}{\text{var}(x_2) * \text{dist}(x_2)} = \frac{\text{var}(x_1) * f(l(x_1))}{\text{var}(x_2) * f(l(x_2))}$$ (5.42)

where $f(l(x))$ is the actual form that the distance term takes, since the distance term is based on $l(x)$. If $r(x_1, x_2) > 1$ then $x_1$ should be sampled before $x_2$.

With the distance term taking several possible forms listed in Eq.5.32.

$$\text{dist}(x) = f(l(x)) = \begin{cases} (1 - l(x))^n \\ \log(1 + \frac{1}{l(x)^n}) \\ \frac{1}{l(x)^n} \end{cases}$$ (5.43)

a set of tests are run on benchmark image “lena” of size 257x257. The test starts with a uniform sampling pattern at sampling distance of 8 and the KbAS algorithms are used to guide the sampling procedure in finding significant pixels. For each iteration, 100 most significant pixels are identified and sampled. At the end of the sampling process (stops at around 4096 samples), the samples are used by linear interpolation to reconstruct for an approximation of the original image. With the variance term defined in Eq.5.31, the resulting performance is shown in Figure ???. In the figure, the two graphs in each row shows the PSNR vs. b/p performance and the shape of $f(l(x))$ respectively.

It can be seen that despite the various forms it takes, the distance term essentially weights the impact of the change of $l(x)$ to the estimation of priority score. All selected options of $f(l(x))$ apart from op1 have a steeper slope when $l(x)$ is close to 0; when $l(x)$ is close to 1 the slope becomes almost flat. This is better shown in Figure 5.18 as the first derivatives $\frac{\partial f(l(x))}{\partial l(x)}$.

This trend of slope changing means that when $l(x)$ is close to 0, or when the likelihood of determining $x$ is considered to be low, the distance term takes dominant presence in the estimation of priority score. A slight change of $l(x)$ will result in a significant change of $f(l(x))$ or $\text{dist}(x)$, resulting in a significant change of priority score. On the contrary when $l(x)$ is close to 1, or when the likelihood of determining $x$ is considered to be high, the priority score is not as...
Figure 5.18: The slopes of different \( f(l(x)) \) choices, shown as \( \frac{\partial f(l(x))}{\partial l(x)} \).
Figure 5.19: The sampling performance using different $f(l(x))$s, and their corresponding shapes.
Chapter 5. Kernel-based Adaptive Image Sampling

Figure 5.20: The sampling patterns at 4096 samples, using different distance terms. (a) op1; (b) op3; (c) op5.

Sensitive to the change of \( l(x) \). In this situation the variance term provides extra information to distinguish between candidate pixels.

The sampling performance in Figure ?? shows that op5 and op6, which are the most true to this implicit balancing strategy, have the best overall performance in terms of PSNR vs. b/p. On the other hand the two options op1 and op2, which are the least true to this balancing strategy, turn out to have the worst performance. The sampling patterns of op1, op3 and op5 are shown in Figure 5.20. It can be seen that without strengthening the power of distance term when \( l(x) \) is close to 0 (Figure 5.20(a)), the samples are overly focused in edged areas due to the high local variance of pixels values. A balanced combination of the variance term and distance term via the slope change of \( f(l(x)) \) reflects both Design Considerations, and results in a better sampling pattern that leads to a higher reconstruction quality.

This is an example of balancing the contribution of the two terms in determining the priority score. Again it is worth emphasising that the actual design is highly application dependent as is explained in the original AFPS paper [ELPZ97].

5.6.2 Evaluation of KbAS Algorithms

In this section, the proposed KbAS algorithms are evaluated on several benchmark images. The evaluation is focused on the image quality (PSNR) vs. number of samples (b/p) ratio. The
higher the ratio is, the better these algorithms can make the trade-off. As is in the previous evaluation tests, the benchmark images are all of size 257x257 and are in grayscale with pixel value ranging from 0 to 255. The sampling starts from a uniform sampling pattern at sampling distance of 8 and in every iteration afterwards 100 most significant pixels are identified by KbAS algorithms and are sampled. The sampling process stops at around 4096 samples, and linear interpolation is used to reconstruct an approximation of the original image using the samples. The evaluation results are shown in Figure 5.22.

In general, because of the use of detailed gradient information, KbAS algorithms produce sampling patterns that result in higher quality reconstructions than that of the grid AFPS algorithm. The selected examples of KbAS algorithms also have higher performance than the method proposed in our previous work (Eq.5.40 op5 in the figure [LBCL14]). Two examples of the reconstruction are provided in Figure 5.21. It can be seen that due to the accurate estimation of underlying image structure, KbAS sampling results in a reconstruction with sharpening effect, producing a better visual quality on top of the higher statistical quality.

5.7 Cost of the Kernel-based Adaptive Sampling Algorithm

In this chapter, Kernel-based Adaptive Sampling is proposed as a family of detailed sampling models of natural images. The discussions are made in the assumption of no overhead computational energy/time in order to investigate for the highest potential image quality vs. b/p performance, which is essential to the proposed Context-based Image Acquisition framework. The detailed modelling of KbAS does require more computational effort to complete than existing methods such as grid AFPS, both in time and energy consumption. In this section, a high-level estimation of the cost of KbAS algorithms is made.

Without implementing the full architecture on practical hardware platforms, the cost of KbAS algorithms and the reference AFPS algorithm [DL07] is estimated by floating-point operation
Figure 5.21: Reconstruction examples using KbAS algorithms, compared with grid AFPS.
Figure 5.22: Sampling/reconstruction results using KbAS algorithms on benchmark images, compared with grid AFPS [DL07]. Five different formulations of KbAS algorithm are evaluated, all producing plausible image quality to b/p ratio. Option 1 in this test, labelled as “op1: rev(1-l(x))”, is the formulation as in Eq 5.30 which is to compute equivalent kernels on sampled pixels.
Chapter 5. Kernel-based Adaptive Image Sampling

<table>
<thead>
<tr>
<th>Operations</th>
<th>flops</th>
</tr>
</thead>
<tbody>
<tr>
<td>addition</td>
<td>1</td>
</tr>
<tr>
<td>multiplication</td>
<td>1</td>
</tr>
<tr>
<td>division</td>
<td>8</td>
</tr>
<tr>
<td>square root</td>
<td>8</td>
</tr>
<tr>
<td>exponential</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 5.1: Flops of example operations. [Min03]

counts (flops). The counts of flops are recorded according to the Lightspeed Toolbox provided by Tom Minka et al. [Min03]. The flop counts of some example operations are listed in Table 5.1. Recorded flop counts during the execution of the sampling algorithms on image “lena” serve as a high-level reference to the amount of computational effort needed.

In this test, the reference AFPS algorithm only computes priority scores on Voronoi centres of the current sampling pattern [ELPZ97, DL07]. At each iteration of sampling, the Delaunay triangulation of the current sampling pattern is incrementally updated by new samples. For the selected KbAS algorithms, each time a new set of samples come in, only the local regions around the new samples are updated with new interpolation results.

In Figure 5.23, the costs (measured in flops) of the chosen algorithms are shown. For KbAS algorithms, option 1 and option 2 from the evaluation in section 5.6.2 Figure 5.22 “lena” are chosen as representatives. Option 1 computes the priority score via constructing equivalent kernels around sampled pixels as in Eq 5.30; option 2 (Eq 5.28) is a representative of the rest of KbAS algorithms which computes priority scores around unsampled candidate pixels. The flop counts in this figure is shown in its logarithm form for visual display. It can be seen that (data points marked in the figure), both KbAS algorithms require much higher flops to finish: KbAS op1 requires \(10^{(8.894 - 7.468)} \approx 27\) times of flops to process 0.5 b/p equal of pixels compared with that of the reference AFPS algorithm; KbAS op2 on the other hand requires \(10^{(9.819 - 7.468)} \approx 224\) times of flops compared with that of the reference AFPS.

In terms of the composition of the computational cost, the majority of the AFPS algorithm is spent on maintaining Delaunay triangulation of the sampling pattern, as well as updating the interpolation of the image with newly sampled pixels. The cost of updating pixel priority scores is the least significant among various types of cost. In the case of the two KbAS algorithms
5.7. Cost of the Kernel-based Adaptive Sampling Algorithm

Figure 5.23: Breakdown of the cost of reference AFPS algorithm, and selected KbAS algorithms.

however, the logic overhead (cost of controlling logics such as finding maximum value) becomes significant because the pool of candidates is larger. Moreover, the cost of pixel priority score computation becomes the dominant part of the overall cost. Additionally, because of the high amount of unsampled pixels compared with the limited amount of sampled pixels, KbAS op2 is significantly more costly than KbAS op1 because it requires priority score updating on each unsampled pixel. Therefore, while achieving a similar image quality to b/p ratio in this case, KbAS op1 is more cost effective than op2.

