A FRAMELESS IMAGING SYSTEM: ARCHITECTURAL DESIGN AND SIMULATION

Pintian Zhang

A thesis submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in the Department of Computer Science.

Chapel Hill
2016

Approved by:

Montek Singh
Ketan Mayer-Patel
Fabian Monrose
ABSTRACT

Pintian Zhang: A Frameless Imaging System: Architectural Design and Simulation
(Under the direction of Montek Singh)

This report presents the work on the architectural design and simulation of the IMAGIN sensing system, a novel frameless camera sensor system that can provide a high dynamic range and fine tonal sensitivity, and allow for different update rates from different regions-of-interest. My contribution to the IMAGIN project is threefold. First, starting with a high-level functional concept of the sensor that had been developed by the group, I took it through several design and implementation steps: from functional model to top-level architecture, to pixel-level microarchitecture, to gate-level design of the core sensor array. Second, I validated the behavior and characterized the performance of the sensor via extensive simulation using state-of-the-art tools (Verilog simulation using Xilinx’s Vivado suite). Based on simulation results, I refined and optimized the design by tuning several parameters. Third, I developed equations to compute the dynamic range and tonal sensitivity from design parameters.
# TABLE OF CONTENTS

LIST OF FIGURES .......................................................................................................................... viii

LIST OF ABBREVIATIONS ............................................................................................................... ix

LIST OF SYMBOLS ........................................................................................................................ x

1 Introduction .................................................................................................................................... 1

1.1 Motivation .................................................................................................................................. 1

1.1.1 High Dynamic Range (HDR) ............................................................................................... 1

1.1.2 Smooth Intensity Gradation (Tonal Sensitivity) ................................................................. 2

1.1.3 Frameless Video .................................................................................................................... 3

1.2 Organization of this document ................................................................................................. 4

2 Background: Traditional Sensors and their Limitations ............................................................... 5

2.1 Typical organization .................................................................................................................. 5

2.2 Architecture of a pixel .............................................................................................................. 5

2.2.1 Photodetector ....................................................................................................................... 6

2.2.2 Capacitor ............................................................................................................................. 7

2.2.3 In-pixel processing .............................................................................................................. 7

2.3 Limitations of Traditional Approaches .................................................................................... 8

2.3.1 Analogy: Measure rainfall ................................................................................................... 8
2.3.2 Rain gauge .......................................................................................................................... 9

2.3.3 Tipping-bucket rain gauge ............................................................................................... 11

3 A Better Approach: UNC’s IMAGIN Sensor ........................................................................... 13

3.1 Key Idea ............................................................................................................................... 13

3.2 Design overview ................................................................................................................ 13

3.3 Key components ................................................................................................................ 14

3.3.1 Asynchronous pipeline .................................................................................................. 14

3.3.2 In-pixel frequency decimation ...................................................................................... 15

4 My Contribution: Overview ................................................................................................. 18

4.1 Architectural Design .......................................................................................................... 18

4.1.1 From top-level behavioral concept to detailed microarchitecture ......................... 18

4.1.2 Converted microarchitecture into gate-level model using Mousetrap .................... 20

4.1.3 For off-sensor processing: developed a behavioral mode ........................................ 21

4.1.4 Explored design space for efficient implementation .................................................. 21

4.2 Simulation .......................................................................................................................... 21

4.2.1 Still image & video and Parameter tuning .................................................................. 21

4.3 Analysis .............................................................................................................................. 21

4.3.1 Dynamic range & Intensity gradation ........................................................................ 21

4.4 Collaboration ..................................................................................................................... 21

5 Architectural Design Detail ................................................................................................. 23
5.1 Column pipeline .................................................................................................................. 23

5.1.1 Arbitration and Conditional select ............................................................................... 23

5.2 A Pixel ................................................................................................................................ 23

5.2.1 Photodetector & capacitor .............................................................................................. 23

5.2.2 Decimator ....................................................................................................................... 23

6 Simulation ................................................................................................................................ 25

6.1 Simulation environment ....................................................................................................... 25

6.1.1 Verilog language, Vivado tool suite from Xilinx ............................................................ 25

6.2 Approach ................................................................................................................................ 25

6.2.1 Still image for HDR ........................................................................................................ 25

6.2.2 Moving video for frameless ............................................................................................ 26

6.2.3 Decimation strategy ......................................................................................................... 27

6.3 Results .................................................................................................................................. 27

6.3.1 Still image ........................................................................................................................ 27

6.3.4 Moving video .................................................................................................................... 29

7 Analysis .................................................................................................................................... 30

7.1 HDR ..................................................................................................................................... 30

7.1.1 Equations ........................................................................................................................ 30

7.1.2 Result ................................................................................................................................ 31

7.2 Gradation (Tonal Sensitivity) .............................................................................................. 32
7.2.1 Equation ........................................................................................................... 32

7.2.2 Result .................................................................................................................. 33

8 Conclusion and Future work ...................................................................................... 36

8.1 Summary of Contribution ...................................................................................... 36

8.2 Key Results and Conclusions ............................................................................... 36

8.3 Future work ............................................................................................................ 37

REFERENCES ............................................................................................................. 38
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Images of varying exposure and dynamic range.</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Varying precision in brightness gradation.</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Typical structure of photo-sensor.</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Smartphone camera pixel count trend over the years.</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>Rain gauge for measuring rainfall.</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>Size dilemma of gauge reading approach.</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>A tipping-bucket rain sensor.</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>Column pipeline of the sensor.</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>High-level design of the sensor column.</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>Microarchitecture of the pixel and column node.</td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>High level concept of the sensor codec system.</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>Still image simulation.</td>
<td>28</td>
</tr>
<tr>
<td>13</td>
<td>Frames from the moving image simulation.</td>
<td>29</td>
</tr>
<tr>
<td>14</td>
<td>Dynamic Range of cameras from 2002 to 2016.</td>
<td>32</td>
</tr>
<tr>
<td>15</td>
<td>Relative changes in intensity comparison between existing sensor and IMAGIN sensor</td>
<td>35</td>
</tr>
</tbody>
</table>
## LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHz</td>
<td>Gigahertz</td>
</tr>
<tr>
<td>HDR</td>
<td>High Dynamic Range</td>
</tr>
<tr>
<td>MHz</td>
<td>Megahertz</td>
</tr>
<tr>
<td>ms</td>
<td>Millisecond</td>
</tr>
<tr>
<td>ns</td>
<td>Nanosecond</td>
</tr>
<tr>
<td>s</td>
<td>Second</td>
</tr>
</tbody>
</table>
LIST OF SYMBOLS

