Abstract—As far as wireless sensor networks (WSN) are concerned, memory and energy are the two main constraints. In this paper, we investigate a framework of compressive sensing that makes use of the discrete cosine transform used in conventional image compression algorithms such as JPEG and Noiselet transform which is solved using second order cone programming (SOCP), and analyze its impact on the constraints of the WSN. The energy consumed for transmitting the image using the developed compressed sensing (CS) framework is calculated. Rate distortion analysis is also done.

Keywords—Compressed Sensing, DCT, Energy, Noiselet, WSN

I. INTRODUCTION

The areas in which wireless sensor networks are used are very remote and sometimes in inaccessible terrains. The nodes in a WSN generally run on one-way batteries and therefore the usage of the energy is of critical importance. Hence the need for power efficiency becomes more important. Another constraint in WSN scenario is that of the available memory. The nodes in such a network need to be cheap and simple, so only a limited amount of memory is available in the nodes. Therefore for maximizing battery lifetime and reducing memory occupied, it is essential, for each sensor, to locally compress its data [1]. A node also has limited computing and communication capabilities. Complex processes cannot be executed on these nodes and data formation, collection and transmission needs to be as effective (in terms of energy consumed) as possible. In a WSN scenario as shown in Fig 1, the nodes gather the information and transmit it to the sink/receiver where all the required processing is done. There is not much processing done at the node. The receiver is assumed to be a device capable of performing complex processes with no power constraint issues. Now-a-days nodes are equipped with camera on board, so they capture images and transmit it towards the sink (destination node). The processing will be done on the node itself or in the nodes through which the data is forwarded to the sink.

Image compression algorithms convert high-resolution images into a relatively small bit streams (while keeping the essential features intact), in effect turning a large digital data set into a substantially smaller one. The concept of compressive sensing is to build the compression during acquisition of the image and not to perform the compression after the complete acquisition of the image. The idea is to acquire lesser number of data and try to recover the complete picture from reduced data which seems to be highly incomplete information. Thus compressive sensing enables this data compression by working on random transform coefficients. This paper proposes a methodology that can be employed in a WSN scenario to overcome its main constraints, energy and memory.

Fig.1 A wireless sensor Network

J. Romberg [2] proposed the recovery of the images from reduced measurements using the CS technique. He used DCT and Noiselet measurements taken from the whole image. The processing is done with the image as a whole which will lead to high energy consumption in case of WSN. In the methodology introduced by Jiantao Wen, local 8*8 DCT is performed on the original image and then the most significant K1 coefficients are selected as per the sorted list over all the blocks of the image [3]. After this step, K2 Noiselet coefficients are obtained and sent to the decoder. The number of coefficients contributed by each 8*8 block differs in this case which also increase the transmission burden as the number of dct elements in each block has to be transmitted to the receiver. Image compression algorithm using DCT, CS and vector quantization is proposed by Dipti Bhatnagar [4] which has a performance limitation due to the exponentially growing codebook size. Hence for better compression ratios, the complexity of the system increases obviously. This paper proposes a methodology with the combination of binDCT and...
Noiselet; the measurements are taken by dividing the image into smaller blocks. The number of DCT and Noiselet coefficients from each block remains fixed once the application is chosen by the sink. Hence there is no need of extra bits to be transmitted in [3]. It also calculates the energy consumption of the image transmission over WSN considering MICA2 motes.

The rest of the paper is organized as follows: section II provides the introduction for CS, the processes done at the transmitting end is discussed in section III, energy computation procedure is given in section IV, reconstruction methodology and the sink node operations are provided in section V, the results are discussed in section VI and the section VII concludes the paper.

II. COMPRESSED SENSING

The image is first represented in terms of the basis vectors \( \{\psi_i\}_{i=1}^N \), corresponding to the transformation:

\[
x = \sum_{i=1}^{N} s_i \psi_i \tag{1}
\]

Forming the coefficient vector \( s \) and the \( N \times N \) basis matrix \( \psi := [\psi_1, \psi_2, \ldots, \psi_N] \) by stacking the vectors \( \{\psi_i\} \) as columns, we can write the samples as \( x = \psi s \).

