

## RESEARCH ARTICLE

# CCD Sensor Based Cameras for Sustainable Streaming IoT Applications With Compressed Sensing

RAMACHANDRA GAMBHEER<sup>ID</sup>, (Member, IEEE), AND M. S. BHAT<sup>ID</sup>, (Senior Member, IEEE)

National Institute of Technology Karnataka Surathkal, Surathkal 575025, India

Corresponding author: Ramachandra Gambheer (ramachandra.gambheer@ieee.org)

**ABSTRACT** This paper presents a comprehensive study of compressed sensing (CS) techniques applied to Charge Coupled Device (CCD) and Complementary Metal-Oxide Semiconductor (CMOS) sensor-based cameras. CS is a powerful technique for reducing the number of measurements required to capture high-quality images while maintaining a high signal-to-noise ratio (SNR). In this study, we propose a novel CS method for CCD and CMOS sensor-based cameras that combines a new sampling scheme with a sparsity-inducing transform and a reconstruction algorithm to achieve high-quality images with fewer measurements. This paper focuses on an efficient CCD image capturing system suitable for embedded IoT applications. Hardware implementation has been done for proof of concept with an onboard Field Programmable Gate Array (FPGA) performing the compression. This hardware module is used over a wireless network to transmit and receive images under different test conditions with both CMOS and CCD sensors. For each use case, Peak Signal to Noise Ratio (PSNR), average power, and memory usage are computed under different ambient lighting conditions from dark to very bright. The results show that, a  $640 \times 480$  CCD sensor with compressed sensing with a sparsity of 0.5, provides 13% power saving and 15% memory saving compared to uncompressed sensing in no-light condition, resulting in 25.76 dB PSNR. Whereas, in no light condition, CMOS sensor does not capture any image at all. These results shows that the CCD image capturing system with compressed sensing can be conveniently used for embedded IoT applications. The data recovery from wireless sensor network is done at a central office where computing time and processing power resources are not constrained. The weight of the CCD camera is approximately 100 grams with modular build approach.

**INDEX TERMS** CCD imager, CMOS imager, compressive sensing, dynamic range, fill-factor, global shutter, IoT, LVDS, PSNR, quantum efficiency, rolling shutter, wearable camera.

## I. INTRODUCTION

Wearable video cameras that can transmit streaming videos will be of great aid in the surveillance applications. The concern with wearable cameras is the quality of video, power dissipation and resource requirements such as memory, bandwidth etc. Video quality depends on whether the sensor is a Complementary Metal Oxide Semiconductor (CMOS) or Charge Coupled Device (CCD) [1]. Even though CMOS

sensors consume less power and provide high-quality picture frames, they fail under poor ambient lighting conditions because a minimum cut-in voltage (and hence a minimum light) is required for the CMOS image sensor to function. On the other hand, CCD sensors provide adequate quality pictures even in the darkest ambiance where the human eye cannot see anything. This is possible because of the inherent characteristics of the CCD cells. However, CCD consumes more power and hence results in higher heat dissipation and shorter battery life, making its use challenging in wearable devices. To transmit a streaming video from a wearable

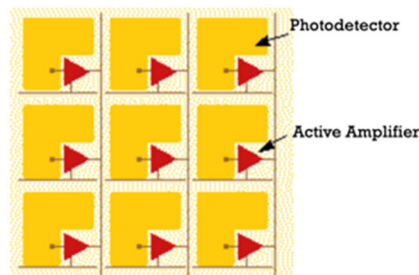
The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Pau<sup>ID</sup>.

**TABLE 1. Important characteristics Of CMOS and CCD image sensors.**

Characteristics	CCD Image Sensor	CMOS Image Sensor
Output electrical signal	Analog	Digital
Pixel Sensor (PPS/APS)	Passive	Active
Readout Speed	Relatively Low	Relatively High
Frame Rate	Low	High
Fixed Pattern Noise	Low	High
Pixel Size	Large	Small
Resolution	Low - High	Low - High
Fill Factor	High (100%)	Moderate (60-75%)
Shutter Type	Global Shutter	Rolling Shutter
Skew	No	Yes
On-Chip Voltage	Multiple Voltages	Single Voltage
Power Consumption	Relatively High (5W)	Low (few mW)
Dynamic Range	High	Moderate
Thermal Noise / Dark Current	Low	High
Smearing & Blooming Effect	More	Less
Quantum Efficiency	High	Low

device that can be reproduced at the receiver with high fidelity, we either need a higher bandwidth or incorporate video compression techniques before transmission. Need for a higher bandwidth is obviously not a choice for any sustainable applications. Hence, the only option is to perform video compression before transmission. Video compression involves image processing using a suitable processor and memory. This results in a higher power requirement and hence a need to use a heat sink. These additional components not only increase the power requirement but also the weight of the wearable device, once again making it unsustainable. In this paper, we present the results of our research using a novel technique that addresses all these issues. We used a compressive sensing technique that avoids the use of higher memory and on-board video processing, thereby limiting the power requirement. We implemented this with a CCD imager to obtain decent quality video frames under low illumination. Combining these two will make a high-quality wearable Internet of Things (IoT) camera a reality, solving the power, need for higher memory, quality, and weight issues, thus resulting in a sustainable system.

This study is divided into six sections. In second section, we presented an overview of CMOS and CCD image sensor characteristics along with their merits and demerits. In third section, we presented the requirements of an image sensor for wearable IoT applications. In fourth and fifth sections, we presented a compressed sensing and retrieval technique that can be used for video frame compression. In sixth section, we presented the hardware implementation. In seventh section, we presented the results of the work we performed by combining the image compression coupled with the CCD image sensor. We have carried out experiments with CMOS image sensor as well for comparison purpose. In the concluding section, we presented the challenges in image compression in IoT applications as a future research direction.

**FIGURE 1. Concept of fill factor; Courtesy: Silicon imaging [9].**

## II. OVERVIEW OF CCD AND CMOS IMAGERS

There is an extensive body of literature available on construction techniques for both CCD [2], [3], [4] and CMOS [1], [5], [6], [7] sensors for image capture.

Although the main purpose of CCD technology when invented in Bell Labs in 1969 was for data storage [8], its capability to transport electronic charges made it possible to use it as an image sensor to convert light to analog pixel information. Table 1 shows the various characteristics and parameters of CCD and CMOS image sensors.

From the above list, we will explain a few important characteristics of both the CCD and CMOS image sensors.

### A. ACTIVE OR PASSIVE PIXEL SENSOR

The pixel sensor can be active or passive, depending on the construction of the sensor. In a Passive Pixel Sensor (PPS), light is converted to voltage by employing photosites that convert photons into voltage. Photosites are tiny light collectors. No amplification was performed in the passive pixel sensor. In an Active Pixel Sensor (APS) [6], light is converted to voltage using an active electronic circuitry, including photodiodes and amplifiers. The CCD image sensor is built using PPS, whereas the CMOS uses both PPS and APS.

