DIGITAL CAMERA SYSTEM SIMULATOR AND APPLICATIONS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Ting Chen

June 2003

© Copyright by Ting Chen 2003

All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Abbas El Gamal
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Robert M. Gray

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Brian A. Wandell

Approved for the University Committee on Graduate Studies:

Abstract

Digital cameras are rapidly replacing traditional analog and film cameras. Despite their remarkable success in the market, most digital cameras today still lag film cameras in image quality and major efforts are being made to improve their performance. Since digital cameras are complex systems combining optics, device physics, circuits, image processing, and imaging science, it is difficult to assess and compare their performance analytically. Moreover, prototyping digital cameras for the purpose of exploring design tradeoffs can be prohibitively expensive. To address this problem, a digital camera simulator - vCam - has been developed and used to explore camera system design tradeoffs. This dissertation is aimed at providing a detailed description of vCam and demonstrating its applications with several design studies.

The thesis consists of three main parts. vCam is introduced in the first part. The simulator provides physical models for the scene, the imaging optics and the image sensor. It is written as a MATLAB toolbox and its modular nature makes future modifications and extensions straightforward. Correlation of vCam with real experiments is also discussed. In the second part, to demonstrate the use of the simulator, the application that relies on vCam to select optimal pixel size as part of an image sensor design is presented. In order to set up the design problem, the tradeoff between sensor dynamic range and spatial resolution as a function of pixel size is discussed. Then a methodology using vCam, synthetic contrast sensitivity function scenes, and the image quality metric S-CIELAB for determining optimal pixel size is introduced. The methodology is demonstrated for active pixel sensors implemented in CMOS processes down to 0.18um technology. In the third part of this thesis vCam

is used to demonstrate algorithms for scheduling multiple captures in a high dynamic range imaging system. In particular, capture time scheduling is formulated as an optimization problem where average signal-to-noise ratio (SNR) is maximized for a given scene probability density function (pdf). For a uniform scene pdf, the average SNR is a concave function in capture times and thus the global optimum can be found using well-known convex optimization techniques. For a general piece-wise uniform pdf, the average SNR is not necessarily concave, but rather a difference of convex (D.C.) function and can be solved using D.C. optimization techniques. A very simple heuristic algorithm is described and shown to produce results that are very close to optimal. These theoretical results are then demonstrated on real images using vCam and an experimental high speed imaging system.

Acknowledgments

I am deeply indebted to many people who made my Stanford years an enlightening, rewarding and memorable experience.

First of all, I want to thank my advisor Professor El Gamal. It has been truly a great pleasure and honor to work with him. Throughout my PhD study, he gave me great guidance and support. All these work would not have been possible without his help. I have benefited greatly from his vast technical expertise and insight, as well as his high standards in research and publication.

I am grateful to Professor Gray, my associate advisor. I started my PhD study by working on a quantization project and Professor Gray was generous to offer his help by becoming my associate advisor. Even though the quantization project did not become my thesis topic, I'm very grateful that he is very understanding and would still support me by serving on my orals committee and thesis reading committee.

I would also like to thank Professor Wandell. He also worked on the programmable digital camera project with our group. I was very fortunate to be able to work with him. Much of my research was done directly under his guidance. I still remember the times when Professor Wandell and I were sitting in front of a computer and hacking on the codes for the camera simulator. It is an experience that I will never forget.

I want to thank Professor Mark Levoy. It is a great honor to have him as my oral chair. I also want to thank Professor John Cioffi, Professor John Gill, and Professor Joseph Goodman for their help and guidance.

I gratefully appreciate the support and encouragement from Dr. Boyd Fowler and Dr. Michael Godfrey.

I gratefully acknowledge my former officemates Dr. David Yang, Dr. Hui Tian, Dr. Stuart Kleinfelder, Dr. Xinqiao Liu, Dr. Sukhwan Lim, and current officemates Khaled Salama, Helmy Eltoukhy, Ali Ercan, Sam Kavusi, Hossein Kakavand and Sina Zahedi, and group-mates Peter Catrysse, Jeffery DiCarlo and Feng Xiao for their collaboration and many interesting discussions we had over the years. Special thanks go to Peter Catrysse with whom I collaborated in many of our research projects.

I would also like to thank our administrative assistants, Charlotte Coe, Kelly Yilmaz and Denise Murphy for all their help.

I also like to thank the sponsors of programmable digital camera (PDC) project, Agilent Technologies, Canon, Hewlett-Packard, Kodak, and Interval Research, for their financial support.

I would also like to thank all my friends for their encouragements and generous help.

Last but not the least, I am deeply indebted to my family and my wife Ami. Without their love and support, I could not have possibly reached at this stage today. My appreciation for them is very hard to be described precisely in words, but I am confident they all understand my feelings for them because they have alway been so understanding. This thesis is dedicated to them.

Contents

Abstract **iv**

Acknowledgments **vi**

1 Introduction **1**

1.1 Digital Camera Basics 1

1.2 Solid State Image Sensors 4

1.2.1 CCD Image Sensors 5

1.2.2 CMOS Image Sensors 8

1.3 Challenges in Digital Camera System Design 11

1.4 Author's Contribution 13

1.5 Thesis Organization 14

2 vCam - A Digital Camera Simulator **15**

2.1 Introduction 15

2.2 Physical Models 16

2.2.1 Optical Pipeline 17

2.2.2 Electrical Pipeline 28

2.3 Software Implementation 41

2.3.1 Scene 41

2.3.2 Optics 43
2.3.3 Sensor 44
2.3.4 From Scene to Image 47
2.3.5 ADC, Post-processing and Image Quality Evaluation 47
2.4 vCam Validation 51
2.4.1 Validation Setup 51
2.4.2 Validation Results 53
2.5 Conclusion 53

3 Optimal Pixel Size **56**
3.1 Introduction 56
3.2 Pixel Performance, Sensor Spatial Resolution and Pixel Size 58
3.2.1 Dynamic Range, SNR and Pixel Size 59
3.2.2 Spatial Resolution, System MTF and Pixel Size 60
3.3 Methodology 61
3.4 Simulation Parameters and Assumptions 64
3.5 Simulation Results 66
3.5.1 Effect of Dark Current Density on Pixel Size 68
3.5.2 Effect of Illumination Level on Pixel Size 70
3.5.3 Effect of Vignetting on Pixel Size 72
3.5.4 Effect of Microlens on Pixel Size 73
3.6 Effect of Technology Scaling on Pixel Size 75
3.7 Conclusion 75

4 Optimal Capture Times **78**
4.1 Introduction 78
4.2 Problem Formulation 80
4.3 Optimal Scheduling for Uniform PDF 83
4.4 Scheduling for Piece-Wise Uniform PDF 84
4.4.1 Heuristic Scheduling Algorithm 91
4.5 Piece-wise Uniform PDF Approximations 92

4.5.1 Iterative Histogram Binning Algorithm 93
4.5.2 Choosing Number of Segments in the Approximation 95
4.6 Simulation and Experimental Results 95
4.7 Conclusion 99

5 Conclusion **103**
5.1 Summary 103
5.2 Future Work and Future Directions 104

Bibliography **106**

List of Tables

2.1 Scene structure 42
2.2 Optics structure 44
2.3 Pixel structure 46
2.4 ISA structure 48

4.1 Optimal capture time schedules for a uniform pdf over interval $(0, 1]$. 85

List of Figures

1.1 A typical digital camera system 2
1.2 A CCD Camera requires many chips such as CCD, ADC, ASICs and memory 3
1.3 A single chip camera from Vision Ltd. [75] Sub-micron CMOS enables camera-on-chip 3
1.4 Photocurrent generation in a reverse biased photodiode 5
1.5 Block diagram of a typical interline transfer CCD image sensor 6
1.6 Potential wells and timing diagram during the transfer of charge in a three-phase CCD 7
1.7 Block diagram of a CMOS image sensors 9
1.8 Passive pixel sensor (PPS) 10
1.9 Active Pixel Sensor (APS) 11
1.10 Digital Pixel Sensor (DPS) 12

2.1 Digital still camera system imaging pipeline - How the signal flows . . 17
2.2 vCam optical pipeline 18
2.3 Source-Reciever geometry 20
2.4 Defining solid angle 21
2.5 Perpendicular solid angle geometry 23
2.6 Imaging geometry 24
2.7 Imaging law and $f/\#$ of the optics 26
2.8 Off-axis geometry 27

2.9 vCam noise model 32
2.10 Cross-section of the tunnel of a DPS pixel leading to the photodiode . 34
2.11 The illuminated region at the photodiode is reduced to the overlap between the photodiode area and the area formed by the projection of the square opening in the 4th metal layer 36
2.12 Ray diagram showing the imaging lens and the pixel as used in the uniformly illuminated surface imaging model. The overlap between the illuminated area and the photodiode area is shown for on and off-axis pixels 37
2.13 An n-diffusion/p-substrate photodiode cross sectional view 38
2.14 CMOS active pixel sensor schematics 40
2.15 A color filter array (CFA) example - Bayer pattern 49
2.16 An Post-processing Example 50
2.17 vCam validation setup 52
2.18 Sensor test structure schematics 53
2.19 Validation results: histogram of the % error between vCam estimation and experiments 54

3.1 APS circuit and sample pixel layout 58
3.2 (a) DR and SNR (at 20% well capacity) as a function of pixel size. (b) Sensor MTF (with spatial frequency normalized to the Nyquist frequency for 6μm pixel size) is plotted assuming different pixel sizes. 60
3.3 Varying pixel size for a fixed die size 62
3.4 A synthetic contrast sensitivity function scene 62
3.5 Sensor capacitance, fill factor, dark current density and spectral response information 65
3.6 Simulation result for a 0.35μ process with pixel size of 8μm. For the ΔE error map, brighter means larger error 67
3.7 Iso-$\Delta E = 3$ curves for different pixel sizes 69
3.8 Average ΔE versus pixel size 69
3.9 Average ΔE vs. Pixel size for different dark current density levels . . 70

3.10 Average ΔE vs. Pixel size for different illumination levels 71
3.11 Effect of pixel vignetting on pixel size 73
3.12 Different pixel sizes suffer from different QE reduction due to pixel vignetting. The effective QE, *i.e.*, normalized with the QE without pixel vignetting, for pixels along the chip diagonal is shown. The X-axis is the horizontal position of each pixel with origin taken at the center pixel. 74
3.13 Effect of microlens on pixel size 75
3.14 Average ΔE versus pixel size as technology scales 76
3.15 Optimal pixel size versus technology 76

4.1 (a) Photodiode pixel model, and (b) Photocharge $Q(t)$ vs Time t under two different illuminations. Assuming multiple capture at uniform capture times $\tau, 2\tau, \ldots, T$ and using the LSBS algorithm, the sample at T is used for the low illumination case, while the sample at 3τ is used for the high illumination case. 81
4.2 Photocurrent pdf showing capture times and corresponding maximum non-saturating photocurrents. 83
4.3 Performance comparison of optimal schedule, uniform schedule, and exponential (with exponent = 2) schedule. $E\,(\text{SNR})$ is normalized with respect to the single capture case with $i_1 = i_{\max}$. 86
4.4 An image with approximated two-segment piece-wise uniform pdf . . 87
4.5 An image with approximated three-segment piece-wise uniform pdf . 87
4.6 Performance comparison of the Optimal, Heuristic, Uniform, and Exponential (with exponent = 2) schedule for the scene in Figures 4.4. $E\,(\text{SNR})$ is normalized with respect to the single capture case with $i_1 = i_{\max}$. 89
4.7 Performance comparison of the Optimal, Heuristic, Uniform, and Exponential (with exponent = 2) schedule for the scene in Figures 4.5. $E\,(\text{SNR})$ is normalized with respect to the single capture case with $i_1 = i_{\max}$. 90

4.8 An example for illustrating the heuristic capture time scheduling algorithm with $M = 2$ and $N = 6$. $\{t_1, \ldots, t_6\}$ are the capture times corresponding to $\{i_1, \ldots, i_6\}$ as determined by the heuristic scheduling algorithm. For comparison, optimal $\{i_1, \ldots, i_6\}$ are indicated with circles. 92
4.9 An example that shows how the Iterative Histogram Binning Algorithm works. A histogram of 7 segments is approximated to 3 segments with 4 iterations. Each iteration merges two adjacent bins and therefore reduces the number of segments by one. 94
4.10 E[SNR] versus the number of segments used in the pdf approximation for a 20-capture scheme on the image shown in Figure 4.5. E[SNR] is normalized to the single capture case. 96
4.11 Simulation result on a real image from vCam. A small region, as indicated by the square in the original scene, is zoomed in for better visual effects 98
4.12 Noise images and their histograms for the three capture schemes . . . 99
4.13 Experimental results. The top-left image is the scene to be captured. The white rectangle indicates the zoomed area shown in the other three images. The top-right image is from a single capture at 5ms. The bottom-left image is reconstructed using LSBS algorithm from optimal captures taken at $5, 15, 30$ and 200ms. The bottom-right image is reconstructed using LSBS algorithm from uniform captures taken at $5, 67, 133$ and 200ms. Due to the large constrast in the scene, all images are displayed in $\log 10$ scale. 100

Chapter 1

Introduction

1.1 Digital Camera Basics

Fueled by the demands of multimedia applications, digital still and video cameras are rapidly becoming widespread. As image acquisition devices, digital cameras are not only replacing traditional film and analog cameras for image captures, they are also enabling many new applications such as PC cameras, digital cameras integrated into cell phones and PDAs, toys, biometrics, and camera networks. Figure 1.1 is a block diagram of a typical digital camera system. In this figure, a scene is focused by a lens through a color filter array onto an image sensor which converts light into electronic signals. The electronic output then goes through analog signal processing such as correlated double sampling (CDS), automatic gain control (AGC), analog-to-digital conversion (ADC), and a significant amount of digital processing for color, image enhancement and compression.

The image sensor plays a pivotal role in the final image quality. Most digital cameras today use charge-coupled device (CCD) image sensors. In these types of devices, the electric charge collected by the photodetector array during exposure time is serially shifted out of the sensor chip, thus resulting in slow readout speed

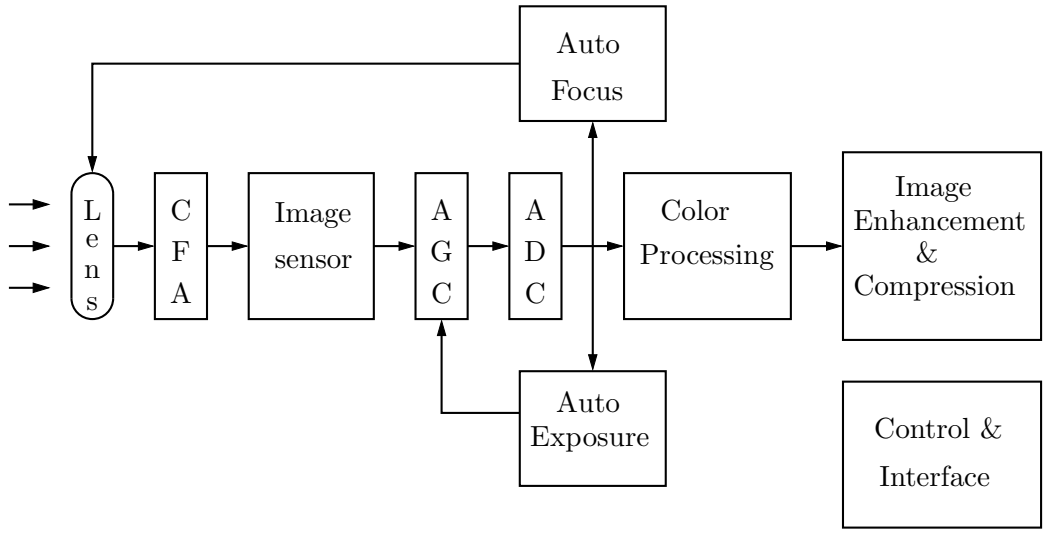


Figure 1.1: A typical digital camera system

and high power consumption. CCDs are fabricated using a specialized process with optimized photodetectors. To their advantages, CCDs have very low noise and good uniformity. It is not feasible, however, to use the CCD process to integrate other camera functions, such as clock drivers, time logic and signal processing. These functions are normally implemented in other chips. Thus most CCD cameras comprise several chips. Figure 1.2 is a photo of a commercial CCD video camera. It consists of two boards and both the front and back view of each board are shown. The CCD image sensor chip needs support from a clock driver chip, an ADC chip, a microcomputer chip, an ASIC chip and many others.

Recently developed CMOS image sensors, by comparison, are read out in a manner similar to digital memory and can be operated at very high frame rates. Moreover, CMOS technology holds out the promise of integrating image sensing and image processing into a single-chip digital camera with compact size, low power consumption and additional functionality. A photomicrograph of a commercial single chip CMOS camera is shown in Figure 1.3. On the downside, however, CMOS image sensors generally suffer from high read noise, high fixed pattern noise and inferior photodetectors due to imperfections in CMOS processes.

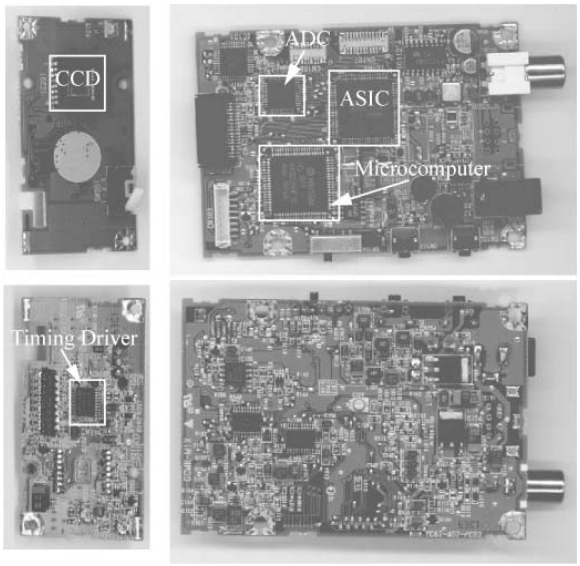


Figure 1.2: A CCD Camera requires many chips such as CCD, ADC, ASICs and memory

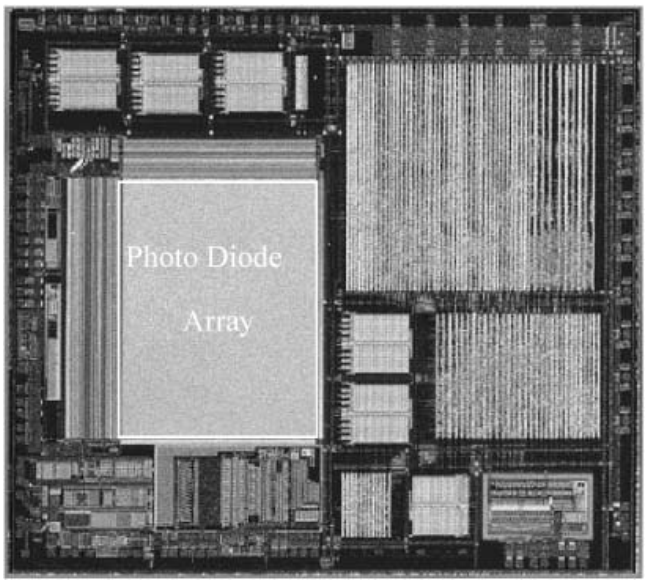


Figure 1.3: A single chip camera from Vision Ltd. [75] Sub-micron CMOS enables camera-on-chip

An image sensor is at the core of any digital camera system. For that reason, let us quickly go over the basic characteristics of solid state image sensors and the architectures of commonly used CCD and CMOS sensors.

1.2 Solid State Image Sensors

The image capturing devices in digital cameras are all solid state image sensors. An image sensor array consists of $n \times m$ pixels, ranging from 320×240 (QVGA) to 7000×9000 (very high end scientific applications). Each pixel contains a photodetector and circuits for reading out the electrical signal. The pixel size ranges from 15μm$\times 15\mu$m down to 3μm$\times 3\mu$m, where the minimum pixel size is typically limited by dynamic range and cost of optics.

The photodetector [59] converts incident radiant power into photocurrent that is proportional to the radiant power. There are several types of photodetectors, the most commonly used is the photodiode, which is a reverse biased pn junction, and the photogate, which is an MOS capacitor. Figure 1.4 shows the photocurrent generation in a reverse biased photodiode [84]. The photocurrent, i_{ph}, is the sum of three components: i) current due to generation in depletion (space charge) region, i_{ph}^{sc} — almost all carriers generated are swept away by strong electric field; ii) current due to holes generated in n-type quasi-neutral region, i_{ph}^{p}— some diffuse to space charge region and get collected; iii) current due to electrons generated in p-type region, i_{ph}^{n}. Therefore, the total photo-generated current is

$$i_{ph} = i_{ph}^{sc} + i_{ph}^{p} + i_{ph}^{n}.$$

The detector spectral response $\eta(\lambda)$ is the fraction of photon flux that contributes to photocurrent as a function of the light wavelength λ, and the quantum efficiency (QE) is the maximum spectral response over λ.

The photodetector dark current i_{dc} is the detector leakage current, *i.e.*, current not induced by photogeneration. It is called "dark current" since it corresponds to the

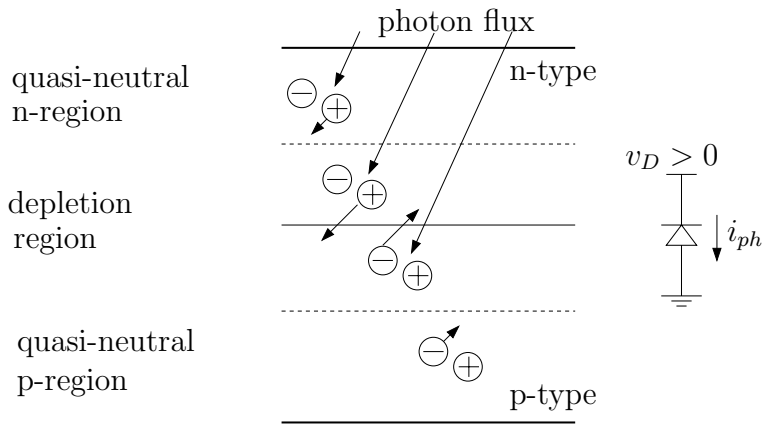


Figure 1.4: Photocurrent generation in a reverse biased photodiode

photocurrent under no illumination. Dark current is caused by the defects in silicon, which include bulk defects, interface defects and surface defects. Dark current limits the photodetector dynamic range because it reduces the signal swing and introduces shot noise.

Since the photocurrent is very small, normally on the order of tens to hundreds of fA, it is typically integrated into charge and the accumulated charge (or converted voltage) is then read out. This type of operation is called direct integration, the most commonly used mode of operation in an image sensor. Under direct integration, the photodiode is reset to the reverse bias voltage at the start of the image capture exposure time, or integration time. The diode current is integrated on the diode parasitic capacitance during integration and the accumulated charge or voltage is read out at the end via the help of readout circuitry. Different types of image sensors have very different readout architectures. We will go over some of the most commonly used image sensors next.

1.2.1 CCD Image Sensors

CCD image sensors [86] are the most widely used solid state image sensors in today's digital cameras. In CCDs, the integrated charge on the photodetector is read out

using capacitors. Figure 1.5 depicts the block diagram of the widely used interline transfer CCD image sensors. It consists of an array of photodetectors and vertical and horizontal CCDs for readout. During exposure, the charge is integrated in each photodetector, and it is simultaneously transferred to vertical CCDs at the end of exposure for all the pixels. The charge is then sequentially read out through the vertical and horizontal CCDs by charge transfer.

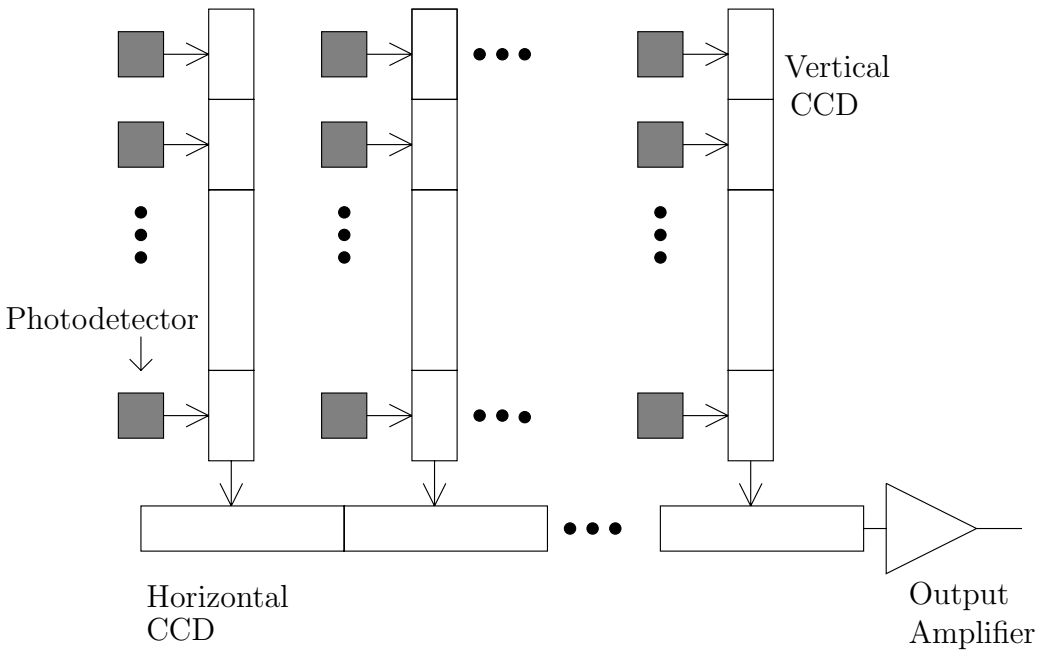


Figure 1.5: Block diagram of a typical interline transfer CCD image sensor

A CCD is a dynamic charge shift register implemented using closely spaced MOS capacitors. The MOS capacitors are typically clocked using 2, 3, or 4 phase clocks. Figure 1.6 shows a 3-phase CCD example where ϕ_1,ϕ_2 and ϕ_3 represent the three clocks. The capacitors operate in deep depletion regime when the clock voltage is high. Charge is transferred from one capacitor whose clock voltage is switching from high to low, to the next capacitor whose clock voltage is switching from low to high at the same time. During this transfer process, most of the charge is transferred very quickly by repulsive force among electrons, which creates self-induced lateral drift, the remaining charge is transferred slowly by thermal diffusion and fringing fields.

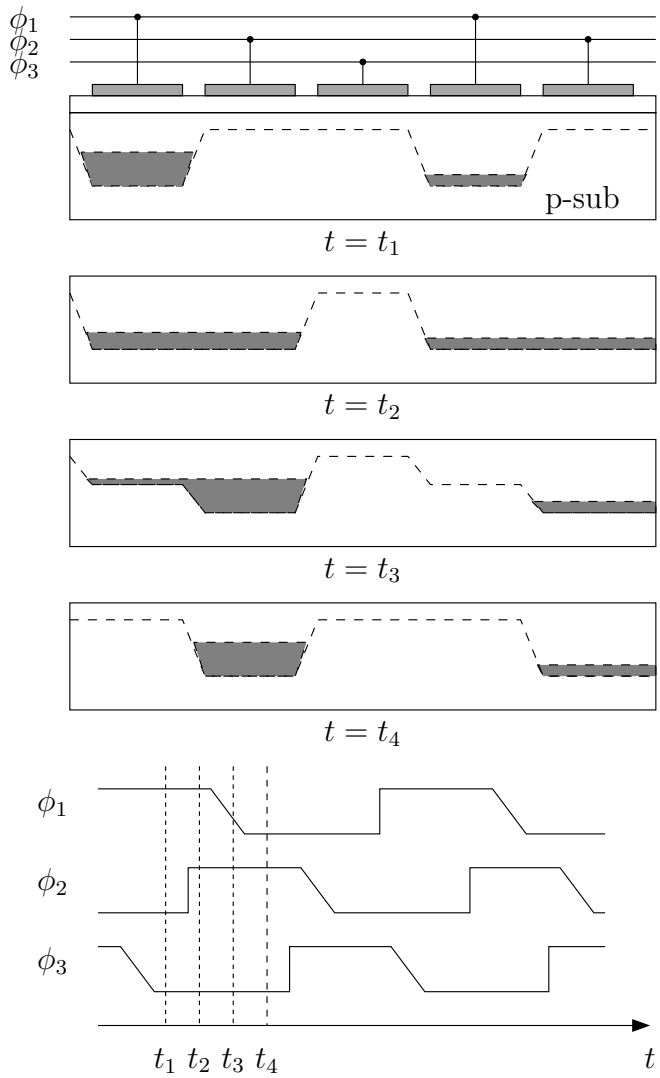


Figure 1.6: Potential wells and timing diagram during the transfer of charge in a three-phase CCD

The charge transfer efficiency describes the fraction of signal charge transferred from one CCD stage to the next. It must be made very high (≈ 1) since in a CCD image sensor charge is transferred up to $n+m$ CCD stages for an $m \times n$ pixel sensor. The charge transfer must occur at high enough rate to avoid corruption by leakage, but slow enough to ensure high charge transfer efficiency. Therefore, CCD image sensor readout speed is limited mainly by the array size and the charge transfer efficiency requirement. As an example, the maximum video frame rate for an 1024×1024 interline transfer CCD image sensor is less than 25 frames/s given a 0.99997 transfer efficiency requirement and 4μm center to center capacitor spacing[1].

The biggest advantage of CCDs is their high quality. They are fabricated using specialized processes [86] with optimized photodetectors, very low noise, and very good uniformity. The photodetectors have high quantum efficiency and low dark current. No noise is introduced during charge transfer. The disadvantages of CCDs include: i) they can not be integrated with other analog or digital circuits such as clock generation, control and A/D conversion; ii) they have very limited programmability; iii) they have very high power consumption because the entire array is switching at high speed all the time; iv) they have limited frame rate, especially for large sensors due to the required increase in transfer speed while maintaining an acceptable transfer efficiency.

1.2.2 CMOS Image Sensors

CMOS image sensors [65, 93, 72, 61] are fabricated using standard CMOS processes with no or minor modifications. Each pixel in the array is addressed through a horizontal word line and the charge or voltage signal is read out through a vertical bit line. The readout is done by transferring one row at a time to the column storage capacitors, then reading out the row using the column decoders and multiplexers. This readout method is similar to a memory structure. Figure 1.7 shows a typical CMOS image sensor architecture. There are three commonly seen pixel architectures: passive pixel sensor (PPS), active pixel sensor (APS) and digital pixel sensor (DPS).