The cost of running these sampling algorithms is considered to be cost overhead over the conventional image access procedure. In Figure 5.24, these overhead costs are added to the cost of pixel accessing from source memory, assuming each data access has the same amount of cost as one flop. It is also assumed that the access of each pixel has the same amount of cost, regardless of previous accessing sequence and pixel location. The overall cost of image acquisition is normalized to the reduced portion of pixel access cost, as is defined in Eq 5.44:

\[
\text{norm\_cost} = \frac{\text{flops} \cdot c_{\text{flop}}}{(N - M) \cdot c_{\text{access}}} \tag{5.44}
\]

where \(c_{\text{flop}}\) and \(c_{\text{access}}\) are the cost per flop and per pixel access respectively (assumed to be equal); \(M\) is the amount of pixels sampled in current sampling pattern and \(N\) is the total pixels in the target image. The result of the normalized cost is also presented in their logarithm form.
for visual display. Therefore the value of 0 on both graphs in Figure 5.24 show the place where progressive sampling breaks even with conventional image accessing procedure in terms of the cost. It can be seen that, in order for the AFPS algorithm to be beneficial in reducing the overall cost of image acquisition at 0.5 b/p sampling rate, the cost (time or energy consumption) of computing one flop needs to be at least about $10^{2.6} \approx 427$ times less costly than that of accessing one pixel; the KbAS op1 requires the cost of computing one flop at least about $10^{4.1} \approx 11858$ times less costly, and KbAS op2 requires the cost of computing one flop at least about $10^5 = 100000$ times less costly.

The discussion of point sampling algorithms in this chapter is to investigate for the best potential image quality vs. b/p performance of progressive sampling procedures. Referring to the test described in section 3.3.2 on the platform of Hardcopy IV structured-ASIC, accessing one pixel from a DDR3 memory equals to performing about 250 ADD or 64 MULT operations in energy consumption. While the proposed KbAS algorithms provide detailed modelling of natural images and achieve better quality vs. b/p ratio, it is unlikely to be able to reduce the overall cost of image acquisition process if implemented on existing hardware platforms.
5.8 Conclusion

In this section, the best potential quality vs. b/p ratio of progressive point sampling of images are investigated. The generalized Kernel-based Adaptive Sampling is proposed as a detailed modelling of natural images, which leads to a better image quality vs. b/p performance than existing point sampling methods. The proposed KbAS algorithms come with high computational overheads as a drawback and is not beneficial to Context-based Image Acquisition framework in current hardware environments. Nevertheless, the design of KbAS algorithms sets a new upperbounds of achievable image quality given a fixed number of samples. Moreover, the Design Considerations as well as the generalized KbAS framework offer guidance for the design of CbIA applicable image point sampling strategies.

The discussion of image point sampling algorithms in this chapter remains based on the basic assumption of signal continuity of natural images. In the next chapter, explicit use of prior knowledge learned from a known class of images is discussed.
Chapter 6

Domain Specific Image Acquisition of Face Images

6.1 Introduction

The main challenge of the CbIA framework is the design of sampling-reconstruction algorithm pair that estimates the ground truth image data using as few samples as possible. In the previous chapters, this problem is approached by exploiting the signal continuity of the natural image data which serves as *a priori*. The discussions made are therefore applicable to a wide range of natural images. In this chapter, the design of CbIA systems is discussed with emphasis on the explicit use of prior knowledge, in the form of a collection of example images.

Example images that share similar features or structures as the target image does have the potential to offer additional statistical information to the sampling and reconstruction of the target image. Among various classes of images that share intra-class content similarities, the class of human face images exemplifies the benefit of using learned prior knowledge in many applications [BK00, BK02, WYG+09]. Targeting this particular class of images, the discussion in this chapter is made to investigate and design a specialized CbIA procedure - the “domain-specific CbIA procedure” - that deals with in-class face images. The proposed domain-specific CbIA procedure for human faces is to deal with the following problem: given a collection of
human face images as training database, the procedure should be able to acquire an unknown face image from the source memory with efficiency, i.e. achieving a high reconstruction quality using as few samples as possible.

Through the use of learned prior knowledge, domain-specific CbIA procedure is expected to have a better image quality vs. b/p ratio when the target image is indeed of the same class as the training examples are. At the same time the use of learned statistics in sampling and reconstruction process introduces computational overheads, compared with that of the basic point sampling and interpolation method in the prototype system explained in chapter 4. This chapter is dedicated to the discussion of such trade-offs and the impact of using explicit prior knowledge in the design of CbIA systems.

It is worth noting that, although the proposed CbIA procedure for face images breaks down face images into small blocks/patches to process, the application of this work is different from those in previous chapters in that it is no longer in a macroblock-based manner. Whereas methods proposed in previous chapters work to acquire a macroblock from the target image at the requested location, the domain specific CbIA procedure for face images works to acquire a complete target face image and store the reconstruction in the local buffer. From another perspective of view, the difference with methods in previous chapters is that for this chapter, the “macroblocks” to acquire are known to belong to the specific class of face images, each of them being one human face.

In the rest of this chapter, section 6.2 introduces the relevant research fields and connects them with the proposed CbIA system design; section 6.3 gives an overview of the domain-specific CbIA procedure; section 6.4 explains the design of the domain-specific procedure in detail, covering the design of the reconstruction algorithm and the corresponding sampling procedure; in section 6.5 evaluation results are given, showing the performance of the design; section 6.6 discusses the cost of implementing and executing the domain-specific CbIA procedure in high-level simulation; finally section 6.7 concludes this chapter.

The main contributions of this chapter include:
1. The problem of CbIA procedure design is extended and formulated around the projection of image data onto example space, enabling the discussion of the use of explicit prior knowledge of a particular class of images.

2. An example domain-specific CbIA procedure for face images is designed and evaluated, demonstrating the impact of explicit use of example images in the process of sampling and reconstruction.

### 6.2 Review: the Hallucination of Face Images

For the proposed “Context-based Image Acquisition (CbIA)” framework, the problem setting is essentially image restoration which is similar to that of the still image Super-Resolution problems in that only fractions of the target image are acquired and the rest has to be estimated afterwards.

In the context of image restoration and image upsampling/Super-Resolution, interpolation and regression techniques are often used to fill in the missing parts of the target image. Although some algorithms (such as the “New Edge-Directed Interpolation (NEDI)” [LO01]) introduce more complex models of the image data, they are based on the fundamental assumption of signal continuity of natural images and therefore are applicable to a wide range of image candidates. Particularly in the field of image upsampling/Super-Resolution, when the target image in question is known to be of a particular class, more information can be gathered and learned from examples of the same class of image data. This additional information is then used to achieve an output with better quality (lower MSE and/or sharper features etc.). This is often seen as example-based image Super-Resolution.\(^1\) The core of using examples is to add in extra information on top of what can be gathered from available low resolution input. By the explicit use of these extra information, Super-Resolution algorithms can break the limits posed by the ill-defined problem of image reconstruction [BK02].

\(^1\)Here “example-based Super-Resolution” is a general term describing the collection of Super Resolution algorithms that make explicit use of image examples. To be differed from the actual algorithm “Example-based Super-Resolution” proposed by Freeman et al. [FJP02].
There are many algorithms developed to make use of the learned statistics from examples to infer high resolution counterparts of the input. Examples of these algorithms include the well established framework proposed by Freeman et al. [FJP02]. Out of different classes of natural data, human face images are of particular interests to researchers due to their well structured features and their potential in many applications. For face images, the problem is often referred to as face hallucination [BK00].

In the work of Wang et al. [WT05], the estimated high resolution face $\hat{I}_H$ of a low resolution input $I_L$ is formed as a weighted sum of a collection of example high resolution faces. The weights are computed by projecting the low resolution input $I_L$ to the eigenspace trained from the low resolution version of the example database, to mimic the impact of downsampling (Figure 6.1). Later in the work of Liu et al. [LSF07], the use of example face eigenspace is integrated into a two stage hallucination framework. In this framework, following a global (full face) hallucination process using the low resolution input, the hallucinated face with higher resolution is further used as a base on which more local details are added.

Figure 6.1: Face hallucination via eigen transformation [WT05]. The projection coefficients are computed using a LR version of the example database. The hallucination is done by mapping these coefficients back to their HR counterparts and guide the reconstruction with the HR example database.

The works of Wang and Liu are representatives of face hallucination by assembling example face images. Later works that share the same concept also employ the method of sparse representation [YTMH08], as is used in the field of face recognition [WYG+09].

On top of these projection-based hallucination algorithms, there are other methods developed
to hallucinate faces by examples. The work of Hu et al. [HLQS11] uses example high resolution faces to extract local pixel structures. Several high resolution examples which are most similar to the input low resolution face $I_L$ are first selected. These examples are then warped to match the features of $I_L$ and local pixel structures (statistical relationship between pixels) are extracted from these warped examples to guide the estimation of missing pixels of $\hat{I}_H$ (Figure 6.2).