### B. FILL FACTOR

The percentage of photo-site or pixel devoted to collecting the light or the percentage of photo-site that is sensitive to light is referred to as pixel's Fill Factor (FF). A low FF requires a higher exposure time and bright light to capture an image of decent quality. CCD sensors have a 100% FF, whereas CMOS sensors have a poor FF that is much less than that of CCD sensors. This is because, in a CMOS image sensor, each photo-site includes circuitry for filtering noise and amplifying the signal; hence, the area that is sensitive to light is significantly reduced (typically 60%-75% of CCD). The concept of FF is illustrated in [9] as shown in Fig. 1.

The FF refers to the percentage of photo-sites that are sensitive to light. If the circuits cover 25% of each photo site, the sensor is said to have an FF of 75%. The higher the fill factor, the more sensitive the sensor is [9]. Recent studies [10] have shown that a fill factor of the order of 61% and slightly above is achievable in CMOS image sensors with Single Photon Avalanche Diode (SPAD) detector arrays.

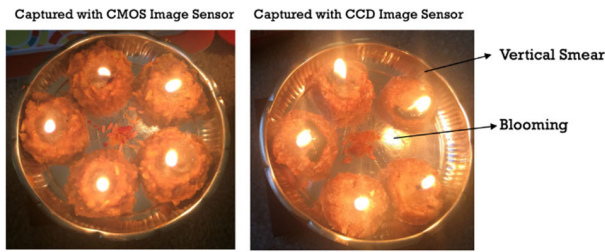


FIGURE 2. Smearing and blooming effect in CMOS & CCD.

### C. SHUTTER TYPE

Two types of shutters are used in image sensors: rolling shutters and global shutters [11]. Owing to the nature of the construction mechanism, all CCD sensors have a global shutter and all CMOS sensors have rolling shutters. Performance of rolling shutter and global shutter cameras are explained in [12].

In the rolling shutter mechanism, the pixels of each row of the frame are exposed one row at a time and then converted into digital signals one at a time. The exposure times for all the rows are the same. Whereas in global shutter, all the pixels are exposed to the light simultaneously and then read sequentially one at a time. In global shutter, since all pixels are captured at the same time, the image quality is superior to that of the rolling shutter.

### D. DYNAMIC RANGE

The dynamic range of an image sensor is defined as the ratio of the maximum achievable signal (proportional to light) to the sensor noise. The dynamic range of the image sensors can be calculated using the photon transfer curve, as detailed in [15]. The dynamic range is the range between the readable brightest and darkest pixel areas that can be captured in a single image. CCD image sensors have a better dynamic range than CMOS image sensors, because there is no minimum light required for the CCD to represent the charge.

### E. SMEARING & BLOOMING EFFECT

The phenomenon of corrupting the acquired image owing to the leakage of charge from one pixel to the adjacent pixel is known as the horizontal blooming effect or pixel saturation in image sensors. Both smearing (a type of contrast reduction) and blooming effects are more prominent in CCD than in CMOS technology. These effects are noticeable when a bright light is captured with CCD image sensor. Fig. 2 shows the smearing and blooming effect comparison in both CMOS and CCD image sensors captured with the same exposure time and ambient lighting conditions. There are several compensation algorithms proposed [16] for removing smearing effects.

### F. QUANTUM EFFICIENCY

Quantum Efficiency (QE) is a measure of the efficiency with which a sensor device converts incident light or photons

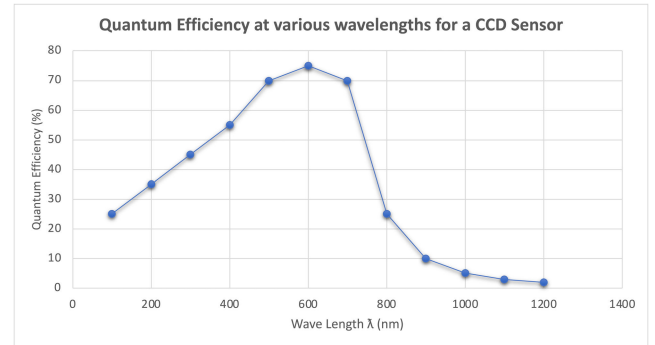


FIGURE 3. Quantum efficiency of CCD image sensor at various photon wavelengths ( $\lambda$  nm).

into electrical charge carriers or electrons. The formula for quantum efficiency is:

$$QE = (\text{number of charge carriers generated}) / (\text{number of photons incident}) \times 100\%$$

For example, if a sensor produces  $X$  electrons when it is exposed to  $X$  photons, the QE is said to be 100%. Generally, the QE is measured for various photon wavelengths for a given image sensor. The calculation of quantum efficiency is influenced by various factors, such as temperature, voltage bias, and material quality. Therefore, it is essential to perform the measurements under controlled conditions to obtain accurate results. The QE of a CCD sensor can be modeled as a product of three components: the quantum efficiency of the sensor's photodiode, the reflectance of the sensor's surface, and the transmission of the sensor's cover glass. The overall QE of the sensor is the product of these three components. For a typical CCD sensor, the QE is highest in the visible range, with a peak value around 500-600 nm. At longer wavelengths, the QE drops rapidly due to absorption in the silicon substrate of the sensor. The QE is typically less than 10% beyond 900 nm. Fig. 3 shows the QE values captured at various photon wavelengths.

Studies have shown [17] that the QE of a CCD image sensor at a wavelength of 550 nm is 70%, whereas that of a CMOS sensor is only 37% at the same wavelength.

## III. IMAGE SENSOR REQUIREMENTS FOR EMBEDDED IOT APPLICATIONS

In the introduction section, we presented a brief description of a few issues and challenges associated with using either CCD or CMOS image sensors for wearable IoT applications. In this section, we will highlight the sensor requirements for embedded IoT applications in terms of their technical characteristics.

One of the main image sensor applications of IoT devices is wearable cameras for surveillance used by law-and-order control officers or night watchman drones. Many times, the requirement might be to grab a video in very low visibility or in absolute dark conditions and transmit it to a remote place. To capture the picture frame that can be decently reproduced, the sensor should have the following specifications to make a minimum viable product (MVP).