[1]For more details, please refer to [1]

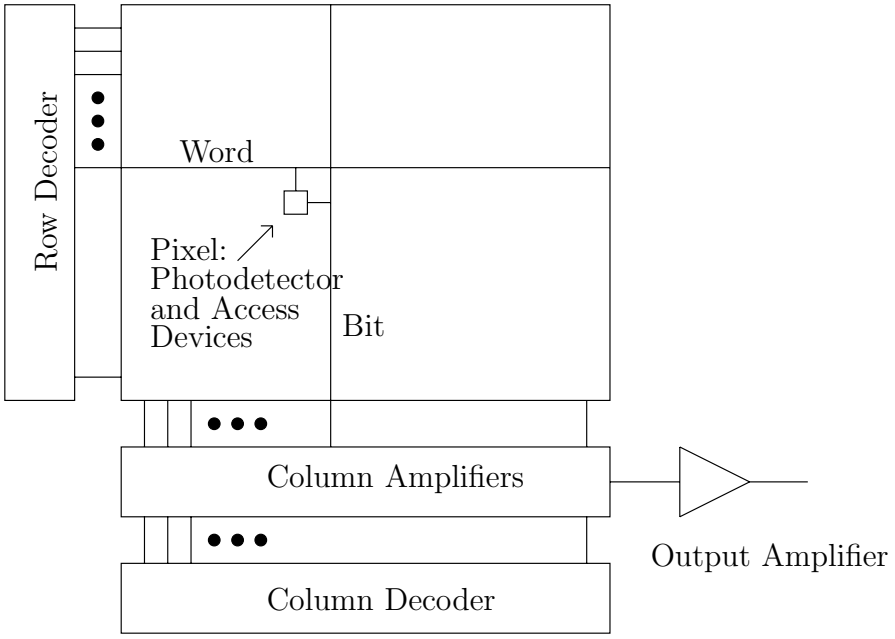


Figure 1.7: Block diagram of a CMOS image sensors

CMOS Passive Pixel Sensors

A PPS [23, 24, 25, 26, 42, 45, 39] has only one transistor per pixel, as shown in Figure 1.8. The charge signal in each pixel is read out via a column charge amplifier, and this readout is destructive as in the case of a CCD. A PPS has small pixel size and large fill factor[2], but it suffers from slow readout speed and low SNR. PPS readout time is limited by the time of transferring a row to the output of the charge amplifiers.

CMOS Active Pixel Sensors

An APS [94, 29, 67, 78, 66, 64, 100, 33, 34, 27, 49, 98, 108, 79, 17] normally has three or four transistors per pixel, where one transistor works as a buffer and an amplifier. As shown in Figure 1.9, the output of the photodiode is buffered using a pixel level follower amplifier. The output signal is typically in voltage and the reading is not destructive. In comparison to a PPS, an APS has a larger pixel size and a lower fill

[2]fill factor is the fraction of the pixel area occupied by the photodetector

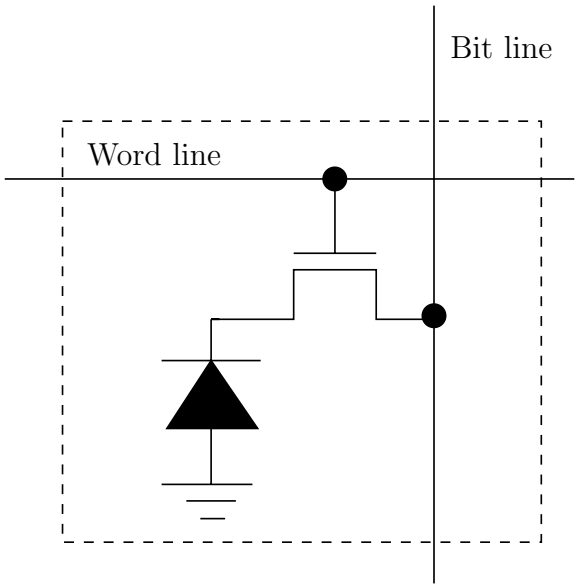


Figure 1.8: Passive pixel sensor (PPS)

factor, but its readout is faster and it also has higher SNR.

CMOS Digital Pixel Sensors

In a DPS [2, 36, 37, 107, 106, 103, 104, 105, 53], each pixel has an ADC. All ADCs operate in parallel, and digital data stored in the memory are directly read out of the image sensor array as in a conventional digital memory (see Figure 1.10). The DPS architecture offers several advantages over analog image sensors such as APSs. These include better scaling with CMOS technology due to reduced analog circuit performance demands and the elimination of read related column fixed-pattern noise (FPN) and column readout noise. With an ADC and memory per pixel, massively parallel "snap-shot" imaging, A/D conversion and high speed digital readout become practical, eliminating analog A/D conversion and readout bottlenecks. This benefits traditional high speed imaging applications (*e.g.*, [19, 90]) and enables efficient implementations of several still and standard video rate applications such as sensor

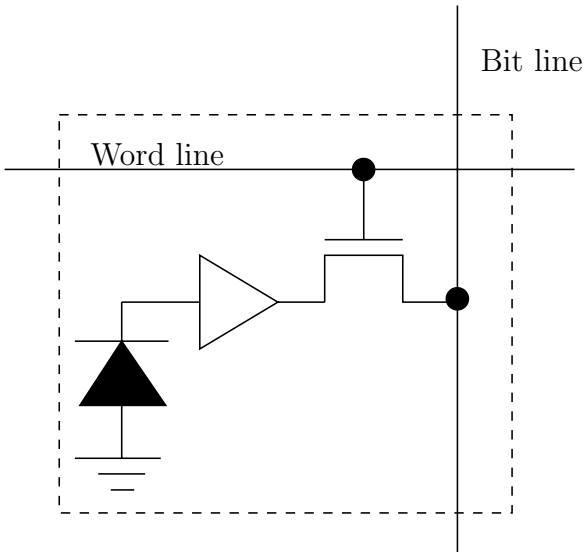


Figure 1.9: Active Pixel Sensor (APS)

dynamic range enhancement and motion estimation [102, 55, 56, 54]. The main drawback of DPS is its large pixel size due to the increased number of transistors per pixel. Since there is a lower bound on practical pixel sizes imposed by the wavelength of light, imaging optics, and dynamic range considerations, this problem diminishes as CMOS technology scales down to 0.18μm and below. Designing image sensors in such advanced technologies, however, is challenging due to supply voltage scaling and the increase in leakage currents [93].

1.3 Challenges in Digital Camera System Design

As we have seen from Figure 1.1, a digital camera is a very complex system consisting of many components. To achieve high image quality, all of these components have to be carefully designed to perform well not only individually, but also together as a complete system. A failure from any one of the components can cause significant degradation to the final image quality. This is true not just for those crucial components such as the image sensor and the imaging optics. In fact, if any one of the

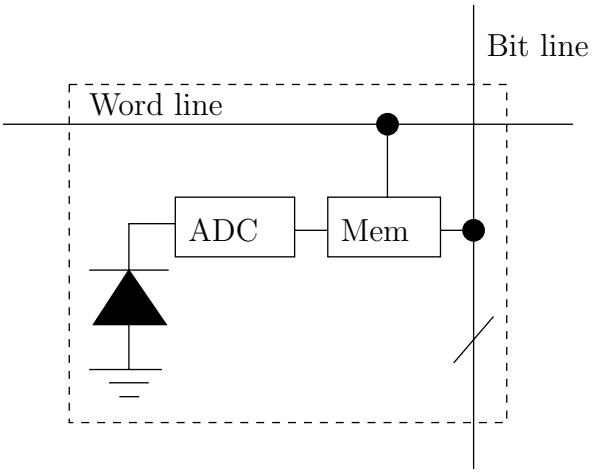


Figure 1.10: Digital Pixel Sensor (DPS)

color and image processing steps, such as color demosaicing, white balancing, color correction and gamut correction, or any one of the camera control functions, such as exposure control and auto focus, is not carefully designed or optimized for image quality, then the digital camera as a system will not deliver high quality images. Because of the complex nature of a digital camera system, it is extremely difficult to compare different system designs analytically since they may differ in many aspects and it is unclear how those aspects are combined and contribute to the ultimate image quality. While building actual test systems is the ultimate way of designing and verifying any practical digital camera product, it also requires significant amount of engineering and financial resources and often suffers from the long design cycle.

Since both prototyping actual hardware test systems and analyzing them theoretically have their inherent difficulties, it becomes clear that simulation tools that can model a digital camera system and help system designers fine tuning their designs are very valuable. Traditionally many well-known ray tracing packages such as the Radiance [69] do provide models for 3-D scenes and are capable of simulating the image formation through optics, they do not provide simulation capabilities of image sensors and camera controls that are crucial for a digital camera system. While complete digital camera simulators do exist, they are almost exclusively proprietary.

The only published articles on a digital camera simulator [9, 10] describe a somewhat incomplete simulator that lacks the detailed modeling of crucial camera components such as the image sensor. So in this thesis, I will introduce a digital camera simulator - vCam - that is from our own research effort. vCam can be used to examine a particular digital camera design by simulating the entire signal chain, from the scene to the optics, to the sensor, to the ADC and entire post processing steps. The digital camera simulator can be used to gain insights on each of the camera system parameters. We will then present two applications of using such a digital camera simulator in actual system designs.

1.4 Author's Contribution

The significant original contributions of this work include

- Introduced a complete digital camera system simulator that was jointedly developed by Peter Catrysse, Professor Brian Wandell and the author. In particular, the modeling of image sensors, the simulation of a digital camera's main functionality - converting photons into digital numbers under various camera controls, and the simulation of all the post processing come primarily from the author's effort.

- Developed a methodology for selecting the optimal pixel size in an image sensor design with the aid of the simulator. This work has provided an answer to an important design question that has not been thoroughly studied in the past due to its complex nature. The methodology is demonstrated for CMOS active pixel sensors.

- Performed the first investigation of selecting optimal multiple captures in a high dynamic range imaging system. Proposed competitive algorithms for scheduling captures and demonstrated those algorithms on real images using both the simulator and an experimental imaging system.

These contributions appear in Chapters 2, 3 and 4.

1.5 Thesis Organization

This dissertation is organized into five chapters of which this is the first. Chapter 2 describes vCam. The simulator provides models for the scene, the imaging optics, and the image sensor. It is implemented in Matlab as a toolbox and therefore is modular in nature to facilitate future modifications and extensions. Validation results on the camera simulator is also presented.

To demonstrate the use of the simulator in camera system design, the application that uses vCam to select the optimal pixel size as part of an image sensor design is then presented in Chapter 3. First the tradeoff between sensor dynamic range (DR) and spatial resolution as a function of pixel size is discussed. Then a methodology using vCam, synthetic contrast sensitivity function scenes, and the image quality metric S-CIELAB for determining optimal pixel size is introduced. The methodology is demonstrated for active pixel sensors implemented in CMOS processes down to 0.18um technology.

In Chapter 4 the application of using vCam to demonstrate algorithms for scheduling multiple captures in a high dynamic range imaging system is described. In particular, capture time scheduling is formulated as an optimization problem where average SNR is maximized for a given scene marginal probability density function (pdf). For a uniform scene pdf, the average SNR is a concave function in capture times and thus the global optimum can be found using well-known convex optimization techniques. For a general piece-wise uniform pdf, the average SNR is not necessarily concave, but rather a difference of convex functions (or in short, a D.C. function) and can be solved using D.C. optimization techniques. A very simple heuristic algorithm is described and shown to produce results that are very close to optimal. These theoretical results are then demonstrated on real images using vCam and an experimental high speed imaging system.

Finally, in Chapter 5, the contributions of this research are summarized and directions for future work are suggested.

Chapter 2

vCam - A Digital Camera Simulator

2.1 Introduction

Digital cameras are capable of capturing an optical scene and converting it directly into a digital format. In addition, all the traditional imaging pipeline functions, such as color processing, image enhancement and image compression, can also be integrated into the camera. This high level of integration enables quick capture, processing and exchange of images. Modern technologies also allow digital cameras to be made with small size, light weight, low power and low cost. As wonderful as these digital cameras seem to be, they are still lagging traditional film cameras in terms of image quality. How to design a digital camera that can produce excellent pictures is the challenge facing every digital camera system designer.

Digital cameras, however, as depicted in Figure 1.1, are complex systems combining optics, device physics, circuits, image processing, and imaging science. It is

difficult to assess and compare their performance analytically. Moreover, prototyping digital cameras for the purpose of exploring design tradeoffs can be prohibitively expensive. To address this problem, a digital camera simulator - vCam - has been developed and used to explore camera system design tradeoffs. A number of studies [13, 16] have been carried out using this simulator.

It is worth mentioning that our image capture is mainly concentrated on capturing the wavelength information of the scene by treating the scene as a 2-D image and ignoring the 3-D geometry information. Such a simplification can still provide us with reasonable image irradiance information on the sensor plane as inputs to the image sensor. With our expertise in image sensor, we have included detailed image sensor models to simulate the sensor response to the incoming irradiance and to complete the digital camera image acquisition pipeline.

The remainder of this chapter is organized as follows. In the next section we will describe the physical models underlying the camera simulator by following the signal acquisition path in a digital camera system. In Section 2.3 we will describe the actual implementation of vCam in Matlab. Finally in Section 2.4 we will present the experimental results of vCam validation.

2.2 Physical Models

The digital camera simulator, vCam, consists of a description of the imaging pipeline from the scene to the digital picture (Figure 2.1). Following the signal path, we carefully describe the physical models upon which vCam is built. The goal is to provide a detailed description of each camera system component and how these components interact to create images. A digital camera performs two distinct functions: first, it acquires an image of a scene; second, this image is processed to provide a faithful yet appealing representation of the scene that can be further manipulated digitally if necessary. We will concentrate on the image acquisition aspect of a digital camera system. The image acquisition pipeline can be further split into two parts, an optical pipeline, which is responsible for collecting the photons emitted or reflected from

the scene, and an electrical pipeline, which deals with the conversion of the collected photons into electrical signals at the output of image sensor. Following image acquisition, there is an image processing pipeline, consisting of a number of post processing and evaluation steps. We will only briefly mention these steps for completeness in Section 2.3.

2.2.1 Optical Pipeline

In this section we describe the physical models used in the optical pipeline [1]. The front-end of the optical pipeline is formed by the scene and is in fact not part of the digital camera system. Nevertheless, it is very important to have an accurate yet tractable model for the scene that is going to be imaged by the digital camera. Specifically, we depict how light sources and objects interact to create a scene. [2]

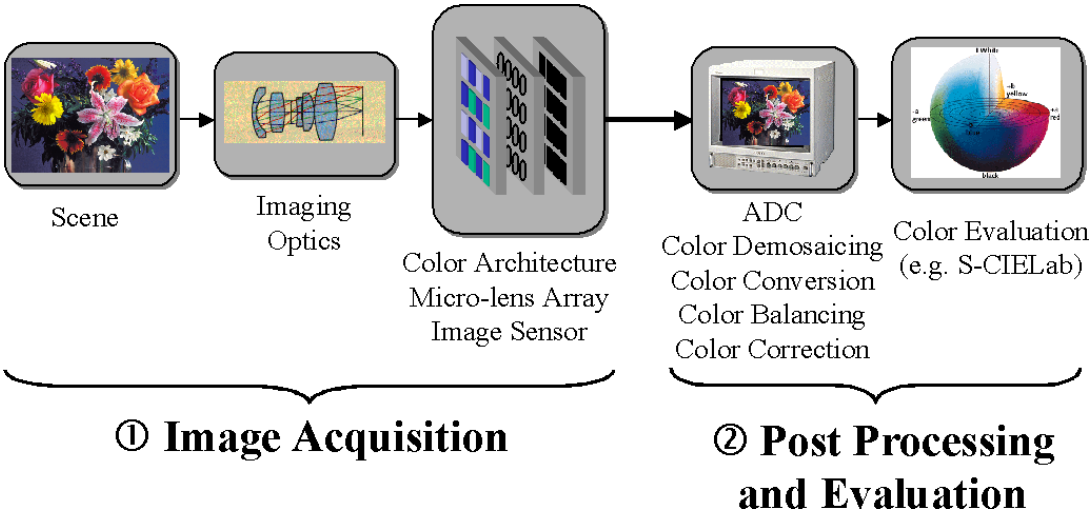


Figure 2.1: Digital still camera system imaging pipeline - How the signal flows

[1] Special acknowledgments go to Peter Catrysse who implemented most of the optical pipeline in vCam and contributed to a significant amount of writing in this section.

[2] In its current implementation, vCam assumes flat, extended Lambertian sources and object surfaces being imaged onto a flat detector located in the image plane of lossless, diffraction-limited imaging optics.

We will follow the photon flux, carrier of the energy, as it is generated and propagates along the imaging path to form an image. We begin by providing some background knowledge on calculating the photon flux generated by a Lambertian light source characterized by its radiance. In particular, we point out that the photon flux scattered by a Lambertian object is a spectrally filtered version of the source's photon flux. We continue with a description of the source-receiver geometry and discuss how it affects the calculation of the photon flux in the direction of the imaging optics. Finally, we incorporate all this background information into a radiometrical optics model and show how light emitted or reflected from the source is collected by the imaging optics and results image irradiance at the receiver plane. The optical signal path can be seen in Figure 2.2.

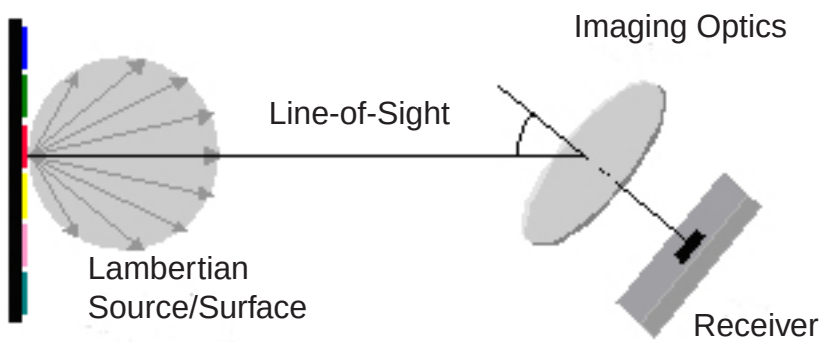


Figure 2.2: vCam optical pipeline

Our final objective is to calculate the number of photons incident at the detector plane. In order to achieve that objective we take the approach of following the photon flux, *i.e.*, the number of photons per unit time, from the source all the way to the receiver (image sensor), starting with the photon flux leaving the source.

Lambertian Light Sources

The photon flux emitted by an extended source depends both on the area of the source and the angular distribution of emission. We, therefore, characterize the source by its

emitted flux per unit source area and per unit solid angle and call this the radiance L expressed in $[watts/m^2 \cdot sr]$ [3]. Currently vCam only allows flat extended sources of the Lambertian type. By definition, a ray emitted from a Lambertian source is equally likely to travel outwards in any direction. This property of Lambertian sources and surfaces results in a radiance L_o that is constant and independent of the angle between the surface and a measurement instrument.

We proceed by building up a scene consisting of a Lambertian source illuminating a Lambertian surface. An extended Lambertian surface illuminated by an extended Lambertian source acts as a secondary Lambertian source. The (spectral) radiance of this secondary source is the result of the modulation of the spectral radiance of the source by the spectral reflectance of the surface [4]. This observation allows us to work with the Lambertian surface as a (secondary) source of the photon flux. To account for non-Lambertian distributions, it is necessary to apply a bi-directional reflectance distribution function (BRDF) [63]. These functions are measured with a special instrument called a goniophotometer (an example [62]). The distribution of scattered rays depends on the surface properties, with one common division being between dielectrics and inhomogeneous materials. These are modeled as having specular and diffuse terms in different ratios with different BDRFs.

Source-Receiver Geometry and Flux Transfer

To calculate the total number of photons incident at the detector plane of the receiver, we must not only account for the aforementioned source characteristics but also for the geometric relationship between the source and the receiver. Indeed, the total number of photons incident at the receiver will depend on source radiance, and on

[3] sr, in short for steradian, is the standard unit of a solid angle.

[4] For an extended Lambertian source, the exitance M (the concept of exitance is similar to radiance. It represents the radiant flux density from a source or a surface and has a unit in $[watts/m^2]$) into a hemisphere is given by $M_{\text{source}} = \pi L_{\text{source}}$. If the surface can receive the full radiant exitance from the source, the radiant incidence (or irradiance) E on the surface is equal to the radiant exitance M_{source}. Thus $E = \pi L_{\text{source}}$ and before being re-emitted by the surface it is modulated by the surface reflectance S. Therefore the radiant exitance becomes $M = SM_{\text{source}}$ and since the surface is Lambertian, $M = \pi L$ holds for the surface as well. This means that the radiance L of the surface is given by SL_{source}. For more details, see [76].

the fraction of the area at the emitter side contributing to the photon flux at the receiver side. Typically this means we have to calculate the projected area of the emitter and the projected area of the receiver using the angles between the normal of the respective surfaces and the line-of-sight between them. This calculation yields the fundamental flux transfer equation [92].

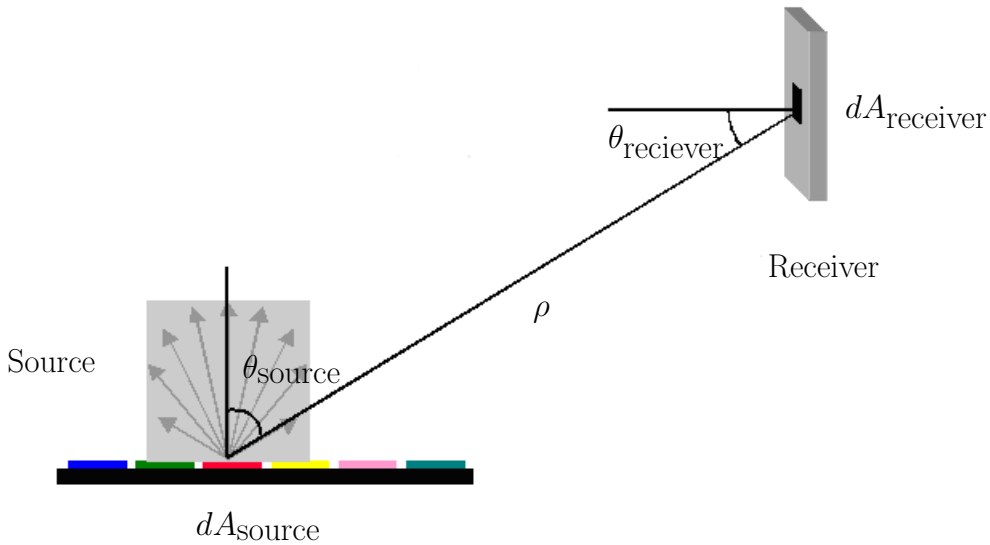


Figure 2.3: Source-Reciever geometry

To describe the flux transfer between the source and the receiver, no matter how complicated both surfaces are and irrespective of their geometry, the following fundamental equation can be used to calculate the transferred differential flux $d^2\Phi$ between a differential area at the source and a differential area at the receiver

$$d^2\Phi = L\frac{dA_{\text{source}} \cdot \cos\theta_{\text{source}} \cdot dA_{\text{receiver}} \cdot \cos\theta_{\text{receiver}}}{\rho^2}, \tag{2.1}$$

where as shown in Figure 2.3, L represents the radiance of the source, A represents area, θ is the angle between the respective surface normals and the line of sight between both surfaces, and ρ stands for the line-of-sight distance. This equation

specifies the differential flux radiated from the projected differential area $dA_{\text{source}} \cdot \cos\theta_{\text{source}}$ of the source to the projected differential area $dA_{\text{receiver}} \cdot \cos\theta_{\text{receiver}}$ of the receiver. Notice that this equation does not put any limitations on L, nor does it do so on any of the surfaces.

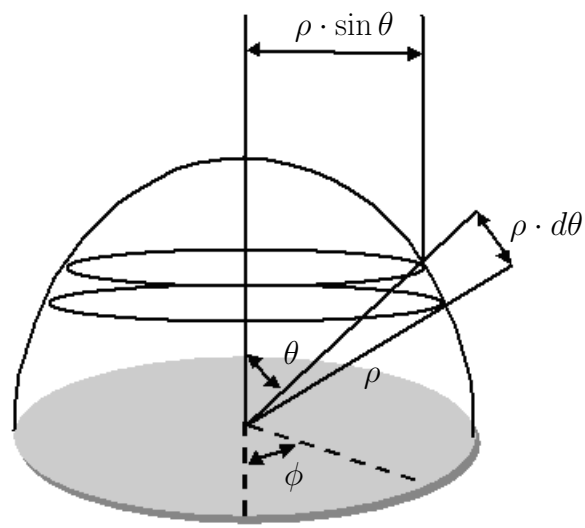


Figure 2.4: Defining solid angle

Solid Angle

Before we use Equation (2.1) to derive the photon flux transfer from the source to the reciever, let us quickly review some basics of solid angle. A differential element of area on a sphere with radius ρ (refer to Figure 2.4) can be written as

$$dA = \rho \cdot \sin\theta \cdot d\phi \cdot \rho \cdot d\theta, \tag{2.2}$$

where ϕ is the azimuthal angle. To put into the context of source-receiver geometry, θ is the angle between the flux of photons and the line-of-sight. This area element can be interpreted as the projected area $dA_{\text{receiver}} \cdot \cos\theta_{\text{receiver}}$ in the fundamental flux transfer equation, *i.e.*, the area of the receiver on a sphere centered at the source

with radius the line-of-sight distance ρ.

By definition, to obtain the differential element of solid angle we divide this area by the radius squared, and get

$$d\Omega_{\text{receiver/source}} = \frac{dA_{\text{receiver}} \cdot \cos\theta_{\text{receiver}}}{\rho^2} = \sin\theta d\theta d\phi, \tag{2.3}$$

where $d\Omega_{\text{receiver/source}}$ represents the differential solid angle of the receiver as seen from the source. Insert Equation (2.3) into the fundamental flux transfer equation, we get

$$d^2\Phi = L \cdot dA_{\text{source}} \cdot \cos\theta_{\text{source}} \cdot d\Omega_{\text{receiver/source}}. \tag{2.4}$$

Typically we are interested in the total solid angle formed by a cone with half-angle α, centered on the direction perpendicular to the surface [5], as seen in Figure 2.5, since this corresponds to the photon flux emitted from a differential area dA_{source} and reached the receiver. Such a total solid angle can be written as

$$\Omega = \int_{\text{perpendicular}} d\Omega = \int_0^{2\pi} \int_0^{\alpha} \sin\theta d\theta d\phi, \tag{2.5}$$

and apply Equation (2.5) to Equation (2.4), we get

$$d\Phi = L \cdot dA_{\text{source}} \int_0^{2\pi} \int_0^{\alpha} \cos\theta \sin\theta d\theta d\phi = \pi L \cdot dA_{\text{source}} (\sin\alpha)^2. \tag{2.6}$$

[5] If the cone is centered on an oblique line-of-sight, then in order to maintain the integrability of the flux based on a Lambertian surface, we now have a solid angle whose area on the unit-radius sphere is not circular but rectangular, limited by 4 angles. This will break the symmetry around the line-of-sight and complicate any further calculations involving radial symmetric systems such as the imaging optics. For this reason, vCam currently only supports the case of a perpendicular solid angle.

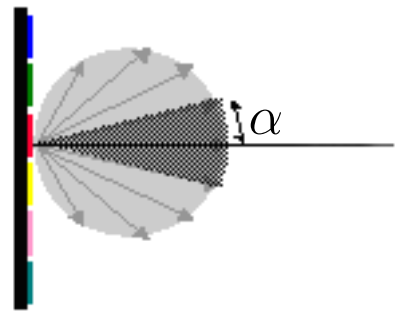


Figure 2.5: Perpendicular solid angle geometry

Radiometrical Optics Model

Imaging optics are typically used to capture an image of a scene inside digital cameras. As an important component of the digital camera system, optics needs to be modeled. What we have derived so far in Equation (2.6) can be viewed as the photon flux incident at the entrance aperture of the imaging optics. What we are interested in is the irradiance at the image plane where the detector is located. In this section we will explain how, once we know the photon flux at the entrance aperture and the properties of the optics, we can compute the photon flux and the irradiance at the image plane where the sensor is located. And this irradiance is the desired output at the end of the optical pipeline.

We introduce a new notation better suitable for the image formation using a radiometrical optics model and restate the derived result in Equation (2.6) with the new notation. Consider now an elementary beam, originating from a small part of the source, passing through a portion of the optical system, and producing a portion of the image, as seen in Figure 2.6. This elementary beam subtends an infinitesimal solid angle $d\Omega$ and originates from an area dA_o with Lambertian radiance L_o. From Equations (2.3) and (2.4), the flux in the elementary beam is given by

$$d^2\Phi = L_o \cdot \cos\theta \cdot dA_o \cdot d\Omega_o = L_o \cdot dA_o \cdot \cos\theta \sin\theta d\theta d\phi. \tag{2.7}$$

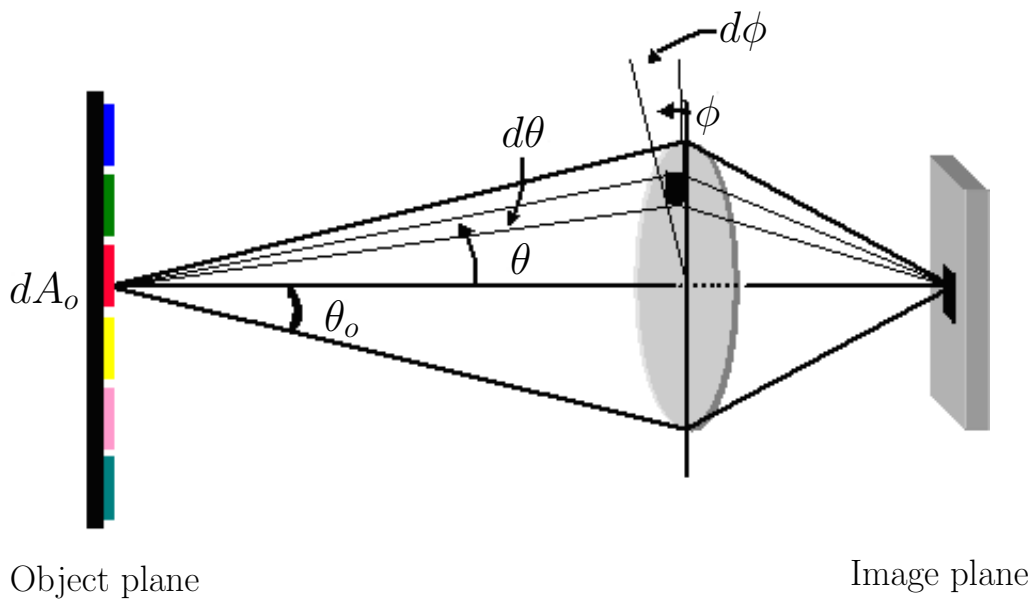


Figure 2.6: Imaging geometry

We follow the elementary beam until it arrives at the entrance pupil or the first principle plane [6] of the optical system. If we now consider a conical beam of half angle θ_o, we will have to integrate the contributions of all these elementary beams,

$$d\Phi_o = L_o \cdot dA_o \int_0^{2\pi} d\phi \int_0^{\theta_o} \cos\theta \sin\theta \cdot d\theta = \pi \cdot L_o \cdot (\sin\theta_o)^2 \cdot dA_o. \tag{2.8}$$

This is the result obtained in the previous section using the new notation. We now proceed to go from the flux at the entrance pupil, *i.e.*, the first principle plane of the optical system to the irradiance at the image plane at the photodetector.

