Design and Implementation of IP Core for Contourlet-Based Image Compression

Truong Quang Vinh
Ho Chi Minh City University of Technology (HCMUT)
tqvinh@hcmut.edu.vn

***Abstract*—This paper presents a Contourlet-based image compression algorithm and its hardware IP core design. The proposed algorithm is based on the Set Partitioning in Hierarchical Trees (SPIHT) coding method in Contourlet domain. Three new ideas for this algorithm are proposed: finding significant subbands by analyzing and comparing the distribution of coefficients in both Contourlet and Wavelet subbands; rearranging the order of these subbands based on their significant level; coding more significant subbands first by more bits to reduce compression loss. Thanks to these improvements, the proposed algorithm outperforms the other methods using SPIHT in Contourlet or Wavelet domain. In order to increase the processing time of the proposed algorithm, its hardware IP core is designed and verified on Cyclone IV FPGA. The architecture of the IP core supports Avalon bus interface and is able to be easily integrated into a system-on-chip (SoC). The complete verification of the IP core is described to demonstrate its correctness and applications in image compression.**

***Index Terms*— Contourlet Transform; Wavelet Transforms; Image Compression; SPIHT.**

I. Introduction

The popular image compression techniques usually utilize scalar quantization and vector quantization. However, both of these methods cannot produce embedded coding bit-stream. Some schemes which can generate an embedded bit-stream nowadays are Embedded Block Coding with Optimized Truncation (EBCOT) [1], Embedded Zerotrees of Wavelet transforms (EZW) [2], and Set Partitioning in Hierarchical Trees (SPIHT) [3]. Among those methods, SPIHT is one of the prominent algorithms for coding transformed coefficients efficiently with low computing complexity.

The SPIHT algorithm is originally implemented in the Wavelet domain. However, Wavelet coefficients represent a limited number of directions and cannot express smooth contours in natural images. Contourlet transform is known as a method which can form a multi-resolution and multi-directional representation of images by using the directional filter banks [4]. Consequently, Contourlet transform produces less distortion than Wavelet in some image regions such as contours and textures. Therefore, the combination of Contourlet transform and SPIHT coding algorithm is a potential approach that some researchers have applied for their proposed image compression methods.

Based on Contourlet transform, the author Haohao Song proposed an adjusted SPIHT algorithm for image coding [5]. He has analyzed the distribution of significant Contourlet coefficients in different subbands and decided to change the order of bandpass subbands for coding the most significant subband first. This helps to scan and code significant coefficients as much as possible when the bitstream is truncated at any point in order to satisfy the demand for embedded coding. In the research of Xi Zhi-Hong [6], Contourlet-based SPIHT coding method is also proposed for image compression. The author has optimized the lowpass subband coefficients by reducing the amplitude of these coefficients before coding and then adding reduced amplitude to reconstructed coefficients. This helps to avoid loss of energy. By another approach, Tan Peipei proposed a coding method using the adjusted SPIHT based on Contourlet transform [7]. The Contourlet transform is sparsified by the iterative thresholding method before coding.

In all the above approaches, the authors have tried to code the more significant Contourlet coefficients first. However, they did not notice how coefficients in Contourlet bandpass subbands are distributed at different levels. The numbers of significant coefficients at the same level are very different. The author of the work [5] also performed an analysis of the distribution of significant Contourlet coefficients in different subbands, and then he changed the order of subbands needed to be coded. By using more effective way, we rearrange the subbands in the virtual lowpass image [5] based on their significant levels, and then code more significant subbands by more number of bits. Consequently, these subbands could be reconstructed with less loss.

The hardware implementation for the SPIHT algorithm is also considered by many researchers. Thomas W. Fry et al. presented an implementation of the SPIHT algorithm using the folded Discrete Wavelet Transform (DWT) [8]. J. Yotheswar et al. also proposed FPGA implementation of modified SPIHT [9] using lifting-based DWT. Although many hardware designs of SPIHT have been proposed, the design of SPIHT algorithm in Contourlet domain is still an open problem due to the high complexity of Contourlet transform for hardware implementation. In the previous research [10], we have designed a full pipeline architecture for Contourlet transform. This hardware architecture will be applied for our SPIHT IP core to compress images in real-time.

