Real-Time Image and Video Processing 2012, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE Vol. 8437, 843704 - © 2012 SPIE - doi: 10.1117/12.924327

A contourlet transform based algorithm for real-time video encoding

Stamos Katsigiannis[a], Georgios Papaioannou[b], Dimitris Maroulis[a]
[a]Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens, Panepistimioupoli, Ilisia, 15703, Athens, Greece;
[b]Dept. of Informatics, Athens University of Economics and Business, 76, Patission Str., 10434, Athens, Greece

ABSTRACT

In recent years, real-time video communication over the internet has been widely utilized for applications like video conferencing. Streaming live video over heterogeneous IP networks, including wireless networks, requires video coding algorithms that can support various levels of quality in order to adapt to the network end-to-end bandwidth and transmitter/receiver resources. In this work, a scalable video coding and compression algorithm based on the Contourlet Transform is proposed. The algorithm allows for multiple levels of detail, without re-encoding the video frames, by just dropping the encoded information referring to higher resolution than needed. Compression is achieved by means of lossy and lossless methods, as well as variable bit rate encoding schemes. Furthermore, due to the transformation utilized, it does not suffer from blocking artifacts that occur with many widely adopted compression algorithms. Another highly advantageous characteristic of the algorithm is the suppression of noise induced by low-quality sensors usually encountered in web-cameras, due to the manipulation of the transform coefficients at the compression stage. The proposed algorithm is designed to introduce minimal coding delay, thus achieving real-time performance. Performance is enhanced by utilizing the vast computational capabilities of modern GPUs, providing satisfactory encoding and decoding times at relatively low cost. These characteristics make this method suitable for applications like video-conferencing that demand real-time performance, along with the highest visual quality possible for each user. Through the presented performance and quality evaluation of the algorithm, experimental results show that the proposed algorithm achieves better or comparable visual quality relative to other compression and encoding methods tested, while maintaining a satisfactory compression ratio. Especially at low bitrates, it provides more human-eye friendly images compared to algorithms utilizing block-based coding, like the MPEG family, as it introduces fuzziness and blurring instead of artificial block artifacts.

Keywords: Contourlet transform, real-time video encoding, GPU computing, denoising, video-conferencing, surveillance video

1. INTRODUCTION

In recent years, the use of broadband internet connections made possible the transmission over the network of high quality multimedia content. Real-time video communication over the internet has been widely utilized for applications like video conferencing, video surveillance, etc. Streaming live video over heterogeneous IP networks, including wireless networks, requires highly efficient video coding algorithms that can achieve good compression while maintaining satisfying visual quality and real-time performance. Most state of the art video compression techniques like the H.264, DivX/XVid, MPEG2 have computational complexities that require dedicated hardware to achieve real-time performance. Another drawback of these methods is the lack of support for multiple quality levels on the same video stream. Solutions proposed based on these algorithms constitute extensions that were not taken into consideration when the algorithms were designed. Furthermore, in order to achieve optimal compression and quality efficiency, these methods utilize statistical and structural analysis of the whole video content which is not available in cases of live content creation and demand a lot of computational time and resources.

The aim of this work is the design and development of a novel algorithm for high-quality real-time video encoding, for content obtained from low resolution sources like web cameras, surveillance cameras, etc. The desired characteristics of such an algorithm would be: 1) low computational complexity, 2) real-time encoding and decoding capabilities, 3) the ability to improve the visual quality of content obtained from low quality visual sensors, 4) the support of various levels of quality in order to adapt to the network end-to-end bandwidth and transmitter/receiver resources and 5) the resistance to packet losses that might occur during transmission over a network. The selection of a suitable image representation method is critical for the efficiency of a video compression algorithm. Texture representation methods utilizing the Fourier transform, the Discrete Cosine Transform, the wavelet transform as well as other frequency domain methods have been extensively used for video and image encoding. Some limitations of these methods have been partially addressed by the Contourlet Transform [1] which can efficiently approximate a smooth contour at multiple resolutions. Additionally, it offers multiscale and directional decomposition, providing anisotropy and directionality, features missing from traditional transforms like the Discrete Wavelet Transform [1]. The Contourlet Transform has been successfully used in a variety of texture analysis applications, including SAR [2], medical and natural image classification [3], image denoising [4], despeckling of images, image compression, etc. Combined with the computational power offered by modern graphics processing units (GPUs), the contourlet transform can provide an image representation method with advantageous characteristics while maintaining real time capabilities. Considering these facts, the contourlet transform was selected as the core element of the proposed video encoding algorithm.

The rest of this paper is organized in four sections. Section 2 introduces the methods and knowledge needed for better understanding of this work, while section 3 presents the proposed algorithm, including a detailed explanation of its components. An experimental study for the evaluation of the algorithm is presented on section 4, whereas conclusions and future perspectives are presented in section 5.

