Smart Readout Electronics for CMOS Image Sensors

A Dissertation

Presented in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

with a

Major in Electrical Engineering

in the

College of Engineering

University of Idaho

by

Mohamed R. Elmezayen

Major Professor: Suat U. Ay, Ph.D.

Committee Members: Ting-Yen Shih, Ph.D.; Yacine Chakhchoukh, Ph.D.; Ahmed Ibrahim, Ph.D.

Department Administrator: Joseph Law, Ph.D.

December 2020

Authorization to Submit Dissertation

This dissertation of Mohamed R. Elmezayen, submitted for the degree of Doctor of Philosophy with a Major in Electrical Engineering and titled "Smart Readout Electronics for CMOS Image Sensors," has been reviewed in final form. Permission, as indicated by the signatures and dates below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor: _____ Date: _____
Suat U. Ay, Ph.D.

Committee Members: _____ Date: _____
Ting-Yen Shih, Ph.D.

_____ Date: _____
Yacine Chakhchoukh, Ph.D.

_____ Date: _____
Ahmed Ibrahim, Ph.D.

Department Administrator: _____ Date: _____
Joseph Law, Ph.D.

Abstract

A massive number of images and videos are captured by mobile phone cameras every day. Thus, the built-in cameras become essential in almost all produced mobile phones and devices. To compete, device manufacturers improve image sensors' performance in camera modules and optimize it for high frame rate, better image quality, low power consumption, low cost, and added features.

Solid-state image sensor is the core of today's camera systems. Two well-known technologies exist for image sensors, which are the charge-coupled devices (CCD) and complementary metal-oxide-semiconductor (CMOS) image sensors (CIS). CCD was the dominant and mature technology for a long time until CIS appears in the early 1990s to solve CCD drawbacks. CIS has several crucial advantages over CCD, including integrating peripheral circuits on the same chip, minimizing overall system size, and forming a camera system-on-chip (SOC). Besides other advantages, just these advantages of CIS have qualified it to be integrated with the modern mobile phone systems.

Typically, the built-in camera module's power consumption specification is crucial for mobile devices' overall battery life. Besides low-power consumption, image sensors in these camera modules require high-resolution and high-quality image reproduction capabilities to provide a competitive edge for the manufacturers to dominate in the marketplace. Today, CIS is the technology of choice due to its low-power consumption, high resolution, and increased integration capabilities. One of the essential blocks in the CIS that affects its speed, power consumption, image quality, and resolution at the same time is the analog-to-digital converter (ADC). Thus, optimizing and improving the performance of integrated ADCs is crucial for CIS performance. Many ADC types have been used in CIS. One of the most efficient types is the integrating (ramp) type ADCs, as they are small, easy to integrate, consume minimal power, and fulfill high-resolution requirements easily while providing low-noise operation. However, they suffer from the conversion speed problem when bit resolution is increased. Many solutions were proposed to improve the conversion speed of ramp ADCs in CISs while maintaining acceptable bit resolution and, consequently, reproduced image resolution and quality at the same time.

In this research, two speedup techniques (SuPTs) are proposed to improve the conversion speed of ramp-type ADCs integrated with a CIS column-parallel architecture (CPA). They are the single-slope look-ahead ramp (SSLAR) and accelerated single-slope look-ahead ramp (ASSLAR) ADCs. Measurements of the SSLAR SuPT in a 200 × 150 pixel CIS chip showed that a 6x frame rate increase could be achieved while reducing power consumption 13% without compromising image quality. The ASSLAR SuPT, on the other hand, improved the performance of the SSLAR and the well-know accelerated ramp (AR) SuPT by enhancing the speedup ratio (SuPR) by 20% on average

while keeping the structural similarity of the reproduced image not affected by the proposed algorithm and structural similarity index of over 98%.

These new CIS SuPTs have some inherent settings that control speed-up ratio (SuPR) and ADC bit-resolution. As a result, power consumption, frame rate, and image quality of the CIS can intelligently be controlled and optimized. This intelligent control requires predicting the content and complexity of captured scenes without requiring complex computational resources. To do this, a new image quality (IQ) metric, called conversion complexity metric (CCM), was developed to simply and quantitatively measure complexity for any scene captured by CIS or a still image stored on a medium. It provides an index number for smart adjustment of the performance parameters of CIS electronics, including on-chip ADCs. The CCM was proven to be bounded, monotonic, 99% linear, and 316% sensitive. It is a computationally efficient single-image quality metric that no other metrics could provide for CIS to intelligently adjust and optimize on-chip analog and digital signal processing operations. The new CCM can also be used for comparing different SuPT for different ramp ADCs used in CIS CPAs.

A new image quality and complexity comparison methodology is also proposed based on the CCM index and existing image quality metric, known as structural similarity metric (SSIM). This new methodology is proposed to set up a fair comparison between different SuPTs. Each SuPT has its controlling parameters (CPs) to adjust SuPR while trading off some CIS performance parameters, such as reproduced image quality of which, if not considered during the comparison procedure, it may result in a misleading performance advantage. Using the proposed comparison methodology guarantees that a fair comparison and judgment of different SuPTs is possible. A case study was developed to apply this new comparison methodology to compare the new ASSLAR and the existing AR SuPTs. This case study resulted in a process to hybridize these two powerful SuPTs to get superior performance. It was found that for the highly complex images with a CCM index of 0.5 or lower, the AR SuPT has to be used. For less complex images with a CCM value of 0.5 or more, it is beneficial to use the proposed ASSLAR SuPT to provide 20% or more SuPR compared with AR SuPT in CIS CPA.

Acknowledgments

Throughout my Ph. D. study, my research, and this dissertation's writing, I have received a great deal of support and assistance from many sources.

All praises to ALLAH and his blessing for the completion of this thesis. I thank ALLAH for all the opportunities, trials, and strength showered on me to work on this research topic and finish writing this thesis. I acquired so much experience during this process, not only from the academic point of view but also personally. My humblest gratitude to the Holy Prophet Muhammad (Peace be upon him), whose way of life has been continuous guidance for me.

I would first like to thank my supervisor, Professor Suat U. Ay, whose expertise helped me formulate the research questions and methodology. His insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. He also gave me lots of life and work experience that always enrich my mind and experience. I would like to thank him always and forever for supporting and helping me to reach what I was always dreaming about, the Ph.D. degree.

I would like to acknowledge my committee members Dr. Shih, Dr. Chakhchoukh, and Dr. Ibrahim, for their time and flexibility. Their opinions and questions are always pushing me forward and enrich my research and enhance my academic work.

Dedication

I dedicate this thesis to my mom, my dad's soul, my wife, Nourhan, and my beloved sons, Eyad & Adam, and all my family for their constant support and unconditional love. I love you all dearly.

I would also like to thank my friends, Dr. Shreif, Dr. Sobhy, Dr. Ahmed, Dr. Hefieda, Dr. Sameh, and their families for being our wonderful friends and their containment of my family since we arrive in the United States. We never felt that we are away from our home country or family because they compensated for their absence. Thank you all so much, and I hope we keep friends forever.

Table of Contents

List of Tables

List of Figures

Statement of Contributions

This research resulted in advancements in the field of CMOS image sensors. Speed up techniques for ramp ADC are proposed and implemented. They are the single-slope look-ahead ramp ADC and accelerated single-slope look-ahead ramp ADC technologies. Measurements of the SSLAR ADC in a 200x150 pixel CIS showed that a 6x frame rate increase could be achieved while reducing power consumption 13% without compromising image quality. The study and the implementation of ASSLAR showed that the ASSLAR improved the performance of the well-known AR SuPT by enhancing the speedup ratio (SuPR) by 20% on average for less complex images while keeping the structural similarity not affected and the SSIM is over 98% [1].

A new reference-free image conversion complexity metric was developed and tested using hundreds of reference and standard database images. The new metric has proven to be bounded, monotonic, achieves 99% linearity, and 316% sensitivity. It provides a computationally efficient single-image quality metric that no other metrics provide for CIS to intelligently adjust and optimize on-chip analog and digital signal processing operations [2] [3].

A new comparison methodology is proposed based on the CCM index in addition to the SSIM index. This new methodology is proposed to set up a fair comparison and judgment between different SuPTs. A comparison between the ASSLAR and the AR SuPTs is chosen as a case study that proposes the hybridization of these two powerful SuPTs to get a superior performance based on the input image CCM index [4] [5].

Chapter 1: Introduction

CMOS image sensors (CISs) became the dominant image sensor technology through the last several decades. This dominance resulted from its high integration compatibility, low power consumption, and low cost of manufacturing. In contrast to charge-coupled device (CCD) imagers, CIS did not require dedicated manufacturing processes. They follow the mainstream CMOS manufacturing process used today's advanced analog, digital, and mixed-signal integrated circuits (ICs). Indeed, CIS can easily be integrated with analog and digital electronics and result in a complete camera system integrated on a single IC, hence cost, interfacing problems, and power consumption can be reduced dramatically [6].

Historically, CMOS fabrication technology has grown faster than any technology man developed. Integrated transistors became smaller, faster, and cheaper over the past 50 years, which allowed integration of more and more components on-chip leading to building a high-performance system on chip ICs. As they became smaller, faster, and cheaper, the demand for portable battery-powered systems increased with the condition of keeping the cost down. Optimization techniques aim to improve speed, performance, and power consumption, especially for these portable devices and their subsystems. Mobile phones and hand-held multimedia systems have been the dominant driving force behind these developments and related markets.

Today, almost all mobile phones, tablets, and multimedia devices have a built-in camera, which plays an essential role in overall device power consumption, performance, and cost. Low-cost image sensors consuming low-power with highly integrated functionality are needed for these portable devices. So, minimizing power consumption while keeping acceptable imager performance is one of the fundamental challenge image sensors, and camera sub-systems face since the beginning.

Portable devices put a limit on the physical size and weight of the power source. Therefore, limited battery capacity mandates low-power designs, not only image sensor level but also on the system level. A typical CCD image sensor dissipating 3 watts (W) of power will run less than an hour on a 1.2V battery with 2000mAh capacity, while a low-power CIS consuming 30μW will run nine years on the same battery.

CIS consists of different blocks, including a 2-dimensional (2D) array of photosensitive elements (pixels), pixel amplifiers, analog signal processors (ASP), and ADCs. Typically, the performance of the integrated ADC determines the overall performance of a CIS. The ADC conversion speed should accommodate the camera system's target frame rate while consuming minimum power. ADC is the most power-consuming block on the analog signal chain (ASC) of the CIS. Therefore, performance optimization and smart tradeoffs among image quality, speed, and

power consumption of integrated ADCs in CIS need to be considered carefully to meet the system, performance, and market requirements.

Many speed-up techniques (SuPTs) have been developed to enhance the analog-to-digital conversion speed of integrated ADCs in CIS. However, as the high-performance CIS demand becomes significant, the speed-up ratio (SuPR) of the existing SuPTs should be improved either by developing new SuPTs or enhancing the existing ones, or even hybridizing some of them to obtain better SuPR and, consequently, better performance for the CIS.

Speed-up, most of the times, is achieved at the expense of the image quality, power consumption, size, and/or design complexity. Except for image quality, power, size, and design could easily be evaluated objectively. However, no objective image quality metric exists to allow a fair comparison and assessment between all existing speed-up techniques. This metric has to be reference-free (an independent metric that does not require reference image to compare), bounded to have maximum and minimum index values (i.e., between 0 and 1), sensitive to changes on processed images, and have a monotonic response. This metric is necessary for the intelligent adaptation of image sensor electronics, setting up some parameters to optimize ADC speed (as a result, power) while trading image quality after assessing image conversion complexity before the conversion occurs.

There is a competition between all available CIS SuPTs, and each is trying to prove its superiority over the other. To prove this, they may use a simple image that would require a less complex conversion process. This would blur the real advantage or disadvantage of the competing techniques. Thus it is necessary to use objective and fair comparison methodology to avoid this issue. This methodology should consider the conversion complexity of the input image, quality of the reproduced output image, and the common controlling parameters (CPs) of the SuPT to set up a rule set that every competitor follows to prove their performance SuPT.

Motivation and Goals

This research's primary focus is to develop a high-speed CIS that adopts integrating (ramp) type ADC to take advantage of its design simplicity, low-power consumption, and small silicon footprint and mitigate its main drawback of slow conversion speed. Besides, as outlined in the previous section, a new reference-free image quality metric is developed to adapt CIS electronics intelligently for changing scene and imaging conditions, optimizing speed, power, and image quality. Finally, since there are many SuPT for ramp ADC integrated with CIS CPA, it is challenging to compare them as each SuPT uses different types of images, and it has many CPs that can be adjusted to prove that their performance is the best. A comparison methodology is required to set up a rule of thumb for the SuPTs to allow objective comparison of their performance.

Hence, this research has three main goals. The first goal is to design a new speedup technique for ramp ADC used in CIS to allow faster conversion speed and minimize power consumption and maintain a high quality of processed image using this technique. The second goal includes developing an objective reference-free image quality metric to assess the processed images' conversion complexity by different ADC SuPTs to use it for intelligent adjustment of the circuit parameters. The final goal is to produce a comparison methodology to compare the speed and power performance of different SuPTs, including the proposed new technique.

Applications

The smartphone market is growing and has a tremendous driving force behind global business developments. This industry worth billions of dollars and increasing every year. Vendors are trying to improve their product by offering high-performance smartphones with the most extended battery life and high-quality sub-systems such as microphones, speakers, projection capable modules, and multiple camera modules. Among them, camera modules' capabilities are among the most important features that customers are always searching for. Also, camera performance affects the overall performance and battery life of smartphones. When the camera consumes more power, battery life will be shorter than competitor products. In addition, the camera may consume less power but produces low image quality or slow response for high speed moving objects. This research targets improvement of the camera sensor and its electronics such that it produces high-quality and responsive images with low power consumption.

Organization of the thesis

Chapter 1 introduces the research motivation, goals, and applications. It also summarizes the research contributions and the organization of this thesis.

Chapter 2 gives a general background on CMOS active pixel sensor (APS) imagers. First, general information on CMOS image sensors (CIS) is provided. Second, a brief historical background on CMOS image sensors is presented, followed by a comparison of CMOS APS and CCD technologies and trends. Third, CMOS image sensor architectures, functional sub-blocks, and design requirements are discussed.

Chapter 3 explains a new speed-up technique used to enhance column ramp ADC's CIS speed to improve the overall frame rate. It starts with an illustration of a quick background about current speed-up techniques. Then, it shows the operation of the new speedup technique, the single-slope look-ahead ramp (SSLAR) ADC. Measurement results are also presented in this chapter.

Chapter 4 discusses the new image quality (IQ) metric, called conversion complexity metric (CCM), used for comparing different speedup techniques to offer a reference for fair assessment

between processed images by any speedup techniques. This chapter started with a background of image quality metrics and how they can be categorized from different perspectives. Then, it shows how this image quality metric can be calculated with a numerical example and equations. After that, the discussion of how to find and calculate and test its extreme limits is illustrated. Finally, testing the new image quality metric is shown on different standard and commercial databases with different image resolutions and sizes to generalize it.

Chapter 5 illustrates the development of the accelerated single-slope look-ahead ramp (ASSLAR) ADC SuPT and its design concept, control parameters (CPs), and optimization of CPs to allow maximum performance with minimum IQ degradation. After that, the new comparison methodology is discussed in detail and applied to a case study to compare the ASSALR and AR SuPTs. This chapter concludes with an important conclusion of the hybridization of powerful techniques based on the input image CCM index.

Chapter 6 summarizes the significant accomplishments and contributions achieved in this research and presents ideas for future research.

Chapter 2: CMOS Image Sensors (CIS)

CMOS images sensors (CISs) are widely used today in almost every camera system. The camera system is built by integrating three main blocks; optics, image sensors, and digital signal processors. This thesis is concerned with the image sensor block. This investigation starts with a historical and technological literature survey. It includes the charge-coupled device (CCD) history, complementary metal-oxide-semiconductor (CMOS) image sensor (CIS) basics and sub-system components, and a study of the latest developments for improving the performance of CIS.

Introduction

The image sensor is the next block of the imager system that comes after the optics block. It translates collected photons by imager optics to an electrical signal. The solid-state image sensor consists of several blocks that handle signals through different stages until delivered to the digital signal processing (DSP) block. The solid-state image sensor signal chain's front end includes micro-lens that direct and focus incident photons to the image sensor's photosensitive area to increase quantum efficiency. For color images, color filters are added between micro-lens and photosensitive areas to filter the incident light (photons) according to their wavelengths to separate different colors' intensities. The photosensitive material absorbs photons and allows free carrier (electron or hole) generation resulting in ideally one electron-hole pair per absorbed photon with higher released energy than the bandgap of the substrate semiconductor. The electron-hole pair is separated and converted into an electrical signal. The electrical signal is buffered, locally amplified, sampled, and held before it goes through an analog to digital converter whose output is passed to DSP; the last block of an image sensor system that regenerates the scene image digital form.

A brief technological and historical background on CMOS image sensors emphasizing active pixel sensor (APS) technologies is presented in this chapter. First, a brief history of CCD is reviewed. Second, the basics of CMOS image sensors is explained. Third, current state-of-the-art speed-up techniques for ramp ADCs used in CIS are presented. Finally, a summary of the chapter is given.

Charge-Coupled Device (CCD)

The charge-coupled device (CCD) was the technology of choice for a very long time since it was invented in 1969 at Bell Labs by Boyle and Smith [7]. CCDs are highly sensitive photon detectors that consist of an array of a large number of pixels, which are the photosensitive part of CCDs. When photons incident on pixels, it converts photons to one or more electrons, which are proportional to the number of incident photons representing the current scene of an image. Hence, each pixel holds an amount of charge that represents the number of collected photons. These charges

need to move out of pixels to allow pixels to detect a new image frame. CCD is clocked out such that charges in pixels are transferred from one pixel to adjacent pixel until all charges are transferred to the output stage. Originally, CCDs are an analog shift register used to transfer charges from the image sensor to output stages. An increased number of transfers require CCDs to utilize a manufacturing process that allows perfect charge transfer. CCD evolved and flourished due to its high sensitivity, high quantum efficiency, and large format. Its high sensitivity results from its high quantum efficiency on the order of 40%, low noise output amplifier, and high fidelity of readout [8]. Modern CCDs had a high fill factor of 80% - 100%. Many research efforts have been dedicated to improving the CCD performance because it was very promising at that time of invention. As a result, CCDs accomplished low readout noise, excellent photo-responsivity, high dynamic range, minimum pixel dark current, and high quantum efficiency.

CCD requires near-perfect charge transfer from pixel to pixel. This results in, however, difficulty in reproducibility in large CCD arrays. They cannot be integrated with on-chip electronics, spectral responsivity cannot be extended through using different materials, and the readout rate is limited. The charge transfer operation needs a high voltage supply, which is not suitable for standard low power and low voltage applications. Although CCD has many advantages like high dynamic range and high responsivity, mentioned issues prevent the integration of on-chip supplementary circuits like timing generators, analog to digital converters (ADCs), clock drivers, and signal processing. These circuits are usually built off-chip, resulting in high power consumption and consuming significant supply current used to drive output pad capacitors.

Due to CCDs' disadvantages, CMOS APS replaced CCD as APS preserve CCD's high performance and eliminate the need for perfect charge transfer. Fossum in [8] proposed that APS will replace CCD and become the next successor. In the next Section, the CMOS APS and its main components are explored.

CMOS Image Sensor (CIS)

A brief history of complementary metal-oxide-semiconductor (CMOS) image sensor (CIS) is reviewed in this section. Then, CMOS pixels and its different types and their main ideas are explored. Finally, the rest of the CIS sub-systems are illustrated.

CMOS History in a Glance

Historically, metal oxide semiconductor (MOS) image sensors were invented before CCDs and CMOS APS around the 1960s. At that time, some research groups worldwide were able to build the first solid-state image sensor. Weckler in 1967 [9] described a technique for operating a

photodiode (PD) p-n junction in the integration mode of photon flux and then charging the p-n junction in reverse bias voltage and discharge it in two different illumination conditions; zero-incident illumination and under illumination. He concluded that the discharging rate is independent of the p-n junction area in case of no illumination. In contrast, in the case of illumination, the charge's decay rate depends on light intensity. This means that the MOS structure can easily be integrated for image sensor arrays. Besides, sensitivity can be controlled electronically. In 1968, a PD array of 100 X 100 was reported by Dyck and Weckler [10], and it was working in photon flux integration mode. In the same year, Noble [11] proposed some configurations of self-scanned silicon image detector arrays and their circuitries to construct 2D images. In addition, he discussed how to reduce dark current by burying a PD below the semiconductor surface, as well as using regular unburied PDs. He used MOS source follower in the pixel for the first time to buffer the readout signal. Early MOS devices were suffering from fixed pattern noise (FPN) because of variation between pixels. FPN was explored by Fry et al. in 1970 [12]. In the same year, 1970, CCD was first introduced [7] with low FPN and small pixel size, and that is why it was adopted for decades earlier until the rebirth of CIS in the 1990s.

CIS had been adopted again to solve CCDs‘ problems. Many types of research have been focused on CMOS image sensors because it was very promising and advancements in fabrication processes at the time helped for its development and improvement rapidly. It was developed to overcome CCDs‘ main problems and, most importantly, commercialize it in a cost-efficient manner to the masses. The CIS's ability to integrate on-chip supplementary circuits eliminates many issues that CCDs have. Integration minimizes interface problems and minimizes driving current consumed by output pads and hence minimizes overall power consumption. In addition, low voltage supply can be used in CIS as now there are no capacitors exist to transfer pixel charges. CIS allows low noise performance, no lag, no smear, better blooming control, simple clocking, and monolithic integration. Finally, pixels can be accessed randomly since the output of pixels do not need to shift sequentially.

CMOS Image Sensors System

Generally, CIS consists of pixels array, row and column decoders, analog signal processor(s), and analog to digital converter(s), as shown in Figure 2.1. The pixel array is the photon sensitive part of CIS. When photons of light incident on the pixel array, it generates electron-hole pairs as long as

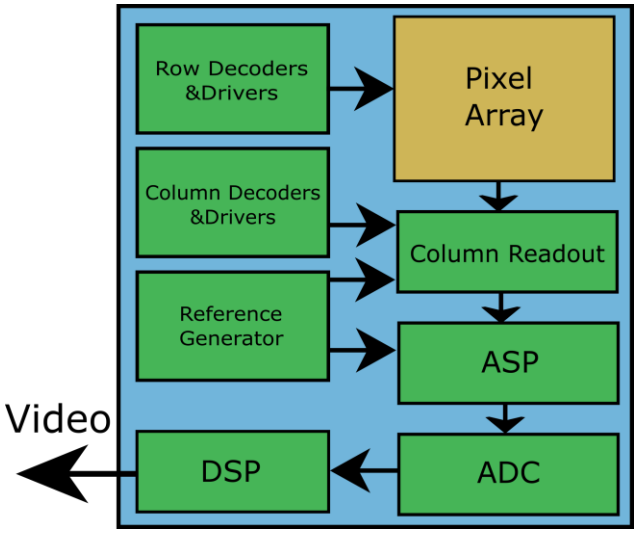


Figure 2.1. Functional blocks in a CMOS image sensor (CIS).

photon energy is greater than the semiconductor material's energy gap. Generated electron-hole pairs pass through a chain of several steps before an image is captured. Figure 2.2 illustrates the journey of photons through CIS until it becomes a video streaming. Negatively charged electrons and positively charged holes are willing to recombine unless they are separated, i.e., by using an electric field. These charges can be stored in a capacitor and converted to a voltage, or they could be quantified as current according to pixel structure. The readout stage comes after the conversion of charge to voltage or current. Each pixel signal is read out according to CIS structure (pixel by pixel or row by row, etc.). The analog signal is buffered and processed by ASP(s) then converted to a digital signal by ADC(s). Finally, the digital signal is processed through digital signal processing (DSP) to an image that is displayed on the screen.

Architectures of CMOS Image Sensors

CIS components can be arranged in many ways to fulfill specific requirements. There are four main types of CIS architectures; (1) column-parallel architecture (CPA), (2) column-series

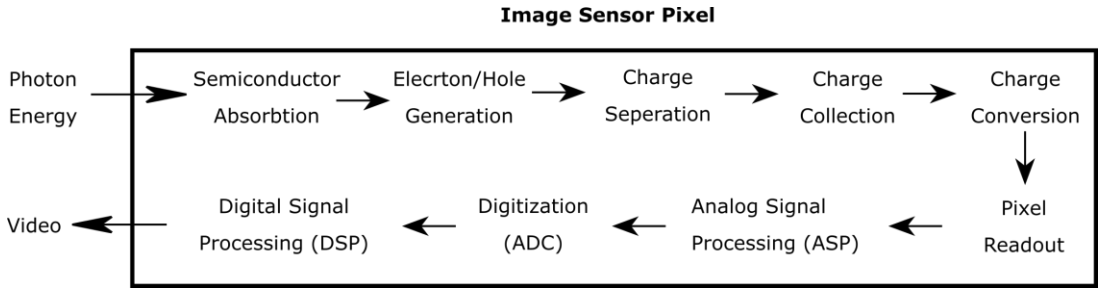


Figure 2.2. Photon's path through CIS

architecture (CSA), (3) pixel-parallel architecture (PPA), (4) pixel-series architecture (PSA). These four types will be illustrated in the following sub-sections.

Column-Parallel Architecture (CPA)

Figure 2.3 shows CIS with CPA. It has a single ASP and single ADC per each column. Signals from rows are sampled on column ASP(s) and digitized by column ADC(s) in parallel and readout from columns in the digital domain. As only one ASP and ADC process whole column data in parallel, ADC should have high conversion speed. CPA has a relatively low fill factor as it has a number of ASPs and ADCs equal to the number of columns, which consumes extra area. Some other CPAs may split ASP and ADC into more than a single ASP and ADC per column to relax the speed limit but at the fill factor expense.

Column-Series Architecture (CSA)

CSA consists of a single ADC per chip and single ASP per column, as shown in Figure 2.4. Row(s) of pixels are sampled on column (or global) ASP(s) and digitized by global ADC(s) in series. ADC of CSA should have a very high conversion speed as it converts all columns in series. CSA has

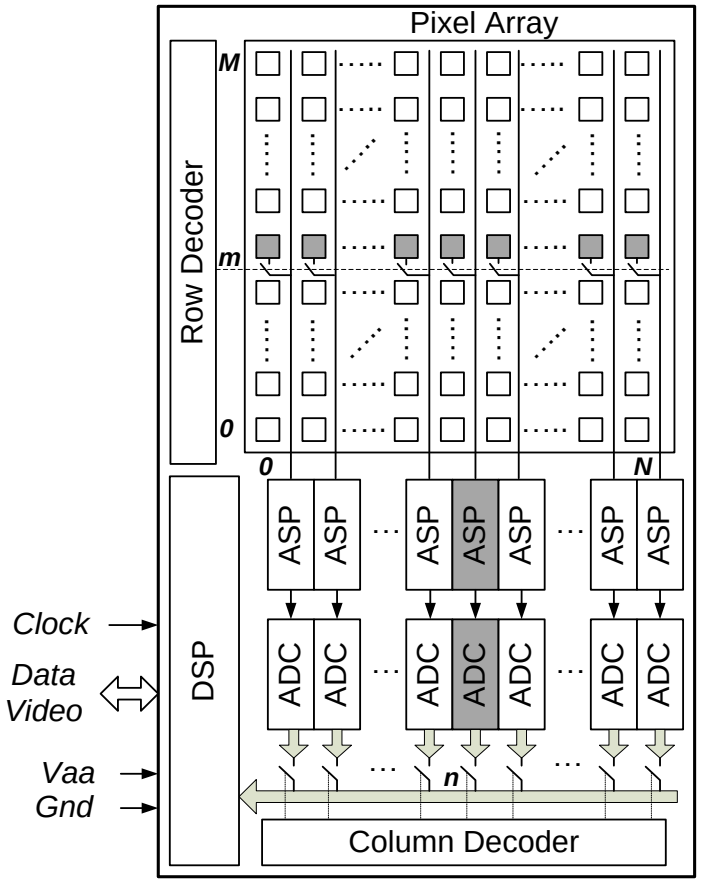


Figure 2.3. Column-parallel architecture (CPA).

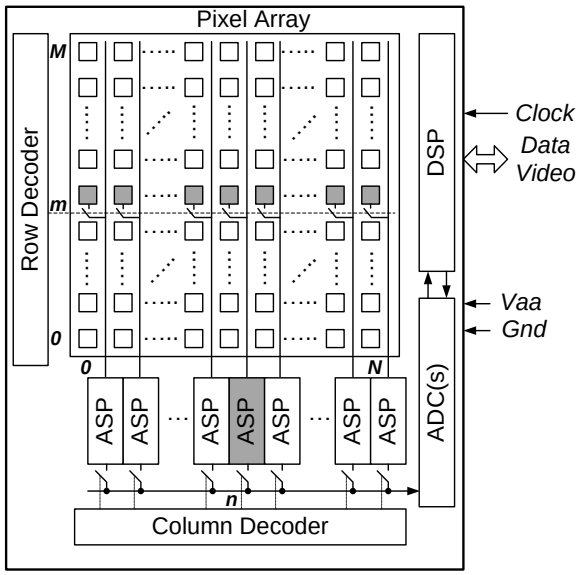


Figure 2.4. Column-series architecture (CSA)

a better fill factor than CPA because of the single ADC per chip saves more area for sensing elements.

Pixel-Series Architecture (PSA)

Figure 2.5 shows PSA; it has only a single ASP and ADC per chip. Individual pixel is selected in series and read through global ASP and global ADC. This architecture has the best fill factor because most of the area is dedicated to sensing elements. However, ADC should fulfill the extremely high-speed requirement to maintain a reasonable frame rate.

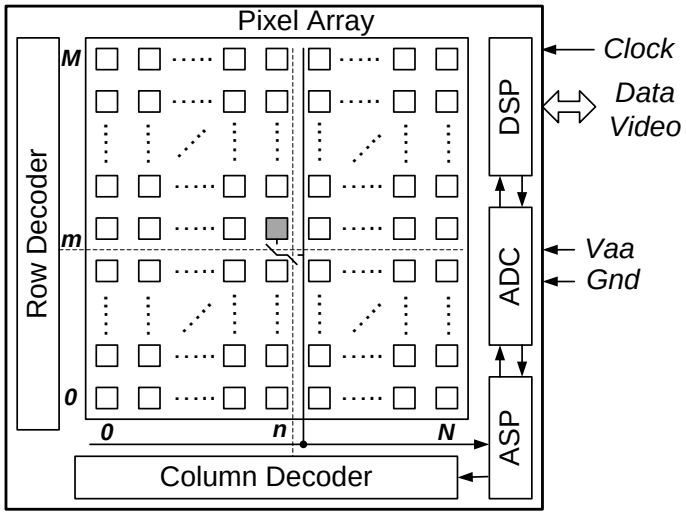


Figure 2.5. Pixel-series architecture (PSA)

Pixel-Parallel Architecture (PPA)

PPA is depicted in Figure 2.6. each pixel has its ASP and ADC. Individual pixel is sampled in parallel and read through pixel ASP(s) and pixel ADC(s). this architecture has the worst fill factor while it releases the speed limit requirement for ADC.

CMOS Pixel Technologies

CIS pixels have three basic approaches; PD-type passive pixel, PD-type active pixel, and photogate-type active pixel [6]. The main difference between active and passive pixels is that active pixels have an amplifier at each pixel for buffering and driving the output signal to busses. In contrast, passive pixel connects the PD output directly to the output bus.

Passive CMOS Pixels

The PD passive CMOS pixel consists of a PD and access switch. Figure 2.7 illustrates the basic structure of a CMOS passive pixel and column charge amplifier. Passive CMOS pixel was proposed by Weckler in 1967 [9], [10]. The PD is connected to the column bus when the access switch is turned on, and the voltage of the column bus is kept constant thanks to the column charge amplifier, which helps to reduce thermal noise. The PD voltage is reset to column bus voltage when PD is accessed; then, the PD charge is converted to a voltage by charge column amplifier. The simplicity of passive CMOS pixel results in a very high fill factor, allowing very small pixel size. Passive CMOS pixel has only one transistor. However, another transistor could be added to allow X

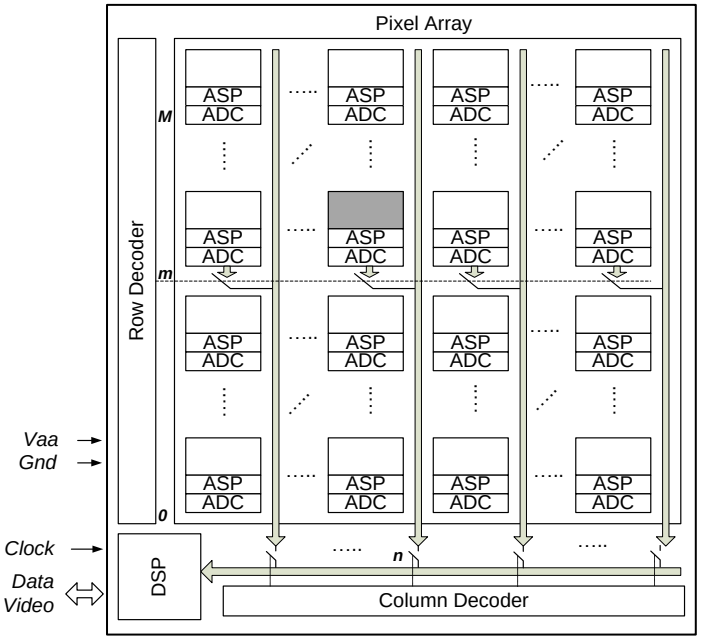


Figure 2.6. Pixel-Parallel Architecture

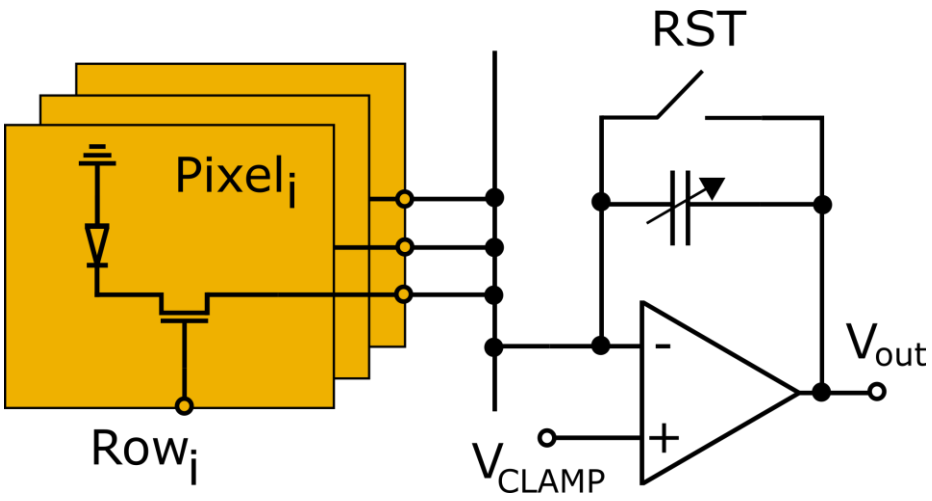


Figure 2.7. Passive CMOS pixel schematic and column amplifier.

and Y addresses. Because of the large fill factor of passive CMOS pixel, quantum efficiency could be relatively high.

Passive CMOS pixels suffer from several issues like readout noise, FPN, and scalability. The readout noise of passive CMOS pixel is in the range of hundred(s) of electrons r.m.s. But for CCDs, it is less than a few electrons r.m.s. Also, the column amplifier causes a large FPN. The passive CMOS pixel cannot be scaled easily to a large array format or faster readout rate, and this is due to the large column bus capacitance the result from connecting more pixels to it. Large capacitance results in higher power consumption and higher readout noise.

CMOS Active Pixel Sensor (APS) Technologies

Active pixel sensors (APS) are proposed [11] [13] [14] [15] to solve issues with passive CMOS pixel. Each APS has, typically, a source follower amplifier to buffer the photon detector signal. The pixel source follower is turned on only during the readout of the pixel. So, active pixels have relatively low power consumption. There are two main types of active pixel sensors; photodiode (PD) type CMOS APS and photogate (PG) type CMOS APS

Photodiode (PD) Type CMOS APS

Figure 2.8 illustrates a schematic diagram of a typical photodiode (PD) type APS that uses three transistors (3T). It consists of reset transistor (M1), pixel source follower (M2), and row select transistor (M3). It has high quantum efficiency due to there is no polysilicon overlap.