**Figure 6.2:** The method proposed by Hu et al. [HLQS11], in which the example HR images are first warped to match the structure of the input LR image. The warped HR examples are then used to learn local pixel structures for the regression of missing pixel values in the LR image.

Despite of the different ways of using example faces, face hallucination algorithms all emphasize on the explicit use of example faces. This is because of the special characteristics of this particular class of image data such as shared structures and significant feature landmarks.
These designs of face hallucination methods demonstrate the significant benefit of employing learned prior knowledge.

In previous chapters, interpolation/regression-based designs of CbIA systems are discussed using a minimal amount of \textit{a priori} for a universal application. In this chapter, the explicit use of learned prior knowledge in the design of CbIA systems is discussed with human face images as an example. The discussion is to show that under the framework of CbIA, the use of learned prior knowledge in the sampling-reconstruction algorithm pair has the potential of improving the quality of the final output.

### 6.3 Overview of the Domain-specific Point Sampling of Faces

Following the general design of CbIA framework described in chapter 3, the designed image acquisition procedure still progressively samples pixels from a target image to acquire. During the sampling process the system decides whether or not enough pixels have been sampled and the process should stop. At the end of each sampling steps, acquired pixels are used to hallucinate the ground truth face image. The key concept in the domain-specific CbIA system is the use of hallucination algorithm in image reconstruction. Compared with interpolation/regression algorithms, it employs more prior knowledge learned from a given database. Therefore the domain-specific algorithm is expected to be able to compensate for the artefacts generated by point sampling, and therefore to achieve a better quality vs. b/p ratio in general if the target image to acquire is indeed of the same class as the database. It is reported in various related literatures that when such projection based method is applied to images that is of different class as the example database, the reconstruction result is unsatisfying (e.g. the work of Wright et al. [WYG+09]).

As is explained in section 5.2, the design of CbIA procedure is essentially to design the sampling-reconstruction pair. Based on the different \textit{a priori} employed by the chosen reconstruction
algorithm, the sampling process is to be designed accordingly to acquire the most needed
information to improve reconstruction quality. This is also one of the focuses of designing
domain-specific CbIA procedure.

In general, the algorithm listed in algorithm 2 is proposed as an example of the face-oriented
domain-specific CbIA procedure. The objective is to reconstruct an approximation $\hat{I}_H$ to the
ground truth $I_H$ using as few samples from $I_H$ as possible, in order to reduce the effort of memory
accessing as is the case for discussions in previous chapters. A set of sampling patterns and the
eigenspace $B$ are pre-learned off-line from the given database. The system works on patches of
the ground truth instead of the whole image, for the reasons explained in the following section.
The process is shown in Figure 6.3.

Algorithm 2 Overview of the proposed method

Require: for each patch location, the learned eigenspace $B$ and a set of sampling patterns $S$
of different resolution level; target face image $I_H$ to acquire; an initial sampling of $I_H$ at the
lowest resolution level ($S_0$) of each patch location and the initial reconstruction of each patch

Ensure: the updated reconstruction of the image $I_H^*$

For each patch location, advance to the next sampling resolution level, sample more pixels
according to $S_1$

Compute the validation error $e$ of each patch location, defined as the mean squared error of
the newly sampled pixels and the previous approximations of these pixels

Reconstruct using learned eigenspace at each patch location, using existing samples

while $I_H^*$ does not meet the validation requirement do

At the patch location with highest $e$, advance to the next sampling resolution level by
sampling more pixels according to $S_{i+1}$

Update the validation error $e$ by comparing the newly sampled pixels and their approximations before

Update the reconstruction of the patch

end while

return updated $I_H^*$ as an approximation to $I_H$

6.4 Design of the Domain Specific Point Sampling of Faces

In this section, the details of face-oriented domain-specific CbIA procedure are explained. The
reconstruction problem, which is the core idea of domain-specific CbIA, is first formulated and
Figure 6.3: Overview of the face-oriented domain-specific sampling-reconstruction process. For a given patch location, e.g. the region marked in the figure, a set of sampling patterns are learned off-line from the training database. Samples retrieved are used for patch reconstruction with learned codebook in the form of eigenspace.

discussed. Based on the reconstruction algorithm, a sampling procedure is designed to access pixels that provide the most information of the target image to the reconstruction.

6.4.1 Reconstruction by Hallucination

The reconstruction problem is modelled in a similar way to the global hallucination problem in the work of Liu et al. [LSF07]. While the main body of the hallucination method is the same as in Liu’s work, the application of such method in the context of image sampling poses different challenges and require different treatments. In this section the application of their face hallucination method in the context of image progressive sampling is explained.

Formulation of the Hallucination Problem

A target face image is treated as a column vector $I_H$ of dimension $M$. Given a collection of $N$ example face images of the same size and alignment, a example space $B$ ($M \times r$ where $r < N$) is built. The hallucination of $I_H$ is to find the optimal set of projection coefficients $g^{opt} (r \times 1)$ that results in a projection $\hat{I}_H^{opt}$ of the least residual:

$$g^{opt} = \arg\min_g \| I_H - B \ast g \|_2$$  \hspace{1cm} (6.1)
Chapter 6. Domain Specific Image Acquisition of Face Images

\[ I_H^{opt} = B \ast g^{opt} \]  

(6.2)

Assuming that the example space \( B \) is indeed capable of producing such a projection \( \hat{I}_H^{opt} \) with acceptable residual error, then for a hallucination-based SR problem the objective is to approximate the underlying optimal coefficients \( g^{opt} \) with only the low resolution version of the target image \( I_L \) as a constraint. In the context of image point sampling process in the CbIA system, the effect of sampling is denoted by applying the sampling matrix \( S \) (\( n \times M \)) to the target image to acquire a vector of sampled pixel values: \( I_L = S \ast I_H \).

\[
S = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \ldots \\
& & & & & & & & & \\
& & & & & & & & & \\
& & & & & & & & & \\
\ldots & & & & & & & & &
\end{pmatrix}
\]  

(6.3)

This is similar to the generation of low resolution inputs in SR problem with the differences lying only in the contents of the downsampling matrix \( S \). For point sampling problem, \( S \) is a \( n \times M \) matrix mostly filled with zeros while in the SR problem \( S \) also introduces low-pass filtering effects. Nevertheless, in the discussion of CbIA system design the same notations are used: \( I_L \) represents the input fractions of the image data and \( I_H \) represents the ground truth image data with full information. In the CbIA system, the reconstruction problem (Figure 6.4) has the objective of:

\[ g^* = \arg \min_g \| I_L - S \ast B \ast g \|_2 \]  

(6.4)

and consequently the reconstructed approximation is:

\[ I_H^* = B \ast g^* \]  

(6.5)

The learning and construction of example space \( B \) may take various approaches, including methods from both compact coding [WT05] and sparse coding (e.g. the work of Yang et
6.4. Design of the Domain Specific Point Sampling of Faces

al. [YTMH08]). In this chapter eigenspace of the original example database is trained using Principle Component Analysis (PCA). The eigenspace $B$ is formed of principle components as column vectors in the descending order of their corresponding eigenvalues. The objective function Eq 6.4 is then changed to:

$$g^* = \arg\min_g \| I_L - S^* (B^* g + \mu) \|_2$$

(6.6)

where $\mu$ is the mean face of the examples. The reconstruction $I_H^*$ is then:

$$I_H^* = B^* g^* + \mu$$

(6.7)

Solution of the Hallucination Problem

Due to the nature of the image sampling problem, the added sampling matrix $S$ is constantly changing. This poses challenges to the otherwise fixed hallucination problem. In this section,
the solving of hallucination problem based on image sampling is discussed.

In the work of Liu et al. [LSF07], the solving of this optimization problem is categorized into two situations, which are soft constraint and hard constraint. The soft constraint is the situation where \( r < n \) which means the number of eigenvectors is smaller than the dimension of \( I_L \). The problem is over-constrained in this case and can be solved. But to further guarantee a face-like reconstruction, additional constraints in the form of weighted \( l_2 \) norm of \( g \) is added as a regularization term:

\[
g^* = \arg \min_g (\| I_L - S \ast (B \ast g + \mu) \|_2 + \| \Lambda^{-\frac{1}{2}} \ast g \|_2) \tag{6.8}
\]

where \( \Lambda \) is the diagonal eigenvalue matrix. The hard constraint describes the situation where \( r > n \), i.e. the number of eigenvectors is greater than the dimension of \( I_L \). In this situation \( g \) is under-constrained and there is enough freedom to precisely formulate the constraint. Additional constraints have to be added to find a solution. In the work of Liu et al. [LSF07], the problem under hard constraint is solved to first fully meet the constraint of \( I_L = S \ast (B \ast g + \mu) \). Then the final \( g \) is solved by minimizing the same weighted \( l_2 \) norm of \( g \) as is the case for soft constraint.