If the system is lossless, the image formed on the first principle plane is converted without loss into a unit-magnification copy on the second principle plane and we have

[6] Principle planes are conjugate planes; they are images from each other like the object and the image plane. Furthermore principal planes are planes of unit magnification and as such they are unit images of each other. In a well-corrected optical system the principal planes are actually spherical surfaces. In the paraxial region, the surfaces can be treated as if they were planes.

conservation of flux

$$d\Phi_i = d\Phi_o \tag{2.9}$$

Using Abbe's sine relation [7], we can derive that not only flux but also radiance is conserved, i.e. $L_i = L_o$ for equal indices of refraction $n_i = n_o$ in object and image space. The radiant or luminous flux per unit area, i.e. the irradiance, at the image plane will be the integral over the contributions of each elementary beam. A conical beam of half angle θ_i will contribute

$$d\Phi_i = \pi \cdot L_i \cdot (\sin\theta_i)^2 \cdot dA_i \tag{2.10}$$

in the image space. Dividing the flux $d\Phi_i$ by the image area dA_i, we obtain the image irradiance in image space

$$E_i = \frac{d\Phi_i}{dA_i} = \pi L_i (\sin\theta_i)^2. \tag{2.11}$$

We now use the conservation of radiance law, yielding

$$E_i = \pi L_o (\sin\theta_i)^2. \tag{2.12}$$

Irradiance in terms of f-number.

The expression for the image irradiance in terms of the half-angle θ_i of the cone in the image plane, as derived above, can be very useful by itself. In our simulator, however, we use an expression which includes only the f-number ($f/\#$) and the magnification (besides the radiance, of course). We show now how to derive this expression starting with a model for the diffraction-limited imaging optics which uses the f-number.

The f-number is defined as the ratio of the focal length f to the clear aperture

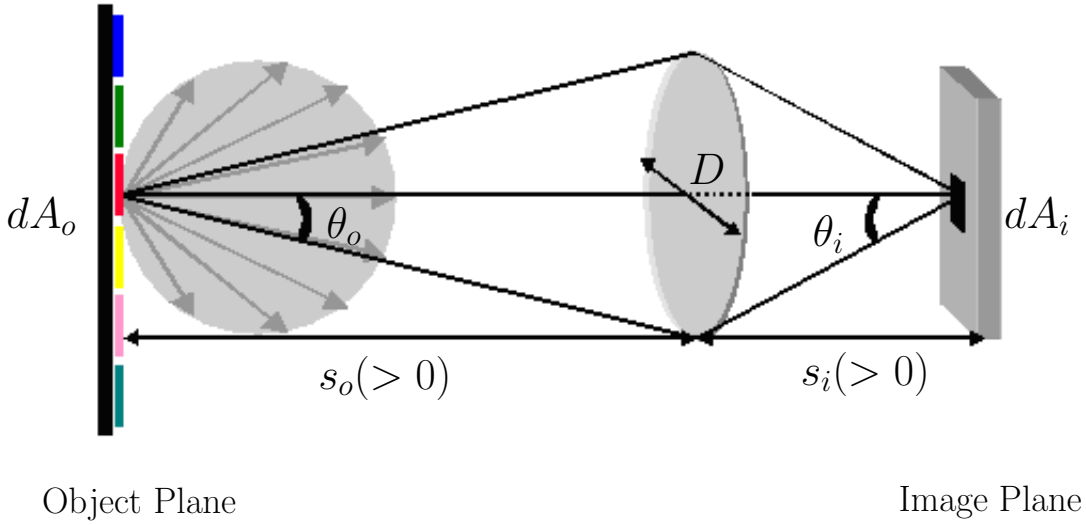


Figure 2.7: Imaging law and $f/\#$ of the optics

diameter D of the optical system,

$$f/\# = \frac{f}{D}. \tag{2.13}$$

Using the lens formula [80], where $s_o(> 0)$ represents the object distance and $s_i(> 0)$ the image distance,

$$\frac{1}{f} = \frac{1}{s_o} + \frac{1}{s_i}, \tag{2.14}$$

we derive an expression of the magnification and the image magnification

$$m = -\frac{s_i}{s_o} = 1 - \frac{s_i}{f} < 0 \tag{2.15}$$

and

$$s_i = (1 - m)f. \tag{2.16}$$

We rewrite the sine

$$(\sin\theta_i)^2 = \frac{1}{1+4(\frac{s_i}{D})^2} \tag{2.17}$$

and finally get an expression for the irradiance in terms of f-number and magnification

$$E_i = \pi \cdot L_o \frac{1}{1+4(f/\#(1-m))^2} \tag{2.18}$$

with $m < 0$.

Off-axis image irradiance and cosine-fourth law

In this analysis we will study the off-axis behavior of the image irradiance, which we have not considered so far. We will show how off-axis irradiance is related to on-axis irradiance through the cosine-fourth law [7]. If the optical system is lossless, the irradiance at the entrance pupil is identical to irradiance at the exit pupil due to conservation of flux. Therefore we can start the calculations with the light at the exit pupil and consider the projected area of the exit pupil perpendicular to an off-axis ray.

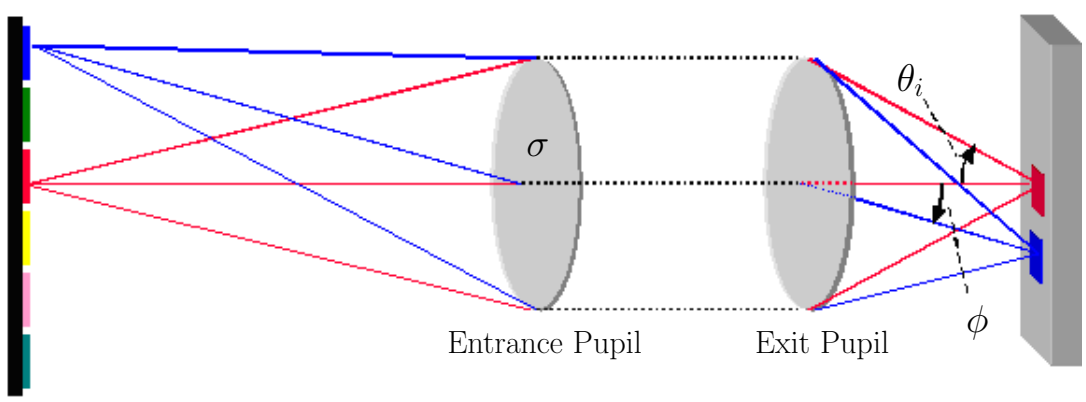


Figure 2.8: Off-axis geometry

[7] The "cosine-fourth law" is not a real physical law but a collection of four separate cosine factors which may or may not be present in a given imaging situation. For more details, see [52].

The solid angle subtended by the exit pupil from an off-axis point is related to the solid angle subtended by the exit pupil from an on-axis point by

$$\Omega_{\text{off-axis}} = \Omega_{\text{on-axis}}(\cos\phi)^2. \tag{2.19}$$

The exit pupil with area σ is viewed obliquely from an off-axis point, and its projected area $\sigma_\perp$ is reduced by a factor which is approximately $\cos\phi$ (earlier referred to as $\cos\theta_{\text{receiver}}$),

$$\sigma_\perp = \sigma\cos\phi. \tag{2.20}$$

This is a fair approximation only if the distance from the exit pupil to the image plane is large compared to the size of the pupil. The fourth and last cosine factor is due to the projection of an area perpendicular to the off-axis ray onto the image plane. Combining all these separate cosine factors yields,

$$E_i = \pi \cdot L_o \frac{1}{1 + 4(f/\#(1-m))^2}(\cos\phi)^4. \tag{2.21}$$

Equation (2.21), however, does include one approximate cosine factor. A more complicated expression [31] for the irradiance which takes care of this approximation and is accurate even when the exit pupil is large compared with distance is

$$E_i = \frac{\pi \cdot L_o}{2}\left(1 - \frac{1 - (\tan\theta_i)^2 + (\tan\phi)^2}{\sqrt{(\tan\phi)^4 + 2(\tan\phi)^2(1-(\tan\theta_i)^2) + 1/(\cos\theta_i)^4}}\right). \tag{2.22}$$

2.2.2 Electrical Pipeline

In this section we will describe the vCam electrical model, which is responsible for converting incoming photon flux or the image irradiance on the image sensor plane to electrical signals at the sensor outputs. The analog electrical signals are then converted into digital signals via an ADC for further digital signal processing. The

sensing consists of two main actions, spatial/spectral/time integration, and the addition of temporal noise and fixed pattern noise; and a number of secondary but yet very complicated effects such as diffusion modulation transfer function and pixel vignetting. We will describe them one by one in the following subsections. To model these operations of image sensors, it is necessary to have the knowledge of key sensor parameters. Sensor parameters are best characterized via experiments. For the cases when experimental sensor data are not available, we will show how the parameters can be estimated.

Spectral, Spatial and Time Integration

Image sensors all have photodetectors which convert incident radiant energy (photons) into charges or voltages that are ideally proportional to the radiant energy. The conversion is done in three steps : incident photons generate electron-hole (e-h) pairs in the sensor material (e.g. silicon); the generated charge carriers are converted into photocurrent; the photocurrent (and dark current due to device leakage) are integrated into charge. Note that the first step involves photons coming at different wavelengths (thus different energy) and exciting e-h pairs, therefore to get the total number of generated e-h pairs, we have to sum up the effect of photons that are spectrally different. The resulting electrons and holes will move under the influence of electric fields. These charges are integrated over the photodetector area to form the photocurrent. Finally the photocurrent is integrated over a period of time, which generates the charge that can be read out directly or converted into voltage and then read out. It is evident that the conversion from photons to electrical charges really involves a multi-dimensional integration. It is a simultaneously spectral, spatial and time integration, as described by Equation (2.23),

$$Q = q \int_0^{t_{int}} \int_{A_D} \int_{\lambda_{min}}^{\lambda_{max}} E_i(\lambda) s(\lambda) d\lambda dA dt, \tag{2.23}$$

where Q is the charge collected, q is the electron charge, A_D is the photodetector area, t_{int} is the exposure time, $E_i(\lambda)$ is the incoming photon irradiance as specified in the previous section and $s(\lambda)$ is the sensor spectral response, which characterizes the fraction of photon flux that contributes to photocurrent as a function of wavelength λ. Notice that the two inner integrations actually specify the photocurrent i_{ph}, i.e.,

$$i_{ph} = q \int_{A_D} \int_{\lambda_{min}}^{\lambda_{max}} E_i(\lambda) s(\lambda) d\lambda dA. \tag{2.24}$$

In cases where voltages are read out, given the sensor conversion gain g (which is the output voltage per charge collected by the photodetector), the voltage change at the sensor output is

$$v_o = g \cdot Q. \tag{2.25}$$

This voltage can then be converted into a digital number via an ADC.

Additive Sensor Noise Model

An image sensor is a real world device which unfortunately is subjected to real world non-idealities. One of such non-idealities is noise. The sensor output is not a pure and clean signal that is proportional to the incoming photon flux, instead it is corrupted with noise. In our context, such a noise corruption to the sensor output refers to the inclusion of temporal variations in pixel output values due to device noise and spatial variations due to device and interconnect mismatches across the sensor. Such temporal variations result temporal noise and spatial variations cause fixed pattern noise.

Temporal noise includes primarily thermal noise and shot noise. Thermal noise is generated by thermally induced motion of electrons in resistive regions such as polysilicon resistors and MOS transistor channels in strong inversion regime. Thermal noise typically has zero mean, very flat and wide bandwidth, and samples that follows Gaussian distributions. Consequently it is modeled as a white Gaussian noise (WGN).

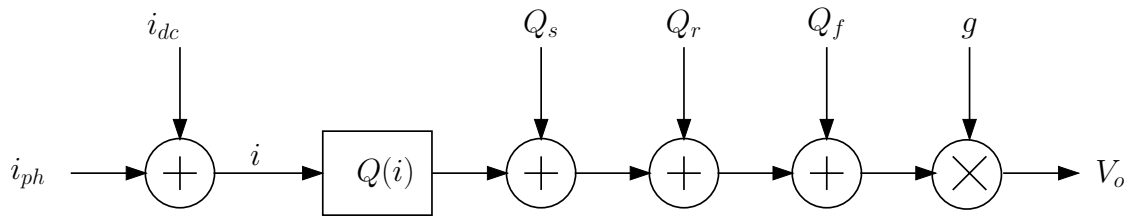
For an image sensor, the read noise, which is the noise occurred during reset and readout, is typically thermal noise. Shot noise is caused either by thermal generation within a depletion region such as in a pn junction diode, or by the random generation of electrons due to the random arrival of photons. Even though the photon arrivals are typically characterized by Poisson distributions, it is common practice to model shot noise as a WGN since Gaussian distributions are very good approximations of Poisson distributions when the arrival rate is high. Spatial noise, or fixed pattern noise (FPN), is the spatial non-uniformity of an image sensor. It is fixed for a given sensor such that it does not vary from frame to frame. FPN, however, varies from sensor to sensor.

We specify a general image sensor model including noise, as shown in Figure 2.9, where i_{ph} is the photo-generated current, i_{dc} is the photodetector dark current, Q_s is the shot noise, Q_r is the read noise, and Q_f is the random variable representing FPN. All the noises are assumed to be mutually independent as well as signal independent. The noise model is additive and with noise, the output voltage now becomes

$$v_o = g \cdot (Q(i_{ph} + i_{dc}) + Q_s + Q_r + Q_f). \tag{2.26}$$

Diffusion Modulation Transfer Function

The image sensor is a spatial sampling device, therefore the sampling theorem applies and sets the limits for the reproducibility in space of the input spatial frequencies. The result is that spatial frequency components higher than the Nyquist rate cannot be reproduced and cause aliasing. The image sensor, however, is not a traditional point sampling device due to two reasons: photocurrent is integrated over the photodetector area before sampling; and diffusion photocurrent may be collected by neighboring pixels instead of where it is generated. These two effects cause low pass filtering before spatial sampling. The degradation on the frequencies below Nyquist frequency is usually measured by modulation transfer function (MTF). It can be seen that the overall sensor MTF includes the carrier diffusion MTF and sensor aperture integration



Where

- the charge
$$Q(i) = \begin{cases} \frac{1}{q}(it_{int}) \text{ electrons} & \text{for } 0 < i < \frac{qQ_{max}}{t_{int}} \\ Q_{max} & \text{for } i \geq \frac{qQ_{max}}{t_{int}} \end{cases}$$
- shot noise charge $Q_s \sim \mathcal{N}(0, \frac{1}{q}it_{int})$
- read noise charge $Q_r \sim \mathcal{N}(0, \sigma_r^2)$
- FPN Q_f is zero mean and can be represented as sum of pixel and column components
$$Q_f = \frac{1}{g}(X + Y)$$
or offset and gain components
$$Q_f = \frac{1}{g}(\Delta H \cdot j_{ph} + \Delta V_{os})$$
- g is the sensor conversion gain in V/electron

Figure 2.9: vCam noise model

MTF. Though it may not be entirely precise [82], it is common practice to take the product of these two MTFs as the overall sensor MTF. This product may overestimate the MTF degradation, but it can still serve as a fairly good worst-case approximation. The integration MTF is automatically taken care of by collecting charges over the photodetector area as described in Section 2.2.2. We will introduce the formulae for calculating diffusion MTF in this section.

It should be noted that diffusion MTF in general is very difficult to find analytically and in practice it is often measured experimentally. Theoretical modeling of the diffusion MTF can be found in two excellent papers by Serb [73] and Stevens [81]. The formulae we implemented in vCam correspond to a 1-D diffusion MTF model and are shown in Equations (2.27)-(2.28) for a n-diffusion/p-substrate photodiode. The full derivation of those formulae is available at our homepage [1].

$$\text{diffusion MTF}(f) = \frac{D(f)}{D(0)} \tag{2.27}$$

and

$$D(f) = \frac{q(1 + \alpha L_f - e^{-\alpha L_d})}{1 + \alpha L_f} - \frac{qL_f \alpha e^{-\alpha L_d}(e^{-\alpha L} - e^{-\frac{L}{L_f}})}{(1 - (\alpha L_f)^2)\sinh(\frac{L}{L_f})} \tag{2.28}$$

where α is the absorption coefficient of silicon and is a function of wavelength λ. L_f is defined in Equation (2.29) with L_n being the diffusion length of minority carriers (i.e. electrons) in p-substrate for our photodiode example. L is the width of depletion region and L_d is the width (i.e. thickness) of the (p-substrate) quasi-neutral region. f is the spatial frequency.

$$L_f^2 = \frac{L_n^2}{1 + (2\pi f L_n)^2}. \tag{2.29}$$

Pixel Vignetting

Image sensor designers often take advantage of technology scaling either by reducing pixel size or by adding more transistors to the pixel. In both cases, the distance from the chip surface to the photodiode increases relative to the photodiode planar

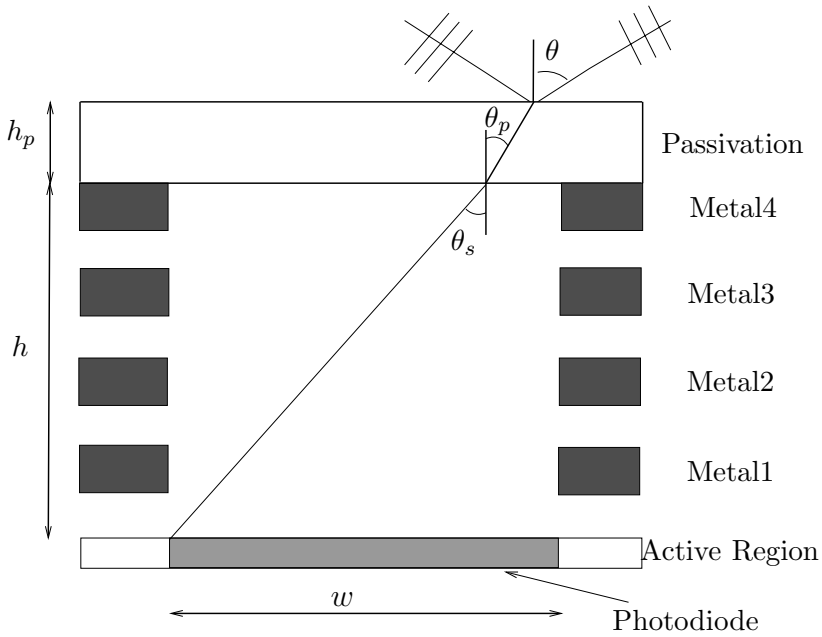


Figure 2.10: Cross-section of the tunnel of a DPS pixel leading to the photodiode

dimensions. As a result, photons must travel through an increasingly deeper and narrower "tunnel" before they reach the photodiode. This is especially problematic for light incident at oblique angles where the narrow tunnel walls cast a shadow on the photodiode. This severely reduces its effective quantum efficiency. Such a phenomenon is often called pixel vignetting. The QE reduction due to pixel vignetting in CMOS image sensors has been thoroughly studied by Catrysse *et al.* [8] in [14] and in that paper a simple geometric model of the pixel and imaging optics is constructed to account for the QE reduction. vCam currently implements such a geometric model.

First consider the pixel geometric model of a CMOS image sensor first. Figure 2.10 shows the cross-section of the tunnel leading to the photodiode. It consists of two layers of dielectric: the passivation layer and the combined silicon dioxide layer. An incident uniform plane wave is partially reflected at each interface between two layers. The remainder of the plane wave is refracted. The passivation layer material is Si_3N_4. It has an index of refraction n_p and a thickness h_p, while the combined oxide layer

[8]Special acknowledgments go to Peter Catrysse and Xinqiao Liu for supplying the two figures used in this section

has an index of refraction n_s and a thickness h_s. If a uniform plane wave is incident at an angle θ, it reaches the photodiode surface at an angle

$$\theta_s = \sin^{-1}(\frac{\sin\theta}{n_s}).$$

Assuming an incident radiant photon flux density E_{in}(photons/s·m^2) [9] at the surface of the chip, the photon flux density reaching the surface of the photodiode is given by

$$E_s = T_p T_s E_{in},$$

where T_p is the fraction of incident photon flux density transmitted through the passivation layer and T_s is the fraction of incident photon flux density transmitted through the combined SiO_2 layer. Because the plane wave strikes the surface of the photodiode at an oblique angle θ_s, a geometric shadow is created, which reduces the illuminated area of the photodiode as depicted in Figure 2.11. Taking this reduction into consideration and using the derived E_s we can now calculate the fraction of the photon flux incident at the chip surface that eventually would reach the photodiode

$$\text{QE reduction factor} = T_s T_p (1 - \frac{h}{w}\tan\theta_s)\cos\theta_s$$

To complete our geometric model, we include a simple geometric model of the imaging lens. The lens is characterized by two parameters: the focal length f and the $f/\#$. As assumed in Section 2.2.1, we consider the imaging of a uniformly illuminated Lambertian surface. Figure 2.12 shows the illuminated area for on- and off-axis pixels. Since the incident illumination is no longer a plane wave, it is difficult to analytically solve for the normalized QE as before. Instead, in vCam we numerically solve for the incident photon flux assuming the same tunnel geometry and lens parameters.

[9] Since we are using geometric optics here we do not need to specify the spectral distribution of the incident illumination.

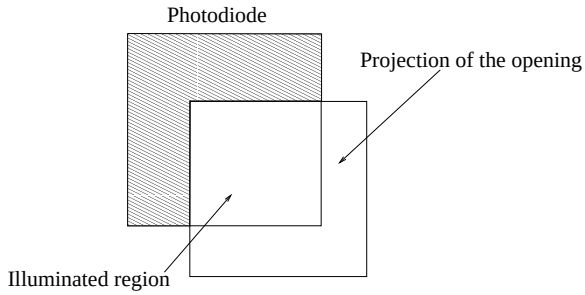


Figure 2.11: The illuminated region at the photodiode is reduced to the overlap between the photodiode area and the area formed by the projection of the square opening in the 4th metal layer

Sensor Parameter Estimation

From previous sections it is apparent that several key sensor parameters are required in order to calculate the final sensor output. In this section we will describe how these parameters can be derived if not given directly.

A pixel usually consists of a photodetector over which photon-excited charges are accumulated, and some readout circuitry for reading out the collected charges. The photodetector can be a photodiode or a photogate. And depending on its photon-collecting region, the photodetector can be further differentiated. Two examples may be n-diffusion/p-substrate photodiodes and n-well/p-substrate photodiodes. There are two important parameters that are used to describe the electrical properties of a photodetector: dark current density and spectral response. Ideally these parameters are measured experimentally in order to achieve a high accuracy. In reality, however, measurement data are not always available and we will have to estimate these parameters using the information we have access to. For instance, technology files are required by image sensor designers to tape out their chips. With the help of the technology files, SPICE simulation can be used to estimate some of the photodetector electrical properties such as the photodetector capacitance. Device simulators such as Medici [4] can also be used to help determine photodetector capacitance, dark current density and spectral response. For cases where even simulated data are not available, we will have to rely on results based on theoretical analysis. We will use

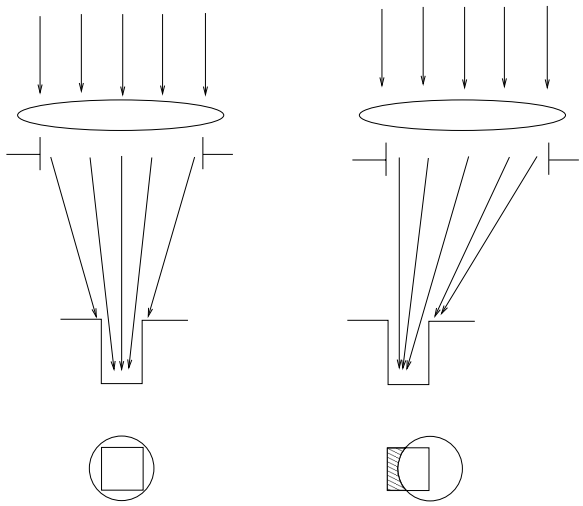


Figure 2.12: Ray diagram showing the imaging lens and the pixel as used in the uniformly illuminated surface imaging model. The overlap between the illuminated area and the photodiode area is shown for on and off-axis pixels

an n-diffusion/p-substrate photodiode to illustrate our ideas here.

Figure 2.13 shows a cross sectional view for the photodiode. With a number of simplifying assumptions including abrupt pn junction, depletion approximation, low level injection and short base region approximation, the spectral response of the photodiode can be calculated [1] as

$$\eta(\lambda) = \frac{1}{\alpha}\left(\frac{(1-e^{-\alpha x_1})}{x_1} - \frac{(e^{-\alpha x_2} - e^{-\alpha x_3})}{x_3 - x_2}\right) \text{ electrons/photons,} \tag{2.30}$$

where α is the light absorption coefficient of silicon. And the dark current density is determined as

$$\begin{aligned} j_{dc} &= j_{dc}^{p} + j_{dc}^{n} + j_{dc}^{sc} \\ &= qD_p\frac{n_i^2}{N_d x_1} + qD_n\frac{n_i^2}{N_a(x_3 - x_2)} + \frac{qn_i}{2}\left(\frac{x_n}{\tau_p} + \frac{x_p}{\tau_n}\right). \end{aligned} \tag{2.31}$$

This analysis ignores reflection at the surface of the chip, it also ignores the reflections

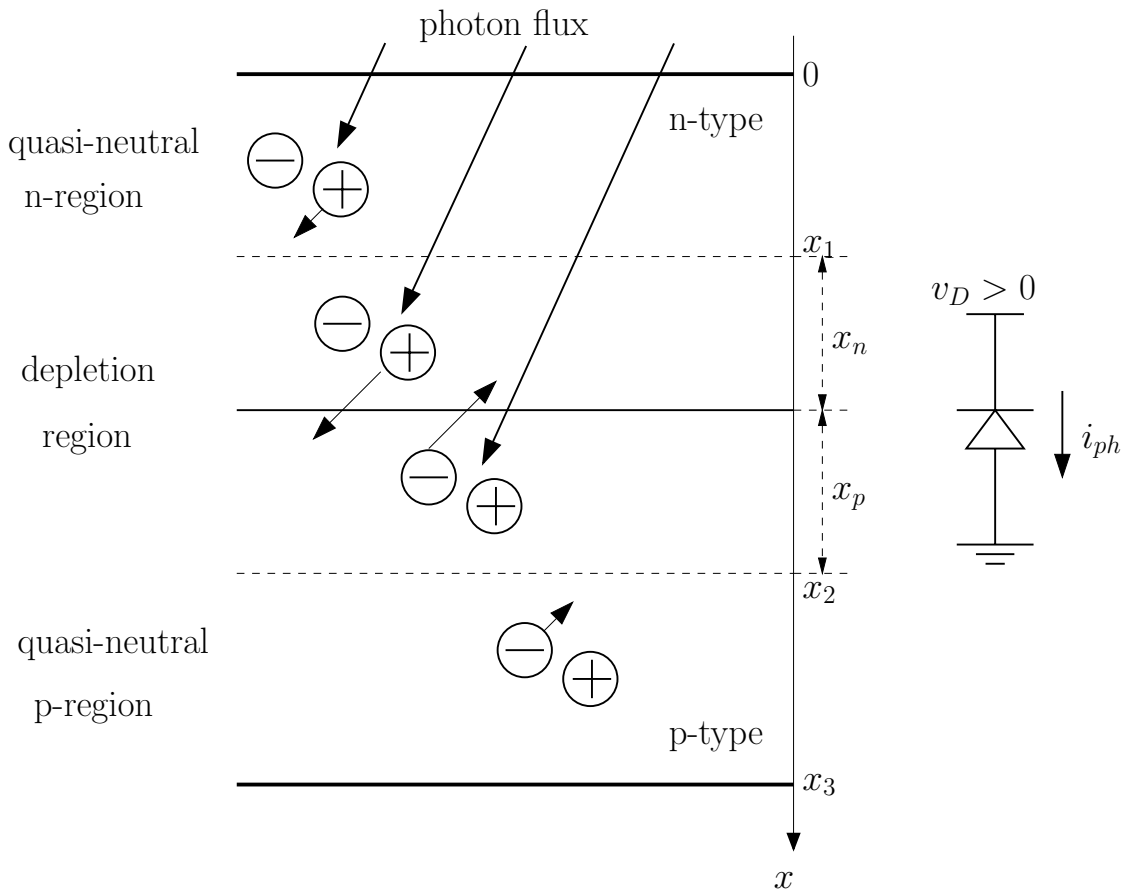


Figure 2.13: An n-diffusion/p-substrate photodiode cross sectional view

and absorptions in layers above the photodetector. It does not take into account the edge effect as well. So the result of this analysis is somewhat inaccurate, but it is helpful in understanding the effect of various parameters on the performance of the sensor. Evaluating the above equations require process information such as the poly thickness, well depth, doping densities and so on. Unfortunately process information is not necessarily available for various reasons. For instance, a chip fabrication factory may be unwilling to release the process parameters, or an advanced process has not been fully characterized. For such cases, process parameters need to be estimated. Our estimation is based on a generic process in which all the process parameters are known, and a set of scaling rules specified by the International Technology Roadmap

for Semiconductors (ITRS) [50].

Besides specifying the photodetector, to completely describe a pixel, we also need to specify its readout circuitry, which also uniquely determines the type of the pixel architecture, *i.e.*, a CMOS APS, a PPS or a CCD etc. The readout circuitry often includes both pixel-level circuitry and column-level circuitry. The readout circuitry decides two important parameters of the pixel, the conversion gain and the output voltage swing. The conversion gain determines how much voltage change will occur at the sensor output for the collection of one electron on the photodetector. The output voltage swing specifies the possible readout voltage range for the sensor and is essential for determining the well capacity (the maximum charge-collecting capability of an image sensor) of the pixel. Obviously both parameters are closely dependent on the pixel architecture. For example, for CMOS APS, whose circuit schematics is shown in Figure 2.14, the conversion gain g is

$$g = \frac{q}{C_D} \tag{2.32}$$

with q being the electron charge and C_D the photodetector capacitance. The voltage swing is

$$\begin{aligned} v_s &= v_{omax} - v_{omin} \\ &= (v_{DD} - v_{TR} - v_{GSF}) - (v_{bias} - v_{TB}) \end{aligned} \tag{2.33}$$

where v_{TR} and v_{TB} are the threshold voltages of reset and bias transistors, respectively. v_{GSF} is the gate-source voltage of the follower transistor. Notice that all the variables used in the above equations can be derived from technology process information if not given directly.