2. BACKGROUND

2.1 The Contourlet Transform

The Contourlet Transform (CT) is a directional multiresolution image representation scheme proposed by Do and Vetterli, which is effective in representing smooth contours in different directions of an image, thus providing directionality and anisotropy [1]. The method utilizes a double filter bank (Figure 1) in which, first the Laplacian Pyramid (LP) [5] detects the point discontinuities of the image and then the Directional Filter Bank (DFB) [6] links point discontinuities into linear structures. The LP provides the means to obtain multiscale decomposition. In each decomposition level it creates a downsampled lowpass version of the original image and a more detailed image with the supplementary high frequencies containing the point discontinuities. This scheme can be iterated continuously in the lowpass image and is restricted only by the size of the original image due to the downsampling. The DFB is a 2D directional filter bank that can achieve perfect reconstruction. The simplified DFB used for the contourlet transform consists of two stages, leading to 2^l subbands with wedge-shaped frequency partitioning [7]. The first stage is a two-channel quincunx filter bank [8] with fan filters that divides the 2D spectrum into vertical and horizontal directions. The second stage is a shearing operator that just reorders the samples. By adding a shearing operator and its inverse before and after a two-channel filter bank, a different directional frequency partition is obtained (diagonal directions), while maintaining the ability to perfectly reconstruct the original image.

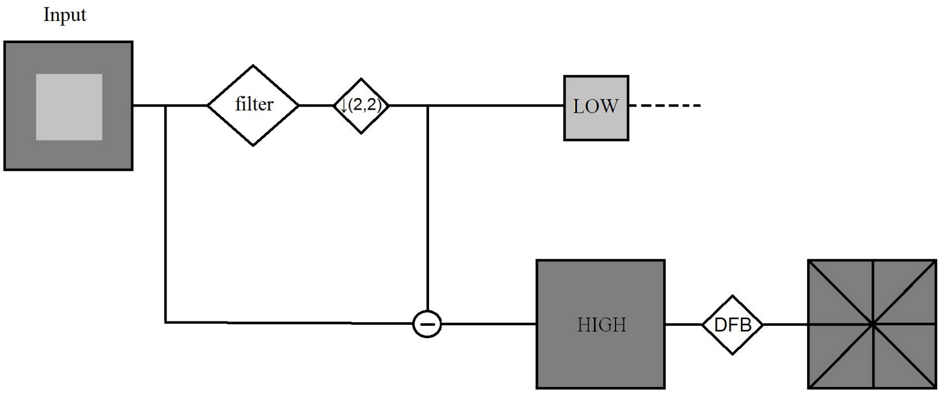


Figure 1. The Contourlet Filter Bank.

By combining the LP and the DFB, a double filter bank named Pyramidal Directional Filter Bank (PDFB) is obtained. Bandpass images from the LP decomposition are fed into a DFB in order to capture the directional information. This scheme can be repeated on the coarser image levels, restricted only by the size of the original image. The combined result is the contourlet filter bank. The contourlet coefficients have a similarity with wavelet coefficients since most of them are almost zero and only few of them, located near the edge of the objects, have large magnitudes [9]. In this work, the Cohen and Daubechies 9-7 filters [10] have been utilized for the Laplacian Pyramid. For the Directional Filter Bank, these filters are mapped into their corresponding 2D filters using the McClellan transform as proposed by Do and Vetterli in [1]. The creation of optimal filters for the contourlet filter bank remains an open research topic.

2.2 General purpose GPU computing

The most computationally intensive part of the contourlet transform is the calculation of all the 2D convolutions needed for complete decomposition or reconstruction. Classic CPU implementations based on the 2D convolution definition are not suitable for real-time applications since their computational complexity is a major drawback for performance. Utilizing the DFT or even FFT for better performance provides significantly faster implementations but still fails to achieve satisfactory real-time performance, especially in mobile platforms such as laptops and tablet PCs. In order to fully exploit the benefits of the FFT for the calculation of 2D convolution, an architecture supporting parallel computations can be utilized. Apart from the CPU, modern personal computers are commonly equipped with powerful graphics cards, which, in this particular case, are underutilized. This “dormant” computational power can be harnessed for accelerating intensive computations that can be computed in parallel. General purpose computing on graphics processing units (GPGPU) is the set of techniques that use a GPU, which is primarily specialized in handling computations for the display of computer graphics, to perform computations in applications traditionally handled by the CPU.

2.3 The YCoCg color space

It is well established in literature that the human visual system is significantly more sensitive to variations of luminance compared to variations of chrominance. Encoding the luminance components of an image with more accuracy than the chrominance components provides an easy to implement low complexity compression scheme while maintaining satisfactory visual quality. Many widely used image and video compression algorithms take advantage of this fact to achieve increased efficiency.