The above is how the optimization problem is solved in conventional face Super Resolution task. While in the problem of image Super Resolution the number of eigenvectors \( r \) and the dimension \( n \) of low resolution input are often fixed, their relationship is constantly changing in the context of progressive sampling in CbIA framework: as more and more samples are accessed \( n \) increases accordingly. The CbIA procedure may start with \( n < r \) and proceed to a point where \( n > r \). This change of relationship between \( r \) and \( n \) makes the problem of CbIA posses new challenges on top of the conventional hallucination problem. On one hand, a balance has to be made dynamically between the weightings of residual error and regularization term in the objective function; on the other hand, there will be a transition from hard constraint solution to soft constraint solution as \( n \) increases and it needs to be carefully designed to avoid over-fitting (explained later).
Therefore, instead of breaking down the problem into hard constraint and soft constraint situations, in this work a Bayesian treatment is given to the problem to form a unified maximum a posteriori (MAP) problem:

\[
p(I_H \mid I_L) \propto \frac{p(I_L \mid I_H) \ast p(I_H)}{p(I_L)} \tag{6.9}
\]

Since the estimate of the target image is determined solely by \( g \) in the reconstruction function (Eq. 6.7), the MAP problem is to estimate for the \( g \):

\[
g^* = \arg \max_g [p(I_L \mid g) \ast p(g)] \tag{6.10}
\]

With likelihood gauges the residual error of the reconstruction:

\[
p(I_L \mid g) = \frac{1}{\text{const}} \exp\left\{-\frac{1}{\sigma^2_{n,r}} [S(Bg + \mu) - I_L]^T [S(Bg + \mu) - I_L] \right\} \tag{6.11}
\]

The prior is the same as in Liu's work [LSF07], which adds constraint in the form of weighted \( l_2 \) norm of \( g \) to guarantee a face-like reconstruction:

\[
p(g) = \frac{1}{\text{const}} \exp\left\{-\frac{1}{2}g^T \Lambda^{-1}g \right\} \tag{6.12}
\]

where \( \Lambda \) is a diagonal matrix with entries being the eigenvalues corresponding to eigenvectors in \( B \).

Compared with the hard/soft constraint solutions in the work of Liu [LSF07], there are two major differences due to the nature of progressive sampling:

Firstly, to adapt to the constantly changing dimension \( n \) of the input degraded face (sampled pixels in this case), the parameter \( \sigma^2_{n,r} \) is added to the likelihood (Eq. 6.11). This parameter determines the balance between likelihood and prior and is set to take into account different amount of pixels available in \( I_L \): \( \sigma^2_{n,r} = c \ast \frac{n}{r} \), where \( c \) is a constant coefficient. This parameter makes sure the contributions of likelihood and prior to the posterior are of equal ratio \( c \), not
mater how \( n \) and \( r \) change during the CbIA procedure.

Secondly, there is no separation between hard and soft constraint. In both situations of \( r \geq n \) and \( r < n \) the solution is produced from Eq. 6.10. It is because when the dimension \( n \) is constantly changing, using the solution for hard constraint situation proposed by Liu [LSF07] may lead to unsatisfying result. Because hard constraint solution is forced to completely eliminate the residual error first, when \( n \) is smaller but close to the number of eigenvectors \( r \), it is likely to cause unsatisfying reconstruction due to over-fitting. To elaborate this, the following test is conducted.

**Test:** A collection of 500 examples are used as training database to learn for the eigenspace \( B \). The eigenspace contains a variate number \( (r) \) of eigenvectors. Another set of 100 face images are used as testing targets, each of them uniformly sampled at decreasing sampling distances to get \( I_L \) which is then used to hallucinate for an approximation \( \hat{I}_H \) of the original \( I_H \). The hallucination is done using the combination of hard constraint and soft constraint solutions from the work of Liu [LSF07] but with the proposed adaptive parameter \( \sigma^2_{n,r} \). The whole test works on 17x17 patches \( (M = 289) \) of the face image instead of the full image, and eigenspace \( B \) is learned separately for each patch location.

As the sampling distance decreases from 16 to 8 and eventually to 2, more pixels are sampled from the target image which results in \( I_Ls \) with increasing dimensions. With this increment in input information, an improvement in the hallucination quality of \( \hat{I}_H \) is expected. However this is not the case across the test with different number \( (r) \) of eigenvectors as is shown in Figure 6.5(a).

Four points, A, B, C, and D break the expected continuous increment of reconstruction quality as the number of samples increases. Table 6.1 lists the occurring conditions of these over-fitting points. It can be seen that when \( n \) increases and approaches \( r \) the over-fitting happens. In these situations although there is enough freedom to precisely formulate the likelihood constraint, there is little freedom remaining to prevent over-fitting. This problem can be mitigated in an image SR task by careful selection of \( n \) and \( r \). However in the context of image progressive sampling in CbIA systems, it is impractical to have a set of fixed \( n \) and \( r \) and therefore the
Figure 6.5: Over-fitting of hard constraint solution. In this graph, the reconstruction qualities under various $n$ and $r$ are plotted. a) is the solution using the hard/soft constraints in Liu’s work [LSF07]; b) is the solution using the unified MAP formulation in Eq. 6.10.

unified Bayesian treatment (Eq 6.10) of the optimization problem is introduced which, when working on the same test, results a smooth performance plot in Figure 6.5(b).

<table>
<thead>
<tr>
<th>Point ID</th>
<th>sampling distance</th>
<th>n</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>16</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>B</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>25</td>
<td>40</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>25</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 6.1: The over-fitted data points in Figure 6.5

The Closed Form Solution

The proposed optimization problem above has the final closed form solution as follows:

$$g^* = (B^T S^T S B + \sigma_{n,r}^2 \Lambda^{-1})^{-1} B^T S^T (I_L - S \mu)$$

To stay true to the sampled information, in the domain-specific CbIA system only the missing pixels are filled in with the hallucinated result while the samples remain unchanged.
6.4.2 Patches vs. Full Image

As is mentioned in the overview of domain-specific CbIA system, the proposed design works on patches of the target face image instead of the full image. The reconstruction algorithm explained in the previous section is applicable to either image patch or full image. Working on image patches, or applying the objective function locally, allows the eigenspace projections ($g$) better flexibility to adapt to local samples with a fixed number of example faces (refer to the work of Jung et al. [JLG11]).

More importantly, breaking the image into patches makes local updating of the sampling/reconstruction during sampling iterations possible. Such flexibility leads to the possibility of focusing the otherwise limited computational power/bandwidth on local regions of the image which are considered to be of highest priority. During each iteration of the progressive sampling process, it is then possible to identify and refine the one patch with highest estimated error, without spending effort in other patches. This is another layer of sampling priority estimation, on top of the estimation of pixel priorities as is the case in previous chapters. More on the sampling order of the pixels and patches is explained in section 6.4.4.

6.4.3 Learning from Database

The ability to hallucinate missing details comes from a collection of well selected example faces, as well as a well trained example space $B$. The “quality” can be regarded as how much the input image shares the similar structure and features as database examples. Ideally if the matrix $B$ contains all information available from the examples, then the projection of the ground truth target image should result in a minimum residual error:

$$
\| I_H - I_H^{opt} \|_2 = \epsilon_{min} < \epsilon_0
$$

(6.14)

where $\epsilon_0$ is the maximum acceptable error. This sets the upper-bound of the reconstruction quality using a particular set of examples, and the progressive updating of $g^*$ is to approximate $g^{opt}$ with an increasing number of samples. The upper-bound in Eq 6.14 requires that the
example database contains example images/features with high similarity to the target image. Therefore training from a specific class of images can only be expected to achieve good reconstruction quality of an input image of the same class. Human face images are a class of images that exemplifies this characteristic. Although the textures and details are different between different faces, they all share a similar structure. Therefore even face images from different subjects (i.e. people) can be used to as reconstruction examples.

The upper bound in Eq 6.14 also requires that the re-organization or training of the final example space $B$ preserves enough features to be able to contain a potential target image $I_H$. Eigenspaces are trained on different patch locations in the example images (Figure 6.6). From the full eigenspace, eigenvectors with the highest corresponding eigenvalues are preserved in order to reduce the computational cost and noise effect.

