Strip-based Multi-view Image Compression for Visual Sensor Networks

Nguyen Ho Minh Quang[1], Wai Chong Chia[1], KahPhooiSeng[1], Li MinnAng[2]
[1]Department of Computer Science and Networked System
Sunway UniversityKuala Lumpur
Malaysia
[2]School of Engineering
Edith Cowan University
Australia
12057634@imail.sunway.edu.my

ABSTRACT: *This paper proposed an energy efficient multi-view visual processing scheme for the visual sensor networks. The aim of this scheme is to reduce the energy consumption by reducing the amount of data that needs to be transmitted in the network. Two approaches are introduced to achieve this purpose. First, an image compression method based on the Set Partitioning in Hierarchical Trees [1] algorithm is implemented to reduce the size of images before transmitting them over the network. Second, in cases where the Field-of-View of different visual nodes is overlapped, the overlapping regions (redundant information) between the images will be identified at the host workstation to prevent them from being transmitted multiple times. This is to further reduce the data transmission. In this case, only one visual node is required to transmit the overlapping regions. The information obtained from the aforementioned visual node will be used to reconstruct the overlapping regions of other visual nodes at the host workstation. In other words, other visual nodes are only required to transmit the non-overlapping regions. Instead of transmitting the entire image over the network, only part of it will be transmitted. The simulations results indicate that the proposed scheme can reduce the data transmission up to 50%. In addition, strip-based processing is adopted to reduce the memory consumption of the visual node. In this case, the images will be transferred and encoded in a strip-by-strip manner. This can help to lessen the on-chip memory requirement since it is not necessary to process the entire image at the same time.*

Keywords: Visual Sensor Networks, Set Partitioning in Hierarchical Trees, Image Compression, Multi-view Images, Strip-based Processing

Received: 12 July 2012, Revised 11 August 2012, Accepted 18 August 2012

1. Introduction

Nowadays, wireless sensor network (WSN) has becomes one of the most attractive topics in research community, due to its potential in developing new practical applications [2]. WSN is usually deployed to monitor and measure scalar data, such as light intensity, pressure, temperature, etc. Recently, the advancement in micro-electronic technologies has led to the development of Visual Sensor Network (VSN). With the integration of low-power and inexpensive image sensor [3], a VSN is now able to retrieve not only scalar data but also multimedia data such as images, sounds, or video sequences.

However, the use of image sensors in VSN also brings with it a new set of challenges. One of the challenges is the large amount of data generated by the image sensors. Since most of the devices in VSN are powered by batteries, the increase in data transmission, which also resulted in the increase of energy consumption, appears to be one of the critical issues in VSN. In fact, it has been shown in [4] that the energy used for connection and communication is much higher than that of other operations. In this case, the energy consumption of a visual node in an operating cycle (one hour long) was investigated. The visual node consists of a Mircochip PIC16LF877 microcontroller and a 384 × 288 pixels gray scale digital camera with Bluetooth connection. The operating cycle of sensor node was carried out through five stages, which denotes as standby, sensing, processing, connection and communication. The result obtained is shown in Table 1 [4]. It can be observed from Table 1 that the connection and communication are the most energy-consuming operations (consumed more than 90% of the overall energy consumption). In detail, this amount of energy is used to transmit only 108 kB of data, with the total transmission time of 14s.

Operating State	Required Time (mS)	Required Current (mA)	Charge (Current *Time) [A*s]
Standby	3564535	0.0301	0.107
Sensing	2035	46.030	0.093
Processing	0	26.030	0.000
Connection	19207	80.430	1.545
Communication	14223	70.030	0.996
Total	3600000	N/A	2.742

Table 1. Summary of the power budgeting in the hour operating cycle [4]

Hence, many researchers have been trying to reduce the amount of data that needs to be transmitted in the network, in order to reduce the energy consumption of the visual nodes. One of the common solutions is to reduce the data transmission by using compression schemes. The basic idea is to compress the data before transmission takes place. Distributed source coding (DSC) is one of the examples. The DSC is motivated by either Slepian-Wolf [5] or Wyner-Ziv [6] theorems. It allows the compression of multiple correlated sources without the need of any intercommunication between them. The joint decoding of the compressed data is then performed at the host workstation. Although this will increase the computational burden of the host workstation, the simple-encoder complex-decoder paradigm is acceptable in the case of VSN. Following this, various works have been presented [7-9]. However, their performances are still far from the theoretical bound stated in [5] and [6]. This is due to the difficulty in obtaining accurate side information that is required for the joint decoding.

In [10], the authors tried to solve the problem by using a more specific solution, known as image registration. Image registration is used to recognize the overlapping regions and prevents them from being transmitted for multiple times. First, a shape context algorithm is used to extract feature points from correlated images taken by different visual nodes. The overlapping regions are then determined and transmitted between the primary and secondary nodes. Instead of transmitting the images in full-resolution, the secondary nodes are only required to transmit images in lower resolution. At the decoder, these low resolution images will be used to reconstruct the original image by using super-resolution technique. One disadvantage of this approach is that the secondary sensors have to be distributed close to the primary nodes. Moreover, communication between primary and secondary nodes is required to exchange the overlapping information. A similar framework is proposed in [11]. The images have to be transmitted from one node to another by following a fixed route before reaching the decoder. Each node analyses the information from the previous node, and only the non-overlapping information is transmitted to the next node.