At the starting of integration time, M1 is switched on to set the reset voltage (V_R) of PD. The depletion region capacitance holds the reset voltage of PD, which increases the negative built-in electric field from N to P region. M1 is switched off during integration time, causing PD to float

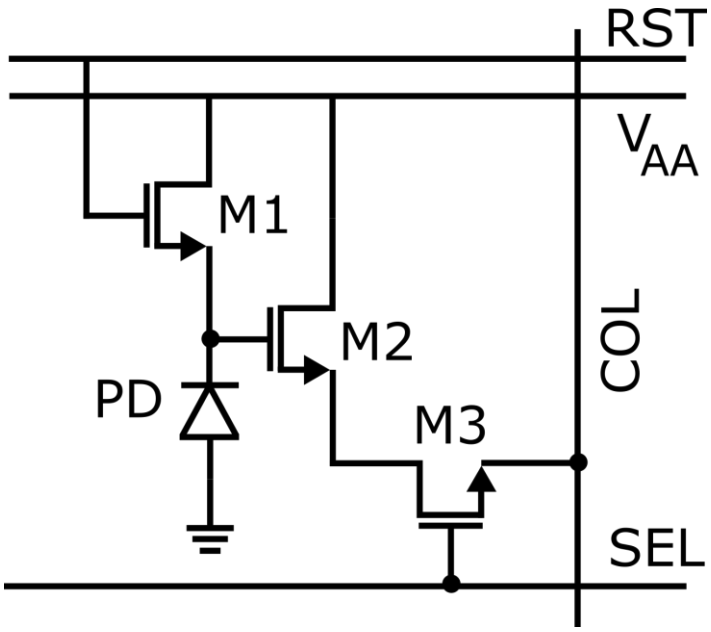


Figure 2.8. schematic diagram of a typical photodiode

while exposing to light. Hence, excess electron-hole pairs are photo-generated, and the electric field drives electrons to P-region and holes to N-regions (the charge separation function). These drifted charges start to discharge the PD capacitance during integration time with a rate proportional to the amount of charge generated by light, corresponding to light intensity level. M2 is dedicated to the buffering function and to drive the capacitive column. It isolates PD from readout electronics and enables the non-destructive readout process. M3 is a select transistor that connects the selected row to the column bus. This basic operation is followed by all CMOS APS, even though it may differ in the number of transistors or schematic to enhance performance.

Figure 2.9 shows the cross-section of APS PD. For standard CMOS process. Extended field oxide (FOX) is formed around n+ regions causing dark current due stress centers [16]. In addition, there are two other sources of dark current, which are surface defects and surface recombination centers. These two dark current sources absorb any photo-generated charges near to surface like short-wavelength photons corresponding to the blue/ultraviolet (UV) spectrum. The structure shown in Figure 2.9 has poor blue/UV response due to the high dark current near-surface.

The solution of the blue/UV response problem related to surface defects is depicted in Figure 2.10. PD is buried under a thin surface p+ region. This structure is called a buried photodiode or pinned photodiode (PPD), and it needs a special process for manufacturing. This thin layer doping concentration should be well controlled to form potential well of PD correctly [17] to allow complete charge transfer from PD to floating diffusion (FD) through the transfer gate (TX). FD and TX are added for low noise readout operation. This structure is working as follows. First, FD is reset to reset

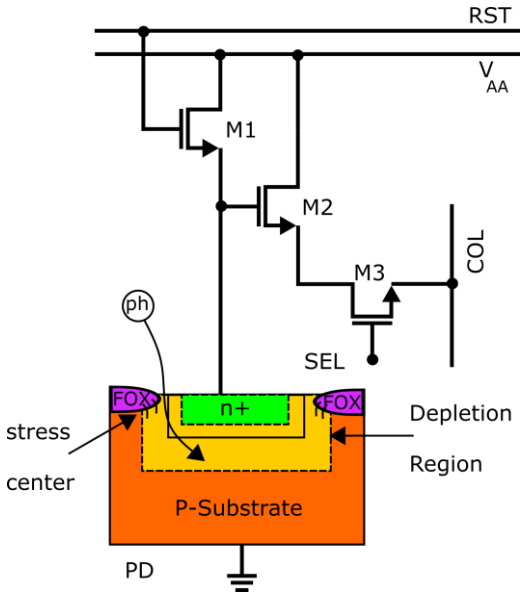


Figure 2.9. Cross sectional view of photodiode in 3T CMOS APS Pixel

voltage, and this level is read out at the starting of integration time. Then, TX is switched on to transfer charges from PD to FD. After that, FD is readout again after integration time, and this level represents a signal level. The difference between these readout values after and before transferring charges is the absolute signal, proportional to the light level. This readout operation is called correlated double sampling (CDS). CDS suppress 1/f noise from the in-pixel source follower, thermal noise caused by pixel reset, and fixed pattern noise due to pixel-to-pixel variation in pixel transistors.

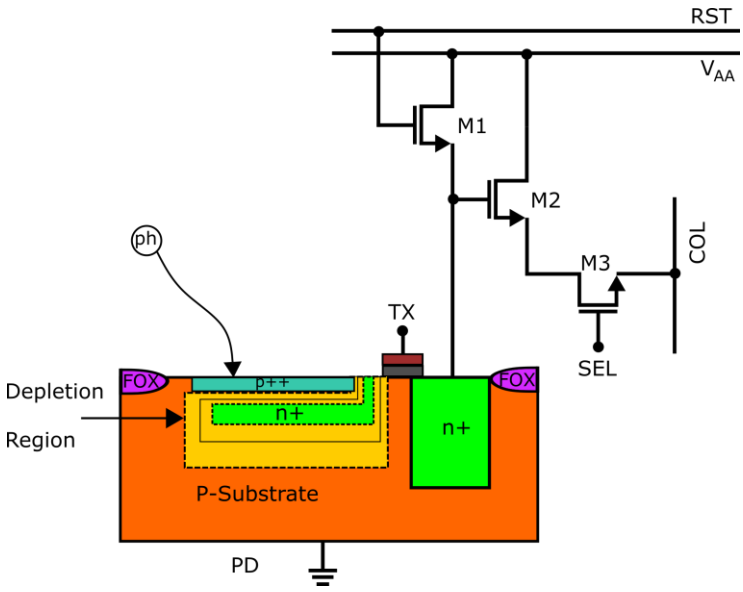


Figure 2.10. Cross sectional view of a buried (pinned) photodiode (PPD) in CMOS APS pixel

Photogate (PG) Type CMOS APS

Photogate (PG) type active pixel sensor was first introduced in 1993 at JPL [13] [14] [15]. Figure 2.11 illustrates a cross-section of the PG type active pixel sensor. It consists of PG, transfer gate (TX), and floating diffusion in addition to the 3T like PD. Low noise charge collection of CCDs technology was inspiring for proposing a PG type active pixel sensor. In Figure 2.11, if a positive voltage is applied to the PG terminal, the depletion region is formed as P-substrate's holes are repelled away from the surface. The depletion region collects photo-generated electrons due to incident photons, while holes flow to the ground because of the applied electric field across the depletion region. The surface of silicon collects the generated charges like in PD. Still, for PG, this surface has better quality than in PD, which results in low surface dark current and no stress-related dark current in PG. Therefore, the PG dark current is shallow, and surface effects can be reduced using buried PG.

In Figure 2.11, the RST transistor sets FD to a reset level, which is read first. Then, TX is pulsed, allowing charge under PG to transfer to FD, and the FD level is reread after transferring is completed. The readout signal difference after and before charge transfer during the integration period is the readout signal, and this is the correlated double sampling process. The PG type active pixel sensor is typically larger than PD because of the extra components

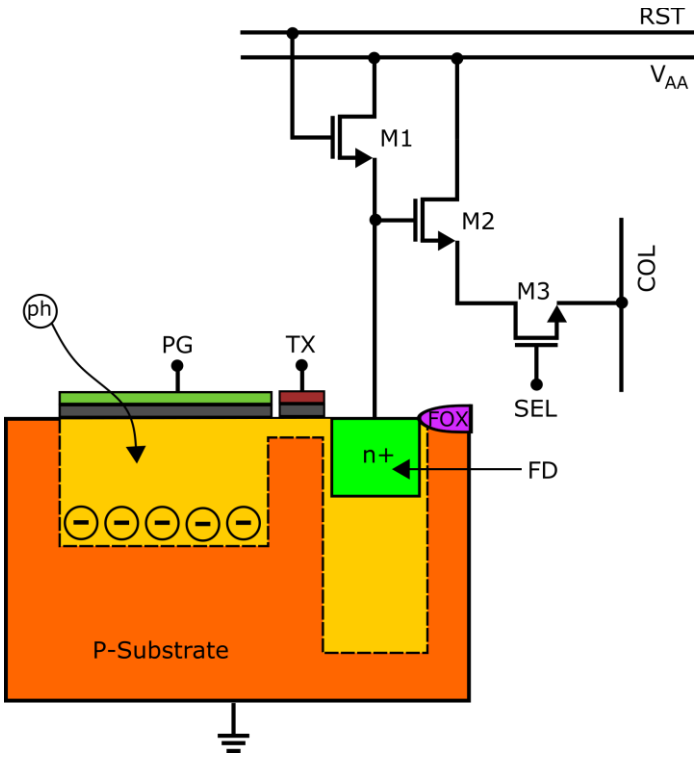


Figure 2.11. Cross sectional view of a photogate (PG) type CMOS APS pixel.

Analog Signal Processor (APS)

For the passive pixel sensor, a charge integration amplifier is used for the readout signal, while for the active pixel sensor, sample and hold circuits are used. The readout process is completed when the analog signal is stored in sample and hold capacitors. The programmable gain amplifier typically follows the sample and hold circuit. ASP performs correlated double sampling (CDS) to suppress FPN, reset noise, and 1/f noise. ASP can also perform some other functions like level shifting, offset cancellation, delta-double sampling, etc. The analog output signal from ASP is fed to the analog-to-digital converter (ADC) for converting to a digital signal.

Analog-to-Digital Converter (ADC)

Depending on imager architecture, the analog signal can be converted to digital form at different levels. It can be converted in pixel, column, or global level. However, on-chip A-to-D conversion is recommended because digital signals have better noise immunity than analog signals; also, the system's total cost will decrease as a result of more components integrated with one chip. In addition, the integration of ADC(s) minimizes power consumption of CIS as there is no need for the powerful amplifier that will drive analog output pads or off-chip capacitive loads. However, ADC(s) should be optimized for minimum power consumption minimum silicon footprint for CMOS imager's better performance. At the same time, ADC(s) must have the capability of supporting video rates with at least 8-bit resolution and high frame rate. Integral nonlinearity (INL) and differential nonlinearity (DNL) should be minimized to reduce image quality degradation. To avoid the switching noise coupling of ADC(s) to the pixel array or analog components, ADC(s) must be well isolated from the pixel array and analog components.

There are many ways to implement ADC(s) in CMOS imagers. It can be a single ADC with very high speed for the whole imager running in a serial fashion or multiple ADCs with lower speed running in parallel; this depends on imager architecture and how different blocks are arranged. As the number of parallel running ADCs increased, the speed limit for them can be relaxed, and the power consumption limit. ADC speed limit, size, and power consumption are the key points for choosing ADC type for CMOS imagers; for example, successive approximation register (SAR) ADC has a higher speed and less power consumption than ramp ADC; however, it needs a large area that is not fit into CMOS imager. Table 2.1 shows a comparison of different ADC types that are available for CMOS image sensors. Flash ADC is the fastest type at all and provides high bandwidth. However, it suffers from the most important needed features; high power consumption and large die size in addition to high cost. So, flash ADC is not suitable for CMOS image sensors. Sigma- Delta ADC provides high

Table 2.1. Comparison of ADC topologies for CMOS image sensors

ADC Type	Resolution	Power	Energy/sample	Clock cycle	Speed	Pros/Cons
Flash	8-bit	>50mW	~50nJ/S	1	10Msps - 1Gsps	+Extremely Fast +High Input Bandwidth -Highest Power Consumption -Large Die Area -Expensive
SAR	8-16 bit	~100µW	~100pJ/S	N	75Ksps - 2Msps	+High Resolution And Accuracy +Low Power Consumption +Few External Components -Low Input Bandwidth -Limit Sampling Rate -Input Voltage Must Remain Constant During Conversion
Ramp	>14 bit	~1mW	~10nJ/S	2^N	<100Ksps	+High Resolution +Low Supply Current +Excellent Noise Rejection -Low Speed
Sigma-Delta	>14 bit	>10mW	~5nJ/S	$<2^N$	>200Ksps	+High Resolution + High Input Bandwidth +Digital On-Chip Filtering -External T/H -Limited Sampling Rate
Pipeline	10-14 bit	>10mW	10nJ/S	N+1	10Msps - 100Msps	+High throughput rate +digital error correction and on-chip self-calibration -typically requires 50% duty cycle -requires minimum clock frequency

resolution and high bandwidth, but it needs an external track and hold circuit and has a limited sampling rate.

Additionally, Sigma- delta ADC typically consumes more than10 mW of power, which means that it will increase total chip power consumption if used for each CMOS image sensor columns. This is the same reason why pipeline ADC is not used for CMOS image sensors, although it has a high throughput rate and digital error correction and on-chip self-calibration. Ramp or integrating ADCs fulfill the most requirements of CMOS image sensors. It has high resolution and low power consumption in addition to excellent noise suppression and a small silicon footprint due to its simplicity; however, it suffers from low conversion speed. It needs 2^N clocks to convert an analog signal. Speed enhancement of ramp ADCs becomes a hot topic of research that will be reviewed in the next section.

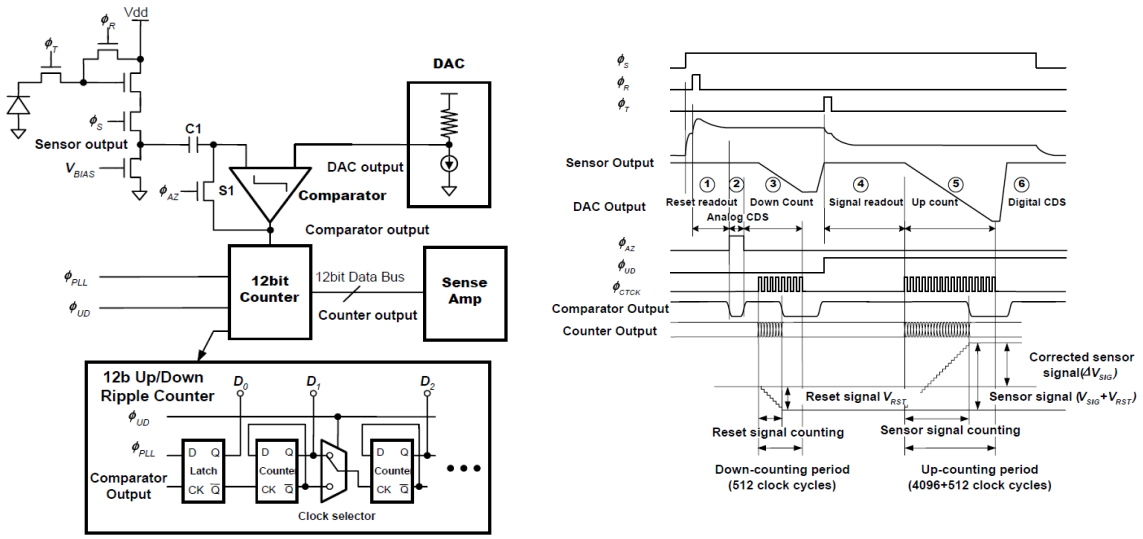
Speed-Up Techniques for CIS Integrating Type ADCs

The demand for high-frame-rate CMOS image sensors is steadily increasing. Column-parallel integrating (single-slope ramp, SSR) type ADCs were widely used in CMOS image sensors because they can be implemented with a small area, low noise, and high energy efficiency. However, conventional SSR ADCs have a speed limitation, which has led to various architectural improvements. In the following sub-sections, different methods of improving these limitations are presented. These improvements can be divided into three main categories; single-step sampling, multi-step sampling, and hybrid single-slope ramp ADCs.

Single-Step Sampling Ramp ADCs

Clock Frequency Increase for Single-Slope Single-Ramp ADC

In 2006, a research group in Japan [18] [19] [20] proposed CIS that has a single master clock operating at 74.25MHz and an ADC clock of 297MHz generated by an on-chip phase-locked loop (PLL) from the master clock. ADC clock was used for generating single-slope ramp signal using a high-speed digital-to-analog converter (DAC) driving column SSR ADCs and was also used for low voltage differential signal (LVDS) interface, as shown in Figure 2.12 [18]. This high-speed clock (297MHz) is also used to minimize the double digital sampling period of digital double-sampling (digital CDS) and analog CDS. Also, a high-speed clock and CDS help to achieve 150MHz of data rate with 12-bit resolution. Frame rate can increased up to 180 frames/second (fps) with maximum 600 MHz data rate but in the expense of resolution that drops to 10b.This design achieved readout



(a) (b)

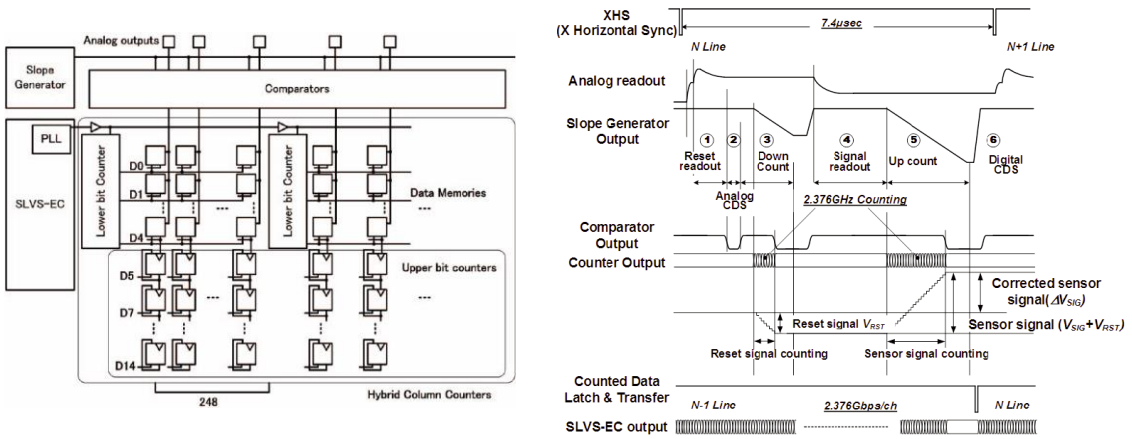
Figure 2.12. (a) Column-inline dual CDS architecture as presented in [18] that uses shigh speed SSR and its (b) timing diagram.

random noise of 4.8 e^-_{rms} for 60 fps and 12-bit resolution and this readout noise increases to 5.2 e^-_{rms} if the frame rate increased to maximum of 180 fps as the noise increased for voltage supply.

In 2011, Toyama et al. [21] presented high image quality 34.8 Gb/s CIS, which realized 17.7 M pixels with 12-bit resolution at 120 fps and 75 dB dynamic range, as shown in Figure 2.13. They have two solutions for the speed issue of conventional single slop ADC; the first one utilizes hybrid column counters that allow rapid A/D conversion with minimum power consumption. The second solution is a Scalable Low Voltage Signaling interface with Embedded Clock (SLVS-EC), which results in 2.376 Gb/s/channel.

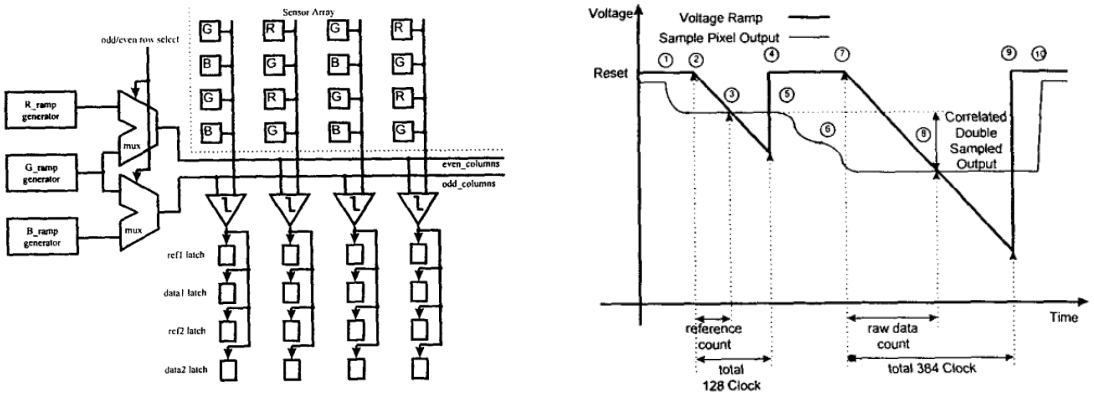
Dual-Slope Single-Ramp ADCs

As shown in Figure 2.14, the double-slop technique is proposed by Oh-Bong et al. [22] [23] to enhance the resolution of images with low luminance conditions and improve dynamic range. Three analog ramp generators with programmable step size multiplexed into odd and even sub banks of comparators for individual ADC gain control for red, green, and blue pixels arranged in Bayer pattern, as illustrated in Figure 2.14. (a). This technique utilizes gamma characteristics to give higher resolution and better dynamic range. It differentiates between high and low illumination levels of the pixel. The high illumination pixel output is limited by shot noise, and low-resolution conversion is sufficient, while in low light condition, pixel noise is dominated by the quantization noise of ADC and needs high-resolution conversion for better image quality. Hence, a double slope exploits this characteristic and implemented such that its slop is shallow in case of low pixel output values to



(a) (b)

Figure 2.13. (a) Block diagram of this SS-ADC architecture as presented in [21] that uses high speed SSR and its (b) timing diagram



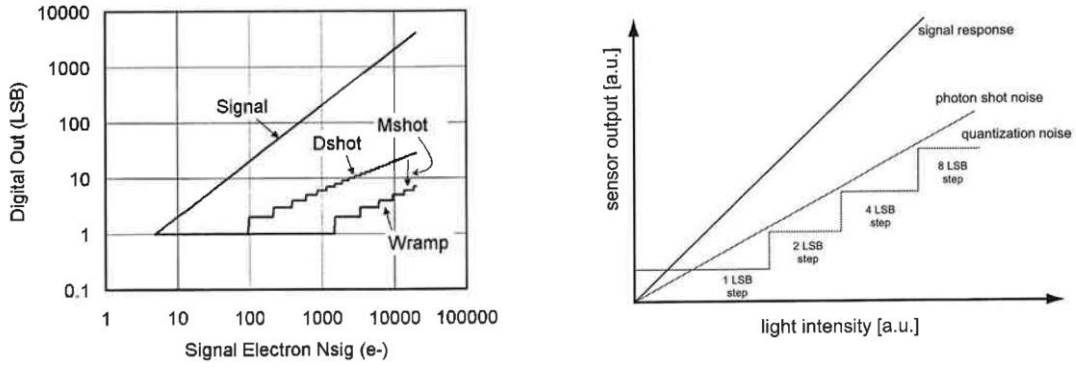
(a) (b)

Figure 2.14. (a) Block diagram of dual slope ADC as presented in [22] and its (b) conversion cycle

provide high resolution. On the other hand, its slope is steeper when the signal level is strong enough with low resolution to provide a high dynamic range.

Accelerated Ramp (AR) ADC

In 2005, Otaka et al. [24] and Snoeij et al. [25] proposed enhancement of ramp ADC conversion speed using accelerated ramp based on the photon shot noise limit of an image signal as depicted in Figure 2.15. (a). The photon shot noise dominates when the signal level increase. For large input signals, if ADC steps are increased linearly, its performance is higher than needed, ie. ADC quantization steps can be increased, as shown in Figure 2.15. (b) without affecting the overall signal to noise ratio. Otaka defined an image signal's shot noise as the square root of this signal and related ADC ramp steps to photon shot noise rather than linear steps. Accelerated ramp step size should increase to decrease the overall number of required steps and perform conversion faster. Otaka also introduced another parameter called shot noise margin and defined it as the accelerated ramp step



(a) (b)

Figure 2.15. (a) shot noise limit as in [24] (b) Increased ADC quantization steps as in [25]

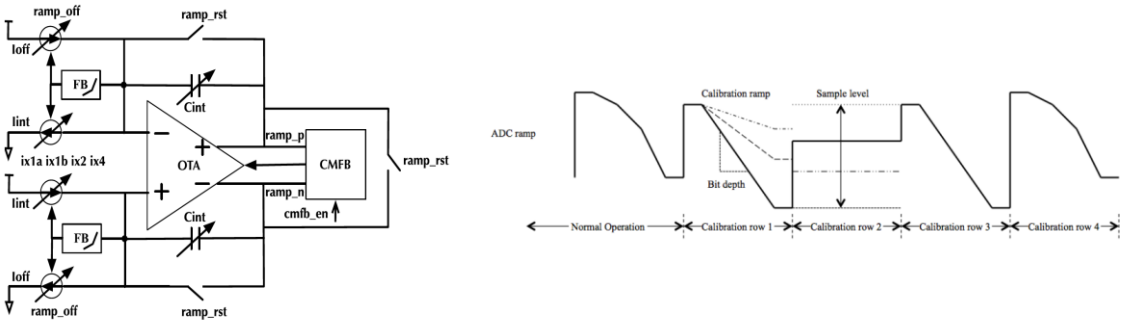
size ratio to signal shot noise. The recommended shot noise margin is 1/2 to reduce the possible appearance of contours effectively.

In 2013, a differential ramp generator for SSR ADC with an accelerated ramp was proposed [26], as illustrated in Figure 2.16 (a). The main idea is built in the same theory of photon shot noise limitation and implement an on-chip continuous-time ramp generator to replace discrete-time implementation to minimize glitch noise resulted from discrete-time ramp high-speed clocking. Figure 2.16 (b) illustrates a real-time calibration sampling and timing scheme for a 14-bit accelerated ramp generator for column-parallel CIS introduced by Bergey [27]. The ramp generator creates multiple slope ramp that allows ADC resolution scaling for a given shot noise limit. Hence, tradeoffs between frame rate, resolution, and power dissipation can be made easily without a redesign depending on the application.

Simultaneous Multiple-Slope Ramp (SMSR)ADC

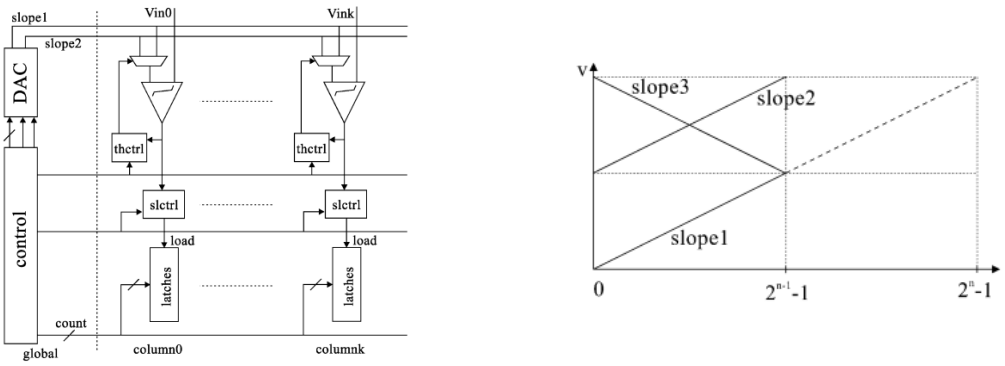
Simultaneous multiple-slope ramp (SMSR) ADC was first proposed by Lindgren [28], as illustrated in Figure 2.17 (a). It achieves almost twice the speed of conventional ADC by adding small extra circuitry. This type of ADC uses the same circuitry as conventional single-slope ADC except that it has two different phases of conversion; comparison phase and slope phase. During the slope phase, many slopes are used in parallel, such that each slope covers a section of the total signal swing. The additional circuits over conventional single-slope ADC are some control circuits and analog multiplexers added at comparator input. The available slopes are applied to an analog multiplexer allowing each column to choose the appropriate slope to connect with.

A comparison phase is the start of the conversion process. All slopes are first set to an initial value and applied to multiplexers, as illustrated in Figure 2.17 (b). After that, the column multiplexer's output is scanned and compared to the column input signal. The comparator output of



(a) (b)

Figure 2.16. (a) Differential Ramp Generator as in [26] (b) Real-Time Calibration Scheme as in [27]



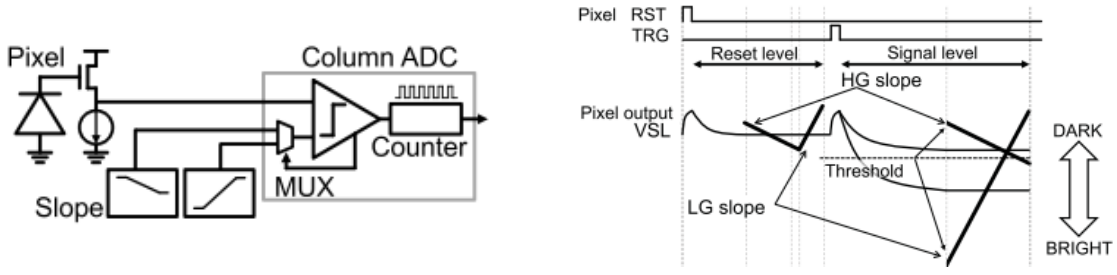
(a) (b)

Figure 2.17. (a) Block Diagram of the SMSR Architecture (b) Slopes of SMSR Architecture [28]

each corresponding slope is stored in a register in each column. The stored comparator output will be used in the next slope phase. For the slope phase, the actual slope operation is performed, and the global counter output is fed to the slope generator and columns. Once the output of the comparator is changed, the counter value is latched. The combination of the stored threshold value and latched counter value results in the digital result.

Dual-Gain Single-Slope Ramp ADCs

Figure 2.18 (a) illustrates a new proposed gain adaptive column ADC based on single-slop ramp ADC. It was presented in [29] to fulfill the high frame rate requirements and high dynamic range of CIS. It applied two slope references of different gain instead of the programmable gain amplifier, as shown in Figure 2.18 (b). Pixel output levels of each column are responsible for switching between the two slope references. This technique adds a small area and power consumption without adding any noise source. It can also suppress the nonlinearity of the sensor due to real-time gain correction and small FPN without any necessary calibration. That paper achieved 480 fps input-referred dark random noise of 140 µVrms with 923mV for input-referred full-scale readout.



(a) (b)

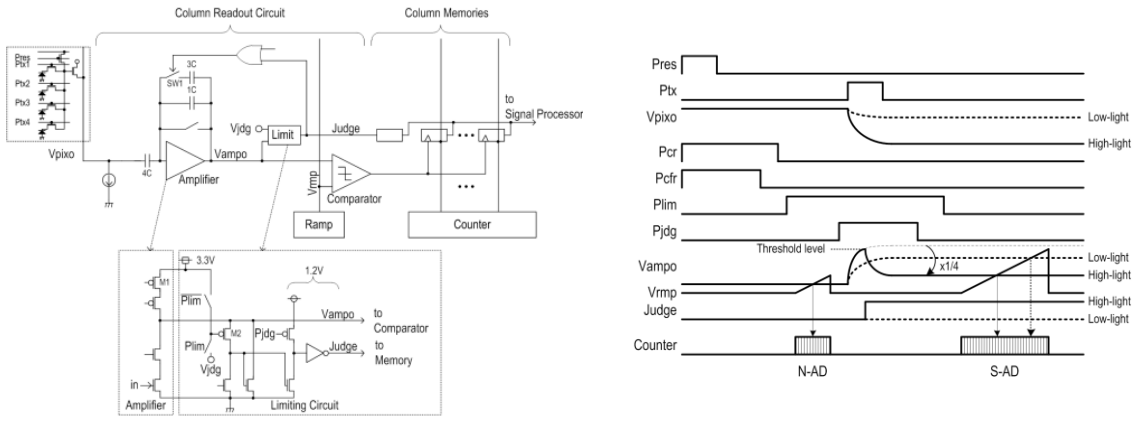
Figure 2.18. (a) Gain Adaptive Column ADC (b) Concept of A/D operation with two slope references as in [29].

Single-Slope Ramp (SSR) ADCs with Dual-Gain Amplifier

Single-slope ramp (SSR) ADC with dual-gain (SSDG) amplifier idea was proposed in [30] to speed-up the conventional SSR ADC and widen the dynamic range without increasing in power consumption in CIS CPA. The block diagram of the SSDG ADC is illustrated in Figure 2.19 (a), and the timing chart is shown in Figure 2.19 (b). A column amplifier gain is set to high gain (4X) if the pixel signal level is below a threshold level (approximately 1/4 of pixel signal value) in order to minimize random noise (RN). After that, the column amplifier gain is set to low gain (1X) if the pixel signal level exceeds that threshold to prevent exceeding the amplifier output range. In that case, the dynamic range is wider than SSR as an RN is suppressed, and the signal level is kept the same. This work achieved 5 fps for full pixel readout, 24fps at 8k4k, and 48fps at 4k2k. The chip power consumption was 1.97W at 5 fps full-pixel readout and 12-bit resolution, while the dynamic range was 6 dB wider than SSR ADC and had a value of 66.7 dB

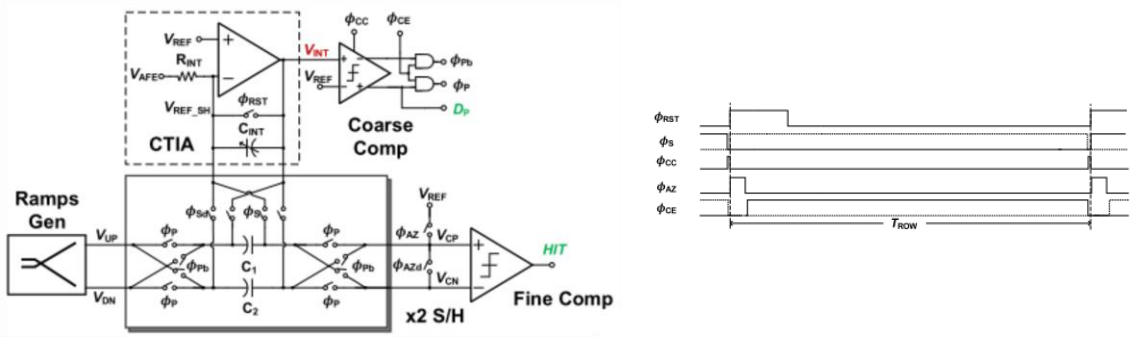
Differential-Ramp Single-Slope (DRSS) ADC

Differential-ramp single slop (DRSS) ADC was introduced in [31]. This ADC consists of four main circuits: pre-comparator, main comparator, course-fine time to digital converter (TDC), and low pass filtered offset. The DRSS scheme and its timing diagram are shown in Figure 2.20. The differential ramp SS ADC is based on the capacitor trans-impedance amplifier (CTIA) with a two-step coarse fine conversion scheme. At the end of each line, the coarse comparator completes 1-bit coarse quantization. This scheme helped reduce the counter frequency to 70MHz instead of 500MHz that cannot be fulfilled by that technology achieving 2X faster conversion speed and better ADC noise performance with reduced power and the chip area.



(a) (b)

Figure 2.19. (a) The block diagram of the SSDG (b) The Timing chart of the SSDG as in [30].

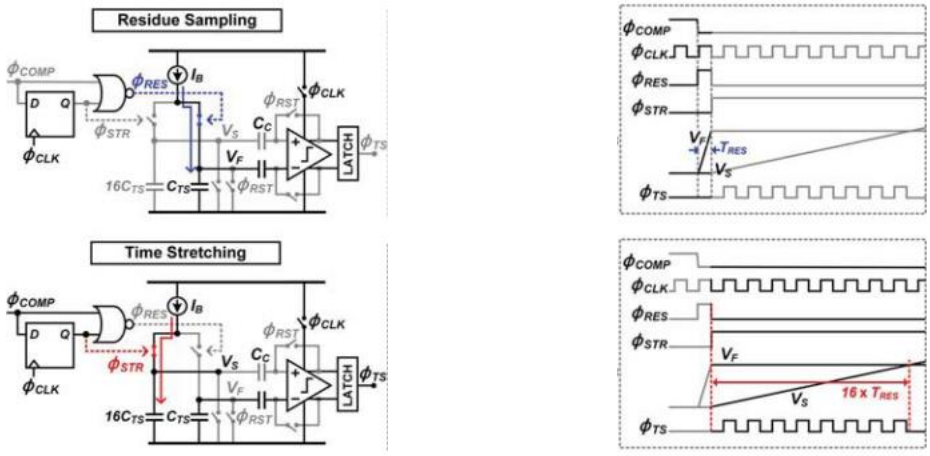


(a) (b)

Figure 2.20. (a) The block diagram of the DRSS (b) The Timing chart of the DRSS as in [31].

Time-Stretched Single-Slope Ramp (TS-SSR) ADC

A new technique was presented in 2019 by Injun et al. [32] called time-stretched single-slope ramp (TS-SSR) ADC. The block diagram and the timing scheme is illustrated in Figure 2.21. It achieved 500 fps, 1095e^{-} random noise, and 76 mW power consumption. It is proposed a column-parallel time stretcher that expands SS ADC's time residue to 16 times and reduces the conversion cycle of SS ADC 80 cycles for 10-bit resolution. This allowed using the clock with a lower frequency, which leads to reducing power consumption significantly. This architecture uses a single ramp generator for SS ADC so. It cannot degrade noise performance and gain the function of SS ADC.



(a) (b)

Figure 2.21. (a) The block diagram of the TS-SSR (b) The Timing chart of the TS-SSR as in [32].

