Strip Based Embedded Coding of Wavelet Coefficients for Large Images

Raghunadh K Bhattar
EE Department
Indian Institute of Science
Bangalore
raghu@sac.ernet.in

K.R.Ramakrishnan
EE Department
Indian Institute of Science
Bangalore
krr@ee.iisc.ernet.in

K.S.Dasgupta
ADCTG
Space Applications Centre
Ahmedabad.
ksd@sac.ernet.in

Abstract

Wavelet based embedded coders such as EZW and SPIHT require full wavelet transform of the image to be buffered for coding. Further, since the transform coefficients are required to be stored in high precision, the buffering requirements are prohibitively high for large images, such as remote sensing images, space borne images, medical images etc. In this paper we investigate, embedded coding of wavelet coefficients using zero trees, with reduced memory requirements, for large images. The wavelet coefficients are buffered in a 'strip buffer', capable of holding few lines of wavelet coefficients from all the subbands belonging to the same spatial location. For above applications, we also develop a pipeline architecture for real time embedded coding.

1 Introduction

Images acquired through remote sensing satellites are generally large in size. These images, acquired line-by-line by an optical sensor, are to be transmitted in compressed form on a constant bit rate channel to ground station for archival and analysis purpose.

Wavelet based compression techniques, such as EZW and SPIHT have become benchmark techniques for state of art compression. Besides, better compression and better image quality, they provide embedded bitstream, which can be truncated at any point and reconstruction carried out at varying quality. The price to be paid for these advantages is in the form of large memory requirements. For coding, the transform coefficients of full image need to be buffered in high precision, thus making memory requirements a bottle neck for hardware implementation. This problem is very serious in compression of satellite images on-board, where memory is limited because of power constraints. While wavelet transform can be implemented by buffering only few lines[4], embedded coding is non-compatible with low memory environment. In this paper, we propose a strip based embedded coding for large images, with minimal buffering of wavelet coefficients. The image is processed line-by-line and the wavelet transform is performed over a set of scan lines. The wavelet coefficients are buffered in a strip buffer for embedded coding. The strip based embedded wavelet image coder is shown in fig.1. The size of the strip is dependent on the image width and number of levels in wavelet decomposition, but independent of wavelet filter length and image height.

JPEG2000, the emerging still image compression standard addresses the compression of large images as a part of its standardization process[1]. Line based wavelet transform is being considered for compression, with limited memory utilization . This paper achieves the embedded coding of large images with reduced memory requirements using line based wavelet transform.

The paper is organized as follows. Section **2**, we briefly describe the line based wavelet transform using absolute low memory[4]. The advantages of embedded coders such as EZW[3] and SPIHT[2] are outlined in section **3**. Section **4** explains the proposed strip based embedded image coding. A pipeline architecture for real time strip based coding is described in section **5** and compression results are reported in section **6**.

2 Line Based Implementation of Wavelet Transform

A separable wavelet transform can be implemented using only 2L equivalent number of image lines, where L is the filter length. The L lines of input image need to be buffered (line width equal to image width 'W') for first level decomposition. This L line buffer acts as a sliding window for vertical decomposition. After sufficient number of lines acquired (refer section **5**), which depends on the filter length, the filtering is carried out in the vertical direction. The vertical filtered outputs, approximate coefficients(low pass output) and detail coefficients(high pass output), having the same width as image, are subjected for horizontal wavelet decomposition. The approximate coefficients from lowpass filtered lines corresponding to Low-Low(LL) subband, are sent for next level(i.e. second level) decomposition and are buffered in another line buffer with L lines, with ½ of the original image width i.e. W/2. The other three set of lines from Low-High(LH), High-Low(HL) and High-High(HH) subbands are sent for coding. For third level decomposition again, a L line buffer, with the width equal to 'W/4', is needed for buffering the lines from the LL subband of the second level and so on. For any number of levels of decomposition, the total memory is upper bounded by $\sum_{n=0}^{\infty} 2^{-n} L = 2L$ equivalent number of image lines. The wavelet coefficients obtained are exactly equal to the coefficients, which would have been obtained while the full image is buffered.