Besides, another new theorem based on the sparsity of signal, which also known as Compressed Sensing (CS), is introduced in 2006 [12, 13]. In general, Nyquist sampling theorem stated that a minimum sample rate equals to two times the highest frequency of a signal is required to fully reconstruct the original signal. However, CS stated that if the signal has sparse representation or compressible in a certain transformed domain, it can be reconstructed exactly with a lower sampling rate. Instead of capturing all of the information and then remove the unimportant parts, CS attempts to acquire only the important information. There are several approaches that applied the CS theorem [14-16]. However, the practical implementation of CS theorem is still limited and needs further study. According to [17], noise is one of the common problems when applying the CS theorem. Even though [18] and [19] tried to resolve the noise problem by using objective~~s~~ functions, the performance is still falling behind the theoretical bound.

In this paper, a multi-view visual processing scheme for VSNs that is able to reduce the energy consumption by reducing the amount of transmitted data is proposed. A block diagram of the proposed scheme is shown in Figure 1. In this proposed system, the Set Partitioning in Hierarchical Trees (SPIHT) [1] algorithm is implemented at the visual sensor nodes to compress the images before transmitting them over the network .In cases where the Field-of-View (FoV) of different visual sensor nodes is overlapped, the overlapping regions (redundant information) between the images (multi-view images) will be identified at the host workstation to prevent them from being transmitted multiple times. In this case, only one visual node is required to transmit the overlapping regions, and the information obtained from the aforementioned visual node will be used to reconstruct the overlapping regions of other visual nodes at the host workstation. In other words, other visual nodes are only required to transmit the non-overlapping regions. Instead of transmitting the entire image over the network, only part of it will be transmitted.

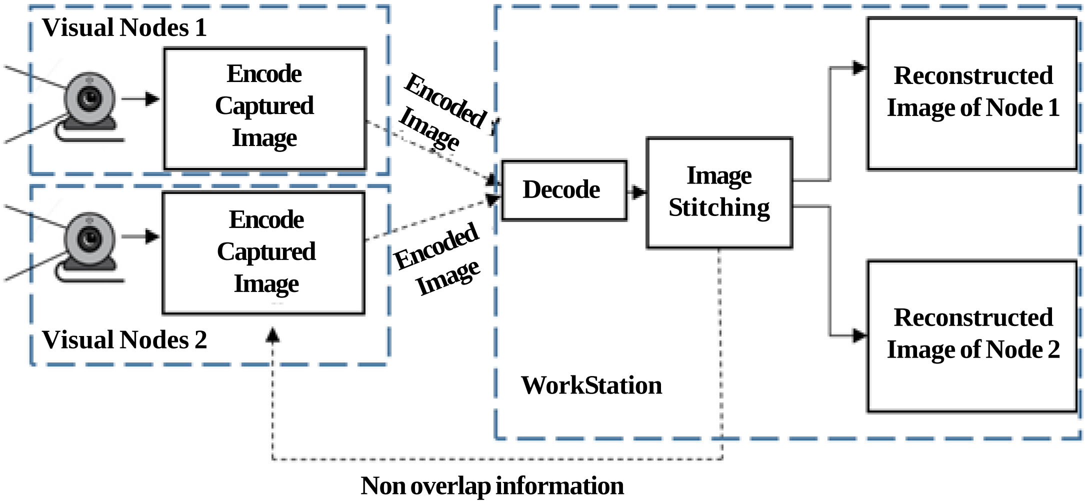

Figure 1. Block diagram of proposed system

Besides reducing the data that has to be transmitted in the network, strip-based processing proposed in [20] is adopted to reduce the need of on-chip memory. Typically, an image has to be transferred to the on-chip memory for processing. If the entire image has to be transferred at once, it would require a large amount of on-chip memory. When strip-based processing is adopted, the captured image will be transferred and processed in a strip-by-strip manner as shown in Figure 2. For example, if a grayscale image with the resolution of 1024×1024 is to be transferred to the on-chip memory at once, 1MB is required to store the image. If the image is to be divided and processed in 16 strips, then only 62.5KB is needed.

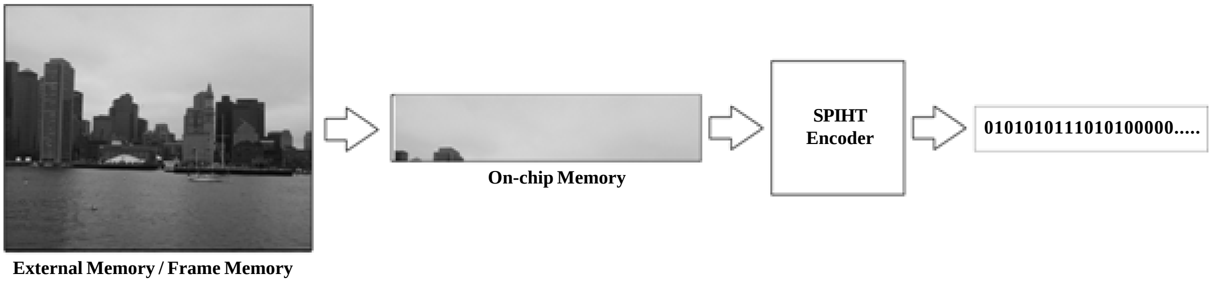

Figure 2. Strip-based processing

This paper is organized as follows. First, the proposed scheme is described in detail in Section 2. This is followed by the image reconstruction process and the modified SPIHT coding that are described in Section 3 and 4 respectively. All the simulation results are presented and discussed in Section 5. Finally, the main findings and the ideas for future works are explained in Section 6.

2. Proposed Scheme

For simplicity, the proposed scheme is explained by assuming that there are only two visual nodes and one joint decoder (workstation) as depicted in Figure 3. However, it should be noted that the proposed scheme can be easily extended to work with more than two visual nodes. The operation of the proposed scheme is described as follows.

Step 1: For the very first time, the two visual nodes will capture the images, compress them by using the region-based SPIHT coding, and then transmit them to the joint decoder. The use of region-based SPIHT coding allows an arbitrary part of the image to be encoded efficiently. This will be explained in more detail in Section 4. It should be noted that the region-based SPIHT will still encode the image in a strip-by-strip basis.