Two-Step Sampling Ramp ADCs

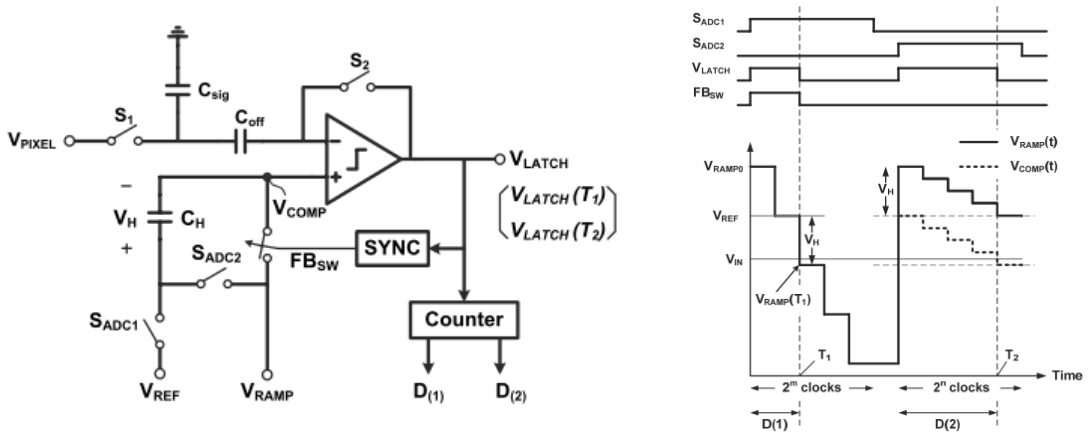
The second category of ramp ADCs converts analog signal using two steps or phases; coarse and fine phases. According to a different setting in each technique, the switching from the coarse phase to the fine phase occurred.

Two-Steps Single-Slope Ramp (2S-SSR) ADC

Lee, in [33], [34] presented CIS CPA integrated two-step single-slope ramp (2S-SSR) ADC to enhance the sampling rate in order to increase the speed of SSR ADC. This scheme is divided into two steps; m-bit coarse SS ADC and n-bit fine SS ADC as shown in Figure 2.22. The first step has 2^m steps to scan full range and quantizing m-most significant bits (MSBs) while the fine ramp steps are 2^n to quantize n-LSBs. However, the fine steps ranged only within coarse steps with the same slope and smaller steps sizes. This makes the required number of clocks for a sample is (2^m+2^n) instead of (2^{n+m}) for conventional SS ADC. In addition, this work introduces a solution of missing codes between coarse steps of ADC by doubling the fine range to cover the boundary between coarse steps. As a result, speed-up the conventional SS ADC by factor of 10 in addition to theoretical maximum frame rate of 1000 fps but power consumption increases of 25%.

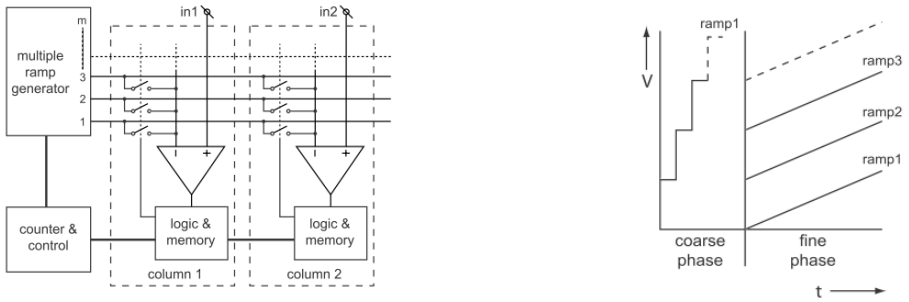
Multiple-Ramp Single-Slope (MRSS) ADC

The development of the latter idea was proposed in [35] by Snoeij. Multiple ramp single-slope ADC idea is introduced to solve the speed problem of SS ADC by trading power consumption and speed and keep the advantages of SS ADC like simplicity. The column circuit consists of one comparator that can connect to multiple ramp voltages simultaneously, as shown in Figure 2.23. The conversion process is divided into coarse and fine phases. For the coarse phase, the first ramp is



(a) (b)

Figure 2.22. (a) The block diagram of the 2S-SSR (b) The Timing chart of the 2S-SSR as in [33].



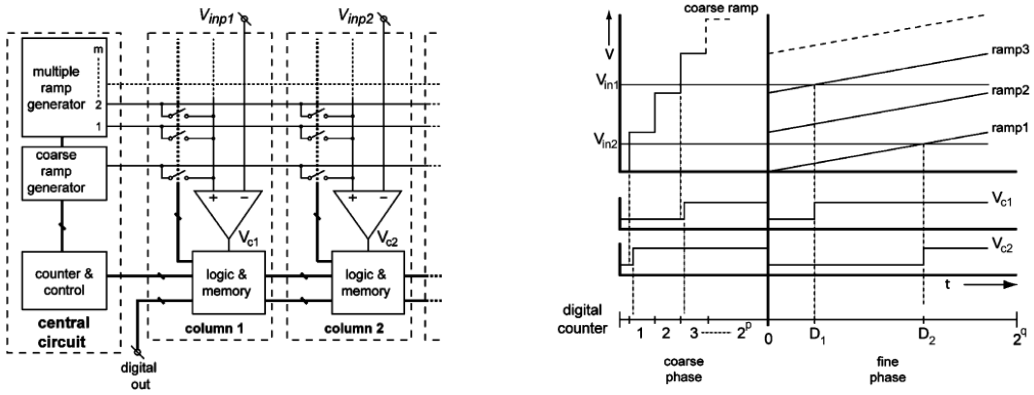
(a) (b)

Figure 2.23. (a) The block diagram of the MRSS (b) The Timing chart of the MRSS as in [35].

connected to all comparators. Coarse conversion is performed to provide an initial coarse estimation (p-bit) of input signal level during the fine phase. All ramps with the same slope are scanning a specific range of coarse ramp concurrently, and each ramp is connected to appropriate comparator according to coarse phase results. This technique reveals a reduction in conversion time to 3.3Xwith an increase of frame rate 2.8X, and power consumption is increased by 24%.

Multiple-Ramp Multiple-Slope (MRMS) ADC

The same author of MRSS developed the latter technique to be multiple ramp multiple slope ADC [36], which achieved a 25% faster speed with the same power consumption. In this work, the photon shot noise amplitude-dependent nature of imager signals is exploited to exhibit a companding characteristic. Figure 2.24 illustrates the block diagram and the timing scheme of the MRMS. As MRMS is MRSS-based, it follows the same operation theory except for the fine ramp signals; instead of the same slope for each ramp, the shot-noise-based accelerated ramp is used to speed up MRSS the same power consumption.



(a) (b)

Figure 2.24. (a) The block diagram of the MRMS (b) The Timing chart of the MRMS as in [36].

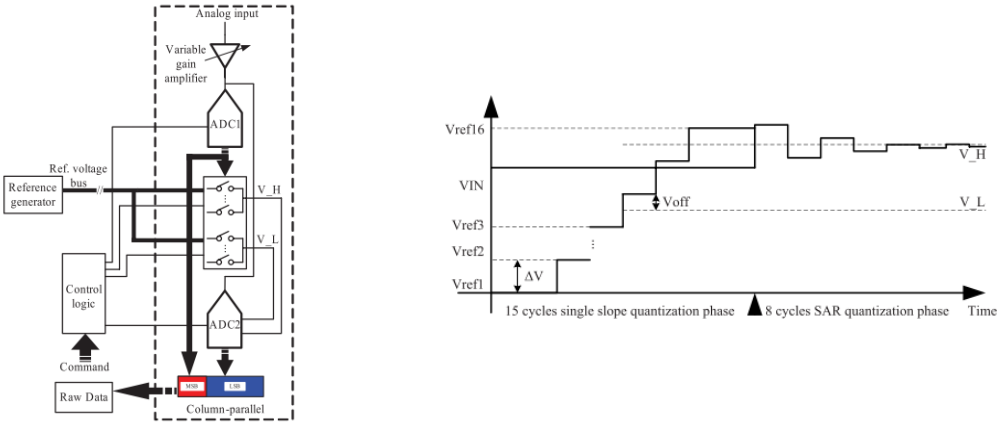
Hybrid Sampling Ramp ADCs

The last category of ADCs is hybrid sampling and operation ADC techniques. It tries to take advantage of the high conversion speed of ADCs like successive-approximation register (SAR) or flash type ADCs and the advantage of simplicity and area efficiency of SSR ADC. However, these techniques are successful, somehow trade the gained advantages with other requirements.

Hybrid SS-ADC/ SAR type

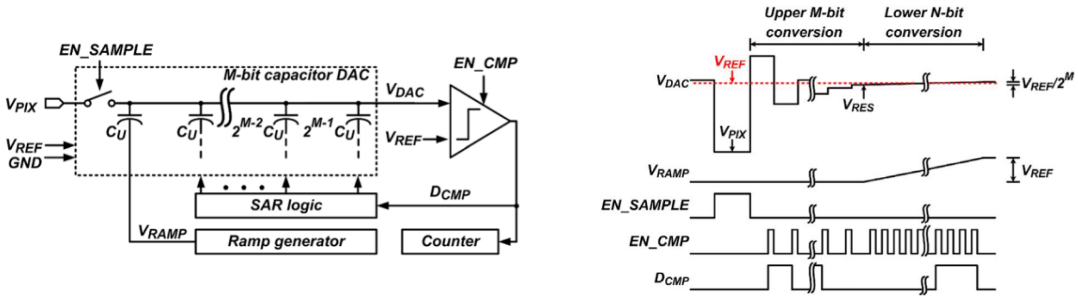
The combination of SS-ADC with SAR ADC was first reported in [37], as shown in Figure 2.25. 11-bit, two steps, CPA ADC, was realized in that work. 3-bit SS-ADC realized the 11-bit Digital output as MSB and 8-bit SAR as the rest of the digital word. It is claimed that power consumption is reduced compared to SSR ADC. The chip area was reduced compared to 11-bit SAR ADC as an 8-bit resolution relaxed the capacitor array matching and eliminated the need for any calibration to guarantee the monolithic quantization response. However, in order to achieve high resolution, a lot of accurate reference voltages are required because reference voltages of lower bit conversion depend on upper bit reference voltages.

Another hybridization between SS-ADC and SAR ADC is presented in [38]. This technique realizes 12-bit hybrid SS/SAR ADC with less power consumption by sharing analog circuits between SS and SAR ADC. 6-bit SAR-based MSB conversion is followed by a 6-bit SS-based LSB conversion of the residue to achieve a 12-bit final output word, as illustrated in Figure 2.26. However, hybrid multi-step SS /SAR converters are more complex due to additional circuitry involved in residue extraction and resampling.



(a) (b)

Figure 2.25. (a) The block diagram of the SS-ADC/SAR (b) The Timing chart of the SS-ADC/SAR as in [37].



(a) (b)

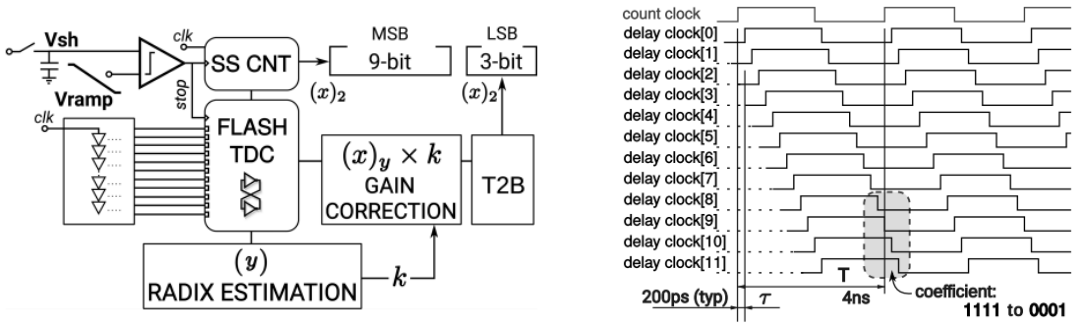
Figure 2.26. (a) The block diagram of the SS-ADC/SAR (b) The Timing chart of the SS-ADC/SAR as in [38].

Hybrid Single-Slope Time to Digital Conversion (TDC) and Flash Type ADC

CPA TDC Flash-type ADC is combined with SS-ADC in [39]; it solves the problem of multiphase clock period of flash TDC without using a delay-locked loop. Figure 2.27 (a) illustrates the system-level concept of the interpolation method with gain calibration. Figure 2.27 (b) shows the principle of flash TDC-interpolated SS ADCs and calibration proposal. This technique uses open-loop delay elements for clock generation to gain calibration using per column digital multiplication operation. The correction scheme is applied after each conversion to save less than 5% of ramp time.

Conclusion

This chapter started with a quick revision of CCD history and how it was very dominant and mature for years in the industry. Then, how CIS emerged and how it was very promising at that time is also explained the reasons that forced researchers to spend more time to explore and develop it until it dominates in the industry today are illustrated. CIS system level and different architectures were explained, showing the advantages and disadvantages of each architecture. Each block of the system



(a) (b)

Figure 2.27. (a) System-level concept of the interpolation method with gain calibration (b) The Principle of Flash TDC-Interpolated SS ADCs and calibration as in [39].

was explored to understand each block's function and how it affects the overall function of CIS. A literature survey of speed-up techniques was performed, revealing the amount of research done to speed up SSR ADC as the only drawback is the conversion speed. The next section will illustrate the new speed-up technique for CPA CIS called single-slop look-ahead ramp ADC.

Chapter 3: Single Slop Look-Ahead Ramp (SSLAR) Analog to Digital Converter (ADC)

"Single-Slope Look-Ahead Ramp ADC for CMOS Image Sensors." IEEE Transactions on Circuits and Systems I: Regular Papers, Early Access, 2020, pp. 1-10.

Integrating (ramp) type analog-to-digital converters (ADC) used in column-parallel CMOS image sensors trade conversion speed with size, power, and circuit complexity to achieve optimal performance. A new integrating (ramp) type ADC architecture called single-slope look-ahead ramp (SSLAR) ADC is introduced in this chapter. It utilizes a statistical approach and code-prediction methods to improve the standard single-slope ramp (SSR) ADC conversion speed. It is shown that SSLAR ADC reduces power consumption while achieving an increased frame rate. This is achieved by the SSLAR algorithm optimized for column-parallel CMOS active pixel sensor (APS) imager architecture. The characterization result of a 10-bit SSLAR ADC designed in a 0.5µm CMOS (2P3M) process and integrated with a column-parallel CMOS image sensor with 200 × 150 array with 15µm pixels is presented. Measurements showed that a six times (6x) frame rate increase could be achieved while reducing power consumption by 13% with minimal impact on image quality.

Introduction

The CMOS image sensor is the technology of choice due to its advantages over the charge-coupled device (CCD) in today's still image and video capturing consumer devices [8] Particularly, CMOS APS imagers have two main image readout architectures that provide these advantages [40]. Column series architecture (CSA) is used in most low-resolution, high-volume, and highly commoditized CIS based camera systems. The high-resolution, high-speed CMOS image sensors take advantage of the second architecture called column-parallel architecture (CPA). CPA utilizes column level low-speed, low-noise, analog domain signal processing, and digitizing circuits. Correlated double sampling (CDS), analog noise cancellation, sampling, amplification, level shifting, and dynamic range adjustment are typical operations performed on pixel signals in column analog signal processors (CASP) before ADC digitizes them.

Historically, four types of ADCs have been used in CPAs without degrading the competitive advantage of CMOS APS imagers. They are successive approximation register (SAR), [14] [41] [42], integrating [43] [44] [45] [46]**,** sigma-delta [47] [48] [49] [50] [51], and cyclic or algorithmic type ADCs, [52] [53] [54] [55] [56] [57] [58] [59] [60], Integrating type ADCs are also known as single-slope or multiple-slope ramp (SSR or MSR) ADCs. Among these ADC topologies, the most noteworthy one for CPA is the SSR ADC [22] [23] [24] [27] [26] [35] [36] due to its advantages on

multiple fronts, including ease of integration, smaller silicon footprint, low-power consumption, low-voltage operation, high bit resolution, low-noise operation, and low design complexity. However, they suffer from conversion speed, especially when bit resolution is increased to 10-bit or more.

In the past, many solutions have been proposed to overcome the resolution and speed issues of SSR ADCs used in CIS CPAs, [22] [23] [24] [27] [26] [35] [36]. These solutions can be categorized into four different techniques; double-slope ramp (DSR) [22] [23], accelerated ramp (AR) [19] [27] [26], multiple-ramp single-slope (MRSS) [35], and multiple-ramp multiple-slope (MRMS) [36]. These solutions reduce conversion time by using added global or column level circuits and architecture design techniques. All existing solutions perform analog-to-digital conversion blindly scanning all possible codes between 0 and (2^n-1) using n-bit digital counter and ramp generator. Indeed, this is the main reason why there is a tradeoff between speed and resolution. On the other hand, if the code distribution of the sampled pixel signals (or row histogram in CPA) is known, some code ranges can be skipped on analog (ramp) and digital domains (counter) resulting in improved conversion speed and reduced power consumption, [61] [62]. In this chapter, implementation details and measurement results of a new method to predict code distribution, and a ramp ADC algorithm to accelerate analog-to-digital conversion operation for column-parallel CMOS image sensors is presented. Both were implemented on hardware achieving code range look-up, jump, and fallback operations on analog and digital domains, and used for developing the proposed single-slope look-ahead ramp (SSLAR) ADC. 10-bit SSLAR ADC is integrated with a column-parallel CMOS APS image sensor having a 200 × 150 array of 15µm × 15µm three-transistor (3T) APS pixels. Design is fabricated in a standard 0.5µm, 2P3M CMOS process and measurement results are presented

This chapter is organized as follows. First, the operational principle of the SSLAR ADC algorithm is provided. Then, the implementation and operational limitations of the SSLAR ADC algorithm are described. The Circuit level implementation of the SSLAR ADC algorithm and integration of the design on a CMOS image sensor are presented afterward. The measurement techniques, metrics to quantify the image quality performance of SSLAR ADC, and the measurement results are presented before the discussion and conclusion.

SSLAR ADC Operation Principle

Focal plane imaging systems compose of optics focusing scene images on its field-of-view (FOV) onto image capturing arrays of photosensitive elements confined in pixels. Today's mainstream imaging devices use a so-called electronic rolling shutter operation that allows capturing and processing two-dimensional (2D) scene images sequentially. Typically, this is done row-by-row and by using fixed integration times. Analog pixel signals on the addressed row are sampled and hold

for further processing on CASPs. Typically, reference (reset) and light-induced signal voltages of each pixel on the addressed row are sampled and processed in CASP. These pixel signals are subtracted (correlated double sampling-CDS), noise-shaped, level-shifted, and, most of the time, amplified in the analog domain by the CASP circuits. In CPA architectures, they are digitized by an ADC placed alongside CASP circuits on each column. Digitized pixel signals are sequentially sent to digital signal processing (DSP) circuits for further processing.

A Block diagram of a CMOS CPA image sensor composing of SSR ADC is shown in Figure 3.1. Each column ADC has a low-speed comparator and n-bit transparent latch following the CASP circuits. Effective pixel voltage ($V_{in}[i]$) is processed and held at a comparator's input. The other input of it is connected to a global ramp generator, V_{ramp}. The comparator drives transparent latches, which pass digital inputs when its control input is high and holds the last digital inputs when it is low. Digital inputs of the latches are connected to the n-bit global counter. After comparator inputs are settled, the global ramp generator and counter are activated, generating analog and digital ramps. When comparator output changes its state from logic 1 to 0 due to $V_{ramp}>V_{in}[i]$, latches on column *i* holds the last counter bits digitizing the input voltage, $V_{in}[i]$. Since each column might have a different input voltage, the ramp signals have to scan full analog and digital ranges blindly. This blind ramping operation is depicted in Figure 3.2(b) for *row[j]* shown in Figure 3.1. Code distribution of

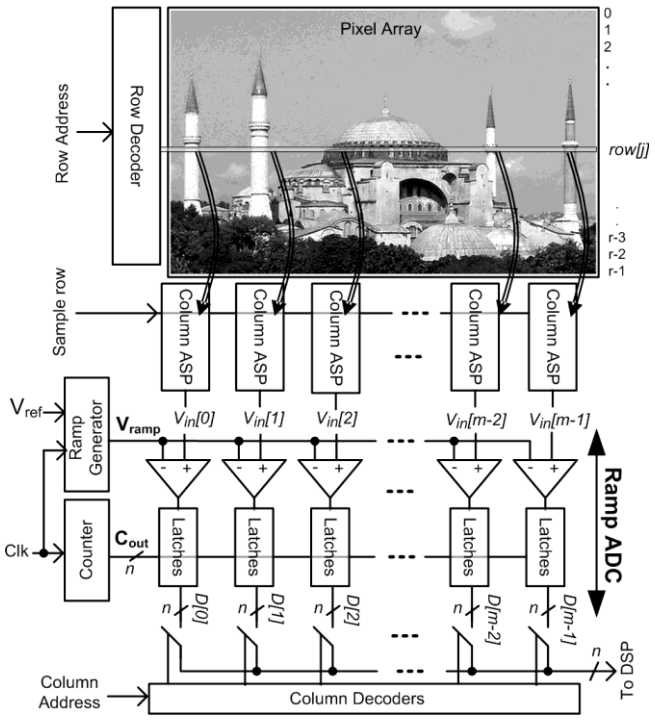


Figure 3.1. Building blocks of a CMOS APS image sensor with CPA and integrated single-slope ramp ADC.

row[j] signals (row histogram) is shown in Figure 3.2(a) over the 8-bit code range of the ADC. The vertical axis in Figure 3.2(a) represents the number of columns that have the particular ADC code. For example, none of the columns have codes between 0 and 48, 62 and 90, and 220 and 255 for this particular row of pixels. This is a typical occurrence for captured images that row analog signals do not exist in all code ranges of digitizing column-parallel ADCs. Another example, For example, if row#2 of the image in Figure 3.1 is digitized, it can be found that 90% of the digital outputs are between 64 and 71, and the rest is some individual codes outside this majority code range that can be skipped. Thus, scanning code ranges between 0 and 63, and 72 and 255 is a wasteful operation for column-parallel SSR ADC. Indeed, suppose the distribution of the ADC codes of the sampled column signals is predicted. In that case, digitization of entire row signals can be accelerated while reducing power consumption and conversion time, as seen in Figure 3.2(c). However, this requires a code look-ahead scheme for forecasting when to perform ramp, jump, and fallback operations for digital counter and analog ramp generators [63] with controlled steps and thresholds. This is exactly what the proposed single-slope look-ahead ramp (SSLAR) ADC algorithm does. It considers the general nature of the everyday scenes that are captured by an image sensor, concluding that spatial variation and distribution of row of pixel signals might have localized distributions on digital code domain, allowing code skipping during digitization and resulting in opportunities to the speed-up conversion process and to reduce power consumption. Best images that have this condition are the ones in which all row pixel signals are accumulated in very small code ranges, allowing few clock cycles to be used for completing analog-to-digital conversion using the look-ahead ADC algorithm. Row #2 of the

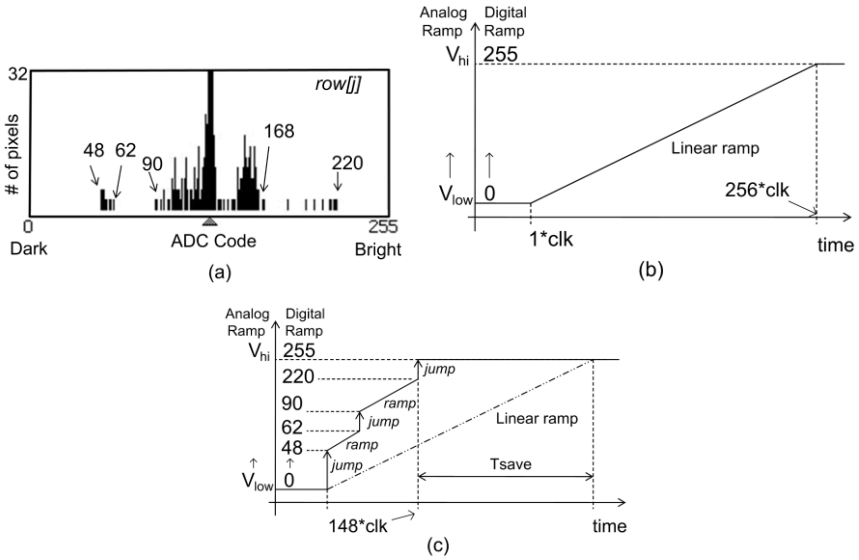


Figure 3.2. ADC operation of a captured image by a CPA CIS; a) 8-bit code distribution of pixel signals on row[j] on Figure 3.1, b) blind A-to-D conversion of pixel signals on row[j], c) ideal look-ahead ramp ADC operation of row[j].

image in Figure 3.1, for example, requires less than 16 clock cycles for ideal look-ahead analog-to-digital conversion resulting in more than sixteen times (16x) speed and/or power improvement opportunity. On the other hand, in worst cases, codes are uniformly distributed, resulting in full scanning of all codes using 256 clock cycles. Thus, the ideal look-ahead conversion would result in a "blind" conversion for the worst case as it will scan all codes without performing any jumping processes. As a result, ideally, digitization of an array of pixel signals using the proposed SSLAR ADC algorithm results in always faster than a blind one [61]. The next section reviews how would the proposed look-ahead ADC behaves under non-ideal conditions.

SSLAR ADC Algorithm

As mentioned earlier, if the digital code distribution of input analog voltages is known, ramp ADC operation can be modified and accelerated. A code look-ahead or prediction can be made instantaneously for an ideal case. However, in reality, this prediction requires a computation time in the form of a number of clock cycles, which can be defined as (*h*). Indeed, *h-clock cycles* have to be used when a look-ahead operation is performed.

Another deviation from the ideal look-ahead operation is related to how to quantify if there is (or is not) enough number of columns fell in the look-ahead code range. A look-ahead range and a threshold have to be used in real implementation to make a judgment resulting in code jump or fallback operations. Thus, the code look-ahead range (both in analog and digital domains) has to be set reasonably such that the number of columns falling in this range (z) has to be counted and compared against a threshold level (s). In the SSLAR algorithm, *step size* (*k*) is the code look-ahead range, and *jump threshold* (*s*) is the maximum allowed number of columns to approve the look-ahead process.

A Block diagram of the SSLAR ADC architecture for column-parallel CMOS image sensors is shown in Figure 3.3. Comparing with a typical SSR ADC shown in Figure 3.1, SSLAR ADC has one extra block called predictor on each column. The global ramp generator and counters were modified [63], and extra two blocks, event detector, and a global look-ahead controller (LAC) were added to implement the SSLAR ADC algorithm [62]. Modified global ramp generator and counter could be able to jump k-step and k/2-code (i.e., least significant bit, LSB) ahead from their current, voltage, or code level, respectively. They also accommodate the fallback of k-step and k/2-code if the LAC block does not approve jump operation. The predictors' function is to generate a signal

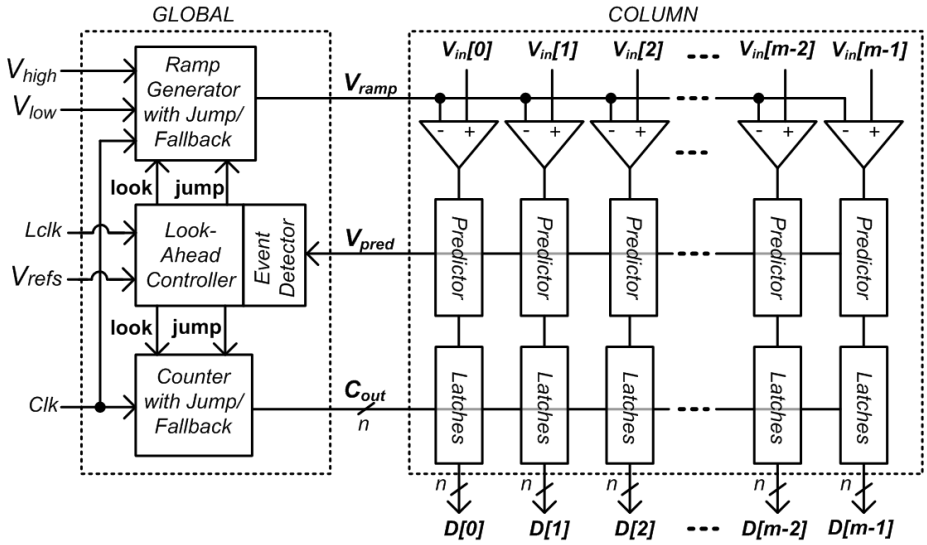


Figure 3.3. Block diagram of SSLAR ADC integrated in CMOS APS imager.

collectively, informing the block about the number of column comparators that changed their state during look-ahead operation. Using these extra blocks and operation principles, the SSLAR ADC algorithm could be

implemented following the steps given in Table 3.1. After a row of pixel signals sampled on *m*-number of CASPs, and global ramp and counter circuits are reset, the analog-to-digital conversion starts with *k* look-ahead steps. Before the look-ahead operation, LAC initializes the event detector and asserts a jump signal. Before the ramp voltage is increased, the counter code is incremented (*k/2*)-LSB and let column latches to pass this level first. Only *k/2* LSB is assigned for the counter, as this value is the best digital representation of columns in the code range if a jump is approved. Next, ramp voltage is increased *k*-LSB equivalent voltage given with equation (3-1)

$$\Delta V_{ramp} = k \cdot \left(\frac{V_{high} - V_{low}}{2^n}\right) \tag{3-1}$$

Right after ramp and counter signals settled on their jump levels, the event detector will process the column predictor outputs quantifying the number of comparators that changed states due to jump performed by the ramp generator. Simultaneously, column latches will lock on the counter's current output if its control input from the comparator is changed. At this point, LAC receives information (*z*) from column predictors, which are proportional to the number of column comparators that have changed their state, and compares this number with the jump threshold (*s*). If *z* is smaller than *s*, then LAC will approve jump, incrementing the counter by another k/2-LSB. This means that there might be a number of columns (*z*) exist that their sampled pixel voltages fall in between the *k*-LSB look-ahead range, but they are less than the threshold (*s*). Hence, the SSLAR algorithm blindly

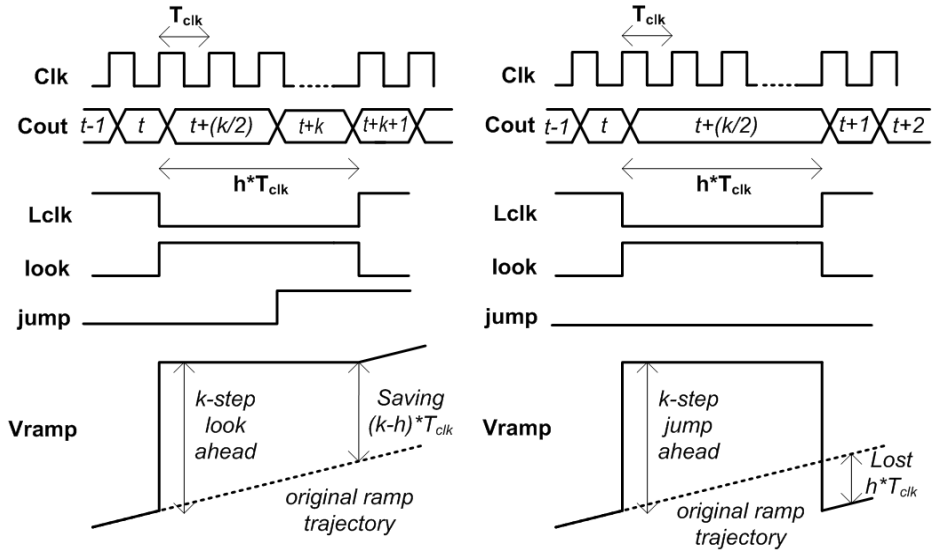
Table 3.1. Single-Slope Look-Ahead Ramp (SSLAR) ADC Algorithm

Step		Operation
1		Sample a row of pixel signals on column ASP circuits
2		Reset global ramp generator and n-bit counter
3		Initialize column predictors for possible jump operation
4		Increment global counter (k/2)-LSB
5		Increase ramp voltage k-LSB equivalent analog level
6		Check column predictors
7		Are there enough column comparators changing their outputs in k-LSB code range?
	A:	If YES: Fallback k-LSB on ramp voltage and (k/2)-LSB on counter output. Ramp and count 1-LSB at a time for k-LSB, then go to Step 8.
	B:	If NO: Do not change analog ramp voltage, increment counter (k/2)-LSB step more, go to Step 8
8		Have you reached 2^n-1 bit range?
		If NO, go to Step 3,
		If YES, go to Step 9
9		Have you read all *m* rows?
		If NO, increment row address, go to Step 1,
		If YES, end of frame.

quantizes these *z* numbers of column voltages uniformly with value equals to the previous code plus *k/2*-LSB or the mid-code value of (C_{out}+*k*/2) where C_{out} is the last counter output. This is registered as a quantization error or noise during the conversion. The worst-case occurs when all columns in every *k*-step range have the same voltages and are located at the far end of the look-ahead range. So, the total quantization error per row due to the proposed look-ahead algorithm is given by equation (3-2).

$$E_{row} = \frac{k}{2} \times \sum_{j=1}^{\frac{m}{k}} z_j \tag{3-2}$$

where *m* is the number of columns, suppose the z is larger than or equal to s. In that case, LAC will disapprove jump operation because many columns exist in the look-ahead range, and assigning mid-code for all pixels will result in a large quantization error. Thus, it will force the ramp to fallback *k*-step and counter to fallback *k/2*-LSB. It then asks the ramp generator and counters to increment one LSB at a time for *k*-steps during the next *k* clock cycles.



(a) (b)

Figure 3.4. Timing diagram of LAC, counter and ramp generator blocks of SSLAR ADC a) jump is approved b) jump is denied [56]

The two cases that a jump is approved and denied by LAC are depicted in Figure 3.4. When a jump is approved, (*k*-*h*) number of clock cycles are saved. If a jump is denied, *h* clock cycles would be lost. In the worst case, the proposed algorithm will slow down the conversion operation. In the best case, all column voltages will be in one *k*-step range of the SSLAR ADC. Indeed, this case could be observed for the first few rows seen in Figure 3.1. The worst case is observed when more than s number of pixel signals exist on every *k*-step range of the SSLAR ADC. This is the case when the sampled image has pixel values covering the full ADC range and *s*<*k*. Thus, the best and the worst conversion time of SSLAR ADC can be calculated with equations (3-3) and (3-4), respectively.

$$T_{SSLAR,Best} = \left(\frac{2^n}{k} \cdot h + k\right) \cdot T_{clk} \tag{3-3}$$

$$T_{SSLAR,Worst} = \left(\frac{2^n}{k}\right) \cdot (h + k) \cdot T_{clk} \tag{3-4}$$

For all cases, standard SSR ADC has the same conversion time of 2^n clock cycles. To quantify the speed-up ratio (R_{sup}) with the new ADC algorithm, the SSR ADC conversion time is normalized with the conversion time of the SSLAR ADC for best and worst cases given with equation (3-5).

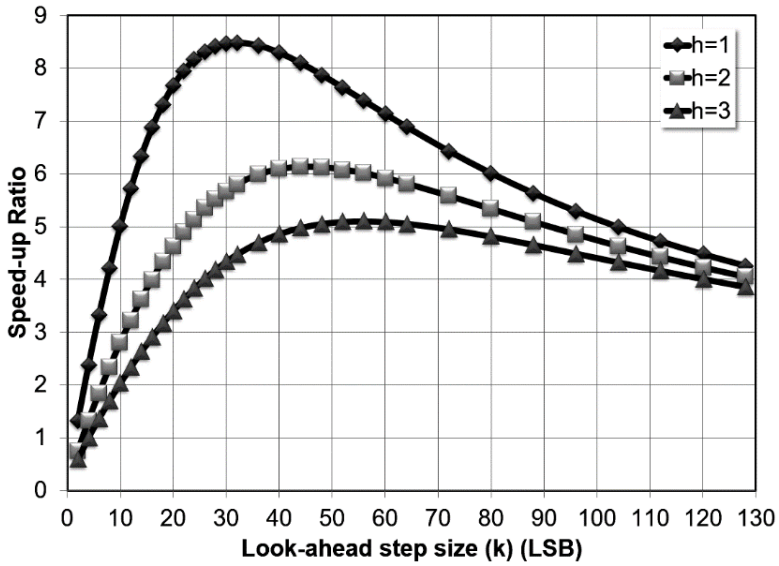


Figure 3.5. Average speed-up ratio of SSLAR ADC

$$R_{sup} = \frac{T_{SSR}}{T_{SSLAR}} = \begin{cases} \left(\frac{k}{h + k^2 \cdot 2^{-n}}\right) & Best \\ \left(\frac{k}{k+h}\right) & Worst \end{cases} \quad (3\text{-}5)$$

The average speed-up ratio, where half of the pixel signals have the worst case pattern, and the half has the best case, is plotted in Figure 3.5 for 10-bit SSLAR ADC for different *h* and *k* values. *h* is the cost of look-ahead operation and could be 1, 2, or 3 clock cycles. For this particular case, the SSLAR algorithm could result in at least six times (6x) speed improvement.

One important feature of the SSLAR algorithm is that it works like a regular SSR ADC when *k* and *s* are set to 1-LSB. In this case, however, SSLAR ADC works slower than SSR ADC, as seen in Figure 3.5. This feature allows us to evaluate the SSLAR ADC algorithm on the same focal plane and quantify speed-up ratio and noise level or image quality (IQ) degradation.