C  Size of capacitor

$dp$  The differential change in perception.

dS  The differential increase in the stimulus

$I$  Luminance intensity

$K$  Experiment constant.

$Q$  Charge of capacitor

$S$  The instantaneous stimulus.

$t$  Time between capacitor fill events

$V$  Supply voltage

$MHz$  Megahertz
1 Introduction

1.1 Motivation

Despite the advances in camera sensor technology, there is an unsatisfied consumer desire for greater dynamic range as demonstrated by the proliferation of software apps to obtain high dynamic range images. Recently, a new sensor called IMAGIN was introduced by our research group at UNC [4], which utilizes an entirely new approach for measurement of intensities—one based on “spiking” pixels—and promises drastically better quality of the visuals captured while remaining cost and power efficient. The highlights of the IMAGIN system are high dynamic range image capture, precise luminance gradation, and native support of frameless video.

1.1.1 High Dynamic Range (HDR)

The human eye, through adaptation of the iris and other methods, can adjust to perceive a wide range of brightness present in our everyday life. Although it's desired to be able to capture the that same range of luminance with a camera, standard digital imaging and photographic technique falls short in producing visual with such high dynamic range. This difference between our eyes and average grade camera can be seen in the images below. While our eye can adjust and effectively perceive both the direct sunlight in the background and the shade in the foreground, cameras with standard or low dynamic range would lose information for either the higher end or lower end of the luminance spectrum (Figure 1). High dynamic range, or HDR, images can represent range of brightness higher than that can be achieved using more traditional methods. Currently, HDR imaging is often done in post processing by taking multiple different narrow range exposure of the same scene and then combine into a single HDR image. The UNC IMAGIN sensor model intend increase the dynamic range of the camera sensor during capture.
Figure 1. Images of varying exposure and dynamic range. Standard dynamic range and short exposure (top left) loses the dark region in the foreground. Standard dynamic range and long exposure (top right) loses the bright region in the background. High dynamic range image (bottom) can capture both the light and dark region.¹

1.1.2 Smooth Intensity Gradation (Tonal Sensitivity)

As the display technology improved over the decades, our monitors and TVs are capable of showing more shades of gradient. As seen in the image below, as the gradation of the brightness improve, the transition become smoother and similar to the continuous scene in the real world. This measure of

quality can also be applied to the photo-sensors of camera. Arguably, the luminance gradation or sensitivity of a camera sensor is even more important than that of the display as a camera with coarse gradation will bottleneck display. The brightness sensitivity of a camera sensor is dependent upon the precision of analog to digital conversion. Precise A-to-D conversion in a traditional imaging scheme can be very area intensive. This problematic as a camera sensor should be allocating as much circuitry area as possible for the photodetectors to collect light. Our sensor model will explore an A-to-D approach that result in smooth luminance gradation while being area efficient.

![Figure 2. Varying precision in brightness gradation. Lowest precision at the left and highest at the right.](https://en.wikipedia.org/wiki/Colour_banding)

1.1.3 Frameless Video

Currently the way in which we capture, encode, and display visuals are done in a frame-oriented (or, “frame-ful”) manner. In the frame-ful scheme, the temporal transition of a visuals are represented by periodic snapshots of the entire scene. While intuitive and widely adopted, there are many limitation of this scheme. For video processing, one such limitation is that there are built in support for discriminating a region of pixel over others. This can be inefficient during encoding as not all region are of the same

---

importance during playback, but the entire scene refreshes every frame. Under the frame-ful scheme, in order to increase the visual fidelity when recording video, a camera sensor and encoder would either have to use much more data than necessary to capture an entire frame in the desired quality or apply non-trivial post process-optimization. Alternatively, a frameless way of capturing and encoding video would lay the foundation for intuitive pixel/region specific processing. The frameless framework would allow for different bandwidth allocation to different part of the screen. And for live streams, the encoder can interface with the sensor to dynamically adjust the throughput of desired pixel and/or region. The UNC IMAGIN design incorporates the frameless concepts and provides interface to sensor encoder communication.

1.2 Organization of this document

The rest of this document is organized as follows. Section 2 provides background on traditional sensors and their limitations. Section 3, introduces UNC’s IMAGIN frameless sensing approach. Section 4 gives an overview of my contribution, with details in subsequent sections: Section 5 (architectural design), Section 6 (simulation), and Section 7 (analysis). Finally, Section 8 gives conclusions and some directions for future work.
2 Background: Traditional Sensors and their Limitations

2.1 Typical organization

A typical camera sensor (Figure 3) is structured as a matrix of pixel-photodetectors. The main processing logic is usually separated from the photodetector array. The two modules are typically connected via communication pipelines that run through a column of pixels.

2.2 Architecture of a pixel

Within each pixel module, there are three key components: the photodetector, the capacitor, and the in-pixel processing unit.

![Figure 3. Typical structure of photo-sensor.](image)
2.2.1 Photodetector

The photodetector is the component that converts the luminance perceived by the pixel into electrical currents. A typical photodetector is a photodiode, a semiconductor that, when struck by sufficient energy, knocks off electrons and generates photocurrent. The amount of current generated correlates positively with the surface area of the diode. For a camera, large photodetector surface means smaller exposure time and less noise. As such, a pixel sensor’s design prioritize allocating as much area as possible for the photodetector in order to collect as much light as possible.