Step 2: At the joint decoder, the two received images will be decoded. Then, image stitching technique is adopted to identify the overlapping regions in each image. A small amount of overlap information is then generated and transmitted back to one of the visual nodes (i.e. node 2). The detail of this step will be described in Section 3.

Step 3: From now on, only one of the visual nodes (i.e. node 1) is needed to compress and transmit the entire image that it captured. Another visual node (i.e. node 2) is only required to encode and transmit the non-overlapping part of the captured image.

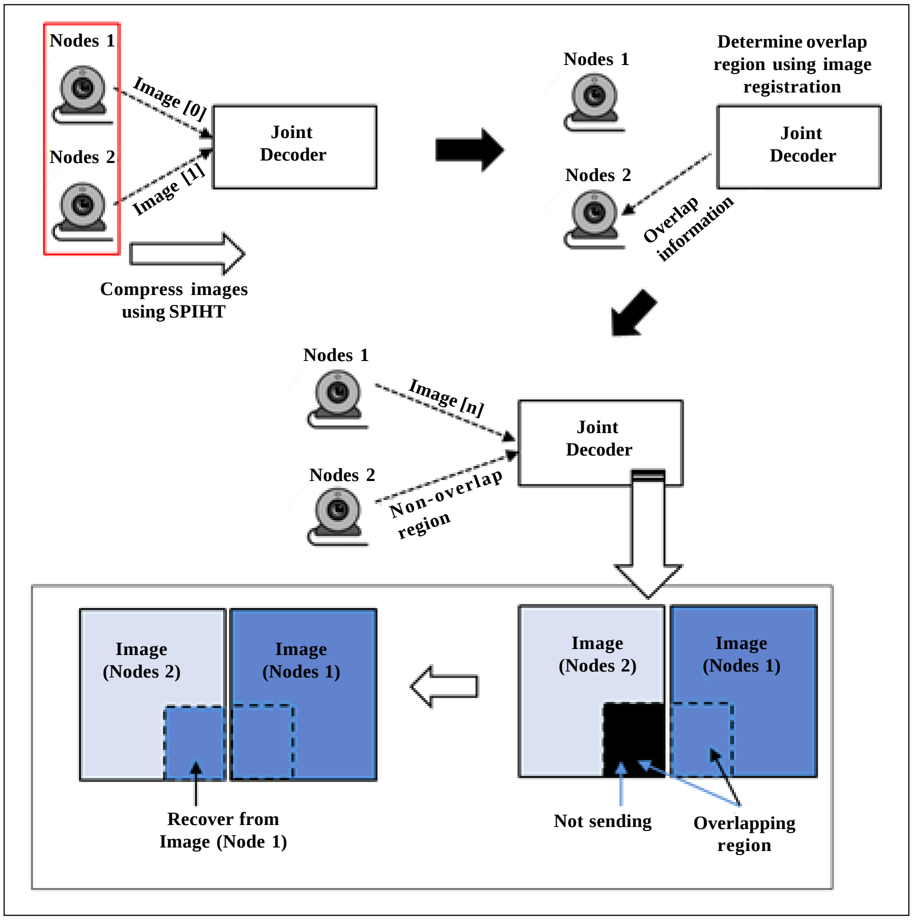

Figure 3. Diagram of proposed scheme

3. Image Reconstruction

The process of reconstructing an image by using the overlap information from another image will be explained in this section. In this case, image stitching is implemented to identify the overlapping regions between different images. The steps that will be performed to complete the reconstruction are described as follows.

3.1 Feature points matching

Feature points in each image will be detected and extracted by using the Laplacian-of-Gaussian (LoG) detector [21]. It is a type of blob (local maxima and local minima) detection algorithm. This algorithm is capable of locating the same feature points even when there are changes in terms of scale and rotation. After the extraction, each feature point is assigned with a "*description*". The "*description*" serves as the identity for each feature point, and it will be used for the matching process.

For simplicity, the LoG detector is adopted in the proposed system. However, in order to handle multi-view images that involve complex changes (i.e. viewing angle, size, illumination and etc.), the LoG detector can be replaced by more advanced techniques, such as the Scale-Invariant Feature Transform (SIFT) [22]. Since the feature extraction algorithm is implemented in the joint decoder (host workstation), the use of a more complex algorithm is acceptable.

3.2 Image matching

After matching the two sets of feature points, the Random Sample Consensus (RANSAC) [23] is used to select the most suitable set of inliers to match the images and calculate the transform parameters between the two images. The transformation parameters will determine how an image will be transformed and placed in the new coordinate system that is agreed by another.

3.3 Determine the overlapping region

After the transformation, the overlapping regions between two images are identified by comparing the two transformed images. Once the overlapping regions in the new coordinate system are known, it is necessary to determine how the overlapping regions will reflect in the original image.

3.4 Reconstruct of overlapping region

An easy way is to shift the pixels of the overlapping region from the transformed coordinates to the original image coordinates (forward mapping). However, forward mapping usually creates holes in unparalleled line transformation. This draw back will affect the quality of our reconstructed image. In this case, inverse mapping is adopted to resolve the aforementioned issue. Instead of determining the new coordinates for each pixel in the transformed image, inverse mapping determines the source coordinates for each pixel in the original image. Inverse mapping is used to fully fill up the overlapping regions by using the information from another image.