SSLAR ADC and Image Sensor Design

Column Analog Signal Processor (ASP) Circuits

Analog signal processor (ASP) circuits from the pixel photodiode (PD) node to digital outputs are shown in Figure 3.6. 3T CMOS APS pixel composes of reset (M_1), select (M_3), and source follower (M_2) transistors. Column ASP comprises a programmable charge amplifier (A1) that performs CDS and sample-and-hold (S/H) circuit composed of a switch and C_1 capacitor. The bottom plate of C_1 is connected to the global ramp signal (V_{RAMP}). Amplified pixel voltage (ΔV_{sh}) is sampled and held at the ADC comparator's input with respect to the clamp voltage, $V_{CM,}$ as in (3-6).

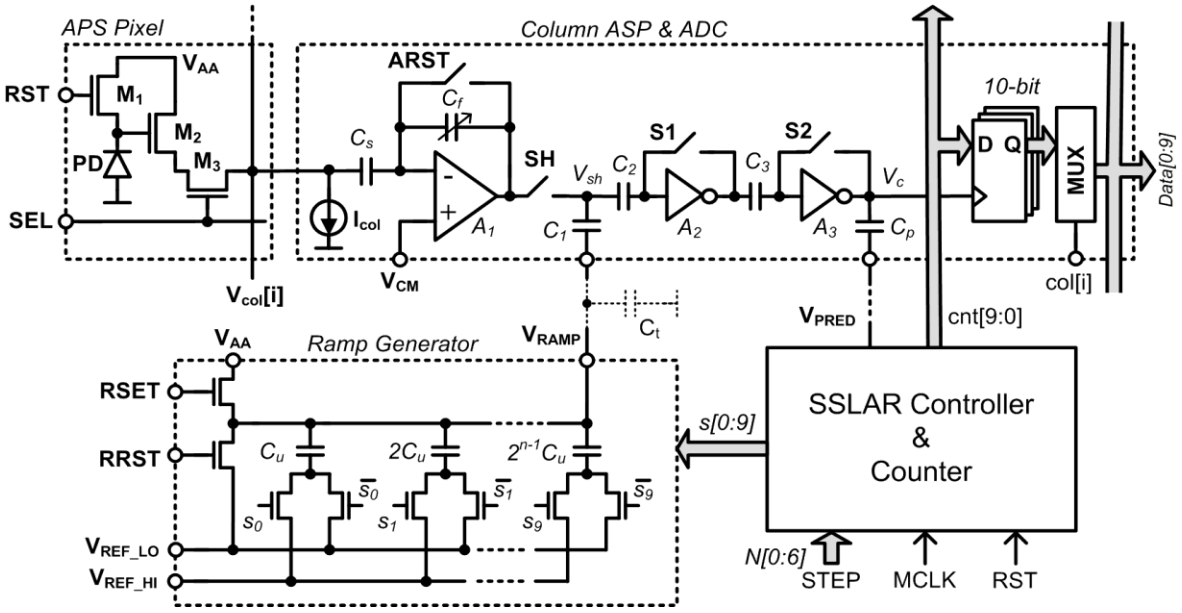


Figure 3.6. Column ASP and ADC circuits of the SSLAR imager.

$$\Delta V_{sh} = V_{CM} - A_{SF} \cdot \left(\frac{C_s}{C_f}\right) \cdot \left(V_{PD,rst} - V_{PD,sig}\right) \tag{3-6}$$

where A_{SF} is the gain of pixel source follower, $V_{PD,rst}$ is PD reset, and $V_{PD,sig}$ is PD signal voltages. ADC comparator is formed by two inverting amplifiers (A_2, A_3), offset nulling capacitors (C_2, C_3), and switches (S1, S2). The column predictor composes of a single capacitor, C_P. It is connected between the output of the comparator and the global event detector bus (V_{PRED}). The comparator output is connected to 10 transparent data latches. Latches pass the global counter signals when the comparator output is low and hold last known inputs when it is high. The timing diagram of the readout chain is shown in Figure 3.7.

The global binary-weighted ramp is directly coupled to the column ASP through the S/H capacitor, C_1. A voltage buffer was not used to drive the ramp generator voltage for saving power. Indeed, ramp drivers are one of the main power consumers in a typical SSR ADC used in a CIS due to their large parasitic loads (C_t). This capacitor is proportional to the number of columns (m) and the effective column S/H capacitor's size. Assuming the first comparator amplifier (A2) input capacitance is much smaller than its nulling capacitor, C_2, the ramp step size can be found by using equation (3-7).

$$\Delta V_{\mathrm{RAMP}} = \frac{C_u \cdot \left(V_{\mathrm{REF\_HI}} - V_{\mathrm{REF\_LO}}\right)}{(2^n - 1) \cdot C_u \cdot \left(1 + \frac{C_2}{C_1}\right) + m \cdot C_2} \tag{3-7}$$

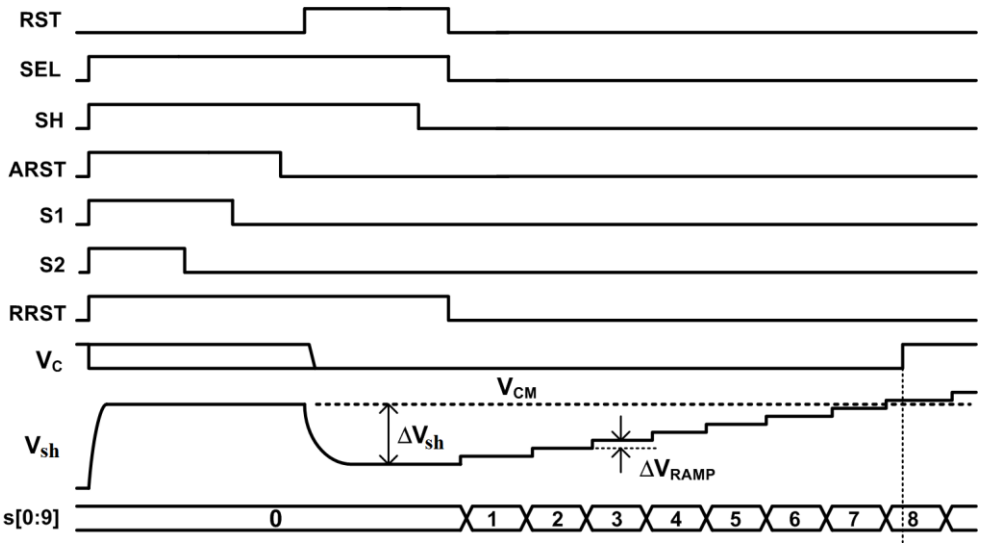


Figure 3.7. Timing diagram and associated node signals of the ASC and ADC

where C_u is the unit capacitor used in the ramp generator. In this design, ΔV_{REF}=2.5V, C_u=50fF, C_1=520fF, C_2=260fF, n=10, and m=200 are used. Thus, the minimum ramp step was 0.97mV with a 10-bit resolution and 1V ADC input range.

Global SSLAR ADC Blocks

The global section of the SSLAR ADC composes of three blocks; SSLAR controller (CONT), event detector (ED), and ramp-count generator (RCG), as shown in Figure 3.8. . CONT block implements the SSLAR algorithm using feedback from ED. It also generates control and clock signals (C0, C1, C2, Lclk) for the RCG block. Look-ahead threshold (*s*) was implemented in the analog domain using two bias voltages to the ED while the look-ahead step size (*k*) is applied digitally. 10-bit SSLAR ADC was designed with 7-bit look-ahead step size control (N[0:6]).

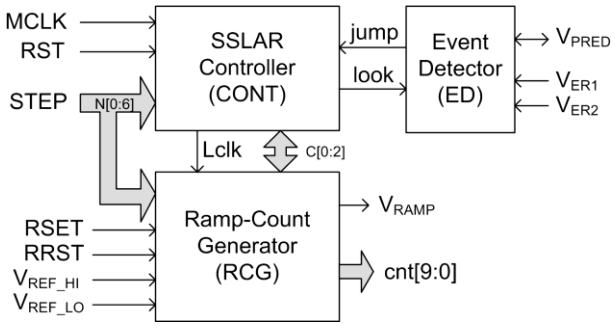


Figure 3.8. Global SSLAR ADC blocks.

Event Detector Design

ED block composes of a high-speed open-loop comparator and a CMOS switch, as shown in Figure 3.9. It is connected to column predictor capacitors and generates the *jump* signal after the *look* signal is asserted by CONT. When the *look* signal is high, the predictor bus (V_{PRED}) is set to $V_{ER1,}$ and the analog ramp signal is frozen while the ramp counter increments 1 or *k/2*-LSBs. During this time, none of the column comparator outputs changes its state. When the *look* signal is asserted low, the analog ramp signal rises 1 or *k*-step equivalent voltage causing *z* number of comparators to change their state from low to high. As a result, predictor bus voltage increases as given in (3-8).

$$\Delta V_{PRED} = \frac{z \cdot C_P \cdot V_{AA}}{(m - z) \cdot C_P + C_W} \tag{3-8}$$

where C_W is the parasitic capacitance on the predictor's bus. If ΔV_{PRED} is larger than the (V_{ER2}-V_{ER1}) difference, then the event detector pulls *jump* signal high confirming that the number of columns in the *k* range is less than *s* and lets the CONT block know that the jump operation can be approved. For a given *s*, event reference voltages have to be adjusted as given in (3-9).

$$V_{ER2} = V_{ER1} + \frac{s \cdot C_P \cdot V_{AA}}{(m - s) \cdot C_P + C_W} \tag{3-9}$$

In this design m=200, V_{AA}=3.3V, C_p=50fF, and C_W=400fF. For example, s=2 results in an event detector bias difference of 32mV, defining the required accuracy of the ED comparator.

SSLAR Controller Design

The SSLAR controller unit is the central part of the SSLAR ADC. It generates unique control signals for ED and RCG blocks implementing the SSLAR ADC algorithm. The internal making of the controller unit is shown in Figure 3.10. The finite state machine (FSM), shown in Figure 3.11,

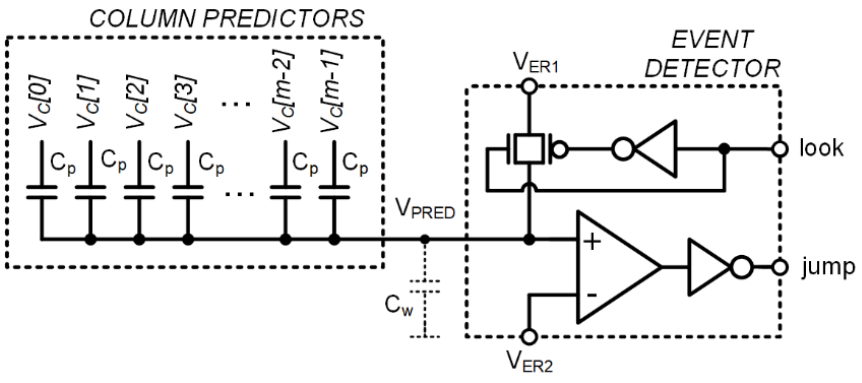


Figure 3.9. Event detector circuits.

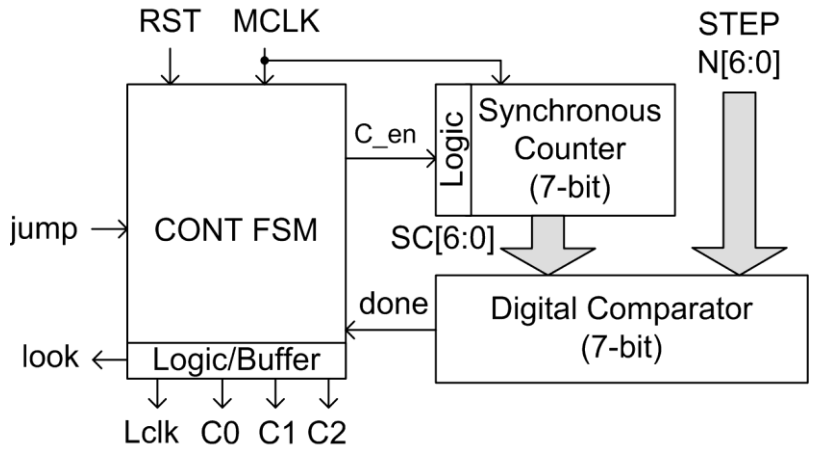
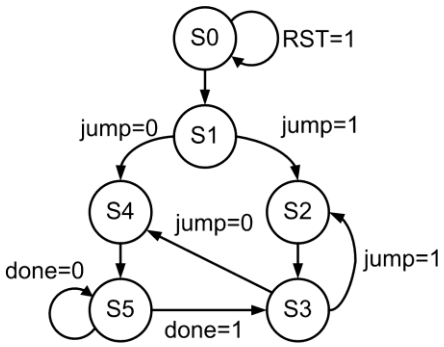


Figure 3.10. Block diagram of SSLAR ADC controller unit

implements the algorithm while 7-bit synchronous counter and digital comparators are used when jump operation is not approved, incrementing the analog and digital ramp 1-LSB at a time.

The controller FSM has six (6) states to generate five control signals, as listed on the table in Figure 3.11. Some of the signals are used directly by the ED and RCG units, while others are used internally. Four inputs define the state of FSM. These signals are; *jump* signal from the event detector, *done* signal from the internal counter, *MCLK* signal for the master clock, and *RST* signal for reset. The FSM changes its state at the rising edge of the MCLK signal conditionally or unconditionally. Unconditional state changes only exist from states S4 to S5 and from S2 to S3. Other state transitions depend on the *jump*, *done*, and RST inputs. If the RST is high, the state machine goes to state S0 and waits until the RST signal is cleared. *Done* signal is generated in the CONT unit by 7-bit synchronous counter and comparators. The operation of the counter/comparator combination is enabled by the



State	Look	C0	C1	C2	C_en	RST
Sx	1	0	1	0	0	1
S0	1	0	1	0	0	0
S1	0	1	0	1	0	0
S2	1	1	0	0	0	0
S3	0	1	0	1	0	0
S4	1	0	0	0	0	0
S5	1	0	1	0	1	0

Figure 3.11. Finite state machine diagram and the state assigned outputs of the SSLAR controller unit.

FSM through a counter enable signal (C_en). If C_en=1, then the counter starts counting while the comparator is checking whether the counter value (SC[6:0]) is equal to the look-ahead step (N[6:0]) or *k*. If they are equal, it asserts *done* signal to high for FSM to take action. If the *done* signal is high, then the counter enables the signal is de-asserted (C_en=0), changing the FSM state from S5 to S3. When C_en=0, the synchronous counter is reset to 0 and waits for C_en to be high again.

SSLAR Ramp-Counter Generator Design

The Block diagram of the RCG unit is shown in Figure 3.12. It generates an analog ramp signal (V_{RAMP}) and 10-bit digital counter outputs (Cnt[0:9]). The unit composes of two multiplexers; one carry look-ahead (CLA) full-adder, one full adder latches (FAL), two CLA subtractors, and one binary weighted charge scaling ramp generator. Look-ahead, jump, and fallback operations are controlled by adequately timing the blocks without requiring clocked synchronous counters in the RCG unit. The only clocked unit is the 10-bit FAL, whose clock (Lclk) is generated in the CONT unit asynchronously. 10-bit inputs (W_0[9:0]) to CLA full-adder are provided by the 4:1 multiplexer unit. Other inputs (W_2[9:0]) come from the 10-bit latch outputs, which change state at the rising edge of Lclk.

Depending on FSM's state in the CONT unit, one of four words is passed to the CLA unit through a 4-to-1 multiplexer using inputs C1 and C0. As a result, CLA full-adder works like a 10-bit counter (W_1[9:0]) without requiring a clock signal. It either stops counting for "11" control inputs or increments one LSB at a time for "10", or counts *k*-by-*k* (N[6:0]=*k*) for "01".

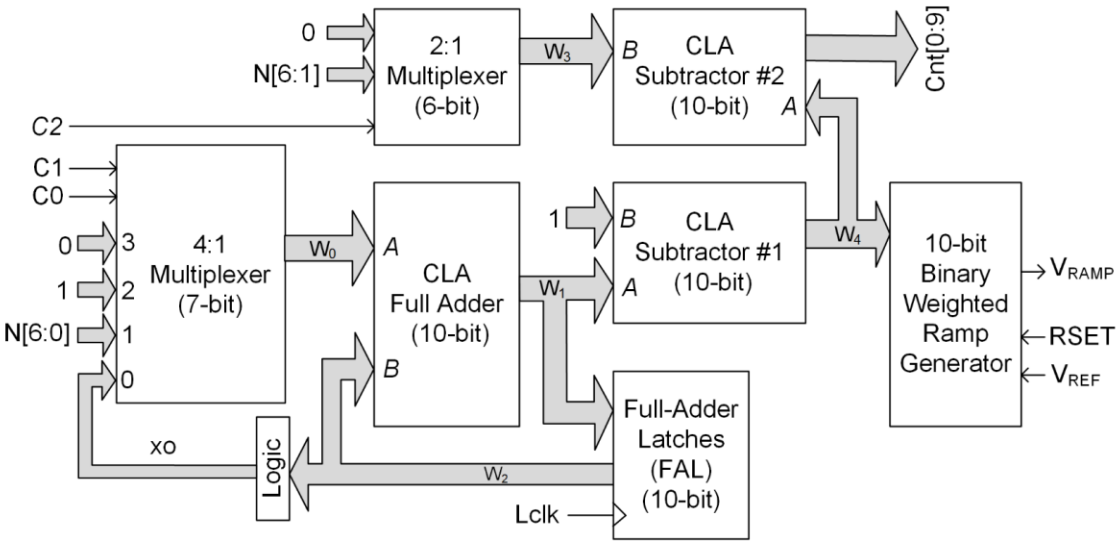


Figure 3.12. Block diagram of SSLAR ADC ramp-count generator unit

Carry look-ahead type adders were used for reducing glitches at the counter outputs, which was used directly by the binary-weighted ramp generator (BRG) block. CLA subtractor #1 subtracts 1 from the CLA full adder block outputs (W_1[9:0]). This allows digital bits used by BRG to be between 0 and 1023 for 10-bit. 2-to-1 multiplexer passes a half-step programming word or zero (0) to the subtractor #2. This way, half and full step counter increment operations of the algorithm are implemented.

Operation Modes

SSLAR ADC can work as a standard SSR ADC when the *jump* signal is driven externally and by setting both *k* and *s* to 1-LSB (*SSR-mode*). In this mode, ADC resolution can be reduced from 10-bits to 9, 8, or down to 2-bits by changing *k* to 2, 4, 8, or 128 LSB, respectively. Thus, SSLAR ADC architecture allows resolution change on the fly achieving high or low-resolution acquisition of predefined regions or rows of the pixel array, further improving frame rate as needed.

SSLAR ADC could be clocked to run at the same frame rate as it is in SSR mode by setting *k* and *s* and driving the *jump* signal from the ED block instead of externally. In this mode, acceleration would be achieved, and rows can be converter faster depending on the sampled row's code distribution and *k* and *s* values. If the same frame rate as in SSR-mode is needed or a fixed frame rate is set (i.e., 5 frames per second, fps), part of the analog and support electronics could be shut down (*SSLAR-mode-0*) during the saved periods reducing the power consumption. If a high frame rate is needed, these blocks left running and saved time periods could be used to increase the frame rate of the imager (*SSLAR-mode-1*). In later mode, the frame rate is increased with the expense of the overall power consumption.

The SSLAR based CMOS image sensor was designed with multiple levels of power-saving possibilities. Analog biasing signals were generated using on-chip current and voltage mode digital-to-analog converters (DAC), and a bandgap reference circuit is integrated. Power to these analog and mixed-signal blocks can be turned on or off through internal program registers accessed and set through a scan chain when a global power-down pin is asserted.

Measurement Results

The SSLAR ADC based CMOS image sensor was designed and fabricated in a 0.5µm, 2P3M CMOS process. Standard 3T CMOS APS pixel with a 15µm × 15µm pixel size was integrated with a 200 × 150 pixel array and 3.9mm × 4.1mm total die size. The micrograph of the fabricated chip is shown in Figure 3.13. Analog and digital power domains were separated in the chip but fed from a single 3.3V single supply on the PCB, allowing measurement of power consumption on each domain.

The SSLAR imager's several characteristics were measured using a custom-designed FPGA test board and program, as shown in Figure 3.13. An FPGA generated timing and control signals, and frames were transferred to a PC through a USB2 port. The master clock frequency for FPGA was 20MHz, while the ADC clock rate was set internally 4x smaller or to 5MHz to allow controlling signals to overlap correctly. Internal program registers of the imager are controlled through the user interface program running on the PC. The program in real-time analyzes captured images. Most measurements were performed in an isolated environment while a scene (a one-dollar US bill) is flood illuminated or a uniform light source/projector is used (Davidson TV Optoliner K-1000V).

Captured images in *SSR-mode* and *SSLAR-mode-1* are shown in Figure 3.14 for different *k* and *s* settings, which results in different frame rates, as shown in Figure 3.15. The imager captures 3.65 fps in *SSR-mode* and up to 22.5 fps in *SSLAR-mode-1* for k=64, s=32. Reduced integration time for high-frame rates is compensated by increasing the scene's illumination level; that average of captured images was set at about half-full scale level of 412-LSB. As shown in Figure 3.15, the frame rate could be increased more than six times (6x) by changing *k* and *s*. Measurements showed that an optimum k and s setting exists for higher frame rates in SSLAR-mode1 for the captured scene of Figure 3.14.

Full-chip power consumption of the imager running in *SSR-mode* and different settings in other SSLAR modes were measured as shown in Figure 3.16 Imager consumes 8.4mW full-chip power in *SSR-mode*, while this can be 4.6mW or 47% lower in *SSLAR-mode-0* with *s* between 2 and

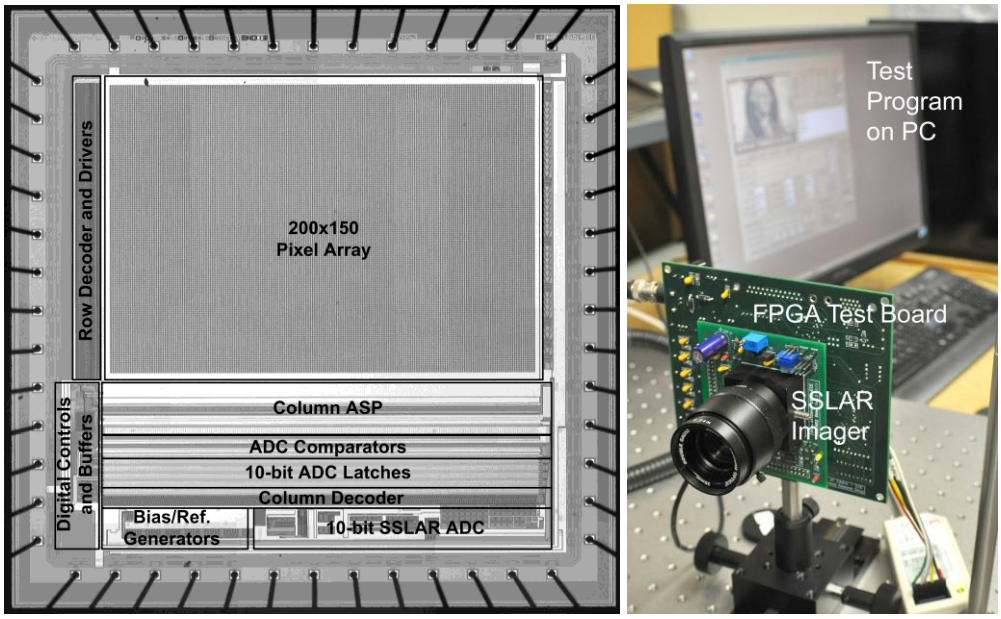


Figure 3.13. Die micrograph and test board of the CIS with 10-bit SSLAR ADC.

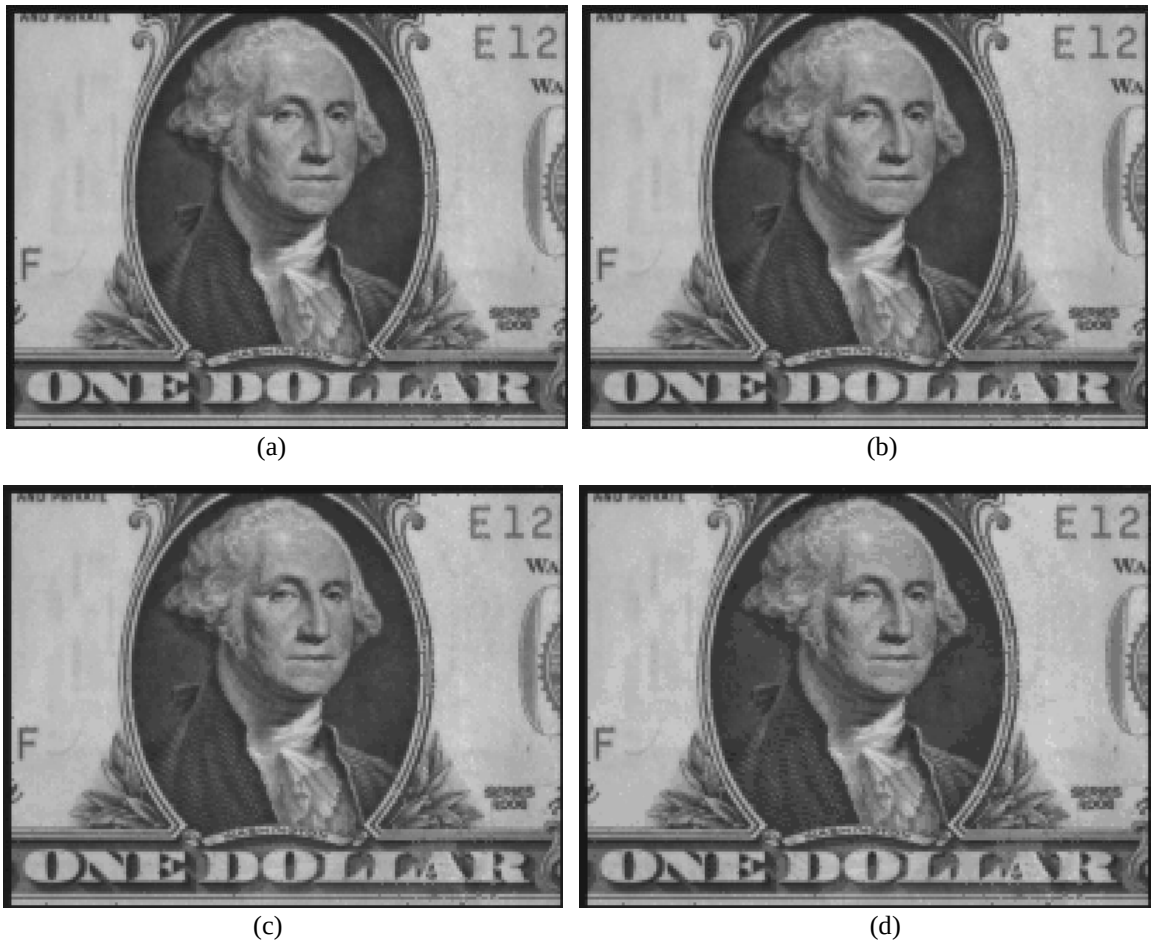


Figure 3.14. Images captured at different operation modes: (a) SSR at 3.65fps (b) SSLAR (mode1): k=16, s=32 at 16.5fps, (c) SSLAR (mode1): k=32, s=32 at 20.7fps (d) SSLAR (mode1): k=64, s=32 at 22.5fps.

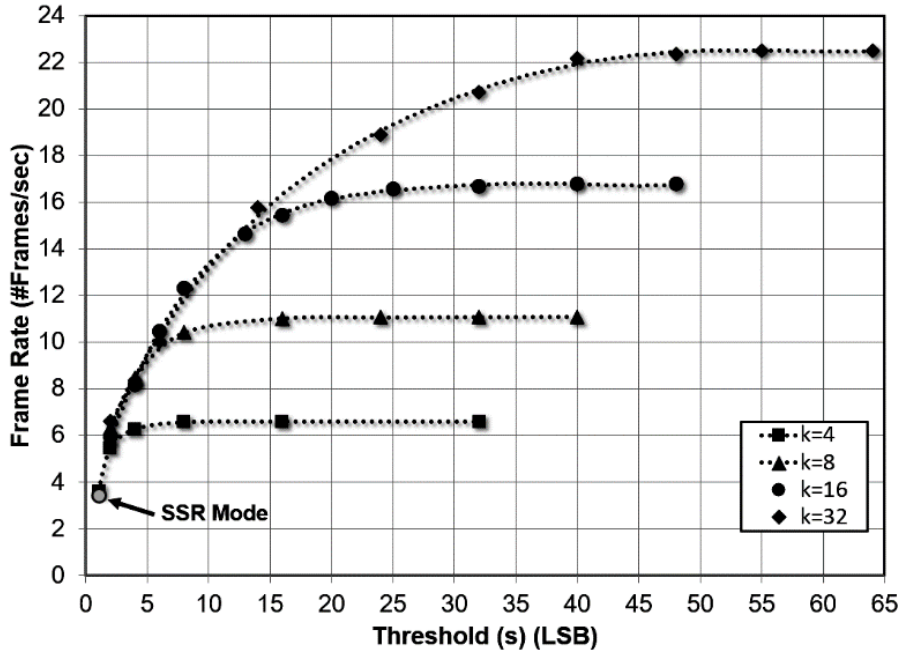


Figure 3.15. Measured frame rate versus threshold (s) for different step (k) sizes in *SSR-mode* and *SSLAR-mode-1* operations.

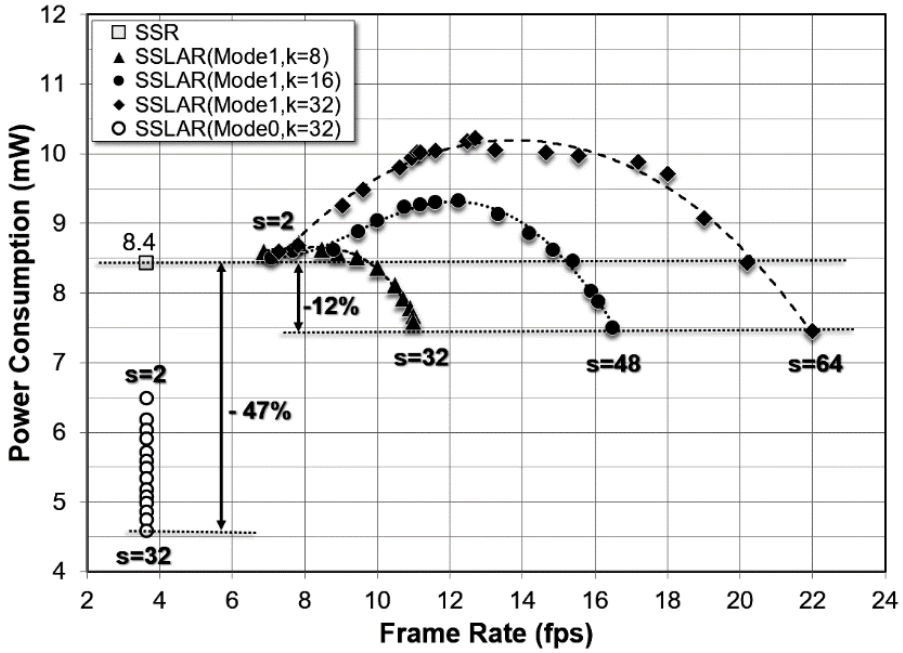


Figure 3.16. Measured full-chip power consumption of SSLAR image sensor in SSR and SSLAR modes and achieved frame rates.

32 and k=32 settings with which frame rate was fixed at 3.65 fps. In *SSLAR-mode-1*, power consumption tends to increase about 20% while the frame rates more than triples (3x) and ultimately drop below the SSR-mode level, achieving more than six times (6x) rate increase and more than 10% power reduction. What is more significant about this measurement is that the imager's power consumption is maintained at SSR-mode ranges while the frame rate is doubled or tripled for optimum step and threshold values. This shows that the proposed SSLAR algorithm not only improves the frame rate but also reduces the power consumption by skipping code ranges.

The ultimate goal of any mobile phone built-in cameras is to provide clear and intelligent information to the human visual system (HVS), which is very good at detecting any artifact produced by the imaging electronics or algorithms used in the video acquisition path. Thus, it is necessary to subjectively quantify if the SSLAR algorithm introduces perceivable artifacts on the captured images. This image quality (IQ) measure or metric has to be bounded, subjective, and ideally does not require multiple images to do the assessment. Today, the most commonly used IQ metrics for this purpose are the mean squared error (MSE) and peak signal to noise ratio (PSNR) for assessing picture quality/distortion for additive noise, even though they are known to be uncorrelated with the perceived quality by HVS. However, it was also reported that complicated IQ metrics based on human perception gain no clear advantage over simple mathematical measures such as PSNR [64]. Thus, MSE and PSNR were used for quantifying image distortion caused by the SSLAR algorithm.

Technically, MSE and PSNR measure image difference and fidelity, i.e., how closely an image resembles a reference image. Thus, it is necessary to capture two frames to calculate them. The MSE can be calculated by (3-10).

$$MSE = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left(x_{new}(m,n) - x_{ref}(m,n)\right)^2 \tag{3-10}$$

where M and N are the pixel array dimensions, $x_{ref}(m,n)$ is the reference frame, while $x_{new}(m,n)$ is the new frame with added noise. In this case, the reference frame is the image captured when the imager is working in *SSR-mode*, and the new frame is the one captured in *SSLAR-mode-0* or *SSLAR-mode-1*. Since the SSLAR imager could change the mode of operation seamlessly, it is possible to capture two consecutive frames in SSR and SSLAR modes. One difficulty is that the integration times will be different because the frame rates of SSLAR modes are much faster than that of the SSR mode for given threshold and step values. This is avoided by calculating the frame average at specific illumination levels (i.e., 412 LSB) for SSR-mode and increasing the illumination level until the frame average for SSLAR modes becomes the same with better than 0.5 LSB accuracy. Another issue is the

inherent temporal noise that varies from frame to frame for the SSR and SSLAR modes. This is reduced by averaging several consecutive images to generate reference and SSLAR frames to calculate MSE.

Figure 3.17 illustrates measured MSE versus frame rates for both *SSR-mode* and *SSLAR-mode-1* as LSB and percentage error when scene illumination was adjusted, and frame averages were 412LSB. *k* and *s* were the same data points as in Fig 15, and the image under test was the images in Fig. 14. Percent error was calculated using MSE in LSB and normalizing it with a full SSLAR ADC resolution-scale of 1024 LSB. As shown, *SSLAR-mode-1* introduces less than 0.7% or less than 7LSB error on a 200 × 150 pixel array with a 6X frame rate increase. The step size and the threshold are the key parameters for determining how the SSLAR speeds up and the output image quality. Choosing and optimizing these parameters depends on the maximum allowed MSE and the required speedup ratio. So, there is a trade-off between these parameters.

PSNR is also determined in decibel by using measured MSE data and by (3-11).

$$PSNR = 10 \cdot log_{10}\left(\frac{I^2}{MSE}\right) \tag{3-11}$$

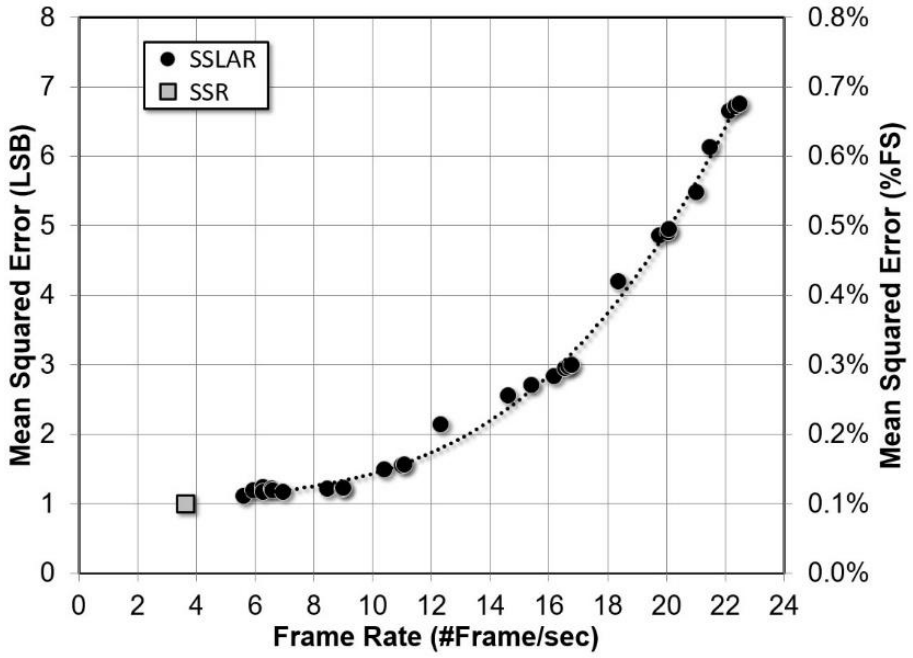


Figure 3.17. Measured MSE versus frame rate in SSR and SSLAR modes.

where I is the maximum value that a pixel could take (i.e., 1023 for 10-bit and 255 for an 8-bit image), measured PSNR versus frame rates for SSR, and SSLAR modes are shown in Figure 3.18, PSNR decreases as the frame rate increases.