Determining the optimal size of photodetector is non-trivial. We want to maintain or improve quality of image taken even as area allocated for photo-sensor gets smaller. Perhaps the device that highlight this dynamic the most is smartphone. Figure 4 below shows a table with the camera pixel count of the flagship smartphone from 2008 to 2013. The table demarcate a clear increase in camera pixel count for these portable devices. However these smartphone cameras typically allocate very little real estate for the camera sensor, meaning that with every iteration, more pixels need to be packed inside the sensor chip. For instance, the main back side camera the Samsung Galaxy S6 released in 2015 is able to take picture with resolution of 16 Megapixels using a 1/2.6” photo-sensor [9]. For the cameras on portable devices such as smartphones, the trade-off between allocating area to photodetector and in-pixel processing becomes more significant. Given that a larger detector leads to better light collection, it is important to limit the size of the processing unit within each pixel.
2.2.2 Capacitor

The capacitor is a buffer that collects the charge generated by the photodetector. The energy is stored in an electric field and will be processed by the in-pixel processor into meaningful data about the visual.

2.2.3 In-pixel processing

Before the data is fed into the communication pipeline, it first goes through an in-pixel processing unit that formats the raw electrical charges into meaningful data for the processor. One of the decisions that must be made in the design of in-pixel processor is whether or not to apply analog to digital conversion (ADC) before sending the data into the column network.

---

*Per-pixel ADC:* The pixel processor can converts the raw electrical charge into a digital value before forwarding them to the pipeline. The advantage of performing ADC before the data are put on the column network is that the network can effectively pipeline digital value, which increases the throughput of the network. Digital value is also much more prone to noise than its analog counterpart. However implementing precise ADC within each pixel is area demanding. Have A-to-D conversion within every single pixel usually leads to compromise in the precision of the ADC to achieve smaller circuit size. This compromise would then leads to a less precise conversion, resulting coarse luminance gradation.

*Per-column ADC:* Alternatively, a sensor model can delegate the digital conversion to the more powerful main processor unit separate from the pixel matrix. The main processor does not compete for circuitry area with the photodetector, which allows for a more powerful ADC unit shared between multiple pixels. However, this requires the column network to relay analog signal. Since the network cannot buffer analog signal, only one pixel can send information down the network at any given time, which would lower the maximum throughput. The analog data are also more subjected to noise.

### 2.3 Limitations of Traditional Approaches

#### 2.3.1 Analogy: Measure rainfall

There are many existing ways of processing the data from the photodetector. An intuitive analogy for this is the process of measuring rainfall. One way to measure rainfall is using a standard rain gauge with graduated marking on the side. The gauge collects rain much like the capacitor for a photodetector. The water it collects is then used to determine the heaviness of the rain. There are two main rainfall measuring approach, each with its corresponding approach in image sensing: the rain gauge reading approach and the tipping bucket approach.
2.3.2 Rain gauge

The gauge reading approach pre-elect a fixed observation period and measure the fullness of the rain gauge afterwards to determine the heaviness of the rain. Similarly, in photo sensing, the sensor model derive the perceived luminance by measuring the fill of the capacitor.

The advantage of processing data using the gauge reading approach is that there are no compromise in the precision of the data collected. Because the data in question is kept in a continuous format, it is as true to the value captured by the photodetector as possible.

A drawback to the gauge reading approach is that it requires a very efficient ADC module to reap the benefit in the precision of the data collected. Because it is generally desired to transmit digital signals in the column network, The ADC unit would need to be implemented within each pixel. Thus, in order to perform high quality ADC, the in-pixel processing unit would need to occupy greater area of the circuit, which would lead to less area for the photodetector.

Another issue to consider for the gauge reading approach is the size of the capacitor. The size of the capacitor cannot be too small, as very bright intensity will quickly fill up the capacitor and result in data loss from overflow. The capacitor also cannot be too big since a large capacitor would require a higher amount of current to raise the voltage from 0 to a distinguishable level. This would cause the capacitors of

---

the dark pixels to fill very slowly. Large capacitor will be unable to distinguishable shades of dark unless the sensor run for extended period of time, which could lead the brighter pixels to overflow again.

It is worth noting that although determining the size of the capacitor is a non-trivial decision, the capacitor size itself does not affect the dynamic range of the sensor. As seen in the rearranged capacitance equation below (Equation 1), the charge is positively correlated with the capacitance and the voltage. However, capacitance is predetermined and will not change during image capture. Any changes in the capacitor charge would solely be reflected in changes of the voltage. Thus, the capacitor size will not have any effect on the Dynamic range ratio. To increase the HDR ratio, either the maximum volt be made bigger or the minimum voltage made smaller. Increasing the voltage maximum is not practical as it lowers battery life. Typical camera sensor runs on 1 to 1.25 volt with about 12 to 14 bit of HDR. A smartphone camera running on 1V with 12 bit HDR would have a minimum voltage of $2^{-12}$. Lowering the minimum voltage for such a device would require a more powerful analog to digital converter that would take up larger area of the sensor.

$$ Q = CV $$

*Equation 1. Capacitance formula.*

*Here, $Q$ = charge, $C$ = capacitance (size of capacitor), and $V$ = supply voltage.*

$$ Q_{\text{max}} = CV_{\text{max}} $$

*Equation 2. Maximum charge of the photodetector capacitor.*

$$ Q_{\text{min}} = CV_{\text{min}} $$

*Equation 3. Minimum charge of the photodetector capacitor.*
\[
\text{HDR Ratio} = \frac{Q_{\text{max}}}{Q_{\text{min}}} = \frac{V_{\text{max}}}{V_{\text{min}}}
\] (4)

*Equation 4. HDR ratio derived from equation 2 and 3.*

---

**Figure 6. Size dilemma of gauge reading approach. Small gauge overflow easily. Large gauge takes long time to fill for drizzle.**

---

### 2.3.3 Tipping-bucket rain gauge

A more sophisticated approach to measuring rainfall is the tipping bucket approach. A diagram of this approach is shown below. In this approach a small bucket is used and as the bucket fills to a fixed level, the bucket will tip over, releasing all the water, and a magnet will record the “tipping” of the bucket. At the end of the fixed observation interval, the number of bucket fill will be used to calculate the heaviness of the rainfall. Like the tipping bucket sensor for rainfall, a photo-sensor can use this approach by setting a small sized capacitor that empties as soon as it is full and record the number of fills in a fixed observation interval.