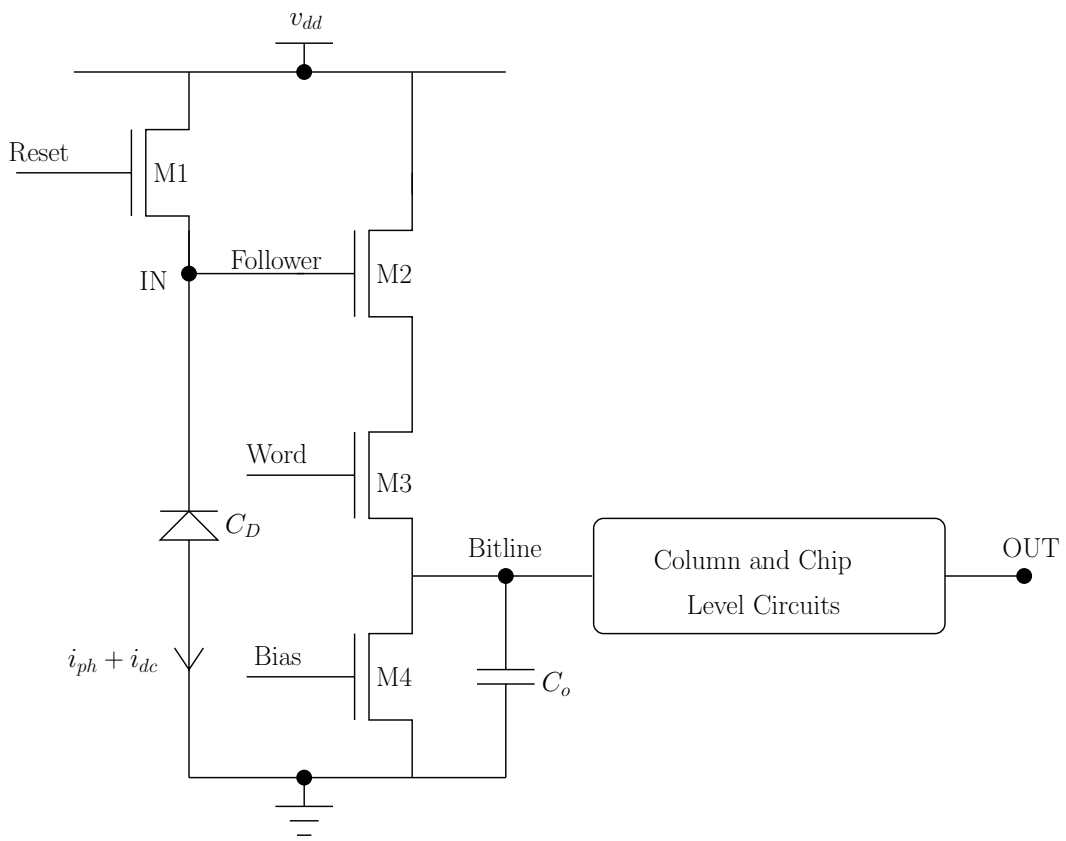


Figure 2.14: CMOS active pixel sensor schematics

2.3 Software Implementation

The simulator is written as a MATLAB toolbox and it consists of many functional routines that follow certain input and output conventions. Structures are used to specify the functional blocks of the system and are passed in and out of different routines. To name a few, a *scene* structure and an *optics* structure are used to describe the scene being imaged and the lens used for the digital camera system, respectively. Each structure contains many different fields, each of which describes a property of the underlying structure. For instance, *optics.fnumber* is used to specify the f/# of the lens. We have carefully structured the simulator into many small modules in hope that future improvements or modifications on the simulator need to be made on relevant modules only without affecting others. An additional advantage of such an organization is that any customization on the simulator is permitted and can be implemented easily.

There are three input structures that need to be defined before the real camera simulation can be carried out. This includes defining a scene, specifying the camera optics and characterizing the image sensor. We will describe how these three input structures are implemented. Once these three structures are completely specified, we can then apply the physical principles as described in Section 2.2 and follow the imaging pipeline to create a camera output image.

2.3.1 Scene

The scene properties are specified in the structure *scene*, which is described in table 2.1. Most of the listed fields in the structure are straightforward, consequently we only mention a few noteworthy ones here. The resolution of a real world scene is infinite, hence we would need an infinite number of points to represent the real scene. Simulation requires digitization, which is an approximation. Such an approximation is reflected in substructure *resolution*, which specifies how fine the sampling of the real scene is, both angularly and spatially. The most crucial information about the scene is contained in *data*, where a three dimensional array is used to specify the scene

radiance in photons at the location of each scene sample and at each wavelength.

Substructure/Field		Class	Unit	Parameter Meaning
distance		double	m	distance between scene and lens
magnification		double	N/A	scene magnification factor
resolution	angular	double	sr	scene angular resolution
	spatial	double	m	scene spatial resolution
height	nRows	integer	N/A	number of rows in the scene
	angular	double	sr	scene vertical angular span
	spatial	double	m	scene vertical dimension
width	nCols	integer	N/A	number of columns in the scene
	angular	double	sr	scene horizontal angular span
	spatial	double	m	spatial horizontal dimension
diagonal	angular	double	sr	scene diagonal angular span
	spatial	double	m	scene diagonal dimension
spatial-Support	rowCo-ordinates	array	m	horizontal and vertical positions of the scene samples
	colCo-ordinates	array	m	
frequency-Support	max-Frequency	double	lp/mm	maximum spatial frequency in the scene
	fx	array	lp/mm	horizontal and vertical spatial frequencies of scene samples
	fy	array	lp/mm	
spectrum	nWaves	integer	N/A	number of wavelengths
	wavelength	array	nm	wavelengths included in data
data	photons	array	$\text{sec}^{-1}\cdot \text{sr}^{-1}\cdot \text{m}^{-2}\cdot \text{nm}^{-1}$	scene radiance in photons

Table 2.1: Scene structure

A scene usually consists of some light sources and some objects that are to be imaged. And the scene radiance can be determined using the following equation,

$$L = \int_{\lambda_{min}}^{\lambda_{max}} L(\lambda)d\lambda = \int_{\lambda_{min}}^{\lambda_{max}} \Phi(\lambda)S(\lambda)d\lambda, \tag{2.34}$$

where L represents the total scene radiance, $L(\lambda)$ is the the scene radiance at each

wavelength, $\Phi(\lambda)$ is the light source radiance, $S(\lambda)$ is the object surface reflectance function, λ_{max} and λ_{min} determine the range of the wavelength which often corresponds to the human's visible wavelength. In order to specify the scene radiance, we need to know both the source radiance and the object surface reflectance. In practice, however, we often do not have all this information. To work with a large set of images, we allow vCam to handle three different types of input data. The first type is hyperspectral images. Hyperspectral images are normal images specified at multiple wavelengths. In terms of dimension, normal images are two-dimensional, while hyperspectral images are three-dimensional with the third dimension representing the wavelength. Having a hyperspectral image is equivalent to knowing the scene radiance $L(\lambda)$ directly without the knowledge of the light source and surface reflectance. Hyperspectral images are typically obtained from tedious measurements that involve measuring the scene radiance at each location and at each wavelength. For this reason, the availability of hyperspectral images is limited. Some calibrated hyperspectral images can be found online [8, 70]. The second type of inputs that vCam handles is B&W images. We normalize a B&W image between 0 and 1. The normalized image is assumed to be the surface reflectance of the object. As a result, the surface reflectance is independent of wavelength. Using a pre-defined light source, we can compute the scene radiance from Equation (2.34). The third type is RGB images. For this type of inputs, we determine the scene radiance by assuming that the image is displayed using a laser display with source wavelengths of 450nm, 550nm and 650nm. These three wavelengths correspond to the three color planes of blue, green and red, respectively. The scene radiance at each wavelength is specified by the relevant color plane and the integration in Equation (2.34) is reduced to a summation of three scene radiance. The last two types of inputs have enabled vCam to cover a vast set of images that can be easily obtained in practice.

2.3.2 Optics

The camera lens modeled in vCam are restricted to diffraction-limited lens for simplicity currently. All the information related to the camera lens is contained in the

structure *optics*, which is further described in Table 2.2. Two out of the three parameters, *fnumber*, *focalLength* and *clearDiameter* need to be specified and the third one can be derived thereafter using Equation (2.13). Function *cos4th* is used to take into account the effect of off-axis illumination and will be computed on-the-fly during simulation as described in Section 2.2. Similarly Function *OTF* specifies the optical modulation transfer function of the lense and is also executed during simulation.

Substructure/Field	Class	Unit	Parameter Meaning
fnumber	double	N/A	f/# of the lens
focalLength	double	m	focal length of the lens
NA	double	N/A	numerical aperture of the lens
clearDiameter	double	m	diameter of the aperture stop
clearAperture	double	m^2	area of the aperture stop
cos4th	function	N/A	function for off-axis image irradiance correction
OTF	function	N/A	function for calculating OTF of the lens
transmittance	array	N/A	transmittance of the lens

Table 2.2: Optics structure

2.3.3 Sensor

An image sensor consists of an array of pixels. To specify an image sensor, it is reasonable to start by modeling a single pixel. Once a pixel is specified, we can arrange a number of pixels together to form an image sensor. Such an arrangement includes both positioning pixels and assigning appropriate color filters to form the desired color filter array pattern. In the next two subsections We will describe how to implement a single pixel and how to form an image sensor with these pixels, respectively.

Implementing a Single Pixel

A pixel on a real image sensor is a physical entity with certain electrical functions. Consequently in order to describe a pixel, both its electrical and geometrical properties need to be specified. A *pixel* structure, as shown in Table 2.3, is used to describe the pixel properties. Sub-structure *GP* describes the pixel geometrical properties, including the pixel size, its positioning relative to adjacent pixels, the photodetector size and position within the pixel. Similarly, sub-structure *EP* specifies the pixel electrical properties, including the dark current density and spectral response of the photodetector, conversion gain and voltage swing of the pixel readout circuitry. Also the parameters used to calculate diffusion MTF is specified in *EP.pd.diffusionMTF* and noise parameters are contained in *EP.noise*. Notice that all the fields under sub-structures *GP* and *EP* are required for the simulator to run successfully. On the other hand, fields under *OP* are optional properties of the pixel. These parameters are the ones that may be helpful in specifying the pixel or may be needed to derive those fundamental pixel parameters, but they themselves are not required for future simulation steps. The fields listed in the table are only examples of what can be used, not necessarily what have to be used. One thing that is worth mentioning is the sub-structure *OP.technology*. It contains essentially all the process information (doping densities, layer dimensions and so on) related to the technology used to build the sensor and it can be used to derive other sensor parameters if necessary.

Implementing an Image Sensor

Once an individual pixel is specified, the next step is to arrange a number of pixels together to form an image sensor. The properties of the image sensor array (ISA) is completely specified with structure *ISA*, which is listed in Table 2.4. Forming an image sensor includes both assigning a position for each pixel and specifying an appropriate color filter according to a color filter array (CFA) pattern. This is described by sub-structure *array*. Fields *DeltaX* and *DeltaY* are the projections of center-to-center distances between adjacent pixels in horizontal and vertical directions. *unitBlock* has to do with the fundamental building blocks of the image sensor array. For instance, a

Substructure/Field			Class	Unit	Parameter Meaning
GP[a]	pixel	width	double	m	pixel width
		height	double	m	pixel height
		gapx	double	m	gap between adjacent pixels
		gapy	double	m	
		area	double	m^2	pixel area
		fillFactor	double	N/A	pixel fill factor
	pd[b]	width	double	m	photodetector width
		height	double	m	photodetector height
		xpos	double	N/A	photodetector position in reference to the pixel upper-left corner
		ypos	double	N/A	
		area	double	m^2	photodetector area
EP[c]	pd	darkCurrent-Density	double	nA/ cm^2	photodetector dark current density
		spectralQE	array	N/A	photodetector spectral response
		diffusionMTF	struc-ture	N/A	information for calculating diffusion MTF
	roc[d]	conversion-Gain	double	V/e-	pixel conversion gain
		voltageSwing	double	V	pixel readout voltage swing
	noise	readNoise	double	e-	read noise level
OP[e]	pixelType		string	N/A	pixel architecture type
	pdType		string	N/A	photodetector type
	pdCap		double	F	photodetector capacitance
	noiseLevel		string	N/A	specify what noise source to be included
	FPN		struc-ture	N/A	information for calculating sensor FPN
	technology		struc-ture	N/A	technology process information

[a]GP: geometrical properties
[b]pd: photodetector
[c]EP: electrical properties
[d]roc: readout circuitry
[e]OP: optional properties

Table 2.3: Pixel structure

Bayer pattern [5] has a building block of 2×2 pixels with 2 green pixels on one diagonal, one blue pixel and one red pixel on the other, as shown in Figure 2.15. Once a *unitBlock* is determined, we can simply replicate these unit blocks and put them side by side to form the complete image sensor array. *config* is a matrix of three columns with the first two columns representing the coordinates of each pixel in absolute units in reference to the upper-left corner of the sensor array. The third column contains the color index for each pixel. Using absolute coordinates to specify the position for each pixel allows vCam to support non-rectangular sampling array patterns such as the Fuji "honeycomb" CFA [99].

The sub-structure *color* determines the color filter properties. Specifically it contains the color filter spectra for the chosen color filters. This information is later combined with the photodetector spectral response to form the overall sensor spectral response. Structure *pixel* is also attached here as a field to *ISA*. Doing so allows compact arguments to be passed in and out of different functions.

2.3.4 From Scene to Image

Given the scene, the optics and the sensor information, we are ready to estimate the image at the sensor output. This has been described in detail in Section 2.2. The simulation process can be viewed as two separate steps. First, using the scene and optics information, we can produce the spectral image right on the image sensor but before the capture, this is essentially the optical pipeline. Then the electrical pipeline applies and an image represented as analog electrical signals is generated. Camera controls such as auto exposure are also included in vCam.

2.3.5 ADC, Post-processing and Image Quality Evaluation

After the detected light signal is read out, many post-processing steps are applied. First comes the analog-to-digital conversion, followed by a number of color processing steps such as color demosaicing, color correction, white balancing and so on. Other steps such as gamma correction may also be included. At the end to evaluate the

Substructure/Field			Class	Unit	Parameter Meaning
pixel			structure	N/A	see Table 2.3
array	pattern		string	N/A	CFA pattern type
	size		array	N/A	2x1 array specifying number of pixels
	dimension		array	m	2x1 array specifying size of the sensor
	DeltaX		double	m	center-to-center distance between adjacent pixels in horizontal and vertical directions
	DeltaY		double	m	
	unit-Block	nRows	integer	N/A	size of fundamental building block for the chosen array pattern
		nCols	integer	N/A	
		config	array	N/A	(Number of pixels)x3 array, where the 1st two columns specify pixel positions in reference to upper-left corner and the last column specifies the color. "Number of pixels" refers to the pixels in the entire sensor in *array.config* and only those in the fundamental building block for *array.unitBlock.config*
	config		array	N/A	
color	filterType		string	N/A	color filter type
	filterSpectra		array	N/A	color filter response

Table 2.4: ISA structure

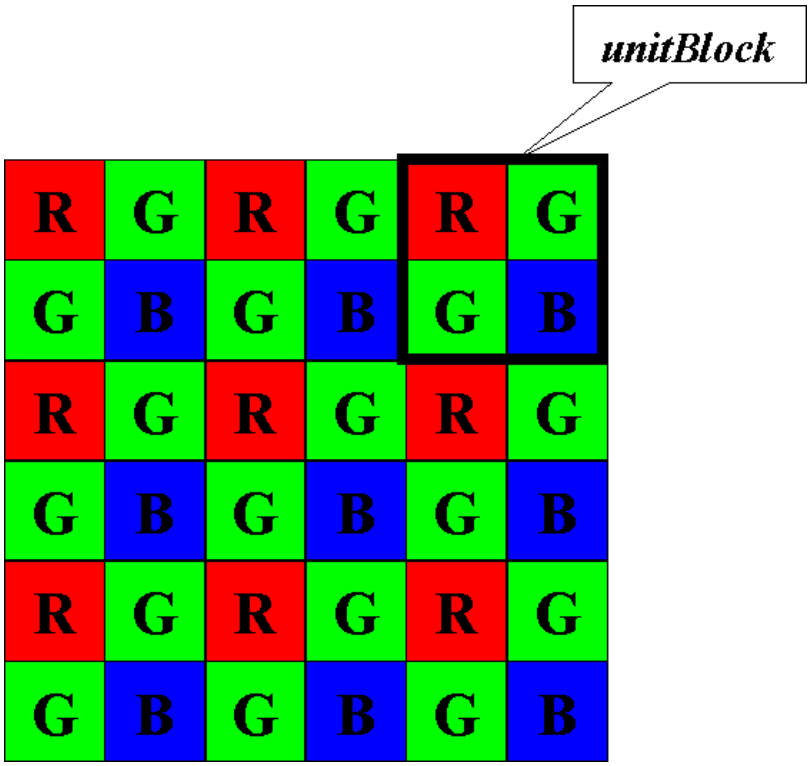


Figure 2.15: A color filter array (CFA) example - Bayer pattern

image quality, metrics such as MSE, S-CIELAB [109] can be used. All these processings are organized as functions that can be easily added, removed or replaced. Basically the idea here is that as soon as the sensor output is digitized, any digital processing, whether it is color processing, image processing or image compression, can be realized. So the post-processing simulation really consists of numerous processing algorithms, of which we only implemented a few in our simulator to complete the signal path. For ADC, we currently support linear and logarithmic scalar quantization. Bilinear interpolation [21] is the color demosaicing algorithm adopted for Bayer color filter array pattern. A gray-world assumption [11] based white balancing algorithm is implemented, "bright block" method [89], which is an extension to the gray-world algorithm, is also supported. Because of the modular nature of vCam, it is straightforward to insert any new processing steps or algorithms from the rich color/image

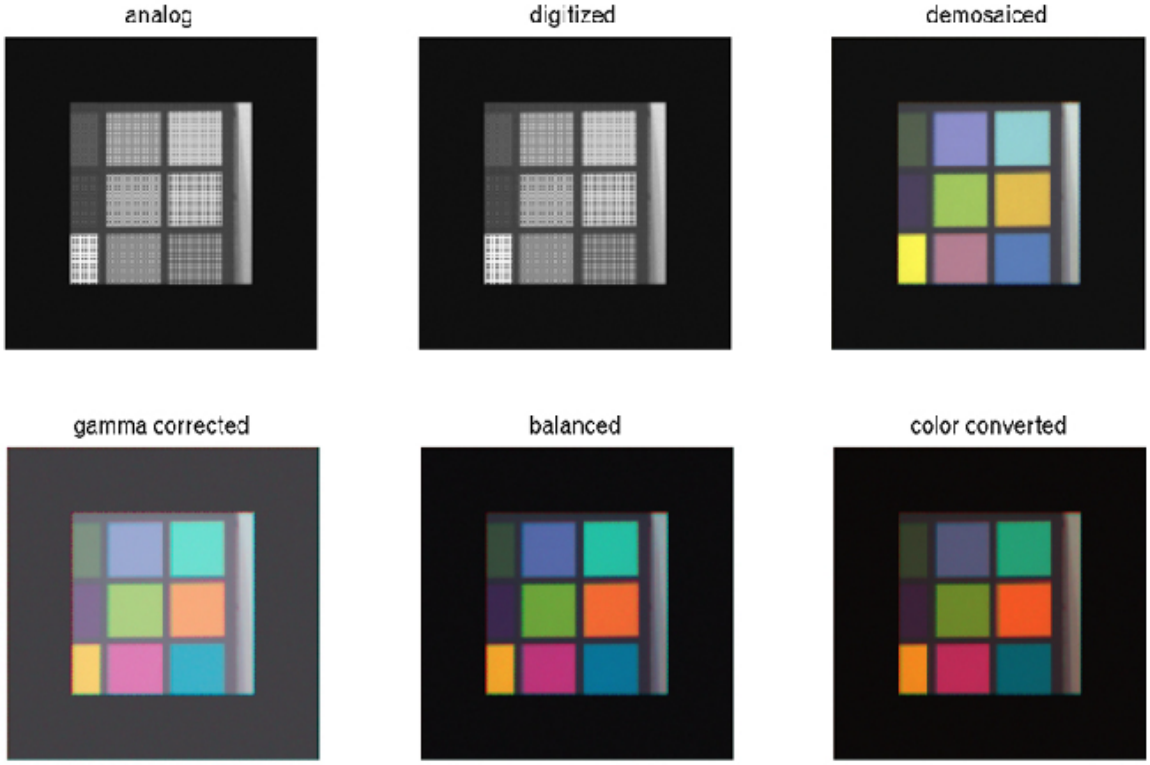


Figure 2.16: An Post-processing Example

processing field into this post-processor. Figure 2.16 shows an example from vCam, where an 8-bit linear quantizer, bilinear interpolation on a Bayer pattern, white balancing based on gray-world assumption, and a gamma correction with gamma value of 2.2 are used.

2.4 vCam Validation

vCam is a simulation tool and it is intended for sensor designers or digital system designers to gain more insight about how different aspects of the camera system perform. Before we can start trusting the simulation results, however, validation with real setups in practice is required. As a partial fulfillment of such a purpose, we validated the vCam using a $0.18\mu m$ CMOS APS test structure [88] built in our group. The vCam simulates a complex system with a rather long signal path, a complete validation on the entire signal chain, though ideal, is not crucial in correlating the simulation results with actual systems. For instance, all the post-processing steps are standard digital processings and need not to be validated. So instead we chose to validate the analog, *i.e.*, sensor operation only, mainly because this is where the real sensing action occurs and the multiple (spectral, spatial and temporal) integrations involved impose the biggest uncertainty in the entire simulator. Furthermore, since a single pixel is really the fundamental element inside an image sensor, we will concentrate on validating the operations of a signal pixel. In the following subsections, we will describe our validation setup and present results obtained.

2.4.1 Validation Setup

Figure 2.17 shows the experimental setup used in our validation process. The spectroradiometer is used to measure the light irradiance on the surface of the sensor. It measures the irradiance in unit of $[W/m^2 \cdot sr]$ for every wavelength band of 4nm wide in the visible range from 380nm to 780nm. A reference white patch is placed at the sensor location during the irradiance measurement, and the light irradiance is determined from the spectroradiometer data assuming the white patch has perfect reflectance. The light irradiance measurement is further verified by a calibrated photodiode. We obtain the spectrum response of the calibrated photodiode from its spec sheet and together with the measured light irradiance, we compute the photocurrent flowing through the photodiode under illumination using Equation (2.24). On the other hand, the photocurrent can be simultaneously measured with a standard amp

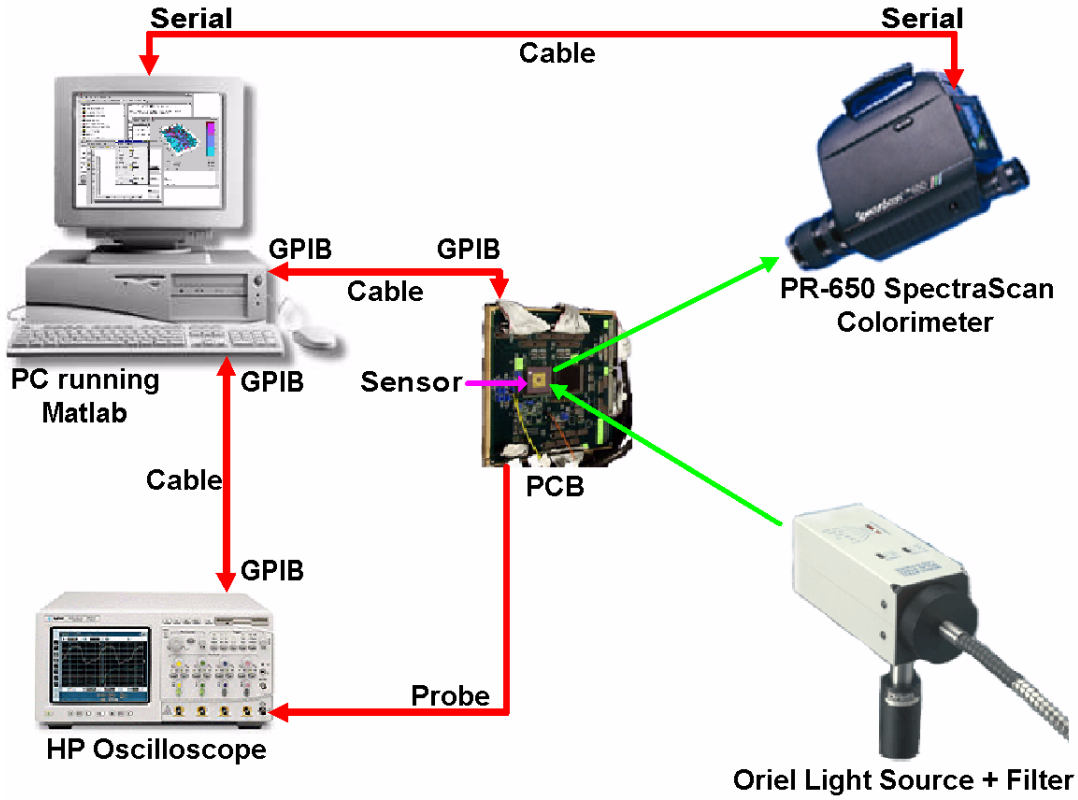


Figure 2.17: vCam validation setup

meter. The discrepancy between the two photocurrent measurements is within 2%, which assures us high confidence on our light irradiance measurements.

The validation is done using a $0.18\mu m$ CMOS APS test structure with a $4 \times 4\mu m^2$ n+/psub photodiode. The schematic of the test structure is shown in Figure 2.18. First of all, by setting *Reset* to V_{dd} and sweeping V_{set}, we are able to measure the transfer curve between V_{in} and V_{out}. Given the known initial reset voltage on the photodetector at the beginning of integration, we are able to predict V_{in} at the end of integration from vCam. Together with the transfer curve, we can decide the estimated V_{out} value, finally this estimated value is compared with the direct measurement from the HP digital oscilloscope.

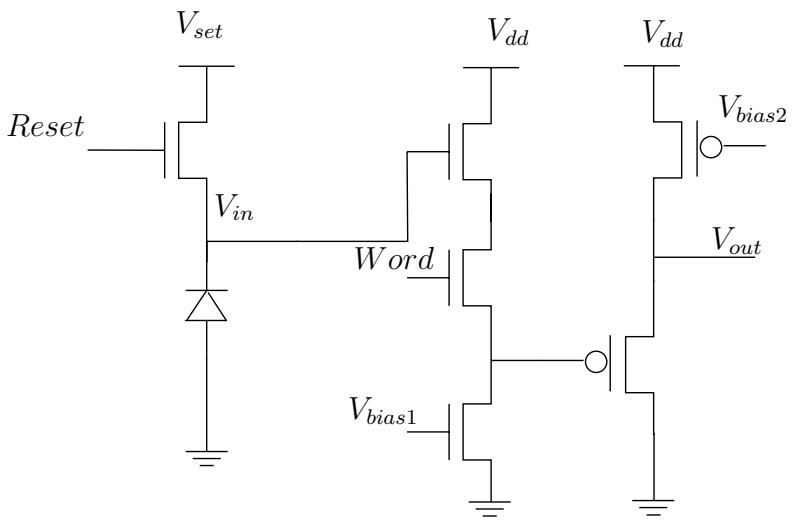


Figure 2.18: Sensor test structure schematics

2.4.2 Validation Results

We performed the measurements on the test structure aforementioned. We experimented with a day light filter, a blue light filter, a green light filter, a red light filter and no filter in front of the light source. For each filter, we also tried three different light intensity levels. Figure 2.19 shows the validation results from these measurements. It can be seen that the majority of the discrepancy between the estimation and the experimental measurements are within ±5%, while all of them are within ±8%. Thus vCam's electrical pipeline has been shown to produce results well correlated to actual experiments.

2.5 Conclusion

This chapter is aimed at providing detailed description of a Matlab-based camera simulator that is capable of simulating the entire image capture pipeline, from photons at the scene to rendered digital counts at the output of the camera. The simulator vCam includes models for the scene, optics and image sensor. The physical models upon

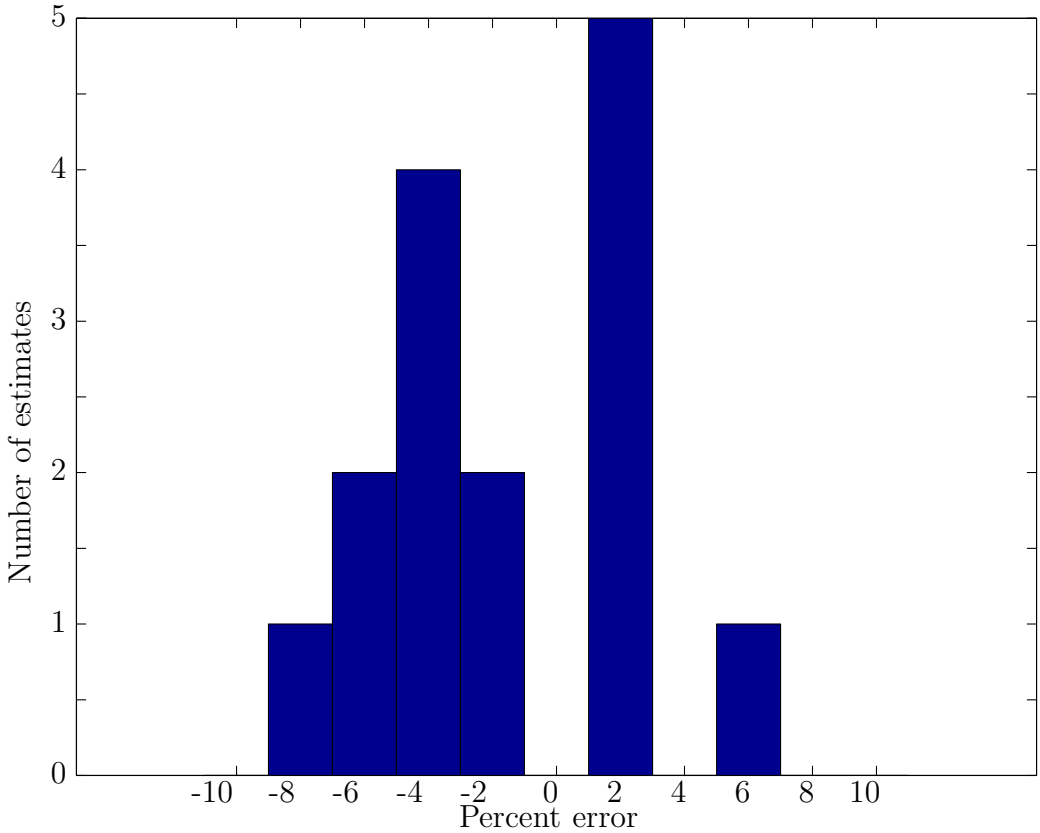


Figure 2.19: Validation results: histogram of the % error between vCam estimation and experiments

which vCam is built are presented in two categories, optical pipeline and electrical pipeline. Implementation of the vCam is also discussed with emphasis on setting up the simulation environment, the scene, the optics, the image sensor and the camera control parameters. Finally, partial validation on vCam is demonstrated via a 0.18μ CMOS APS test structure.

Chapter 3

Optimal Pixel Size

After introducing vCam in the previous chapter, we are now ready to look at how the simulator can help us in camera system design. The rest of this dissertation will describe two such applications of vCam. The first application is selecting optimal pixel sizes for the image sensor.

3.1 Introduction

Pixel design is a crucial element of image sensor design. After deciding on the photodetector type and pixel architecture, a fundamental tradeoff must be made to select pixel size. Reducing pixel size improves the sensor by increasing spatial resolution for fixed sensor die size, which is typically dictated by the optics chosen. Increasing pixel size improves the sensor by increasing dynamic range and signal-to-noise ratio. Since spatial resolution, dynamic range, and SNR are all important measures of an image sensor's performance, special attention must be paid to select an optimal pixel size that can strike the balance among these performance measures for a given set of process and imaging constraints. The goal of our work is to understand the tradeoffs involved in selecting a pixel size and to specify a method for determining such an

optimal pixel size. We begin our study by demonstrating the tradeoffs quantitatively in the next section.

In older process technologies, the selection of an optimal pixel size may not have been important, since the transistors in the pixel occupied such a large area relative to the photodetector area that the designer could not increase the photodetector size (and hence the fill factor) without making pixel size unacceptably large. For an example, an active pixel sensor with a $20 \times 20\mu m^2$ pixel built in a 0.9μ CMOS process was reported to achieve a fill factor of 25% [28]. To increase the fill factor to a decent 50%, the pixel size needs to be larger than $40\mu m$ on a side. This would make the pixel, which is initially not small, too big and thus unacceptable for most practical applications. As process technology scales, however, the area occupied by the pixel transistors decreases, providing more freedom to increase the fill factor while maintaining an acceptably small pixel size. As a result of this new flexibility, it is becoming more important to use a systematic method to determine the optimal pixel size.