Other pixel-based metrics were also used during measurements [65] [66] [67]. However, it was found that although some of these metrics can predict subjective ratings quite successfully for a given compression technique or type of distortion, they are not reliable in this case as they are not fulfilling the IQ metric requirements.

Conclusion

Although many ADC algorithms exist addressing speed issues of SSR ADC in CMOS image sensors, the proposed SSLAR ADC algorithm outperforms them on many fronts. Mainly, the proposed SSLAR algorithm provides speed improvement up to six times (6x) of a regular SSR ADC operation while reducing overall power consumption (12% to 47%) of the imager with minimal degrading image quality (MSE<0.7%). MSE image quality metric is used to assess the output image's degradation as it is the most commonly used metric. However, developing a new image quality metric will help the SSLAR algorithm to optimize its parameters like *s* and *k* for captured scenes. SSLAR algorithm also allows seamless mode change providing low (SSR mode) and high resolution (SSLAR modes) image capturing capabilities of regions or rows. The proposed ramp ADC algorithm achieves these advantages by opportunistically looking ahead on code ranges during ramp operation, predicting a number of pixels in the range, and skipping or scanning the range during ADC operation while maintaining high-resolution. To show these capabilities, a CMOS APS imager was designed in a

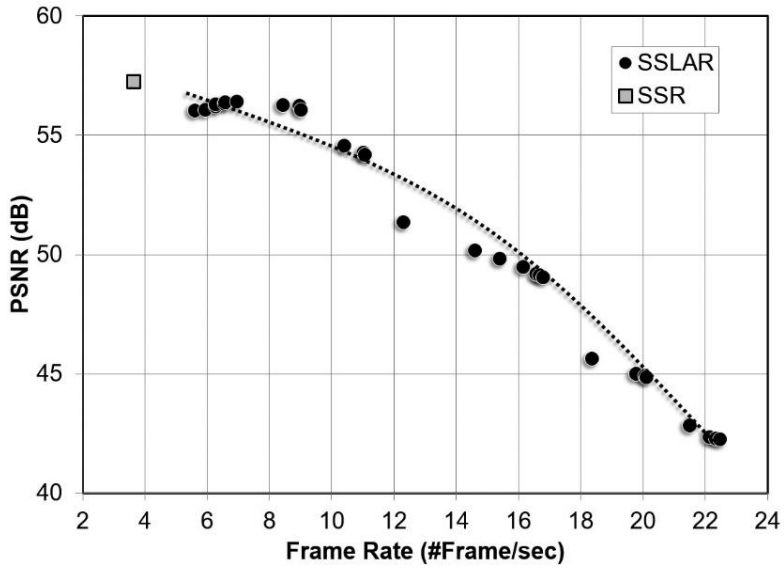


Figure 3.18. Measured PSNR versus frame rate in SSR and SSLAR modes.

0.5µm CMOS (2P3M) process. 10-bit SSLAR ADC was designed and integrated with column-parallel architecture along with 15µm × 15µm 3T APS image sensor pixels in a 200 × 150 array.

Chapter 4: Conversion Complexity Metric (CCM) for CIS

"A New blind image conversion complexity metric for intelligent CMOS image sensors."

Forthcoming in IET Image Processing Journal, 2020.

Many algorithms have been developed for complementary metal-oxide-semiconductor (CMOS) image sensors to speed-up analog-to-digital (A-to-D) conversion of captured images. However, there is no objective blind-image quality metric to compare and quantify the quality and effectiveness of these speed-up algorithms. In this work, I developed a blind -image quality and complexity metric for this purpose. The proposed metric relies on counting the successive zeros in a code histogram (CH). The proposed metric is called the conversion complexity metric (CCM). The CCM is designed to quantify how complex and to predict how much time and power-consuming a captured image is for A-to-D conversion, mainly by integrating (ramp) type analog to digital converter (ADC) used in column-parallel architectures (CPA) of a CMOS image sensor (CIS). The proposed metric, CCM, is tested for linearity, monotonicity, and sensitivity to many types of introduced distortion. The CCM is compared with other no-reference and full-reference image quality and complexity metrics. It accomplished, for brightness, change distortion, 99% linearity, and 316 % sensitivity, providing a computationally efficient blind-image quality metric that no other metrics provide for CIS to adjust and optimize on-chip analog and digital signal processing intelligently.

Introduction

Today, a massive number of people carry mobile phones that have built-in cameras. Millions of images are captured every minute by these cameras, and this number is increasing yearly [68]. Hence, the image sensor and camera markets are huge and very competitive. Cameras are expected to capture high-quality and high-resolution images while intelligently minimizing power consumption, especially in battery-powered mobile devices and in future internet of things (IoT) devices. Besides, cameras should respond and dynamically adapt to different scene conditions providing a wide dynamic range operation capability. Nowadays, mainly CISs are at the core of these cameras, in which their performance determines the quality and the value of them. CIS captures images in the analog domain in the form of voltage, current, or charge and perform some analog signal processing (ASP) before converting them into the digital domain by on-chip ADCs for further digital image processing. All on-chip analog or digital signal processing and conversion operations have to be optimized for low-power and high-speed performance. Many algorithms were proposed to speed up these operations [69] [61] [36] [34]. However, there is no objective image

quality (IQ) metric to compare the effectiveness or quality of these algorithms proposed and used. The IQ metric should consider the complexity of the images captured and the difficulty of analog and digital processing and conversion and provide intelligent information to processing and conversion algorithms to optimize conversion quality, speed, and power consumption of the image sensor in both analog and digital domains. It should also be computationally fast and straightforward to be implemented and should not require large analog or digital storage capacity (i.e., two frames digital memory).

Image capture, conversion, and processing techniques introduce undesirable effects to digital images, causing a loss of quality and important information. These may occur during any stage(s) of image reproduction processes like image capture, conversion, processing, compression, transmission, storage, or retrieval. Hence, assessing image properties (mainly IQ) becomes paramount to be able to judge the effects of the aforementioned processes. Fundamentally, there are two methods [70] available to evaluate and to compare image quality, which are; subjective methods [71] [66] and objective methods [72] [73]. The subjective methods are built on human perception quantifying image properties with a scale based on a human observer's judgment, while the objective methods are based on obvious numerical and mathematical calculations of image parameters. Both of these IQ assessment methods are trying to correlate the index assessed by the human visual system (HVS) to an index obtained from mathematical calculations. Until today, over 100 metrics have been developed in both methods to assess IQ, as summarized by Pedersen [74] and Marius [75]. These metrics' common purpose is to quantify the distortion that may occur, IQ monitoring, process optimization, benchmark production, and problem area identification [76]. Some IQ metrics are designed for colored images, while others are for greyscale images. There are many ways to classify these metrics based on the way and purpose that they have been developed.

Researchers have classified IQ metrics in different ways [76] Avcibas [66] classified IQ metrics into six groups according to the information that image carries: (1) pixel difference-based distortion measure like mean-square error (MSE), (2) correlation-based measure, (3) edge-based measure such as edge position displacement and its consistency for different resolution levels, (4) spectral distance-based measure, (5) context-based measure, and (6) HVS-based measure which are either dependent on (dis)similarity criteria used in image-based browsing functions. Callet and Barba [77] categorized IQ metrics into two different groups. The first group is IQ metrics, which utilize the HVS model as a low-level perception like masking effect and sub-band decomposition to compute a distortion map. The second metric group uses little information from the HVS model to represent errors and form a prior knowledge for introduced distortions. Chandler and Hemami [78] classified

IQ metrics into three categories. First, metrics that can be obtained mathematically that is, depend on distortion intensity. Second, near-threshold psychophysics-based metrics in which the visual detectability of distortion is considered. Third, overarching principles-based metrics that extract information or structure. Wang and Bovik [79] divided IQ metrics into three groups. The first group is based on the use of a reference image, which is: no-reference, full-reference, and reduced-reference based IQ metrics. Further, Thung and Raveendran [80] sub-divided full-reference IQ metrics into three groups as mathematical, HVS-based, and other. The second group is based on the coverage of the IQ metrics as application-specific or general-purpose. Third, how the IQ metric is structured either bottom-up or top-down. These three groups and their IQ metrics use the original image, a distortion process, and HVS knowledge. Pedersen and Hardeberg [76] had also proposed to divide IQ metrics into four groups. First is mathematically based metrics that use distortion intensity such as MSE and peak signal-to-noise ratio (PSNR). Second is low-level metrics that depend on distortion recognition using contrast sensitivity functions (CSFs), which is used in spatial-CIELAB (S-CIELAB), [81]. The third is high-level metrics such as structural similarity metric (SSIM) [82], which depends on structural content or the visual image fidelity (VIF) [83]. This group of metrics depends on the statistical properties of the captured scenes. Forth are other metrics that mix two or more strategies of the above groups such as visual signal-to-noise ratio (VSNR) [78] which considers both low- and mid-level scene properties and utilizes mathematical models to get the IQ metric score in the final stage.

In summary, IQ metrics are classified in literature based on the fundamental answers to the following questions: Does the given metric require single (no-reference) or multiple images (full- or reduced-reference) to compare with? Is the metric objective or subjective? If it is subjective, does the metric provide a low or high target perception level? If it is objective, what is the image information (pixel, row, edge, window, etc.) used by the metric? Choosing an IQ metric will be a simple task if all these questions are answered fully. However, each application has its unique requirements that none of the existing IQ metrics may fulfill all.

The metric should be reference independent (single-image), objective, and use image information row-by-row for intelligent CIS operations. The metric index should also be bounded (i.e., between 0 and 1) and computationally simple to implement on-chip intelligent operations and learn to adjust CIS electronics' analog and digital characteristics.

IQ Metric for CIS Electronics

The classifications mentioned in the previous section are based on an area of applications and points of view of the importance of specific requirements. Among all IQ metrics mentioned in

[76], except for several of them that will be investigated in the following sub-sections, none of the metrics can be used for quantifying the performance requirements of image processing electronics to convert an image from the analog to the digital domain. The IQ metrics MSE [67], SSIM [82], histogram flatness measure (HFM) [84], histogram spread (HS) [84], blind/reference-less image spatial quality evaluator (BRISQUE) [85], natural image quality evaluator (NIQE) [86] and perception-based image quality evaluator (PIQUE) [87] are checked if they are suitable for intelligent and evaluation of CIS.

Mean Square Error (MSE)

MSE is one of the oldest and most widely used IQ metrics [79] because of its computational simplicity, ease of analytical tractability, clear physical meaning, and it is mathematically convenient in the context of optimization [82]. Its index is calculated by adding and averaging the squared difference between the original and processed image element values [76]. Thus, MSE is an objective IQ metric that needs two images (reference and processed) and uses pixel information for calculations. However, MSE has a drawback that it is not correlated to perceived images by HVS [78] [79] [88] [64]. Indeed, some images that have the same MSE perceived very differently from each other, as outlined in [82] and in [79]. Furthermore, MSE does not satisfy the aforementioned metric requirements for intelligent CIS as it needs two images to quantify IQ, and its index is not bounded a frame-based.

Structural Similarity Metric (SSIM)

SSIM is one of the most popular IQ metrics due to the drawbacks of MSE [82]. It is built on the universal IQ metric (UIQ) index [89] and takes into account the HVS model to overcome issues MSE possesses [70]. It quantifies IQ using a combination of luminance, contrast, and structure comparisons of the original and processed images [90]. These comparisons are performed for local image windows, and then the SSIM metric is calculated as the mean of all these local windows. The SSIM index is bounded between zero and one. It is a symmetric metric and has a unique maximum. However, the SSIM index cannot be used for intelligent CIS because it needs a reference image and uses the window of pixels for calculations. It is more suitable for human perception assessment of still image frames not for local assessments (i.e., row-based) to assist intelligent image capture and digitization processes in CIS.

Histogram Flatness Measure (HFM) and Histogram Spread (HS)

IQ metrics that use histogram distribution like HFM and HS [84] have the potential to be used in the intelligent CIS. They quantify the contrast level of images using full-frame image histograms, which makes these metric references independent, or single image metrics. The HFM

index is calculated as the ratio between the geometric mean and the arithmetic mean of image histogram intensities. Hence, its index is bounded between 0 and 1 because the geometric mean always less than or equal to the arithmetic mean of the same data set. As a result, HFM has a low index value for low contrast images that has a narrow and peaky histogram and vice versa. HS is also a single image metric that uses the histogram of the full image frame. Its index is calculated by determining the ratio of the quartile distance (the difference between the 3rd quartile and the 1st quartile of a cumulative histogram) to the full range of the image histogram. Thus, the HS index is bounded between 0 to 0.5 for uniformly distributed images and 0 to 1.0 for binary images. The HS index is low for narrow and peaky image histograms, which has low contrast images, and vice versa. Although the HFM and the HS are image reference independent and objective metrics that use image histogram information, they are not fully suitable for the intelligent CIS. They depend on whole frame histograms and computationally expensive.

Blind/Reference-less Image Spatial Quality Evaluator (BRISQUE)

Anish Mittal has proposed the BRISQUE metric [85] for the blind IQ assessment of natural scenes based on statistics and training models. It is designed to quantify a lack of naturalness of an image due to any distortion that may present in it. The BRISQUE generates an index value by a model called support vector regression (SVR), which is trained on an image database with Differential Mean Opinion Score (DMOS) values. Images in this database have well-known distortions for natural images, and hence the BRISQUE index is limited to evaluating IQ for the same type of distortions only. Hence, it cannot compute specific distortion features like blurring and blocking. Although the BRISQUE is a single image metric, it is an unbounded metric, and its index computational complexity makes it unsuitable for CIS intelligent electronics.

Natural Image Quality Evaluator (NIQE)

NIQE was proposed by Mittal [86] to solve issues surrounding BRISQUE. It uses a model that is not trained on human-rated distorted images or even exposed to any distorted image. Thus, it is considered a completely blind metric. NIQE model uses measurable deviations from natural image regularities. This model is built on quality-aware collections derived from statistical features of a simple and successful space domain Natural Scene Statistic (NSS) model. These statistical features are based on undistorted and natural images. NIQE is developed by fitting quality-aware features to a multivariate Gaussian (MVG) model and measure the distance between this MVG model and the NSS feature extracted from the test image. This measured distance has no limit, so NIQE is an unbounded metric. Despite NIQE being a no-reference metric, it is not applicable for CIS intelligent

electronics because it is unbounded and derived from a distortion-aware training model that does not consider how images will be processed through CIS electronics.

Perception-based Image Quality Evaluator (PIQUE)

Venkatanath proposed PIQUE [87] to quantify the distortion of an image without the need for any training data, so it is an opinion-unaware metric. PIQUE predicts image quality based on extracting image local features that help to create a fine-grained block-level distortion map. This approach does not need any statistical learning data for IQ assessment, but it is based on a test image local block/patch level characteristic. Each local block level has a size of n × n. PIQUE methodology classifies the given block into a distorted or not distorted block and assigns a score for each block where it then calculates the PIQUE overall score. Thus, PIQUE is an unbounded metric designed for specific applications like feature point extraction, object detection, and compression, not for CIS intelligent electronics.

In summary, none of the existing IQ metrics are suitable for assessing captured image quality or complexity to assist the intelligent operation of CIS. A new metric is proposed to close this gap. It is a histogram-based, no-reference, an objective metric that is called Conversion Complexity Metric based on a successive zeros histogram, as explained in the next section.

Conversion Complexity Metric (CCM)

Since there is no clear definition for what "image complexity" is [91], quantifying it is not an easy undertaking [92]. Thus, I defined image complexity from an A-to-D conversion complexity point of view considering how difficult it is (simple/complex timing, low/high power consumption, low/high speed, small/large silicon footprint, etc.) for image sensor electronics to convert a captured image from analog to digital domain. This is very critical to judge and compare between different analog and digital processing algorithms and choose which algorithm is better for specific applications to intelligently maximize or predict the performance and efficiency of sensor electronics. As mentioned before, it is necessary for the metric to be reference independent (single image), objective, bounded, and row-by-row based. I will start by illustrating the electronic system requirements and describing the new philosophy of the proposed metric afterward where some examples and discussion will follow.

IQ and Image Complexity Metric Requirements

After a CIS captures a scene image, it passes through a long chain of electronics, including ASP(s), ADC(s), and finally, a digital signal processor (DSP). These blocks are integrated into different orders, forms, and shapes by different CIS architectures. Four fundamental architectures

can be constructed; column-parallel architecture, column-series architecture (CSA), pixel-parallel architecture (PPA), and pixel-series architecture (PSA). The captured image data in the form of voltage, current, or charge can be converted and processed in digital form in different stages and locations of these architectures. It may be processed and converted in each pixel in parallel in PPA, or out of the pixels sequentially in PSA, or row-by-row sequentially in CSA, or row-by-row in parallel in CPA [40]. Today, CPAs are widely used in CIS to satisfy the high frame rate market-driven requirements, large pixel arrays, smaller pixels, and low-power consumption. Integrating (ramp) ADC topologies are the best fit for CPAs as they provide the best monotonic, low-noise, and power-efficient conversion capabilities with smaller silicon footprints [36] [18] [93].

Intelligent image sensor operation requires an image complexity metric that relates captured or incoming image pixel, row, or frame information to readout electronics' performance parameters in the sensor architecture. I proposed a metric that is designed for the most commonly used CIS architecture, the CPA. In CPA, the image is processed row-by-row, and each column has its own or shared ASP and ADC. The row decoder selects one row at a time, and analog signals from this selected row of pixels are transferred to column ASPs for pre-processing in the analog domain. Then, these signals are transferred to ADCs, and finally, digitized pixel data is transferred to DSP or stored in memory for further digital processing. Each column level circuits read the next row of pixel data in a pipelined fashion. The selected row is reset before the next row is accessed, allowing rolling-shutter frame integration operation.

CIS readout electronics' performance efficiency in different image sensor architectures is affected by the complexity of the scene image. For example, in CSA, if an image has well-distributed grey levels or a wide histogram (I consider them a complex image), its ASP stage consumes more power than when the image or row has a narrow code distribution or a peaky histogram (I consider them less complex). For the same scenario, ADCs in CPA work more power efficiently for non-complex images than complex ones. Digital power consumed during the data transfer period from ADC to DSP in both architectures would be lower for non-complex images. Suppose a metric is available for a row or full-frame image complexity. In that case, it will be easy to gear up and down the biasing conditions in ASPs, or ADCs, or regulation efficiencies in on-chip voltage regulators, or other parts of the readout chain that could be tuned for optimum and uniform performance of the CIS. One such example is shown in Figure 4.1, where a multi-mode ramp ADC [1] is used in a CIS with CPA, and full-chip power consumption was measured from the imager in [1] at 20 frames per second (FPS) for 1 minute, capturing the same scene images. Mode-1 is the standard ADC mode where the ADC works normally without any speedup process, so it has the

maximum power consumption as it works for the longest time. On the other hand, different speedup techniques can be applied to minimize the ADC conversion time, resulting in minimizing the power consumption by shutting down or idling circuit blocks during the saved times. Modes 2 and 3 represent the power consumption after applying two of these speedup techniques. However, mode-3 is a higher speedup mode than mode-2 and shorter ADC period. Thus, the power consumption of mode-3 is lower than mode-2. This speedup comes at the expense of image quality. Thus, based on the new CCM index, the speedup mode can be determined intelligently. As the scene content and complexity change from frame to frame, the power consumption of the CIS can be reduced up to 10%.

The CCM Algorithm

Images stored in digital mediums consist of a two-dimensional (2D) array of rows and columns with specific digital numbers representing the intensity of the scene pixels captured by image sensors that have ADCs with a resolution of n-bits. Thus, each pixel holds a digital number or code in the range between 0 and $(2^n - 1)$ if binary representation is used. However, in a typical scene image, not all codes appear in each row or frame, which could be seen in the histogram distribution of codes, as shown in Figure 4.2. The evaluation of these CHs would reveal the

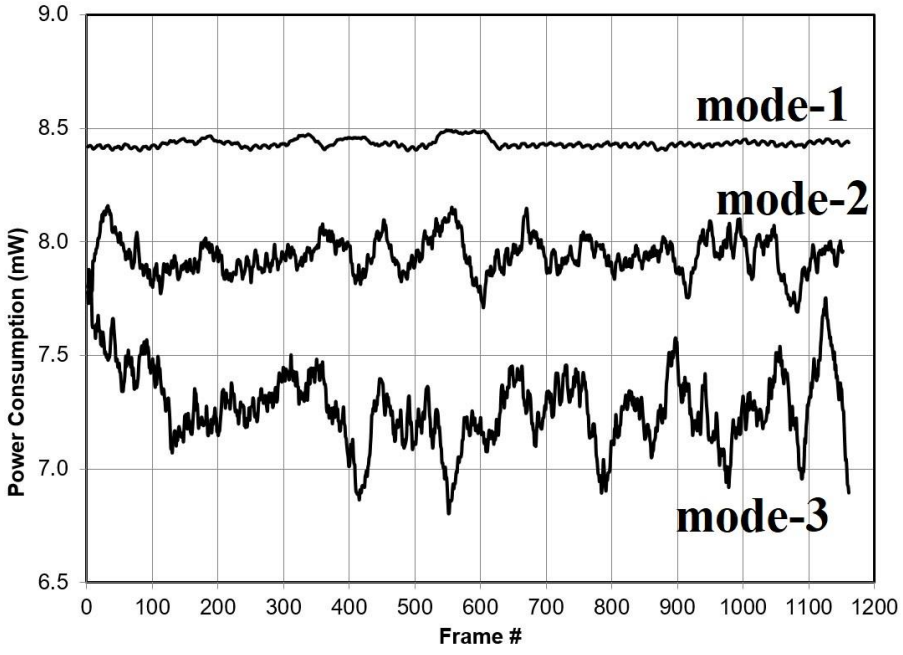


Figure 4.1. Measured full-chip power consumption of a CIS with CPA and integrated ramp-ADC in [35] for different modes.(a) mode-1 for normal ADC operation (b) mode-2 is medium ADC speedup operation (c) mode-3 is the maximum ADC speedup operation.

Figure 4.2. Code histograms of different type of rows of an image.

fundamental characteristics of images, rows, or columns, i.e., whether the selected row is bright, dark, or well-distributed.

CH is constructed by counting the number of pixels in the selected region (row(s), column(s), sub- or full-area(s) of image) that have the same code values and assigning this number as a hit for the given code. For example, a dark region will have a CH that has all the hits appear in lower code ranges, and the rest of the range will have no hits (zero), and vice versa for bright regions, as seen in Figure 4.2. If the number of pixels (N) in the selected region is less than the code range or resolution of the image ($2^n - 1$), then some number of zero hits will appear in the CH of the selected region. Depending on the distribution of the code hits in the histogram, the number of successive zero code hits will vary. For example, all pixels will have a 0 code as a binary number if the image is completely dark, and as a result, code zero (0) will be repeated for N number of hits. The rest of the codes (2^n) will have zero hits or will have one 2^n number of successive zeros in the CH. This is the same for a wholly saturated or bright image that the last code number ($2^n - 1$) will have N number of hits and the rest of the codes (2^n) will have zero hits resulting in one 2^n number of successive zeros in the CH. While on the other hand, for the graded greyscale region, code hits are well distributed, minimizing the number of successive zeros. The proposed metric is based on the number of successive zeros in a CH for the image row region.

As the number of successive zeros in the CH of a row increases, the analog signals become easier and faster to be converted into the digital domain by row-level ADCs in CIS CPA. That is because if a code histogram has a large number of successive zeros, the ADC can skip scanning all

codes that do not appear in the CH and save the conversion time for these absent codes resulting in faster operation and lower power consumption [1]. The opposite is true if there are few or no zeros appear in the CH, which means the pixel values are well distributed in this row, and ADCs will consume more power to convert the analog content of the rows into the digital domain. Indeed, a metric based on the successive zeros in CHs will provide valuable information not only for the electronics but also for assessing the complexity of the images' regions (row, column, area, of the full image). Based on these understandings, I call the new metric the Conversion Complexity Metric because it relies on the successive zeros in the CHs of a given region of the image.

Because CCM is a histogram-based metric, it does not need any reference image and uses simple mathematical processes to count the number of successive zeros in the CH and calculate the average image conversion complexity. Although it was developed for assessing the complexity of captured image rows in CMOS CPA imagers, it can also be used as a metric to quantify the quality or complexity of any regions (i.e., sub-image area, full image, row(s), column(s)) of a single image. Its index does not depend on human evaluation and uncertainty. As presented in the next sections, the CCM index is bounded between zero (0) and one (1) and inversely proportional to the image conversion complexity, i.e., uniformly distributed CH of a row have minimal CCM index (difficult to convert) and ideally have an index of zero. In contrast, very dark or very bright (easy to convert) rows have a CCM index of near unity.

The CCM index value of an image row is calculated as follows:

1- Start reading rows from the image or pixel array (i=1),
2- Read row(i) from image sensor array or image file,
   a. Construct CH of a row(i),
   b. Construct successive zero histogram (SZH) from the CH of the row(i),
   c. Calculate $CCM_{row}(i)$ index for row(i) using equation (4-1),
   d. Increment row address (i=i+1),
3- Go to step-1 until all rows ($i=n_{row}+1$) are read, otherwise go to step-4,
4- Calculate CCM_{image} index for the whole image using equation (4-2)
5- Calculate the final CCM index by using the mapping function (F) in equations (4-3) and (4-8).

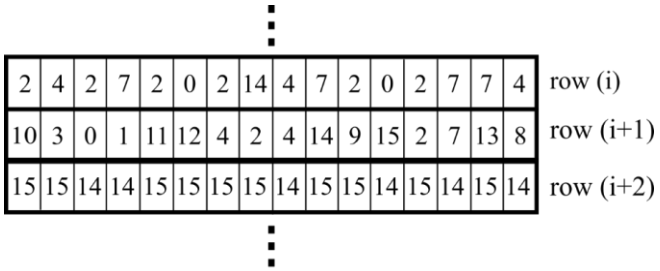
$$CCM_{row}(i) = \frac{\sum_{j=1}^{2^n} \frac{j}{Z_j}}{2^n} \tag{4-1}$$

$$CCM_{image} = \frac{\sum_{i=1}^{n_{row}} CCM_{row}(i)}{n_{row}} \quad (4\text{-}2)$$

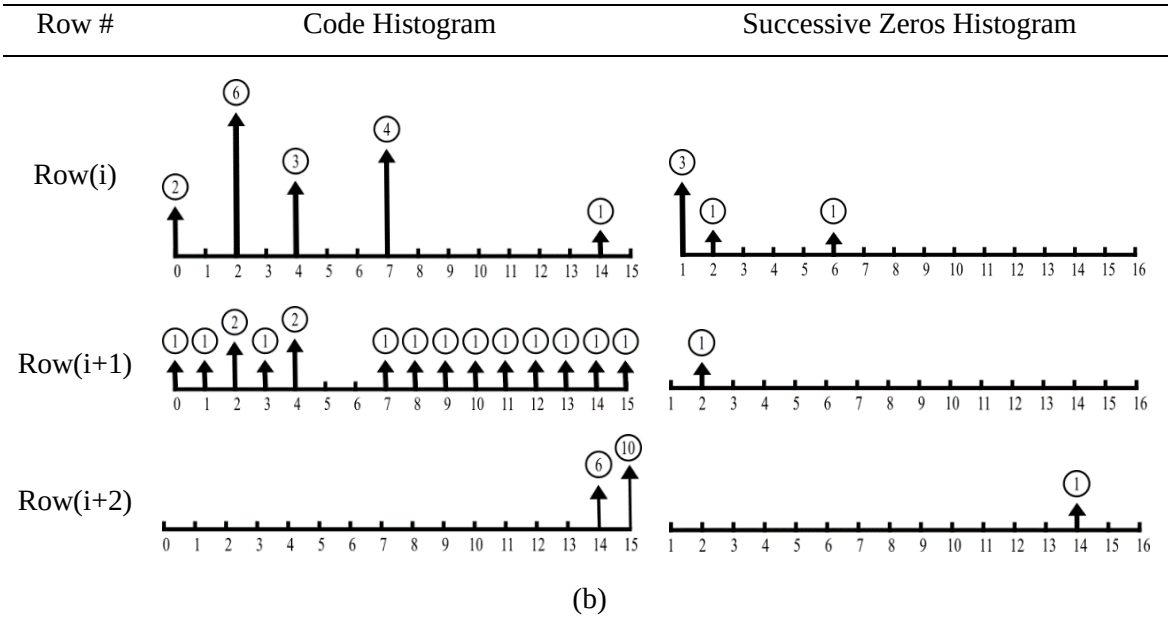
$$CCM = F\{CCM_{image}\} = \frac{CCM_{image} - CCM_{min}}{CCM_{max} - CCM_{min}} \quad (4\text{-}3)$$

where j is the index of SZH representing the number of successive zeros ($j \neq 0$), Z_j represents the repetition of the successive zeros of the (j) number. n is the ADC resolution, and n_{row} is the total number of rows of an image. CCM_{min} and CCM_{max} are the minimum and maximum theoretical limits of the CCM index, respectively, and they are derived in the following sub-section.

Figure 4.3 illustrates a simple example of how CCM works and its index is calculated. Assume a row of an image (row(i)) that has 16 columns (n_{col}=16) with 4-bits resolution (n=4). Thus, the code range is between 0 and 15. Figure 4.3.a shows three rows of such data, of which values lie in the code range 0 to $(2^n - 1)$. Figure 4.3.b shows the CH and SZH of each row. For example, in the row(i); 6 columns (or pixels) have a binary code of 2, 4 pixels have a code of 7, 3 pixels have a



(a)



(b)

Figure 4.3. CCM index calculation example (a) sample of row image data (b) code histogram and successive zero histogram of the rows' example

code of 4, 2 pixels have a code of 0, and 1 pixel has code of 14. So, except for these five codes, the rest of the code did not appear in a row(i). The CH for the row(i) could be constructed using these values, as shown in Figure 4.3.b. Using the CH, SZH for the row(i) can be constructed by counting the successive zeros (j) of the CH. For example, there are six (6) successive zeros exist one (1) time between codes 7 and 14 in CH, two(2) successive zeros appear one (1) time between codes 4 and 7 in CH, and a single (1) zero exists three(3) times as seen in the CH of row(i). Thus, the Zj becomes {3,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0} for j between 1 and 16. Then, the CCM_{row} (i) index for the row(i) is calculated from equation (4-1) as;

$$CCM_{row}(i) = \frac{\frac{1}{3} + \frac{2}{1} + \frac{6}{1}}{2^4} = 0.52 \tag{4-4}$$

Finally, if I have the data for the rest of the image rows, the average of all rows (CCM_{image}) will be calculated according to equation (4-3) for the entire image. The CCM_{row} index of 0.52 for row(i) indicates that this row has a moderate conversion complexity as it contains a relatively large number of successive zeros and the small number of single and double successive zeros. A row becomes complex for processing and conversion electronics if all grey level codes appear in its CH, as in row(i+1) in Figure 4.3, which gives the CCM index value of 0.125. It becomes less complex if all pixels have the same value, as in row(i+2) in Figure 4.3, which increases the number of successive zeros and results in a larger CCM index value of 0.875.

The simple example in Figure 4.3 shows the proposed algorithm to calculate the CCM index value and how to apply it for given image data. However, the theoretical limits of the proposed metric need to be well understood, derived and calculated to make sure its value is bounded. Besides, different combinations of the number of columns (n_{col}) and the bit resolution (n) of the images or ADCs should be considered to create more generic image quality and complexity metrics. This is explained in the next sub-section, which results in a mapping function.

Theoretical Limits of The CCM Index

The CH distribution of natural images is relatively random, as some of them may be dark themed images while others may be brighter or well distributed. Because of this variability, theoretical limits should be established, and a mapping function is defined to keep the index values bounded. Moreover, to generalize the CCM as an image quality and complexity metric, the relationship between the number of image elements and the bit resolution (n) of the ADCs in image sensors should be considered. Image elements could be selected as a sub-array of pixels, or in this case, the number of columns (n_{col}) of image pixels of the selected row sampled in the column electronics of CPA CISs

If the number of columns (n_{col}) (or number of pixels in the full or sub-array of an image) is less than the code range (2^n), not all codes will appear in CHs. As a result, at least ($2^n - n_{col}$) number of extra zeros will appear in row CH. Using these extra zeros during the calculation of the CCM index may result in saturation. They are considered and counted because, fortunately, these extra zeros help ADCs to convert these signals faster. The CCM upper and lower limits for this case will be investigated first.

The upper theoretical limit of the CCM index (ideally 1) for ($n_{col} < 2^n$) occurs when all column data in the selected row are zero (i.e., totally dark row of an image). In this case, only one code appears in the CH at code index 0. The number of occurrences or hits of code 0 equals the number of columns (n_{col}), and the rest of the histogram indices will have zero hits, as shown in Figure 4.4.a The number of successive zeros (j) in this case is equal to ($2^n - 1$) and appear one time (Z_j=1) as illustrated in the SZH in Figure 4.4.b. So, the maximum CCM index value can be calculated using equation (4-1) as;

$$CCM_{max} = \frac{(2^n - 1)}{\frac{1}{2^n}} = 1 - \frac{1}{2^n} \tag{4-5}$$

The lower theoretical limit of the CCM (ideally 0) can be calculated for ($n_{col} < 2^n$) when a CH is well distributed, as in Figure 4.4.c. In this case, each code will appear only one time in the

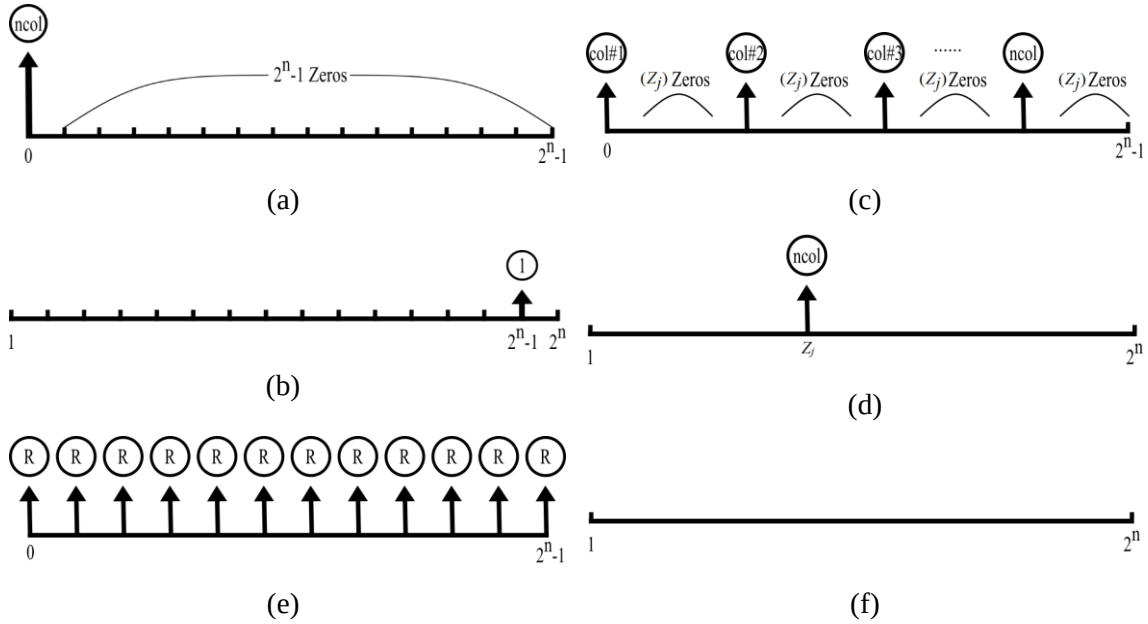


Figure 4.4. Theoretical limit cases for CCM index calculation (a) CH for upper theoretical limits for n_{col} < or ≥ 2^n (b) SZH of CH upper limit for n_{col} < or ≥ 2^n (c) CH for lower theoretical limits for $n_{col} < 2^n$ (d) SZH of CH lower limit for $n_{col} < 2^n$ (e) CH for lower theoretical limits for $n_{col} \geq 2^n$ (f) SZH of lower limit for $n_{col} \geq 2^n$

CH. However, because n_{col} is less than code range (2^n), there will be extra zeros between CH indices. In the worst-case these extra successive zeros (j) can be calculated as;

$$j = \frac{2^n}{n_{col}} - 1 \tag{4-6}$$

and the (j) number will be repeated for n_{col} times so, $Z_j = n_{col}$. The SZH is obtained from row CH as illustrated in Figure 4.4.d, and the theoretical lower limit of the CCM index is calculated using equation (4-1), as:

$$CCM_{min} = \frac{\frac{j}{Z_j}}{2^n} = \frac{\frac{2^n}{n_{col}} - 1}{2^n \cdot n_{col}} = \frac{2^n - n_{col}}{2^n \cdot n_{col}^2} \tag{4-7}$$

There is another case that needs to be investigated; that is when the number of columns (n_{col}) is larger than or equal to the code range (2^n) for the n-bits resolution of the ADC. The upper theoretical CCM limit when ($n_{col} \geq 2^n$) is the same as the case when ($n_{col} < 2^n$), as explained before and as shown in Figure 4.4.a and 4.b. It happened when all the columns in a row have zero values, and it is calculated from equation (4-5) as well. For the lower theoretical limit when ($n_{col} \geq 2^n$), in the worst case, all codes in the CH have at least one or repeated number of hits (R= n_{col} /2^n) as shown in Figure 4.4.e. Thus, no successive zeros appear in the CH (j=0), and as a result, the successive zero histogram has no hits, as shown in Figure 4.4.f. In this case, the lower theoretical limit CCM_{min} will be zero (0).