For patch location $(i, j)$ of size $u \times v$, the system preserves the main fraction of the power of eigenvectors trained from $N$ database examples. The number of eigenvectors to preserve $r_{i,j}$ equals to:

$$r_{i,j} = \min r \text{ s.t. } \sum_{i=1}^{r} ||\lambda_i|| > q \times \min (u \times v, N-1) \sum_{i=1}^{\min (u \times v, N-1)} ||\lambda_i||$$

(6.15)

Given the same threshold $q$, for patches of smaller cross-image variance the number of eigenvectors to preserve will be smaller as well. A larger sized training database will also require more eigenvectors to describe. Figure 6.7 shows the number of eigenvectors in different patch locations under various setups (patches are of size 17x17 and are defined in the same fashion as in Figure 6.3). It can be seen that the graphs roughly resembles the structure of face im-
ages. In locations of complex facial features such as eyes and mouth, more eigenvectors are preserved. Increasing $q$ leads to larger number of eigenvectors being preserved, and a bigger example database also requires more eigenvectors to store the feature information.

![Figure 6.7: Number of eigenvectors in different regions, given different sized training database and different threshold $q$.](image)

### 6.4.4 Sampling Order and Validation

The sampling process should be designed to compensate for the reconstruction process, aiming to increase the information gain per transferred pixels. Data of higher potential priority should be sampled first. To allow for hierarchical refinement over sampling iterations, two types of sampling priorities are defined for domain-specific CbIA sampling process: patch priority and pixel priority. The task is to identify both the patch and in-patch pixel locations (unsampled
6.4. Design of the Domain Specific Point Sampling of Faces

sites) that are likely to bring most information gain when sampled in the next iteration.

**Patch Level Priority**

Unlike previous generic CbIA designs, hallucination based reconstruction algorithm works on regions of the image instead of individual pixels. Therefore the proposed design breaks down the target image into patches to introduce the patch level priority. Priority of patches is computed during runtime by the validation error between newly sampled pixels from this iteration and their approximation from previous iteration. At every iteration during the sampling procedure, the system picks the patch location that has the highest average validation error to sample from. The whole process stops when validation errors in all patch locations are below a threshold. The use of validation process requires a reconstruction process be done at each iteration for patch locations that are updated with new samples. This introduces additional computational overhead to the system. There are potentially other methods to estimate for the patch level priority using existing samples without reconstruction of the patch. This will be part of the future research plan of this project.

**Pixel Level Priority**

Within each patch, pixel level priority is learned off-line. Within the patch, in order to determine the priority of each pixel, the extended point sampling framework proposed in section 5.3 is used:

\[ p(x_i) = \text{var}(x_i) \otimes \text{dist}(x_i, P) \]  

(6.16)

Where \( \text{dist}(x_i, P) \) measures the “distance” from pixel \( x_i \) to the current sampled pixels, i.e. the likelihood of determining pixel \( x_i \) with existing samples. This distance term includes but is not limited to Euclidean distance of pixel coordinates, which is used in previous point sampling methods (e.g. the work of Eldar [ELPZ97]). The variance term \( \text{var}(x_i) \) is the estimated variance of pixel \( x_i \). To accommodate the hallucination process we observe that convergence of the hallucination algorithm is determined by whether or not the pixels of most variance across
Figure 6.8: Learning for sampling patterns: (a) the variation map of the patch location marked in Figure 6.3; (b) the initial ($S_0$) sampling pattern with only 4 samples, one at each corner (white dots are pixel locations to sample); (c) the priority map computed at each pixel by Eq 6.16; (d) sampling pattern $S_1$ at level 1 iteratively picks pixel locations with highest priority in (c) and update the priority map accordingly; (e) updated priority map after $S_1$.

database examples have been sampled. Therefore we model the two terms as:

$$\text{dist}(x_i, P) = \min_j (1 - \text{corr}_{i,j}), \quad x_j \in P \quad (6.17)$$

$$\text{var}(x_i) = \text{var}(I_k(x_i, y_i)), k = 1, 2, \ldots N \quad (6.18)$$

Where $\text{corr}_{i,j}$ is the correlation between pixel $x_i$ and $x_j$ in database, and $I_k$ is the $k$th example image in database. An example is given in Figure 6.8. In this example, examples in a single patch location of size 17x17 are used to estimate the priority of each pixel location within the patch. Combining the variation (Eq. 6.18) of values at each pixel location, as well as the correlation (Eq. 6.17) of each pixel with existing samples, an initial priority map is estimated and shown in Figure 6.8(c). Notice that when pixel locations with high priority are picked iteratively (Figure 6.8(d)), the priority scores of surrounding locations with high correlation to them are lowered (Figure 6.8(e)). This reflects the conventional concept of “the distance to the current sampled pixels”, but in a form more suitable to the chosen hallucination-based reconstruction algorithm.

The learning process iteratively sets up several levels of sampling patterns according to this priority off-line, with more pixels sampled at higher level (Figure 6.3). During reconstruction, every time when a patch location is called to be sampled next, the system advances to a higher level of sampling pattern of this patch and samples more pixels accordingly from external data source.
Summary of Sampling Order

By a combination of patch level and in-patch pixel level priorities, the system is able to identify patch locations with high estimated reconstruction error in each iteration, and call for sampling procedure to refined this patch by accessing the missing pixels with highest priority which is learned from examples off-line. Both priority scores help to improve the reconstruction quality using the fewest sampling effort.

6.5 Evaluations

The proposed method is evaluated on the FERET database [PWHR98] of 1752 frontal faces, and the ORL database [SH94] of 400 faces. All faces were resized to $129 \times 113$ and were broken down to $17 \times 17$ patches overlapping each other by 1 column/row. For each patch location, sampling patterns containing 10, 30, 60, 100 and 150 pixels are pre-learned. For each test out of a total of 5, a number of training face are randomly selected from the database. The reported performance below is an average across repeated tests. For demonstration purpose, the sampling process in all tests stops when about 15% of the total pixels are sampled, showing the most informative data during the sampling process.

Because different persons (testing subjects) have different facial features, face images of the same person/testing subject are considered to be “close examples” to each other (Figure 6.9). Two sets of evaluations (using the same settings explained above) are therefore performed with and without close examples of the target face image, to investigate the impact of such close examples to the quality of the database as well as the performance of the proposed CbIA procedure.

As is the case in Chapter 5 (Section 5.6), the proposed method is compared with the grid AFPS [DL07] to show the impact of introducing domain specific knowledge to sampling without pre-processing or compressing the target image. It is worth noting that although the proposed method is inspired by hallucination and Super Resolution algorithms, the problem settings
Chapter 6. Domain Specific Image Acquisition of Face Images

Figure 6.9: Examples of face images of the same testing subject in FERET database.

are different. The proposed Domain-specific CbIA procedure remains an image progressive sampling procedure which combines hallucination-based reconstruction with tailored sampling patterns. Therefore the evaluation is conducted to compare the proposed method with reference image progressive sampling algorithm, which is grid AFPS, instead of hallucination or Super Resolution algorithms.

6.5.1 Experiments without close examples of the testing subject

In the first series of tests, training faces images that are drawn from the database do not include any close examples from the same testing subject (person) of the target image. The proposed method is evaluated under various levels of thresholds ($q$) and various number of training images.

An example of reconstruction is shown in Figure 6.11. In this particular example, the eigenspace codebook was trained from 500 random faces from the FERET database, excluding close examples of the testing subject. It can be seen that the proposed method can achieve a better approximation quality (in PSNR) than state-of-art method does [DL07], especially in early stages. The faces reconstructed by the proposed method also exhibit much sharper features, by
virtue of the hallucination based reconstruction algorithm. Even though the training database is randomly selected and does not include examples of the same testing subject, the codebook learned can still resemble the target face by filling in the missing pixels with hallucinated data (Eq. 6.13). More examples of various testing subjects are given in Figure 6.10 showing a similar trend. For all four face images in Figure 6.10 reconstructions in (b)(c) are based on grid AFPS sampling, which show lower PSNR than that of (d)(e), the reconstruction from the proposed method.

Figure 6.10: Additional examples of the performance comparison at the iteration when 5% and 12% pixels are sampled. Same as the test in Figure 6.11, 500 faces are randomly selected for training, excluding any examples of the testing subject. For each testing face shown in this graph, (a) is the ground truth image; (b)(c) are reconstructions from global grid AFPS and triangulation-based linear interpolation; (d)(e) are reconstruction from the proposed sampling and reconstruction method.
Figure 6.11: Example reconstructions with different amount of pixels sampled; (b)-(e) are reconstruction examples obtained from global grid AFPS and triangulation-based linear interpolation; (f)-(i) are reconstruction examples obtained from the proposed method with $q = 99.9\%$ and 500 training images in the database, excluding any examples of the testing subject. The locations of sampled sites for these reconstructions are shown as well, below their corresponding reconstructions.
Figure 6.12 shows the overall sampling performance of both the reference method and the proposed method. The performance is taken at each sampling iteration given $q = 99.9\%$ (left) and $q = 99.5\%$ (right), and is measured in reconstruction PSNR vs. percentage of samples required. It can be seen that a larger training database allows for a higher flexibility of the projection to fit samples from target image, due to increased number of eigenvectors in the codebook, and therefore provides better reconstruction results. This difference becomes more significant in late stages of the sampling process, when a large amount of pixels are sampled and the relatively smaller codebook is unable to fit for the numerous samples.