It is difficult to determine an optimal pixel size analytically because the choice depends on sensor parameters, imaging optics characteristics, and elements of human perception. In this chapter we describe a methodology for using a digital camera simulator [13, 15] and the S-CIELAB metric [109] to examine how pixel size affects image quality. To determine the optimal pixel size, we decide on a sensor area and create a set of simulated images corresponding to a range of pixel sizes. The difference between the simulated output image and a perfect, noise-free image is measured using a spatial extension of the CIELAB color metric, S-CIELAB. The optimal pixel size is obtained by selecting the pixel size that produces the best rendered image quality as measured by S-CIELAB.

We illustrate the methodology by applying it to CMOS APS, using key parameters for CMOS process technologies down to 0.18μ. The APS pixel under consideration is the standard n+/psub photodiode, three transistors per pixel circuit shown in Figure 3.1. The sample pixel layout [60] achieves 35% fill factor and will be used as a basis for determining pixel size for different fill factors and process technology generations.

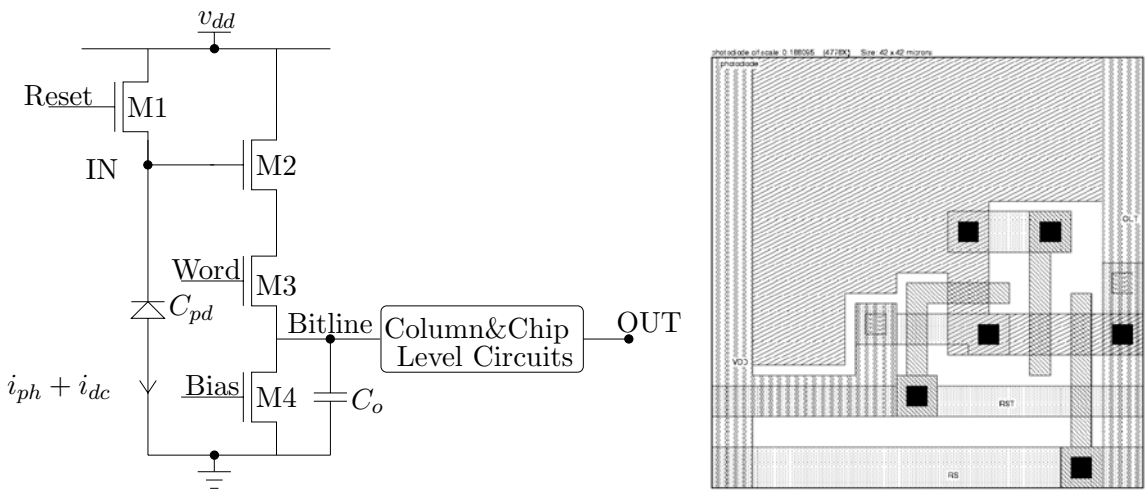


Figure 3.1: APS circuit and sample pixel layout

The remainder of this chapter is organized as follows. In Section 3.2 we analyze the effect of pixel size on sensor performance and system MTF. In Section 3.3 we describe the methodology for determining the optimal pixel size given process technology parameters, imaging optics characteristics, and imaging constraints such as illumination range, maximum acceptable integration time and maximum spatial resolution. The simulation conditions and assumptions are stated in Section 3.4. In Section 3.5 we first explore this methodology using the CMOS APS 0.35μ technology. We then investigate the effect of a number of sensor and imaging parameters on pixel size. In Section 3.6 we use our methodology and a set of process parameters to investigate the effect of technology scaling on optimal pixel size.

3.2 Pixel Performance, Sensor Spatial Resolution and Pixel Size

In this section we demonstrate the effect of pixel size on sensor dynamic range, SNR, and camera system MTF. For simplicity we assume square pixels throughout this

chapter and define pixel size to be the length of the side. The analysis in this section motivates the need for a methodology for determining an optimal pixel size.

3.2.1 Dynamic Range, SNR and Pixel Size

Dynamic range and SNR are two useful measures of pixel performance. Dynamic range quantifies the ability of a sensor to image highlights and shadows; it is defined as the ratio of the largest non-saturating current signal i_{max}, *i.e.* input signal swing, to the smallest detectable current signal i_{min}, which is typically taken as the standard deviation of the input referred noise when no signal is present. Using this definition and the sensor noise model it can be shown [101] that DR in dB is given by

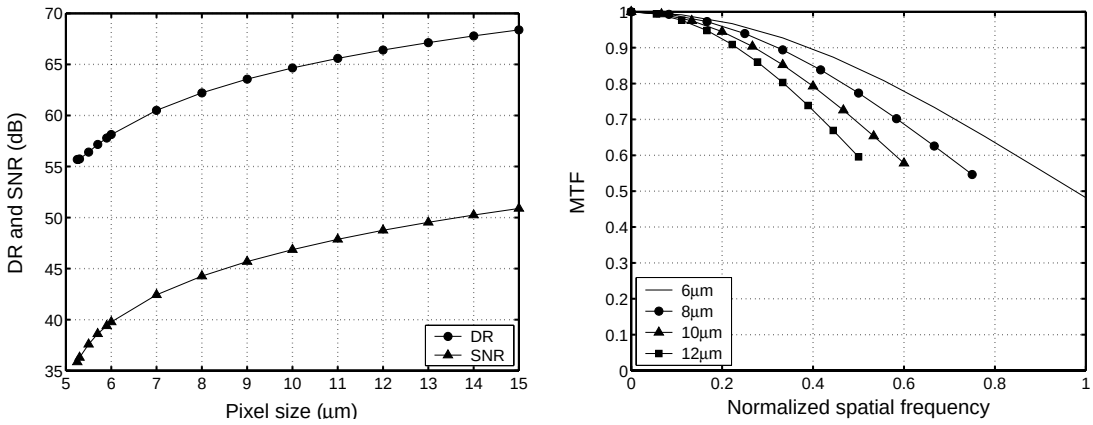
$$\mathrm{DR} = 20\log_{10}\frac{i_{max}}{i_{min}} = 20\log_{10}\frac{q_{max} - i_{dc}t_{int}}{\sqrt{\sigma_r^2 + qi_{dc}t_{int}}}, \tag{3.1}$$

where q_{max} is the well capacity, q is the electron charge, i_{dc} is the dark current, t_{int} is the integration time, σ_r^2 is the variance of the temporal noise, which we assume to be approximately equal to kTC, *i.e.* the reset noise when correlated double sampling is performed [87]. For voltage swing V_s and photodetector capacitance C the maximum well capacity is $q_{max} = CV_s$.

SNR is the ratio of the input signal power and the average input referred noise power. As a function of the photocurrent i_{ph}, SNR in dB is [101]

$$\mathrm{SNR}(i_{ph}) = 20\log_{10}\frac{i_{ph}t_{int}}{\sqrt{\sigma_r^2 + q(i_{ph} + i_{dc})t_{int}}}. \tag{3.2}$$

Figure 3.2(a) plots DR as a function of pixel size. It also shows SNR at 20% of the well capacity versus pixel size. The curves are drawn assuming the parameters for a typical 0.35μ CMOS process which can be seen later in Figure 3.5, and integration time $t_{int} = 30ms$. As expected, both DR and SNR increase with pixel size. DR increases roughly as the square root of pixel size, since both C and reset noise (kTC)



(a) (b)

Figure 3.2: (a) DR and SNR (at 20% well capacity) as a function of pixel size. (b) Sensor MTF (with spatial frequency normalized to the Nyquist frequency for 6μm pixel size) is plotted assuming different pixel sizes.

increase approximately linearly with pixel size. SNR also increases roughly as the square root of pixel size since the RMS shot noise increases as the square root of the signal. These curves demonstrate the advantages of choosing a large pixel. In the following subsection, we demonstrate the disadvantages of a large pixel size, which is the reduction in spatial resolution and system MTF.

3.2.2 Spatial Resolution, System MTF and Pixel Size

For a fixed sensor die size, decreasing pixel size increases pixel count. This results in higher spatial sampling and a potential improvement in the system's modulation transfer function provided that the resolution is not limited by the imaging optics. For an image sensor, the Nyquist frequency is one half of the reciprocal of the center-to-center spacing between adjacent pixels. Image frequency components above the Nyquist frequency can not be reproduced accurately by the sensor and thus create aliasing. The system MTF measures how well the system reproduces the spatial structure of the input scene below the Nyquist frequency and is defined to be the ratio of the output modulation to the input modulation as a function of input spatial

frequency [46, 91].

It is common practice to consider the system MTF as the product of the optical MTF, geometric MTF, and diffusion MTF [46]. Each MTF component causes low pass filtering, which degrades the response at higher frequencies. Figure 3.2(b) plots system MTF as a function of the input spatial frequency for different pixel sizes. The results are again for the aforementioned 0.35μ process. Note that as we decrease pixel size the Nyquist frequency increases and MTF improves. The reason for the MTF improvement is that reducing pixel size reduces the low pass filtering due to geometric MTF.

In summary, a small pixel size is desirable because it results in higher spatial resolution and better MTF. A large pixel size is desirable because it results in better DR and SNR. Therefore, there must exist a pixel size that strikes a compromise between high DR and SNR on the one hand, and high spatial resolution and MTF on the other. The results so far, however, are not sufficient to determine such an optimal pixel size. First it is not clear how to tradeoff DR and SNR with spatial resolution and MTF. More importantly, it is not clear how these measures relate to image quality, which should be the ultimate objective of selecting the optimal pixel size.

3.3 Methodology

In this section we describe a methodology for selecting the optimal pixel size. The goal is to find the optimal pixel size for a given process parameters, sensor die size, imaging optics characteristics and imaging constraints. We do so by varying pixel size and thus pixel count for the given die size, as illustrated in Figure 3.3. Fixed die size enables us to fix the imaging optics. For each pixel size (and count) we use vCam with a synthetic contrast sensitivity function (CSF) [12] scene, as shown in Figure 3.4 to estimate the resulting image using the chosen sensor and imaging optics. The rendered image quality in terms of the S-CIELAB ΔE metric is then determined. The experiment is repeated for different pixel sizes and the optimal

pixel size is selected to achieve the highest image quality.

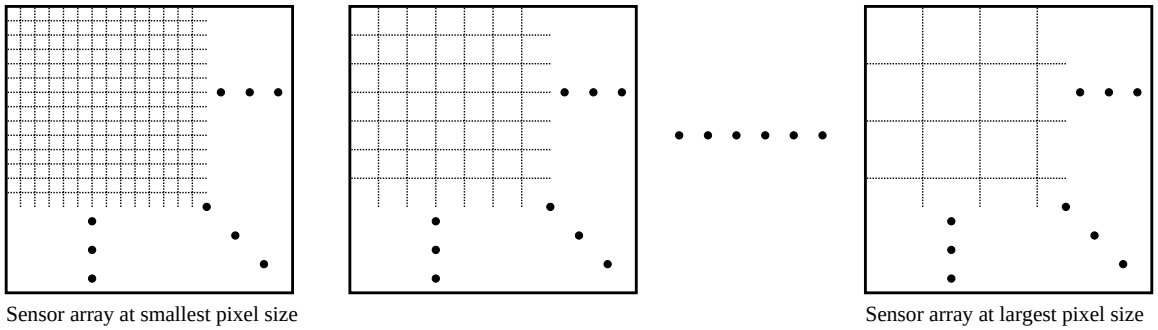


Figure 3.3: Varying pixel size for a fixed die size

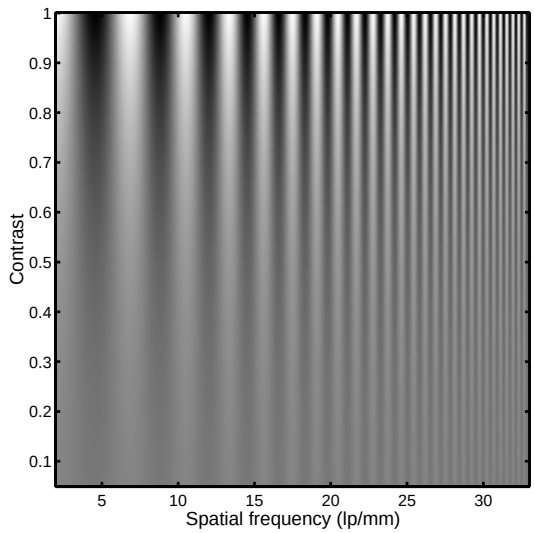


Figure 3.4: A synthetic contrast sensitivity function scene

The information on which the simulations are based is as follows :

- A list of the sensor parameters for the process technology.
- The smallest pixel size and the pixel array die size.
- The imaging optics characterized by focal length f and $f/\#$.

- The maximum acceptable integration time.
- The highest spatial frequency desired.
- Absolute radiometric or photometric scene parameters.
- Rendering model including viewing conditions and display specifications

The camera simulator [13, 15], which has been thoroughly discussed in the previous chapter, provides models for the scene, the imaging optics, and the sensor. The imaging optics model accounts for diffraction using a wavelength-dependent MTF and properly converts the scene radiance into image irradiance taking into consideration off-axis irradiance. The sensor model accounts for the photodiode spectral response, fill factor, dark current sensitivity, sensor MTF, temporal noise, and FPN. Exposure control can be set either by the user or by an automatic exposure control routine, where the integration time is limited to a maximum acceptable value. The simulator reads spectral scene descriptions and returns simulated images from the camera.

For each pixel size, we simulate the camera response to the test pattern shown in Figure 3.4. This pattern varies in both spatial frequency along the horizontal axis and in contrast along the vertical axis. The pattern was chosen firstly because it spans the frequency and contrast ranges of normal images in a controlled fashion. These two parameters correspond well with the tradeoffs for spatial resolution and dynamic range that we observe as a function of pixel size. Secondly, image reproduction errors at different positions within the image correspond neatly to evaluations in different spatial-contrast regimes, making analysis of the simulated images straightforward.

In addition to the simulated camera output image, the simulator also generates a "perfect" image from an ideal (*i.e.* noise-free) sensor with perfect optics. The simulated output image and the "perfect" image are compared by assuming that they are rendered on a CRT display, and this display is characterized by its phosphor dot pitch and transduction from digital counts to light intensity. Furthermore, we assume the same white point for the monitor and the image. With these assumptions, we use the S-CIELAB ΔE metric to measure the point by point difference between the simulated and perfect images.

The image metric S-CIELAB [109] is an extension of the CIELAB ΔE metric, which is one of the most widely used perceptual color fidelity metric, given as part of the CIELAB color model specifications [18]. The CIELAB ΔE metric is only intended to be used on large uniform fields. S-CIELAB, however, extends the ΔE metric to images with spatial details. In this metric, images are first converted to a representation that captures the response of the photoreceptor mosaic of the eye. The images are then convolved with spatial filters that account for the spatial sensitivity of the visual pathways. The filtered images are finally converted into the CIELAB format and perceptual distances are measured using the conventional ΔE units of the CIELAB metric. In this metric, one unit represents approximately the threshold detection level of the difference under ideal viewing conditions. We apply S-CIELAB on gray scale images by considering each gray scale image as a special color image with identical color planes.

3.4 Simulation Parameters and Assumptions

In this section we list the key simulation parameters and assumptions used in this study.

- Fill factors at different pixel sizes are derived using the sample APS layout in Figure 3.1 as the basis and their dependences on pixel sizes for each technology are plotted in Figure 3.5.
- Photodetector capacitance and dark current density information are obtained from HSPICE simulation and their dependencies on pixel sizes for each technology are again plotted in Figure 3.5.
- Spectral response in Figure 3.5 is first obtained analytically [1] and then scaled to match QE from real data [88, 95].
- Voltage swings for each technology are calculated using the APS circuit in Figure 3.1 and are shown in table below. Note that for technologies below 0.35μ, we have assumed that the power supply voltage stays one generation behind.

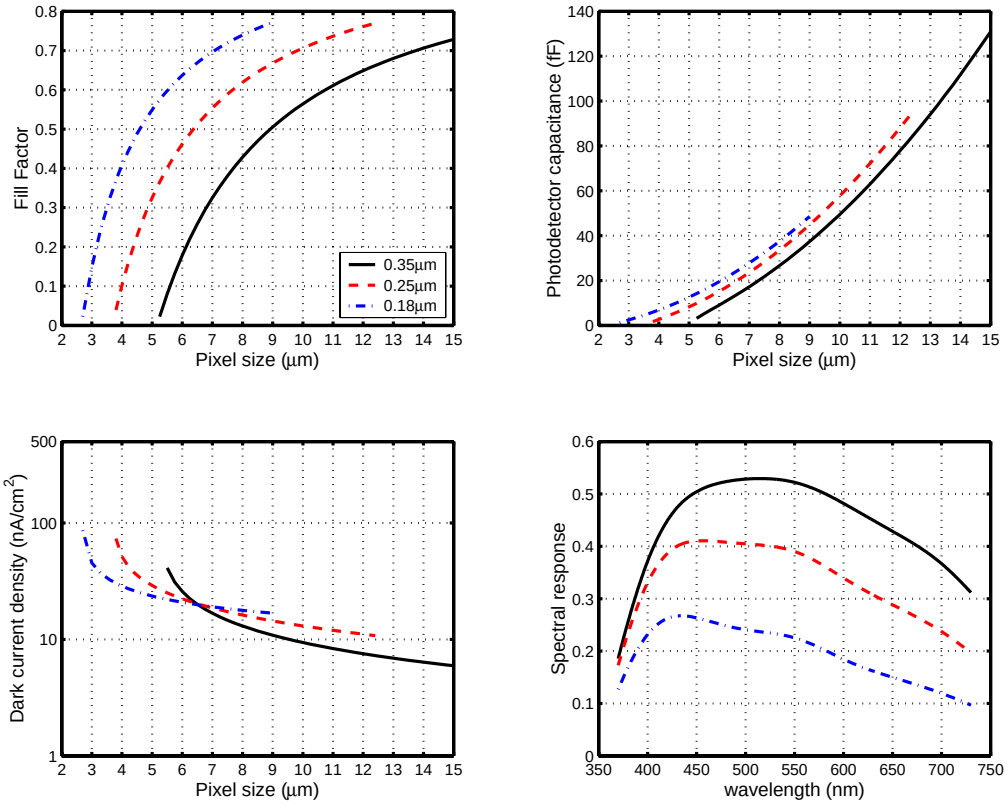


Figure 3.5: Sensor capacitance, fill factor, dark current density and spectral response information

Technology	Voltage Supply (volt)	Voltage swing (volt)
0.35μm	3.3	1.38
0.25μm	3.3	1.67
0.18μm	2.5	1.12

- Other device and technology parameters when needed can be estimated [93].
- The smallest pixel size in μm and the corresponding 512×512 pixel array die size in mm. The array size limit is dictated by camera simulator memory and speed considerations. The die size is fixed throughout the simulations, while pixel size is increased. The smallest pixel size chosen corresponds to a very low fill factor, *e.g.* 5%.

- The imaging optics are characterized by two parameters, their focal length f and $f/\#$. The optics are chosen to provide a full field-of-view (FOV) of 46°. This corresponds to the FOV obtained when using a 35mm SLR camera with a standard objective. Fixing the FOV and the image size (as determined by the die size) enables us to determine the focal length, *e.g.* $f = 3.2mm$ for the simulations of 0.35μ technology. The $f/\#$ is fixed at 1.2.
- The maximum acceptable integration time is fixed at 100ms.
- The highest spatial frequency desired in lp/mm. This determines the largest acceptable pixel size so that no aliasing occurs, and is used to construct the synthetic CSF scene.
- Absolute radiometric or photometric range values for the scene
  - radiance: up to 0.4 W/(sr $\cdot$m^2)
  - luminance: up to 100 cd/m^2
- Rendering model: The simulated viewing conditions were based on a monitor with 72 dots per inch viewed at a distance of 18 inches. Hence, the 512x512 image spans 7.1 inches (21.5 deg of visual angle). We assume that the monitor white point, *i.e.* [R G B] = [111], is also the observer's white point. The conversion from monitor RGB space to human visual system LMS space is performed using the L, M, and S cone response as measured by Smith-Pokorny [77] and the spectral power density functions of typical monitor phosphors.

3.5 Simulation Results

Figure 3.6 shows the simulation results for an 8μm pixel, designed in a 0.35μ CMOS process, assuming a scene luminance range up to 100 cd/m^2 and a maximum integration time of 100ms. The test pattern includes spatial frequencies up to 33 lp/mm, which corresponds to the Nyquist rate for a 15μm pixel. Shown are the perfect CSF

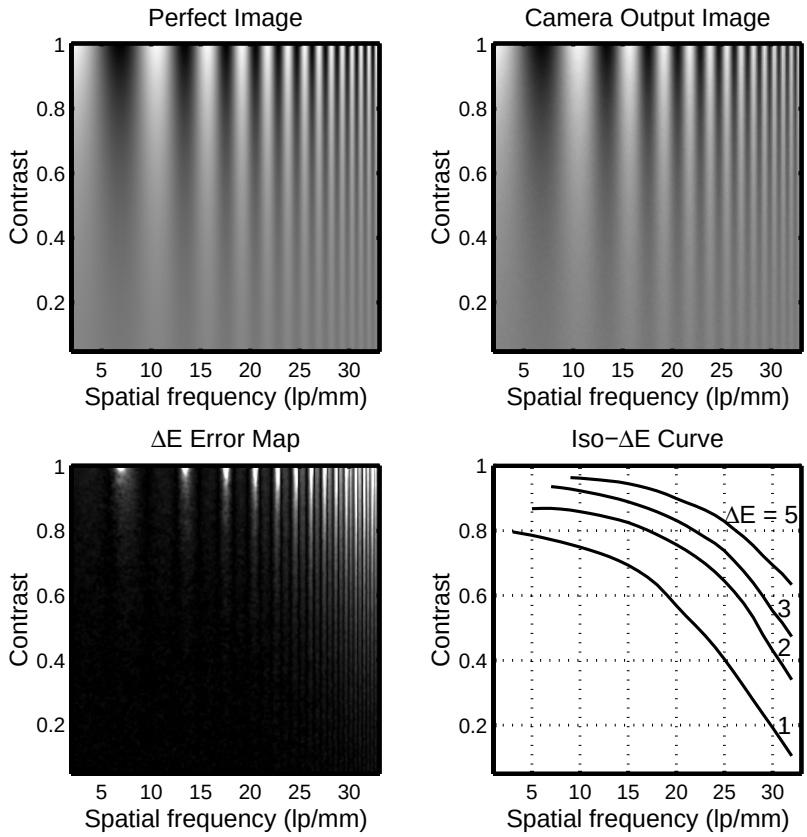


Figure 3.6: Simulation result for a 0.35μ process with pixel size of 8μm. For the ΔE error map, brighter means larger error

image, the output image from the camera simulator, the ΔE error map obtained by comparing the two images, and a set of iso-ΔE curves. Iso-ΔE curves are obtained by connecting points with identical ΔE values on the ΔE error map. Remember that larger values represent higher error (worse performance).

The largest S-CIELAB errors are in high spatial frequency and high contrast regions. This is consistent with the sensor DR and MTF limitations. For a fixed spatial frequency, increasing the contrast causes more errors because of limited sensor dynamic range. For a fixed contrast, increasing the spatial frequency causes more errors because of more MTF degradations.

Now to select the optimal pixel size for the 0.35μ technology we vary pixel size as

discussed in the Section 3.3. The minimum pixel size, which is chosen to correspond to a 5% fill factor, is 5.3μm. Note that here we are in a sensor-limited resolution regime, *i.e.* pixel size is bigger than the spot size dictated by the imaging optics characteristics. The minimum pixel size results in a die size of 2.7×2.7 mm^2 for a 512×512 pixel array. The maximum pixel size is 15μm with a fill factor of 73%, and corresponds to maximum spatial frequency of 33 lp/mm. The luminance range for the scene is again taken to be within 100 cd/m^2 and the maximum integration time is 100ms.

Figure 3.7 shows the iso-$\Delta E = 3$ curves for three different pixel sizes. Certain conclusions on the selection of optimal pixel size can be readily made from the iso-ΔE curves. For instance, if we use $\Delta E = 3$ as the maximum error tolerance, clearly a pixel size of 8μm is better than a pixel size of 15μm, since the iso-$\Delta E = 3$ curve for the 8μm pixel is consistently higher than that for the 15μm pixel. It is not clear, however, whether a 5.3μm pixel is better or worse than a 15μm pixel, since their iso-ΔE curves intersect such that for low spatial frequencies the 15μm pixel is better while at high frequencies the 5.3μm pixel is better.

Instead of looking at the iso-ΔE curves, we simplify the optimal pixel size selection process by using the mean value of the ΔE error over the entire image as the overall measure of image quality. We justify our choice by performing a statistical analysis of the ΔE error map. This analysis reveals a compact, unimodal distribution which can be accurately described by first order statistics, such as the mean. Figure 3.8 shows mean ΔE versus pixel size and an optimal pixel size can be readily selected from the curve. For the 0.35μ technology chosen the optimal pixel size is found to be 6.5μm with roughly a 30% fill factor.

3.5.1 Effect of Dark Current Density on Pixel Size

The methodology described is also useful for investigating the effect of various key sensor parameters on the selection of optimal pixel size. In this subsection we examine the effect of varying dark current density on pixel size. Figure 3.9 plots the mean ΔE as a function of pixel size for different dark current densities. Note that the optimal

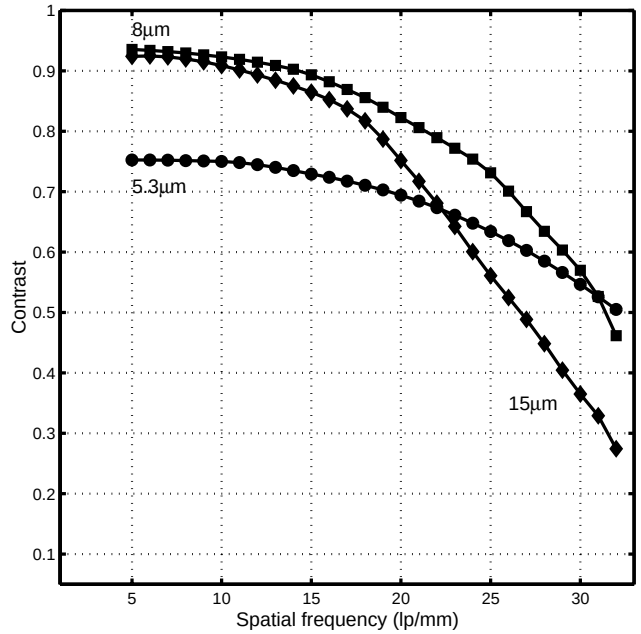


Figure 3.7: Iso-$\Delta E = 3$ curves for different pixel sizes

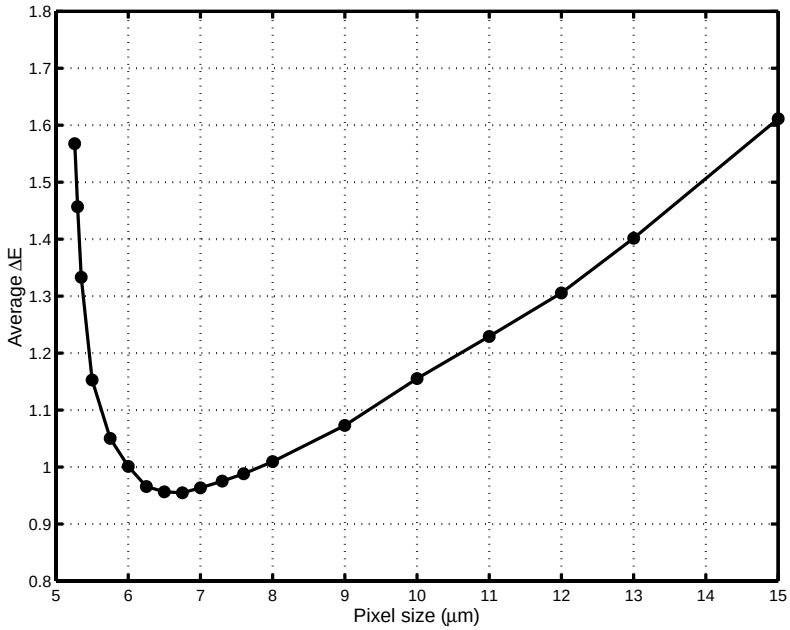


Figure 3.8: Average ΔE versus pixel size

pixel size increases with dark current density increase and in the case when the dark current density is increased by 10 times, the optimal pixel size is increased from 6.5μm to roughly 10μm. This is expected since as dark current increases sensor DR and SNR degrade. This can be somewhat overcome by increasing the well capacity, which is accomplished by increasing the photodetector size thus the pixel size. As expected the mean ΔE at the optimal pixel size also increases with dark current density increase. On the other hand, in the case when the dark current density is reduced by 10 times, not surprisingly the optimal pixel size is also reduced to 5.7μm due to the fact that smaller pixel size can also achieve reasonably good sensor DR and SNR (because we have such a good photodetector) while at the same time improve the resolution.

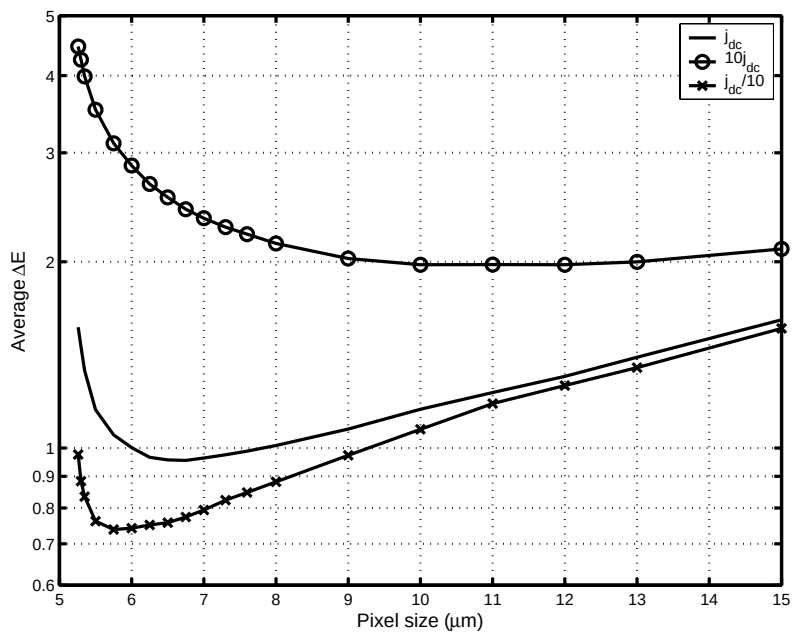


Figure 3.9: Average ΔE vs. Pixel size for different dark current density levels

3.5.2 Effect of Illumination Level on Pixel Size

We look at the effect of varying illumination levels on the selection of optimal pixel size in this subsection. Figure 3.10 plots the mean ΔE as a function of pixel size for

different illumination levels. It appears that illumination level has a similar effect on pixel size as dark current density. Under strong lights, because there are so many photons available, first of all getting good sensor SNR is not a big problem even for small pixels. Moreover, strong lights allow fast exposure which results in small dark noise and increases the sensor dynamic range. This explains why in Figure 3.10 the optimal pixel size is reduced to 5.5μm when the scene luminance level is increased by a factor of 10. On the other hand, when there is not sufficient light, getting good sensor responses becomes more challenging. For example, in order to get the same SNR, under weak lights we have to increase the exposure time which in turn requires us to use a larger pixel if we also want to maintain the same dynamic range. This is why the optimal pixel size is increased to about 10μm when the scene luminance level is reduced by 10 times.