3 Advantages of Embedded Coders

The embedded coders such as EZW and SPIHT generate the bits in the bit stream in the order of importance. They provide a better rate-distortion tradeoff and can be truncated at any point. The truncated bit stream can be reconstructed with the quality corresponding to the bit rate. Thus the target bit rate can be met exactly without any complicated models. One can use embedded coding for progressive transmission of wavelet coefficients. This enables the embedded coders to compute reduction in distortion, at every instance and encoding can be stopped when the target distortion measure is reached. Another significant advantage of the embedded coders lies in the way the bit planes are transmitted with reduced number of alphabet size for coding. The reduced number of alphabet size increases the coding gain[13]. In view of these advantages the embedded coders become synonymous with wavelet based coding. Embedded coders generally require maintenance of several 'lists' and complicated data structures which consume huge memory, besides memory required for buffering the full wavelet image. The PCs will fail for moderately large images and even the work station fail to code very large images. One way to overcome this problem is to resort to tile the image. Tiling in spatial domain is more involved with boundary problems and overlapping problems. Alternatively, the wavelet coefficients can be partitioned for coding.

We are proposing a strip based embedded coding for large images in wavelet domain using zerotree structure. The embedded coding is done strip wise on the wavelet coefficients, to exploit the efficient coding of bit planes inherent with the embedded coders.

4 Strip Based Embedded Coding

The wavelet decomposition and zerotree relationship for embedded coding for full wavelet image is shown in fig.2. Zerotrees are formed with the wavelet coefficients belonging to same spatial location in all levels of decomposition. Hence it is possible to collect the wavelet coefficients corresponding to the same spatial location from all the subbands and group them in a small strip of memory as shown in fig.3. The strip width is equal to image width in horizontal direction and height is equal to 2^N lines in vertical direction where, N is the number of levels in wavelet decomposition. The wavelet transform can be implemented with low memory as described in the section II, and the wavelet coefficients generated can be collected into a strip, while the image is being continuously acquired from the sensor. In a strip, we have one line in each subband LL, HL, LH and HH with line width equal to $W/2^N$ in Nth level and 2^{N-1} lines in first level width equal to W/2. Line wise occupation of subbands in a strip is shown in fig.4. The embedded coding is applied on this strip. EZW algorithm can be implemented on this strip without any modifications. The SPIHT algorithm requires a group of 2x2 adjacent coefficients in LL band. The memory requirements double to meet this condition . So this condition can be relaxed and SPIHT algorithm may be implemented with zerotree roots starting from HL, LH and HH rather than from LL band, with a little loss of performance. As a further tradeoff between performance and memory requirements, we can apply Listless Zerotee Coding(LZC) algorithm[12] on this strip.

Once the embedded coding is done on one strip, the other strip with another set wavelet coefficients should be ready for coding. Alternatively, the coding of one strip should be finished before the other strip is ready for real time encoding. This requires a pipeline architecture which is discussed in the next section.

5 Pipeline Architecture for Strip Based Coding

In real time applications, the data from the source is continuously available and the data is also continuously used by the end user. So the processing units in between the source and end user, such as compression engine, transmission media etc., should also be capable of receiving the data continuously and deliver the processed data to the next processing unit or to the end user continuously as the case may be. In on-board applications, the remote sensing images are acquired continuously from the sensor and they should be compressed for efficient utilization of bandwidth in real time without having to buffer the full image. The compression engine should work in pipeline mode. We now propose a pipeline architecture for strip based embedded coding of wavelet coefficients described in the previous section.