![Graph showing performance evaluation](image)

Since the hallucination based method is mainly designed to improve the reconstruction quality early on when samples are relatively sparse, performance measurements in early stages are of particular interests. A significant improvement in PSNR can be seen in this graph compared with that of the reference point sampling/interpolation scheme especially in early stages. This difference in reconstruction PSNR diminishes as the number of samples increases. When about 15%-20% total pixels are sampled, conventional point sampling and interpolation can achieve a similar PSNR as the proposed method. At this point, the PSNR of the reconstruction is (typically) above 35 dB already. In situations where such reconstruction quality is still considered to be unsatisfying, conventional method can simply take place and keep refining the
patches. Therefore the system can benefit from the early high performance of the proposed method, while being compatible with conventional point sampling methods.

Additionally, the dashed line in Figure 6.12 shows the reconstruction of the target image using triangulation based linear interpolation, but with samples retrieved by the sampling pattern generated for the proposed reconstruction algorithm. It can be seen that different sampling patterns serve different reconstruction algorithms: while the learned sampling patterns improve the reconstruction quality of the hallucination based algorithm, it is not derived from the continuity assumption of images, which is the foundation of interpolation algorithms. Therefore the good sampling performance of the proposed system comes from both the use of domain specific codebook for reconstruction, and the specially tailored sampling patterns.

6.5.2 Experiments with close examples of the testing subject

In the second series of tests, training faces images that are drawn from the database include several close examples from the same testing subject (person) of the target image. However, the exact target image is still not included in the training database. The proposed method is evaluated under various levels of thresholds ($q$) and various number of training images.

As discussed earlier, larger threshold $q$ during training process will preserve more eigenvectors and therefore bring more flexibility to the system to solve the projection problem. On the other hand, the selection of database affects the performance of the proposed system as well. A large database will often provide more information about structures of face images. Given a fixed threshold $q$, more eigenvectors are often needed to meet the threshold when describing a larger database. What is also essential to the performance of the system is whether or not there are example faces/patches that are close to the target image. Therefore, on top of the previous experiments, tests with training database including close examples of the target subject were carried out. Such close examples can be seen in Figure 6.9. The results of these tests are given in Figure 6.13.

With close examples of the target subject included, the system is more capable of modeling
6.6 The Cost of Domain-specific Sampling

In previous sections, the discussion of the proposed Domain-specific CbIA procedure for human faces are based on software simulation and it is evaluated as an image point sampling algorithm. It is shown that by utilizing learned prior knowledge, the domain-specific CbIA procedure is able to trade image quality for reduced bandwidth requirement (b/p) more effectively than state-of-art point sampling algorithm. This section is dedicated to the evaluation of the cost of the proposed Domain-specific CbIA.
<table>
<thead>
<tr>
<th>Num. examples</th>
<th>q</th>
<th>99.5%</th>
<th>99.7%</th>
<th>99.9%</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>q</td>
<td>4.3/21%</td>
<td>5.7/28%</td>
<td>9.9/49%</td>
</tr>
<tr>
<td>400</td>
<td>q</td>
<td>5.7/28%</td>
<td>7.9/39%</td>
<td>13.6/67%</td>
</tr>
<tr>
<td>600</td>
<td>q</td>
<td>6.0/29%</td>
<td>8.3/41%</td>
<td>14.8/73%</td>
</tr>
</tbody>
</table>

Table 6.2: On-chip memory bits required for the storage of learned prior knowledge. The memory bits are measured in Mega bits, and is compared with the total amount of block RAM bits available to Stratix IV EP4SGX530KH40C2 [Alt12].

### 6.6.1 Storing and Accessing Learned Prior Knowledge

A major difference between Domain-specific CbIA and generic point sampling is that Domain-specific CbIA makes use of learned prior knowledge, which is in the form of eigenspace matrix $B$ for each patch location. These learned matrices are accessed repetitively during the CbIA procedure, and are stored in local buffers such as scratchpad memory or BRAMs. From the closed form solution in Eq. 6.13, it can be seen that since the sampling patterns are learned off-line, the majority of this equation can be computed off-line as well to minimize the computational cost of the procedure. In detail, the following part can be computed off-line for each patch location at each sampling level:

$$B_{i,j}' = (B_j^T S_i^T S_i B_j + \sigma_{n_i,r_j}^2 \Lambda_j^{-1})^{-1} B_j^T S_i^T$$  \hspace{1cm} (6.19)

where $i$ and $j$ are sampling level index and patch location index respectively. Each pre-computed $B_{i,j}'$ is of size $r_j \times n_i$ and it is computed for each sampling level leading to a total size of $r_j \times \sum_i n_i$, whereas the original $B_j$ is of size $M \times r_j$ ($M = 17 \times 17$ in this evaluation). For the evaluation example in previous section, the sampling levels are $n_i \in \{10, 30, 60, 100, 150\}$ which is a total size of $r_j \times 350$. In Table 6.2 the usage of memory bits is listed under various $q$. To make a reference, the usage of memory bits is compared with the total available block memory bits on a Stratix IV device [Alt12].

The required local buffer storage for prior knowledge is an inherent characteristic of domain-specific method. It can be seen that the required local buffer is much larger than what is needed to buffer the whole face image (0.11 Mb). This makes the proposed Domain-specific
CbIA most useful in situations where the generation and storage of the target face image to acquire is completely separate from the computing engine in question. In other cases where the target image to acquire is, for example, generated as an intermediate image data from the computing engine in question itself, it might be more beneficial to buffer the whole image directly in the local buffer.

### 6.6.2 The Computational Cost

In the following, a high-level estimation of the cost in time and energy consumption of running the proposed method is provided. This high-level estimation is based on the number of floating point operations (flops) during the executing of the proposed method. Although in practical implementation, the exact cost depends on the chosen hardware platform, the flops estimation provided in this section serves as a guideline of how much effort the proposed method takes to complete.

In figure 6.14, the normalized cost of running the domain-specific CbIA procedure for human faces is shown. The normalized cost has the same definition as in section 5.7:

\[
\text{norm\_cost} = \frac{\text{flops} \cdot c_{\text{flop}}}{(N - M) \cdot c_{\text{access}}} \tag{6.20}
\]

where \(c_{\text{flop}}\) and \(c_{\text{access}}\) are abstract cost per flop and per pixel access respectively (assumed to be equal); \(M\) is the amount of pixels sampled in current sampling pattern and \(N = 129 \times 113\) is the dimension of target images. Therefore the normalized cost measures that under the assumption of \(c_{\text{flop}} = c_{\text{access}}\), how much times the cost of running the proposed method is compared to the conventional image acquisition method which is to access every pixel of the target image. In other words, the normalized cost provides an estimation that in order for the proposed method to be reducing the overall cost of image acquisition, how much times does computing one flop has to be less costly than that of accessing the pixel directly from the memory.

From Figure 6.14 it can be seen that at PSNR of 36 dB, for the reference grid AFPS algorithm to be able to reduce the overall cost of the image acquisition process, the cost of computing one
Chapter 6. Domain Specific Image Acquisition of Face Images

Figure 6.14: Normalized cost of the proposed domain-specific CbIA procedure. The graph shows the proposed method with 400 training images with and without close examples. While higher energy threshold \( q \) leads to better PSNR vs. b/p ratio (figure 6.12), it also leads to more costly computation because the example space is more complex.

It is worth noting that while higher energy threshold \( q \) leads to better PSNR vs. b/p ratio (Figure 6.12), it also leads to more costly computation because the example space is more complex. Therefore in Figure 6.14 it can be seen that to achieve a same PSNR, algorithm with lower \( q \) is less costly. Compared with the reference grid AFPS, the proposed domain-specific CbIA procedure achieves better PSNR vs. b/p (figure 6.12) and at the same time has a comparable cost to PSNR ratio (Figure 6.14). This is resulted from the explicit use of learned prior knowledge from given examples of the same image class.
6.7 Conclusion

In this chapter the design of Domain-specific CbIA procedure for face images is proposed, which utilizes learned prior knowledge from example images of the same class as the target image is. The discussion is made on human faces as example and the evaluation shows that in the domain-specific scenario of faces, the CbIA procedure is able to achieve improved image quality vs. b/p ratio than reference point sampling algorithm while being less costly. It is of the future plan that the proposed method be extended to a broader scope of image classes, to investigate the effectiveness of Domain-specific CbIA in other applications.