In this paper, we propose a modified SPIHT method that rearranges the order of significant subbands before coding the coefficients. To do that, we analyze the distribution of significant coefficients at a different level. After rearranging the order of significant subbands, we code the more bit number for the more significant subbands. By that way, the loss of compression method can be reduced. Moreover, in this paper, we also propose a hardware design of the Contourlet-based SPIHT algorithm. The design is implemented as an IP core which is able to be integrated into a system on chip with Avalon bus architecture. The IP core is verified on Altera Cyclone IV FPGA to prove its correctness.

II. PROPOSED CONTOURLET-BASED IMAGE COMPRESSION

We propose the idea that using more bits to code more significant subbands to reduce the compression loss. In order to find how the Contourlet and Wavelet coefficients are significant in different levels, we analyze the distribution of significant coefficients in both Contourlet and Wavelet subbands with different levels. From that, we decide to apply the coding process on appropriate subbands.

A. *The Analysis of the Distribution of Significant Coefficients*

A number of significant coefficients of each subband are given by the equation:

$$S = \sum_x \sum_y \alpha(x, y) \tag{1}$$

where,

$$\alpha(x, y) = \begin{cases} 1 & p(x, y) \geq T \\ 0 & others \end{cases} \tag{2}$$

According to the result analyzed in [5], the suitably chosen threshold T is 32. $p(x,y)$ is the magnitude of the coefficient at (x,y) coordinates of each subband.

The number of significant coefficients in the subband decides the significance of that subband. It means that the more significant coefficients a subband contains, the more significant the subband is. The subband is the most significant one if it has the largest number of significant coefficients, and vice versa.

From the number of significant coefficients at each subband, we calculate the significant coefficient ratio R which is the ratio of a number of the most significant coefficients and the least significant coefficients at the same level.

The ratio R is given by the equation:

$$R = \frac{S_{the\,most\,significant\,subband}}{S_{the\,least\,significant\,subband}} \tag{3}$$

Table 1 shows a number of significant coefficients of Contourlet and Wavelet subbands of 8 standard test images (8 bpp, 512x512) at different scales. In there, significant coefficients of each subband are counted for its sum numbers.

Where, M_W and M_C are the numbers of significant coefficients of the most significant subband in Wavelet and Contourlet, respectively. L_W and L_C are the numbers of significant coefficients of the least significant subband in Wavelet and Contourlet, respectively. R_W and R_C are the ratios of a number of the most significant coefficients and the least significant coefficients at the same level in Wavelet and Contourlet, respectively.

From Table 1, we conclude that R is generally higher at the higher level in Wavelet and Contourlet subbands. It is noticed that at high levels, R_C is much greater than R_W. For example, at the 3rd level of Barbara image, R_W is 2.01, while W_C is 8.59. At 4th level, R_C is more hundreds of times than R_W. Moreover, the fact that R_C is high means that almost all significant Contourlet coefficients are contained in only some significant subbands, while other subbands have few significant coefficients or even no significant coefficients. Therefore, we decide to use Contourlet coefficients to encode significant subbands for image compression. To do that, we have to rearrange the order of each subband based on its significant level.

Table 1
Distribution of Significant Coefficients of Wavelet and Contourlet Subbands with Four Levels