The compression stage consists of computing certain measurements from vector \( x \) by using a collection of vectors \( \{\phi_i\}_{i=1}^N \). So measurements are calculated as follows:

\[
y = \phi x = \phi \psi s = \Theta s \tag{2}
\]

The measurement is non-adaptive, as in \( \phi \) does not depend on the image. There is a dimensionality reduction while transforming from \( x \) to \( y \) thus loss in information. Since \( M < N \), there are many \( x^* \) such that \( \phi x^* = y \).

The magic of CS [5] is that \( \phi \) can be designed such that \( x \) can be approximately recovered by \( y \). If a signal is \( S \)-sparse under representation basis \( \psi \), then the number of samples it takes to faithfully reconstruct the signal using the basis \( \phi \) is determined by the correlation of \( \phi \) and \( \psi \), known as the restricted isometric property. Lesser the coherence between the two bases, lesser the number of elements required to reconstruct the image. Natural images are sparse under decompositions using DCT. Noiselets are noise-like in nature, and are known to be uncorrelated with respect to many used decompositions using DCT. Noiselets are noise-like in nature, reconstruct the image. Natural images are sparse under two bases, lesser the number of elements required to represent the image using the sensing basis \( \phi \).

The next step is to determine the \( K_2 \) noiselet coefficients to be transmitted. The location of these coefficients is determined by the sink and transmitted to the source rather than being computed at the source node itself. This is to avoid the energy being spent at the source node in determining these locations and then transmitting the location data to the sink. This is also to avoid the overhead of communicating these locations. This is because, the distribution of noiselet coefficients is random in nature and more image-dependent, while in the case of DCT coefficients, the significant coefficients lie in the lower frequencies.

A. Noiselet

The family of noiselets [6] is constructed on the interval \([0,1]\) as in (3) to (5)

\[
f_1(x) = \chi_{(0,1)}(x) \tag{3}
\]

\[
f_{2,2n+1}(x) = (1+i)f_{n}(2x) + (1-i)f_{n}(2x-1) \tag{4}
\]

\[
f_{2,2n}(x) = (1-i)f_{n}(2x) + (1+i)f_{n}(2x-1) \tag{5}
\]

The extension of these noiselet functions to the interval \([0, 2^n - 1]\) generates a sequence of noiselet matrices \(N_1, N_2, N_4, \ldots, N_2^n \) of size \(1 \times 1, 2 \times 2, 4 \times 4, \ldots, 2^n \times 2^n \).

The noiselet matrix \((n \times n)\) can be constructed recursively according to the formulae (6) and (7).

\[
N_n(k, k) = \frac{1}{2} (1-i)(1+i) \otimes N_{n/2}(k-1, k) \tag{6}
\]

when \( k = 0, 2, 4, \ldots, n-2 \) and

\[
N_n(k, *) = \frac{1}{2} (1+i)(1-i) \otimes N_{n/2}(k-1, *) \tag{7}
\]

when \( k = 1, 3, 5, \ldots, n-1 \)

The implementation of this algorithm requires \( n \log_2 n \) additions, where \( n \) is the order of the image [2].
B. Measurements

The \( K_2 \) noiselet coefficients taken randomly are appended to the set of \( K_1 \) DCT coefficients to form a column vector and are quantized by normal bit-shifting of the elements. The quantization level can be varied depending upon the quality of reconstructed image required. The quantization is restricted to uniform bit-shifting of the elements in vector rather than other complex processes keeping in mind the energy constraints at the source node.

\[
y_q = \frac{y}{\Delta_q}
\]

(8)

where \( \Delta_q = 2^n \)

Depending upon the quality required the value of \( n \) can be varied. The resulting quantized vector is Huffman encoded by using the Huffman table that is assumed to be stored in the node. The encoded vector is transmitted to the sink node.