Between the basic sampling procedure of adaptive refine adopted in chapter 4 and the complex modelling of KbAS proposed in chapter 5, the explicit use of prior knowledge is introduced in this chapter to explore the design space. In the presence of example face images, learned statistics can contribute to the performance of image point sampling and reconstruction tasks which are essential to the proposed CbIA procedure. In the next chapter, more details are provided regarding the design space of CbIA procedure which serve as a conclusion to the thesis.
Chapter 7

Conclusion

7.1 Summary

In this thesis, the concept of Context-based Image Acquisition (CbIA) is proposed as a novel approach that deals with the ever increasing presence of the cost of image acquisition process. The main idea behind CbIA is to propose such a hardware architecture that is able to dynamically and adaptively sample from a target image stored in source memory, and prepare a reconstructed approximation of the image inside the computing engine for potential client image processing application to use. The sampling nature of CbIA procedures essentially encourages the trading of image quality and computation power, with the bandwidth, time, and energy consumption of memory accessing. Due to the performance gap between computing engines and memory systems, CbIA architecture is expected to achieve a reduced overall cost of image acquisition process. Throughout the thesis, the major performance metrics of evaluating the image acquisition process are as established in Chapter 3:

1. **Image quality**: the quality of the acquired image data, which is often measured in PSNR against the ground truth image stored in the source memory. In the framework of CbIA methods, part of the image quality is traded for a reduced effort (the following three metrics) of image acquisition process.
2. **Acquisition time**: the overall time required between the acquisition order is issued and the requested image data is acquired from the source memory. This measures how much time it is going to take for the complete image processing task to finish.

3. **Energy consumption**: the overall energy required to perform the acquisition of target image data, including the energy consumption of both the source memory and the computing engine.

4. **Bandwidth**: the total amount of data (measured in bits) accessed from the source memory and transmitted to the computing engine in a fixed period of time. The bandwidth requirement here measures how frequent the source memory is occupied during the image acquisition process. Lower bandwidth requirement means the source memory can be freed to serve other potential hardware entities.

Among these metrics, the latter three are jointly denoted as the “cost” of image acquisition for simplicity.

In Chapter 3, the concept of CbIA framework is introduced and explained under a generic scenario set for custom hardware. Some pre-emptive discussion and analysis are given to establish a guideline for the detailed designs and discussions in later chapters.

In Chapter 4, a CbIA architecture is designed and evaluated on FPGA and structured ASIC devices. This CbIA architecture is based on a progressive image sampling algorithm, the Adaptive Refine, modified from existing ray tracing algorithm to adapt to DRAM characteristics. The implemented architecture shows its ability to significantly reduce the overall bandwidth, time, and energy consumption of the image acquisition process, at the expense of image quality and hardware resource.

Chapter 5 takes the discussion of CbIA procedure one step further, to investigate for the best PSNR vs. b/p performance without considering the overhead computational cost of the algorithm. The problem of image progressive sampling is revisited and the Kernel-based Adaptive Sampling (KbAS) is proposed as a generic image blind sampling procedure. The KbAS method is based on the construction of spatial equivalent kernel around pixels, and is evaluated to show
a superior PSNR vs. b/p performance than state-of-art reference sampling algorithm. The
discussion of KbAS method gives a bold estimation of what CbIA procedure can achieve, if the
overhead cost of the sampling algorithm itself is not a concern.

Finally in Chapter 6, the explicit use of learned prior knowledge in the task of image point
sampling is discussed. The Domain-specific CbIA procedure is proposed which, using human
faces as an example image class, learns the sampling locations as well as the reconstruction
eigenspace from a given set of image examples. Compared with state-of-art point sampling
algorithm, the proposed Domain-specific CbIA procedure is able to achieve a better PSNR vs.
b/p performance.

Both Chapter 5 and Chapter 6 are focused on the design of sampling procedure and reconstruc-
tion algorithm which, as analysed in Section 5.2 are key to the design of CbIA architecture. In
these two chapters, a brief discussion of the estimated cost of performing the proposed algo-
rithms is provided. The estimation of the cost is based on the floating point operations (flop)
count, which is able to describe the cost in an abstract level.

Under the framework of Context-based Image Acquisition, there are many possibilities of de-
signing the actual CbIA procedure. These different designs make different trade-offs among the
four major metrics of CbIA procedure. The proposed CbIA procedures in this thesis all have
their own advantages and disadvantages compared with others as well. Ultimately, it comes
down to the trade-off between image quality and computational cost.

As a conclusion to the whole thesis, an abstract-level evaluation of this trade-off is carried out
on all proposed CbIA procedures. This is to provide an overall discussion of the CbIA designs
involved in this thesis, as well as to answer to the preliminary analysis of CbIA in Chapter 3.
7.2 Analysis of the Proposed CbIA Concept

The abstract-level evaluation of the CbIA procedures is based on the flops counting, and the normalized cost estimation (which stands for time and energy consumption) below:

\[ \text{norm\_cost} = \frac{flops \cdot c_{flop}}{(N - M) \cdot c_{access}} \]  

(7.1)

where \( c_{flop} \) and \( c_{access} \) is the time/energy cost per flop and per pixel access respectively. In this evaluation, the two are assumed to be equal and therefore the final \( \text{norm\_cost} \) describes the ratio of the costs. It also represents that, for the CbIA procedure in question to be beneficial in reducing the time and energy consumption of the image acquisition, how many times the action of performing one flop has to be less costly than accessing a pixel directly from the source memory.

In order for the Domain-specific CbIA procedure proposed in Chapter 6 to be able to work, the tests are carried out on the FERET and ORL face database. The same test setting is used as is in Chapter 6. The results of the evaluation, which shows the trade-off among all four major metrics of CbIA, are shown in Figure 7.1.

Figure 7.1(a) shows the PSNR vs. b/p performance of the various sampling procedures, which is essentially the ability of the sampling procedures to trade image quality for reduced bandwidth. It can be seen that, both KbAS and Domain-specific CbIA procedure achieve a higher performance than reference AFPS, with the Adaptive Refine process adopted in the designed CbIA architecture being the lowest in this department. From Adaptive Refine, to grid AFPS, and finally to the proposed KbAS, the modelling of the target image structure is more and more accurate which leads to the improved image quality at the end of the process. On the other hand, because an example database that share the same class as the target image is provided, the Domain-specific CbIA achieve an even higher performance than that of the KbAS, being the best sampling procedure in terms of reducing bandwidth requirement.

The normalized cost of sampling procedures is shown in Figure 7.1(b). Essentially this graph shows the estimated time and energy cost of the various sampling procedure, compared with
Figure 7.1: The conclusive evaluation of various CbIA procedures involved in the thesis, as well as the reference grid AFPS algorithm. (a) This graph shows the PSNR vs. b/p performance of the various sampling procedures, which shows the ability of the sampling procedures to trade image quality for reduced bandwidth. (b) This graph is the normalized cost of sampling procedures, which abstracts their time and energy consumption.

The conventional method of accessing every pixel from the target image. It can be seen that, while KbAS can achieve a higher PSNR vs. b/p ratio, its cost is also higher than that of the grid AFPS and Adaptive Refine. The Adaptive Refine in particular, is of the lowest cost due to its simplicity. Finally, the Domain-specific CbIA is able to maintain a cost similar to that of the grid AFPS while achieving the highest PSNR vs. b/p ratio, demonstrating the benefit of using the learned prior knowledge. In detail, at PSNR of 34 dB, for the Adaptive Refine procedure to be able to reduce the overall cost in time and energy of the image acquisition process, the cost of performing one flop has to be $10^{1.9} \approx 80$ times less than the cost of accessing one pixel directly from the source memory. For the grid AFPS, the number is $10^{2.8} \approx 631$; for the Domain-specific CbIA with selected settings, the number changes to $10^{2.6} \approx 398$; for the KbAS procedure using op1, the number changes to $10^{4.4} \approx 25119$.

To have a better interpretation of the results in Figure 7.1(b), two references can be used. Firstly in Chapter 3, the rough evaluation shows that on the platform of Hardcopy IV structured-ASIC, accessing one pixel from a DDR3 memory equals to performing about 250 ADD or 64 MULT operations in energy consumption. Secondly, as is demonstrated in Chapter 4, when
the Adaptive Refine procedure is adopted the CbIA architecture on Hardcopy IV is capable of reducing the overall time and energy consumption of the image acquisition process by over 50%. Based on this reference, a rough estimation can be made of when these sampling procedures will be beneficial to be applied in CbIA architecture. For example, if the CbIA architecture with Adaptive Refine procedure designed in Chapter 4 is considered to be of equal cost with conventional image accessing procedure, then for the Domain-specific CbIA procedure to be able to reduce the overall cost in time and energy consumption the computational ability\(^1\) of the computing engine has to be roughly \(398 \div 80 \approx 5\) times that of the Stratix IV device.