The pipeline architecture is described for 9-7 bi-orthogonal wavelets, but can be generalized for any bi-orthogonal wavelets. Bi-orthogonal wavelets are preferred, because of their linear phase characteristics in image compression. Another advantage with the bi-orthogonal filters is, they can be implemented with symmetric extensions which is compatible with on-line data.

Horizontal decomposition can be done easily, as the entire line is available in buffer. But vertical decomposition need to be done progressively as they are generated from sensor. For 9-7 bi-orthogonal wavelets, the vertical decomposition can be started with symmetric extensions, after acquiring 5 lines. When these five lines are extended with whole sample symmetry at boundary, they occupy 9 lines. Thus the first line in approximate and detail component in level one, are acquired in vertical direction after 5 lines of input. The vertically decomposed lines are then subjected to horizontal decomposition, producing each line in HL, LH and HH sub-bands. The line belonging to LL band processed for next level of decomposition. In the sub-sequent levels also the same symmetric extension is followed.

Now we reorder these coefficients in a strip with 2^N lines as shown in the fig.3, for zerotree coding. The first line in any level is generated after 5 lines are generated in the previous level. Due to this initial delay, the wavelet coefficients belonging to the strips are not generated continuously for N > 1. In one strip we accommodate 2^{N-1} lines belonging to first level, 2^{N-2} lines belonging to second level and so on, and finally only one line,(i.e. $2^0 = 1$)in the Nth level for each subband. The fully occupied strip is shown in the fig.4. Once the first line in Nth level is generated, the strip is released for coding. The number of lines generated in the lower levels, in wavelet decomposition, far exceed the capacity allotted to that level in a strip, before the first line is filled. This calls for more strips, until the line in Nth level is generated in a strip. Occupation of the Nth level guarantees the completeness of the strip. A fully occupied strip is released for coding. The strip which is just used for coding is put back for receiving the data, to maintain the pipeline. Now the question is, how many strips are required for maintaining the pipeline?. The following analysis answers the question.

PresentNL = Number of lines acquired in the present level.
PreviousNL = Number.of lines acquired in the previous level.

$$PresentNL = \left\lfloor \frac{PreviousNL - 5}{2} \right\rfloor + 1 \qquad (1)$$

The delay of 5 lines is because of the wavelet filter length.

N = Number of wavelet decompositions
It can be shown that the number of lines acquired in the level one, when the first line is acquired in the Nth level is ($2^{N+1} - 3$). So we require ($2^{N+1} - 3$) lines to be buffered before releasing the first strip.
Number of lines in the level one in any strip is equal to 2^{N-1}. Hence the number of strips
required to maintain the pipeline is equal to

$$= \left\lceil \frac{2^{N+1} - 3}{2^{N-1}} \right\rceil = \left\lceil 4 - \frac{3}{2^{N-1}} \right\rceil \qquad (2)$$

for N = 1, the number of strips required is one. For N = 2, the numbers of strips required is ‘3’. For N > 2, the number of strips required is ‘4’. This is an important relation shows that for any number of levels, the strips required to maintain the pipe line is only ‘four’. For 9-7 bi-orthogonal wavelets and N = 4, the first strip is fully occupied after 67 lines of input as shown in fig.5a. The status of remaining three strips is shown in fig5b-d. At this point, the fully occupied strip1 is used for embedded coding. Now the strip2 is designated as strip1, strip3 as strip2, strip4 as strip3 and the empty strip4 is added to pipeline. After the first strip released for coding, the subsequent strips with wavelet coefficients are available for every 32 lines of input.