Image	Level	Wavelet domain			Contourlet domain		
		M_W	L_W	R_W	M_C	L_C	R_C
Barbara	1	679	371	1.83	207	125	1.66
	2	1206	437	2.76	402	54	7.44
	3	1662	828	2.01	696	81	8.59
	4	2754	861	3.20	1780	0	1780/0
Goldhill	1	719	400	1.80	217	97	2.24
	2	1547	479	3.23	562	49	11.47
	3	1569	575	2.73	879	11	79.91
	4	1731	211	8.20	800	0	800/0
Mandrill	1	727	614	1.18	199	129	1.54
	2	2021	1676	1.21	603	285	2.12
	3	3474	2273	1.53	1301	354	3.68
	4	4711	2552	1.85	874	0	874/0
Lena	1	651	380	1.71	204	123	1.66
	2	1289	600	2.15	420	141	2.98
	3	1395	407	3.43	517	61	8.48
	4	1474	449	3.28	340	0	340/0
Baboon	1	705	559	1.26	198	122	1.62
	2	1856	1471	1.26	574	253	2.27
	3	3560	2146	1.66	1419	314	4.52
	4	5079	2892	1.76	3092	60	51.53
Camera-man	1	456	328	1.39	180	97	1.86
	2	1011	564	1.79	384	136	2.82
	3	1263	459	2.75	603	43	14.02
	4	1627	221	7.36	339	0	339/0
Peppers	1	670	403	1.66	218	124	1.76
	2	1066	422	2.53	427	78	5.47
	3	909	457	1.99	427	19	22.47
	4	1002	197	5.09	322	2	161
Pirate	1	729	558	1.31	207	121	1.71
	2	1496	807	1.85	525	137	3.83
	3	1694	904	1.87	712	56	12.71
	4	1958	630	3.11	535	4	133.75

B. *Proposed algorithm*

According to the analysis result of the distribution of significant coefficients, we propose the adjusted SPIHT algorithm based on the significant levels of Contourlet subbands. The algorithm can be described as the following steps:

1. Applying Contourlet transform to the image: The input image is transformed to Contourlet coefficients with subbands
2. Constructing virtual lowpass image: Virtual lowpass image [5] has the same size and a same number of pixels as the original lowpass image but is partitioned imaginarily (Figure 1). It makes the lowpass image has both size and direction suitable to join making Spatial Orientation Tree. By that way, the tree will be higher, and the compression performance will be better.
3. Rearranging the order of 16 subbands in all 4 levels based on significant levels of subbands: We count the number of significant coefficients in each subband that is given by formula (1). Then, we sort subbands by their significant number. The more significant subband is, the higher order in its pyramid level is.
4. Coding the more significant subbands first by more bits with the adjusted SPIHT algorithm: To code the more significant subbands first, we have to decide how many bits needed to code the more significant subbands. We assume that each couple of adjacent significant subbands

is different from N bits. In our experiment, N is set to be 200 in order to give the best performance from the bit-rate 0.15 bpp to the bit-rate 0.25 bpp. The coding method is applied by an adjusted SPIHT algorithm using Spatial Orientation Tree with both lowpass and bandpass images [5].

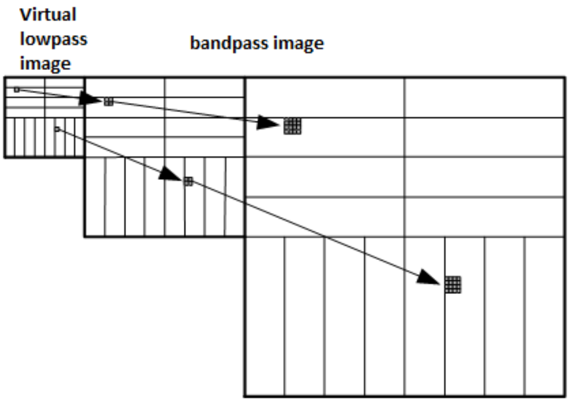


Figure 1: Structure of the virtual lowpass image and bandpass image

III. IP CORE DESIGN

The IP core of the proposed SPIHT image compression algorithm is designed with the following features:

- Perform image compression of image stream with user-defined frame size.
- Support Avalon-ST to input and output image data stream.
- Support Avalon-MM to interface with the NIOS processor.
- Verified on Altera Cyclone IV FPGA.

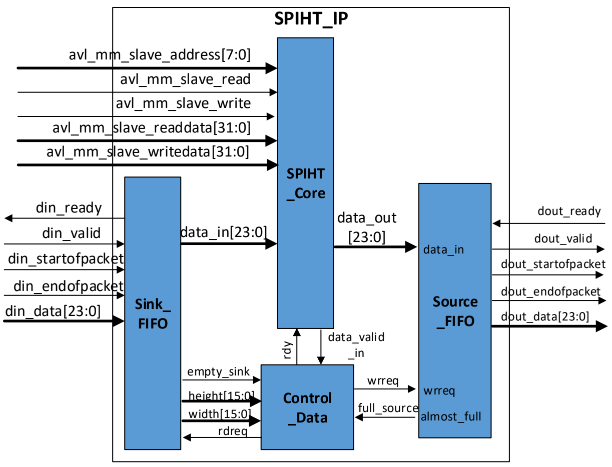


Figure 2: Block diagram of the SPIHT IP core

The block diagram of the SPIHT IP core is divided into 4 main blocks as shown in Figure 2.