- High fill factor – so that the image can be captured in very low visibility or in the absence of light where the human eye cannot see anything clearly.
- Large pixel size, so that image quality is high even under low ambient lighting conditions.
- High dynamic range – so that a decent quality of image can be captured in low-light conditions.
- Global Shutter – so that fast-moving objects can be captured without distortion.
- Low power consumption – this results in less heat dissipation, leading to a longer battery life.
- Low thermal noise, so that finer details in the captured image (such as license plate number) are undistorted.
- Less Smear and Blooming Effect – images captured in high-intensity light are not distorted.
- High Quantum Efficiency – so that lossless (high quality) images are captured.
- Uses as much less on-board memory as possible

Most of the above conditions are satisfied with CCD sensor, which has the disadvantage of high-power consumption. If we choose a CMOS sensor for the sake of low power consumption, we cannot capture any image in poor visibility or in the absence of light (dark ambient conditions) as the CMOS transistor requires a minimum threshold voltage for conduction. To overcome the challenges of high-power, and high memory requirements while using CCD, we present a novel technique of compressed sensing, where the image is compressed while capturing. The results show that the average power consumption is relatively low for the CCD sensor based wearable IoT device with onboard compressed sensing.

#### IV. COMPRESSED SENSING

Compressed sensing (CS) was first demonstrated by Donoho in 2006 [18] and Cande’s et al. [19], where a technique of simultaneous image sensing and compression was used, which is also known as compressive sensing. The main idea involved in CS is to recover a sparse signal by means of nonadaptive linear measurements with convex optimization. This method involves recovering a multidimensional sparse vector with a dimension-reduction step.

A sparse matrix or sparse array is a matrix with a majority of zero elements. Sparsity refers to the nonzero elements of the matrix. The sparse matrix is said to be dense if most of the elements are non-zero. The sparsity of the matrix was obtained by dividing the number of zero-valued elements by the total number of elements. Compressive sensing is suitable for reducing the bandwidth and power of imaging applications. In the CS technique [20], [21], [22], while acquiring the image itself, only the non-redundant required image is sensed or captured, instead of capturing the entire frame, including redundant information, and then removing the redundancy. This technique is presented in [23] with relevant algorithm.

A schematic of the CS implementation is shown in Fig. 4 and the compression process is shown in Fig. 5. The

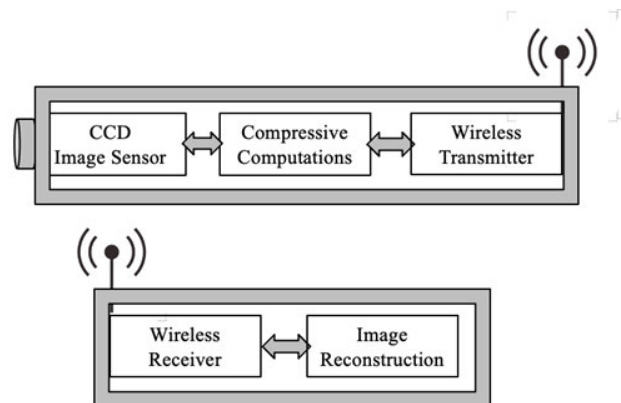


FIGURE 4. Implementation block schematic for compressed sensing.

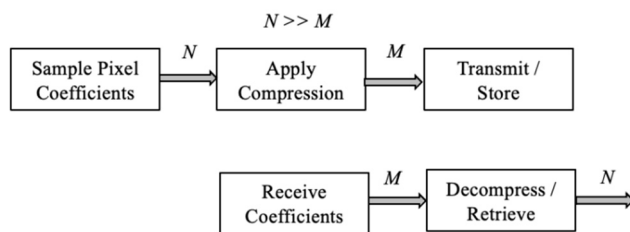


FIGURE 5. Compressed sensing process flow.

streaming video image captured by the CCD image sensor was processed frame-by-frame and transmitted over a wireless sensor network. The received frames are reconstructed at the receiver end.

Generally, the acquired images are sparse in transform domains such as DCT, DFT, or DWT. To understand compressed sensing, consider a transform domain basis  $\Psi$  represented as an  $n \times n$  matrix such that the image  $x$  represented as a column vector,  $n \times 1$  is sparse in  $\Psi$ . This can be expressed as shown in (1) [21], [22]:

$$x = \Psi\theta; \|\theta\|_0 \leq k \tag{1}$$

where  $\theta$  is a  $n \times 1$   $k$ -sparse vector; meaning,  $\theta$  has at most  $k$  non-zero elements, where  $k \ll n$ .

In practice, the transformed image of  $x$ , denoted by  $y$ , is a column vector  $m \times 1$ , where  $m < n$  is measured. In a linear measurement model represented by a matrix  $\Phi$ ,  $y$  is expressed as in (2).

$$y = \Phi x = \Phi\Psi\theta = \Phi'\theta \tag{2}$$

In compressed sensing computations,  $\Phi$  is a random matrix of order  $m \times n$ , known as a mixing matrix. Using this random matrix, the transformed image  $y$  can be represented using a basis, whose vectors are random linear combinations of the original basis  $\Psi$ . In addition,  $\Phi'$  is expressed as  $\Phi' = \Phi\Psi$ .

It has been shown [20] that for efficient compression to occur,  $\Phi$  must be incoherent with the basis  $\Psi$ . This condition is satisfied when  $\Phi$  is chosen as a random matrix. Interestingly,

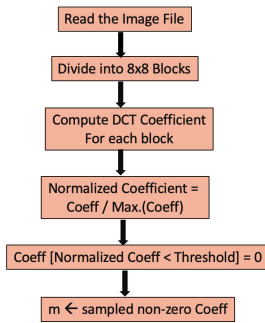


FIGURE 6. Compressed sensing algorithm flow chart.

the sparsity becomes 1 when  $k$  equals  $n$ . This resulted in zero compression. If we chose  $k \ll n$ , this would give reasonably high compression, but owing to limited non-zero samples, the reconstructed image will be very lossy. While computing the CS using equation 2, in the random matrix  $\Phi$  of the order  $m \times n$ , the compression ratio is  $\frac{m}{n}$ , where  $m$  represents no. of measurements, and  $n$  represents no. of pixels. If  $m < n$ , the solution is said to be ill-posed, as there are more unknowns than equations.

This problem can be overcome by using the sparsity  $k$  of  $\theta$ . In this scenario, the measurement vector  $y$  is a linear combination of  $k$  columns of the matrix,  $\Phi\Psi$  with  $\theta \neq 0$ . If we know beforehand the non-zero  $k$  entries of  $\theta$ , then we can form an  $m \times k$  system of linear equations to solve for the non-zero values. In this case, the number of equations  $m$  either equals or exceeds the number of unknowns  $k$  in many cases. The measurement matrix  $\Phi$  must satisfy the Restricted Isometry Property (RIP) which is a necessary and sufficient condition to ensure that the  $m \times k$  system is resolvable [21]. RIP is expressed as shown in (3).

$$(1 - \delta) \leq \frac{\|\Phi\Psi^{-1}v\|_2}{\|v\|_2} \leq (1 + \delta) \quad (3)$$

where  $v$  is any vector with the same nonzero coefficients as that of  $\theta$  and  $0 < \delta < 1$ .