First introduced in H.264 compression, the RGB to YCoCg transform decomposes a color image into luminance and chrominance components and has been shown to exhibit better decorrelation properties than YCbCr and similar transforms [11]. The transform is calculated by the following equations:

$$Y = R/4 + G/2 + B/4 \qquad (1) \qquad\qquad R = Y + Co - Cg \qquad (4)$$

$$Co = R/2 - B/2 \qquad (2) \qquad\qquad G = Y + Cg \qquad (5)$$

$$Cg = -R/4 + G/2 - B/4 \qquad (3) \qquad\qquad B = Y - Co - Cg \qquad (6)$$

In order for the reverse transform to be perfect and to avoid rounding errors, the Co and Cg components should be stored with one additional bit of precision compared to the RGB components. Experiments using the Kodak image suite showed that using the same precision for the YCoCg and RGB data when transforming to YCoCg and back result in an average PSNR of 52.12dB. This loss of quality cannot be perceived by the human visual system making it insignificant for our application. Nevertheless, it indicates the highest quality possible when used for image compression.

Real-Time Image and Video Processing 2012, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE Vol. 8437, 843704 - © 2012 SPIE - doi: 10.1117/12.924327

3. METHOD OVERVIEW

Figure 3 depicts the outline of the algorithm. Raw input frames are considered to be in the RGB format. The first step of the algorithm is the transform from the RGB color space to YCoCg for further manipulation of the luminance and chrominance channels. Chrominance channels are then subsampled by a user-defined factor *N*, while the luminance channel is decomposed using the contourlet transform. Then, contourlet coefficients of the luminance channel are dropped, retaining only a user-defined amount of the most significant ones, while the precision allocated for storing the contourlet coefficients is reduced. Figure 2 shows an example of a decomposed luminance channel, containing three scales, each decomposed into four directional subbands. All computations up to this stage are performed on the GPU, avoiding needless memory transfers from the main memory to the GPU memory and vice versa. After manipulating the contourlet coefficients of the luminance channel, the directional subbands are encoded using a run length encoding scheme that encodes only zero valued elements. The large sequences of zero valued contourlet coefficients make this run length encoding scheme suitable for their encoding.

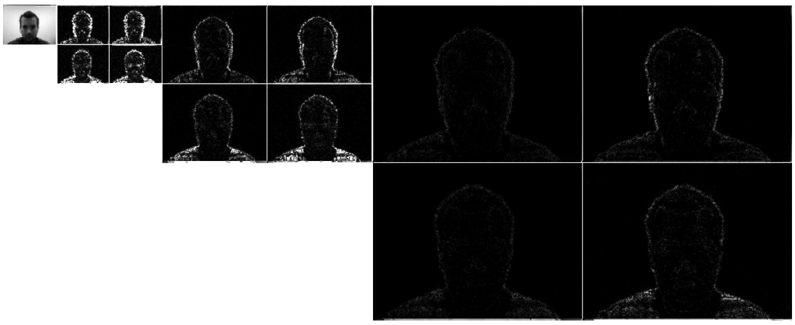


Figure 2. Example of CT decomposition of the luminance channel. Three levels of decomposition with the Laplacian Pyramid were applied, each then decomposed into four directional subbands using the Directional Filter Bank.

The algorithm divides the video frames into two categories; key frames and internal frames. Key frames are frames that are encoded using the steps described in the previous paragraph. The frames between two key frames are called internal frames and their number is user defined. When a frame is identified as an internal frame, at the step before the run length encoding, all its components are calculated as the difference between the respective components of the frame and the previous key frame. This step is processed on the GPU while all the remaining steps of the algorithm are performed on the CPU.

Then, run length encoding is applied to the chromatic channels, the low frequency contourlet component of the luminance channel, as well as the directional subbands of the luminance channel. Consecutive frames tend to have small variations from one another, with many regions similar to each other. Exploiting this fact, the calculation of the difference between a frame and the key frame provides components with large sequences of zero values making the run length encoding more efficient. Especially in the case of video-conferencing or surveillance video, the background tends to be static, with slight or no variations at all. When the key and internal frame scheme described above is utilized, the occurrence of static background leads to many parts of the consecutive frames to be identical. Calculating the difference of each frame from its respective key frame provides large sequences of zero values leading to improved compression through the run length encoding stage. Experiments showed that the optimal compression is achieved for a relatively small interval between key frames, in the region of 5-7 internal frames. This fact provides small groups of pictures (GOP) that depend to a key frame, making the algorithm more resistant to packet loses when transmitting over a network. Also, if a scene change occurs, the characteristics of consecutive frames differ drastically and the compression achieved for the internal frames until the next key frame is similar to that of a key frame. Small intervals between key frames reduce the number of non optimally encoded frames. Nevertheless, in cases like surveillance video where the video is expected to be mostly static, a larger interval between key frames will provide considerably better compression.

The last stage of the algorithm consists of the selection of the optimal precision for each video component. The user can select between lossless or lossy change of precision, directly affecting the output's visual quality.