- **Sink_FIFO:** support Avalon-ST sink to receive image data into a FIFO memory.
- **Control_Data:** control input/output data for other blocks.
- **SPIHT_Core:** process SPIHT algorithm for image compression and interface with the Nios2 processor.
- **Source_FIFO:** store the processed data and send to output via Avalon-ST source.

A. Sink FIFO

The Sink FIFO block is to receive input image data from Avalon-ST interface. The data are then stored in the FIFO memory. The Sink FIFO consists of two subblocks: Sink Control and FIFO memory (Figure 3). The Sink Control is to interface the Avalon-ST bus and extract the image data and the size data from the input stream.

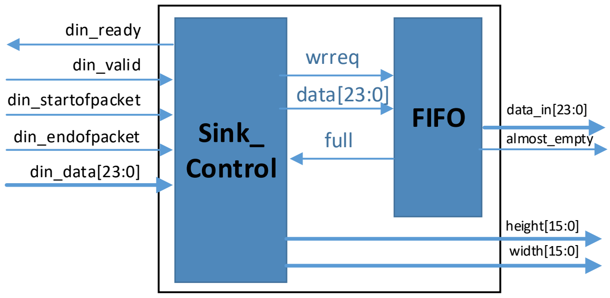


Figure 3: Block diagram of the Sink FIFO

B. Source FIFO

The function of the Source FIFO block is to store the result image data in the FIFO memory and transfer these data to the output through the Avalone-ST interface. The block diagram of Source FIFO is shown in Figure 4. The Source Control is to interface the Avalon-ST bus and package the image data and size data into the output stream.

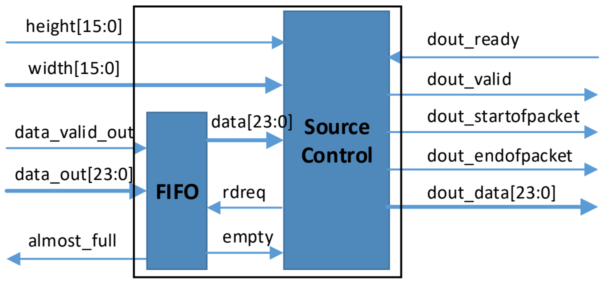


Figure 4: Block diagram of the Source FIFO

C. SPIHT core

The SPIHT core performs the proposed SPIHT algorithm for image compression (Figure 5). The format of input data and output data is YCbCr 4:4:4 in raster scan stream.

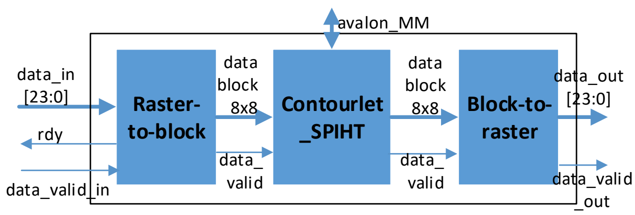


Figure 5: Block diagram of the SPIHT core

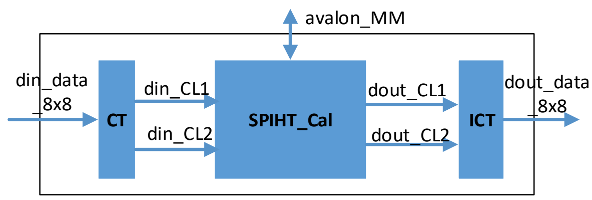


Figure 6: Block diagram of the Contourlet SPIHT block

The block diagram of Contourlet SPIHT subblock is shown in Figure 6. This subblock encodes each image data block by applying the Contourlet transform (CT) for input data, then performing image compression by the proposed SPIHT algorithm. The output encoded data are transferred to a microprocessor through Avalon-MM bus. For the image un-compression process, the encoded image data are entered from memory via Avalon-MM and are decoded by Contourlet_Cal block, and then converted back to image data by Inverse Contourlet Transform (ICT). The hardware architecture of CT and ICT blocks are inherited from our previous research project [10].