4. Image Compression

The image compression algorithm used in this proposed scheme is a combination of the SPIHT coding with the Shape Adaptive Discrete Wavelet Transform (SA-DWT) [24]. In contrast, the original SPIHT coding uses the traditional DWT that can only work on rectangular images. However, in our proposed scheme, only the non-overlapping regions of the image have to be encoded. Therefore, the SA-DWT is used to replace the traditional DWT, due to its ability to transform the arbitrary shaped object (i.e. non-overlapping regions) of an image efficiently for coding with SPIHT.

4.1 Shape-adaptive discrete wavelet transform (SA-DWT)

The SA-DWT presented in [24] is adopted in the proposed scheme. This SA-DWT scheme can be directly applied to encode an arbitrarily shaped region efficiently. It progresses through each row of the segmented objects to obtain the 1-D transform. Based on the shape information, 1-D SA-DWT algorithm treats the arbitrarily shaped region as several continual signal segments and performs the DWT on the texture information for each signal segment independently. The process is then continued by applying the 1-D SA-DWT to each column.

The SA-DWT in this paper is implemented with the global subsampling strategy as introduced in [24] with an exception of the special case of individual isolated pixels (i.e. segments of length $N = 1$). As proposed in [25], individual isolated pixels should be low-pass filtered. But in the proposed scheme, individual isolated pixels are made to follow the global subsampling strategy by applying high-pass filter to odd positioned pixels and low-pass filter to even positioned pixels. The reason of having this exception is to maintain the one-to-one relationship between the transformed domain shape mask and the spatial domain shape mask.

Along with the global subsampling strategy, the memory and computational efficient lifting-based LeGall 5/3-tap DWT filter is also employed [19]. The in-place lifting scheme with symmetrical boundary extension is applied on each segment independently to avoid distortion at the object's border. Additionally, this also helps to ensure that the number of wavelet coefficients after SA-DWT is exactly the same as the number of pixels in the object.

4.2 Modified SPIHT coding

After the image is decomposed into different frequency subbands using the SA-DWT, the SPIHT coding is used to encode the image. However, the original SPIHT coding tends to encode the entire image (include the overlapping regions) which is unnecessary. In our proposed scheme, the SPIHT coding is modified in order to efficiently encode only the non-overlapping regions. The basic idea is to use the shape mask of the overlapping regions to discard the zero trees that contain coefficients belong to the overlapping regions. The shape mask of the overlapping regions is created at the workstation after the overlapping regions between the images are identified. However, instead of sending the entire mask back to the visual nodes, we extracted a much smaller mask that denoted as the LL subband mask to reduce the data transmission. The amount of memory required to store the LL subband mask can be calculated by using Equation (1), whereby *Row* and *Column* are the height and width of the image, and *n* is the level of DWT decomposition.

$$\text{Memory (LL Subband Mask)} = (Row * Column) / (2^n)\ \text{Bit} \qquad (1)$$

In order to create the LL subband mask that can be used by the modified SPIHT coding, the relationship between the wavelet coefficients in different subbands as quad-trees is exploited. The process of creating the LL subband mask is shown in Figure 4. In this case, a coefficient in a lower frequency subband can be thought of as having four descendants in the next higher frequency subband. Starting from the scanning of 2×2 group pixels in the higher frequency subband, a lower level mask will be created. This process is repeated until the size of the mask is equivalent to the size of the lowest frequency subband. If any pixel in the 2×2 group falls inside the object (region of interest), its root will be marked as an inside pixel.

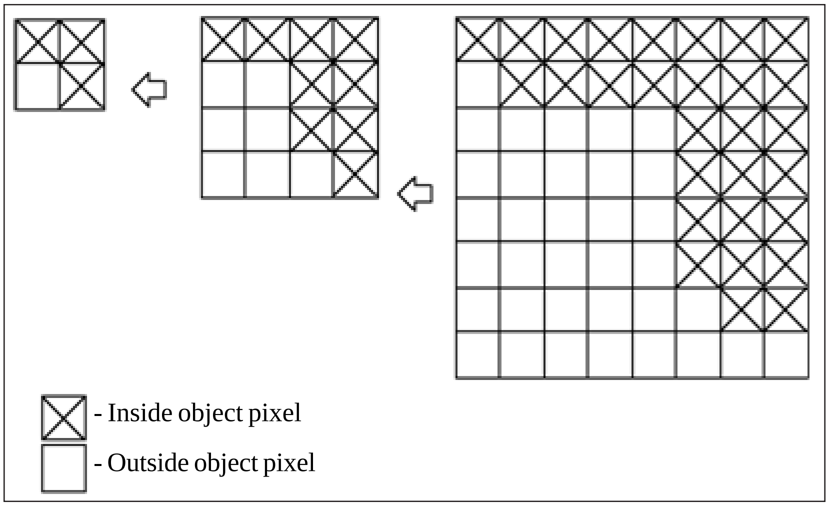

Figure 4. An example of creating the LL subband mask with 2 level of decomposition

After the LL subband mask is created at the workstation, it will be transmitted back to one of the visual nodes. Then, the visual node will make use of this shape mask information to discard the overlapping regions in the next transmission cycle. Our modified SPIHT coding will encode the image according to the mask. At the initialization stage, only the coefficients that belong to the non-overlapping region will be added to the List of Insignificant Pixels (LIP), whereas the others will be discarded. Before adding any entry into the List of Insignificant Sets (LIS), the mask is scanned in the form of 2×2 group pixels. If any one of them (except the "*Star*" coefficient that does not has any descendant) is marked as "*inside*" (i.e. "1") then their coordinates will be added into the LIS, whereas if all of them are marked as "*outside*" (i.e. "0"), then none of them will be added into the LIS. This is to ensure that no distortion can be observed at the border of the encoded object. The rest of our modified SPIHT coding is similar to the original.

5. Simulation Results

The performance of the proposed scheme is presented in this section. In the simulation, the images are assumed to be captured and sent to the workstation for the first time. At the workstation, we tried to determine the overlapping regions between the images and reconstruct one of them using the information obtained from another.