C. Integer DCT

The conventional floating point method of computing the discrete cosine transform is very exhaustive. The floating point method computes the DCT using multiplications and additions. The energy consumed for multiplications is generally high when compared to additions or shifting operations. As we have a limited power supply at the source node, it is important to utilize it effectively. On this regard, we use the concept of integer DCT (binDCT when lifting scheme is utilized) to replace the floating point DCT. Using binDCT, the forward and the inverse transforms can be implemented using only binary shift and addition operations. The energy consumed via this method would be far less than the conventional method.

There are several families of such multiplierless binDCT computations based on Chen’s and Loeffler’s plane rotation based factorization of the DCT matrix [7], [8]. The configuration employed in this work is that of C1, Chen’s factorization. The table below compares the operations in the conventional method and binDCT method.

|TABLE I| COMPARISON OF DCT METHODS USED FOR 8×8 MATRIX|
|:---:|:---:|:---:|
|Type| No. of Multiplications| No. of Additions| No. of shifts|
|Floating Point DCT| 1024| 896| -|
|Integer DCT| -| 672| 368|

The binDCT enjoys the properties of both DCT and the lifting scheme. It inherits the properties of the lifting scheme such as fast implementation, integer-to-integer mapping, and in-place computation, as well as those of DCT such as high coding gain, symmetric basis functions and recursive construction. These features make binDCT an appropriate choice in a WSN scenario.

IV. Energy computation

The energy computed at the source node is broadly of two types: computational and transmission/reception. The computational energy consumed depends upon the microcontroller being used and the transmission/reception energy depends upon the transceiver being used. The computational energy is a sum of the following individual energies consumed as in (9) to (17).

A. BinDCT energy computation

Performing the binDCT for a single row requires 42 additions and 23 shift operations [7]. For an \( 8 \times 8 \) matrix it is required to perform these operations 16 times, for 8 rows and 8 columns. Here \( n \) is the order of the image and \( k \) is the order of the block (in our case \( k = 8 \)).

\[
E_{\text{binDCT}} = \left( \frac{n}{k} \right)^2 \left( 16 \times 42 \times e_{\text{add}} + 23 \times e_{\text{shift}} \right)
\]

(9)

B. Noiselet energy computation

The noiselet energy consumption is computed as given in (10). The implementation uses exactly \( n \times \log_2 n \) additions or subtractions where \( n \) is the order of the image.

\[
E_{\text{noiselet}} = \left( n \times \log_2 n \right) \times e_{\text{add}}
\]

(10)

C. Zigzag ordering energy computation

As we know, the low frequency components contain most of the significant image information, there is a simple rearrangement stage of the \((k^2 - 1)\) coefficients in a zigzag order, from lower to higher frequencies. The coefficients are shifted to their new positions [9]

\[
E_z = \left( \frac{n}{k} \right)^2 \times (k^2 - 1) \times e_{\text{shift}}
\]

(11)

D. Read-write energy computation

This energy includes the energy consumed in reading all the blocks from the image and the energy consumed in writing the required \( K_1 \) DCT coefficients from each block and the \( K_2 \) noiselet coefficients. [10]

\[
E_{\text{read}} = (n \times n) \times e_{\text{read}}
\]

(12)

\[
E_{\text{write}} = \left( K_1 \times \left( \frac{n}{k} \right)^2 + K_2 \right) \times e_{\text{write}}
\]

(13)

E. Quantization energy

The quantization carried out in this work is a simple bit shift of \( n \) bits over the message file of length \( \text{bits} \) over the message file of length \( y \).
\[ E_{\text{quant}} = n \cdot \text{length}(y) \cdot e_{\text{shift}} \]  

(14)

**F. Encoding energy**

The encoding scheme used is Huffman coding. The energy is given by the number of bytes of information multiplied by the energy for entropy encoding per byte. \[ E_{\text{enc}} = \text{no. of bytes} \cdot e_{\text{ent}} \] 

(15)

**G. Transmission/reception energy**

The transmission/reception energy is the sum of \( E_{\text{RX}} \) and \( E_{\text{TX}} \).