Table 4.1 summarizes the upper and the lower theoretical limits of the CCM index values for all cases of n_{col} and the code range (2^n) of the image or ADC. It is found that the maximum CCM value depends on the ADC resolution (n) only and is independent of $n_{col,}$ while the minimum CCM value depends on both the ADC resolution and n_{col}. Besides, the minimum and maximum CCM values are independent of n_{col} when ($n_{col} \geq 2^n$). These maximum and minimum theoretical limits are calculated for CCM "values" of (1) and (0), respectively. However, as stated before, the higher CCM

Table 4.1. Summary of CCM index upper and lower limits.

CCM	$n_{col} < (2^n)$	$n_{col} \geq (2^n)$
CCM_{max} (ideally =1)	$1 - \frac{1}{2^n}$	$1 - \frac{1}{2^n}$
CCM_{min} (ideally =0)	$\frac{2^n - n_{col}}{2^n \cdot n_{col}^2}$	0

index means that the image is less complex, and the lower CCM index means a higher complex image.

CCM Mapping Function for Bounded Index Values

All possible combinations of n_{col} between 2^1 and 2^{16} and resolution (n) between 1 and 16 were tested to find the minimum and maximum ranges of the CCM to verify theoretical limits. It was found that upper theoretical limit CCM (which ideally should equal 1) changes between 0.5 and 0.999984 when resolution varies from 1 bit to 16 bits, respectively. It is independent of n_{col}, as expected. The minimum theoretical limit (which ideally should equal to 0) varies between 0.125 (for n= n_{col} =2) and 4.65×10^{-10} (for n=16, n_{col} =32768) if $n_{col} < 2^n$ and always zero if $n_{col} \geq 2^n$ as expected. Indeed, these checks show that a correction/ mapping function is needed to map the calculated CCM index values to the theoretical limits and bound the metric index between 0 and 1.

Using CCM_{max} and CCM_{min} from Table 4.1 and the linear mapping function in equation (4-3), the un-mapped CCM index in equation (4-2) can be mapped as illustrated in equation (4-8):

$$CCM = \begin{cases} \dfrac{2^n(CCM_{image} \cdot n_{col}^2 - 1) + n_{col}}{2^n(\ n_{col}^2 - 1) + n_{col}(1 - n_{col})} & n_{col} < 2^n \\ \dfrac{2^n \cdot CCM_{image}}{2^n - 1} & n_{col} \geq 2^n \end{cases} \quad (4\text{-}8)$$

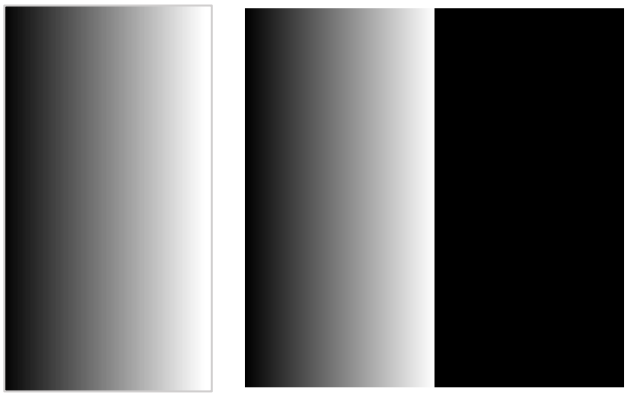
Evaluation of CCM

The first and essential feature that needs to be checked and evaluated is testing the theoretical limits and ensuring that the CCM index follows these limits as designed. Also, the derived mapping function in equation (4-8) needs to be verified. After that, the CCM index will be calculated for the standard images and other images with different resolutions and dimensions. CCM index will be compared with the available blind and objective metrics using the same images. Finally, the verification of linearity, sensitivity, and monotonicity of the CCM will be determined by changing an image parameter and compared with other metrics to check if it is a valid metric for CIS.

Evaluation of Theoretical Limits of CCM

Two types of images were synthesized to test the theoretical upper and lower limits of the CCM for cases when $n_{col} < 2^n$ and for $n_{col} \geq 2^n$ as shown in Figure 4.5.

When $n_{col} < 2^n$, the upper limit is hit when all columns have zero or the maximum ADC resolution value, so the upper limit image for the first case will be synthesized as a whole black or white image. Figure 4.5. .a shows an example of the synthesized test image for the lower theoretical



(a) (b)

Figure 4.5. Sample of synthesized images for testing lower limits of CCM index for (a) $n_{col} < (2^n)$ (b) $n_{col} \geq (2^n)$.

limit when $n_{col} < 2^n$ (n=8 and n_{col} =128). In the synthesized image, only 128 codes out of 256 total codes will appear gradually in the available 128 columns, and the rest of the codes will not appear in the image, so these codes will be the zeros that appear in between codes as (*j*) value in the row histogram in Figure 4.4.c. Note that although the image in Figure 4.5. .a looks very simple and uniform, it is challenging for CPA ADC to convert this well-distributed image because it will scan all codes from zero to the maximum ADC resolution, and it will consume more power and time.

For the second case, when $n_{col} \geq 2^n$, the upper limit remains the same as the first case. All pixels hold zero or the maximum ADC resolution value, and the synthesized images will be a whole black or white image, respectively. For the lower limit, a test image is synthesized such that all available codes appear in columns. If there are any extra columns, it will hold zero value, or it may hold any random values because the CCM index does not care about the repetition of codes; it takes care of the number of successive zeros between codes in the CH. By synthesizing a test image that way, all row histograms were filled up with codes, and there are no available zeros between the codes to count, which is the most challenging case for ADC that hits the theoretical lower limit. Figure 4.5. .b illustrates a synthesized image as an example for n_{col} =512 and n=8.

The synthesized images have two main parameters to create; the number of columns and bit depth or, in other words, resolution. To study the upper and lower limits of CCM and how general is it for all cases, I scanned the number of ADC resolution bits from 1 bit to 16 bits and the number of columns from 2 to 65536 simultaneously. The mapped outputs for all these combinations were as exactly as expected. The CCM is zero for the lower theoretical limit and one for the upper limit for all cases. Please note that the CCM index is inversely proportional to image conversion complexity.

Hence, the CCM is bounded between zero and one, and it will never exceed these theoretical limits. Besides, the mapping function is verified and working correctly.

The CCM Index for The Standard Images

The CCM metric is measured for different groups of images with different sizes and resolutions. Standard images like Lena, Barbra, Baboon, and Peppers (shown in Figure 4.6) were the first group tested. This group was 512 × 512 pixels in size with an 8-bits resolution. Because no IQ metric has the same purpose as CCM, HFM and HS have been selected to evaluate these images too because these two metrics are the nearest metrics that use histograms for calculating the contrast of an image, and they are blind metrics. So, any change in these images' histograms should appear in these metrics' indices. MSE and SSIM are not used in this test as they need a reference image to compare with. They will be used for the next test as a reference for comparing images and comparing metrics. Table 4.2 summarizes the simulated metrics values for the first group of standard images. The Lena image is the most complex image for the ADC to convert while the Baboon is the easiest image relative to this group from CCM's point of view. This group has a CCM index less than 0.5, which means that they all have a high conversion complexity, and that makes sense according to their well-distributed histogram shape that makes the conversion of these images hard and complex.

The standard deviations between all images are calculated for CCM, HS, and HFM to check how these metrics deviate with respect to each other for different images. It is found that CCM and HS have almost the same standard deviation, while HFM has a different value, which is expected.

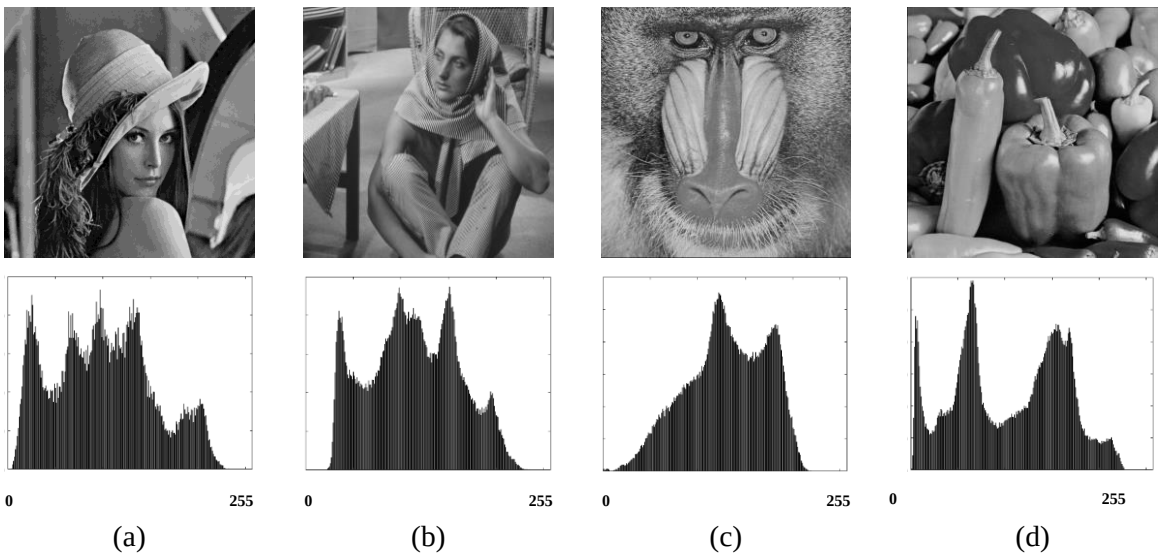


Figure 4.6. 8-bits standard images under test and the corresponding full frame CH (a) Lena (b) Barbra (c) Baboon (d) Peppers.

Table 4.2. Summary of standard images evaluation for different IQ metrics

Image	HFM	HS	CCM
Lena	0.622	0.297	0.239
Peppers	0.756	0.375	0.260
Barbra	0.560	0.277	0.313
Baboon	0.484	0.254	0.352
Standard Deviation	0.099	0.045	0.044

The creator of HS and HFM [84] concluded that HS is more meaningful than HFM for contrast detection as it takes into consideration both histogram counts and histogram bins. Because HS and CCM have the same standard deviation between the same group of images using the same kind of image information, CH, the CCM index can be trusted as it follows the same HS context. So, CCM can track any changes in an image histogram correctly.

The second selected group for the test was 1920 × 1200 pixels with 16-bits resolution images, as illustrated in Figure 4.7. CCM, HS, and HFM are calculated for these images as well. The summary of these calculations is shown in Table 4.3. Note that images (d) and (e) have less conversion complexity (higher CCM index) for ADC as they are mostly bright or dark images, respectively. While on the other hand, images (a) and (b) look very well distributed images; thus, they are more complex to be converted with the lower CCM index. Looking at image (a), the CH is very well distributed, and most of the codes appeared on it. So, this image is very complex to be converted by CPA ADC as it will count all codes starting from 0 to 65535 for every row, which makes the conversion of this image consume more power and time than images like (d) or (e). The later images have somewhat narrow CH in which not all the codes appeared, and successive zeros will exist instead, so CPA ADC can jump over empty (successive zeros) codes to save power and

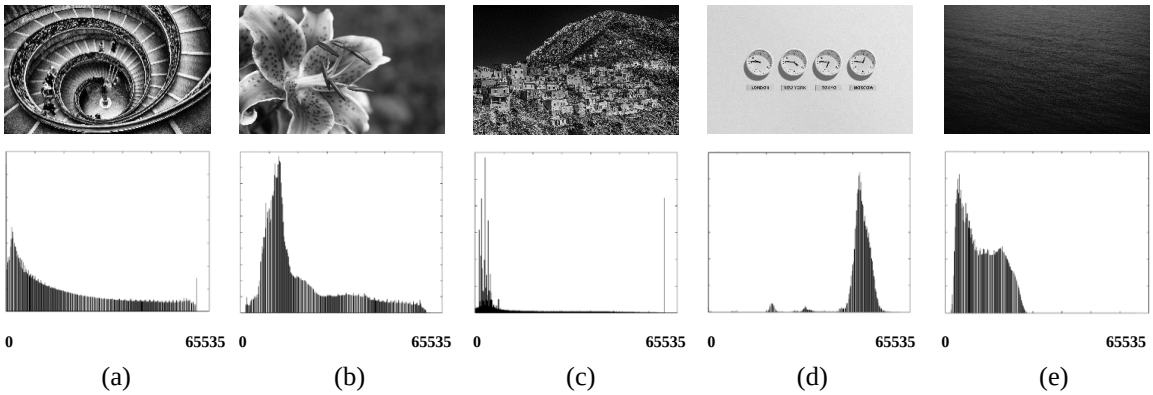


Figure 4.7. 16-bits images under test and the corresponding full frame CH

Table 4.3. Summary of 16-bits images evaluation for different IQ metrics

Image	HFM	HS	CCM
a	0.8165	0.4470	0.0128
b	0.5368	0.2705	0.1557
c	0.6688	0.4019	0.3056
d	0.1076	0.0549	0.6649
e	0.4812	0.1764	0.7868

time of conversion, which proves the theory of successive zeros make an image easier to be converted.

Monotonicity, Sensitivity and Linearity Evaluation

After calculating some IQ metrics for different images, it is very important to calculate the IQ metrics for an image after adding several kinds of distortions and check if these IQ metrics are responsive to the added distortions and also to see if they respond in a monotonic fashion or not. Not only monotonicity needs to be checked, but also the sensitivity of a metric is important too. Sensitivity is calculated to see how much IQ metrics could change corresponding to the amount of linearly introduced distortion to an image. In addition to sensitivity, linearity is also a parameter that should be verified to check how linear the IQ metric is with respect to the linear change in distortion.

Many image distortion types can be used in MATLAB [94] to add distortions to an image, namely, the blurring filter effect, Gaussian white noise, salt and peppers noise, and multiplicative noise. Also, brightness change is an important effect of the new metric that was added to these distortions.

All of these distortion types can be controlled by a specific parameter related to each of them. For example, the variance parameter is used to control the blurring filter and multiplicative noise, while the noise density parameter is used in the salt and peppers noise type. Gaussian white noise is controlled using the variance and mean parameters. To be able to control Gaussian white noise, I fixed one parameter while changing the other one linearly. A fixed number was added to all the image pixels for brightness change and increased this number linearly. The scanning parameter is changed 64 times linearly, adding more distortion to the image linearly. IQ metrics are calculated with respect to this linear distortion change. The 8-bits 512 × 512 pixels Lena image was chosen to be the image under this distortion test. The combination of distortion types and scanning of each distortion parameters created 384 distorted views of the Lena image. Figure 4.8 shows the original Lena image and a sample from each type of distortions that were introduced to the Lena image.

The CCM, HS, HFM, BRISQUE, NIQE, PIQUE, SSIM, and MSE are calculated for the Lena image after applying and linearly scanning the aforementioned types of distortions. All these metrics are checked for monotonicity with respect to the distortion increase. Table 4.4 summarizes the monotonicity features for these metrics. It is found that CCM and MSE are monotonic for all kinds of distortions. HS, PIQUE, and SSIM are monotonic for most distortion types, while HFM, BRISQUE, and NIQE are non-monotonic for almost every distortion type.

These IQ metrics' sensitivity is calculated as the percentage of the difference between the metric values, which corresponds to the maximum and minimum change of the added distortion divided by the original metric value. Table 4.5 summarizes the sensitivity values for all metrics with respect to added distortions. The table is divided into two groups; bounded metrics and unbounded metrics. The bounded group is for those bounded metrics between zero and one, while another group has no limit. This table is split that way to be able to perform a fair comparison. CCM has the highest sensitivity for all types of distortions except for Gaussian white noise with the fixed mean and changing variance; it comes in the second position after SSIM. For the unbounded group, MSE comes as the highest sensitive unbounded metric for every distortion change.

For linearity calculations, the R-square value correlates each metric dataset output to the linearly changed distortion and figuring out how linear the metric corresponds to linear distortion

Figure 4.8. Samples of different types of distortions (a) Original (b) Brightness change (c) Blur distortion (d) Gaussian noise distortion (e) Salt and peppers distortion (f) Multiplicative noise distortion

Table 4.4. Monotonicity summary of all IQ metrics

Distortion type	CCM	HFM	HS	BRISQUE	NIQE	PIQUE	SSIM	MSE
Brightness increase	**yes**	no	yes	no	no	yes	yes	yes
Bluer filter	**yes**	no	yes	no	no	yes	yes	yes
Gaussian white noise with mean change and fixed variance	**yes**	no	yes	no	no	no	no	yes
Gaussian white noise with fixed mean and variance change	**yes**	no	yes	yes	no	yes	yes	yes
Salt and pepper noise	**yes**	yes	yes	no	yes	no	yes	yes
Multiplicative noise	**yes**	no	no	no	no	yes	yes	yes

Table 4.5. Summary of metrics sensitivities to different distortion types

Distortion type	Bounded metrics				Un-Bounded metrics			
	CCM	HFM	HS	SSIM	MSE	BRISQUE	NIQE	PIQUE
Brightness increase	**316%**	61%	100%	65%	**169375%**	339%	175%	441%
bluer filter	**100%**	12%	32%	47%	**15789%**	589%	48%	441%
Gaussian white noise with mean change and fixed variance	**1026%**	99%	100%	19%	**4119%**	9%	3%	15%
Gaussian white noise with fixed mean and variance change	74%	28%	26%	**81%**	**5486%**	60%	155%	101%
Salt and pepper noise	**339%**	60%	230%	100%	**6345%**	32%	5272 %	154%
Multiplicative noise	**55%**	30%	7%	50%	**6131%**	154%	101%	165%

change. Table 4.6 summarizes all R-squared values for all metrics for added distortions. CCM has the maximum linearity value of 99% for the first three distortion types, while MSE has the maximum linearity for the other three types.

CCM Evaluation for brightness change

The CCM theory is based on counting the number of successive zeros in a row histogram. When an image becomes darker or brighter, it becomes less complex for the ADC to convert as a result of increasing the number of successive zeros (see Figure 4.2). Brightness is the most important image parameter that is used to verify the new theory. The 8-bits 512 × 512 pixels Lena image was selected to perform this test because of being the most difficult image of the selected standard

Table 4.6. Summary of R-squared values for all metrics

Distortion type	CCM	HFM	HS	BRISQUE	NIQE	PIQUE	SSIM	MSE
Brightness increase	**99%**	55%	85%	90%	82%	86%	98%	98%
Bluer filter	**99%**	11%	**99%**	43%	15%	29%	73%	**99%**
Gaussian white noise with mean change and fixed variance	**99%**	92%	86%	47%	0%	92%	78%	98%
Gaussian white noise with fixed mean and variance change	83%	69%	99%	78%	89%	62%	73%	**100%**
Salt and pepper noise	59%	63%	86%	0%	55%	45%	46%	**100%**
Multiplicative noise	87%	82%	92%	45%	86%	83%	89%	**100%**

images. Figure 4.9 illustrates the original Lena image, and its brightness is increased linearly by adding 40 to 240 digital values to each pixel in images in Figure 4.9.b to f, respectively. The brightness increase is performed by simply adding a specific digital number to all pixels, which, as a result, makes the histogram shift to the maximum value according to the added number. This process will introduce extra zeros in the lower code range of the row histogram and cause many high illumination pixels to saturate at the row histogram's maximum value.

Table 4.7 summarizes measurements of CCM, HS, HFM, BRISQUE, NIQE, PIQUE, SSIM, and MSE corresponding to the brightness increase of the Lena image. It is obvious that CCM has the highest sensitivity to the brightness increase. CCM changes 316% when brightness increases while no other bounded metric exceeds 100% change for the same brightness change. These results are plotted in Figure 4.10. CCM has a relatively small response to small value brightness increases; then, it increases linearly as brightness increases further, which is the typical case for MSE and SSIM with different rates. The rest of the metrics have a non-monotonic and non-linear response to the brightness increase.

(a) (b) (c)

(d) (e) (f)

Figure 4.9. Brightness increasing of Lena image from +40 to +200 (a) Original (b) +40 (c) +80 (d) +120(e) +160 (f) +200

Conclusion

A new metric, CCM, is designed for the assessment of images before processing by intelligent CPA ADC electronics. The CCM index is an important parameter for intelligent electronics to obtain optimum performance, such as maximizing speed and minimizing the power consumption with the minimum image distortion. The CCM is designed to be bounded between zero and one. These limits are designed and tested for all possible cases of image resolution and image dimensions. Additionally, a mapping function is created to correct the CCM index for specific cases. The mapped output is tested as well, resulting in a successful output as expected for all limits. It

Table 4.7. Metrics measuring values due to brightness change for Lena image

Metric type	Metric	Original	+40	+80	+120	+160	+200	⁄sensitivity
bounded	CCM	0.2388	0.2516	0.3793	0.5216	0.6773	0.9036	**316%**
	HFM	0.6226	0.8649	0.8019	0.6812	0.4268	0.1225	61%
	HS	0.2969	0.2969	0.2969	0.2930	0.1367	0	100%
	SSIM	1	0.8980	0.7672	0.6328	0.5009	0.3709	65%
Un-bounded	MSE	0	1598.49	6117.30	12812.52	19611.28	26242.6	**169375%**
	BRISQUE	10.1529	10.0255	29.9918	29.0638	34.7735	51.715	339%
	NIQE	4.5957	4.6040	4.6502	5.2153	7.1910	10.146	175%
	PIQUE	18.4942	18.5423	22.9832	30.7049	42.2203	62.356	441%

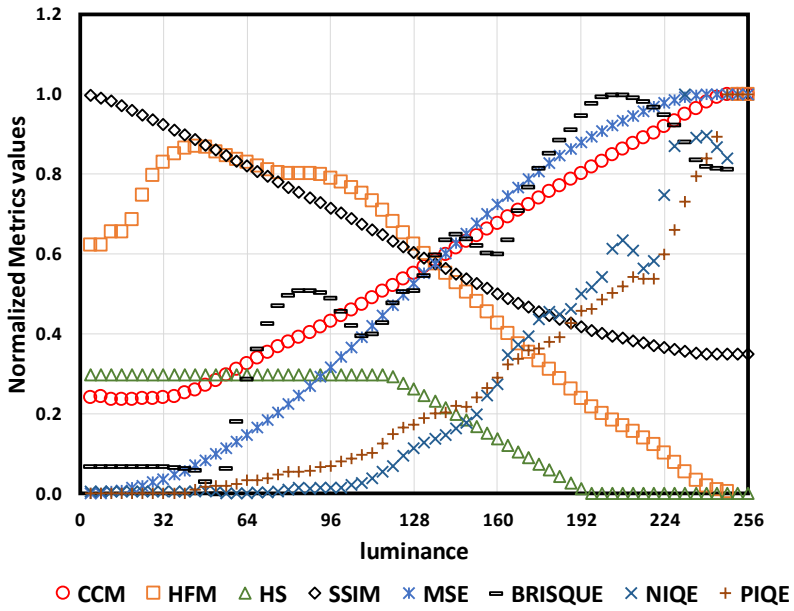


Figure 4.10. Plotting IQ metric versus brightness change (unbounded metrics are normalized to be from 0 to 1)

gives zero for the lower limit and one for the upper limit images. The CCM is an independent matric that does not need any reference image to compare with. This independence makes CCM suitable for analog processing electronics. The CCM uses a simple calculation method based on a histogram of a row that allows fast processing and minimizes hardware implementation complexity. Finally, it is not for human perception as a very simple and uniform image is very challenging for analog electronics to process and consume more power and time.

CCM is tested in different ways. First, the theoretical mapped output is tested using synthesized images to reach the maximum and minimum limits according to the successive zeros theory. Second, CCM is calculated for different standard 8-bits greyscale images resulting in the same behavior as HS, the metric that uses the histogram for calculating image contrast. Third, CCM is calculated for 16-bits images to verify how general it is. This test results in a very well understood result that the most bright or dark images have less conversion complexity than the well-distributed images in the same group. Forth, six types of distortion are introduced linearly to an image, and CCM and other IQ metrics are calculated for these types of distortion. Monotonicity, sensitivity, and linearity are investigated for all available IQ metrics. The CCM results in a monotonic behavior to linearly scanned distortion with linearity of 99% and 316% sensitivity to brightness change. The CCM index is tested for greyscale images, however, for colored images, the CCM index can be calculated for each color channel, and the weighted combination of them can be used to calculate the global CCM index for the colored images.

Chapter 5: Accelerated Single Slop Look-Ahead Ramp (ASSLAR) ADC and CCM Comparison Methodology

"A Simple Image Quality and Complexity Metric for Smart Image Sensor Evaluation."

Forthcoming in IEEE Access Journal, 2020.

Many speedup techniques (SuPT) were developed to enhance the conversion speed of the ramp-type ADCs used in the CMOS image sensors. These techniques trade the speed with other parameters such as image quality, hardware complexity, and silicon area. The accelerated ramp (AR) SuPT was proposed based on the shot-noise limitation, and it achieved a constant and high speedup ratio (SuPR) within the allowed image degradation that cannot be observed by a human. Another SuPT was introduced and implemented in chapter 3, showing excellent performance in terms of the SuPR and power consumption saving, which is the SSLAR. The SSLAR has an interesting property that allows its conversion algorithm to exploit any opportunity to increase the SuPR based on image code distribution, so it is called opportunistic SuPT. In this chapter, the hybridization of these two powerful SuPTs is proposed to take the advantages of both and overcome their disadvantages. The proposed SuPT is called accelerated SSLAR (ASSLAR). The ASSLAR is tested for a commonly used image database from Caltech [95]. The ASSLAR showed promising results; it achieved an average of 20% SuPR enhancement over the AR SuPT for less complex images keeping the image quality unaffected. The ASSLAR investigation and comparison were performed using a new comparison methodology presented in this chapter. The proposed comparison methodology is a generic method that can be applied to compare any SuPTs based on the CCM and the SSIM indexes. The case study of this methodology was about comparing AR and ASSLR SuPTs, and it is presented at the end of this chapter.

Introduction

The SSLAR speedup technique (SuPT) was proven to enhance the speedup ratio (SuPR) of the SSR ADC [1]. However, the SSLAR SuPT needs some improvements in terms of the output image quality. Also, any chance to increase its SuPR should be explored. In this chapter, a new SuPT is proposed to increase the SuPR of the SSLAR by hybridizing the SSLAR with the accelerated ramp (AR) SuPT [24] [36] to develop what is called accelerated SSLAR (ASSLAR). The evaluation of the effectiveness and the comparison of the ASSLAR with other SuPTs is performed using a proposed comparison methodology based on the CCM index accompanied by the SSIM index. This general methodology is proposed for a fair comparison between any SuPTs that are used for speeding up the standard SSR ADC.

The Accelerated Ramp (AR) Speedup Technique

The Accelerated Ramp (AR) Concept

Figure 5.1 illustrates the relationship between light intensity and the CIS output signal response and different kinds of noises. Two types of noise exist; the constant noise and the photon shot noise. The constant noises include the thermal noise and flicker (1/f) noise that form a noise floor [36]. On the other hand, the photon shot noise is not constant and directly proportional to the square root of the sensor signal. The constant noise dominates if the output signal is small, while the photon shot noise dominates when the output signal level increases. The ADCs were designed such that their quantization noise does not exceed the constant noise floor. However, for higher output signal levels, where the shot noise dominates, the ADC's performance is higher than needed, i.e., ADC quantization steps can be increased without affecting the overall signal to noise ratio (SNR). The ramp ADC steps should be related to photon shot noise rather than linear steps, and it should always be less than the overall noise limit shown in Figure 5.1. The AR was proposed in [24] and [36] to enhance ramp ADC conversion speed using accelerated ramp based on the photon shot noise limit of a sensor signal. The AR step size was increased to reduce the overall number of required steps and perform the A-to-D conversion faster, as shown in Figure 5.1. The shot noise margin (SNM) is defined as the ratio of the accelerated ramp step size to signal shot noise and can be calculated from [24]:

$$SNM_{shot} = \frac{W_{shot}}{D_{shot}} \tag{5-1}$$

where W_{shot} is the maximum step size of the accelerated ramp, and D_{shot} is the shot-noise limit. The recommended shot noise margin is 1/2 to reduce contours' possible appearance on the reproduce

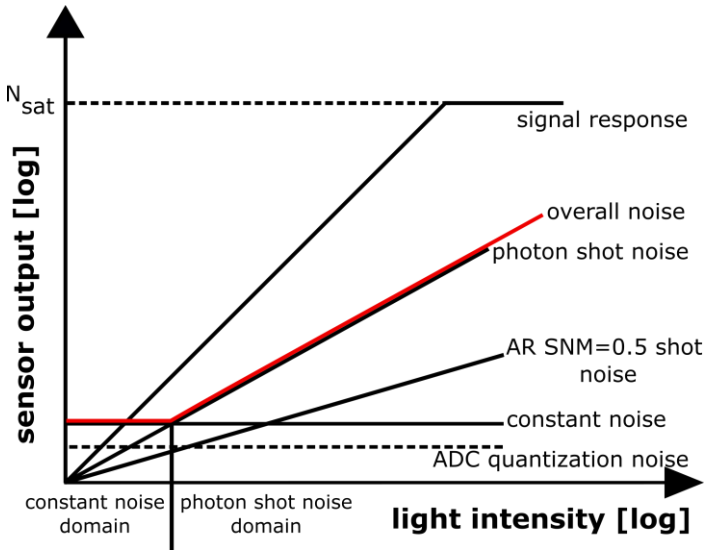


Figure 5.1. Conceptual logarithmic plot of the sensor's response to light and corresponding noise sources.

image [24]. If the SNM is increased, which means the accelerated ramp step size is increased, the AR ADC becomes faster, and the output image quality becomes worse because the ramp steps sizes become closer to the shot-noise limit, or it may exceed it. The inverse is correct, if the noise margin becomes smaller, the conversion process will be slower, but the output image quality will be better. So, there is a tradeoff between image quality and conversion speed.

The Controlling Parameters (CPs) of Accelerated Ramp (AR) SuPT

The AR technique has a fixed SuPR as the total number of required steps is fixed for the specific image or ADC resolution. For example, for the 8-bit images and SNM of 1/2, it will need 61 steps instead of 256 steps required by standard single slope ramp (SSR) ADC. Thus, the AR SuPR of 8-bit images is 4.19. Also, for the 12-bits images and noise margin is 1/2, it will need 606 steps instead of 4095, or it will have a SuPR of 6.70 more than the SSR ADC. For an 8-bits image, if the SNM increased to 1.0 and 2.0, the SuPR will jump to 8 and 14.2, respectively. The only control parameter (CP) that seems to control the SuPR of the AR is the SNM parameter. However, the authors of the AR strongly recommended keeping it at most 1/2 to avoid any output image distortion. So, the AR relatively has no CPs to control its SuPR. It depends only on the image or ADC resolution and provides a constant SuPR for specific image resolution.

Advantages and Disadvantages of the Accelerated Ramp (AR) SuPT

The AR SuPT is a technique that exploits the shot-noise limit to increase the conversion speed of the SSR in image sensors. It has a relatively *fixed* SuPR for each image resolution regardless of the current scene/image's conversion complexity. The SuPR comes at the expense of the output image quality. However, its SuPR can be increased more based on the conversion complexity of the scene itself or the input image; i.e., if the input image is less complex to convert, it should be converted faster and vise versa. So, this feature should be added to the AR SuPT to enhance its SuPR. The AR SuPT needs some improvements to reach the optimum operating condition to get the maximum SuPR and the minimum image quality (IQ) degradation.

The SSLAR Technique

As illustrated in chapter 3, the SSLAR ADC is an opportunistic SuPT that was introduced to speed up the SSR ADC, [1] [61] [62]. The main idea behind it is to look into a code range (defined as to look ahead *step* size, *k*) before converting that range (t to t+k). If no signal exists in the range, then the ADC's ramp signal jumps *k* step, skips scanning the range (*t* to *t+k*), and looks ahead to the next code range (*t+k* to *t+2k*). If, on the other hand, *h* number (defined as the *threshold*) or more signals exist in the look-ahead code range (*t* to *t+k*), ADC's ramp signal scans the core range using the regular ramping step size of 1. If less than *h* number of signals exist in the range, all signals in the

range will be considered having middle code value of the look-ahead range (*t+k/2*), and conversion continues without scanning the range (known as *jump*). As a result, when a jump occurs during A-to-D conversion, the pixels in the *k*-range will incur a *k/2*-LSB error in the SSLAR SuPT. This process is repeated until the ramp reaches the 2^n value, where n is the image resolution.

The Controlling Parameters (CP) of SSLAR SuPT

The step size (*k*) and the threshold (*h*) are the main CPs for the SSLAR. The threshold is defined as the maximum allowed number of codes that can make the ramp skip during the jump process, while the step size is a predefined parameter that is used in the acceleration process. Both CPs are used to control the tradeoff between the SuPR and the output image quality. As the threshold and the step size increase, the conversion speed increases, and the output image quality degrades. However, this relationship is not linear; if the threshold and/or the step size increased further, the SuPR would saturate, or it may decrease [1] [61], [62]. So, there are optimum settings for these parameters to reach the maximum SuPR.

Advantages and Disadvantages of the SSLAR SuPT

The SSLAR SuPT is a technique for exploiting any chance to increase the conversion speed based on the image/scene nature. So, if the scene is not complex in terms of details and code distribution, SSLAR SuPT can convert images faster than if it is a complex scene with many details and uniform code distributions. It trades image quality (IQ) for extra speedup and reduced power consumption. However, the SSLAR has one issue regarding the output IQ: it does not consider the shot-noise limit. If the step size is set too large, the ramp will exceed the shot-noise limit, especially when the signal level is still small. Figure 5.2 shows an example of an 8-bits resolution ADC to illustrate this issue. In this figure, the X-axis is the number of required steps for an ADC to finish the conversion, while the Y-axis is the bit resolution of an image. The SSLAR ramp is plotted for the step sizes of 1, 2, 3, 4, 5, 10, and 16. As the step size increases, the SSLAR finished the conversion faster. However, the SSLAR linear ramps may exceed the shot-noise limit. The shot noise limit is plotted in this graph as well to show the maximum allowed step size. It is found that for the SSLAR's step size equal to or larger than 5, its ramp always exceeds the shot-noise limit, which may affect the quality of the produced image negatively. For the step size less than 5, the SSLAR ramp exceeds the limit only when the signal level is small.

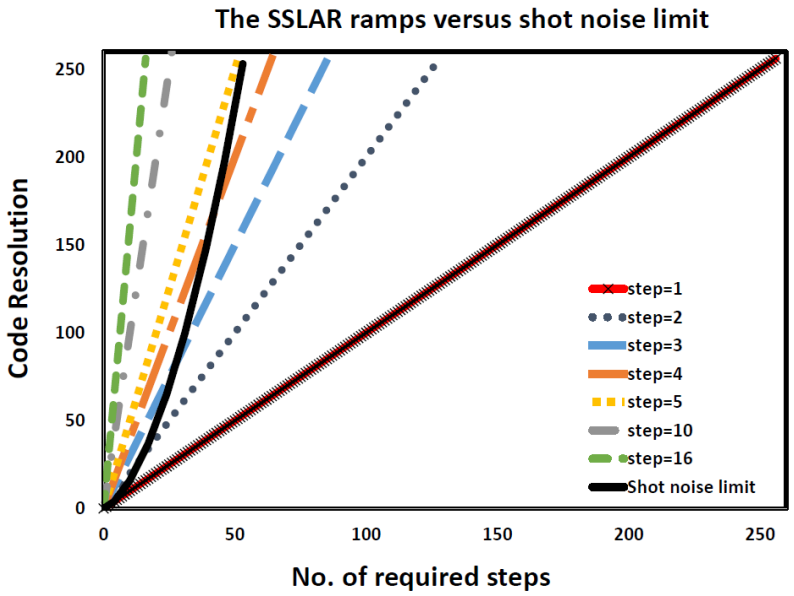


Figure 5.2. The SSLAR ramps of different step sizes compared with the shot noise limit.

Proposed Accelerated SSLAR Technique

The AR and SSLAR SuPTs are to enhance SSR ADC speed. Each has its advantages and disadvantages. I propose to hybridize these techniques to develop the accelerated-SSLAR ADC to take the advantages of both and overcome their disadvantages.

The ASSLAR concept

The ASSLAR SuPT hybridizes the properties of the SSLAR and AR SuPTs, maintaining the benefits of both techniques while achieving faster conversion speed and the best image quality than both. In SSLAR SuPT, two ramps exist; the main ramp has a step size larger than 1, and the fall-back ramp, which is used when the jump is declined and increases linearly by the step size of 1. The proposed ASSLAR SuPT will replace the main and the fall-back ramps of the SSLAR SuPT with two accelerated ramps that follow the same theory as the AR SuPT. By doing this, the SSLAR SuPR will be increased because the accelerated ramp is faster than the linear ramp as it follows the shot-noise limit. The AR technique always blindly (regardless of the processed image complexity) follows the shot-noise limit by a SNM of 1/2, which means that its value is always below half of the shot-noise limit. However, the opportunistic property of the SSLAR SuPT should be exploited based on the image complexity. So, the proposed ASSLAR SuPT will have the chance to exceed the SNM limit for the accelerated ramps to find any opportunity to speed up the conversion process without degrading image quality.