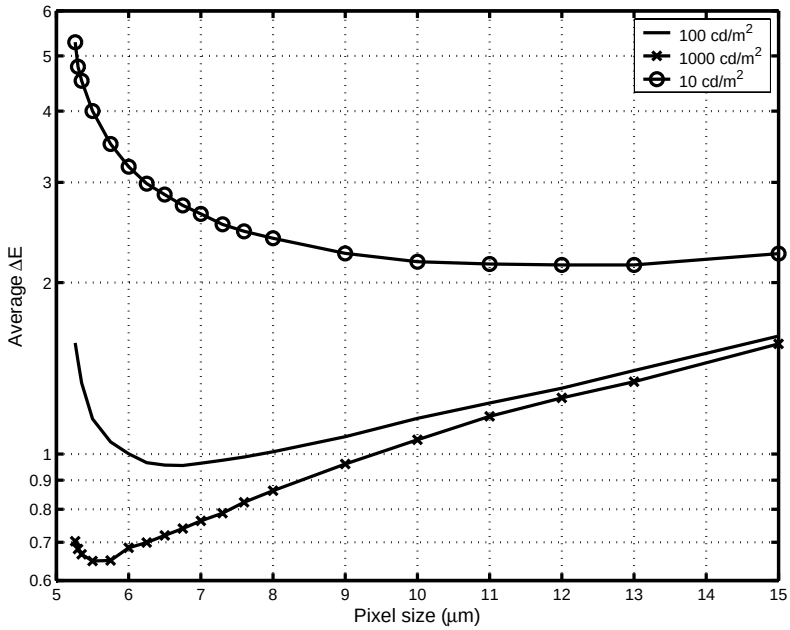


Figure 3.10: Average ΔE vs. Pixel size for different illumination levels

3.5.3 Effect of Vignetting on Pixel Size

Recent study [14] has found that the performance of CMOS image sensors suffers from the reduction of quantum efficiency (QE) due to pixel vignetting, which is the phenomenon that light must travel through a narrow "tunnel" in going from the chip surface to the photodetector in a CMOS image sensor. This is especially problematic for light incident at an oblique angle since the narrow tunnel walls cast a shadow on the photodetector which will severely reduce its effective QE. It is not hard to speculate that vignetting will have some effects on selecting the pixel size since the QE reduction due to pixel vignetting directly depends on the size of the photodetector (or the pixel). In this subsection, we will investigate the effect of pixel vignetting on pixel size following the simple geometrical model proposed by Catrysse *et al.* [14] for characterizing the QE reduction caused by the vignetting.

We use the same 0.35μm CMOS process and a diffraction-limited lens with fixed focal length of 8mm. Figure 3.11 plots the average ΔE error as a function of pixel size with and without the pixel vignetting included. It is observed that pixel vignetting in this case has significantly altered the curve, the optimal pixel size has been increased to 8μm (from 6.5μm) to combat with the reduced QE. This should not come as a surprise since smaller pixels clearly suffer more QE reduction since the tunnels the light has to go through are also narrower. In fact in our simulation, we have observed that the QE reduction for a small off-axis 6μm pixel is as much as 30%, compared with merely an 8% reduction for a 12μm pixel. This is shown in figure 3.12 where we have plotted the normalized QE (with respect to the case with no pixel vignetting) for pixels along the chip diagonal, assuming the center pixel on the chip is on-axis. The figure also reveals that there are larger variations of the QE reduction factors between the pixels on the edges and in the center of the chip for smaller pixel sizes. This explains why there are large increases of average ΔE error for small pixels in figure 3.11. As pixel sizes increase initially, these QE variations between the center and the perimeter pixels are quickly reduced, *i.e.*, the curve is flatter in figure 3.12 for the larger pixel. Consequently the average ΔE error caused by pixel vignetting is also getting smaller.

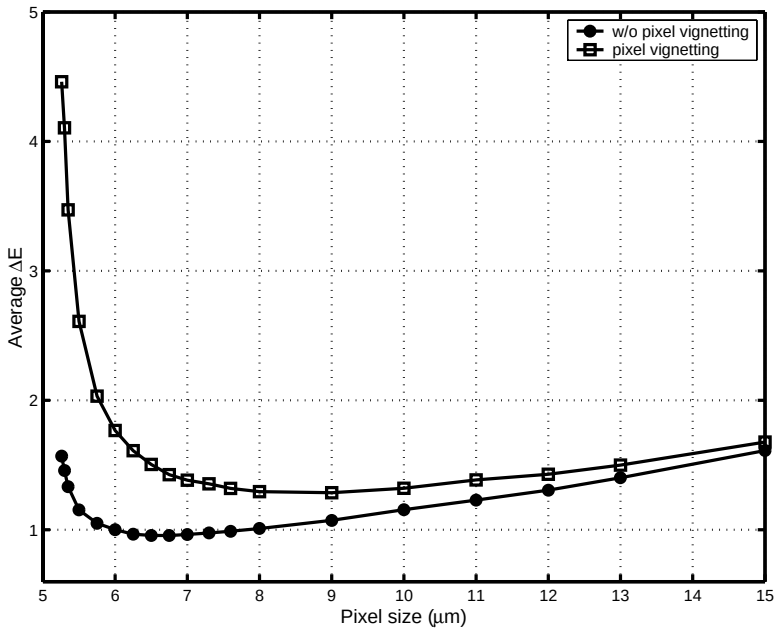


Figure 3.11: Effect of pixel vignetting on pixel size

3.5.4 Effect of Microlens on Pixel Size

Image sensors typically use a microlens [6], which sits directly on top of each pixel, to help direct the photons coming from different angles to the photodetector area. Using a microlens can result an effective increase in fill factor, or in sensor QE and sensitivity. Using our methodology and the microlens gain factor reported by TSMC [96], we performed the simulation for a 0.18μm CMOS process with and without a microlens. The results are shown in Figure 3.13, where as we can see, without a microlens, the optimal pixel size for this particular CMOS technology is 3.5μm; and with a microlens, the optimal pixel size decreases to 3.2μm. This is not surprising since using a microlens effectively increases sensor's QE (or sensitivity) and thus makes it possible to achieve the same DR and SNR with smaller pixels. The overall effect on pixel size due to the microlens is very similar to having stronger light.

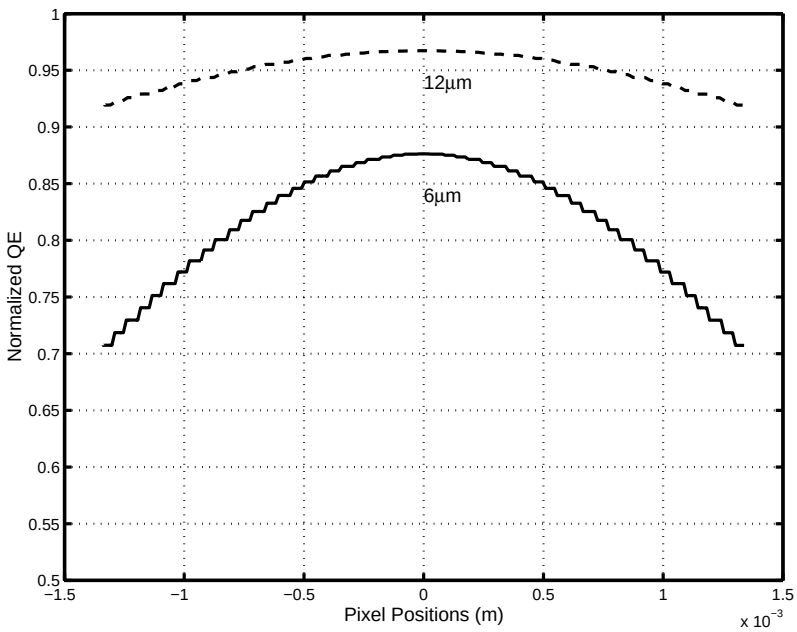


Figure 3.12: Different pixel sizes suffer from different QE reduction due to pixel vignetting. The effective QE, *i.e.*, normalized with the QE without pixel vignetting, for pixels along the chip diagonal is shown. The X-axis is the horizontal position of each pixel with origin taken at the center pixel.

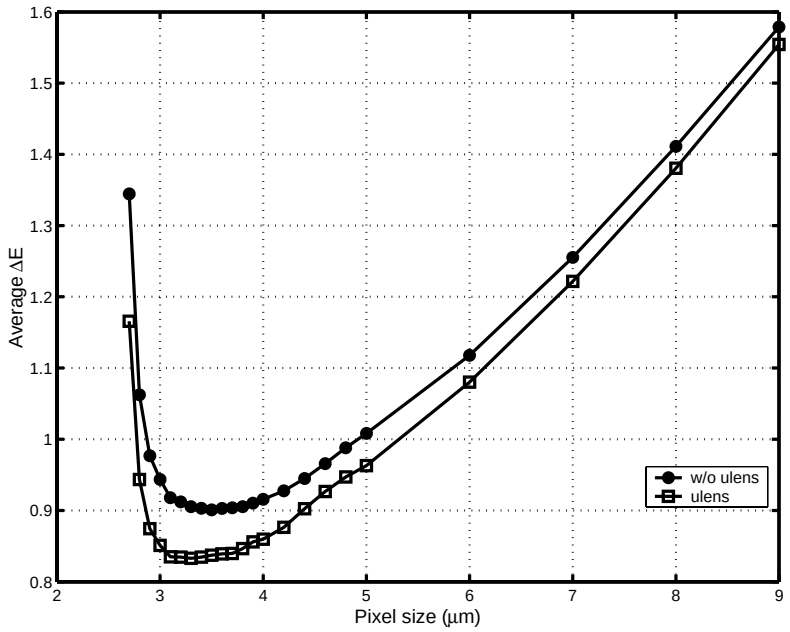


Figure 3.13: Effect of microlens on pixel size

3.6 Effect of Technology Scaling on Pixel Size

How does optimal pixel size scale with technology? We perform the simulations discussed in the previous section for three different CMOS technologies, 0.35μ, 0.25μ and 0.18μ. Key sensor parameters are all described in Section 3.4. The mean ΔE curves are shown in Figure 3.14. It can also be seen from Figure 3.15 that the optimal pixel size shrinks, but at a slightly slower rate than technology.

3.7 Conclusion

We proposed a methodology using a camera simulator, synthetic CSF scenes, and S-CIELAB for selecting the optimal pixel size for an image sensor given process technology parameters, imaging optics parameters, and imaging constraints. We applied the methodology to photodiode APS implemented in CMOS technologies down to 0.18μ and demonstrated the tradeoff between DR and SNR on one hand and

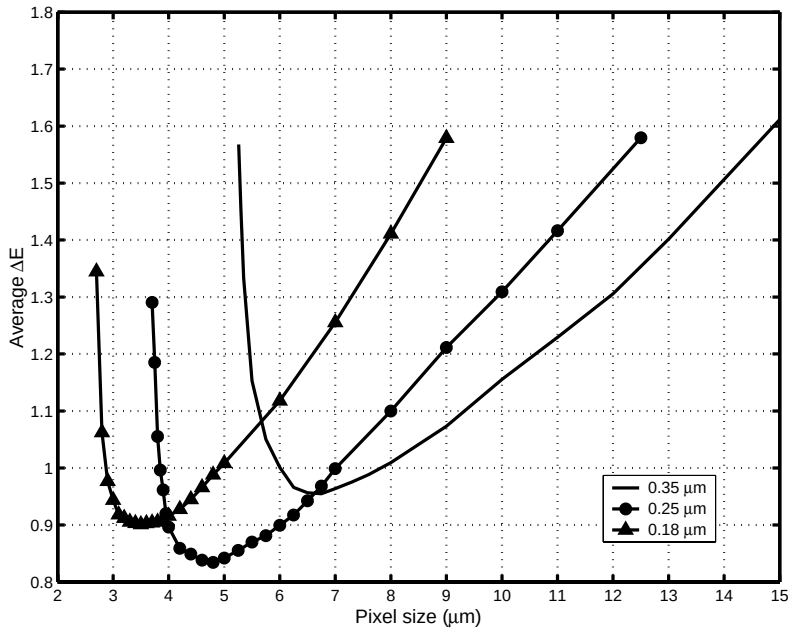


Figure 3.14: Average ΔE versus pixel size as technology scales

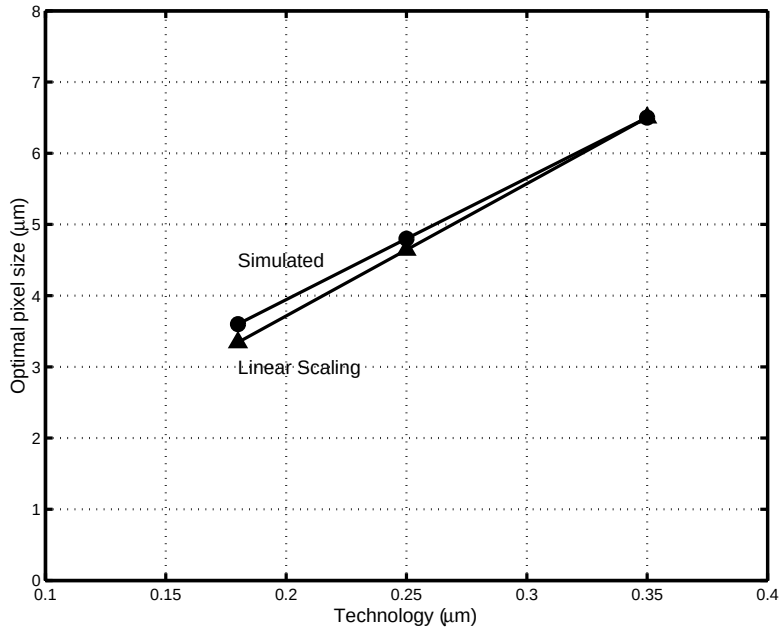


Figure 3.15: Optimal pixel size versus technology

spatial resolution and MTF on the other hand with pixel size. Using the mean ΔE as an image quality metric, we found that indeed an optimal pixel size exists, which represents the optimal tradeoff. For a 0.35μ process we found that a pixel size of around 6.5μm with fill factor 30% under certain imaging optics, illumination range, and integration time constraints achieves the lowest mean ΔE. We found that the optimal pixel size scales with technology, albeit at a slightly slower rate than the technology.

The proposed methodology and its application can be extended in several ways:

- The imaging optics model we used is oversimplified. A more accurate model that includes lens aberrations is needed to find the effect of the lens on the selection of pixel size. This extension requires a more detailed specification of the imaging optics by means of a lens prescription and can be performed by using a ray tracing program [20].

- The methodology needs to be extended to color.

Chapter 4

Optimal Capture Times

The pixel size study as described in the previous chapter is one of those vCam's applications where the entire study is based on the vCam simulation. We now look at another application where we use vCam to demonstrate our theoretical ideas. This brings us to the last part of this dissertation, where we look at the optimal capture time scheduling problem in a multiple capture imaging system.

4.1 Introduction

CMOS image sensors achieving high speed non-destructive readout have been recently reported [53, 43]. As discussed by several authors (e.g. [97, 101]), this high speed readout can be used to extend sensor dynamic range using the multiple-capture technique in which several images are captured during a normal exposure time. Shorter exposure time images capture the brighter areas of the scene while longer exposure time images capture the darker areas of the scene. A high dynamic range image can then be synthesized from the multiple captures by appropriately scaling each pixel's last sample before saturation (LSBS). Multiple capture has been shown [102] to achieve

better SNR than other dynamic range extension techniques such as logarithmic sensors [51] and well capacity adjusting [22].

One important issue in the implementation of multiple capture that has not received much attention is the selection of the number of captures and their time schedule to achieve a desired image quality. Several papers [101, 102] assumed exponentially increasing capture times, while others [55, 44] assumed uniformly spaced captures. These capture time schedules can be justified by certain implementation considerations. However, there has not been any systematic study of how optimal capture times may be determined. By finding optimal capture times, one can achieve the image quality requirements with fewer captures. This is desirable since reducing the number of captures reduces the imaging system computational power, memory, and power consumption as well as the noise generated by the multiple readouts.

To determine the capture time schedule, scene illumination information is needed. In this chapter, we assume known scene illumination statistics, namely, the probability density function (pdf)[1] and formulate multiple capture time scheduling as a constrained optimization problem. We choose as an objective to maximize the average pixel SNR since it provides good indication of image quality. To simplify the analysis, we assume that read noise is much smaller than shot noise and thus can be ignored. With this assumption the LSBS algorithm is optimal with respect to SNR [55]. We use this formulation to establish a general upper bound on achievable average SNR for any number of captures and any scene illumination pdf. We first assume a uniform pdf and show that the average SNR is concave in capture times and therefore the global optimum can be found using well-known convex optimization techniques. For a piece-wise uniform pdf, the average SNR is not necessarily concave. The cost function, however, is a difference of convex (D.C.) function and D.C. or global optimization techniques can be used. We then describe a computationally efficient heuristic scheduling algorithm for piece-wise uniform distributions. This heuristic scheduling algorithm is shown to achieve close to optimal results in simulation. We also discuss how an arbitrary scene illumination pdf may be approximated by piece-wise uniform pdfs. The effectiveness of our scheduling algorithms is

[1] In this study, pdfs refer to the the marginal pdf for each pixel, not the joint pdf for all pixels.

demonstrated using simulations and real images captured with a high speed imaging system [3].

In the following section we provide background on the image sensor pixel model, define sensor SNR and dynamic range, and formulate the multiple capture time scheduling problem. In Section 4.3 we find the optimal time schedules for a uniform pdf. The piece-wise uniform pdf case is discussed in Section 4.4. The approximation of an arbitrary pdf with piece-wise uniform pdfs is discussed in Section 4.5. Finally, simulation and experimental results are presented in Section 4.6.

4.2 Problem Formulation

We assume image sensors operating in direct integration, *e.g.*, CCDs and CMOS PPS, APS, and DPS. Figure 4.1 depicts a simplified pixel model and the output pixel charge $Q(t)$ versus time t for such sensors. During capture, each pixel converts incident light into photocurrent i_{ph}. The photocurrent is integrated onto a capacitor and the charge $Q(T)$ is read out at the end of exposure time T. Dark current i_{dc} and additive noise corrupt the photocharge. The noise is assumed to be the sum of three independent components, (i) shot noise $U(T) \sim \mathcal{N}(0, q(i_{\text{ph}} + i_{\text{dc}})T)$, where q is the electron charge, (ii) readout circuit noise $V(T)$ with zero mean and variance σ_V^2, and (iii) reset noise and FPN C with zero mean and variance σ_C^2. [2] Thus the output charge from a pixel can be expressed as

$$Q(T) = \begin{cases} (i_{\text{ph}} + i_{\text{dc}})T + U(T) + V(T) + C, & \text{for } Q(T) \leq Q_{\text{sat}} \\ Q_{\text{sat}}, & \text{otherwise,} \end{cases}$$

[2] This is the same noise model in Chapter 2 except that read noise is split into readout circuit noise and reset noise, and the reset noise and FPN are lumped into a single term. This formulation distinguishes read noises independent of captures (*i.e.*, reset noise) from read noises dependent on captures (*i.e.*, readout noise) and is commonly used when dealing with multiple capture imaging systems [55].

where Q_{sat} is the saturation charge, also referred to as well capacity. The SNR can be expressed as[3]

$$\text{SNR}(i_{\text{ph}}) = \frac{(i_{\text{ph}}T)^2}{q(i_{\text{ph}} + i_{\text{dc}})T + \sigma_V^2 + \sigma_C^2} \qquad \text{for } i_{\text{ph}} \leq i_{\text{max}}, \tag{4.1}$$

where $i_{\text{max}} \approx Q_{\text{sat}}/T$ refers to the maximum non-saturating photocurrent. Note that SNR increases with i_{ph}, first at 20dB per decade when reset, FPN and readout noise dominate, then at 10dB per decade when shot noise dominates. SNR also increases with T. Thus it is always preferable to have the longest possible exposure time. However, saturation and motion impose practical upper bounds on exposure time.

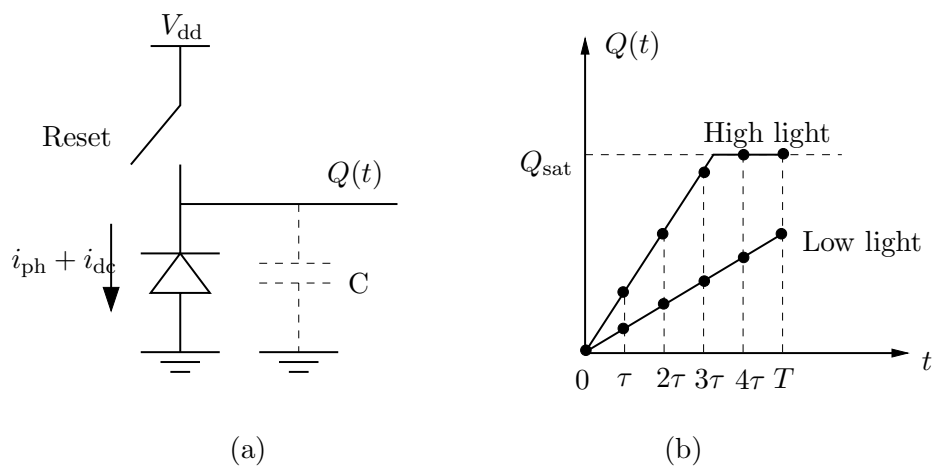


Figure 4.1: (a) Photodiode pixel model, and (b) Photocharge $Q(t)$ vs Time t under two different illuminations. Assuming multiple capture at uniform capture times $\tau, 2\tau, \ldots, T$ and using the LSBS algorithm, the sample at T is used for the low illumination case, while the sample at 3τ is used for the high illumination case.

Sensor dynamic range is defined as the ratio of the maximum non-saturating photocurrent i_{max} to the smallest detectable photocurrent $i_{\text{min}} = \frac{q}{T}\sqrt{\frac{1}{q}i_{\text{dc}}T + \sigma_V^2 + \sigma_C^2}$ [1]. Dynamic range can be extended by capturing several images during exposure time without resetting the photodetector [97, 101]. Using the LSBS algorithm [101]

[3]This is a different version of Equation (3.2), in which σ_r^2 can be regarded as the sum of σ_V^2 and σ_C^2.

dynamic range can be extended at the high illumination end as illustrated in Figure 4.1(b). Liu *et al.* have shown how multiple capture can also be used to extend dynamic range at the low illumination end using weighted averaging. Their method reduces to the LSBS algorithm when only shot noise is present [55].

We assume that scene illumination statistics are given. For a known sensor response, this is equivalent to having complete knowledge of the scene induced photocurrent pdf $f_I(i)$. We seek to find the capture time schedule $\{t_1, t_2, ..., t_N\}$ for N captures that maximizes the average SNR with respect to the given pdf $f_I(i)$ (see Figure 4.2). We assume that the pdf is zero outside a finite length interval $(i_{\min}, i_{\max})$. For simplicity we ignore all noise terms except for shot noise due to photocurrent. Let i_k be the maximum non-saturating photocurrent for capture time t_k, $1 \leq k \leq N$. Thus

$$t_k = \frac{Q_{\text{sat}}}{i_k},$$

and determining capture times $\{t_1, t_2, ..., t_N\}$ is equivalent to determining the set of photocurrents $\{i_1, i_2, ..., i_N\}$. Following its definition in Equation (4.1), the SNR as a function of photocurrent is now given by

$$\text{SNR}(i) = \frac{Q_{\text{sat}} i}{q i_k}$$

for $i_{\text{k+1}} < i \leq i_{\text{k}}$ and $1 \leq k \leq N$. To avoid saturation we assume that $i_1 = i_{\max}$.

The **capture time scheduling problem** is as follows:

Given $f_I(i)$ and N, find $\{i_2, ..., i_N\}$ that maximizes the average SNR

$$E\left(SNR(i_2, ..., i_N)\right) = \frac{Q_{\text{sat}}}{q} \sum_{k=1}^{N} \int_{i_{k+1}}^{i_k} \frac{i}{i_k} f_I(i)\, di, \tag{4.2}$$

subject to: $0 \leq i_{\min} = i_{N+1} < i_N < \ldots < i_k < \ldots < i_2 < i_1 = i_{\max} < \infty$.

Upper bound: Note that since we are using the LSBS algorithm, $\text{SNR}(i) \leq \frac{Q_{\text{sat}}}{q}$

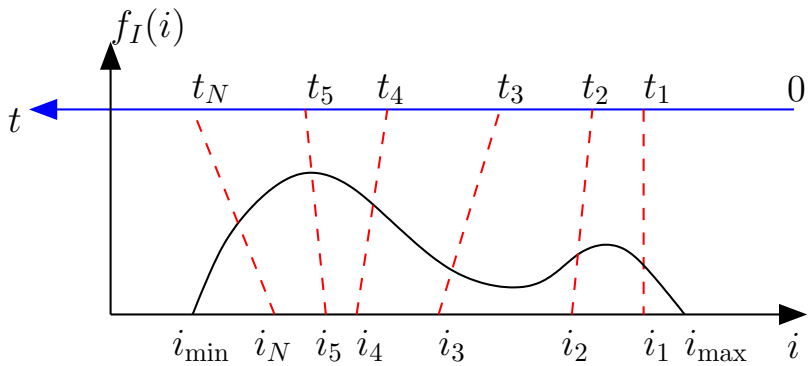


Figure 4.2: Photocurrent pdf showing capture times and corresponding maximum non-saturating photocurrents.

and thus for any N,

$$\max E\left(\mathrm{SNR}(i_1, i_2, ..., i_N)\right) \leq \frac{Q_{\mathrm{sat}}}{q}.$$

This provides a general upper bound on the maximum achievable average SNR using multiple capture. Now, for a single capture with capture time corresponding to $i_{\max}$, the average SNR is given by

$$E\left(\mathrm{SNR}_{\mathrm{SC}}\right) = \frac{Q_{\mathrm{sat}}}{q} \int_{i_{\min}}^{i_{\max}} \frac{i}{i_{\max}} f_I(i)\, di = \frac{Q_{\mathrm{sat}} E(I)}{q i_{\max}},$$

where $E(I)$ is the expectation (or average) of the photocurrent i for given pdf $f_I(i)$. Thus for a given $f_I(i)$, multiple capture can increase average SNR by no more than a factor of $i_{\max}/E(I)$.

4.3 Optimal Scheduling for Uniform PDF

In this section we show how our scheduling problem can be optimally solved when the photocurrent pdf is uniform. For a uniform pdf, the scheduling problem becomes:

Given a uniform photocurrent illumination pdf over the interval $(i_{\min}, i_{\max})$ *and* N,

find $\{i_2, ..., i_N\}$ *that maximizes the average SNR*

$$E\left(SNR(i_2, ..., i_N)\right) = \frac{Q_{\text{sat}}}{2q(i_{\max} - i_{\min})} \sum_{k=1}^{N} (i_k - \frac{i_{k+1}^2}{i_k}), \tag{4.3}$$

subject to: $0 \leq i_{\min} = i_{N+1} < i_N < \ldots < i_k < \ldots < i_2 < i_1 = i_{\max} < \infty$.

Note that for $2 \leq k \leq N$, the function $(i_k - \frac{i_{k+1}^2}{i_k})$ is concave in the two variables i_k and i_{k+1} (which can be readily verified by showing that the Hessian matrix is negative semi-definite). Since the sum of concave functions is concave, the average SNR is a concave function in $\{i_2, ..., i_N\}$. Thus the scheduling problem reduces to a convex optimization problem with linear constraints, which can be optimally solved using well known convex optimization techniques such as gradient/sub-gradient based methods. Table 4.1 provides examples of optimal schedules for up to 10 captures assuming uniform pdf over $(0, 1]$. Note that the optimal capture times are quite different from the commonly assumed uniform or exponentially increasing time schedules. Figure 4.3 compares the optimal average SNR to the average SNR achieved by uniform and exponentially increasing schedules. To make the comparison fair, we assumed the same maximum exposure time for all schedules. Note that using our optimal scheduling algorithm, with only 10 captures, the E(SNR) is within 14% of the upper bound. This performance cannot be achieved with the exponentially increasing schedule and requires over 20 captures to achieve using the uniform schedule.

4.4 Scheduling for Piece-Wise Uniform PDF

In the real world, not too many scenes exhibit uniform illumination statistics. The optimization problem for general pdfs, however, is very complicated and appears intractable. Since any pdf can be approximated by a piece-wise uniform pdf[4], solutions for piece-wise uniform pdfs can provide good approximations to solutions of the general problem. Such approximations are illustrated in Figures 4.4 and 4.5. The

[4]More details on this approximation in the next subsection.

Capture Scheme	Optimal Exposure Times (t_k/t_1)									
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
2 Captures	1	2	–	–	–	–	–	–	–	–
3 Captures	1	1.6	3.2	–	–	–	–	–	–	–
4 Captures	1	1.44	2.3	4.6	–	–	–	–	–	–
5 Captures	1	1.35	1.94	3.1	6.2	–	–	–	–	–
6 Captures	1	1.29	1.74	2.5	4	8	–	–	–	–
7 Captures	1	1.25	1.61	2.17	3.13	5	10	–	–	–
8 Captures	1	1.22	1.52	1.97	2.65	3.81	6.1	12.19	–	–
9 Captures	1	1.20	1.46	1.82	2.35	3.17	4.55	7.29	14.57	–
10 Captures	1	1.18	1.41	1.71	2.14	2.76	3.73	5.36	8.58	17.16

Table 4.1: Optimal capture time schedules for a uniform pdf over interval $(0, 1]$

empirical illumination pdf of the scene in Figure 4.4 has two non-zero regions corresponding to direct illumination and the dark shadow regions, and can be reasonably approximated by a two-segment piece-wise uniform pdf. The empirical pdf of the scene in Figure 4.5, which contains large regions of low illumination, some moderate illumination regions, and small very high illumination regions is approximated by a three-segment piece-wise uniform pdf. Of course better approximations of the empirical pdfs can be obtained using more segments, but as we shall see, solving the scheduling problem becomes more complex as the number of segments increases.