Energy spent on reception of the \( K_2 \) noiselet locations.

\[ E_{\text{RX}} = K_2 \cdot e_{\text{RX}} \]  

(16)

Energy spent on transmitting the encoded message

\[ E_{\text{TX}} = \text{no of bytes for transmission} \cdot e_{\text{TX}} \]  

(17)

The adopted input parameters and energy computations refer to the characteristics of MICA2 motes. These devices consist of a low-power ATmega 128L microcontroller, a Chipcon CC1000 transceiver and an Atmel AT45DB041 serial Flash memory with 512K bytes for data storage [10].

**TABLE II**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Operation performed</th>
<th>Energy consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_{\text{add}} )</td>
<td>addition over 1 byte</td>
<td>3.3nJ</td>
</tr>
<tr>
<td>( e_{\text{shift}} )</td>
<td>shift over 1 byte</td>
<td>3.3nJ</td>
</tr>
<tr>
<td>( e_{\text{read}} )</td>
<td>reads 1 byte from the flash memory</td>
<td>0.26µJ</td>
</tr>
<tr>
<td>( e_{\text{write}} )</td>
<td>writes 1 byte in the flash memory</td>
<td>4.3µJ</td>
</tr>
<tr>
<td>( e_{\text{TX}} )</td>
<td>transmits 1 byte</td>
<td>24.96µJ</td>
</tr>
<tr>
<td>( e_{\text{RX}} )</td>
<td>receives 1 byte</td>
<td>18.72µJ</td>
</tr>
<tr>
<td>( e_{\text{ent}} )</td>
<td>encodes 1 byte</td>
<td>160 nJ</td>
</tr>
</tbody>
</table>

**V. RECEIVER**

The sink node receives the message and decodes the information using the Huffman table. The decoding process gives us back the DCT and Noiselet coefficients, it is illustrated in Fig.3.

The image reconstruction problem is solved as an SOCP [11] with the constraint \( Ax^* = y \). Solving the problem in this manner is equivalent to getting an estimate of the original image using \( K_1 \) DCT coefficients and constructing the original image using compressive sensing. The construction of a matrix with \( K_1 \) DCT rows and \( K_2 \) Noiselet rows models the prediction phase (with \( K_1 \) DCT) and compressive sensing phase (with \( K_2 \) phase).

Since the signal is a 2D image, the recovery model is that the gradient is sparse. Let \( x_{i,j} \) represent the pixel in the \( i \)th row

\[ \sum_{i,j} \sqrt{(D_{ni,j}x)^2 + (D_{nj,j}x)^2} = \sum_{\gamma} \|D_{\gamma}x\|_2 \leq \varepsilon \]  

(21)

where \( y \) measurement observed based on the DCT and noiselet information and \( \varepsilon \) is a user-defined parameter.

This can be solved by the log-barrier algorithm, but the initial point is acquired by using the conjugate gradient method. The conjugate gradient method is an iterative method that is applied to sparse systems that are large and cannot be handled by direct methods effectively.

**VI. RESULTS AND DISCUSSION**

As per the methodology proposed Noiselet and the BinDCT coefficients are combined to get the actual measurements that are to be transmitted. The quantized measurements are then encoded using Huffman coding. At the sink node CG solve and barrier iterations are used to recover the original image. The methodology is tested for the various combinations of the measurements (BinDCT + Noiselet) and the performance metrics calculated are PSNR, Bpp, compression ratio and energy consumed.