In the simulation, one visual sensor node is assumed to have received the LL subband mask from the workstation. Hence, the visual node will only compress the non-overlapping regions of the captured image. In this case, the two images denoted as "*Boston*" in the data set obtained from [26] are adopted. The two original images (with size of 1712×1368 pixels) are resized to 512×512 pixels by using cubic interpolation method, and they are converted to gray-scale as shown in Figure 5. The testing images are resized in order to accommodate the use of the modified SPIHT coding. The steps applied throughout the simulation are described in more detail in the next two subsections.

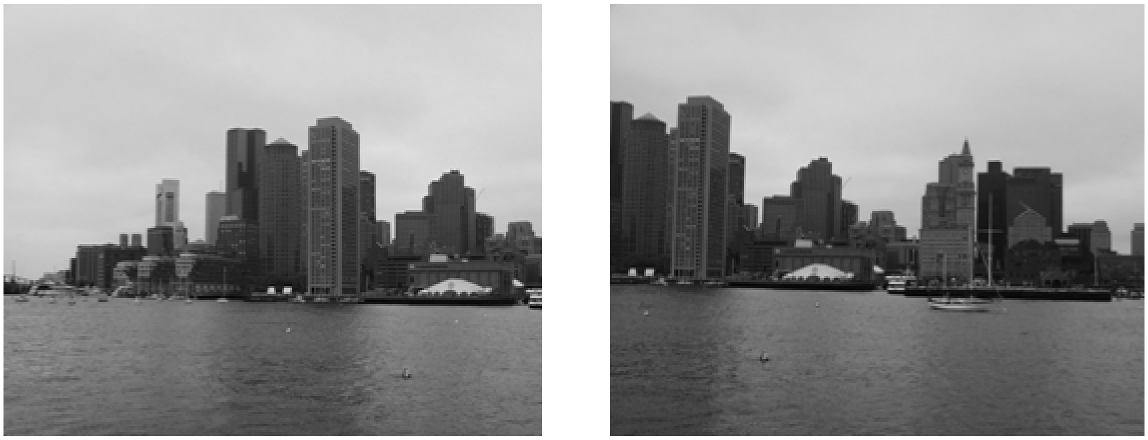

Figure 5. The two images used in the simulation (left: image *A*; right: image *B*)

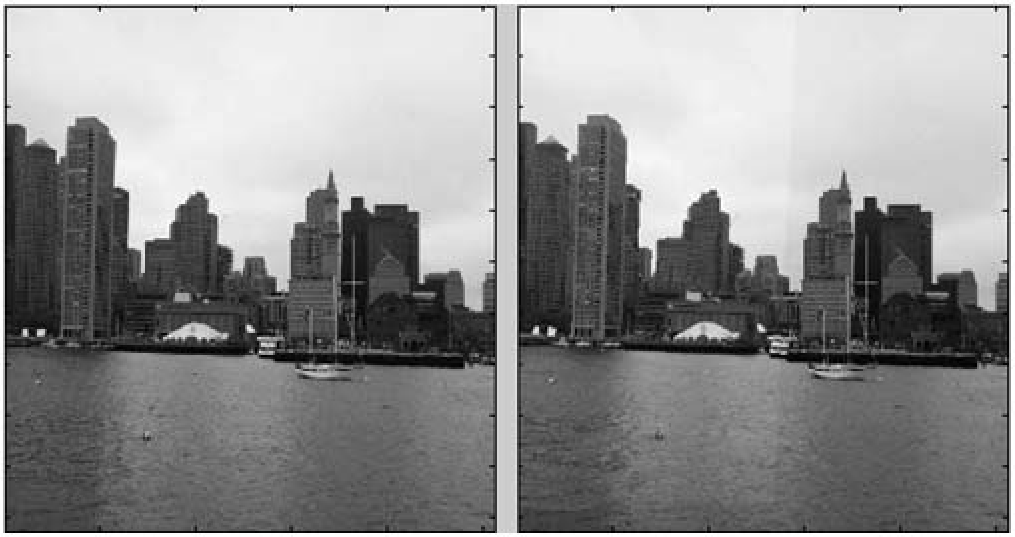

Figure 6. Original image *B* (left) and reconstructed image *B* (right)