6 Discussion and Simulation Results

The 9-7 biorthogonal[5] wavelets were used for simulation. Symmetric extensions [6],[7] were used at boundaries in both horizontal and vertical directions. The SPIHT algorithm is used, with relaxed condition, for embedded coding given in section **4**. The computational complexity of the coder remains same as that of the SPIHT algorithm. The compression results are reported with binary-uncoded data. No arithmetic coding or any other entropy coding was used. The target bit rate was divided equally among all the strips. The decoder need to decode all the bits encoded by the encoder for each strip to maintain the synchronization between encoder and decoder, which is not a mandatory requirement for embedded coders when whole image is coded as one unit. The strip based algorithm was applied on a remote sensing image and JPEG2000 test image ‘bike’. The results were compared with JPEG compression algorithm. The compression results for embedded coding were tabulated in table.1. The compressed image coded at the rate 0.25bpp is shown in fig.6b and the original image is given in fig.6a. The same algorithm was applied on small images like ‘lena’, goldhill’, and ‘barbara’ of size 512x512. The PSNR values were slightly inferior to SPIHT for same bit rate, due to reduced memory and lack of global information. The quality obtained was very close to that of the EZW algorithm. The rate vs PSNR values in dB are tabulated in table.2.

The line based wavelet transform which is reported in[4], is based on context modeling and classification along with arithmetic coding and probability estimation. The above techniques are based on input image statistics and the results may vary from image to image and may not be as robust as SPIHT or EZW. Hence it is not considered here for comparison.

7 Conclusion

In this work, we have presented a strip based embedded coding for large images for real time applications. This algorithm reduces the memory requirements significantly and retains all the advantages of embedded coding. The performance degradation is minimal, in view of reduced memory requirements. We have also developed a pipeline architecture for strip based coding, for real time applications.

ACKNOWLEDGMENT

The authors are grateful to Christos Chrysafis, Hewlett-Packard and Antonio Ortega, Integrated Media Systems Center for their executable code and test image provided.

References

[1] *SO/IEC JTC 1/SC 29/WG1 (ITU-T SG8), Coding of Still Pictures, JBIG, JPEG*. October 5, 1998.

[2] A. Said and W. Pearlman, "A New fast and Efficient Image coder Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. On Circuits and systems for Video Technology*, vol 6, no.3, pp 243-250, June 1996.

[3] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Trans.onSignal Processing*, vol.41, pp 3445-3462, Dec 1993.

[4] Christos Chrysafis, Antonio Ortega, "Line Based, Reduced Memory, Wavelet Image Compression," *IEEE Trans.On Image Processing*. Vol:9, No.3, March2000, pp 378-389.

[5] C.Sidney Burrus, Ramesh A. Gopinath, and Haitao Guo, *Introduction to Wavelets and wavelet Transforms*, Prentice-Hall International, inc, 1998.

[6] Pankaj N. Topiwala *Wavelet Image and Video Compression*, Kluwer Academic Publishers, 1998.

[7] Christopher M. Brislawn. "Classification of non-expansive symmetric extension transforms for multi rate filter banks," *Appl. Comput. Harmonic Anal*. 3:337-357, 1996.

[8] M.Viswanath, "The recursive pyramid algorithm for the discrete wavelet transform," *IEEE trans.on Signal Processing*, vol.42, pp 673-676, March 1994

[9] P.Cosman and K.Zeger, "Memory Constrained wavelet-Based Image Coding," *Signal Processing Letters*, vol.5, pp 221-223, September 1998.

[10] Stephen A. artucci, Iraj Sodagar, Tihao hiang, Ya-Qin, "A Zerotree Wavelet Coder," *IEEE Trans.on Circuits and systems for Video technology*, vol.7, no.1 pp 109-118, Feb 1997.

[11] Gregory K.Wallace, "The JPEG Still Picture Compression Standard," *Communications of ACM*, vol.34, no.4, pp 30-44, April 1991.

[12] Wen-Kuo Lin, Brian Wai-Him Ng, and Nei Burgess, "Reduced memory Zerotree Coding Algorithm for Hardware Implementaion" *Proceedings of the IEEE International Conference on Multimedia Computing and Systems* Vo.2, pp 57-61, 1999

[13] Jianhua Chen, Tian-hu Yu and Xiaolin Wu "Non-Embedded Wavelet Image Coding Scheme", *ICIP-97*, pp 584-587.