The Contourlet-Cal subblock preforms rearrangement of Contourlet coefficients as Spatial Orientation Tree and encodes subbands to binary codes. The rearrangement process requires memory for operations. Therefore, it needs a microprocessor to control the process through Avalon-MM bus.

D. *Control Data*

The control data block is used to control the input and output flows of the IP core (Figure 7). The input data are requested from Sink FIFO by the signal *rdreq*. After the data have been processed, the result data are written to the Source FIFO by the signal *wrreq*. The *empty_sink* and *full_source* signals are checked before Sink FIFO is read, and the Source FIFO is written to make sure both of them are ready. The signal *data_valid_out* is active when the output data are valid.

Figure 7: Block diagram of the Control Data

IV. EXPERIMENTAL RESULTS

A. *Algorithm Simulation Results*

The proposed modified SPIHT algorithm for image compression is simulated by Matlab software. For the testing dataset, we select eight standard 8-bit 512x512 images: Lena, Goldhill, Barbara, Mandrill, Baboon, Cameraman, Peppers, and Pirate. In all experiments, 9-7 filter for lowpass stage and PKVA filter for DFB stage are used. 4-scale decomposition and directional decomposition are performed for each scale with 16 directions. The output bitstream is not further compressed with any arithmetic encoders. We compare the proposed method with two reference methods: Contourlet-based adjusted SPIHT [5] and Wavelet-based SPIHT [4].

Table 2 shows the test results of eight images. The proposed method generally has higher PSNR than other methods, especially at low bit rates. For image Cameraman, the performance is the best, 0.15 bpp higher in 0.655 dB comparing with Contourlet-based adjusted SPIHT, and 3.051 dB comparing with Wavelet-based SPIHT.

However, PSNR of the proposed method decreases when the bit rate increases. At a bit rate of 0.25 bpp, the performance of the proposed method is generally slightly higher than Contourlet-based adjusted SPIHT, but it is lower than Wavelet-based SPIHT.

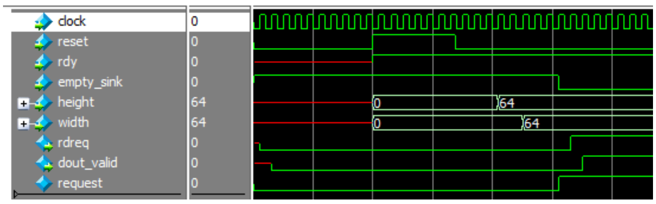
Table 2
The PSNR of Compression Images on Three Coding Algorithms

Image	Bit-rate (bpp)	Proposed (dB)	Method [5] (dB)	Method [4] (dB)
Goldhill	0.15	27.206	26.606	24.834
	0.20	27.712	27.458	26.509
	0.25	27.824	27.992	27.926
Barbara	0.15	23.422	23.061	22.620
	0.20	23.829	23.583	23.743
	0.25	24.083	24.086	24.791
Mandrill	0.15	22.382	22.063	21.671
	0.20	23.282	23.113	22.997
	0.25	24.230	24.002	23.992
Lena	0.15	28.235	27.824	25.852
	0.20	28.904	28.932	28.392
	0.25	29.700	29.392	30.221
Peppers	0.15	27.426	26.938	25.405
	0.20	28.399	28.161	27.876
	0.25	29.265	28.956	30.064
Pirate	0.15	25.631	25.350	24.184
	0.20	26.441	26.192	25.936
	0.25	26.982	26.790	27.390
Camera man	0.15	28.585	27.930	25.534
	0.20	30.144	30.622	28.327
	0.25	30.651	30.116	30.997
Baboon	0.15	20.599	20.647	19.993
	0.20	20.743	21.141	20.948
	0.25	21.537	21.202	21.553