5.1 Image reconstruction

1. Extract the feature points between the two images by using the LoG detector [21].

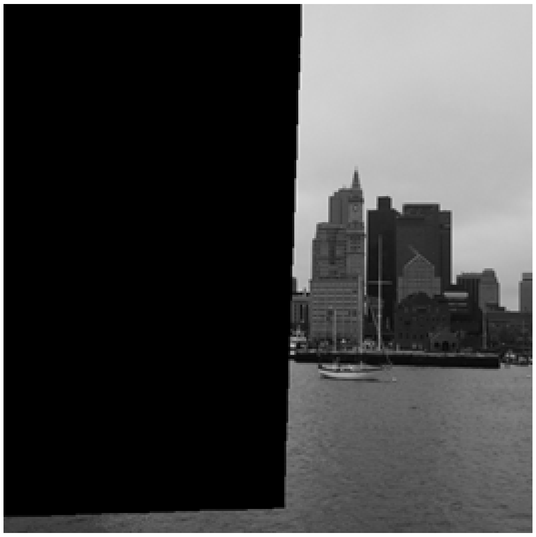

Figure 7. Non-overlapping region

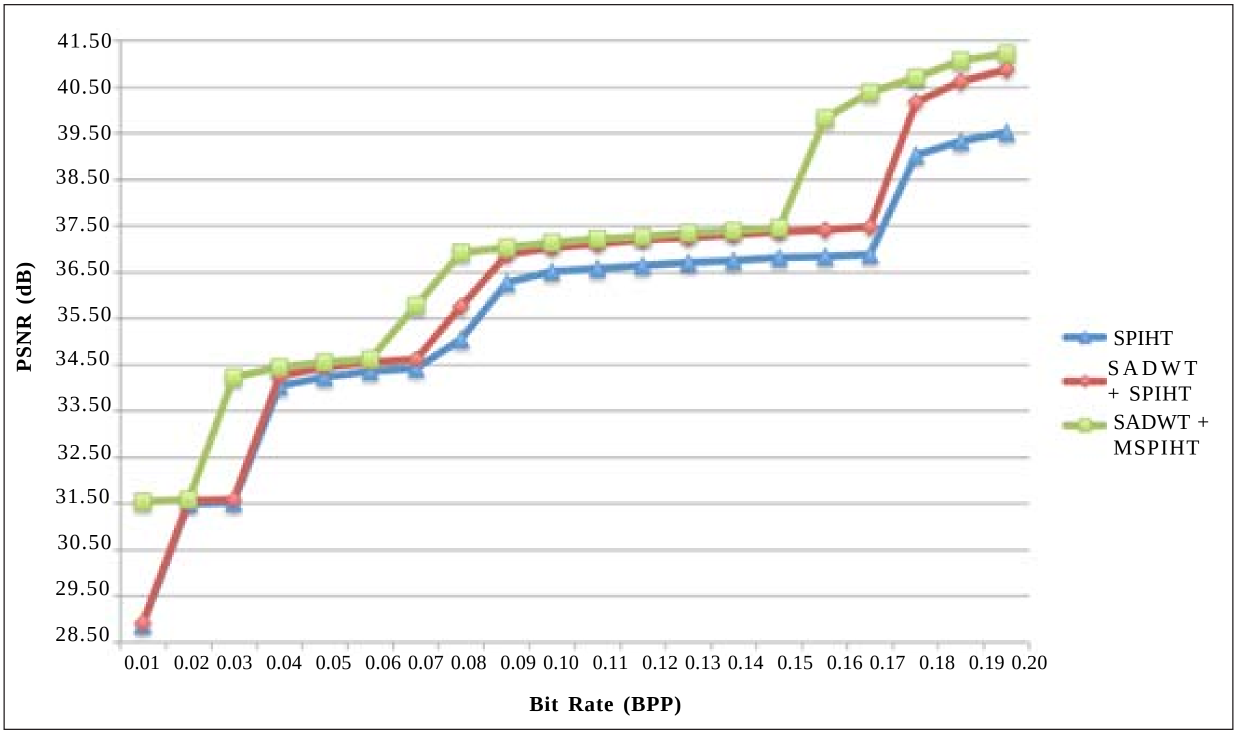

Figure 8. PSNR versus bitrate between SA-DWT and DWT

2. Each extracted feature point in the two images will be given a descriptor vector for matching purpose. The transformation parameters between the two images are calculated using RANSAC [23].

3. By using the transformation parameters obtained from previous step, a transformation matrix is created to transform the right image (*B*) to the same coordinate system agreed by the left image (*A*).

4. The overlapping regions in image *A* are identified by comparing image *A* with transformed image of *B*. Based on the overlapping regions observed, the overlapping regions in original image *B* will be reconstructed by using the information from image *A*. This is achieved by using the inverse mapping.

BPP	PSNR (dB)						
	Non- Strip-based			Strip-based (SA-DWT + M SPIHT)			
	DWT* **+** *SPIHT	***SA-DWT* **+** *SPIHT***	***SA-DWT* + *MSPIHT* (= 1 *Strip*)**	**2 *Strips***	**4 *Strips***	**8 *Strips***	**16 *Strips***
0.01	28.89	28.93	31.53	31.54	31.55	31.68	**32.02**
0.02	31.50	31.57	31.58	31.58	**33.42**	33.28	32.04
0.03	31.52	31.58	**34.24**	34.15	32.73	32.47	33.19
0.04	34.03	34.25	**34.44**	34.33	**34.44**	33.63	33.38
0.05	34.23	34.44	34.56	35.23	34.48	**35.40**	33.90
0.06	34.36	34.56	34.61	**35.68**	34.56	35.09	34.30
0.07	34.43	34.61	35.79	35.72	34.59	35.17	**35.81**
0.08	35.06	35.77	**36.93**	35.80	35.85	36.53	36.61
0.09	36.28	36.87	**37.03**	36.10	35.94	36.59	36.61
0.10	36.52	37.03	37.14	**37.95**	35.98	36.62	36.78
0.11	36.59	37.12	37.21	**38.14**	36.01	37.18	36.91
0.12	36.65	37.19	37.27	38.25	36.82	37.40	**38.34**
0.13	36.70	37.25	37.35	38.35	36.88	37.34	**38.45**
0.14	36.76	37.31	37.40	38.42	37.15	37.95	**38.53**
0.15	36.81	37.38	37.45	38.47	**39.43**	38.65	38.74
0.16	36.84	37.43	39.83	39.34	**40.31**	39.76	38.91
0.17	36.89	37.48	40.39	39.45	**40.49**	39.85	39.04
0.18	39.03	40.16	**40.72**	39.55	40.65	39.99	39.10
0.19	39.34	40.62	**41.07**	39.60	40.74	40.07	39.27
0.20	39.53	40.88	**41.23**	39.64	40.81	40.16	39.32

Table 2. PSNR (dB) achieved at different bit rate (BPP) by using
various non strip-based and strip-based compression schemes

5. Finally, the reconstructed overlapping regions are combined with the non-overlapping part of original image *B*. The complete reconstructed image *B* is shown in Figure 6. When compared with the original image *B,* it can be observed that there is only a small difference between the two images, and it is difficult to distinguish.