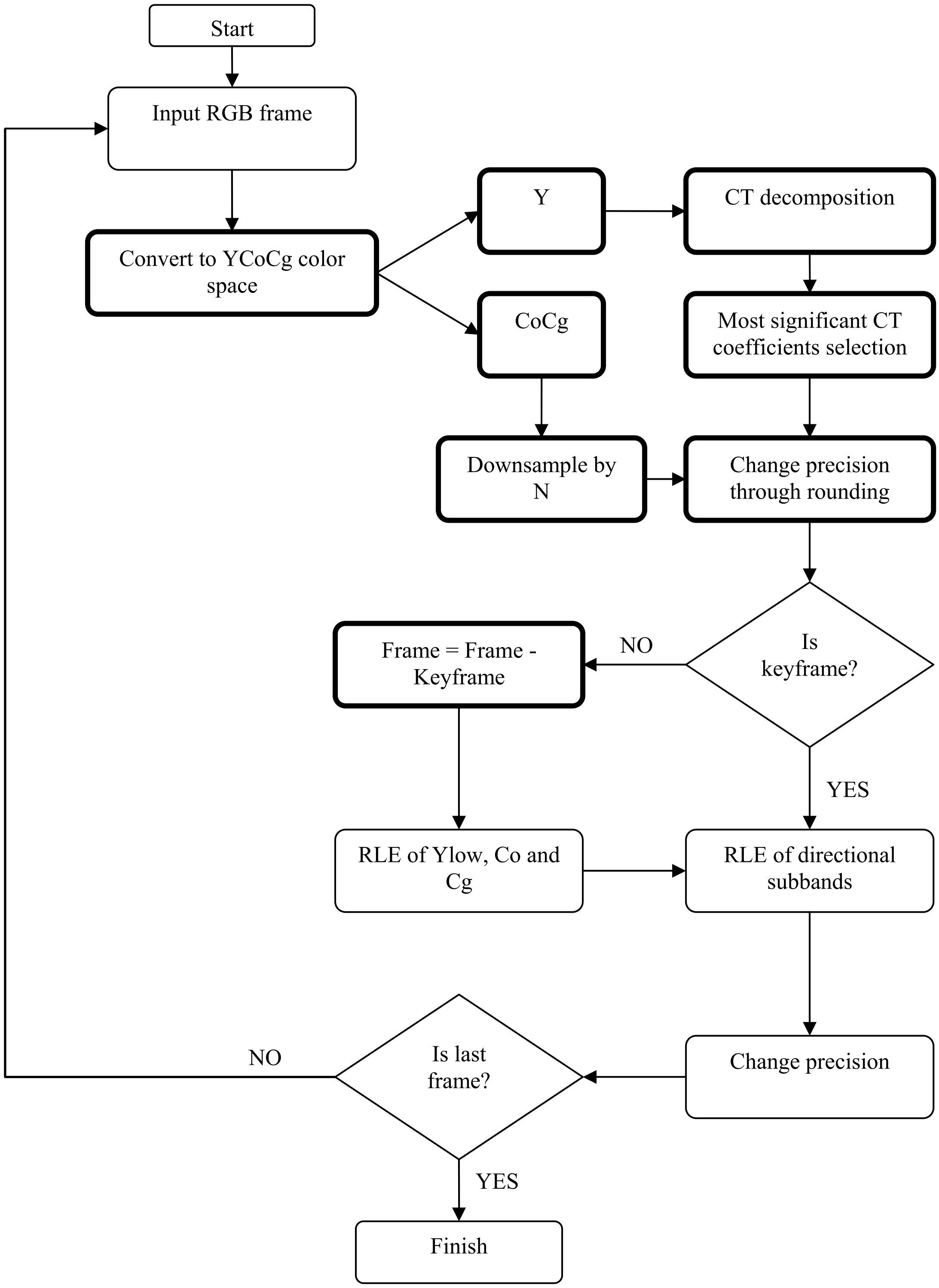


Figure 3. Block diagram of the algorithm. Highlighted blocks refer to calculations performed on the GPU, while the other blocks refer to calculations performed on the CPU.

Real-Time Image and Video Processing 2012, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE Vol. 8437, 843704 - © 2012 SPIE - doi: 10.1117/12.924327

3.1 Chroma subsampling

As mentioned in the introduction, the human visual system is significantly more sensitive to variations of luminance compared to variations of chrominance. Exploiting this fact, the chrominance channels Co and Cg are subsampled by a user-defined factor *N* that directly affects the output's visual quality, as well as the compression achieved. As a result, the chrominance channels are stored in lower resolution, thus providing compression. In order to reconstruct the chrominance channels at the decoding stage, the missing chrominance values are replaced with the nearest available subsampled chrominance values. This approach is simple and naïve compared to other methods like bilinear interpolation but has been selected due to the significantly smaller number of memory fetches and minimal computation cost, factors critical in real time applications. Depending on the subsampling factor, utilizing the nearest neighbor reconstruction approach introduces artifacts in the form of mosaic patterns in regions with strong chrominance transitions. Nevertheless, the receiver can choose to use the bilinear interpolation approach, given adequate computational resources. Figure 4 shows an example of chroma subsampling of the Co and Cg channels by various factors, using the nearest neighbor and the bilinear interpolation approach for reconstruction. For presentation purposes, only a small part of the "baboon" image used is shown.