The Controlling Parameters (CP) of ASSLAR SuPT

Since the ASSLAR SuPT has two different accelerated ramps, each ramp will have its own SNM. The main ramp SNM parameter (M-SNM) is the first CP of the ASSLAR SuPT. The second CP is the fall-back SNM parameter (FB-SNM) that controls the fallback ramp acceleration of the ASSLAR SuPT. Combining these two important CPs will result in different options that will be chosen to get the maximum SuPR with the minimum IQ degradation. As the ASSLAR SuPT is built on the SSLAR SuPT, it is expected that there is (are) some common CP(s) between them like the threshold or the step size. However, the ASSLAR SuPT is based on the AR SuPT that has a fixed step size limited by the shot-noise limit so, the ASSLAR SuPT does *not* have the step size as a CP, but it has only the threshold parameter as a CP similar to the SSLAR SuPT. Another parameter may affect the overall SuPR of the ASSLAR, which is the fall-back cost or penalty. When the SSLAR falls back, it incurs 2 extra clock cycles as a cost of jump failure, which causes a delay of conversion cycles. In the ASSLAR SuPT, the clock frequency is increased to minimize the delay resulting from falling back. Thus, the ASSLAR SuPT has four CPs which are: M-SNM, FB-SNM, threshold, and fall-back cost. These parameters will be investigated to choose the optimum settings for the ASSLAR SuPT.

The ASSLAR Simulation Results

The ASSLAR SuPT is modeled and simulated using the Matlab software. The selected test images were a group of 52, 8-bits, grey-scale images from the Caltech image database [95] that cover almost the maximum range of CCM index from 0 to 1. When an image is processed, the CCM and the SuPR index are calculated, then the SSIM index of the processed image versus the original image is determined. Different combinations of M-SNM and FB-SNM are assigned with the values 1, 2, 4, and 6 for both of them to investigate the effect of these two parameters on the output SuPR and SSIM. This process is repeated for the fall-back cost value of 0.5, 1, and 2 clock cycles to check the effect of increasing the clock frequency of the SSLAR. The different cases of these combinations are illustrated case by case.

The FB-SNM is the first CP to investigate. All other parameters are fixed, and the FB-SNM is assigned to the fixed values of 1, 2, 4, and 6. Figure 5.3 is just an example to show the relationship between the FB-SNM and SuPR. The M-SNM for the data in Figure 5.3 is 1, and the cost is 2. In this figure, it can be observed that the SuPR always increases when the CCM index increases, i.e., when the images become less complex. This is an expected response for the ASSLAR SuPT as its SuPR depends on the input image's complexity because this is one of the inherent characteristics of the

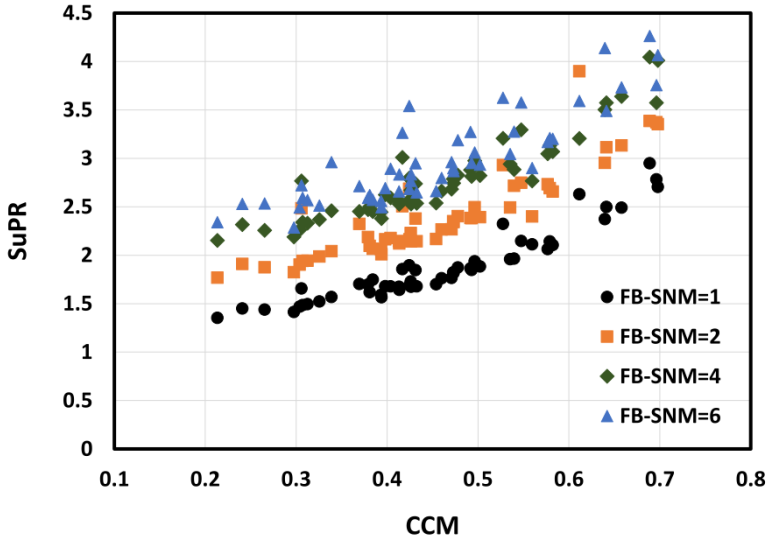


Figure 5.3. The ASSLAR SuPR vs. CCM index for different FB-SNM

SSLAR SuPT. Simulations also showed that the SuPR increases with increased FB-SNM, as expected. That is one of the AR SuPT's features that the ASSLAR SuPT inherited.

Figure 5.4 shows the SSIM for the same conditions and data set, as in Figure 5.3. The SSIM index is used to quantify the similarity between the original image and the processed image. As the SuPR increased, the SSIM index or similarity between the original and processed images is decreased. For this example, due to the increase of FB-SNM, SuPR increased from 1.87 to 2.99 (an average 59.8% increase), while SSIM is decreased from 0.997 to 0.959 (a reduction of only 3.9%).

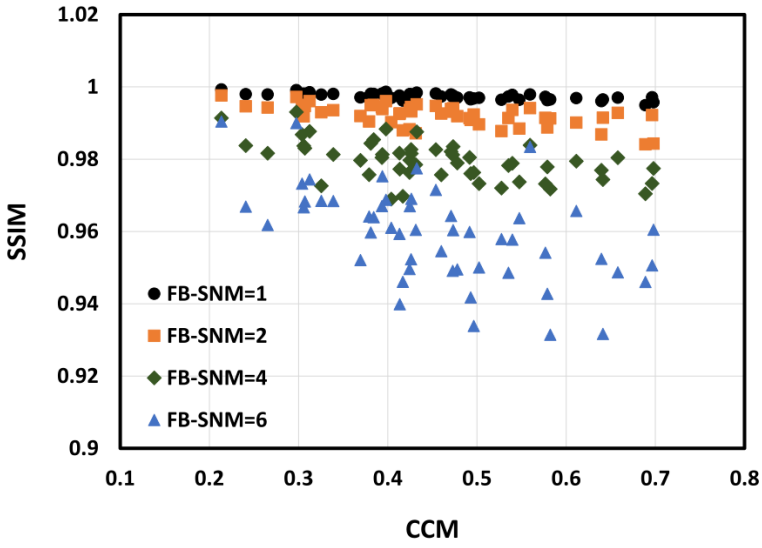


Figure 5.4. The ASSLAR SSIM vs. CCM index for different FB-SNM

This suggests that the ASSLAR SuPT increases the SuPR and keeps the IQ not degraded dramatically. The effect of the SuPTs on IQ will be investigated in detail in the following section.

The second parameter that is investigated is M-SNM CP of the ASSLAR SuPT. Following the same procedures as FB-SNM, all other CPs are fixed, and the M-SNM is assigned to 1, 2, 4, and 6 to investigate its effect on the SuPR and the SSIM index. The other CPs are assigned for this simulation are; FB-SNM = 4 and the cost = 2. Figure 5.5 illustrates the simulation results of the data set showing the SuPR of the ASSLAR SuPT for different M-SNM. The SuPR curves are shifted up with increased M-SNM values showing the relation between SuPR and M-SNM parameters. Also, for the same M-SNM, the SuPR is increasing with the CCM index, which means it follows the CCM theory that states, "if an image is less complex, it will be easy and fast to be converted" as in the SSLAR/ASSLAR SuPT.

Figure 5.6 illustrates the SSIM index of the processed images for different M-SNM values. It is clear that the processed image's IQ becomes worse as the M-SNM increased or, in other words, as the SuPR increased. The SuPR is enhanced on average by 169%, while the SSIM decreased on average by 4.3% only.

The last CP that will be investigated is the fall back cost. In the SSLAR SuPT, the cost is the number of the clock cycles that the SSLAR SuPT will incur if a jump is declined and ramp fall back will occur. It was assigned as 2 clock cycles in the existing SSLAR SuPT. Figure 5.7 shows the enhancement in the SuPR if the cost is reduced to 1.0 or 1/2 clock cycles. This figure's data is the

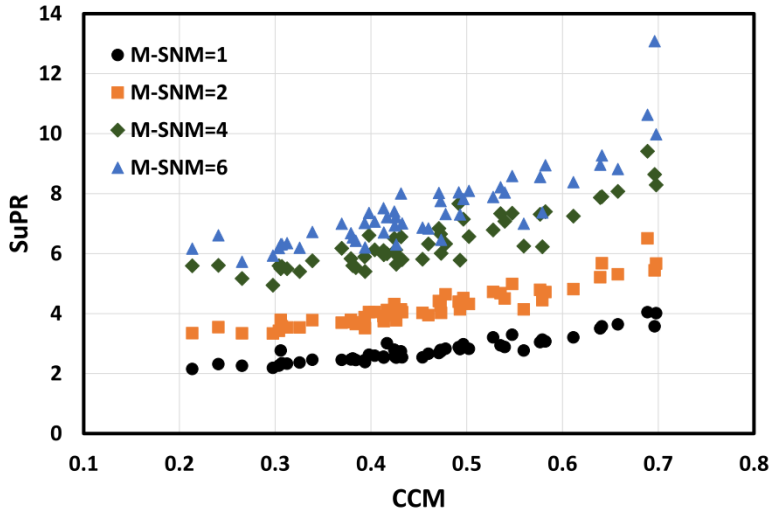


Figure 5.5. The ASSLAR SuPR vs. CCM index for different M-SNM

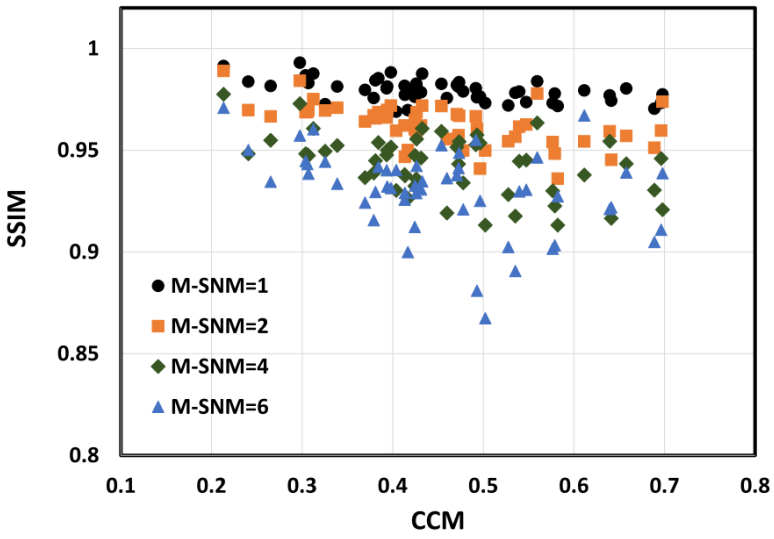


Figure 5.6. The ASSLAR SSIM vs. CCM index for different M-SNM

average of SuPR of all combinations of M-SNM and FB-SNM for the values 1, 2, 4, and 6. The SuPR will be enhanced by 22.9% if the cost CP is assigned to 1/2 clock cycle instead of 2 clock cycles.

ASSLAR Performance Optimization.

Figure 5.8 (a) and (b) illustrates a general view of the average SuPR of the ASSLAR SuPT versus different values of M-SNM and FB-SNM CPs, respectively. As discussed before, as the SNM is increased, the SuPR has increased accordingly, and the resulting IQ is degraded as well. The average SuPR can be increased from 3 up to almost 7 times, according to the SNM CP settings.

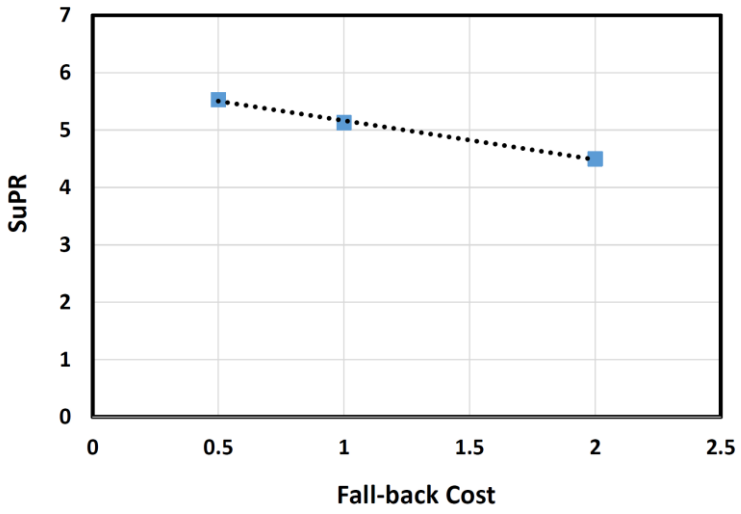
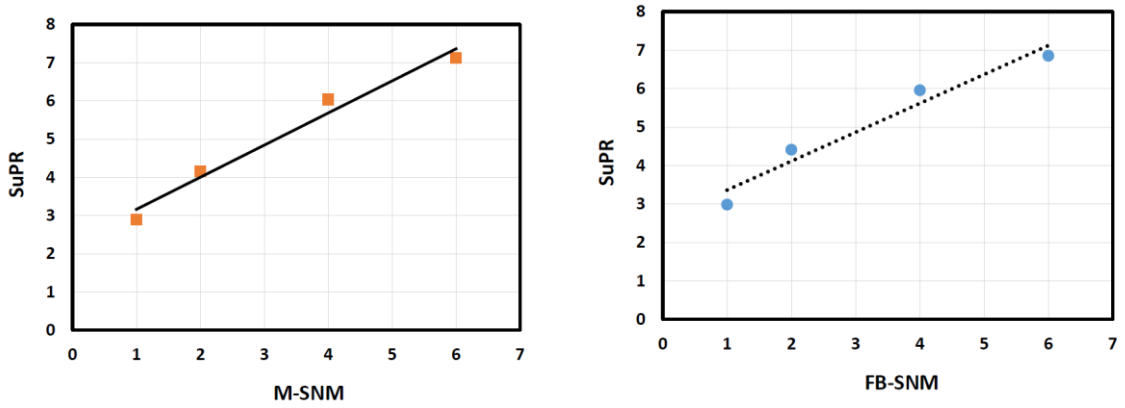


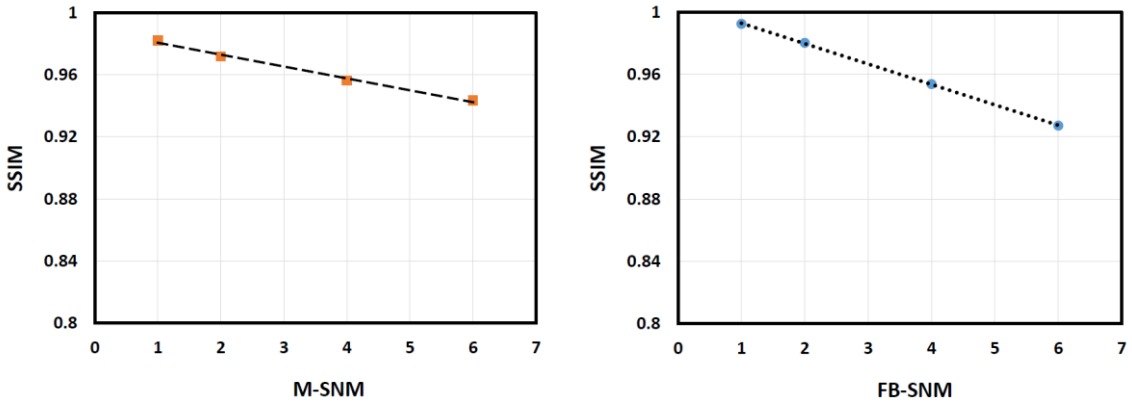
Figure 5.7. The ASSLAR SuPR vs. average fallback cost for all available combination of SNM CPs.



(a) (b)

Figure 5.8. The ASSLAR generic SuPR vs. CCM index for different M-SNM and FB-SNM.

Figure 5.9 (a) and (b) show an average SSIM degradation versus different values of M-SNM and FB-SNM CPs, respectively. The SSIM drops from 98% to 92% for the ASSLAR SNM CPs between 1 and 6. So, there is a tradeoff between the SuPR and IQ of the images, as reflected by the SSIM. The optimum settings of the SNM CPs should be determined to achieve a maximum SuPR and a maximum SSIM index (minimum IQ degradation) at the same time. Table 5.1 shows the ASSLAR operating modes and the corresponding CPs, and average SuPR, SuPR standard deviation, average



(a) (b)

Figure 5.9. The ASSLAR generic SSIM vs. CCM index for different M-SNM and FB-SNM.

Table 5.1. The SNM settings for different operation modes of ASSLAR and resulting average SuPR, and SSIM values for the Caltech database images.

Operation Modes	M-SNM	FB-SNM	SuPR	SuPR Standard Deviation	SSIM index	SSIM Standard Deviation
1	1	1	2.23	0.4064	99.7%	0.0009
2	1	2	3.02	0.4869	99.2%	0.0030
3	1	4	3.69	0.4625	98.0%	0.0054
4	1	6	3.98	0.4686	95.9%	0.0132
5	2	1	2.79	0.4836	99.5%	0.0019
6	2	2	4.10	0.6418	98.6%	0.0043
7	2	4	5.44	0.7323	96.3%	0.0103
8	2	6	6.02	0.7489	94.5%	0.0208
9	4	1	3.60	0.5636	99.1%	0.0034
10	4	2	5.52	0.7685	97.5%	0.0084
11	4	4	8.03	1.0665	94.3%	0.0147
12	4	6	9.22	1.1531	91.6%	0.0243
13	6	1	3.97	0.7600	98.7%	0.0059
14	6	2	6.38	1.2317	96.9%	0.0111
15	6	4	9.15	1.4687	93.0%	0.0205
16	6	6	11.29	2.4845	88.8%	0.0326

SSIM, and SSIM standard deviation for all of the images under test. Each mode is an indication of a specific M-SNM and FB-SNM as a combination. For example, mode-9 means the M-SNM is 4, and the FB-SNM is 1. Figure 5.10 is a plot of the data in Table 5.1. It is clear that the SuPR is increasing as the SNM CPs increase with different rates in both main and fall back ramps. The standard deviation of the average value of the SuPR is plotted to show how these data are deviating around the average value. Following the same fashion, the average SSIM index and its standard deviation are plotted for the corresponding SuPR. As observed before, the SSIM decreases while the SuPR increases. The optimum SuPR and SSIM can be determined from this data, such that the maximum SuPR is obtained for the minimum image distortion (maximum SSIM). Looking at Figure 5.10 or Table 5.1 and starting with the highest SSIM index that exceeds 97%, eight modes can be identified; modes 1, 2, 3, 5, 6, 9, 10, and 13. By looking at the corresponding SuPR of these 8 modes, some of them could be excluded due to their lower SuPR like modes 1, 2, 3, 5, and 9. So, three CPs modes could be selected to be the optimum modes for the ASSLAR, which are modes 6, 10, and 13. These

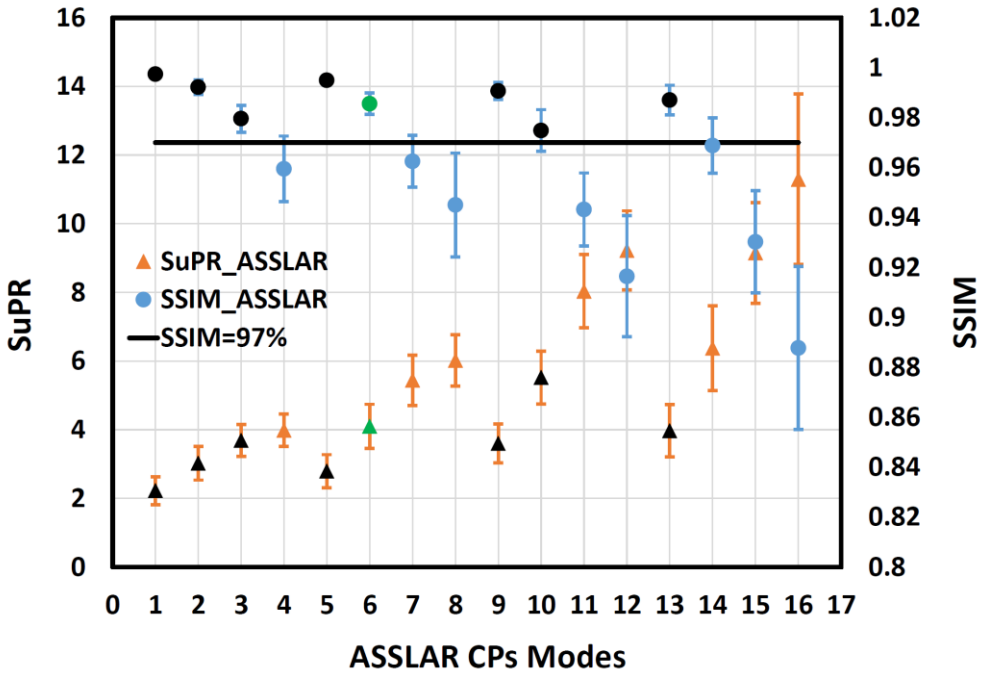


Figure 5.10. ASSLAR modes versus SuPR and SSIM for selecting optimum CP of M-SNM and FB-SNM.

modes have an average SuPR almost equal to or larger than 4 times faster than the standard SSR while keeping the SSIM index larger than 97%. Comparing modes 6 and 13, both have almost the same average SSIM index, but mode 6 has a higher average SuPR. So, mode 13 can be excluded from this selection process, leaving modes 6 and 10.

Among selected modes, the mode that will result in an image that the distortion is not noticeable will yield the optimum ASSLAR CPs. Figure 5.11 shows an example of a visual image comparison. A random image is selected from the Caltech database and processed by the nominated two CPs modes. The output (processed) images are shown in Figure 5.11. Mode-6 resulted in SSIM of 98.5% and SuPR of 5.27, while mode-10 results in SuPR of 7, but it has a lower SSIM index of 95.7%. The lower SSIM index effect can be easily recognized on the processed image as in Figure 5.11(c) on which contours between different areas in the image with varying levels of illumination can be easily observable. Thus, mode-10 cannot be chosen even it provides higher SuPR that mode-6, as image distortion is so clear and visible. Therefore, one can conclude that the mode-6 achieves optimum performance for ASSLAR SuPT that maximizes the SuPR 4 times on average and maximizes the SSIM index (minimize the IQ degradation) up to 98% on average.

(a) Original input image

(b) Mode-6 output image

(c) Mode-10 output image

Figure 5.11. Optimum ASSLAR modes comparison (a) the original input image (b) mode-6 output image, SSIM=98.5% and SuPR=5.27 (c) mode-10 output image, SSIM=95.7% and SuPR=7.0.

After determining the optimum modes for the ASSLAR, it is the time for comparison with other SuPTs. In the following sub-section, a new general comparison methodology is presented and applied to the ASSLAR and AR technique based on the CCM and SSIM metrics.

Proposed Methodology to Compare SuPTs using CCM

The proposed ASSLAR SuPT is a promising technique that takes advantage of two powerful SuPTs; opportunistic SSLAR and shot noise limited AR. However, because many CPs control the

SuPR, a methodology has to be defined to compare and evaluate different SuPTs fairly, including the ASSLAR, SSLAR, and the AR SuPTs.

The CCM index was designed to evaluate ADC SuPTs used in CPA CISs. Some SuPTs may be claimed to be the best. However, less complex scenes might be used to prove that claim. It is common practice to show the original and processed image side by side and let the readers figure out where the differences are. It may also be claimed that the SuPT does not affect IQ. Thus, it is imperative to have a quantitative metric to tell the reader, "*how much does the proposed SuPT degrade the IQ?*" and "*what is the complexity of the original image or scene to start with?*" The role of comparison methodology described in the next section that uses the proposed CCM index becomes significant to answer these pressing questions quantitatively.

The Comparison Method

Most new ADC SuPTs used in image capturing systems have some controlling parameters (CPs) to adjust performance parameters such as speed, power, resolution, or quality using quantization step size, number of steps, threshold, etc. These CPs may be common between different SuPTs or unique due to the addition of elements on a common topology. Here, only the CPs that affect relative SuPR of an A-to-D technique in CPA CIS is considered.

The CCM index will be the reference parameter throughout the comparison of the SuPTs as it is a blind metric quantifying how complex an image is without requiring a reference image. Besides, the SSIM index [82] will be used to compare the original image and the processed image because it is a full reference metric that requires the original image to indicate how similar the processed image is. Using the CCM and SSIM together for the SuPR and similarity will lead to a clear picture of how effective the new SuPT is in terms of speed improvement and output IQ, which could be related easily to power consumption and other performance parameters of a CIS.

The comparison methodology is as follows:

1- Choose a group of test images such that it covers most of the CCM index range from 0 (most complex) to 1 (least complex).
2- Check for the CP(s) of each SuPT and find the common CP(s) (if any).
3- Start with the SuPT, which has the least number of CP(s).
4- Scan the CP(s) resulting in minimum to maximum SuPR and find the SuPR, CCM, and SSIM index values for each corresponding CP(s) for each image.

5- If there is (are) common CP(s) among SuPTs, scan the common CP(s) of the other SuPT and find the CP(s) value(s) that gives the same CCM index value and find the corresponding SuPR and SSIM index value for each image.
6- If there is (are) not common CP(s) between the SuPTs, scan the combination of the CP(s) that gives the same CCM index value and find the corresponding SuPR and SSIM index value for each image.
7- Compare the SuPR of each SuPT for each image with the same CCM index and common CP(s) (if any).
8- Use the corresponding SSIM index to evaluate the processed images' similarity using original images for each SuPT.

We chose two SuPTs used in CPA CIS as a case study to show how the proposed methodology can compare different SuPTs fairly.

Case study: ASSLAR and AR Comparison

The comparison methodology is applied step by step as follows:

1- A group of 52, 8-bits, grey-scale images was chosen from the Caltech image database [95]. The CCM index of each image was calculated and found that they are in the CCM index range between 0.2 and 0.7.
2- SNM is set to 1/2 for AR, thus;
   a- the CP of the AR SuPT is "none,"
   b- no common CPs exist among the two SuPT when SNM is set to 1/2.
3- The AR SuPT is processed first.
4- The AR SuPT is applied to each image, and the corresponding SuPR, CCM, and SSIM are recorded
5- No common CPs exist among the two SuPT when SNM is set to 1/2 for AR SuPT. M-SNM and N-SNM CPs for ASSLAR are set to the optimum value (mode-6), as described in the previous section. The fall back penalty is set to 0.5 clock cycles.
6- The only CP for the ASSLAR SuPT is the threshold. So, the threshold from 1 to 100 was scanned for each image to find which threshold will give the same CCM index value as the AR SuPT. Once this threshold is found, the corresponding SuPR, CCM, and SSIM were recorded for the ASSLAR SuPT.
7- The SuPR of both SuPTs were compared for each image using CCM.
8- The similarity of the output images was compared to the original image for both SuPTs using the recorded SSIM index.

The results of comparing these two SuPTs are illustrated in the following sub-section.

The CCM Comparison Methodology Results

Figure 5.12 shows the resulting SuPR of both AR and ASSLAR SuPTs. The AR SuPT resulted in a constant SuPR as expected and equal to 4.19 with respect to the original SSR ADC. This is because the AR SuPT does not have any CPs (except for the SNM) to vary, and its SuPR depends on the bit resolution of the image, which is a fixed parameter (8-bit) for the test images. This means that the AR SuPT is not affected by the image conversion complexity as it has a fixed number of steps (61 for an 8-bit image) to finish the conversion cycle. On the other hand, the SuPR of the proposed ASSLAR SuPT comes higher than that of the AR SuPT when the CCM index is larger than 0.5; in other words, when the images become less complex. This is the expected behavior of the ASSLAR SuPT because it cares about the image conversion complexity, and its conversion speed is increased when the processed image becomes less complex.

The ASSLAR SuPR in Figure 5.12 is based on mode-6, which is the optimum CP mode that is determined in the ASSLAR. The average ASSLAR SuPR for this group of the image is 4 times faster than the SSR. As shown in Figure 5.12, SuPR can reach almost 6X if the images are very simple, i.e., higher CCM index value. The average ASSLAR enhancement of the SuPR is about 20% for the images with the CCM index larger than 0.5. SuPR of the ASSLAR increases as the image conversion

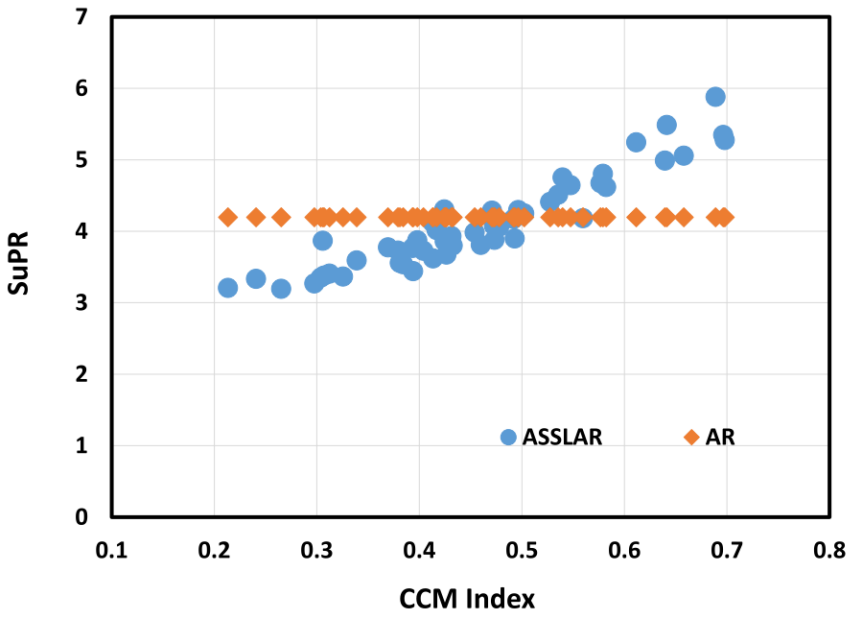


Figure 5.12. AR and ASSLAR SuPRs' comparison using the CCM comparison methodology.

complexity decreases (or the CCM index increases), which are expected, as less complex images create opportunities for SuPTs such as SSLAR to take advantage of without degrading IQ.

The SSIM index plot versus the CCM index of the test images for the AR and ASSLAR SuPTs is shown in Figure 5.13. The average value of the SSIM index for the AR SuPT is 98.6%, and it is the same for the ASSLAR SuPT using mode-6 settings. An average 20% increase in the SuPR for simple images using the ASSLAR SuPT almost did not decrease the SSIM index of the processed images.

Figure 5.14 plots the same data in Figure 5.10 after adding the AR SuPR and the corresponding SSIM to see the complete picture. As illustrated for the AR SuPT, its SuPR is always constant as long as the SNM = 0.5 to keep the SSIM is in the range of 98%. The optimum mode for the ASSLAR SuPT is mode-6, which gives the same average SuPR and SSIM as the AR. However, by looking at the standard deviation of the ASSLAR SuPR, it is found that SuPR can be higher than AR, and the SSIM can slightly be lower than the AR's SSIM. This possibility occurs only when the CCM index is larger than 0.5, because, as stated before, the ASSLAR SuPR increases when the image becomes less complex. The CCM index plays an important role in choosing which SuPT is to be used. So, for the highly complex images with a low CCM index, the AR dominates and has higher SuPR than the ASSLAR. However, if the image becomes less complex with a high CCM index, the ASSLAR becomes faster than the AR SuPT with the same SSIM index, and the IQ will not be affected. The

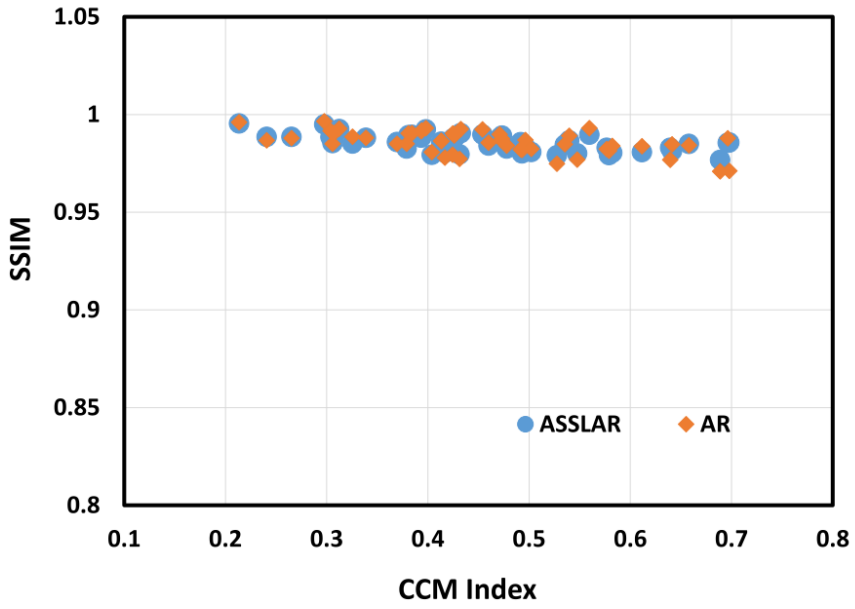


Figure 5.13. The resulting SSIM index versus the CCM index of the test images for AR and ASSLAR techniques after using the proposed CCM comparison methodology.

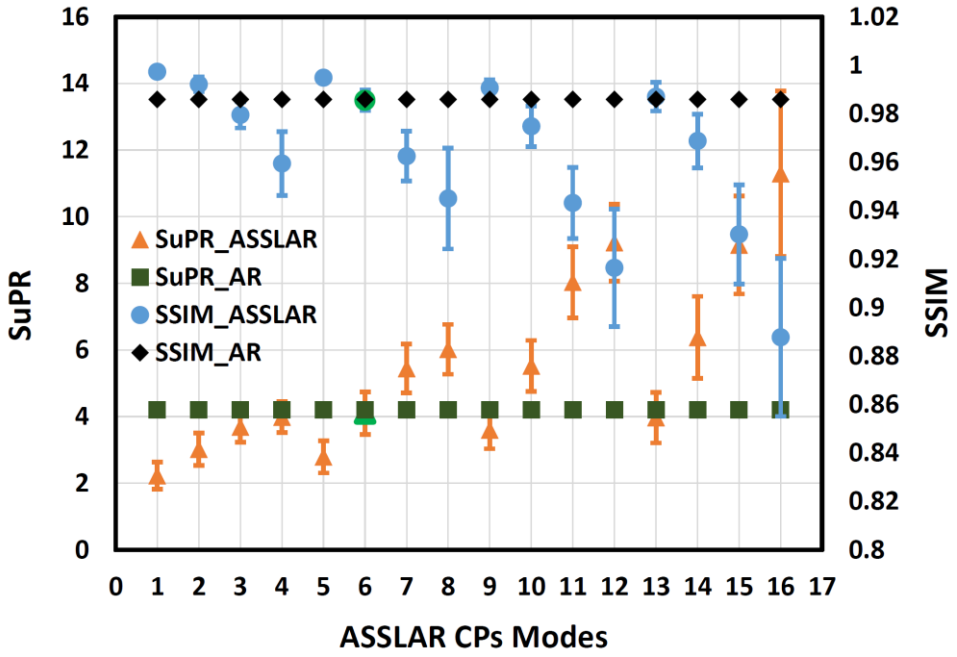


Figure 5.14. The AR and ASSLAR SuPT comparison.

resulting SuPR of this hybridization will always lead to equal or greater than the AR SuPT, resulting in the same IQ.

Figure 5.15, Figure 5.16, and Figure 5.17 show samples of low complexity images processed by the AR and ASSLAR SuPT. The AR SuPT was run using SNM = 0.5, 1, and 2 to see the effect of exceeding SNM for the AR. As can be seen in Figure 5.15 (c) and (d) through Figure 5.17, the IQ degradation is so clear and visible with a lower SSIM index, which means that SNM= 1 and 2 cannot be used in the AR SuPT to increase the SuPR. On the other hand, in Figure 5.15 (e) through Figure 5.17, the output image of the proposed ASSLAR SuPT resulted in almost the same SSIM as the AR technique but higher SuPR, as it is running at mode-6, which has the M-SNM = FB-SNM = 2.

Conclusion

The ASSLAR ADC was built on two powerful SuPTs (SSLAR and AR) to take the advantages of both and overcome their disadvantages. The ASSLAR depends on the main opportunistic property of the SSLAR SuPT to increase the SuPR based on the image complexity. The ASSLAR also inherits how the main ramp is accelerated in the AR SuPT and replaces the main and fallback ramps of the SSLAR with it. The opportunistic property of the SSLAR allowed the ASSLAR to exceed the shot-noise margin (SNM) and search for any chance to increase the SuPR based on the complexity of the input image. The CCM index was used as the metric to figure out which SuPT

should be used based on the image conversion complexity. The ASSLAR SuPT was simulated and compared based on the newly proposed comparison method that is built on the CCM and the SSIM indexes. This general method is applied to the AR and the ASSLAR SuPT as a case study. The comparison methodology resulted in proposing the conditions when using the AR or the ASSLAR. The ASSLAR showed a noticeable SuPR enhancement when the input image is less complex when the CCM index is more than 0.5, while the ASSLAR is better for the higher complex image with the index of 0.5 or less.