PSNR is increased with the increase in both the binDCT and the Noiselet coefficients, based on the requirement (WSN scenario) we aimed at getting the reduced bit rate (<0.5bpp) with reasonable quality. Table III lists the performance metrics calculated for the 128×128 cameraman image. The recovered images are shown in Fig. 4.
The total energy consumed ranges from 110mJ to 280 mJ which is considerably less than the energy required for raw image transmission (840mJ). The energy calculation is done by assuming the transmission from the source node to the sink node directly with a fixed transmission radius. The energy can be reduced further by transmitting the data in distributed fashion through clusters.

The maximum compression ratio listed is 60% which can still be increased with the degradation in image quality. The minimum compression ratio listed is 23% which can still be decreased to get best image quality. The proposed methodology achieves better compression with reasonable image quality. It also has a better image quality. Analysis will be made to source which will reduce the energy consumption further. We will also work on different encoding strategies in future.

### VII. CONCLUSION

The proposed methodology consumes only 33% to 13% of the energy required for the raw image transmission across the WSN. With this strategy it has 20% to 60% compression ratio. It is evident that with the reduction in the energy consumption the life time of the WSN will be accelerated to a greater extent. It also has a better image quality. Analysis will be made to avoid the transmission of the random locations from the sink to source which will reduce the energy consumption further. We will also work on different encoding strategies in future.

### REFERENCES


**R.Hemalatha** received her B.E. in Electronics & Communication Engineering from PGP College of Engineering & Technology, Namakkal, India in 2003 and M.E in Communication Systems from SSN College of Engineering, Chennai, India in 2005. Currently she is a Junior Research Fellow at SSN College of Engineering.

She has 5 years of teaching experience and two years of research experience. Her area of research includes wireless sensor network issues: energy harvesting and image compression for multimedia applications.

Ms. Hemalatha is a life member of ISTE

**S.Radha** is a member of IEEE. She has graduated from Madurai Kamraj University, India, in Electronics and Communication Engineering during the year 1989. She has obtained her Master degree in Applied Electronics with First Rank from Government College of Technology, Coimbatore and Ph.D. degree from College of Engineering, Guindy, Anna University, Chennai.

She has 22 years of teaching and 11 years of research experience in the area of Mobile Ad Hoc Networks. She also worked as a visiting researcher at Carnegie Mellon University for a period of six month with Prof.Rajkumar in the area of Wireless Sensor Networks. At present she is the HOD of ECE Department, SSN College of engineering. She has 65 publications in International and National Journals and conferences in the area of Mobile Ad hoc Network and Wireless Sensor Networks. Her current areas of research include security and architecture issues of mobile ad hoc networks and wireless sensor networks and Cognitive Radios.

Dr. Radha is a Fellow of IETE and Life member of ISTE. She has received IETE – S K Mitra Memorial Award in October 2006 from IETE Council of INDIA, Best paper awards in various conferences and CTS-SSN Best Faculty Award (2007 & 2009) for the outstanding performance.

**Raghuvarman,N, Soumya.B, Vivekanandan.B** received their B.E in Electronics & Communication Engineering from SSN college of Engineering in 2012. Their area of interest is image compression for multimedia applications in WSN.

---

**TABLE III**

<table>
<thead>
<tr>
<th>PSNR (dB)</th>
<th>BPP</th>
<th>CR (%)</th>
<th>Total energy consumed (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1024</td>
<td>0.1348</td>
<td>59.34, 110</td>
</tr>
<tr>
<td>10</td>
<td>4096</td>
<td>0.2551</td>
<td>31.36, 225</td>
</tr>
<tr>
<td>18</td>
<td>4096</td>
<td>0.3077</td>
<td>26, 260</td>
</tr>
<tr>
<td>22</td>
<td>4096</td>
<td>0.3521</td>
<td>22.72, 280</td>
</tr>
</tbody>
</table>

**Fig.4. Recovered Images**

PSNR=27.56dB, bpp=0.1348

PSNR=29.37dB, bpp=0.2551

PSNR=31.72dB, bpp=0.3077

PSNR=32.86dB, bpp=0.3521