Various optimization solutions can be incorporated to determine  $\theta$  from known values of  $y$  and  $\Psi$ . We chose the following optimization method [21], [25] as shown in (4).

$$\hat{\theta} = \min_{\theta} \|\Phi\Psi - y\|_2^2 + \alpha\|\theta\|_1 \quad (4)$$

We obtain the optimal solution of equation (4) using  $l_1$  minimization as shown in Equation (5), where  $\varepsilon$  is the maximum permissible error in the  $l_1$  minimization step.

$$\min \|\theta\|_1 \text{ subject to } \|\Phi\Psi\theta - y\|_2^2 \leq \varepsilon \quad (5)$$

The original image data,  $x$ , can be reconstructed from the obtained solution,  $\hat{\theta}$ , as shown in (6).

$$\hat{x} = \Psi\hat{\theta} \quad (6)$$

We first acquired the image and converted it to a grey scale before dividing it into  $8 \times 8$  blocks. Subsequently, the DCT

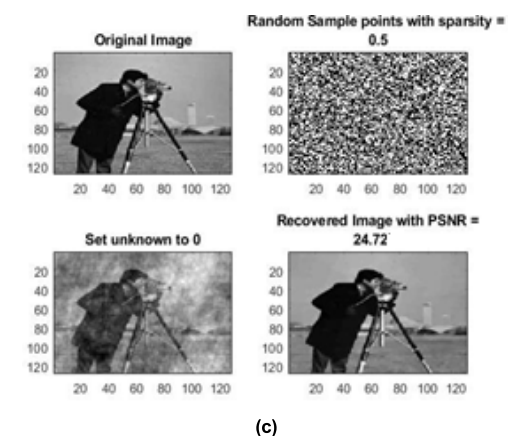
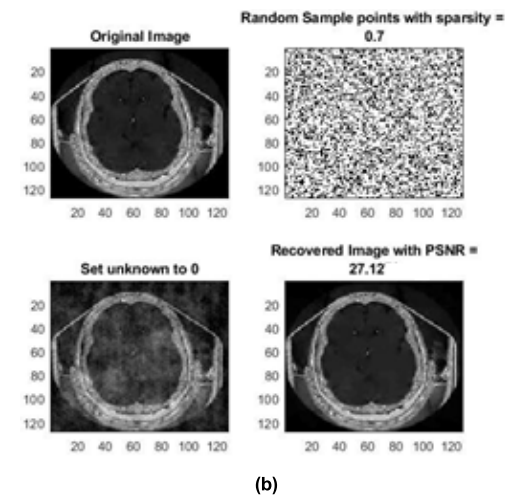
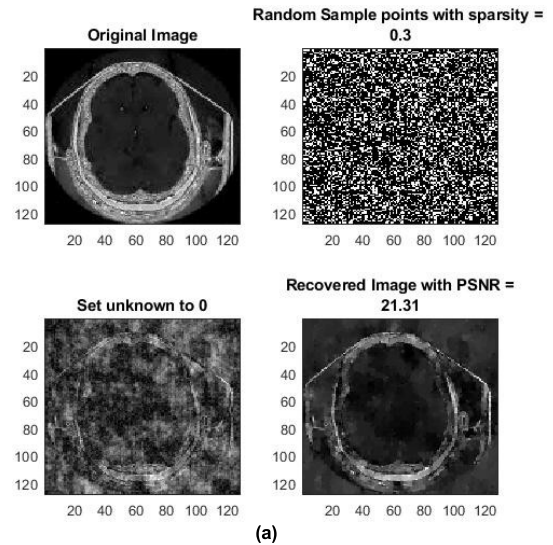


FIGURE 7. MATLAB simulation results with different sparsity values: Figure (a) & (b) are MRI images sampled with sparsity values of 0.3 and 0.7 respectively. Fig. (c) is camera man image sampled with a sparsity 0.5.

coefficients were computed for each block. The coefficients were then normalized using the largest coefficient magnitude. We then chose a threshold value,  $\gamma$ , where any coefficient

value  $< \gamma$  was set to zero. The no. of coefficients  $> \gamma$  are sampled to have ‘ $m$ ’ samples. The value of ‘ $m$ ’ is referred to as ‘sparsity’ or a sparse value. This process is outlined in the flowchart shown in Fig. 6.

MATLAB simulation results [24] for standard images with different sparsity values are shown in Fig. 7.

## V. IMAGE RECONSTRUCTION

In compressed sensing (CS), the goal is to reconstruct an image from a small number of linear measurements, exploiting its sparsity or compressibility in a certain transform domain. There are several familiar reconstruction techniques commonly used to recover sparse signals at various sparse levels. Most used techniques [25] are Basis Pursuit, Matching Pursuit, Orthogonal Matching Pursuit (OMP) and Compressive Sampling Matching Pursuit (CoSaMP) algorithms. CoSaMP is an iterative algorithm that combines ideas from OMP and Iterative Hard Thresholding (IHT). CoSaMP is known for its ability to handle higher levels of sparsity and noisy measurements. We have used CoSaMP algorithm [26] to reconstruct the sparse signal.

Let’s consider a compressive sensing problem where we aim to reconstruct a sparse signal  $x$  from measurements  $y$ . The sensing process can be represented by a measurement matrix  $A$ . The CoSaMP algorithm involves the following five steps:

Initialization:

- Set the solution vector  $x$  as an all-zero vector of appropriate size.
- Compute the initial residual  $r$  by subtracting the measurements  $y$  from the product of the sensing matrix  $A$  and the current solution estimate  $x$ .

Support Set Selection:

- Compute the correlation vector between the residual  $r$  and the columns of  $A$  :  $z = A^T r$
- Select the indices of the  $K$  largest magnitude entries in  $z$ :  $\Omega = \text{topk\_indices}(|z|, K)$ .

Least-Squares Estimation:

- Form a submatrix  $A_{\Omega}$  by selecting the columns of  $A$  indexed by the elements in  $\Omega$ .
- Solve the least-squares problem  $\min \| (A_{\Omega} * x_{\Omega} - y) \|^2$ , where  $x_{\Omega}$  represents the sub-vector of  $x$  corresponding to the indices in  $\Omega$ .
- Obtain the least-squares solution  $x_{\Omega}^*$  within the support set  $\Omega$ .

Where, ‘ $x$ ’ represents the original signal vector, ‘ $\Omega$ ’ refers to the support set, which is a set of indices indicating the selected features or columns of the measurement matrix and ‘ $x_{\Omega}^*$ ’ denotes the estimated or optimal solution for the signal values within the support set ‘ $\Omega$ ’.