(a) The original image

(b) AR. SuPT, SuPR=4.19, SSIM=98.3%, SNM=0.5

(c) AR. SuPT, SuPR=8, SSIM=95.5%, SNM=1

(d) AR. SuPT, SuPR=14.2, SSIM=96.3%, SNM=2

(e) ASSLAR. SuPT, SuPR=4.7, SSIM=98.1%, Mode-6

(f) ASSLAR. SuPT, SuPR=6.13, SSIM=98.0%, Mode-10

Figure 5.15. Samples of processed images using AR and ASSLAR SuPTs and its SuPRs and SSIM indexes. (a) The original image (b) The AR with the recommended SNM=0.5 (c) The AR with SNM=1 (d) The AR with SNM=2 (e) The ASSLAR with mode-6 (f) The ASSLAR with mode-10.

(a) The original image

(b) AR. SuPT, SuPR=4.19, SSIM=98.3%, SNM=0.5

(c) AR. SuPT, SuPR=8, SSIM=94.8%, SNM=1

(d) AR. SuPT, SuPR=14.2, SSIM=88.7%, SNM=2

(e) ASSLAR. SuPT, SuPR=4.95, SSIM=98.1%, Mode-6

(f) ASSLAR. SuPT, SuPR=6.55, SSIM=96.7%, Mode-10

Figure 5.16. Samples of processed images using AR and ASSLAR SuPTs and its SuPRs and SSIM indexes. (a) The original image (b) The AR with the recommended SNM=0.5 (c) The AR with SNM=1 (d) The AR with SNM=2 (e) The ASSLAR with mode-6 (f) The ASSLAR with mode-10.

(a) The original image

(b) AR. SuPT, SuPR=4.19, SSIM=98.7%, SNM=0.5

(c) AR. SuPT, SuPR=8, SSIM=96%, SNM=1

(d) AR. SuPT, SuPR=14.2, SSIM=91.6%, SNM=2

(e) ASSLAR. SuPT, SuPR=5.34, SSIM=98.5%, Mode-6

(f) ASSLAR. SuPT, SuPR=6.9, SSIM=97.5%, Mode-10

Figure 5.17. Samples of processed images using AR and ASSLAR SuPTs and its SuPRs and SSIM indexes. (a) The original image (b) The AR with the recommended SNM=0.5 (c) The AR with SNM=1 (d) The AR with SNM=2 (e) The ASSLAR with mode-6 (f) The ASSLAR with mode-10.

Chapter 6: Contributions and Future Work

In this chapter, the main contribution and possible future research and development directions and works are outlined.

Main Contributions

This research resulted in advancements in the field of CMOS image sensors. Speed up techniques for integrating (ramp) type ADC for CIS with CPA are proposed and implemented. The new ADC topologies are called the single-slope look-ahead ramp (SSLAR) ADC and accelerated single-slope look-ahead ramp (ASSLAR) ADC. Measurements of the SSLAR ADC in a 200 × 150 pixel CIS showed that six times (6x) frame rate improvement could be achieved while reducing power consumption by 13% without compromising image quality. The modification and development of the ASSLAR are performed by combining the SSLAR and accelerated ramp (AR) speed-up techniques (SuPT) to further improve SSLAR ADC performance. It is shown that the ASSLAR SuPT can achieve an additional 20% speed improvement comparing with the AR for the same image quality (with SSIM index over 98%) and less complex images (with CCM index of 0.5 or more).

A new reference-free image quality (IQ) and complexity metric, called conversion complexity metric (CCM), was proposed, developed, and tested using hundreds of reference and standard database images. The new metric has proven to be bounded, monotonic, and achieves 99% linearity and 316% sensitivity. It provides a computationally efficient single-image quality metric that no other metrics provide for CIS to intelligently adjust and optimize on-chip analog and digital signal processing operations.

Using the new and existing IQ metrics, a new performance comparison methodology was proposed to set a fair comparison between the different SuPTs used in CIS to enhance the conversion speed of integrating (ramp) type ADCs. This methodology was used to compare the AR and ASSLAR SuPTs in a case study. This study resulted in developing a hybrid accelerated ramp ADC technique in which the AR for the more complex images (with CCM index 0.5 or less) while the ASSLAR for the less complex images (with CCM index 0.5 or more) to end up with a high-performance SuPT without affecting the output image quality.

The Future Work

This research's outcomes open new research topics to expand the scope and efficiency of several fields, including, but not limited to, hardware-software co-design, power-efficient CIS design, image and video processing, data-driven intelligent electronics, integrated power management, and smart imaging systems. For example, the new CCM index levels the field of algorithmic efficiency of

different techniques (i.e., quantization, compression, conversion, etc.) without requiring a reference image providing image quality and complexity index of any image, still or moving. A simple calculation method of the CCM index gives it an advantage over other techniques. This simplicity makes it a perfect candidate to be integrated with video systems to assess the imaging and scene complexities to intelligently adjust system or circuit level performance (i.e., power, resolution, speed, resource allocation, etc.) parameters. Artificial intelligent (AI) driven next-generation systems to require and find many applications of using simple image/scene/video quality and complexity metrics such as the CCM.

The proposed CCM index can also find many communication and compression applications where image quality has to be assessed with and without an original image to adjust system parameters such as compression ratio, modulation bandwidth, etc.

In the image processing field, standard images have been used for many years. As shown in our research, these images have very balanced CCM index values that make them suitable for such research that no other IQ indexes provide. The question of "*why everybody used LENA or BABOON or BARBARA images in their image processing research?*" can easily be answered by looking at their CCMS indexes. It is basically because the CCM is a reference-free IQ and complexity metric and provides a direct assessment of them. Such an IQ metric has many exciting applications in the field of image processing.

The proposed CCM index also levels the field for a fair comparison of different image or signal processing techniques that no other metric could provide. This is shown in one case study in the field of SuPTs for CIS ADCs in this research. The presented and future CCM base assessment methodologies make the picture clear for researchers to choose the right technique for the specific application they are investigating. Thus, the CCM index has the potential to becomes a new industry standard IQ metric for researchers to discover how to create or choose an image to be a standard image for the image/signal/video processing and CIS research fields.

The ASSLAR SuPT was proven to be better over the existing AR SuPTs by using the proposed fair comparison methodology. It is shown theoretically using many images and through simulations. However, it needs to be implemented in silicon for final verification that is left as future work. In this effort, an efficient way to calculate the CCM index in silicon should be developed and integrated with the new ASSLAR based CIS chip to support the smart decision to improve power consumption. This implementation will include the ASSLAR technique as the main SuPT for the

ramp ADC, and it will be switched to the AR SuPT by overriding the ASSLAR CPs to set it to the same CPs of the AR SuPT. This overriding will be built on the CCM index of the input image.

The new SuPR comparison methodology proposed for CIS should be applied to most or all the existing SuPT to show how effective they are. By doing this, some SuPTs which are thought to be the best may lose value while other SuPT may appear better than it was perceived before. Also, new SuPTs will emerge based on this fair comparison methodology. This requires comprehensive research in this area.

Finally, the CCM index is designed for smart CIS. It should be implemented on any CIS to support it to set up its own CPs to control its SuPR intelligently and the output image quality. This intelligent control will offer power consumption saving, faster conversion speed, and maintaining the IQ not affected. The input image conversion complexity will figure out the optimum CPs that will maximize the SuPR and minimize the IQ distortion intelligentially. How will all be achieved and implemented require further research and left as future work.

References

[1] M. R. Elmezayen, B. Wu, and S. U. Ay, "Single-Slope Look-Ahead Ramp ADC for CMOS Image Sensors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–10, 2020, DOI: 10.1109/TCSI.2020.3007882.

[2] M. R. Elmezayen and S. U. Ay, "Image Quality and Complexity Metric for Smart CMOS Image Sensors," in *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2020, pp. 265–268, DOI: 10.1109/MWSCAS48704.2020.9184591.

[3] M. R. Elmezayen and S. U. Ay, "A New blind image conversion complexity metric for intelligent CMOS image sensors," *IET Image Processing*, 2020, DOI: to be published.

[4] M. R. Elmezayen and S. U. Ay, "A Simple Image Quality and Complexity Metric for Smart Image Sensor Evaluation," *IEEE Access*, 2020, DOI: to be published.

[5] M. R. Elmezayen and S. U. Ay, "Ramp ADC Speed-up Techniques for CMOS Image Sensors with New Image Quality Metric," presented at the IEEE Workshop on Microelectronics and Electron Devices (WMED), Boise, ID, USA, Apr. 2021, DOI: to be published.

[6] E. R. Fossum, "CMOS image sensors: electronic camera-on-a-chip," *IEEE Transactions on Electron Devices*, vol. 44, no. 10, pp. 1689–1698, Oct. 1997, doi: 10.1109/16.628824.

[7] W. S. Boyle and G. E. Smith, "Charge Coupled Semiconductor Devices," *Bell System Technical Journal*, vol. 49, no. 4, pp. 587–593, 1970, doi: 10.1002/j.1538-7305.1970.tb01790.x.

[8] E. R. Fossum, "Active pixel sensors: Are CCDs dinosaurs?," in *Charge-Coupled Devices and Solid State Optical Sensors III*, 1993, vol. 1900, pp. 2–15.

[9] G. P. Weckler, "Operation of p-n Junction Photodetectors in a Photon Flux Integrating Mode," *IEEE Journal of Solid-State Circuits*, vol. 2, no. 3, pp. 65–73, Sep. 1967, doi: 10.1109/JSSC.1967.1049795.

[10] R. H. Dyck and G. P. Weckler, "Integrated arrays of silicon photodetectors for image sensing," *IEEE Transactions on Electron Devices*, vol. 15, no. 4, pp. 196–201, Apr. 1968, doi: 10.1109/T-ED.1968.16166.

[11] P. J. W. Noble, "Self-scanned silicon image detector arrays," *IEEE Transactions on Electron Devices*, vol. 15, no. 4, pp. 202–209, Apr. 1968, doi: 10.1109/T-ED.1968.16167.

[12] P. W. Fry, P. J. W. Noble, and R. J. Rycroft, "Fixed-pattern noise in photomatrices," *IEEE Journal of Solid-State Circuits*, vol. 5, no. 5, pp. 250–254, Oct. 1970, doi: 10.1109/JSSC.1970.1050122.

[13] S. Mendis, S. E. Kemeny, and E. R. Fossum, "CMOS active pixel image sensor," *IEEE Transactions on Electron Devices*, vol. 41, no. 3, pp. 452–453, Mar. 1994, doi: 10.1109/16.275235.

[14] S. K. Mendis *et al.*, "CMOS active pixel image sensors for highly integrated imaging systems," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 2, pp. 187–197, Feb. 1997, doi: 10.1109/4.551910.

[15] S. K. Mendis, S. E. Kemeny, and E. R. Fossum, "A 128×128 CMOS active pixel image sensor for highly integrated imaging systems," *Electron Devices Meeting IEDM '93*, p. pp.583,586, Dec. 1993.

[16] S. U. Ay, "Photodiode Peripheral Utilization Effect on CMOS APS Pixel Performance," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 6, pp. 1405–1411, Jul. 2008, doi: 10.1109/TCSI.2008.918001.

[17] S. U. Ay, *Large Format Cmos Image Sensors: Performance and Design*. Saarbrücken: VDM, Verlag Dr. Müller, 2008.

[18] Y. Nitta *et al.*, "High-Speed Digital Double Sampling with Analog CDS on Column Parallel ADC Architecture for Low-Noise Active Pixel Sensor," in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, Feb. 2006, pp. 2024–2031, doi: 10.1109/ISSCC.2006.1696261.

[19] S. Yoshihara *et al.*, “A 1/1.8-inch 6.4 MPixel 60 frames/s CMOS Image Sensor With Seamless Mode Change,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2998–3006, Dec. 2006, doi: 10.1109/JSSC.2006.884868.

[20] S. Yoshihara *et al.*, “A 1/1.8-inch 6.4MPixel 60 frames/s CMOS Image Sensor with Seamless Mode Change,” in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, San Francisco, CA, USA, Feb. 2006, pp. 1984–1993, doi: 10.1109/ISSCC.2006.1696257.

[21] T. Toyama *et al.*, “A 17.7Mpixel 120fps CMOS image sensor with 34.8Gb/s readout,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, Feb. 2011, pp. 420–422, doi: 10.1109/ISSCC.2011.5746379.

[22] Oh-Bong Kwon *et al.*, “A novel double slope analog-to-digital converter for a high-quality 640/spl times/480 CMOS imaging system,” in *ICVC ‘99. 6th International Conference on VLSI and CAD (Cat. No.99EX361)*, Seoul, South Korea, South Korea, Oct. 1999, pp. 335–338, doi: 10.1109/ICVC.1999.820923.

[23] W. Yang, O.-B. Kwon, J.-I. Lee, G.-T. Hwang, and S.-J. Lee, “An integrated 800/spl times/600 CMOS imaging system,” in *1999 IEEE International Solid-State Circuits Conference. Digest of Technical Papers. ISSCC. First Edition (Cat. No.99CH36278)*, San Francisco, CA, USA, Feb. 1999, pp. 304–305, doi: 10.1109/ISSCC.1999.759261.

[24] Otaka and Toshinori, “12-bit column-parallel ADC with accelerated ramp,” Karuizawa, Nagano, Japan, 2005, pp. 177–180.

[25] M.F.Snoeij, “A Low-Power Column-Parallel 12-bit ADC for CMOS Imagers,” Karuizawa, Nagano, Japan, Jun. 2005, p. R22.

[26] D. Zhang, R. Yassine, L. Truong, J. Rysinski, D. Van Blerkom, and B. Mansoorian, “Single Slope ADC with On-chip Accelerated Continuous-time Differential Ramp Generator for Low Noise Column-Parallel CMOS Image Sensor,” *2013 Intl. Image Sensor Workshop (IISW)*, 2013.

[27] J. Bergey, S. Bagwell, J. Law, and W. Law, “Real-time Calibration of a 14-Bit Single Slope ADC with 290MHz On-chip Accelerated Ramp Generator for Column-Parallel Image Sensors,” *2013 Intl. Image Sensor Workshop (IISW)*, 2013.

[28] L. Lindgren, “A New Simultaneous Multislope ADC Architecture for Array Implementations,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 9, pp. 921–925, Sep. 2006, doi: 10.1109/TCSII.2006.881817.

[29] Y. Oike *et al.*, “8.3 M-Pixel 480-fps Global-Shutter CMOS Image Sensor with Gain-Adaptive Column ADCs and Chip-on-Chip Stacked Integration,” *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 985–993, Apr. 2017, doi: 10.1109/JSSC.2016.2639741.

[30] H. Totsuka *et al.*, “6.4 An APS-H-Size 250Mpixel CMOS image sensor using column single-slope ADCs with dual-gain amplifiers,” in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016, pp. 116–117, doi: 10.1109/ISSCC.2016.7417934.

[31] D. Liu, W. Lu, Z. Chen, Y. Zhang, S. Lei, and G. Tan, “A 14-bit differential-ramp single-slope column-level ADC for 640×512 uncooled infrared imager,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, Montréal, QC, Canada, May 2016, pp. 1922–1925, doi: 10.1109/ISCAS.2016.7538949.

[32] I. Park, C. Park, J. Cheon, and Y. Chae, “A 76mW 500fps VGA CMOS Image Sensor with Time-Stretched Single-Slope ADCs Achieving 1.95e - Random Noise,” in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, Feb. 2019, pp. 100–102, doi: 10.1109/ISSCC.2019.8662388.

[33] J. Lee, S. Lim, and G. Han, *A 10b Column-wise Two-step Single-slope ADC for High-speed CMOS Image Sensor*. 2007.

[34] S. Lim, J. Lee, D. Kim, and G. Han, “A High-Speed CMOS Image Sensor With Column-Parallel Two-Step Single-Slope ADCs,” *IEEE Transactions on Electron Devices*, vol. 56, no. 3, pp. 393–398, Mar. 2009, doi: 10.1109/TED.2008.2011846.

[35] M. F. Snoeij, P. Donegan, A. J. P. Theuwissen, K. A. A. Makinwa, and J. H. Huijsing, "A CMOS Image Sensor with a Column-Level Multiple-Ramp Single-Slope ADC," in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, Feb. 2007, pp. 506–618, doi: 10.1109/ISSCC.2007.373516.

[36] M. F. Snoeij, A. J. P. Theuwissen, K. A. A. Makinwa, and J. H. Huijsing, "Multiple-Ramp Column-Parallel ADC Architectures for CMOS Image Sensors," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, pp. 2968–2977, Dec. 2007, doi: 10.1109/JSSC.2007.908720.

[37] F. Tang, D. G. Chen, B. Wang, and A. Bermak, "Low-Power CMOS Image Sensor Based on Column-Parallel Single-Slope/SAR Quantization Scheme," *IEEE Transactions on Electron Devices*, vol. 60, no. 8, pp. 2561–2566, Aug. 2013, doi: 10.1109/TED.2013.2268207.

[38] M. K. Kim, S. K. Hong, and O. K. Kwon, "An Area-Efficient and Low-Power 12-b SAR/Single-Slope ADC Without Calibration Method for CMOS Image Sensors," *IEEE Transactions on Electron Devices*, vol. 63, no. 9, pp. 3599–3604, Sep. 2016, doi: 10.1109/TED.2016.2587721.

[39] D. Levski, M. Wäny, and B. Choubey, "A 1- μ s Ramp Time 12-bit Column-Parallel Flash TDC-Interpolated Single-Slope ADC With Digital Delay-Element Calibration," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 54–67, Jan. 2019, doi: 10.1109/TCSI.2018.2846592.

[40] I. Takayanagi and J. Nakamura, "High-Resolution CMOS Video Image Sensors," *Proceedings of the IEEE*, vol. 101, no. 1, pp. 61–73, Jan. 2013, doi: 10.1109/JPROC.2011.2178569.

[41] A. Krymski, D. Van Blerkom, A. Andersson, N. Bock, B. Mansoorian, and E. R. Fossum, "A high speed, 500 frames/s, 1024/spl times/1024 CMOS active pixel sensor," in *1999 Symposium on VLSI Circuits. Digest of Papers (IEEE Cat. No.99CH36326)*, Jun. 1999, pp. 137–138, doi: 10.1109/VLSIC.1999.797263.

[42] R. Panicacci, B. Pain, Z. Zhou, J. Nakamura, and E. R. Fossum, "Progress in voltage and current mode on-chip analog-to-digital converters for CMOS image sensors," in *Solid State Sensor Arrays and CCD Cameras*, Mar. 1996, vol. 2654, pp. 63–71, doi: 10.1117/12.236120.

[43] K. Chen, M. Afghani, P. E. Danielsson, and C. Svensson, "PASIC: A processor-A/D converter-sensor integrated circuit," in *1990 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 1990, pp. 1705–1708 vol.3, doi: 10.1109/ISCAS.1990.111947.

[44] C. Jansson, P. Ingelhag, C. Svensson, and R. Forchheimer, "An addressable 256×256 photodiode image sensor array with an 8-bit digital output," *Analog Integr Circ Sig Process*, vol. 4, no. 1, pp. 37–49, Jul. 1993, doi: 10.1007/BF01240678.

[45] A. Dickinson, S. Mendis, D. Inglis, K. Azadet, and E. Fossum, "CMOS Digital Camera With Parallel Analog-ta-Digital Conversion Architecture," in *Proc. IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors (CCD&AIS)*, Dana Point, CA USA, Apr. 1995, p. 32.

[46] A. Simoni, A. Sartori, M. Gottardi, and A. Zorat, "An Intelligent Linear Image Sensor," in *ESSCIRC '93: Nineteenth European Solid-State Circuits Conference*, Sep. 1993, vol. 1, pp. 246–249.

[47] S. K. Mendis, B. Pain, R. H. Nixon, and E. R. Fossum, "Design of a low-light-level image sensor with on-chip sigma-delta analog-to-digital conversion," in *Charge-Coupled Devices and Solid State Optical Sensors III*, Jul. 1993, vol. 1900, pp. 31–39, doi: 10.1117/12.148606.

[48] B. Fowler, A. El Gamal, and D. X. D. Yang, "A CMOS area image sensor with pixel-level A/D conversion," in *Proceedings of IEEE International Solid-State Circuits Conference - ISSCC '94*, Feb. 1994, pp. 226–227, doi: 10.1109/ISSCC.1994.344659.

[49] J. Nakamura, B. Pain, T. Nomoto, T. Nakamura, and E. R. Fossum, "On-focal-plane signal processing for current-mode active pixel sensors," *IEEE Transactions on Electron Devices*, vol. 44, no. 10, pp. 1747–1758, Oct. 1997, doi: 10.1109/16.628832.

[50] Y. Chae and G. Han, "Low Voltage, Low Power, Inverter-Based Switched-Capacitor Delta-Sigma Modulator," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 2, pp. 458–472, Feb. 2009, doi: 10.1109/JSSC.2008.2010973.

[51] Y. Chae *et al.*, "A 2.1Mpixel 120frame/s CMOS image sensor with column-parallel ΔΣ ADC architecture," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, Feb. 2010, pp. 394–395, doi: 10.1109/ISSCC.2010.5433974.

[52] A. Simoni, G. Torelli, F. Maloberti, A. Sartori, M. Gottardi, and L. Gonzo, "256 × 256-pixel CMOS digital camera for computer vision with 32 algorithmic ADCs on board," *IEE Proceedings - Circuits, Devices and Systems*, vol. 146, no. 4, pp. 184–190, Aug. 1999.

[53] S. Decker, D. McGrath, K. Brehmer, and C. G. Sodini, "A 256/spl times/256 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, Dec. 1998, doi: 10.1109/4.735551.

[54] M. Furuta, S. Kawahito, T. Inoue, and Y. Nishikawa, "A cyclic A/D converter with pixel noise and column-wise offset canceling for CMOS image sensors," in *Solid-State Circuits Conference, 2005. ESSCIRC 2005. Proceedings of the 31st European*, Sep. 2005, pp. 411–414, doi: 10.1109/ESSCIR.2005.1541647.

[55] S. Lim *et al.*, "A 240-frames/s 2.1-Mpixel CMOS Image Sensor With Column-Shared Cyclic ADCs," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 9, pp. 2073–2083, Sep. 2011, doi: 10.1109/JSSC.2011.2144010.

[56] K. Kitamura, "A 33 Mpixel,120 fps CMOS Image Sensor for UDTV Application with Two-stage Column-Parallel Cyclic ADCs," in *Proc. International Image Sensor Workshop (IISW)*, Hokkaido, Japan, Jun. 2011, p. R62, pp.343–346.

[57] K. Kitamura *et al.*, "A 33-Megapixel 120-Frames-Per-Second 2.5-Watt CMOS Image Sensor With Column-Parallel Two-Stage Cyclic Analog-to-Digital Converters," *IEEE Transactions on Electron Devices*, vol. 59, no. 12, pp. 3426–3433, Dec. 2012, doi: 10.1109/TED.2012.2220364.

[58] K. Kitamura, "A 14-bit 33-Mpixel 120-fps Image Sensor with DMOS Capacitors in 90-nm/65-nm CMOS," Vaals, The Netherlands, Jun. 2015.

[59] T. Yasue *et al.*, "A 1.7-in, 33-Mpixel, 120-frames/s CMOS Image Sensor With Depletion-Mode MOS Capacitor-Based 14-b Two-Stage Cyclic A/D Converters," *IEEE Transactions on Electron Devices*, vol. 63, no. 1, pp. 153–161, Jan. 2016, doi: 10.1109/TED.2015.2451700.

[60] T. Arai *et al.*, "A 1.1μm 33Mpixel 240fps 3D-Stacked CMOS Image Sensor with 3-Stage Cyclic-Based Analog-to-Digital Converters," Mar. 2016, pp. 21–24, doi: 10.11485/itetr.40.12.0_21.

[61] F. Z. Nelson, M. N. Alam, and S. U. Ay, "A Single-Slope Look-Ahead Ramp (SSLAR) ADC for Column Parallel CMOS Image Sensors," in *IEEE Workshop on Microelectronics and Electron Devices, 2009. WMED 2009*, Apr. 2009, pp. 1–4, doi: 10.1109/WMED.2009.4816152.

[62] F. Z. Nelson and S. U. Ay, "Integration of a New Column-Parallel ADC Technology on CMOS Image Sensor," in *2010 IEEE Workshop on Microelectronics and Electron Devices*, Apr. 2010, pp. 1–4, doi: 10.1109/WMED.2010.5453748.

[63] S. Balagopal and S. U. Ay, "An on-chip ramp generator for single-slope look ahead ramp (SSLAR) ADC," in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*, Aug. 2009, pp. 373–376, doi: 10.1109/MWSCAS.2009.5236076.

[64] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, May 2002, vol. 4, pp. IV-3313-IV–3316, doi: 10.1109/ICASSP.2002.5745362.

[65] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Transactions on Communications*, vol. 43, no. 12, pp. 2959–2965, Dec. 1995, doi: 10.1109/26.477498.

[66] I. Avcibas, B. Sankur, and K. Sayood, "Statistical evaluation of image quality measures," *J. Electronic Imaging*, vol. 11, pp. 206–223, 2002, doi: 10.1117/1.1455011.

[67] H. Marmolin, “Subjective MSE Measures,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 3, pp. 486–489, May 1986, doi: 10.1109/TSMC.1986.4308985.
[68] V. Kumar and V. S. Bawa, “No reference image quality assessment metric based on regional mutual information among images,” *arXiv:1901.05811 [cs, eess]*, Jan. 2019, Accessed: Oct. 27, 2019. [Online]. Available: http://arxiv.org/abs/1901.05811.
[69] S. U. Ay, “Analog-to-digital converter with programmable ramp generator,” US9040889B2, May 26, 2015.
[70] A. Horé and D. Ziou, “Image Quality Metrics: PSNR vs. SSIM,” in *2010 20th International Conference on Pattern Recognition*, Aug. 2010, pp. 2366–2369, doi: 10.1109/ICPR.2010.579.
[71] R. Kreis, “Issues of spectral quality in clinical 1H-magnetic resonance spectroscopy and a gallery of artifacts,” *NMR in Biomedicine*, vol. 17, no. 6, pp. 361–381, 2004, doi: 10.1002/nbm.891.
[72] T. B. Nguyen and D. Ziou, “Contextual and non-contextual performance evaluation of edge detectors,” *Pattern Recognition Letters*, vol. 21, no. 9, pp. 805–816, Aug. 2000, doi: 10.1016/S0167-8655(00)00045-3.
[73] M. Cadik and P. Slavik, “Evaluation of two principal approaches to objective image quality assessment,” in *Proceedings. Eighth International Conference on Information Visualisation*, Jul. 2004, pp. 513–518, doi: 10.1109/IV.2004.1320193.
[74] M. Pedersen and J. Y. Hardeberg, “Survey of full-reference image quality metrics,” *Høgskolen i Gjøviks rapportserie 5, The Norwegian Color Research Laboratory (Gjøvik University College)*, p. 74, Jun. 2009.
[75] M. Pedersen, “Evaluation of 60 full-reference image quality metrics on the CID:IQ,” in *2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 1588–1592, doi: 10.1109/ICIP.2015.7351068.
[76] M. Pedersen and J. Y. Hardeberg, “Full-Reference Image Quality Metrics: Classification and Evaluation,” *CGV*, vol. 7, no. 1, pp. 1–80, Mar. 2012, doi: 10.1561/0600000037.
[77] P. Le Callet and D. Barba, “A robust quality metric for color image quality assessment,” in *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, Sep. 2003, vol. 1, pp. I–437, doi: 10.1109/ICIP.2003.1246992.
[78] D. M. Chandler and S. S. Hemami, “VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images,” *IEEE Trans. on Image Process.*, vol. 16, no. 9, pp. 2284–2298, Sep. 2007, doi: 10.1109/TIP.2007.901820.
[79] Z. Wang and A. C. Bovik, “Modern Image Quality Assessment,” *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 2, no. 1, pp. 1–156, Jan. 2006, doi: 10.2200/S00010ED1V01Y200508IVM003.
[80] K.-H. Thung and P. Raveendran, “A survey of image quality measures,” in *2009 International Conference for Technical Postgraduates (TECHPOS)*, Dec. 2009, pp. 1–4, doi: 10.1109/TECHPOS.2009.5412098.
[81] X. Zhang and B. A. Wandell, “A spatial extension of CIELAB for digital color-image reproduction,” *J. Soc. Inf. Display*, vol. 5, no. 1, p. 61, 1997, doi: 10.1889/1.1985127.
[82] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.
[83] H. R. Sheikh and A. C. Bovik, “Image information and visual quality,” *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, Feb. 2006, doi: 10.1109/TIP.2005.859378.
[84] A. K. Tripathi, S. Mukhopadhyay, and A. K. Dhara, “Performance metrics for image contrast,” in *2011 International Conference on Image Information Processing*, Nov. 2011, pp. 1–4, doi: 10.1109/ICIIP.2011.6108900.
[85] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-Reference Image Quality Assessment in the Spatial Domain,” *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, Dec. 2012, doi: 10.1109/TIP.2012.2214050.

[86] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a ‘Completely Blind’ Image Quality Analyzer,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, Mar. 2013, doi: 10.1109/LSP.2012.2227726.

[87] Venkatanath N, Praneeth D, Maruthi Chandrasekhar Bh, S. S. Channappayya, and S. S. Medasani, “Blind image quality evaluation using perception based features,” in *2015 Twenty First National Conference on Communications (NCC)*, Feb. 2015, pp. 1–6, doi: 10.1109/NCC.2015.7084843.

[88] B. GIROD, “What’s wrong with mean-squared error?,” *Digital Images and Human Vision*, pp. 207–220, 1993.

[89] Zhou Wang and A. C. Bovik, “A universal image quality index,” *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, Mar. 2002, doi: 10.1109/97.995823.

[90] B. Kumar, S. B. Kumar, and C. Kumar, “Development of improved SSIM quality index for compressed medical images,” in *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, Dec. 2013, pp. 251–255, doi: 10.1109/ICIIP.2013.6707593.

[91] R. A. Peters, II, and R. N. Strickland, *Image Complexity Metrics for Automatic Target Recognizers*. 1990.

[92] J. Perkiö and A. Hyvärinen, “Modelling Image Complexity by Independent Component Analysis, with Application to Content-Based Image Retrieval,” in *Artificial Neural Networks – ICANN 2009*, Berlin, Heidelberg, 2009, pp. 704–714, doi: 10.1007/978-3-642-04277-5_71.

[93] K. Findlater *et al.*, “SXGA pinned photodiode CMOS image sensor in 0.35 /spl mu/m technology,” in *2003 IEEE International Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC.*, Feb. 2003, pp. 218–489 vol.1, doi: 10.1109/ISSCC.2003.1234274.

[94] “Add noise to image - MATLAB imnoise.” https://www.mathworks.com/help/images/ref/imnoise.html (accessed Nov. 14, 2019).

[95] “CALTECH Computational Vision group, image archive background.” http://www.vision.caltech.edu/archive.html (accessed Nov. 03, 2019).

Appendix A – Some Codes Subroutines

The SSLAR Subroutine

```
for tresh=1:last_tresh
for step=1:last_step
    num=0;
    for (row=1:nrow) % rows scan loop
       njump(row)=0; % number of ADC code steps per row>>>>>
       ramp=0;       % ramp reset each row
       while (ramp<adc)  %ramp from 0 to 255
                num=0;
                for col=1:ncol    %column scan
                    if (pic(row,col)>=ramp & pic(row,col)<(ramp+step) &
step>1)
                        num=num+1;
                    end;
                end;

                if (num>0) % some number of pixels in step range
                    if(num<=tresh) %jump approved if true
                        for col=1:ncol                %column loop
                            if(pic(row,col)>=ramp &
pic(row,col)<(ramp+step))
                                %picnew(row,col)=ramp+h;
                                picnew(row,col)=ramp+floor((step)*0.5);
                                %picnew(row,col)=ramp+ceil((step)*0.5);
                            end;
                        end;
                        njump(row)=njump(row)+1;
                    else % num>tresh and jump is not aproved, cost has to
be paid
                        njump(row)=njump(row)+cost+step;
                    end;
                else
                  njump(row)=njump(row)+1;
                end;
                % increment the ramp
                ramp=ramp+step;
        end;
    end;

%-------------finding speedup-------------------
total_save =sum(njump);
SSLAR_SUP(step,tresh,file_no) = nrow*adc/total_save;
        end;
    end;
```

The ASSLAR Subroutine

```
for tresh=1:last_tresh
num=0;
    for (row=1:nrow) % rows scan loop
       njump(row)=0; % number of ADC code steps per row
       ramp=0;       % ramp reset each row
       while (ramp<adc)  %ramp from 0 to 255
          % -choose step size based on ramp value
            if (ramp==0) step=1; h=1;
            elseif (ramp>=1 & ramp<=adc-1)
                step= round((sqrt(ramp)*M)); % use round if N<=2
                h=step/2;
            end;

                num=0;
                % check if number of columns having value btw ramp and
ramp+step-1
            for col=1:ncol    %column scan
                if (pic(row,col)>=ramp & pic(row,col)<(ramp+step) &
step>1)
                    num=num+1;
                end;
            end;
            if (num>0) % some number of pixels in step range
                if(num<=tresh) %jump approved if true
                    for col=1:ncol                %column loop
                        if(pic(row,col)>=ramp & pic(row,col)<(ramp+step))
                            picnew(row,col)=ramp+h;
                        end;
                    end;
                    njump(row)=njump(row)+1;
                else % num>tresh and jump is not aproved, cost has to be
paid
                    fbstep=0;
                    ramp2=0;
                    while (ramp2<step)
                        if (ramp2==0) step2=1; h2=1;
                        elseif (ramp2>=1 & ramp2<=step)
                            step2= round((sqrt(ramp2)*N)); % use round if
M<=2
                            h2=step2/2;
                        end;
                        for col=1:ncol                %column loop
                            if(pic(row,col)>=(ramp+ramp2) &
pic(row,col)<(ramp+ramp2+step2))
                                picnew(row,col)=(ramp+ramp2)+h2;
                            end;
                        end;
                        fbstep=fbstep+1;
                        ramp2=ramp2+step2;
                        stepout2(ramp2+1)=step2;
                    end;
                    njump(row)=njump(row)+cost+fbstep;
                    %ramp=ramp-step;
                end;
            else
```

```
                njump(row)=njump(row)+1;
            end;
            ramp=ramp+step;
        end;
    end;
%-------------finding speedup-------------------
total_save =sum(njump);
ASSLAR_SUP(file_no,tresh) = nrow*adc/total_save;
end;
```

The AR Subroutine

```
    for (row=1:nrow) % rows scan loop
       njump(row)=0; % number of ADC code steps per row
       ramp=0;       % ramp reset each row
       while (ramp<adc)  %ramp from 0 to 255
                % -choose step size based on ramp value
            if (ramp==0                    ) step=1;    end;
            if (ramp>=1 & ramp<=adc-1) step= round((sqrt(ramp)*0.5)); end;
            for col=1:ncol                %column loop
                if(pic(row,col)>=ramp & pic(row,col)<(ramp+step))
                 picnew(row,col)=ramp;
                end;
            end;
            stepout(ramp+1)=step;
            njump(row)=njump(row)+1;
            ramp=ramp+step;   % increment the ramp
        end;
    end;
total_save =sum(njump);
ACC_SUP(file_no) = nrow*adc/total_save;
```

The CCM Subroutine

```
row_hist=zeros(nrow,adc);       % row histogram array
img_hist=double(zeros(1,adc));  % image histogram array
zero_hist=zeros(nrow,adc);      % Zeros histogram array/row
img_zero_hist=zeros(1,adc);     % Zeros histogram array/image
cx_row=zeros(nrow,adc);
for row=1:nrow
% find row histogram and whole image histogram
        for col=1:ncol
             indx=double(pic1(row,col)+1); %+1 to avoid error of Matlab as
no 0 index
             % find row histogram
             row_hist(row,indx)=row_hist(row,indx)+1;
             % find picture histogram
             img_hist(1,indx)=img_hist(1,indx)+1;
        end;

% find zeros-band histogram
        i=1;
        while i<=adc
            if row_hist(row,i)==0
                indx=1; % (reset index) but, Matlab has not (0,0) location
this will be corrected later
                while (row_hist(row,i)==0)
                    indx=indx+1;
                    i=i+1;
                    if i>adc break; end;
                end;
                zero_hist(row,indx)=zero_hist(row,indx)+1;
                img_zero_hist(1,indx)=img_zero_hist(1,indx)+1;
            end;
            i=i+1;
        end;
% shifting histogram to correct range from (0,0) to (nrow,ADC-1)
        for indx=1:adc-1
            zero_hist_f(row,indx)=zero_hist(row, indx+1);
            img_zero_hist_f(1,indx)=img_zero_hist(1,indx+1);
        end;

% calculating complexity
            for indx=1:adc-1
                if zero_hist_f(row, indx)~=0
                   cx_row(row,indx)=(indx/zero_hist_f(row, indx));
                end;
            end;
            cx_img(file_no)=(sum(sum(cx_row)))/(nrow*adc); %use if
scanning for threshold required
end;
% % Mapping function
            if adc<=ncol
               ACC_CCM(file_no)=(adc*cx_img(file_no))/(adc-1);
            else
               ACC_CCM(file_no)=(adc*((cx_img(file_no)*ncol^2)-
1)+ncol)/(adc*(ncol^2-1)+ncol*(1-ncol));
            end;
```