5.2 Image compression

Before strip-based processing is applied, the coding performance gain that can be achieved with the use of SA-DWT and modified SPIHT coding is evaluated. To achieve this, the image shown in Figure 7 is compressed by using the combination of DWT with original SPIHT coding (SPIHT), SA-DWT with original SPIHT coding (SADWT+SPIHT), and SA-DWT with modified SPIHT coding (SADWT+MSPIHT). The PSNR (dB) achieved at different bit rate is presented in Figure 8. Four levels of decomposition and LeGall 5/3-tap filter are adopted in all the combinations. It should be noted that all the negative pixel values produced from the reconstruction of the image are rounded to zero, since in grayscale image, the pixel value can only range from 0 to 255.

By referring to the simulation results shown in Figure 8, it can be seen that the combination of SA-DWT with the modified SPIHT coding is more efficient than the other two combinations. Hence, strip-based processing is directly applied to the former

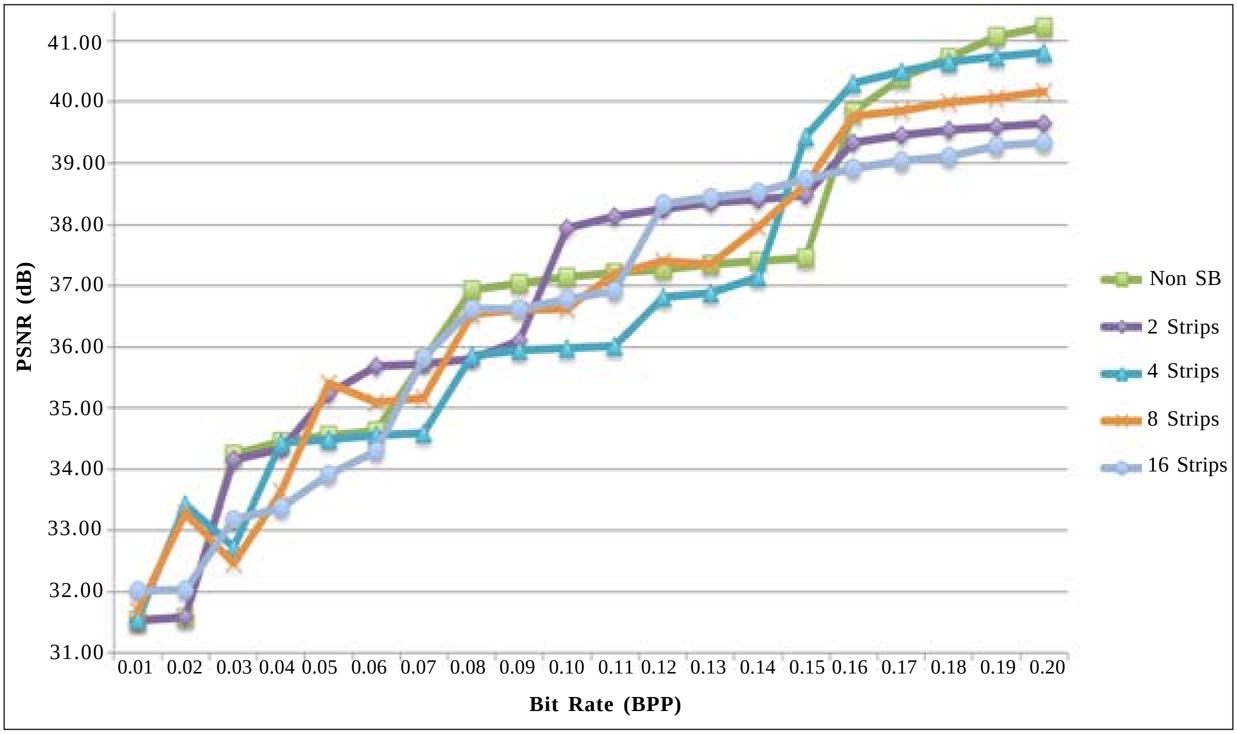

Figure 9. PSNR versus bitrate between original and modified SPIHT

combination instead of the latter two. The coding performance of dividing the image into different number of strips is evaluated and the simulation results is shown in Figure 9. In this simulation, the image is divided into 2, 4, 8, and 16 strips respectively. Then, the combination of SA-DWT with the modified SPIHT coding is applied to encode and decode the strips one-by-one. The coding performance achieved with different number of strips is compared to the case where no strip-based processing is applied (Non SB). For the ease of comparison, the PSNR (dB) used to plot the graphs presented in Figure 8 and Figure 9 is summarized in Table 2.

From the simulation results shown in Figure 9, it can be noticed that the use of strip-based processing only causes the coding performance to drop at high bit rate (³ 0.18 BPP). At lower bit rate (< 0.18BPP), the use of strip-based processing may help to improve the coding performance, but the gain is not very consistent. Overall, it is worth to apply strip-based processing when the image is to be encoded at low bit rate. Not only that it may help to improve the coding performance, it can reduce the amount of on-chip memory that is needed to process the image. Consider the case where the image is divided into 16 strips, the size of each strip is only 32 ´ 512. In this case, only 16 KB of on-chip memory is required.