Thresholding and Solution Update:

- Update the solution estimate  $x$  by setting all elements outside  $\Omega$  to zero and assigning  $x_{\Omega}^*$  to the corresponding elements within  $\Omega$ .

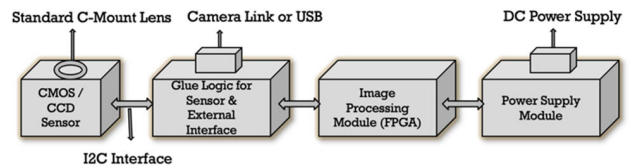


FIGURE 8. System-on-module (SoM) block diagram for image acquisition and processing.



FIGURE 9. System-on-module (SoM) hardware implementation for image acquisition and processing.

- Compute the updated residual  $r$  by subtracting the measurements  $y$  from the product of the sensing matrix  $A$  and the updated solution estimate  $x$ .

Convergence Check:

- If the residual norm  $\|r\|$  is below a predefined threshold  $\epsilon$ , terminate the iterations and return the final solution estimate  $x$ .
- Otherwise, continue to the next iteration.

The algorithm iteratively refines the solution estimate by updating the support set  $\Omega$  and performing a least-squares estimation within the support set. The thresholding step enforces sparsity by setting coefficients outside  $\Omega$  to zero, and the solution update step incorporates the least-squares estimate within  $\Omega$  to improve the solution estimate. The reconstructed images are shown in Table 2 through 10.

## VI. HARDWARE IMPLEMENTATION

In this section, we briefly present an IoT System-On-Module (SoM) built with CCD and CMOS image sensors. We adopted a modular hardware design approach such that the same hardware modules could be used for either of the sensors. A block diagram of the image acquisition and processing of SoM is shown in Fig. 8. The modular design facilitates easier debugging of hardware.

As shown in Fig.9, the hardware implementation involves modular design to make embedded system as compact as possible. It consists of sensor module, compression sensing module (that includes FPGA), an interface module and power supply module. The image sensor module design is performed in such a way that it can hold either a CCD or a CMOS imager with the associated sensor electronics. The overall weight of the above hardware module is approximately 100 grams (without any type of casing) which makes it easier to use in embedded IoT applications. Detailed description of the hardware implementation is beyond the scope of this paper. The completely assembled system of the module is shown in Fig.9. The laboratory test setup is shown in Fig. 10. The tests were repeated for both the CMOS and CCD

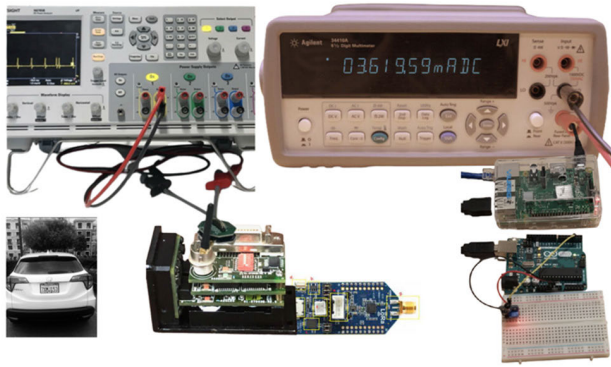


FIGURE 10. Laboratory test & measurement setup acquiring images and measuring average power for various test conditions.

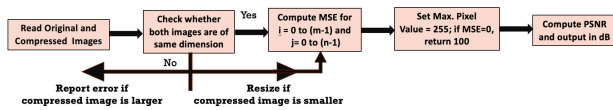


FIGURE 11. PSNR computation algorithm.

TABLE 2. Test Condition 1: Ambient light intensity < 100 Lux; Sparsity = 0.2.

Sensor	Original Image	Reproduced Image	PSNR (dB)	Memory Saving	Power Saving
CMOS			0	NA	NA
CCD			22.75	21%	14%

image sensors under the same test conditions for comparison purpose.

VII. TEST RESULTS

We have summarized the power measurements and memory computation based on the dynamic FIFO register implemented in the FPGA for different test conditions using monochrome CMOS and CCD imagers. Both sensors were set up with the same exposure time, frame rate, and resolution for an apple-to-apple comparison. The Peak Signal-to-Noise Ratio (PSNR) is a measure of the degree to which the original image is corrupted in terms of distortion. The PSNR can be mathematically represented as shown in (7).

$$PSNR = 20 \log_{10} \left( \frac{Max_f}{\sqrt{MSE}} \right) \quad (7)$$

TABLE 3. Test Condition 2: Ambient light intensity < 100 Lux; Sparsity = 0.5.

Sensor	Original Image	Reproduced Image	PSNR (dB)	Memory Saving	Power Saving
CMOS			0	NA	NA
CCD			25.76	15%	13%

TABLE 4. Test Condition 3: Ambient light intensity < 100 Lux; Sparsity = 0.7.

Sensor	Original Image	Reproduced Image	PSNR (dB)	Memory Saving	Power Saving
CMOS			0	NA	NA
CCD			30.71	12%	11%





TABLE 5. Test Condition 4: Ambient light intensity = 1000 Lux; Sparsity = 0.2.

Sensor	Original Image	Reproduced Image	PSNR (dB)	Memory Saving	Power Saving
CMOS			24.93	14%	13%
CCD			25.76	15%	13%





where  $f$  is an  $m \times n$  matrix form of the original image, and MSE is the mean square error, which is expressed as shown in (8).

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |f(i,j) - g(i,j)|^2 \quad (8)$$





**TABLE 6. Test Condition 5: Ambient light intensity = 1000 Lux; Sparsity = 0.5.**

Sensor	Original Image	Reproduced Image	PSNR (dB)	Memory Saving	Power Saving
CMOS			20.25	20%	15%
CCD			25.76	15%	13%

**TABLE 7. Test Condition 6: Ambient light intensity = 1000 Lux; Sparsity = 0.7.**

Sensor	Original Image	Reproduced Image	PSNR (dB)	Memory Saving	Power Saving
CMOS			28.86	12%	10%
CCD			32.71	12%	11%

**TABLE 8. Test Condition 7: Ambient light intensity = 100000 Lux; Sparsity = 0.2.**

Sensor	Original Image	Reproduced Image	PSNR (dB)	Memory Saving	Power Saving
CMOS			20.27	21%	14%
CCD			20.12	21%	14%





where  $g$  is the matrix form of the decompressed image,  $m$  represents no. of pixels in a row,  $i$  is the row index,  $n$  represents no. of pixels in a column,  $j$  is the column index.

The PSNR is computed for each image using the algorithm described in Fig. 11. The results for each test case are listed in Table 2 through 11.