---

5 Image taken from [https://commons.wikimedia.org/wiki/File:Rain_Gauge_Clip_Art.jpg](https://commons.wikimedia.org/wiki/File:Rain_Gauge_Clip_Art.jpg)
Unlike the gauge reading approach, the tipping bucket approach does not have to worry about the size of the capacitor. The size of the capacitor should be as small as possible improve the precision of data discretization. Since the capacitors are meant to fill up and be emptied again, overflow is not an issue.

There are drawbacks to the tipping bucket approach as well. One of the most obvious drawbacks is coarse discretization. Because we are counting the number of discrete event over a fixed interval, any half-filled buckets at the end of the observation period are not taken into consideration when processing the image. This affects dark pixels the most as they generate less bucket fill event over the observation period. Thus, the ratio between brightness loss and brightness collected would be very high for the dark pixels (Equation 5). This lead dark pixels to round off into small sets of brightness and shown as color banding on display.

\[
\frac{\Delta I}{I} = \frac{\Delta \text{BucketCount}}{\text{BucketCount}}
\]

*Equation 5. Relative change in intensity for Tipping bucket approach.*

---

6 Image taken from [http://www.kifissiameteo.gr/Lesson08_Instrument_RainGauge.html](http://www.kifissiameteo.gr/Lesson08_Instrument_RainGauge.html)
3 A Better Approach: UNC’s IMAGIN Sensor

3.1 Key Idea

The UNC IMAGIN sensor model explores an alternative approach in the processing of photocurrent. The approach uses a small sized capacitor that can be emptied when full similar to the tipping bucket method. However, instead of counting the number of bucket fills within a set observation period, the delay between bucket fill is recorded. This delay is inversely proportional to the intensity observed by the pixel.

Similar to the tipping bucket method, there is no tradeoff for specifying the bucket size. Since the capacitor will never overflow, the size can be set reasonably small to minimize exposure time.

Unlike the tipping bucket method, the IMAGIN approach can produce data with much finer precision. In our approach, we derive the perceived luminance intensity using the delay between bucket fill. Since the observation interval is not used in the calculation, it has no effect on the precision of the result. The precision of the derived intensity to the image perceived would depend on the precision of the internal clock used for timestamp, which will have a much finer grained step size than that of the tipping bucket approach.

3.2 Design overview

The top-level design for the IMAGIN model is the same as that of the typical photo-sensor. The pixels would be organized into a matrix structure linked by a row of asynchronous column communication pipeline. Each pixel will process the photocurrent from the photodetector using the tipping bucket method. As a measure of rate control, the bucket tick events will go through an event decimator before entering the column pipeline as a measure against network congestion.
3.3 Key components

3.3.1 Asynchronous pipeline

The column communication pipeline is designed to run asynchronously. The network consist of nodes that each connects to a pixel. The column nodes communicate and forward data via a request and acknowledgement handshake that allow the pipeline to function efficiently without the need of a clock [8].

The job of each column node is to merge traffic from nodes above and neighboring pixel and forward data to the next node of the pipeline. In a case of data arriving at a node from both input at the same time, the node will randomly arbitrate which data to forward. Acknowledgement will not be generated to the wire not chosen by the arbiter, effectively freezing that input pipeline until the node is free to forward the data again.
The data being transmitted consist of the pixel address, decimation value, and other necessary handshake signals. The column pipeline does not carry timestamp as timestamping can be done in the main processing unit at the bottom. This avoid the need to route a fast clock to every single pixel of the sensor.

Since data will be timestamped after it exits the pipeline, significant jitter within pipeline will skew the data. However, network delay does not affect the data since it does not change the time between bucket fill. Any amount of network delay would be canceled out during the difference calculation unless there is network jitter. Fortunately the asynchronous pipeline has shown to have minimal jitter under normal load. Earlier functional simulation by Prof. Montek Singh have only about 1ns of jitter is observed in the column pipeline running at a sub 90% throughput. Any collision can also be quickly arbitrated and resolved without waiting for the clock. Occasional arbitration delay has also shown to be negligible on average.

Lastly, an asynchronous system can be much more power efficient than the synchronous counterpart [11] [12]. Using the request/acknowledgment handshake, there is no need to route a precise clock to the network. For a synchronous network to achieve similar maximum throughput, it would need to run on a very fast clock, which would be taxing on power consumption. Our asynchronous node current has a cycle of about 5ns. To achieve the same maximum throughput using a synchronous pipeline would require a 200 MHz clock routed to each single pipeline node.

3.3.2 In-pixel frequency decimation

Event decimation is another key component of the IMAGIN photo-sensor. The main purpose of the decimator unit is to deal with high intensity inputs. For very bright pixel, the capacitor will generate raw bucket fill event very rapidly. If these raw events are forwarded directly into the column pipeline, the network would likely congest and induce jitter into the network. The decimator would act as congestion control for the system by grouping multiple raw events into a single decimated event. This can be done using a counter. The decimated event will contain the decimation value so that the bottom processor will be able to decode back the raw events.