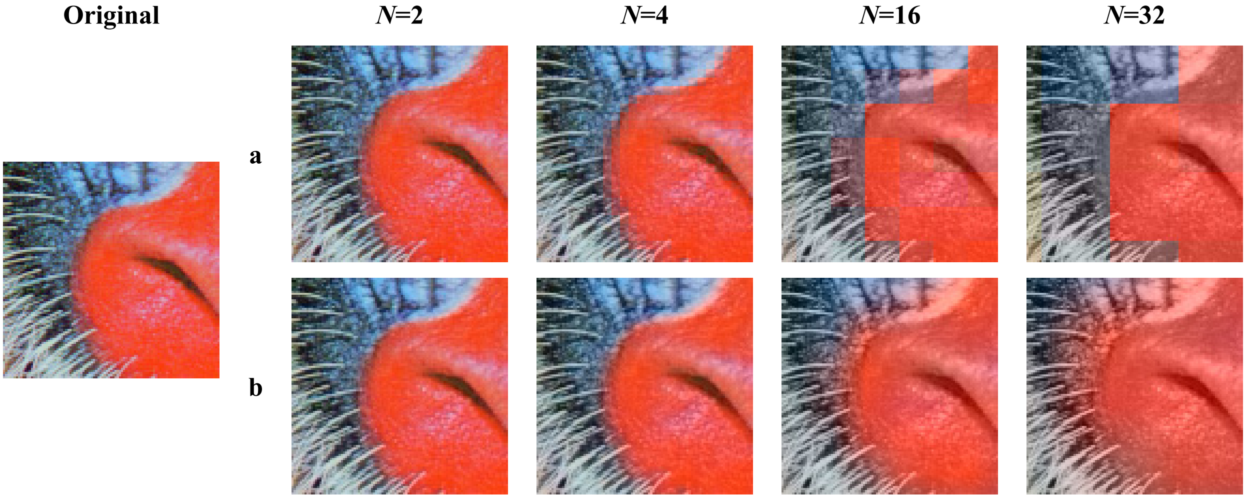


Figure 4. Example of chroma subsampling by factor *N* of the Co and Cg channels of the "baboon" image. Row (a) depicts images reconstructed using the nearest neighbor method, while (b) those reconstructed using bilinear interpolation.

As shown on Figure 4, subsampling by factor 2 or 4 does not affect drastically the visual quality. Further subsampling leads to visible artifacts and as a result, a tradeoff between quality and compression has to be made.

3.2 Contourlet Transform decomposition of luminance channel and quality selection

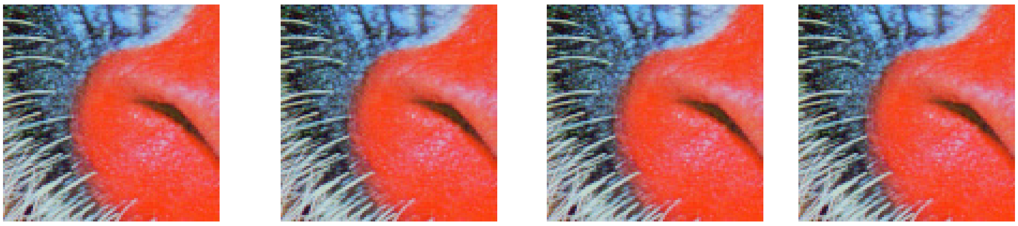
While the chrominance channels are subsampled, the luminance channel is decomposed using the contourlet transform. The levels of decomposition, as well as the filters used are user-defined and directly affect the quality of the output. Decomposition at multiple scales offers better compression while providing scalability. Multiple resolutions inside the same video stream are supported. This characteristic is desired for video coding algorithms in order to adapt to the network end-to-end bandwidth and transmitter/receiver resources. The level of detail for each receiver can be adjusted without re-encoding the video frames, by just dropping the encoded information referring to higher resolution than needed.

After decomposing the luminance channel with the contourlet transform, contourlet coefficients from the directional subbands are dropped by keeping only the most significant coefficients, in order to achieve compression. The amount of coefficients dropped is user-defined and drastically affects the output's visual quality as well as the compression ratio. A common method for selecting the most significant contourlet coefficients is to keep the *M* most significant coefficients while reducing all the others to zero [1]. To provide a more generalized and quantitative parameter for the algorithm, the *M* percentage of the most significant coefficients is retained, *M* being a user-specified quality setting. This procedure leads to a large number of zero-valued sequences inside the elements of the directional subbands, a fact exploited by using run length encoding (as explained in detail at §3.4) in order to achieve even higher compression.