In Table 2 through 11, original image and reconstructed image have the following meanings:

Original Image: At the output of the sensor.

**TABLE 9. Test Condition 7: Ambient light intensity = 100000 Lux; Sparsity = 0.5.**

Sensor	Original Image	Reproduced Image	PSNR (dB)	Memory Saving	Power Saving
CMOS			25.32	15%	14%
CCD			25.42	15%	13%

**TABLE 10. Test Condition 7: Ambient light intensity = 100000 Lux; Sparsity = 0.7.**

Sensor	Original Image	Reproduced Image	PSNR (dB)	Memory Saving	Power Saving
CMOS			32.86	12%	11%
CCD			34.71	13%	12%

Note: All the images in Table 2 through 10 are true images captured with both CMOS and CCD cameras and condensed for printing purposes. The original resolution of each image is  $640 \times 480$  pixels.

**TABLE 11. PSNR Values computed for each test case for different lighting conditions.**

Light Intensity (Lux)	PSNR (CMOS) Sparsity=0.2	PSNR (CCD) Sparsity=0.2	PSNR (CMOS) Sparsity=0.5	PSNR (CCD) Sparsity=0.5	PSNR (CMOS) Sparsity=0.7	PSNR (CCD) Sparsity=0.7
100	0	22.75	0	25.76	0	30.71
200	10.23	23.65	11.75	26.89	12.21	31.32
400	12.45	24.45	12.76	27.65	12.98	32.41
1000	25.23	25.51	27.23	27.93	32.45	32.71
2000	26.56	26.71	28.76	28.88	33.29	33.86
3000	27.44	27.86	29.23	29.86	34.23	34.98
5000	29.64	29.87	31.56	32.78	35.53	35.89
10000	30.55	30.86	32.82	33.18	36.39	36.98
20000	31.66	31.78	33.59	34.12	37.81	37.99
50000	43.45	44.57	46.86	48.55	50.32	53.11
100000	55.43	43.48	58.48	47.69	60.47	52.32

Reconstructed Image: At the IoT receiver in the wireless sensor network.

Constant parameters were:

Sensor resolution: Video Graphic Array (VGA)



**TABLE 12. PSNR power savings for CCD and CMOS image sensors at different sparsity values.**

Light (Lux)	CMOS Power Saving S=0.2	CCD Power Saving S=0.2	CMOS Power Saving S=0.5	CCD Power Saving S=0.5	CMOS Power Saving S=0.7	CCD Power Saving S=0.7
100	0	14%	0	13%	0	11%
200	0	14%	0	13%	0	11%
400	0	14%	0	13%	0	11%
1000	13%	13%	15%	15%	10%	11%
2000	13.2%	13.2%	15.2%	15%	12%	14%
3000	13.4%	13.4%	15.4%	15%	13%	15%
5000	13.6%	13.6%	15.5%	15%	14%	16%
10000	13.8%	13.8%	15.6%	15%	15%	16%
20000	14%	14%	15.7%	14%	16%	16%
50000	14%	14%	15.8%	15%	17%	16%
100000	14%	14%	14%	13%	11%	12%

(640 × 480) pixels;

Exposure time or shutter speed: 10 ms;

Frame rate: 30 fps.

Variable parameters were:

Lighting conditions,

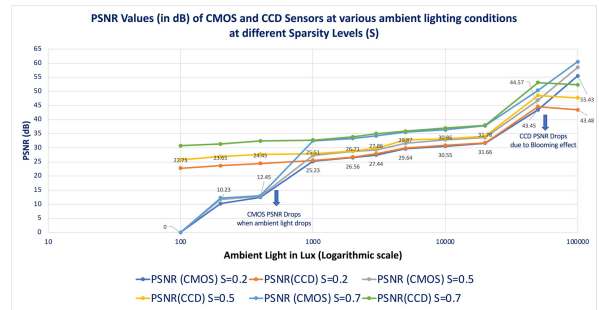
Sparsity (0.2 0.5, 0.7, programmed through FPGA).

Memory and power savings are calculated comparing with uncompressed data. Before we present the test results and analysis, to make it clear, when sparsity is 1, we do not discard any samples from the captured image; hence, compressive sensing is not applied. As a result, we do not obtain any savings in power and memory with a sparsity equal to 1. In other words, PSNR is highest when sparsity is 1 (no compression), and PSNR is lowest when sparsity is close to 0.2 (highest compression). The results for various test conditions are summarized in Table 2 through 10.

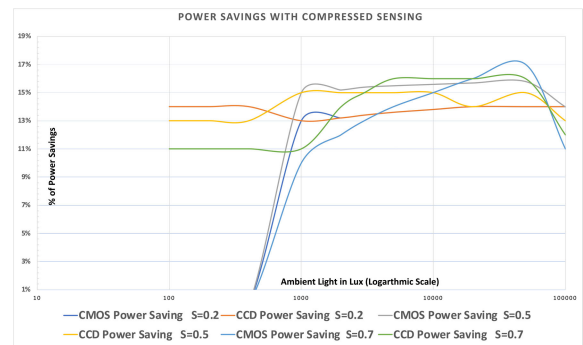
**VIII. CONCLUSION**

The PSNR values computed for each test case are summarized in the Table 11. The PSNR values are plotted for various lighting conditions and sparsity levels for both CCD and CMOS sensors is shown in Fig.12. Percentage of power savings for both CMOS and CCD sensor based embedded system at different sparsity values and light intensity are listed in Table 12. Power savings as shown in Table 12 for various lighting conditions and sparsity values are plotted in Fig. 13 for both CCD and CMOS sensor based embedded system.

Under low ambient light conditions, the CMOS sensor cannot be used, whereas the CCD sensor outperforms it, and the reproduced image is clearly readable. This is exhibited in Table 2, 3 and 4. The resulting plot shown in Fig.12 also



**FIGURE 12. PSNR values at various lighting conditions for CMOS and CCD sensors.**



**FIGURE 13. Power savings for CMOS and CCD sensors at different sparsity values.**

illustrates this phenomenon where PSNR drops for CMOS image sensor when the light intensity falls < 500 Lux. For the same light intensity levels of < 500 Lux, the PSNR values obtained for CCD sensor are considerably high. This is the biggest advantage of using a CCD sensor for low-or no-light conditions, as the CCD imager develops a charge even in the absence of light, which is necessary for wearable IoT devices. Power and memory savings are functions of the sparsity for a given sensor resolution. The savings shown in our test results were for the VGA resolution. For higher resolution image sensors, the savings are significantly high. With the adoption of compressed sensing onboard the CCD image sensor, the memory and power requirements are reduced. Hence, the CCD image sensor with compressed sensing can be used for wearable IoT applications, which makes it sustainable even in low light conditions. The image recovery system can be on a cloud or in a data center with no resource limitations. Using CCD sensors for drone applications [27] has definite advantage.