Aside from congestion control, per-pixel decimation also improves the dynamic range of the sensor. Since each pixel has its own decimator, the bright pixels can adjust to a higher decimation value to
effectively capture the luminance while darker pixels will keep their decimation value low to generate events at a timely pace. This allows both the bright and dark part of the scene to be captured at the same time.

Lastly, decimation can be used as a rate control parameter for pixel update rate. A pixel with low decimation value will generate event frequently and while pixels with high decimation value will generate event at a slower pace. This can be useful in video capture where it's often desirable to have a region of the screen, such as where the main character is, to update at a quicker rate than the rest of the screen.

Implementation

We have considered two approaches in the implementation of the decimator. The first approach is to have the decimator unit receive decimation value from an external module, likely the bottom processor or even a processor outside of the photo-sensor. The second approach is to have each decimator regulate its own decimation value.

The external setting implementation has the advantage of providing an interface for the sensor processor or an external processor to adjust the pixel update rate. The external processor can lower the decimation value for a region of interest in order to achieve faster update rate for that region only.

The self-regulated decimator would adjust its decimation value by probing the rate of which decimated events are being generated. This implementation would need to have a fixed observation interval for probing. The decimator would also need a range of tolerable output amount for the given tau. If the pixel produced more decimated event within a period of tau than that specified by the tolerance range, then the decimation value will increase and vice versa until the desired rate is achieved.

A question to consider when implementing the decimator is when to incorporate a new decimation value mid-run. One of the way is to set the new decimation value when the current counting using the old decimation value finishes. However, this may not be desirable in situation when a pixel’s brightness drastically decrease. In this situation, the decimation value need to decrease as well. However because the
old decimation value is very big, the decimator can take a very long time to produce an event and set the new value.

Another way to set the new decimation value is to clear out the counter and set the new decimation value right away. This approach risk data loss for the benefit of immediate setting of the value. This approach could be viable for still images as the camera can set aside some time for the decimators to converge to the optimal decimation value before taking the shot. Vanilla implementation of this approach would not be suited for video capture due to possibility of data loss mid-recording.

Another approach is to bound the current counter value to the next digit of the counter. Once the counter reached that bounded digit, it will prematurely output a decimated event with the corresponding bounded decimated value and set the new decimation value. This approach does not have to wait a long time for the current count to finish and will not have data loss. However, efficient gate-level implementation may be challenging.
My contribution to the IMAGIN project were threefold. First, I helped flesh out the gate level details of the existing architectural design. With the design, I then developed a gate level model sensor model. Second, I validated the behavior and characterized the performance of the sensor via extensive simulation using state-of-the-art tools (Verilog simulation using Xilinx’s Vivado suite). Third, I developed equations to compute the dynamic range and tonal sensitivity from design parameters.

4.1 Architectural Design

4.1.1 From top-level behavioral concept to detailed microarchitecture

Given the top-level behavioral concept of the sensor (Figure 9), I developed a microarchitecture of the design. I specified the wiring of the request & acknowledgment handshake between column nodes and the interface between arbitration and selection.
Figure 9. High-level design of the sensor column.
4.1.2 Converted microarchitecture into gate-level model using Mousetrap

With the microarchitectural design, I then proceeded to create a gate level model in Verilog using the Mousetrap Asynchronous Primitives developed by Singh and Gill [2]. The arbiter unit chooses takes two input and only forward one at a time based on which input came first. On collision, the arbiter will decide randomly which to forward immediately. The conditional select unit takes a selection wire (from the arbiter in this case) and forward one out of two input data base on that selection. The original design of the arbiter and conditional were not designed to work with each other. I redesigned the arbiter and conditional select module to have them interface efficiently with each other, take up less circuitry area, and run faster for our application.
4.1.3 For off-sensor processing: developed a behavioral mode

For logic units that were not core to the sensor model, such as the bottom logic, I chose to develop them as behavioral model. This is done by algorithmically describing the operation of the module in Verilog. The compiler will take those description and synthesize a circuit design. Another module that was developed behaviorally was the decimator. Because there are different flavor of implementation strategies, I’ve decided to hold off on a gate-level implementation until we decided on a winner.

4.1.4 Explored design space for efficient implementation

I also explored the design space for an efficient implementation of the bounded decimator unit. I have drafted a gate-level design for a decimator that, when received a new decimation value, would immediately bound the current counter to the next reachable digit and set the new decimation value. The design, however, still needs to be further analyzed for correctness and implemented into the simulation model. See section 6 for details.

4.2 Simulation

4.2.1 Still image & video and Parameter tuning

After developing the Verilog model, I took the sensor and ran extensive simulation to collect data. I ran the simulator with both still and moving image input. I also adjusted different parameters for the sensor model to observe their effect on the quality of the image captured.

4.3 Analysis

4.3.1 Dynamic range & Intensity gradation

Finally, using the simulation data, I performed analysis on the HDR ratio and the precision of the brightness gradation achievable by the sensor model.

4.4 Collaboration

Aside from developing the sensor model, our group is also collaborating with Prof. Mayer-Patel and his student Aaron Smith at UNC on a frameless video encoder/decoder. This collaborative project aims to effectively utilize the frameless structure of the sensor in video encoding. The encoder codec will receive per-pixel update data from the photodetectors. The codec will encode the data in a frameless manner. Aside
from encoding the data, the codec will also be able to communicate back to the sensor new decimation values for a specific region on screen. This will allow for dynamic rate control for regions of interest during live encoding. The development is still in its early stage. I have so far adjusted the sensor processor to output the video data in a frameless trace that can be used for the development of the codec.

*Figure 11. High level concept of the sensor codec system.*
5 Architectural Design Detail

5.1 Column pipeline

5.1.1 Arbitration and Conditional select

The asynchronous column pipeline was constructed using modified versions of the Mousetrap primitives designed by Singh and Gill [2]. The arbiter unit utilizes a mutex primitive underneath and the conditional select unit takes the choice from the arbiter and forward the corresponding data to the output wire. The request and acknowledgement wires are both two phase toggle. The data itself is the pixel address. Notice that the wire for the pixel address can easily be adjusted to function as a two way communication line used to relay new decimation value from an external processor.