Another advantage of keeping only the most significant contourlet coefficients is the suppression of noise induced by low-quality sensors usually encountered in web-cameras. Random noise is not likely to generate significant contourlet coefficients [1]. As a result, the application of an approach based on keeping the most significant contourlet coefficients is expected to provide enhanced visual quality. This characteristic is highly desirable since no external manipulation of the video stream is required in order to reduce the level of noise of a noisy video stream.

To avoid rounding errors and precision loss at the contourlet transform decomposition stage, single precision floating point elements that occupy 32 bit of memory are used. Experiments with the precision allocated for the contourlet coefficients showed that the contourlet transform exhibits resistance to quality loss due to loss of its coefficients precision. Exploiting this fact, the precision of the contourlet coefficients is reduced by means of rounding to a specific decimal point. Maintaining only one decimal or more for the contourlet coefficients does not have any effect on the visual quality. Rounding to the integer provides a PSNR of more than 60dB when only the directional subbands' coefficients are rounded, while also rounding the low pass content provides a PSNR of more than 55dB. In both cases, the loss of quality is considered as insignificant due to the fact that it cannot be perceived by the human visual system.

Figure 5 shows an example of precision reduction through rounding of the contourlet coefficients of the "baboon" image. For presentation purposes, only the part of the "baboon" image used in Figure 4 is shown. The image is first transformed into the YCoCg color space. Then the luminance channel is decomposed using the contourlet transform and the contourlet coefficients are rounded. No alteration is done to the chrominance channels. After the manipulation of the contourlet coefficients, the luminance channel is reconstructed and the image is transformed back into the RGB color space



1 decimal, only HIGH PSNR: n/a

1 decimal, HIGH, LOW PSNR: n/a

Integer, only HIGH PSNR: 62.22dB

Integer, HIGH, LOW PSNR: 55.48dB

Figure 5. Example of precision reduction through rounding of the contourlet coefficients of the "baboon" image. LOW refers to the lowpass component obtained through the contourlet transform, while HIGH refers to the directional subbands.

3.3 GPU-based contourlet transform calculations

For the GPU implementation of the contourlet transform, the NVIDIA CUDA architecture has been selected due to the extensive capabilities and specialized API it offers. At first, the image and the filters are transferred from the main memory to the GPU dedicated memory in order to reduce the unnecessary transfers to and from the main memory that introduce delay to the computations. Then, the contourlet transform of the image is calculated at the GPU. The 2D convolutions required are calculated by means of the FFT, as shown on Figure 6. After finishing the calculations, the output is transferred back to the main memory. Taking into consideration the fact that this implementation will be used for video encoding, the filters are loaded once at the GPU memory since they will not change from frame to frame. For performance testing purposes, various implementations of the contourlet transform have been developed, both for the CPU and the GPU. Implementations on the CPU were based on the FFT and the 2D convolution definition. Two types of implementations were developed on the GPU using the CUDA architecture. One based on the FFT as mentioned above and one based on the 2D convolution definition. Except for the basic GPU implementation using the definition, other implementations utilizing memory management schemes in order to support larger frames when the GPU memory is not sufficient were developed based on the 2D convolution definition [12]. The GPU implementation based on the FFT (Figure 6) outperforms all the aforementioned implementations.

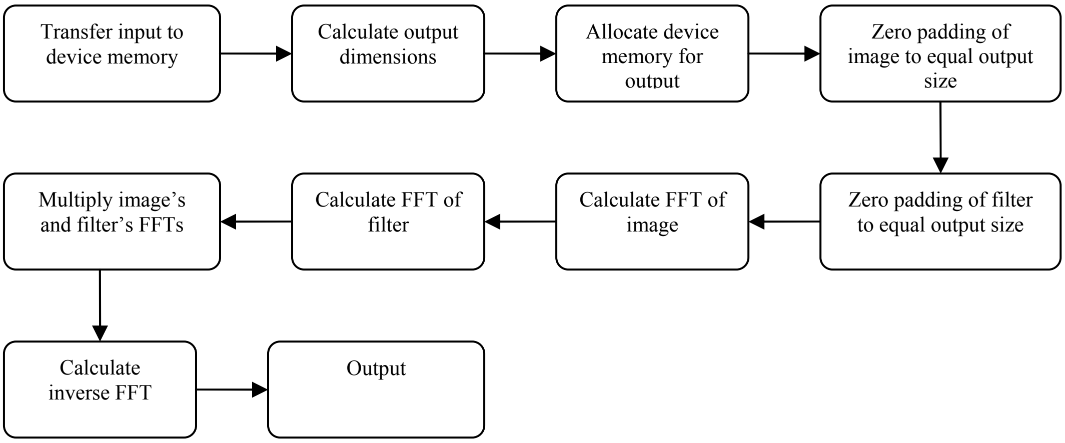

Transfer input to device memory
Calculate output dimensions
Allocate device memory for output
Zero padding of image to equal output size
Zero padding of filter to equal output size
Calculate FFT of image
Calculate FFT of filter
Multiply image's and filter's FFTs
Calculate inverse FFT
Output

Figure 6. Overview of the CUDA implementation of the 2D convolution utilized for the contourlet transform by means of the FFT. Device memory refers to the GPU dedicated memory.