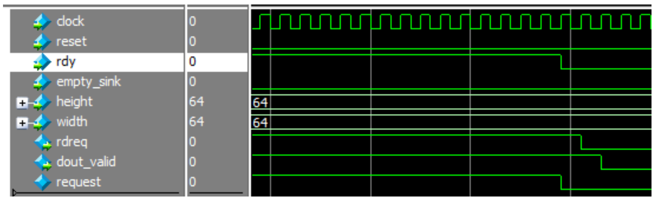
B. *Result of SPIHT IP Core Simulation*

The SPIHT IP core simulation is performed by ModeSim software. The subblocks of the SPIHT are simulated firstly to check the correctness of their operational functionality. Finally, the SPIHT IP Core is simulated with test image data. The input and output data of the simulation result is compared to the simulation result in Matlab.

Figure 8 shows the simulation waveforms of the block Control Data when the signal *request_data* is active and inactive. Figure 9 shows the simulation waveform of the block SPIHT core when the *din_valid* signal is active, and the input data arrive.



(a) When request_data = 1



(b) When request_data = 0

Figure 8: The simulation waveforms of the block Data Control

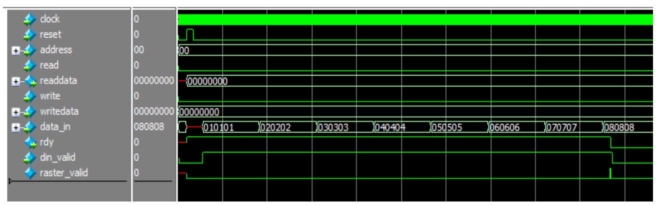


Figure 9: The simulation waveforms of the block SPIHT core

C. *Result of SPIHT IP Core Verification on Cyclone IV FPGA*

The SPIHT IP core is integrated into the SOPC (System on Programmable Chip) using Avalon bus architecture by using Quartus II software. The SOPC architecture is shown in Figure 10.

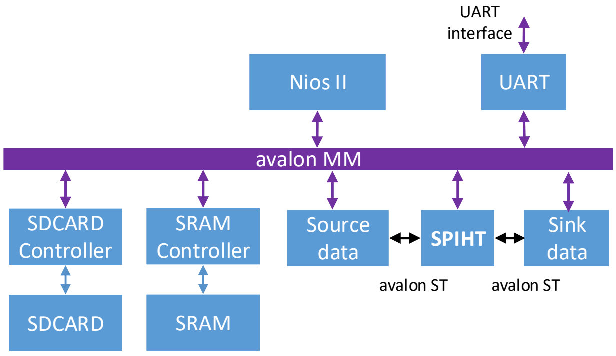


Figure 10: The architecture of the SOPC system

In this architecture, we utilize some IP cores from the Altera library including SD Card Controller, SRAM controller, and UART controller. The SD Card Controller is used to import and export the input and output image files from an SD card. The UART Controller is used to send debugging information to a computer for users.

In the verification process, all operations of the SOPC are controlled by the NIOS processor. An image file in SD card is read by the SD card controller and stored in SRAM. Then, these data are sent to the Source Data. The SPIHT IP core receives input data from Source Data, process them, and send to Sink Data. The processed data are then stored in SD Card. All operation states and parameters are sent to PC via UART for debugging. The experiment has been done on the DE2-115 development board with Altera Cyclone IV FPGA as shown in Figure 11.

Figure 11: Verification process on DE2-115 development board

Table 3 shows the synthesis result of the SPIHT IP core and comparison with other designs. The timing analysis tool of Quartus II software shows that the maximum frequency is 120MHz. In our experiment, the IP core was tested successfully at a clock rate of 50 MHz. In the comparison table, the hardware designs of Wavelet-based SPIHT algorithm [8],[9] provide high throughput thanks to the advanced architectures of DWT. The work [8] used a lifting scheme for DWT, while the work [9] utilized 4-row parallel architecture for DWT. Due to the complexity of Contourlet transform, the throughput of our IP core is lower than the reference designs [8] and [9]. However, Contourlet-based SPIHT is still a potential approach for researchers in the future because of the advantages of Contourlet transform in the representation of contours and textures.