The total estimated cycle time for the column node in our Verilog model is around 5 ns. This means that the maximum throughput of the pipeline is about 200 million events per second. This level of throughput is comparable to that of a synchronous system running on a Gigahertz clock. However, the asynchronous pipeline has the advantage of power efficient and native queuing support.

5.2 A Pixel

5.2.1 Photodetector & capacitor

The photodetector and capacitor within each pixel unit are implemented base on the tipping bucket design. The unit simulates bucket fill against a given intensity input, whether it was a static intensity for still image or a changing intensity for video. When a capacitor is filled, it will clear out the current bucket and output a raw bucket fill event to the decimator. The photodetector and capacitor are both behavioral implementation since the IMAGIN model can interface with existing optimized models available.

5.2.2 Decimator

The decimation unit utilizes a counter to group several raw event into a single decimated event. The decimation value are designed to be powers of two and will always represent one of the digit in the
internal counter. Once the counter reaches the digit specified by the decimation value, it will clear itself and generate a decimated event. The module is implemented behaviorally. The gate-level implementation will be one of the future objective of the project.

The model decimator support both self-regulation and static external setting of the decimation value. For the self-regulation mode, the module is given a static probing period and an event tolerance range. The decimator will adjust its decimation value up or down if the number of decimated value generated does not fall within the tolerance range. Our model currently specifies the tolerance range to be between 2 and 4. For the external setting mode, a fixed decimation value is given to the decimator beforehand.

Any new decimation value is updated after a decimated event has been generated. Although I have drafted a decimator design that can bound the counter as new decimation value comes in, there has not been enough time to validate the design and implement it. However, this would also be one of the future work of the project.
6 Simulation

6.1 Simulation environment

6.1.1 Verilog language, Vivado tool suite from Xilinx

The model and simulation are done in Verilog, a Hardware Description Language, using the Vivado tool suite from Xilinx. All of the simulations were done on a windows machine.

6.2 Approach

6.2.1 Still image for HDR

The simulation will have two different types of input, still images and moving images. The still image input is used to test for the HDR quality of the sensor. The still image used is a ray traced image of a library scene model using Blender, a 3d modeling software. This image has both region of extreme brightness and shade. The decimator will self-regulate its decimation value during the exposure period and output the intensity perceived at the end of the simulation.

During simulation for still image, I tried to tune the capacitor size of the pixels and see its effect on the simulation. The size of the capacitor affected the capture of dark pixels the most. Since larger capacitor will take dark pixel longer to fill, the simulation had to run for longer time to effectively capture the dark areas. For a sensor with small capacitor, the dark pixel will start outputting data faster. It is generally desired for our camera sensor to have a small capacitor because that would effectively reduce the exposure time, which is desirable for a camera because it make possible for capturing quality still image for a moving scene. Something worth noting is that while the minimum capacitor size of the camera is dependent on the decimator and the column throughput, our Verilog simulation had another factor for the floor limit. This was the simulation time scale. We could not simulate a photodetector/capacitor outputting bucket fill event on a bright input if the delay between the raw events ever fell below the timescale step size. For example, say the brightest pixel in the scene can fill about 75,000 capacitor unit per ns and the simulation timescale
was in ns, then the capacitor size should not be less than 75,000 for the simulation else, the delay between raw events will be rounded down to 0.

6.2.2 Moving video for frameless

The moving image input is used to simulate the frameless capture ability of the sensor. The moving images is a simulated cosine waves propagating from a center point on screen. The value of the cosine at each pixel represents the intensity perceived by that pixel at that time.

The moving image simulation utilized both external setting and self-regulation for decimation. The external setting decimator were used to mainly observe the effect on the quality of the output for the development of the frameless encoder in our collaboration Prof. Mayer-Patel’s research group. The value used to the set the decimator was varied from $2^0$ to $2^3$ (i.e., from 1 to 8). Initial results shows slightly more noise in the result when decimation value is low, which is as expected because of higher network traffic produced as a result. The self-regulated decimation was used to test the frameless capture capability of the sensor. An important parameter for this setup was the decimation probing period. The self-regulating decimator will attempt to output 2 events for probing period. Since every decimator output also means update for that pixel, the decimation probing period effectively controls the pixel update rate for the video captured. Using the average frame rate for video as a guide, I chosen the probing interval to be on the order of ms to be a little more aggressive in the pixel update rate. This will lead most pixels, except for the very dark ones, to update roughly on the order of a thousand frames per second.

The simulation data were processed into a frameless output, and also a frame-ful output for backwards compatibility with frame-oriented displays or software. To create a frame-ful output, I introduced a converter module, called a “framifier”, which converts frameless video into a frame-ful one by resampling the current image according to a specified frame rate. This frame-oriented output can then be used to view the result on a traditional display. For the frameless trace, the processor output a list of pixel-timestamp pairs to represent the per-pixel update of the sensor. This trace is intended to be used for the development of the sensor-encoder interface in our collaborative project with Prof. Mayer-Patel’s research group.
6.2.3 Decimation strategy

In terms of decimation, both self-regulation and static external setting were tested for the moving image simulations. The self-regulated decimation is used to test the soundness of the frameless capture. And the externally setting approach is used to see the effect of different decimation value on the quality of the video captured.