While the performance of CbIA procedures is application dependent as well as implementation dependent, the above evaluations in Figure 7.1(b) show a rough estimation of at what conditions the designed procedures are going to be beneficial in reducing time and energy consumption of the image acquisition process.

### 7.3 Potential of Context-based Image Acquisition

By utilizing the contextual information contained in natural images, the proposed Context-based Image Acquisition is essentially a new way of studying the process of image acquisition in hardware systems.

As is demonstrated, the design of CbIA architecture is first and foremost fully capable of reducing the memory bandwidth requirement at the expense of some loss of image quality. Moreover, with the Adaptive Refine shown to be capable of achieving a reduced time and energy consumption in the implemented architecture of CbIA, the potential of the proposed framework in this department is going to be gradually reached as the cost of image access increases, the performance gap between computing engine and memory systems grow bigger.

This project hopes to provide such a novel perspective of thinking, that is to change the conventional architecture of hardware systems and incorporate the knowledge developed in the

\(^1\)The computational ability here refers to the ability of the computing engine to perform computations given a fixed amount of time or energy resource.
field of image processing. The proposed concept of CbIA can also be expanded to fields like video processing, and be designed to be directly compatible with existing methods such as frame re-compression. Additionally, techniques can be developed to let the CbIA sampling procedure take into consideration of the time/energy cost, and achieve better performance of cost vs. achieved PSNR.

All in all, it is the wish of this work that the proposed concept of CbIA can bring inspirations to the field of hardware architecture development.

7.4 Future Work

In this section, future work of this project is discussed.

7.4.1 Short Term Plan: Further Investigations and Modifications

Based on the proposed CbIA framework and designed hardware architecture/algorithms, there are various further investigations and possible modifications that can be made.

Firstly from the hardware perspective, the discussion of the proposed CbIA concept can be taken one step further out of the reconfigurable hardware platforms. In Chapter 4 a hardware architecture of CbIA is designed and evaluated on reconfigurable hardware platforms. As is mentioned in section 4.5.3, ASIC implementation of the proposed architecture is favourable. It is the aim of the proposed CbIA that an ASIC architecture be designed and implemented, serving as an embedded image acquisition subsystem in an otherwise large image processing system to acquire image data for computing engines. It is expected that the ASIC implementation of the proposed architecture can show improved performance than what is reported in the evaluations on FPGA and structured-ASIC in Chapter 4. Performance evaluations of the actual ASIC design of the CbIA framework can provide concrete data to establish a solid reference for the future development of the framework.
Besides the design of CbIA on custom hardware, it is also of great interest of this project to port the design to other hardware platforms such as general purposed processors. The proposed concept of CbIA can be designed and implemented on general purposed processors as generic data acquisition procedures for the computing kernels on the chip, utilizing the available on-chip memory bandwidth and data bus \[RKC13\]. Applying CbIA on these processors such as CPUs would pose further challenges as there are often existing data transmission mechanisms in place, caching scheme of CPUs being an example. Investigating on the possibility of designing a mechanism for the proposed CbIA procedure to co-operate with existing memory hierarchy in these general purposed processors will be a main objective of the future research, as it will increase the range of application of the proposed framework.

Secondly from the algorithmic perspective, the proposed CbIA procedure can be extended from pixel-based sampling, to generic block-based sampling. In this thesis, the discussion of CbIA procedure is focused on pixel-based accessing of data from the source memory. This is to stay true to the basic memory accessing protocol and establish a solid application of the proposed concept. However it is possible to extend the procedure to other types of accessing, which is given the name of “generic block-based accessing” in this case. This can enhance the usefulness of the CbIA procedure, allowing it to be compatible with existing data accessing/manipulation techniques such as frame re-compression, which is introduced in Chapter 2.

One example of the generic block-based accessing is in fact given in Chapter 4 where the designed CbIA architecture works with the pre-fetching functionality of SDRAMs. In this example (Figure 7.2(b)), the designed CbIA architecture simply accepts the additionally pre-fetched pixels from the source memory and put them into the reconstruction of the macroblock. The sampling and reconstruction algorithms are not tailored to specifically account for these extra pixel information. In the future, it is worth investigating on how to design sampling and reconstruction algorithms that are block-based (e.g. Figure 7.2(c)). In general, it is the requirement of the proposed CbIA concept that the source memory and its stored target image must have a certain level of random accessibility. The word “certain” means that it is possible to locate and retrieve a local part/region of the image data out of the stored target image. To find a way to do generic block-based accessing is to allow CbIA procedures account for more
Figure 7.2: The proposed CbIA can be extended from pixel-based access to generic block-based access. This allows the proposed concept of CbIA to be compatible with existing methods such as frame re-compression (e.g. the work of Lee et al. [Lee03, LRL07]). In this figure, sampled pixels by pixel-based CbIA are marked in blue. (a) The current pixel-based CbIA procedures; (b) it is already shown in Chapter 4 that CbIA procedure can adapt to the pre-fetching of memory devices (each sample pixel followed by a burst of pixels, marked in green); (c) the proposed CbIA concept can be extended to generic block-based accessing, which in this example is a $2 \times 2$ block.

types of image storage in hardware system, an example of which being frame re-compression in the work of Lee et al. [LRL07] compresses image data by $4 \times 4$ pixel blocks.

The above are potential short term plans for future research of the topic. These are immediate modifications and/or developments of the work reported in this thesis, and they are to enhance the usefulness of the proposed designs.

### 7.4.2 Long Term Plan: Future Directions of the Work

One of the potential long term directions of this project is the design of dynamic priority estimation, to replace the fixed priority estimation explained in earlier chapters. It can be seen that the proposed CbIA procedures in the thesis priorities pixels by the estimated amount of information the candidate pixel can bring, which improves the reconstruction quality at the end of the process. Such fixed priority estimation can be replaced by a more dynamic mechanism of priority estimation which takes into account not only the reconstruction quality, but also the time and energy consumption of accessing/processing candidate pixels. Weights can be added to the estimated gain and cost each candidate pixel, to form a more informative priority
score. By allowing the user to input the weights of the various performance metrics, the CbIA procedure can better reflect the user requirements. Moreover, run-time evaluations of currently available resources (bandwidth, time, energy) can be done on-line, giving the CbIA procedure the ability to automatically adjust the weights of performance metrics and better adapt to the current hardware environment.

For Domain-specific CbIA, it is of great interests to further investigate on other image classes. The proposed Domain-specific CbIA in this thesis focuses on utilizing the distinct features and structures of face images. Other image classes can pose different challenges such as the problem of mis-alignment, and therefore different sampling/reconstruction methods should be designed. Learning from different images has been a popular topic in the literature of image classification, which also motivates the investigation of applying Domain-specific CbIA procedure on other image classes. The class of face images can also benefit from the investigation on learning methods/dictionary building methods. On top of this, it is also potential to design the Domain-specific CbIA to learn from processed images in an on-line fashion. There have been successful on-line learning of images in the literature of object detection (e.g. the work of Kirstein et al. [KWKOS]), which provides inspirations to the design of CbIA procedures.

Finally, based on the outcome of the above short term and long term research plans, it is the long term vision of this work that an automated tool of CbIA module generation can be designed, providing a complete solution to the task of image acquisition in hardware systems. As is shown in Figure 7.3, the generated CbIA module can be either an IP core for reconfigurable hardware, a design of ASIC architecture, or a procedure for GPUs/CPUs. The main body of the automated tool is the “CbIA Module Generator” which takes in parameters including the size of the requested macroblock, initial sampling pattern, the sampling progression (e.g. how threshold of quality is going to progress) etc.. These parameters, which ultimately determines the output CbIA module, can also be automatically generated by the “Task Analyser”. The Task Analyser takes in a collection of example images, if available, and uses methods such as cross validation to optimize the parameters of the CbIA module. These example images can be domain-specific such as face images to allow the task analyser to extract explicit prior knowledge, or more generic images from which the Task Analyser can cross validate to determine
Figure 7.3: The automated CbIA module generation tool takes into account the task in question, analyse the example images, and generate the CbIA module according to the requirements set by the user.

the more basic parameters such as the initial sampling pattern. For both the CbIA Module Generator and the Task Analyser, the user can input specific requirements such as the weights of various performance metrics and target performance thresholds etc..

Ultimately, it is the aim of the proposed CbIA framework to offer such an automated tool that provides complete solutions to the design of CbIA hardware architecture for the task of image acquisition.
Bibliography


[AMD03] AMD. 128 Megabit (8 M x 16-Bit/16 M x 8-Bit) MirrorBit 3.0 Volt-only Uniform Sector Flash Memory with VersatileI/O Control, Feb 2003.


[Int05] Integrated Silicon Solution, Inc. 256K x 72, 512K x 36, 1024K x 18 18Mb SYNCHRONOUS PIPELINED, SINGLE CYCLE DESELECT STATIC RAM, Feb 2005.