3.4 Run length encoding

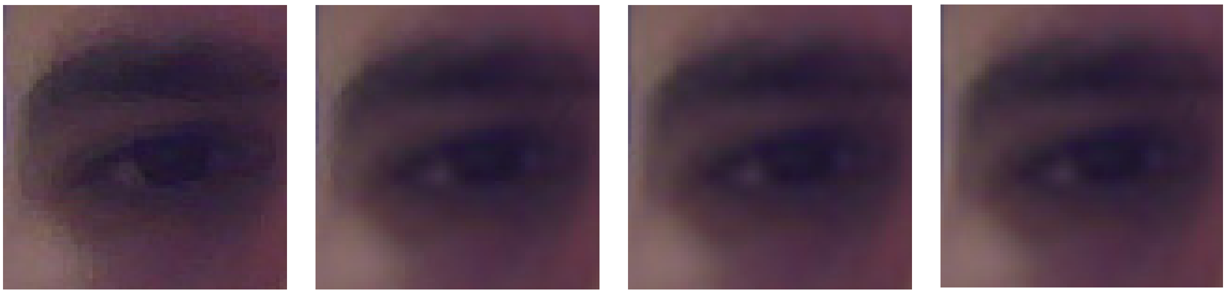
Run length encoding is a simple lossless data compression method in which sequences of the same data value in a data stream are stored as a single data element along with the length of the sequence. Taking into consideration the characteristics of the contourlet transform, the directional subbands of the luminance channel contain large sequences of zero valued contourlet coefficients, making the run length encoding method suitable for their encoding. To reduce the computational cost, only the zero valued coefficients are encoded. Compression gained by run length encoding of all the different values is minimum and does not justify the increased computational cost. Considering the distribution of coefficients of the directional subbands, the optimal direction for the encoding is the horizontal.

4. QUALITY AND PERFORMANCE ANALYSIS

Experiments were conducted using videos captured with a VGA web camera supporting a maximum resolution of 640x480 pixels. Low resolution web cameras are very common on everyday personal computer systems showcasing the need to design video encoding algorithms that take into consideration the problems arising due to low-quality sensors. The videos captured are a typical video-conference video with static background showing the upper part of the human body, containing some motion and a surveillance video depicting an office with no motion.

For the experiments presented in this work, the chrominance channels were subsampled by a factor of 4 and the video stream contained two resolutions, the original VGA (640x480) as well as the lower QVGA (320x240). The method utilized for the reconstruction of the chrominance channels was the nearest neighbor method and the quality parameter adjusted at each encoded video presented was the percentage of the most significant contourlet coefficients of the luminance channel that were retained. Furthermore, at each scale, the luminance channel's high frequency content was decomposed into four directional subbands. It is worth mentioning that dropping all the contourlet coefficients is similar to lowering the luminance channel's resolution while applying a lowpass filter and then upscaling it without reincorporating the high frequency content. In case the user decides to encode the video without the contourlet coefficients, the DFB stage of the contourlet transform does not need to be calculated.

Mosaicing artifacts and noise created due to the low quality of the web camera's sensor are suppressed and replaced by a more fuzzy texture, making the image smoother and more eye friendly, as shown on Figure 7.



Original 5% 2% 0.5%

Figure 7. Example of smoothing due to the dropping of contourlet coefficients. The caption indicates the percentage of the contourlet coefficients retained. Images are cropped and scaled to 140% of their original size.

To show the performance of the algorithm, the sample videos have been encoded using a variety of parameters. A number of percentages of contourlet coefficients to be retained have been selected and the mean PSNR value for each video has been calculated, as well as the compression ratio achieved when using the scheme that incorporates both key frames and internal frames and when compressing all the frames as key frames. The interval between the key frames was set to five frames for the video-conference sample video and to twenty frames for the surveillance video. Detailed results are shown on Table 1 while sample frames of the encoded videos for some settings are shown on Figures 8 and 9.