Table.1 Coding Results on Test Images

Rate(bpp)	Remote Sensing Image(2048x2048)		'BIKE' Image (2560x2048)	
	JPEG (PSNR in dB)	Proposed Method (PSNR in dB)	JPEG (PSNR in dB)	Proposed Method (PSNR in dB)
1.00	45.64	45.68	33.82	34.82
0.50	42.62	42.88	29.89	30.37
0.25	38.90	40.25	26.28	26.49

Table 2. The proposed algorithm applied to standard images of size 512x512. PSNR values in dB.

Rate (bpp)	Lena			Goldhill		Barbara	
	EZW	SPIHT	Proposed Method	SPIHT	Proposed Method	EZW	Proposed Method
1.00	39.55	40.00	39.10	36.00	34.68	35.14	34.06
0.50	36.28	36.80	35.76	32.70	31.52	30.53	29.38
0.25	33.17	33.80	32.20	30.10	29.05	26.77	25.95

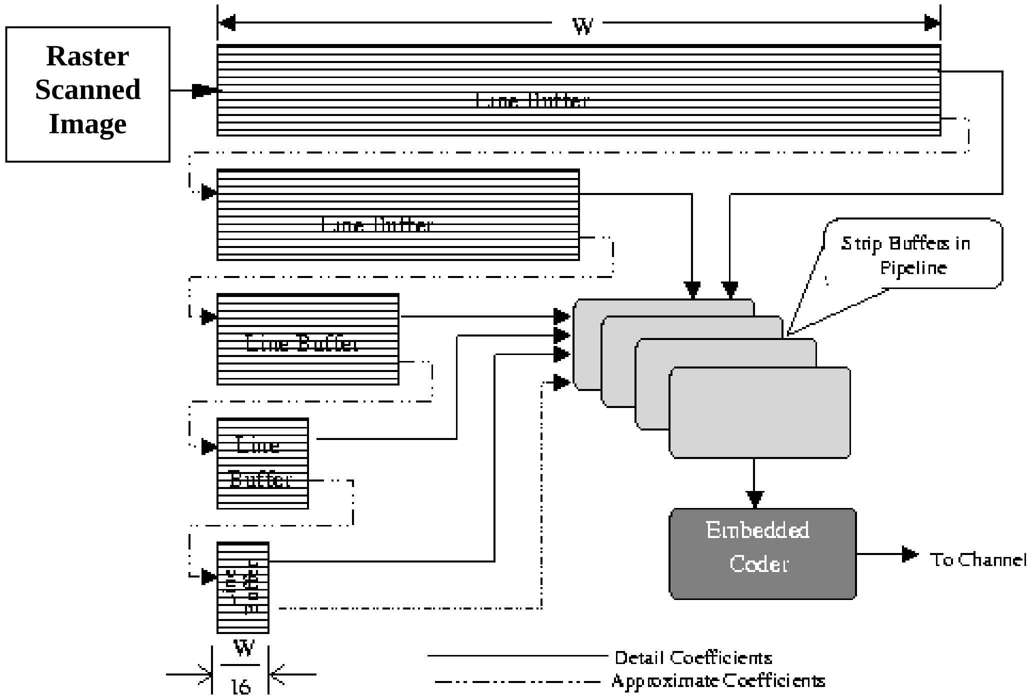

Figure 1 Strip Based Embedded Wavelet Coder

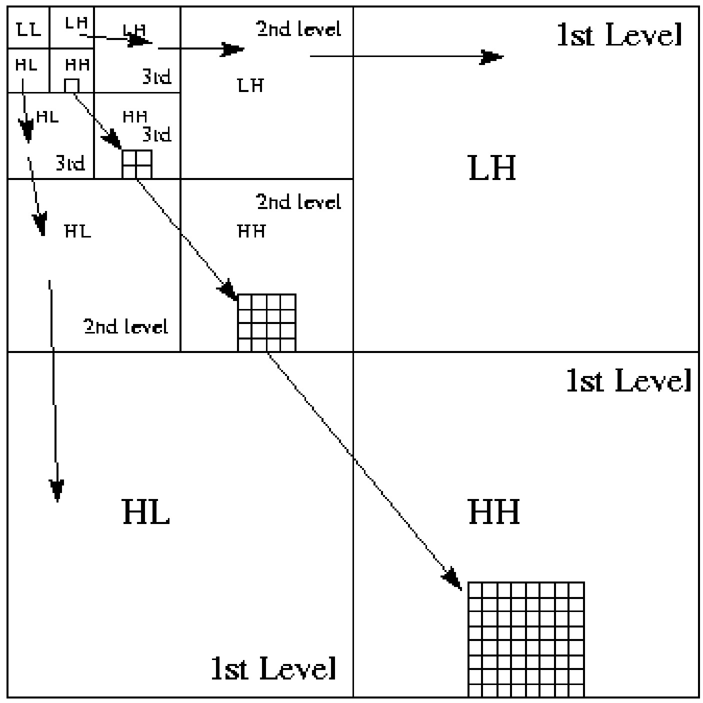

Figure 2 Zerotree Relation in Wavelet Decomposed Image

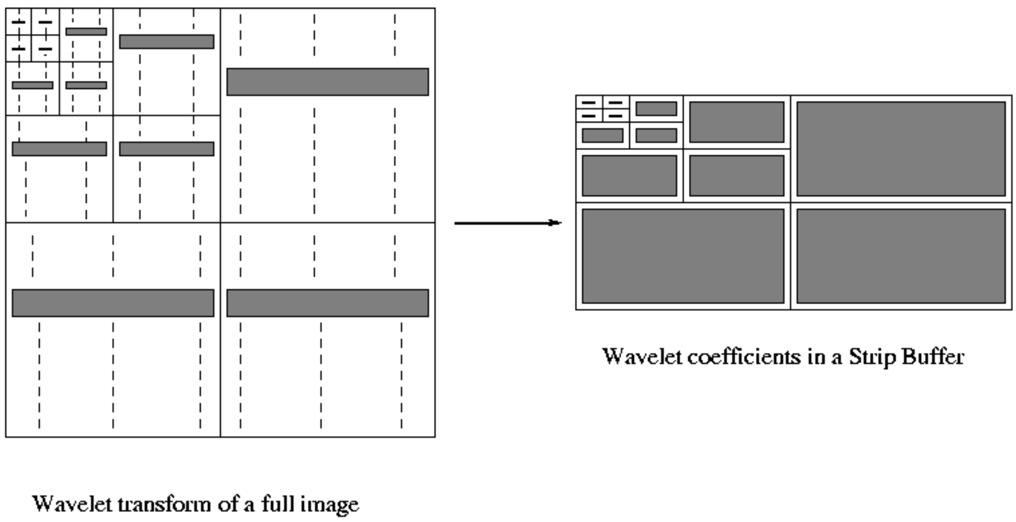

Figure 3 Strip Formation from Full Wavelet Image

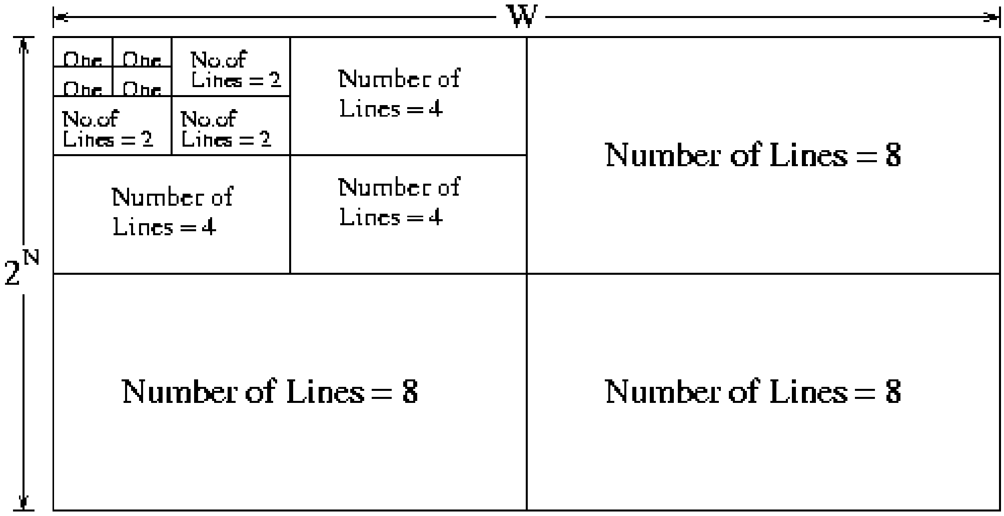