From Table 12 and Fig. 13, we can see that the power saving is function of sparsity values and does not change with ambient lighting conditions. With the combination of CCD sensor using compressed sensing (with sparsity of 0.5), we can achieve reasonably decent image capturing system that can be used in embedded IoT applications. The memory saving too is a function of sparsity level and image sensor resolution. More the resolution of the image sensor, more the memory saving would be at sparsity levels of 0.5. The percentage of

memory savings are shown in Table 2 to Table 10 for all the test conditions.

The entire system can be made even more sustainable by using both CMOS and CCD sensors in the modular design shown in Fig.8 and Fig.9 for night watchman drone applications. Both the sensors can be connected to a single backend module through a multiplexer. At any given point in time, only one, CMOS or CCD sensor module be connected to the backend system. The selected multiplexer signal is controlled by a light sensor. Under adequate ambient lighting conditions, a CMOS sensor can be chosen, and during the dark, a CCD sensor can be chosen. The energy models for sensor nodes in wireless sensor network with compressed sensing is proposed in [28].

The method proposed in this paper can achieve high-quality images with significant reductions in the number of measurements required, making it a promising technique for CCD sensor-based cameras. The simulation and experiment results demonstrate the effectiveness of the proposed method in reconstructing images with a high SNR and without visible artifacts. Future work can investigate the use of different sparsifying transforms, measurement matrices, and reconstruction algorithms to further improve the performance of the proposed method. Additionally, the proposed method can be extended to color images by compressing and reconstructing each color channel separately. Overall, the proposed method has the potential to improve the performance of CCD sensor-based cameras in various imaging applications, where CMOS sensors cannot capture the images.

## REFERENCES

- [1] E. R. Fossum, "CMOS image sensors: Electronic camera-on-a-chip," *IEEE Trans. Electron Devices*, vol. 44, no. 10, pp. 1689–1698, Oct. 1997, doi: [10.1109/16.628824](https://doi.org/10.1109/16.628824).
- [2] M. G. Farrier and R. H. Dyck, "A large area TDI image sensor for low light level imaging," *IEEE J. Solid-State Circuits*, vol. SSC-15, no. 4, pp. 753–758, Aug. 1980, doi: [10.1109/JSSC.1980.1051465](https://doi.org/10.1109/JSSC.1980.1051465).
- [3] L. Zhiyong, Y. Weihua, and D. Xiance, "The analog front end of ultra-high resolution CCD design based on AD9920A," in *Proc. 8th Int. Conf. Intell. Comput. Technol. Autom. (ICICTA)*, Nanchang, China, Jun. 2015, pp. 921–924, doi: [10.1109/ICICTA.2015.234](https://doi.org/10.1109/ICICTA.2015.234).
- [4] Y.-W. Dai, C.-C. Chien, D.-C. Juang, T.-Y. Bin, C.-Y. Hsu, C.-D. Wu, Y.-L. Zhaog, M.-F. Ho, M.-H. Cheng, G.-P. Yang, Y.-F. Luo, and C.-H. Horng, "Color CCD image sensor," in *Proc. Int. Symp. VLSI Technol., Syst., Appl.*, Taiwan, 1991, pp. 126–130, doi: [10.1109/VTSA.1991.246696](https://doi.org/10.1109/VTSA.1991.246696).
- [5] J. Guo and S. Sonkusale, "A high dynamic range CMOS image sensor for scientific imaging applications," *IEEE Sensors J.*, vol. 9, no. 10, pp. 1209–1218, Oct. 2009, doi: [10.1109/JSEN.2009.2029814](https://doi.org/10.1109/JSEN.2009.2029814).
- [6] S. K. Mendis, S. E. Kemeny, R. C. Gee, B. Pain, C. O. Staller, Q. Kim, and E. R. Fossum, "CMOS active pixel image sensors for highly integrated imaging systems," *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 187–197, Feb. 1997, doi: [10.1109/4.551910](https://doi.org/10.1109/4.551910).
- [7] J. Ohta, "Fundamentals of CMOS image sensors," in *Smart CMOS Image Sensors and Applications*, vol. 4, 2nd ed. New York, NY, USA: CRC Press, 2020, ch. 2, pp. 11–57 and 107–142.
- [8] Edmundoptics. (2018). *Understanding Camera Sensors for Machine Vision Applications*. Accessed: Aug. 20, 2021. [Online]. Available: <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-camera-sensors-for-machine-vision-applications/>
- [9] Siliconimaging. (2015). *CMOS Fundamentals*. Accessed: Aug. 20, 2021. [Online]. Available: [http://www.siliconimaging.com/cmos\\_fundamentals.htm](http://www.siliconimaging.com/cmos_fundamentals.htm)
- [10] I. Gyongy, N. Calder, A. Davies, N. A. W. Dutton, P. Dalgarno, R. Duncan, C. Rickman, and R. K. Henderson, "256 × 256, 100 kfps, 61% fill-factor time-resolved SPAD image sensor for microscopy applications," in *IEDM Tech. Dig.*, Dec. 2016, pp. 8.2.1–8.2.4, doi: [10.1109/IEDM.2016.7838373](https://doi.org/10.1109/IEDM.2016.7838373).
- [11] D. Durini, "Charge-coupled device (CCD) image sensors," in *High Performance Silicon Imaging: Fundamentals and Applications of CMOS and CCD Image Sensors*, 2nd ed. Cambridge, MA, USA: Woodhead Publishing, 2019, ch. 2, pp. 75–92.
- [12] T. Le, N.-T. Le, and Y. M. Jang, "Performance of rolling shutter and global shutter camera in optical camera communications," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju Island, South Korea, Oct. 2015, pp. 124–128, doi: [10.1109/ICTC.2015.7354509](https://doi.org/10.1109/ICTC.2015.7354509).
- [13] O. Saurer, K. Köser, J. Bouguet, and M. Pollefeys, "Rolling shutter stereo," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 465–472, doi: [10.1109/ICCV.2013.64](https://doi.org/10.1109/ICCV.2013.64).
- [14] B. Fan, K. Wang, Y. Dai, and M. He, "RS-DPSNet: Deep plane sweep network for rolling shutter stereo images," *IEEE Signal Process. Lett.*, vol. 28, pp. 1550–1554, 2021, doi: [10.1109/LSP.2021.3099350](https://doi.org/10.1109/LSP.2021.3099350).
- [15] D. Levski, M. Wány, and B. Choubey, "Compensation of signal-dependent readout noise in photon transfer curve characterisation of CMOS image sensors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 1, pp. 102–105, Jan. 2021, doi: [10.1109/TCSII.2020.3010366](https://doi.org/10.1109/TCSII.2020.3010366).
- [16] Y. Han, E. Choi, and M. Kang, "Smear removal algorithm using the optical black region for CCD imaging sensors," *IEEE Trans. Consum. Electron.*, vol. 55, no. 4, pp. 2287–2293, Nov. 2009, doi: [10.1109/TCE.2009.5373800](https://doi.org/10.1109/TCE.2009.5373800).
- [17] B. Kozacek, J. Grauzel, and M. Frivaldsky, "The main capabilities and solutions for different types of the image sensors," in *Proc. 12th Int. Conf. ELEKTRO*, Mikulov, Czech Republic, May 2018, pp. 1–5, doi: [10.1109/ELEKTRO.2018.8398278](https://doi.org/10.1109/ELEKTRO.2018.8398278).
- [18] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006, doi: [10.1109/TIT.2006.871582](https://doi.org/10.1109/TIT.2006.871582).
- [19] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006, doi: [10.1109/TIT.2005.862083](https://doi.org/10.1109/TIT.2005.862083).
- [20] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008, doi: [10.1109/MSP.2007.914731](https://doi.org/10.1109/MSP.2007.914731).
- [21] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–124, Jul. 2007, doi: [10.1109/MSP.2007.4286571](https://doi.org/10.1109/MSP.2007.4286571).
- [22] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 14–20, Mar. 2008, doi: [10.1109/MSP.2007.914729](https://doi.org/10.1109/MSP.2007.914729).
- [23] Y. Gui, H. Lu, X. Jiang, F. Wu, and C. W. Chen, "Compressed pseudo-analog transmission system for remote sensing images over bandwidth-constrained wireless channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 3181–3195, Sep. 2020, doi: [10.1109/TCSVT.2019.2935127](https://doi.org/10.1109/TCSVT.2019.2935127).
- [24] G. Ramachandra and M. S. Bhat, "Compressed sensing for energy and bandwidth starved IoT applications," in *Proc. IEEE Distrib. Comput., VLSI, Electr. Circuits Robot. (DISCOVER)*, Mangalore, India, Aug. 2018, pp. 131–134, doi: [10.1109/DISCOVER.2018.8674107](https://doi.org/10.1109/DISCOVER.2018.8674107).
- [25] S. Çelik, M. Basaran, S. Erkiçük, and H. A. Çırpan, "Comparison of compressed sensing based algorithms for sparse signal reconstruction," in *Proc. 24th Signal Process. Commun. Appl. Conf. (SIU)*, May 2016, pp. 1441–1444, doi: [10.1109/SIU.2016.7496021](https://doi.org/10.1109/SIU.2016.7496021).
- [26] X. Zhang, W. Xu, Y. Cui, L. Lu, and J. Lin, "On recovery of block sparse signals via block compressive sampling matching pursuit," *IEEE Access*, vol. 7, pp. 175554–175563, 2019, doi: [10.1109/ACCESS.2019.2955759](https://doi.org/10.1109/ACCESS.2019.2955759).
- [27] T. Yuske and Z. Mbaitiga, "Development of drone detecting free parking space for car parking guidance," in *Proc. Int. Conf. Intell. Informat. Biomed. Sci. (ICIBMS)*, Shanghai, China, Nov. 2019, pp. 385–387, doi: [10.1109/ICIBMS46890.2019.8991452](https://doi.org/10.1109/ICIBMS46890.2019.8991452).
- [28] P. Wei and F. He, "The compressed sensing of wireless sensor networks based on Internet of Things," *IEEE Sensors J.*, vol. 21, no. 22, pp. 25267–25273, Nov. 2021, doi: [10.1109/JSEN.2021.3071151](https://doi.org/10.1109/JSEN.2021.3071151).