6.3 Results

6.3.1 Still image

For the still image simulation, we tested several small sized input and the dynamic range was very promising. The resolution of the testing image was relatively small due to the sheer amount of time to compile a gate-level model of higher solution, which could take up to orders of days to weeks. For our simulations Maximum decimation value was capped at $2^{15}$. A self-regulated decimator is used with a probing period of $6 \times 10^5$ ns and the tolerance range set between 2 and 4 decimated events per probing period. The simulation exposure time was set to 600 ms, which give ample time for the decimators to converge to the optimal decimation values. Our simulation of a 2x2 image with the ratio between highest and lowest intensity of 14,171,171.05:1 (23.7 bit dynamic range) was able to capture the image with a PSNR of 92.5 dB. As the larger size simulation are still running at the time of writing this thesis, I will show some prior high level functional simulation result of a 512x512 image from our group to highlight the HDR benefit from our sensor (Figure 12). As seen below, the model sensor can capture significant detail in both bright and dark regions, whereas the base case approach captures much less information. The dark region captured in the simulation also show smooth gradation of intensity, whereas the base case approach shows significant banding.
Approach                              Base Case

(a) Comparison of high intensities

(b) Comparison of mid-range intensities

(c) Comparison of dark intensities

*Figure 12. Comparison of our high level functional simulation of a still image (left) and base approach (right). The HDR image captured by each is shown by scaling it to three different levels.*
6.3.4 Moving video

I focused the frameless simulation on visualizing the frameless capturing of the moving image using the framifier. To achieve this I set the per frame delay \((5 \times 10^5 \text{ ns})\) to be 1/12 of that of the decimation probing period \((6 \times 10^6 \text{ ns})\). With the decimation tolerance range of 2 to 4, a pixel will only update its value maximum of 1 time per 3 frame. I ran this model against a 100x100 size screen with a cosine wave propagating from the middle (Figure 13). I also manually kept an area on the left to have the maximum intensity in order to highlight the difference in pixel update rate. To show the frameless nature of the sensor, I ran the simulation once normally and once highlighting pixel that produced new event right before the current frame red. Some frames of the results are shown below.

![Frames from the moving image simulation. The frames on top highlighted in red pixels that have updated in that frame. Frames on the bottom are similar to the ones on top but do not highlight any pixel.](image)

The simulation result were as expected. The perceived brightness for a pixel updates as new events arrive at the bottom process unit. The self-regulating decimator also function as expected, converging the update rate of all pixel to an even rate.
7 Analysis

7.1 HDR

7.1.1 Equations

Maximum Intensity: To derive the expected dynamic range of the sensor model, we would first need to calculate the maximum and minimum intensity achievable. The maximum intensity that the IMAGIN sensor can capture is limited by the maximum throughput of the network (Equation 7). If the pixel is so bright that the capacitor generate bucket fill events faster than the column network can handle even after decimating with the max decimation, then there will be significant jitter and data loss. Thus the maximum intensity that the sensor can support is proportional to the maximum throughput of the communications pipeline. Maximum throughput of the asynchronous pipeline is inverse of the cycle time of a single pipeline node (Equation 6).

Minimum Intensity: The minimum intensity that the sensor can capture depends on the maximum expected delay between bucket fill events (Equation 8). With infinite amount of exposure time, this sensor can capture effectively capture very dark areas. However, in actual application, it's desirable to minimize the exposure time of a camera. The maximum refresh delay of a pixel would reflect the maximum exposure time of a camera.

Dynamic Range: With the maximum and minimum intensity defined, we can calculate the dynamic range ratio of the system. The ratio is obtained by dividing the maximum intensity by the minimum intensity. The equation of the ratio is calculated below (Equation 9). The HDR Ratio correlates positively with the Pipeline throughput, maximum decimation value, and the maximum pixel refresh delay and negatively with the number of node in a column pipeline.
\[ \text{MaxColThroughput} = \frac{1}{\text{TotalLatencyOfPipeline}} \]  
(6)  

*Equation 6. Max column throughput*

\[ \text{MaxIntensity} \leq \frac{\text{MaxColThroughput}}{\text{row#}} \ast \text{MaxDecimation} \]  
(7)  

*Equation 7. Maximum intensity supported by sensor*

\[ \text{MinIntensity} \geq \frac{1}{\text{MaxRefreshDelay}} \]  
(8)  

*Equation 8. Minimum intensity supported by sensor*

\[ \text{HDR Ratio} = \frac{\text{MaxColThroughput}}{\text{row#}} \ast \text{MaxDecimation} \ast \text{MaxRefreshDelay} \]  
(9)  

*Equation 9. HDR of sensor*

### 7.1.2 Result

Given the equation for the HDR Ratio, we can calculate the expected dynamic range ratio for the sensor model. The cycle time for a column pipeline node is estimated to be about 5ns. The maximum decimation value used in the simulation is \(2^{15}\). Assuming a maximum pixel refresh delay of 1ms, which is much faster than a frame update for a typical video, and an image dimension of 5000x5000 (25 megapixels), the ratio is about 1,310,720:1 or \(2^{20.3}:1\). Thus, the dynamic range is (slightly) greater than 20 bits, even for a pixel count as high as 25 megapixels. Figure 14 shows the dynamic range of consumer grade cameras (costing less than $1000) with similar pixel counts (i.e., 25 megapixels or less). As seen, these commercial cameras currently only have about 12 to 14 bits of HDR, while our camera sensor is expected to provide 20 bits. Furthermore, note that we have assumed a maximum pixel refresh delay of 1ms for our sensor. This is rather aggressive as most cameras produce video on the order of tens or hundreds frames per second.
If we increase the maximum pixel refresh delay from 1ms to 0.01s, meaning the slowest pixel will now update 100 times per second instead of 1000, the HDR would increase further by a factor of 10, i.e., approximately 13,000,000:1 or 23.6 bits.

![Figure 14. Dynamic Range of cameras under $1000 between 2002 to 2016 with resolution of 25 Megapixels or less. The Y axis is the dynamic range in number of bits [1]. Vertical axis represents dynamic range in bits. Horizontal axis is year of manufacture of the particular camera model.]