Table 3
The synthesis result of the IP Core on FPGA

Specification	[8]	[9]	Proposed
Algorithm	Wavelet-based SPIHT	Wavelet-based SPIHT	Contourlet-based SPIHT
FPGA	Xilinx Virtex 2000E	Xilinx Virtex4 xc4vlx25	Altera Cyclone IV EP4CE115
Clock rate	56 MHz	35 MHz	50 MHz
Logic element	N/A	7,021	5,285
Memory bits	N/A	N/A	82,432
Throughput	224 Mpps	280 Mpps	64 Mpps

Figure 12 shows the reconstructed image which is compressed by the SPIHT IP core. The contour and texture regions of the resulting image are reconstructed at the high subjective quality thanks to Contourlet coefficients.

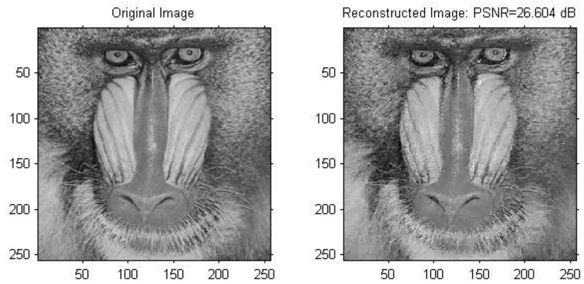


Figure 12: The reconstructed image which is compressed by SPIHT IP core

V. CONCLUSION

This paper has presented an image compression algorithm using the adjusted SPIHT method in Contourlet domain and its hardware implementation. The basic strength of the SPIHT algorithm is to encode more significant coefficients first by more bits to reduce compression loss. Besides, SPIHT algorithm is improved to encode Contourlet coefficients effectively by rearranging Spatial Orientation Tree and number of bits to represent each subband. Experimental results show that the compression performance of the proposed coding algorithm is better than Wavelet-based SPIHT, especially at low bit rates. The hardware design of the proposed SPIHT algorithm which was verified on Cyclone IV FPGA has been proved that the algorithm is applicable for real-time image compression.

REFERENCES

[1] Taubman, D., “High Performance Scalable Image Compression with EBCOT”, *IEEE Trans. on Image Processing*, vol.9, 2000, pp.1158-1170.

[2] Shapiro, J.M., “ Embedded Image Coding Using Zerotrees of Wavelet Coefficient” *IEEE Trans. on Signal Processing*, vol. 41, 1993, pp.3445-3462.

[3] Amir Said, William A.Pearlman, “A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees”, *IEEE Transactions on Circuits and System for Video Technology*, vol. 6, 1996, pp.243-250.

[4] Minh N. Do, and Martin Vetterli, “The Contourlet Transform: An Efficient Directional Multiresolution Image Representation”, *IEEE Transactions on Image Processing*, 14.12, 2005, pp. 2091-2106.

[5] Haohao Song, Songyu Yu, Li Song, Hongkai Xiong, “Contourlet Image Coding Based on Adjusted SPIHT”, *Advances in Multimedia Information Processing – PCM, Lecture Notes in Computer Science*, vol. 3767, 2005, pp. 629-640.

[6] Xi Zhi-hong, Xiao Yi-han, “An Image Compression Scheme Adopted for Contourlet transform”, *2nd International Congress on Image and Signal processing*, 2009, CISP ’09., 2009.

[7] Tan Peipei, Wei Zhao, "Image Coding Based on Sparsified Contourlet and Adjusted SPIHT", *IEEE 11th International Conference on Signal Processing (ICSP)*, vol. 2, 2012.

[8] Thomas W. Fry and Scott A. Hauck, "SPIHT Image Compression on FPGAs", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, 2005, pp.1138-1147.

[9] J. Jyotheswar, Sudipta Mahapatra, "Efficient FPGA implementation of DWT and modified SPIHT for lossless image compression", vol. 53, 2007, pp.369-378.

[10] T.Q. Vinh, L.Q. B. Tri, and N. N. Tai, "A Real-Time Video Denosing Implementation in FPGA Using Contourlet Transform", *The International Conference on Computing, Management and Telecommunications* (ComManTel), pp.203-207, 2013.