Figure 4 Fully Occupied Strip for 4 Levels

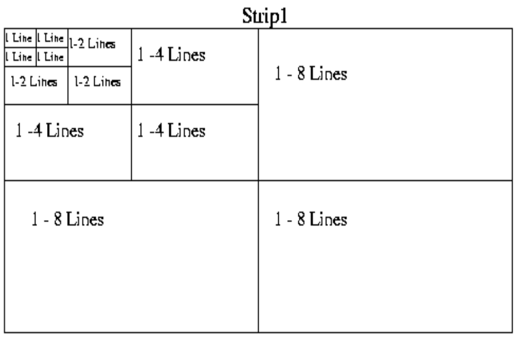

Figure 5a Memory Occupation after 67 Lines of Input

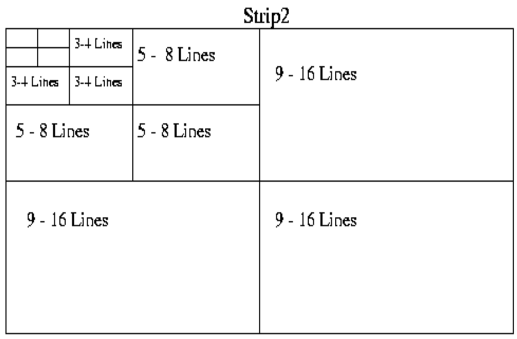

Figure 5b Memory Occupation after 67 Lines of Input

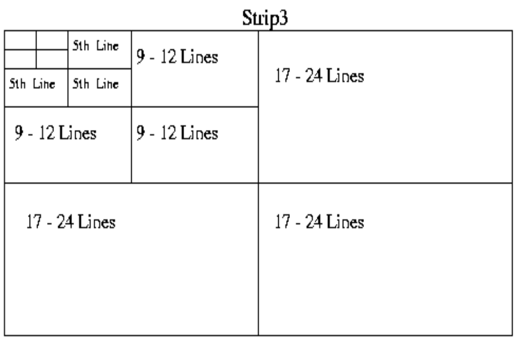

Figure 5c Memory Occupation after 67 Lines of Input

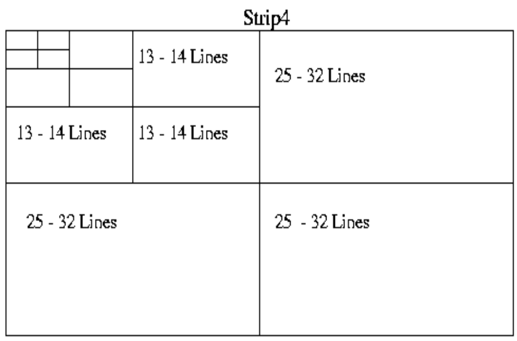

Figure 5d Memory Occupation after 67 Lines of Input

Figure 6a Original Image of 'bike'

Figure 6b Reconstructed Image at 0.25bpp