We first consider the scheduling problem for a two-segment piece-wise uniform pdf. We assume that the pdf is uniform over the intervals $(i_{\min}, i_{\max 1})$, and $(i_{\min 1}, i_{\max})$. Clearly, in this case, no capture should be assigned to the interval $(i_{\max 1}, i_{\min 1})$, since one can always do better by moving such a capture to $i_{\max 1}$. Now, assuming that k out of the N captures are assigned to segment $(i_{\min 1}, i_{\max})$, the scheduling problem becomes:

Given a two-segment piece-wise uniform pdf with k captures assigned to interval $(i_{\min 1}, i_{\max})$ and $N-k$ captures to interval $(i_{\min}, i_{\max 1})$, find $\{i_2, ..., i_N\}$ that maximizes

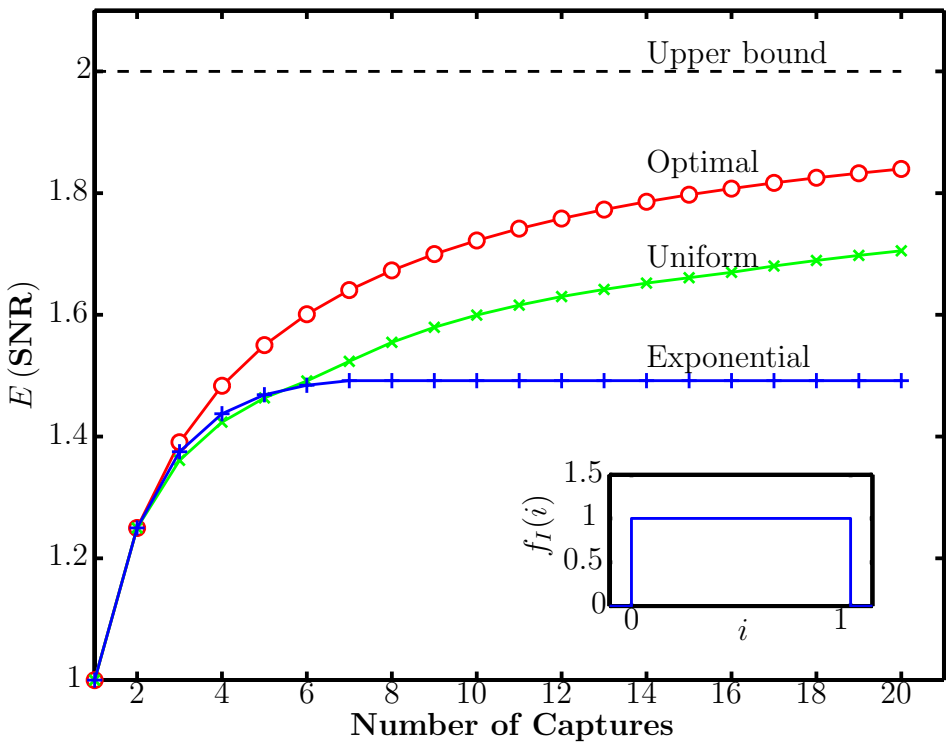


Figure 4.3: Performance comparison of optimal schedule, uniform schedule, and exponential (with exponent = 2) schedule. $E\,(\text{SNR})$ is normalized with respect to the single capture case with $i_1 = i_{\max}$.

the average SNR

$$E(SNR(i_2, ..., i_N)) = \frac{Q_{\text{sat}}}{q}\Bigg(c_1 \sum_{j=1}^{k-1}(i_j - \frac{i_{j+1}^2}{i_j}) + c_1(i_k - \frac{i_{\text{min1}}^2}{i_k}) + c_2\frac{i_{\text{max1}}^2 - i_{k+1}^2}{i_k} + c_2 \sum_{j=k+1}^{N}(i_j - \frac{i_{j+1}^2}{i_j})\Bigg), \tag{4.4}$$

where the constants c_1 and c_2 account for the difference in the pdf values of the two segments,

subject to: $0 \leq i_{\min} = i_{N+1} < i_N < \ldots < i_{k+1} < i_{\text{max1}} \leq i_{\text{min1}} \leq i_k < \ldots < i_2 < i_1 = i_{\max} < \infty$.

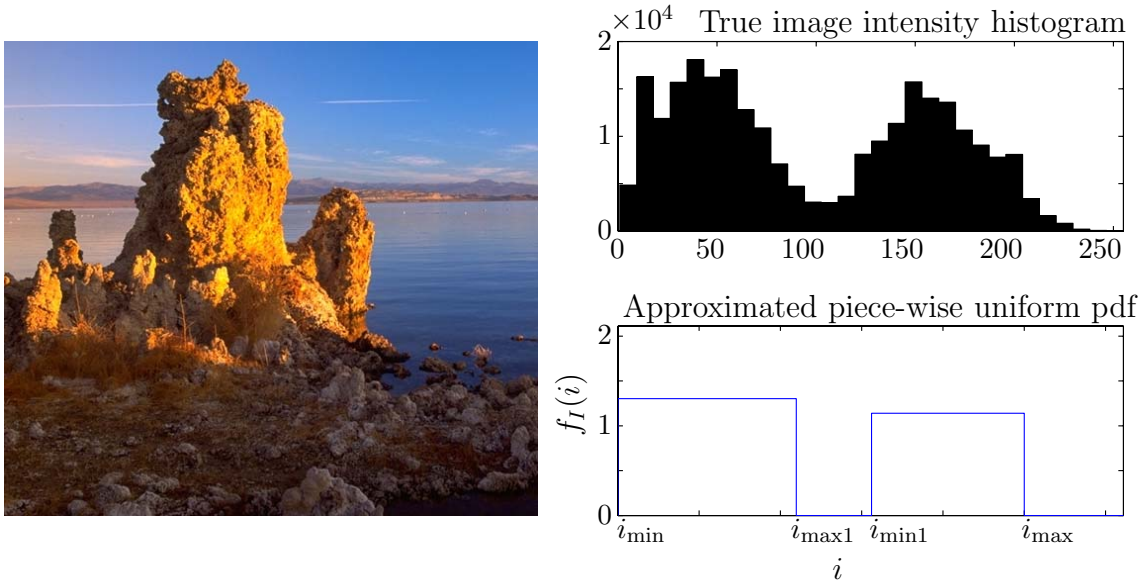


Figure 4.4: An image with approximated two-segment piece-wise uniform pdf

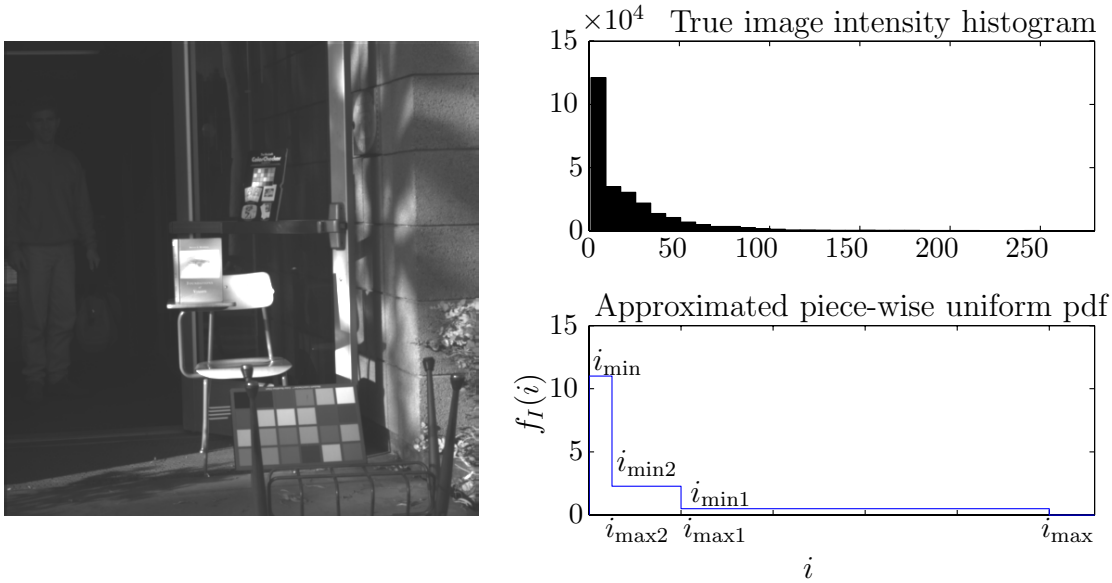


Figure 4.5: An image with approximated three-segment piece-wise uniform pdf

The optimal solution to the general 2-segment piece-wise uniform pdf scheduling problem can thus be found by solving the above problem for each k and selecting the solution that maximizes the average SNR.

Simple investigation of the above equation shows that $E\left(\mathrm{SNR}(i_2, ..., i_N)\right)$ is concave in all the variables except i_k. Certain conditions such as $c_1 i_{\mathrm{min1}}^2 \geq c_2 i_{\mathrm{max1}}^2$ can guarantee concavity in i_k as well, but in general the average SNR is not a concave function. A closer look at equation (4.4), however, reveals that $E\left(\mathrm{SNR}(i_2, ..., i_N)\right)$ is a D.C. function [47, 48], since all terms involving i_k in equation (4.4) are concave functions of i_k except for $c_2 i_{\mathrm{max1}}^2 / i_k$, which is convex. This allows us to apply well-established D.C. optimization techniques (*e.g.*, see [47, 48]). It should be pointed out, however, that these D.C. optimization techniques are not guaranteed to find the globally optimal solution.

In general, it can be shown that average SNR is a D.C. function for any M-segment piece-wise uniform pdf with a prescribed assignment of the number of captures to the M segments. Thus to numerically solve the scheduling problem with M-segment piece-wise uniform pdf, one can solve the problem for each assignment of captures using D.C. optimization, then choose the assignment and corresponding "optimal" schedule that maximizes average SNR.

One particularly simple yet powerful optimization technique that we have experimented with is sequential quadratic programming (SQP) [30, 40] with multiple randomly generated initial conditions. Figures 4.6 and 4.7 compare the solution using SQP with 10 random initial conditions to the uniform schedule and the exponentially increasing schedule for the two piece-wise uniform pdfs of Figures 4.4 and 4.5. Due to the simple nature of our optimization problem, we were able to use brute-force search to find the globally optimal solutions, which turned out to be identical to the solutions using SQP. Note that unlike other examples, in the three-segment example, the exponential schedule outperforms the uniform schedule. The reason is that with few captures, the exponential assigns more captures to the large low and medium illumination regions than the uniform.

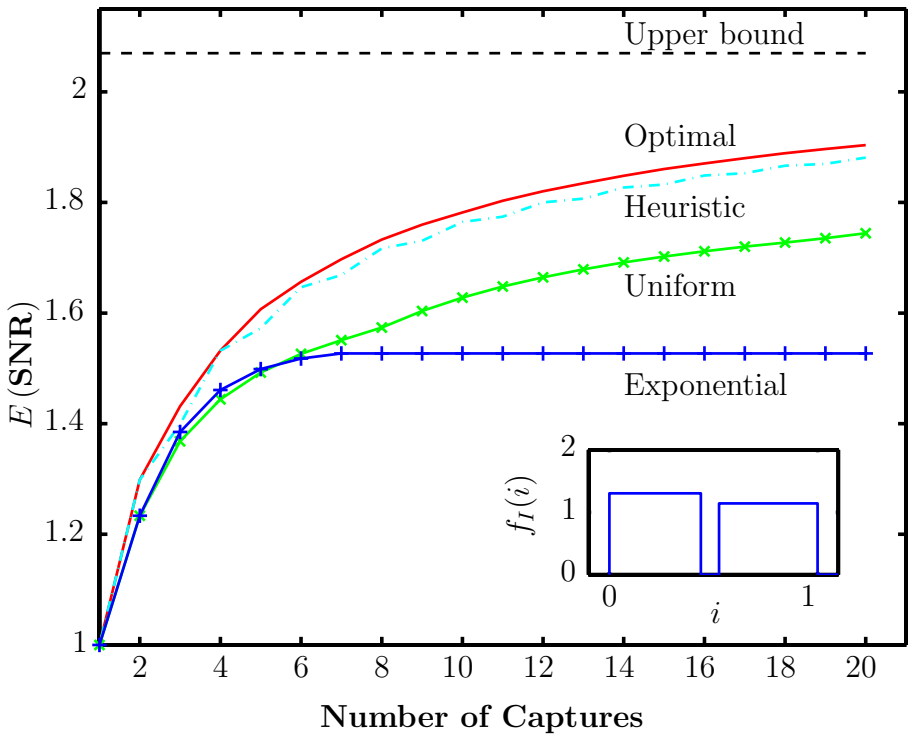


Figure 4.6: Performance comparison of the Optimal, Heuristic, Uniform, and Exponential (with exponent = 2) schedule for the scene in Figures 4.4. E (SNR) is normalized with respect to the single capture case with $i_1 = i_{\max}$.

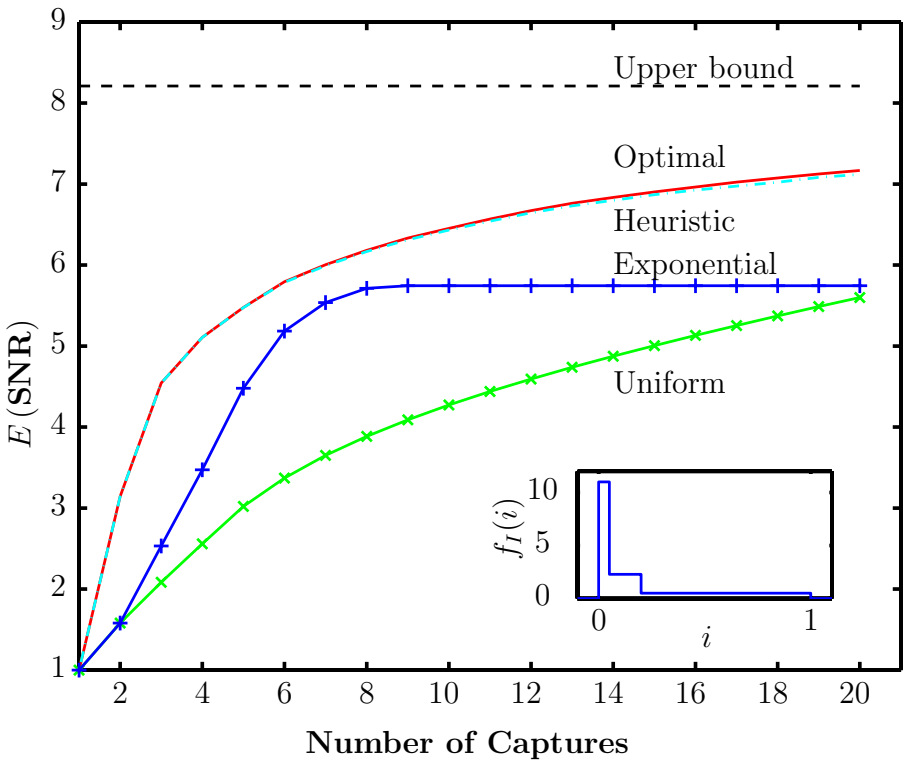


Figure 4.7: Performance comparison of the Optimal, Heuristic, Uniform, and Exponential (with exponent = 2) schedule for the scene in Figures 4.5. E (SNR) is normalized with respect to the single capture case with $i_1 = i_{\max}$.

4.4.1 Heuristic Scheduling Algorithm

As we discussed, finding the optimal capture times for any M-segment piece-wise uniform pdf can be computationally demanding and in fact without exhaustive search, there is no guarantee that we can find the global optimum. As a result, for practical implementations, there is a need for computationally efficient heuristic algorithms. The results from the examples in Figures 4.4 and 4.5 indicate that an optimal schedule assigns captures in proportion to the probability of each segment. Further, within each segment, note that even though the optimal capture times are far from uniformly distributed in time, they are very close to uniformly distributed in photocurrent i. These observations lead to the following simple scheduling heuristic for an M-segment piece-wise uniform pdf with N captures. Let the probability of segment s be $p_s > 0$, $s = 1, 2, \ldots, M$, thus $\sum_{s=1}^{M} p_s = 1$. Denote by $k_s \geq 0$, the number of captures in segment s, thus $\sum_{s=1}^{M} k_s = N$.

1. For segment 1 (the one with the largest photocurrent range), assign $k_1 = \lceil p_1 N \rceil$ captures. Assign the k_1 captures uniformly in i over the segment such that $i_1 = i_{\max}$.

2. For segment s, $s = 2, 3, \ldots, M$, assign $k_s = [(N - \sum_{j=1}^{s-1} k_j)(p_s / \sum_{j=s}^{M} p_j)]$ captures. Assign the k_s captures uniformly in i with the first capture set to the largest i within the segment.

In the first step we used the ceiling function, since to avoid saturation we require that there is at least one capture in segment 1. In the second step $[\cdot]$ refers to rounding. A schedule obtained using this heuristic is given in Figure 4.8 as an example where 6 captures are assigned to 2 segments. Note that the time schedule is far from uniform and is very close to the optimal times obtained by exhaustive search.

In Figure 4.6 we compare the SNR resulting from the schedules obtained using our heuristic algorithm to the optimal, uniform and exponential schedules. Note that the heuristic schedule performs close to optimal for both examples.

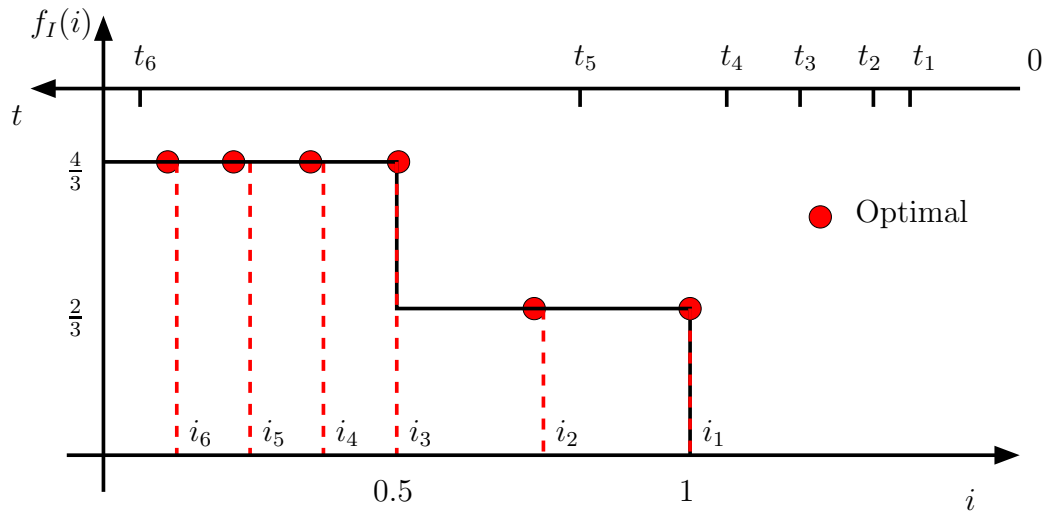


Figure 4.8: An example for illustrating the heuristic capture time scheduling algorithm with $M = 2$ and $N = 6$. $\{t_1, \ldots, t_6\}$ are the capture times corresponding to $\{i_1, \ldots, i_6\}$ as determined by the heuristic scheduling algorithm. For comparison, optimal $\{i_1, \ldots, i_6\}$ are indicated with circles.

4.5 Piece-wise Uniform PDF Approximations

Up to now we have described how the capture time scheduling problem can be obtained for any piece-wise uniform distributions. In general while it is quite clear that any distribution can be approximated by a piece-wise uniform pdf with finite number of segments, issues such that how such approximations should be made and how many segments need to be included in the approximation remain to be answered. Such problems have been widely studied in density estimation, which refers to the construction of an estimate of the probability density function from observed data. Many books [74, 68] offer a comprehensive description of this topic. There exist many different methods for density estimation. Examples are histograms, the kernel estimator [71], the nearest neighbor estimator [57], the maximum penalized likelihood method [41] and many other approaches. Among all these different approaches, the histogram method is of particular interest to us since image histograms are often generated for adjusting camera control parameters in a digital camera, therefore using the histogram method does not introduce any additional requirements on camera

hardwares and softwares. So in this section we first describe an *Iterative Histogram Binning Algorithm* that can approximate any pdf to a piece-wise uniform pdf with prescribed number of segments, we then discuss the choice for the number of segments used in the approximation. It should be stressed that there are many different approaches to solve our problem. For example, our problem can be viewed as the quantization of the pdf, therefore quantization techniques can be used to "optimize" the choice of the segments and their values. What we present in this section is one simple approach that solves our problem and can be easily implemented in practice.

4.5.1 Iterative Histogram Binning Algorithm

The *Iterative Histogram Binning Algorithm* can be summarized into the following steps :

1. Get the initial histogram of the image and start with a large number of bins (or segments);
2. Merge two adjacent bins and calculate the Sum of Absolute Difference (SAD) from the original histogram. Repeat for all pairs of adjacent bins;
3. Merge the two bins that give the minimum SAD (*i.e.*, we have reduced the number of bins, or segments, by one)
4. Repeat 2 and 3 on the updated histogram until the number of desired bins or segments is reached

Figure 4.9 shows an example of how the algorithm works. We start with a seven-segment histogram and want to approximate it with a three-segment histogram. Since at each iteration, the number of segments is reduced by one by binning two adjacent segments, the entire binning process takes four steps.

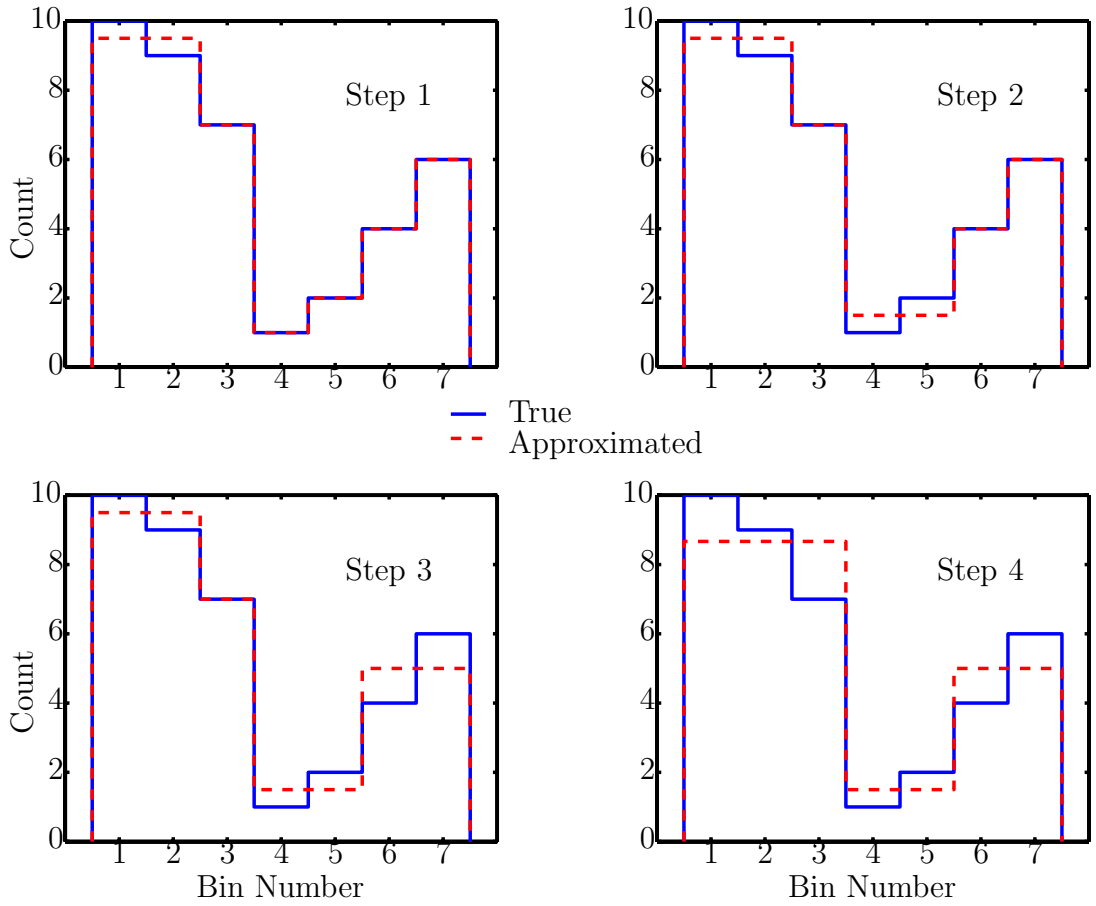


Figure 4.9: An example that shows how the Iterative Histogram Binning Algorithm works. A histogram of 7 segments is approximated to 3 segments with 4 iterations. Each iteration merges two adjacent bins and therefore reduces the number of segments by one.

4.5.2 Choosing Number of Segments in the Approximation

Selecting the number of segments used in the pdf approximation is also a much studied problem. For instance, when the pdf approximation is treated as the quantization of the pdf, selecting the number of segments is equivalent to choosing the number of quantization levels and therefore can be solved as part of the optimization of the quantization levels. While such a treatment is rigorous, in practice it is always desirable to have a simple approach that can be easily implemented. Since using more segments results in a better approximation at the expense of complicating the capture time scheduling process, ideally we would want to work with a small number of segments in the approximation. It is useful to understand how the number of segments in the pdf approximation affects the final performance of the multiple capture scheme. Such an effect can be seen in Figure 4.10 for the image in Figure 4.5, where the $E[\text{SNR}]$ is plotted as a function of the number of segments used in the pdf approximation for a 20-capture scheme. In other words, we first approximate the original pdf to a piecewise uniform pdf, we then use our optimal capture time scheduling algorithm to select the 20 capture times. Finally we apply the 20 captures on the original pdf and calculate the performance improvement in terms of $E[\text{SNR}]$. The above procedures are repeated for each number of segments. From Figure 4.10 it can be seen that a three-segment pdf is a good approximation for this specific image. In general, the number of desired segments depends on the original pdf. If the original pdf exhibits roughly a Gaussian distribution or a mixture of a small number of Gaussian distributions, using a very small number of segments may well be sufficient. Our experience with real images suggests that we rarely need more than five segments, and two or three segments actually work quite well for a large set of images.

4.6 Simulation and Experimental Results

Our capture time scheduling algorithms are demonstrated on real images using vCam and an experimental high speed imaging system [3]. For vCam simulation, we used a 12-bit high dynamic range scene shown in Figure 4.5 as an input to the simulator. We

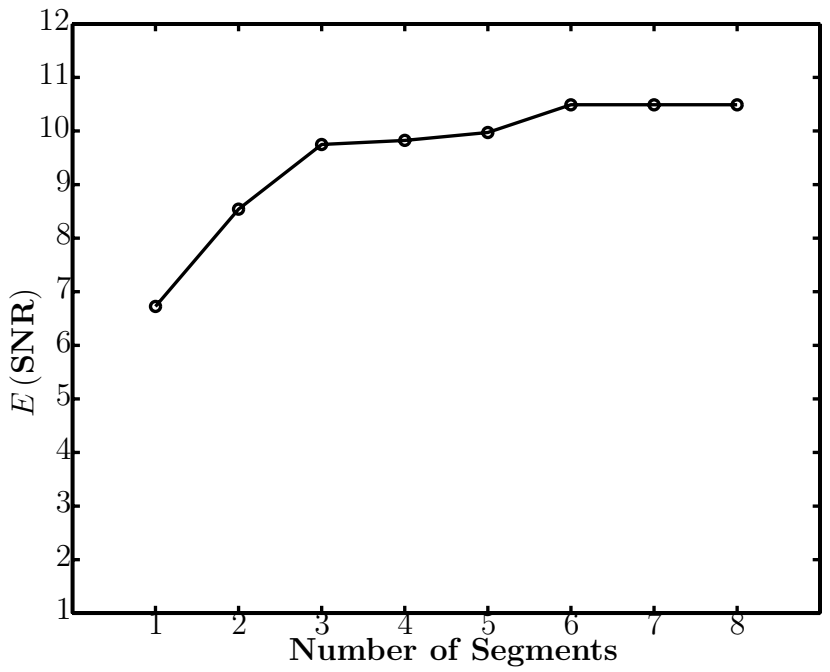


Figure 4.10: E[SNR] versus the number of segments used in the pdf approximation for a 20-capture scheme on the image shown in Figure 4.5. E[SNR] is normalized to the single capture case.

assumed a 256×256 pixel array with only dark current and signal shot noise included. We obtained the simulated camera output for 8 captures scheduled (i) uniformly, (ii) optimally, and (iii) using the heuristic algorithm described in the previous section. In all cases we used the LSBS algorithm to reconstruct the high dynamic range image. For fair comparison, we used the same maximum exposure time for all three cases. The simulation results are illustrated in Figure 4.11. To see the SNR improvement, we zoomed in on a small part of the MacBeth chart [58] in the image. Since the MacBeth chart consists of uniform patches, noise can be more easily discerned. In particular for the two patches on the right, the output of both Optimal and Heuristic are less noisy than Uniform. Figure 4.12 depicts the noise images obtained by subtracting the noiseless output image obtained by setting shot noise to zero from the three output images, together with their histograms. Notice that even though the histograms look similar in shape, the histogram for the uniform case contains more regions with large errors. Finally, in terms of average SNR, Uniform is 1.3dB lower than both Heuristic and Optimal.

We are also able to demonstrate the benefit of optimal scheduling of multiple captures experimentally using an experimental high speed imaging system [3]. Our scene setup comprises an eye chart under a point light source inside a dark room. We took an initial capture with 5ms integration time. The relatively short integration time ensures a non-saturated image and we estimated the signal pdf based on the histogram of the image. The estimated pdf was then approximated with a three-segment piece-wise uniform pdf and optimal capture times were selected for a 4-capture case with initial capture time set to 5ms. We also took 4 uniformly spaced captures with the same maximum exposure time. Figure 4.13 compares the results after LSBS was used. We can see that Optimal outperforms Uniform. This is visible especially in areas near the “F”.