**RAMACHANDRA GAMBHEER** (Member, IEEE) received the Bachelor of Engineering (B.E.) degree from Gulbarga University, in 1990, and the Master of Technology (M.Tech.) degree from NITK Surathkal, in 1998. He started his career as a Faculty Member with the Department of Electronics and Communication Engineering, NITK Surathkal, from 1991 to 2000. He joined industry with Semiconductor Technologies & Instruments (a TI spun off company), in 2000, where he

designed several CCD and CMOS cameras for machine vision applications. He designed a FPGA-based 17mm x 17mm CMOS (using new high speed CMOS sensor designed by then Photobit, Pasadena, USA) embedded camera with a built-in image processing application for chip capacitor inspection which was the smallest camera fitted into a machine vision systems. He worked with various image sensors, including Micron, Phtobit, Kodak, Aptina, and Fill Factory. In 2005, he joined Intellitech, India, as the Director, where he designed several JTAG based functional testers for various contractor manufacturers across the world. He held a startup company for three years, from 2010 to 2013, where he designed several machine vision inspection systems and functional testers for various contractor manufacturers. He joined Cisco Systems, in 2013. He is currently a Senior Technical Operations lead while pursuing part-time research with NITK Surathkal. He has coauthored a textbook *Design of Secure IoT Systems—A Practical Approach Across Industries* (McGraw Hill, New York City). His research interests include the IoT, connected cars, the Internet of Medical Things, and the wearable IoT devices. He was a joint recipient of VLSI'99 International Award for the FPGA-based design done for implementation of MIL-STD-1553 protocols for Indian spacecraft.



**M. S. BHAT** (Senior Member, IEEE) received the M.Sc. degree in physics from Mangalore University, in 1987, the M.Tech. degree from National Institute of Technology Karnataka (NITK) Surthkal, in 1989, and the Ph.D. degree from the Indian Institute of Science, Bengaluru, in 2007. He joined the Department of Electronics and Communication Engineering, NITK Surathkal, as a Faculty Member, in 1989. Currently, he is a Professor with the Department

of Electronics and Communication Engineering. He has been a Visiting Researcher with UMIST, Manchester, under the ODA Program, since 1997, and a Visiting Researcher with TU Eindhoven under the Erasmus Mundus Program, since 2009. At NITK Surathkal, he has executed several sponsored projects—world bank sponsored project IMPACT, from 2000 to 2002, SMDP project in VLSI from Meity, Government of India, from 2007 to 2015, RF-MEMS projects from DIT and NPMASS, Government of India, from 2010 to 2014, energy harvesting from passenger seats funded by ANRC Boeing, from 2014 to 2016, collaborative research in sensor systems for public utilities, from 2016 to 2018, funded by the Royal Academy of Engineering, U.K., and several other collaborative projects. His research interests include low power analog and mixed signal design, submicron devices, and image and signal processing. He is a Life Member of ISSS and ISTE.

• • •