6. Conclusion and Future Works

Through the result obtained from the simulation, there is only a small difference between the original image and reconstructed image. Hence, it is not always necessary to transmit the entire image. Instead, only the non-overlapping regions of image need to be encoded and transmitted. Moreover, the result shows that the amount of image data that need to be transmitted has been greatly reduced, after the SA-DWT and modified SPIHT coding are adopted. For example, to encode the entire image, the original SPIHT coding needs 50.710 kB. In contrast, if only the non-overlap part of the image be encoded (with the modified SPIHT coding), it needs only 23.548 kB. Therefore, more than 50% amount of data is reduced with our proposed scheme. In addition, the possibility to incorporate strip-based processing with the SA-DWT and modified SPIHT coding has been investigated. The simulation shows that the use of strip-based processing may help to improve the coding performance at lower bit rate (<0.18BPP). Therefore, if the image is to be encoded at low bit rate, it is worth to apply strip-based processing. For future works, the proposed scheme could be implemented into hardware in order to construct a real VSN to evaluate the efficiency of the proposed scheme in real-world situation. Besides, more advanced techniques can be applied to improve the performance of the system. For example, the distortion at the border between the non-overlapping and overlapping regions can be neutralized by using image blending.

References

[1] Said, A., Pearlman, W. A. (1996). A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Tree, *IEEE Transactions on Circuits and System for Video Technology*, 6, p. 243-250, June.

[2] Merrett, G. V., Al-Hashimi, B. M., White, N. M., Harris, N. R. (2005). Resource aware sensor nodes in wireless sensor networks, *Journal of Physics: Conference Series*, 15, p. 137-142.

[3] Ian, T. M., Akyildiz, F., Chowdury, Kaushik, R. (2007, December 2007) Wireless Multimedia Sensor Networks: A Survey. *IEEE Wireless Communications*. p. 32-39.

[4] Ferrigno, L., Marano, S., Paciello, V., Pietrosanto, A. (2005). Balancing computational and transmission power consumption in wireless image sensor networks, presented at the VECIMS 2005 - *IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems*, Giardini Naxos, Italy.

[5] Slepian, D., Wolf, J. K. (1973). Noiseless Coding of Correlated Information Sources, *IEEE Trans. Inf. Theory*, IT-19, p. 471-490, July.

[6] Wyner, D., Ziv, J. (1976). The Rate-Distortion Function for Source Coding with Side Information at the Decoder, *IEEE Transaction on Information Theory*, IT-22, p. 1-10, January.

[7] Pradhan, S. S., Ramchandran, K. (1999). Distributed source coding using syndromes (DISCUS): Design and construction, presented at the Procceding IEEE Data Compression Conference, Snowbird, *UT*.

[8] Pradhan, S. S., Ramchandran, K. (2000). Distributed source coding: Symmetric rates and application to sensor networks, presented at the Procceding IEEE Data Compression Conference, Snowbird, *UT*.

[9] Sehgal, A., Jagmohan, A., Ahuja, N. (2004). Wyner-Ziv coding of video: An error-resilient compression framework, *IEEE Trans. Multimedia*, 6, p. 249-258, April.

[10] Wagner, R., Nowak, R., Baraniuk, R. (2003). Distributed image compression for sensor networks using correspondence analysis and super-resolution, *In*: Proc. IEEE Int. Conf. Image Processing, p. 597-600.

[11] Wu, M., Chen, C. W. (2007). Collaborative image coding and transmission over wireless sensor networks, *EURASIP Journal Advance Signal Processing*.

[12] Donoho, D. (2006). Compressed sensing, *IEEE Transactions on Information Theory*, 52, p. 1289-1306.

[13] Candes, E. (2006). Compressive sampling, *In*: Proceedings of the International Congress of Mathematicians, Madrid, Spain, p. 1433-1452.

[14] Trocan, M., Maugey, T., Tramel, E. W., Fowler, J. E., Pesquet-Popescu, B. e. (2010). Multistage Compressed-Sensing Reconstruction of Multiview Images.

[15] Pudlewski, S., Melodia, T., Prasanna, A. (2012). Compressed-Sensing-Enabled Video Streaming for Wireless Multimedia Sensor Networks, *IEEE Transactions on mobile computing*, 11, p. 1060-1072, June.

[16] Trocan, M., Maugey, T., Tramel, E. W., Fowler, J. E., Pesquet-Popescu, B. e. (2010). Compressed sensing of multiview images using disparity compensation, *In*: Proceedings of 2010 IEEE 17th International Conference on Image Processing, Hong Kong, p. 3345-3348.

[17] Wan-zheng, N., Hai-yan, W., Xuan, W., Fu-zhou, Y. (2011). The Analysis of Noise Reduction Performance in Compressed Sensing, *IEEE International Conferrence on Signal Processing, Communications and Computing* (ICSPCC).

[18] Donoho, D. L., Tsaig, Y. (2006). Extensions of compressed sensing, *Signal Processing*, 86, p. 533-548.

[19] Figeiredo, M. A. T., Nowak, R. D., Wright, S. J. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing*, 1, p. 586-598.

[20] Bhattar, R. K., Ramakrishnan, K. R., Dasgupta, K. S. (2002). Strip based coding for large images using wavelets,” Signal Processing, 17 (6) 441-456.

[21] Lindeberg, T. (1998). Feature Detection with Automatic Scale Selection, *IEEE Transactions Pattern Analysis Machine Intelligence*, 30, p. 77-116.

[22] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 60, p. 91-110.

[23] Dirchler, M., Bolles, R. (1982). Random sample consensus: a paradign for model fitting with application to image analysis and automated catography. *Communications of the ACM*, p. 381-395.

[24] Li, S., Li, W. (2000). Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding, *IEEE Trans. Circuits Syst. Video Technol.*, 10, p. 725-743, August.

[25] Zhang, G. X. J. L. S. L. Y.-Q . (2001). Arbitrarily shaped video-object coding by wavelet, *IEEE Trans. Circuits Syst. Video Technol*., 11, p. 1135-1139, Oct.

[26] Stewart, C., Yang, G. (December). Available: http://www.vision.cs.rpi.edu/gdbicp/dataset/