### 7.2 Gradation (Tonal Sensitivity)

#### 7.2.1 Equation

To analyze the brightness gradation of the image derived, we can calculate relative change in the perceived intensity. Firstly, intensity derived by the sensor, denoted as \( I \), is correlated inversely with the time between bucket tick \( t \) (Equation 10). The next higher distinguishable intensity value can be calculated by subtracting a clock tick from the delay between bucket events (Equation 11). Using these definition, we can find the relative change of intensity (Equation 13).

---

7 Image taken from [http://www.dxomark.com/](http://www.dxomark.com/)
\[ I = \frac{\text{Const.}}{t} \]  

*Equation 10. Intensity derived from time bucket fill events.*  

\[ I = \text{intensity, and } t = \text{delay between bucket fill events.} \]

\[ I + \Delta I = \frac{\text{Const.}}{t - \Delta t} \]  

*Equation 11. Sum of intensity and intensity step size derived.*  

\[ \Delta I = \text{intensity step size, and } \Delta t = \text{step size of clock} \]

\[ \Delta I = \frac{\text{Const.} \times \Delta t}{t (t - \Delta t)} \]  

*Equation 12. Step size of intensity.*

\[ \frac{\Delta I}{I} = \frac{\Delta t}{(t - \Delta t)} \propto \frac{\Delta t}{t} \]  

*Equation 13. Relative change of intensity.*

7.2.2 Result

Assuming that the delay between decimated bucket fill events averages to be about 1 ms and the timestamp unit utilizes a 1 ns clock, the relative change of intensity is approximately \(10^{-6} (2^{20})\). This result in a much more finely graded intensity, than achievable by a traditional sensor utilizing a 14 bit ADC module, which is only \(2^{-14}\).

Note that this the relative rate of change for the IMAGIN sensor does not change as intensity gets bigger. By comparison, for a traditional sensor, the relative change in intensity as one moves from one
intensity level to the next higher one, is not constant. At the lowest intensity levels, the next higher intensity is quite a large step up: e.g., the relative change from 1 to 2 is 1.0 (100%); from 2 to 3 is 0.5 (50%), etc. In contrast, for our sensor, the relative change in intensity is $2^{20}$ throughout the range of intensities, from the darkest to the brightest. Thus, in the dark regions of the image, the traditional sensor will show unseemly banding because of sudden large relative changes in intensities, but our sensor shows only smooth tonal gradations (of about 1 part in a million between consecutive intensity levels).

Another way to look at the comparison between sensors with respect to intensity variations is to realize that, instead of a fixed step size like traditional sensors, the intensity of the IMAGIN sensor increases in logarithmic step size (Figure 15). This logarithmic step size is desirable as it emulates how human eyes perceive light. This is known as the Weber-Fechner law (Equation 14), where the just noticeable difference between two brightness is proportional to the brightness’s themselves. Two bright pixels differ by some intensity would appear less different than two dark pixels differ by the same amount of intensity [3]. The intensity output from the IMAGIN sensor adhere to the Weber-Fechner law.
Figure 15. Relative changes in intensity comparison between existing sensor and IMAGIN sensor.

\[
\Delta I / I \approx 1 / \text{Max} \approx 2^{-10} \text{ to } 2^{-14}
\]

\[
\Delta I / I = 1/2
\]

\[
\Delta I / I = 1
\]

Typical Existing Sensor

Proposed Sensor


\[
dp = k \frac{dS}{S},
\]

\(dp = \) the differential change in perception.

\(dS = \) the differential increase in the stimulus

\(S = \) the instantaneous stimulus.

\(K =\) experiment constant.
8 Conclusion and Future work

8.1 Summary of Contribution

My contributions for the IMAGIN sensor project were divided between model development, simulation, and analysis. Given the top-level design of the sensor, I developed the microarchitecture and gate-level implementation in Verilog. I explored various strategies for event decimation and drafted a design for a bounded decimator. I took the Verilog model and ran extensive simulation against various set of parameters. Using the simulation parameters and results, I analyzed the dynamic range and brightness gradation of the sensor.

8.2 Key Results and Conclusions

The HDR and brightness sensitivity analysis of the IMAGIN sensor are promising. Given practical parameters, the sensor’s dynamic range was calculated to be around 1,310,720:1 or $2^{20.3}:1$ for an aggressive 1 ms update rate (and even higher for slower update rates). This ratio is a little over 20 bits and is a significant improvement over current consumer grade cameras. The relative change in perceived intensity of the sensor was calculated to be $2^{-20}$ independent of the intensity. This ratio is much more precise than that of the current cameras, which typically has relative intensity variations from 1.0 down to $2^{-10}$ to $2^{-14}$ at best. Furthermore, the IMAGIN sensor has a constant relative sensitivity (i.e., smallest detectable relative intensity change) throughout the intensity spectrum. This allows the brightness level captured by the IMAGIN sensor to vary logarithmically, which is closer to how human eye perceives brightness.

Aside from providing HDR and precise brightness gradation, the IMAGIN sensor also process the pixel data in a frameless manner. This allow for interfacing with frameless video encoder. The per-pixel decimator can also act as a way of rate control for which a video encoder use to dynamically adjust the flow rate for a region of interest mid-encoding. This allows for many practical application of frameless video such as for security camera. The footage captured on a typical security cameras consist of many region of
still sceneries. Using a frameless video system, the processor can tell the sensor to have the still pixels update very slowly or even turn them off until a burst of movement have been detected by those pixels. This would allow for great potential to save space for such video footage.

8.3 Future work

There are several tasks remaining that will be complete in future work. First, the decimator module, currently only described behaviorally, will be implemented as a gate-level circuit. Once the entire sensor model has been converted into a gate-level circuit, chip floor-planning and layout will be performed next. Then the sensor chip will be sent for fabrication, and resulting physical parts will be mounted a test circuit board and evaluated in a real environment to assess its performance. And lastly, the collaboration with Prof. Ketan Mayer-Patel’s research group will continue on the effort to develop efficient techniques for storage, retrieval and transmission of frameless video.
REFERENCES