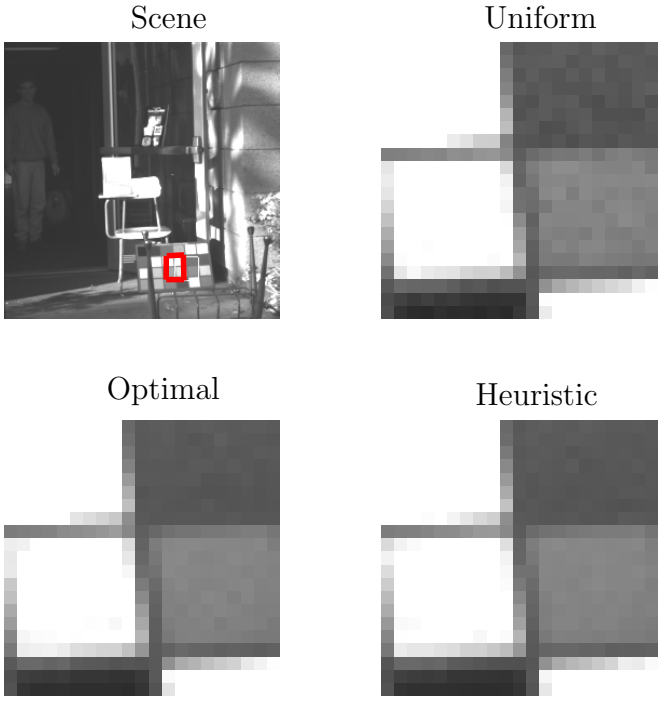


Figure 4.11: Simulation result on a real image from vCam. A small region, as indicated by the square in the original scene, is zoomed in for better visual effects

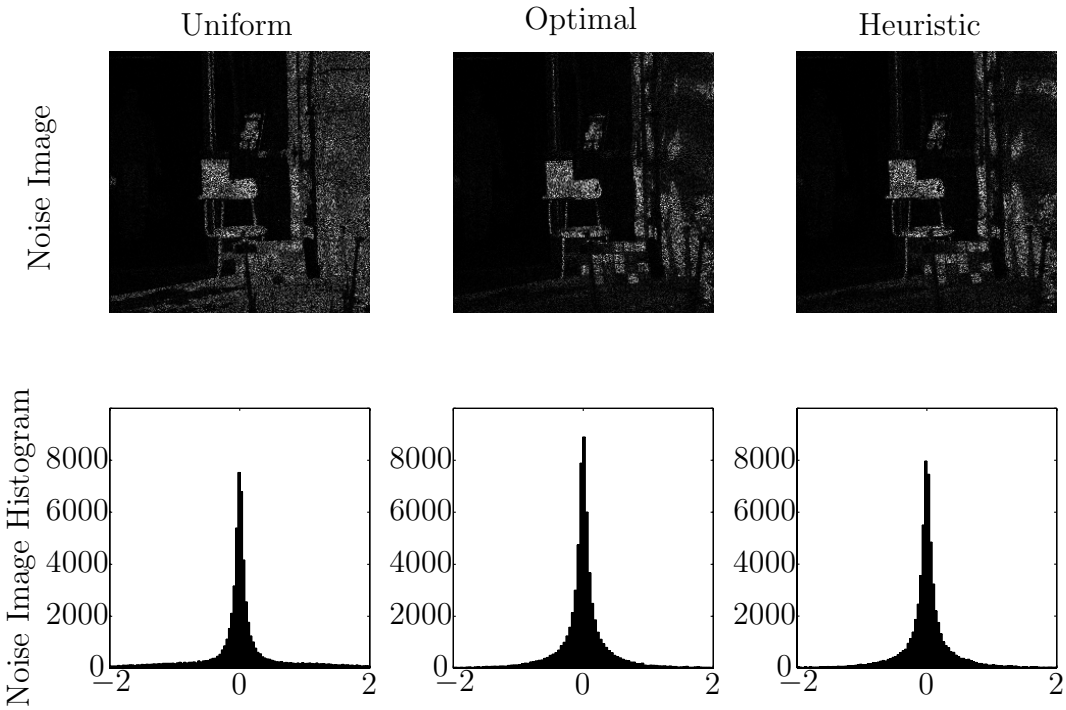


Figure 4.12: Noise images and their histograms for the three capture schemes

4.7 Conclusion

This chapter presented the first systematic study of optimal selection of capture times in a multiple capture imaging system. Previous studies on multiple capture have assumed uniform or exponentially increasing capture time schedules justified by certain practical implementation considerations. It is advantageous in terms of system computational power, memory, power consumption, and noise to employ the least number of captures required to achieve a desired dynamic range and SNR. To do so, one must carefully select the capture time schedule to optimally capture the scene illumination information. In practice, sufficient scene illumination information may not be available before capture, and therefore, a practical scheduling algorithm may need to operate "online", *i.e.*, determine the time of the next capture based on updated scene illumination information gathered from previous captures. To develop understanding of the scheduling problem, we started by formulating the "offline" scheduling problem, *i.e.*, assuming complete prior knowledge of scene illumination pdf, as an optimization

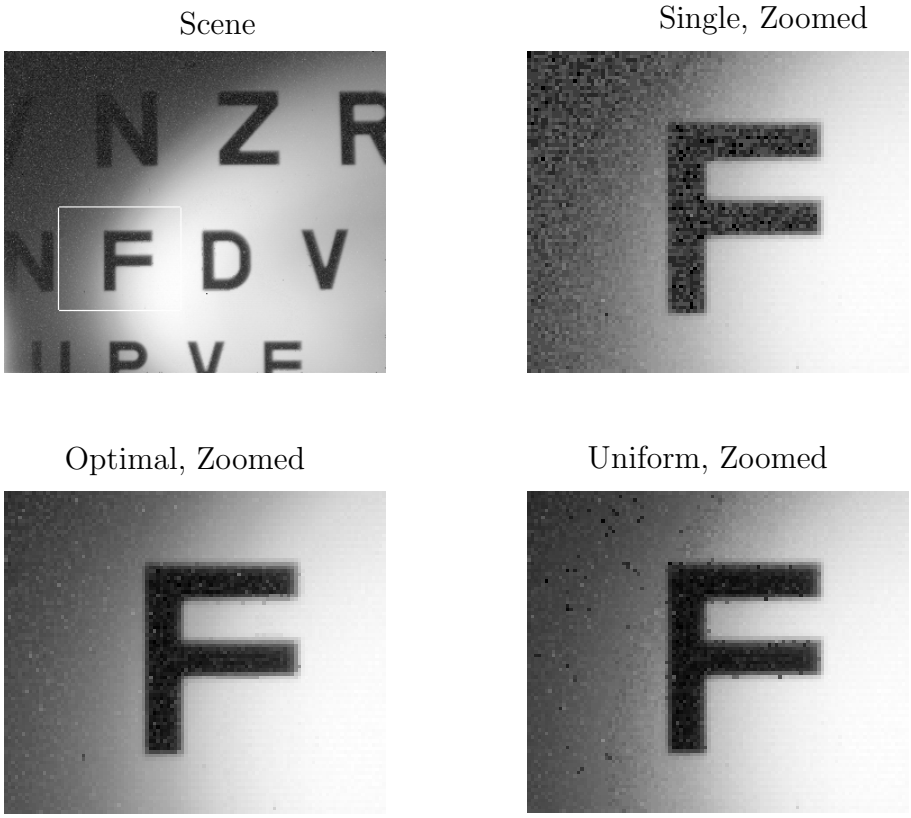


Figure 4.13: Experimental results. The top-left image is the scene to be captured. The white rectangle indicates the zoomed area shown in the other three images. The top-right image is from a single capture at 5ms. The bottom-left image is reconstructed using LSBS algorithm from optimal captures taken at $5, 15, 30$ and 200ms. The bottom-right image is reconstructed using LSBS algorithm from uniform captures taken at $5, 67, 133$ and 200ms. Due to the large constrast in the scene, all images are displayed in $\log 10$ scale.

problem where average SNR is maximized for a given number of captures. Ignoring read noise and FPN and using the LSBS algorithm, our formulation leads to a general upper bound on the average SNR for any illumination pdf. For a uniform illumination pdf, we showed that the average SNR is a concave function in capture times and therefore the global optimum can be found using well-known convex optimization techniques. For a general piece-wise uniform illumination pdf, the average SNR is not necessarily concave. Average SNR is, however, a D.C. function and can be solved using well-established D.C. or global optimization techniques. We then introduced a very simple but highly competitive heuristic scheduling algorithm which can be easily implemented in practice. To complete the scheduling algorithm, we also discussed the issue on how to approximate any pdf with a piece-wise uniform pdf. Finally application of our scheduling algorithms to simulated and real images confirmed the benefits of adopting an optimized schedule based on illumination statistics over uniform and exponential schedules.

The "offline" scheduling algorithms we discussed can be directly applied in situations where enough information about scene illumination is known in advance. It is not unusual to assume the availability of such prior information. For example all auto-exposure algorithms used in practice, assume the availability of certain scene illumination statistics [38, 85]. When the scene information is not known, one simple solution may be that we can take one extra capture initially and derive the necessary information about the scene statistics. How to proceed after that will be exactly the same as described in this paper. The problem is, however, that in reality taking a single capture does not necessarily give us a good complete picture about the scene. If the capture is taken too slowly, we may have missed information about the bright regions due to saturation. On the other hand, if the capture is taken too quickly, we may not get enough SNR on those dark regions so that we don't have an accurate estimate on the signal pdf. Therefore a more general "online" approach that iteratively determines the next capture time based on the updated photocurrent pdf that are derived from all the previous captures appears to be a better candidate for solving the scheduling problem. We have implemented such procedures in vCam and our observations from simulation results suggest that in practice "online" scheduling

can be switched to "offline" scheduling after just a few iterations with negligible loss in performance. So in summary, our approach as discussed in this chapter is mostly sufficient for dealing with practical problems.

Chapter 5

Conclusion

5.1 Summary

We have introduced a digital camera simulator - vCam - that enables digital camera designers to explore different system designs. We have described the modeling of the scene, the imaging optics, and the image sensor. The implementation of vCam as a Matlab toolbox has also been discussed. Finally we have presented the validation results on vCam using real test structures. vCam has found both research and commercial values as it has been licensed to numerous academic institutions as well as commercial companies.

One application that uses vCam to select optimal pixel size as part of the image sensor design is then presented. Without a simulator, such a study can be extremely difficult to be analyzed. In this research we have demonstrated the tradeoff between sensor dynamic range and spatial resolution as a function of pixel size. We have developed a methodology using vCam, synthetic contrast sensitivity function scenes, and the image quality metric S-CIELAB for determining the optimal pixel size. The methodology is demonstrated for active pixel sensors implemented in CMOS processes down to 0.18um technology.

We have described a second application of vCam by demonstrating algorithms for scheduling multiple captures in a high dynamic range imaging system. This is the first investigation of optimizing capture times in multiple capture systems. In particular, capture time scheduling is formulated as an optimization problem where average SNR is maximized for a given scene pdf. For a uniform scene pdf, the average SNR is a concave function in capture times and thus the global optimum can be found using well-known convex optimization techniques. For a general piece-wise uniform pdf, the average SNR is not necessarily concave, but rather a D.C. function and can be solved using D.C. optimization techniques. A very simple heuristic algorithm is described and shown to produce results that are very close to optimal. These theoretical results are finally demonstrated on real images using vCam and an experimental high speed imaging system.

5.2 Future Work and Future Directions

vCam has proven a useful research tool in helping us study different camera system tradeoffs and explore new processing algorithms. As we make continuous improvements to the simulator, more and more studies on the camera system design can be carried out with high confidence. It is our hope that vCam's popularity will help to facilitate the process of making it more sophisticated and closer to reality. We think future work may well follow such a thread and we will group such work into two categories: vCam improvements and vCam applications.

vCam can be improved in many different ways. We only make a few suggestions that we think will significantly improve vCam. First of all, the front end of the digital camera system, including the scene and optics, needs to be extended. Currently vCam assumes that we are only interested in capturing the wavelength part of the scene. While this is sufficient for our own research purposes, real scenes contain not simply photons at different wavelength, but also a significant amount of geometric information. This type of research has been studied extensively in fields such as computer graphics. Borrowing their research results and incorporating them into

vCam seems very logical. Second, in order to have a large set of calibrated scenes to work with, building a database of scenes of different variety (*e.g.*, low light, high light, high dynamic range and so on) will make vCam not only more useful, but also help to build more accurate scene models. Third, more sophisticated optics model will help greatly. Besides the image sensor, the imaging lens is one of the most crucial components in a digital camera system. Currently vCam uses a diffraction-limited lens without any consideration of aberration. In reality aberration always exists and often causes major image degradation. Having an accurate lens model that can account for such an effect is highly desirable.

The applications of vCam in exploring digital camera system designs can be very broad. Here we only mention a few in which we have particular interest. First, to follow the pixel size study, we would like to see how our methodology can be extended to color. Second, to complete the multiple capture time selection problem, it will be interesting to look at how the online scheduling algorithm performs in comparison to the offline scheduling algorithm. Since our scheduling algorithm is based on the assumption that the sensor is operating in a shot noise dominated regime, a more challenging problem is to look at the case when read noise can not be ignored. In that case, we believe linear estimation techniques [55] need to be combined with the optimal selection of capture times to fully take advantage of the capability of a multiple capture imaging system. Another interesting area to investigate is the different CFA patterns versus more recent technologies such as Foveon's X3 technology [35]. It is our belief that vCam allows camera designers to optimize many system components and control parameters. Such an optimization will enable digital cameras to produce images with higher and higher quality. Good days are still ahead!

Bibliography

[1] A. El Gamal, "EE392b Classnotes: Introduction to Image Sensors and Digital Cameras," http://www.stanford.edu/class/ee392b, Stanford University, 2001.

[2] A. El Gamal, B. Fowler and D. Yang. "Pixel Level Processing – Why, What and How?". *Proceedings of SPIE*, Vol. 3649, 1999.

[3] A. Ercan, F. Xiao, S.H. Lim, X. Liu, and A. El Gamal, "Experimental High Speed CMOS Image Sensor System and Applications," *Proceedings of IEEE Sensors 2002*, pp. 15-20, Orlando, FL, June 2002.

[4] http://www.avanticorp.com

[5] Bryce E. Bayer "Color imaging array," U.S. Patent 3,971,065

[6] N.F. Borrelli "Microoptics Technology: Fabrication and Applications of Lens Arrays and Devices," *Optical Engineering*, Vol. 63, 1999

[7] R.W. Boyd, "Radiometry and the Detection of Optical Radiation," Wiley, New York, 1983.

[8] http://color.psych.ucsb.edu/hyperspectral

[9] P. Longere and D.H. Brainard, "Simulation of digital camera images from hyperspectral input," http://color.psych.upenn.edu/simchapter/simchapter.ps

[10] P. Vora, J.E. Farrell, J.D. Tietz and D.H. Brainard, "Image capture: modelling and calibration of sensor responses and their synthesis from multispectral images," *Hewlett-Packard Laboratories Technical Report HPL-98-187*, 1998 http://www.hpl.hp.com/techreports/98/HPL-98-187.html

[11] G. Buchsbaum, "A Spatial Processor Model for Object Colour Perception," *Journal of the Franklin Institute*, Vol. 310, pp. 1-26, 1980

[12] F. Campbell and J. Robson, "Application of Fourier analysis to the visibility of gratings," *Journal of Physiology* Vol. 197, pp. 551-566, 1968.

[13] P. B. Catrysse, B. A. Wandell, and A. El Gamal, "Comparative analysis of color architectures for image sensors," *Proceedings of SPIE*, Vol. 3650, pp. 26-35, San Jose, CA, 1999.

[14] P. B. Catrysse, X. Liu, and A. El Gamal, "QE reduction due to pixel vignetting in CMOS image sensors," *Proceedings of SPIE*, Vol. 3965, pp. 420-430, San Jose, CA, 2000.

[15] T. Chen, P. Catrysse, B. Wandell, and A. El Gamal, "vCam – A Digital Camera Simulator," in preparation, 2003

[16] T. Chen, P. Catrysse, B. Wandell and A. El Gamal, "How small should pixel size be?," *Proceedings of SPIE*, Vol. 3965, pp. 451-459, San Jose, CA, 2000.

[17] Kwang-Bo Cho, *et al.* "A 1.2V Micropower CMOS Active Pixel Image Sensor for Portable Applications," *ISSCC2000 Technical Digest*, Vol. 43. pp. 114-115, 2000

[18] C.I.E., "Recommendations on uniform color spaces,color difference equations, psychometric color terms," *Supplement No.2 to CIE publication No.15(E.-1.3.1) 1971/(TC-1.3)*, 1978.

[19] B.M Coaker, N.S. Xu, R.V. Latham and F.J. Jones, "High-speed imaging of the pulsed-field flashover of an alumina ceramic in vacuum," *IEEE Transactions on Dielectrics and Electrical Insulation*, Vol. 2, No. 2, pp. 210-217, 1995.

[20] *CODE V.40*, Optical Research Associates, Pasadena, California, 1999.

[21] D. R. Cok, "Single-chip electronic color camera with color-dependent birefringent optical spatial frequency filter and red and blue signal interpolating circuit," U.S. Patent 4,605,956, 1986

[22] S.J. Decker, R.D. McGrath, K. Brehmer, and C.G. Sodini, "A 256x256 CMOS Imaging Array with Wide Dynamic Range Pixels and Column-Parallel Digital Output," *IEEE Journal of Solid-State Circuits*, Vol. 33, No. 12, pp. 2081-2091, December 1998.

[23] P.B. Denyer et al. "Intelligent CMOS imaging," *Charge-Coupled Devices and Solid State Optical Sensors IV –Proceedings of the SPIE*, Vol. 2415, pp. 285-91, 1995.

[24] P.B. Denyer et al. "CMOS image sensors for multimedia applications," *Proceedings of IEEE Custom Integrated Circuits Conference*, Vol. 2415, pp. 11.15.1-11.15.4, 1993.

[25] P.Denyer, D. Renshaw, G. Wang, M. Lu, and S. Anderson. "On-Chip CMOS Sensors for VLSI Imaging Systems," *VLSI-91*, 1991.

[26] P.Denyer, D. Renshaw, G. Wang, and M. Lu. "A Single-Chip Video Camera with On-Chip Automatic Exposure Control," *ISIC-91*, 1991.

[27] A. Dickinson, S. Mendis, D. Inglis, K. Azadet, and E. Fossum. "CMOS Digital Camera With Parallel Analog-to-Digital Conversion Architecture," *1995 IEEE Workshop on Charge Coupled Devices and Advanced Image Sensors*, April 1995.

[28] A. Dickinson, B. Ackland, E.S. Eid, D. Inglis, and E. Fossum. "A 256x256 CMOS active pixel image sensor with motion detection," *ISSCC1995 Technical Digests*, February 1995.

[29] B. Dierickx. "Random addressable active pixel image sensors," *Advanced Focal Plane Arrays and Electronic Cameras – Proceedings of the SPIE*, Vol. 2950, pp. 2-7, 1996.

[30] R. Fletcher "Practical Methods of Optimization," *Vol. 1, Unconstrained Optimization, and Vol. 2, Constrained Optimization*, John Wiley and Sons, 1980.

[31] P. Foote "Bulletin of Bureau of Standards, 12," Scientific paper 583, 1915

[32] E.R. Fossum. "CMOS image sensors: electronic camera on a chip," *Proceedings of International Electron Devices Meeting*, pp. 17-25, 1995.

[33] E.R. Fossum. "Ultra low power imaging systems using CMOS image sensor technology," *Advanced Microdevices and Space Science Sensors – Proceedings of the SPIE*, Vol. 2267, pp. 107-111, 1994.

[34] E.R. Fossum. "Active Pixel Sensors: are CCD's dinosaurs," *Proceeding of SPIE*, Vol. 1900, pp. 2-14, 1993.

[35] http://www.foveon.com

[36] B. Fowler, A. El Gamal and D. Yang. "Techniques for Pixel Level Analog to Digital Conversion," *Proceedings of SPIE*, Vol.3360, pp. 2-12, 1998.

[37] B. Fowler, A. El Gamal, and D. Yang. "A CMOS Area Image Sensor with Pixel-Level A/D Conversion," *ISSCC Digest of Technical Papers*, 1994.

[38] Fujii *et al.* , "Automatic exposure controlling device for a camera," U.S. Patent 5452047, 1995.

[39] Lliana Fujimori, et al. "A 256x256 CMOS Differential Passive Pixel Imager with FPN Reduction Techniques," *ISSCC2000 Technical Digest*, Vol. 43. pp. 106-107, 2000

[40] P.E. Gill, W. Murray and M.H.Wright "Practical Optimization," Academic Press, London, 1981

[41] I.J. Good and R.a. Gaskins, "Nonparametric Roughness Penalties for Probability Density," *Biometrika*, Vol. 58, pp. 255-277, 1971

[42] M. Gottardi, A. Sartori, and A. Simoni. "POLIFEMO: An Addressable CMOS 128×128 - Pixel Image Sensor with Digital Interface," Technical report, Istituto Per La Ricerca Scientifica e Tecnologica, 1993.

[43] D. Handoko, S. Kawahito, Y. Todokoro, and A. Matsuzawa, "A CMOS Image Sensor with Non-Destructive Intermediate Readout Mode for Adaptive Iterative Search Motion Vector Estimation," *2001 IEEE Workshop on CCD and Advanced Image Sensors*, pp. 52-55, Lake Tahoe, CA, June 2001.

[44] D. Handoko, S. Kawahito, Y. Takokoro, M. Kumahara, and A. Matsuzawa", "A CMOS Image Sensor for Focal-plane Low-power Motion Vector Estimation," *Symposium of VLSI Circuits*, pp. 28-29, June 2000.

[45] W. Hoekstra et al. "A memory read–out approach for 0.5μm CMOS image sensor," *Proceedings of the SPIE*, Vol. 3301, 1998.

[46] G. C. Holst, "CCD Arrays, Cameras and Displays," JCD Publishing and SPIE, Winter Park, Florida, 1998.

[47] R. Horst, P.Pardalos, and N.V. Thoai, "Introduction to global optimization," Kluwer Academic, Boston, Massachusetts, 2000.

[48] R. Horst and H. Tuy, "Global optimization: deterministic approaches," Springer, New York, 1996.

[49] J.E.D Hurwitz *et al.* "800–thousand–pixel color CMOS sensor for consumer still cameras," *Proceedings of the SPIE*, Vol. 3019, pp. 115-124, 1997.

[50] http://public.itrs.net

[51] S. Kavadias, B. Dierickx, D. Scheffer, A. Alaerts, D. Uwaerts, and J. Bogaerts, "A Logarithmic Response CMOS Image Sensor with On-Chip Calibration," *IEEE Journal of Solid-State Circuits*, Vol. 35, No. 8, pp. 1146-1152, August 2000.

[52] M.V. Klein and T.E. Furtak, "Optics," 2nd edition, Wiley, New York, 1986.

[53] S. Kleinfelder, S.H. Lim, X.Q. Liu, and A. El Gamal, "A 10,000 Frame/s 0.18um CMOS Digital Pixel Sensor with Pixel-Level Memory," *IEEE Journal of Solid State Circuits*, Vol. 36, No. 12, pp. 2049-2059, December 2001.

[54] S.H. Lim and A. El Gamal, "Integrating Image Capture and Processing – Beyond Single Chip Digital Camera", *Proceedings of SPIE*, Vol. 4306, 2001.

[55] X. Liu and A. El Gamal, "Photocurrent Estimation from Multiple Non-destructive Samples in a CMOS Image Sensor," *Proceedings of SPIE*, Vol. 4306, pp. 450-458, San Jose, CA, 2001.

[56] X.Q. Liu and A. El Gamal, "Simultaneous Image Formation and Motion Blur Restoration via Multiple Capture,"] *ICASSP'2001 conference*, May 2001.

[57] D.O. Loftsgaarden and C.P. Quesenberry, "A Nonparametric Estimate of a Multivariate Density Functioin," *Ann. Math. Statist.* Vol. 36, pp. 1049-1051, 1965

[58] C.S. McCamy, H. Marcus and J.G. Davidson, "A Colour-Rendition Chart," *Journal of Applied Photographic Engineering*, Vol. 2, No. 3, pp. 95-99, 1976

[59] C. Mead, "A Sensitive Electronic Photoreceptor". *1985 Chapel Hill Conference on VLSI*, Chapel Hill, NC, 1985.

[60] S. K. Mendis, S. E. Kemeny, R. C. Gee, B. Pain, C. O. Staller, Q. Kim, and E. R. Fossum, "CMOS Active Pixel Image Sensors for Highly Integrated Imaging Systems," *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 2, pp. 187-197, 1997.

[61] S.K Mendis *et al.* . "Progress in CMOS active pixel image sensors," *Charge-Coupled Devices and Solid State Optical Sensors IV –Proceedings of the SPIE*, volume 2172, pages 19–29, 1994.

[62] M.E. Nadal and E.A. Thompson "NIST Reference Goniophotometer for Specular Gloss Measurements," *Journal of Coatings Technology*, Vol. 73, No. 917, pp. 73-80, June 2001

[63] F.E. Nicodemus, J.C. Richmond, J.J. Hsia, I.W. Ginsberg, and T. Limperis, "Geometric Considerations and Nomenclature for Reflectance," *Natl. Bur. Stand. (U.S.) Monogr.* ***160***, U.S. Department of Commerce, Washington, D.C., 1977

[64] R.H. Nixon et al. "256×256 CMOS active pixel sensor camera-on-a-chip," *ISSCC96 Technical Digest*, pp. 100-101, 1996.

[65] "Technology Roadmap for Image Sensors," OIDA Publications, 1998

[66] R.A. Panicacci et al. "128 Mb/s multiport CMOS binary active-pixel image sensor," *ISSCC96 Technical Digest*, pp. 100-101, 1996.

[67] F. Pardo et al. "Response properties of a foveated space-variant CMOS image sensor," *IEEE International Symposium on Circuits and Systems Circuits and Systems Connecting the World – ISCAS 96*, 1996.

[68] P. Rao "Nonparametric Functional Estimation," Academic Press, Orlando, 1983

[69] http://radsite.lbl.gov/radiance/HOME.html

[70] http://www.cis.rit.edu/mcsl/online/lippmann2000.shtml

[71] M. Rosenblatt "Remarks on some nonparametric estimates of a density function," *Ann. Math. Statist.* Vol. 27, pp. 832-837, 1956

[72] A. Sartori. "The MInOSS Project," *Advanced Focal Plane Arrays and Electronic Cameras – Proceedings of the SPIE*, volume 2950, pp. 25-35, 1996.

[73] D. Seib, "Carrier Diffusion Degradation of Modulation Transfer Function in Charge Coupled Imagers," *IEEE Transactions on Electron Devices*, Vol. 21, No. 3, 1974

[74] B.W. Silverman "Density Estimation for Statistics and Data Analysis," Chapman and Hall, London, 1986

[75] S. Smith, *et al.* "A single-chip 306x244-pixel CMOS NTSC video camera". *ISSCC1998 Technical Digest*, Vol. 41, pp. 170-171, 1998

[76] W.J. Smith, "Modern Optical Engineering," McGraw-Hill Professional, 2000.

[77] V. Smith and J. Pokorny, "Spectral sensitivity of color-blind observers and the cone photopigments," *Vision Res.* Vol. 12, pp. 2059-2071, 1972.

[78] J. Solhusvik. "Recent experimental results from a CMOS active pixel image sensor with photodiode and photogate pixels," *Advanced Focal Plane Arrays and Electronic Cameras – Proceedings of the SPIE*, Vol. 2950, pp. 18-24, 1996.

[79] Nenad Stevanovic, *et al.* "A CMOS Image Sensor for High-Speed Imaging". *ISSCC2000 Technical Digest*, Vol. 43, pp. 104-105, 2000

[80] V. Steinhaus, "Mathematical Snapshots," 3rd edition, Dover, New York, 1999.

[81] E. Stevens, "An Analytical, Aperture, and Two-Layer Carrier Diffusion MTF and Quantum Efficiency Model for Solid-State Image Sensors," *IEEE Transactions on Electron Devices*, Vol. 41, No. 10, 1994

[82] E. Stevens, "A Unified Model of Carrier Diffusion and Sampling Aperture Effects on MTF in Solid-State Image Sensors," *IEEE Transactions on Electron Devices*, Vol. 39, No. 11, 1992

[83] Tadashi Sugiki, *et al.* "A 60mW 10b CMOS Image Sensor with Column-to-Column FPN Reduction," *ISSCC2000 Technical Digest*, Vol. 43. pp. 108-109, 2000

[84] S. M. Sze, *Semiconductor Devices, Physics and Technology.* Wiley, 1985.

[85] T Takagi *et al.* , "Automatic exposure device and photometry device in a camera," U.S. Patent 5664242, 1997.

[86] A.J.P. Theuwissen, "Solid-State Imaging with Charge-Coupled Devices," Kluwer, Norwell, MA, 1995.

[87] H. Tian, B. A. Fowler, and A. El Gamal, "Analysis of temporal noise in CMOS APS," *Proceedings of SPIE*, Vol. 3649, pp. 177-185, San Jose, CA, 1999.

[88] H. Tian, X. Q. Liu, S. H. Lim, S. Kleinfelder, and A. El Gamal, "Active Pixel Sensors Fabricated in a Standard 0.18um CMOS Technology," *Proceedings of SPIE*, Vol. 4306, pp. 441-449, San Jose, CA, 2001.

[89] S. Tominaga and B. A. Wandell, "Standard surface-reflectance model and illuminant estimation," *Journal of Optical Society America A*, Vol. 6, pp. 576-584, 1989.

[90] B.T. Turko and M. Fardo, "High speed imaging with a tapped solid state sensor," *IEEE Transactions on Nuclear Science*, Vol. 37, No. 2, pp. 320-325, 1990.

[91] B. A. Wandell, "Foundations of Vision," Sinauer Associates, Inc., Sunderland, Massachusetts, 1995.

[92] William Wolfe, "Introduction to Radiometry," SPIE, July 1998.

[93] H.-S. Wong, "Technology and Device Scaling Considerations for CMOS Imagers," *IEEE Transactions on Electron Devices* Vol. 43, No. 12, pp. 2131-2142, 1996.

[94] H.S. Wong. "CMOS active pixel image sensors fabricated using a 1.8V 0.25um CMOS technology," *Proceedings of International Electron Devices Meeting*, pp. 915-918, 1996.

[95] S.-G. Wuu, D.-N. Yaung, C.-H. Tseng, H.-C. Chien, C. S. Wang, Y.-K. Fang, C.-K. Chang, C. G. Sodini, Y.-K. Hsaio, C.-K. Chang, and B. Chang, "High Performance 0.25-um CMOS Color Imager Technology with Non-silicide Source/Drain Pixel," *IEDM Technical Digest*, pp. 30.5.1-30.5.4, 2000.

[96] S.-G. Wuu, H.-C. Chien, D.-N. Yaung, C.-H. Tseng, C. S. Wang, C.-K. Chang, and Y.-K. Hsaio, "A High Performance Active Pixel Sensor with 0.18um CMOS Color Imager Technology," *IEDM Technical Digest*, pp. 24.3.1-24.3.4, 2001.

[97] O. Yadid-Pecht and E. Fossum, "Wide intrascene dynamic range CMOS APS using dual sampling," *IEEE Trans. on Electron Devices*, Vol. 44, No. 10, pp. 1721-1723, October 1997.

[98] O. Yadid-Pecht et al. "Optimization of noise and responsivity in CMOS active pixel sensors for detection of ultra low–light levels," *Proceedings of the SPIE*, Vol. 3019, pp. 125-136, 1997.

[99] T. Yamada, Y.G. Kim, H. Wakoh, T. Toma, T. Sakamoto, K. Ogawa, E. Okamoto, K. Masukane, K. Oda and M. Inuiya, "A Progressive Scan CCD Imager for DSC Applications," *2000 ISSCC Digest of Technical Papers*, Vol. 43, pp. 110-111, February 2000.

[100] M. Yamawaki *et al.* "A pixel size shrinkage of amplified MOS imager with two-line mixing," *IEEE Transactions on Electron Devices*, Vol. 43, No. 5, pp. 713-719, 1996.

[101] D. Yang, A. El Gamal, B. Fowler, and H. Tian, "A 640x512 CMOS image sensor with ultra-wide dynamic range floating-point pixel level ADC," *IEEE Journal of Solid-State Circuits*, Vol. 34, No. 12, pp. 1821-1834, December 1999.

[102] D. Yang and A. El Gamal, "Comparative Analysis of SNR for Image Sensors with Enhanced Dynamic Range," *Proceedings of SPIE*, Vol. 3649, pp. 197-221, San Jose, CA, January 1999.

[103] D. Yang, B. Fowler, and A. El Gamal. "A Nyquist Rate Pixel Level ADC for CMOS Image Sensors," *Proc. IEEE 1998 Custom Integrated Circuits Conference*, pp. 237 -240, 1998.

[104] D. Yang, B. Fowler, A. El Gamal and H. Tian. "A 640×512 CMOS Image Sensor with Ultra Wide Dynamic Range Fl oating Point Pixel Level ADC," *ISSCC Digest of Technical Papers*, Vol. 3650, 1999.

[105] D. Yang, B. Fowler and A. El Gamal. "A Nyquist Rate Pixel Level ADC for CMOS Image Sensors," *IEEE Journal of Solid State Circuits*, pp. 348-356, 1999.

[106] D. Yang, B. Fowler, and A. El Gamal. "A 128×128 CMOS Image Sensor with Multiplexed Pixel Level A/D Conversion," *CICC96*, 1996.

[107] W. Yang. "A Wide-Dynamic-Range Low-Power photosensor Array," *ISSCC Digest of Technical Papers*, 1994.

[108] Kazuya Yonemoto, *et al.* "A CMOS Image Sensor with a Simple FPN-Reduction Technology and a Hole-Accumulated Diode," *ISSCC2000 Technical Digest*, Vol. 43. pp. 102–103, 2000

[109] X. Zhang and B. A. Wandell, "A Spatial Extension of CIELAB for Digital Color Image Reproduction," *Society for Information Display Symposium Technical Digest* Vol. 27, pp. 731-734, 1996.