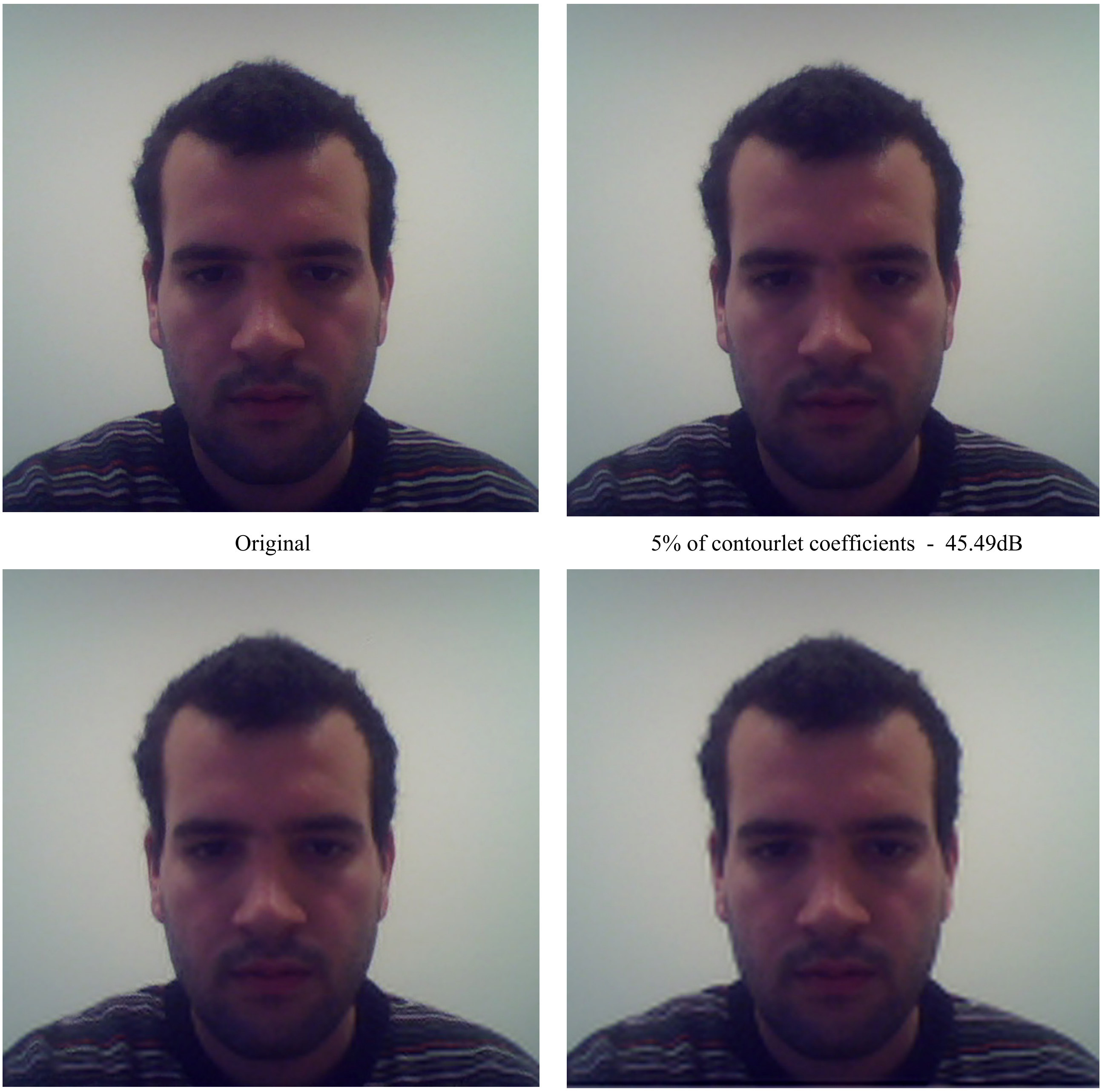
Table 1. PSNR and compression ratios achieved for several settings of the video encoding algorithm. VC refers to the video-conference sample and SU to the surveillance sample video.

Contourlet coefficients retained (%)	**PSNR (dB)**		**Compression ratio**			
			Only key frames		**Key frames and internal frames**	
	VC	SU	VC	SU	VC	SU
10	46.06	40.90	4.86:1	4.46:1	12.97:1	22.32:1
5	45.49	39.76	6.45:1	5.87:1	13.33:1	28.01:1
3	44.83	38.90	7.41:1	6.82:1	15.15:1	31.65:1
1	43.14	37.58	8.73:1	8.29:1	17.73:1	36.90:1
0.5	42.38	37.13	9.09:1	8.79:1	18.38:1	38.61:1
0.2	41.91	36.86	9.30:1	9.10:1	18.76:1	39.53:1
0	39.87	36.29	11.71:1	11.71:1	22.88:1	46.73:1

Examining the compression ratios achieved, it is shown that utilizing both key frames and internal frames outperforms the simple method of encoding all the frames the same way. However, it must be noted that this work is on a preliminary stage and the selection of an efficient entropy encoding algorithm that will further enhance the compression ability of our algorithm is still an open issue.

It is worth mentioning that the contourlet transform exhibits substantial resistance to the loss of contourlet coefficients. Keeping even as few as 5% of its original coefficients, the visual quality of the image is not seriously affected. This fact underlines the efficiency of the contourlet transform in approximating natural images using a small number of descriptors and justifies its utilization in this algorithm.

Real-Time Image and Video Processing 2012, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE Vol. 8437, 843704 - © 2012 SPIE - doi: 10.1117/12.924327



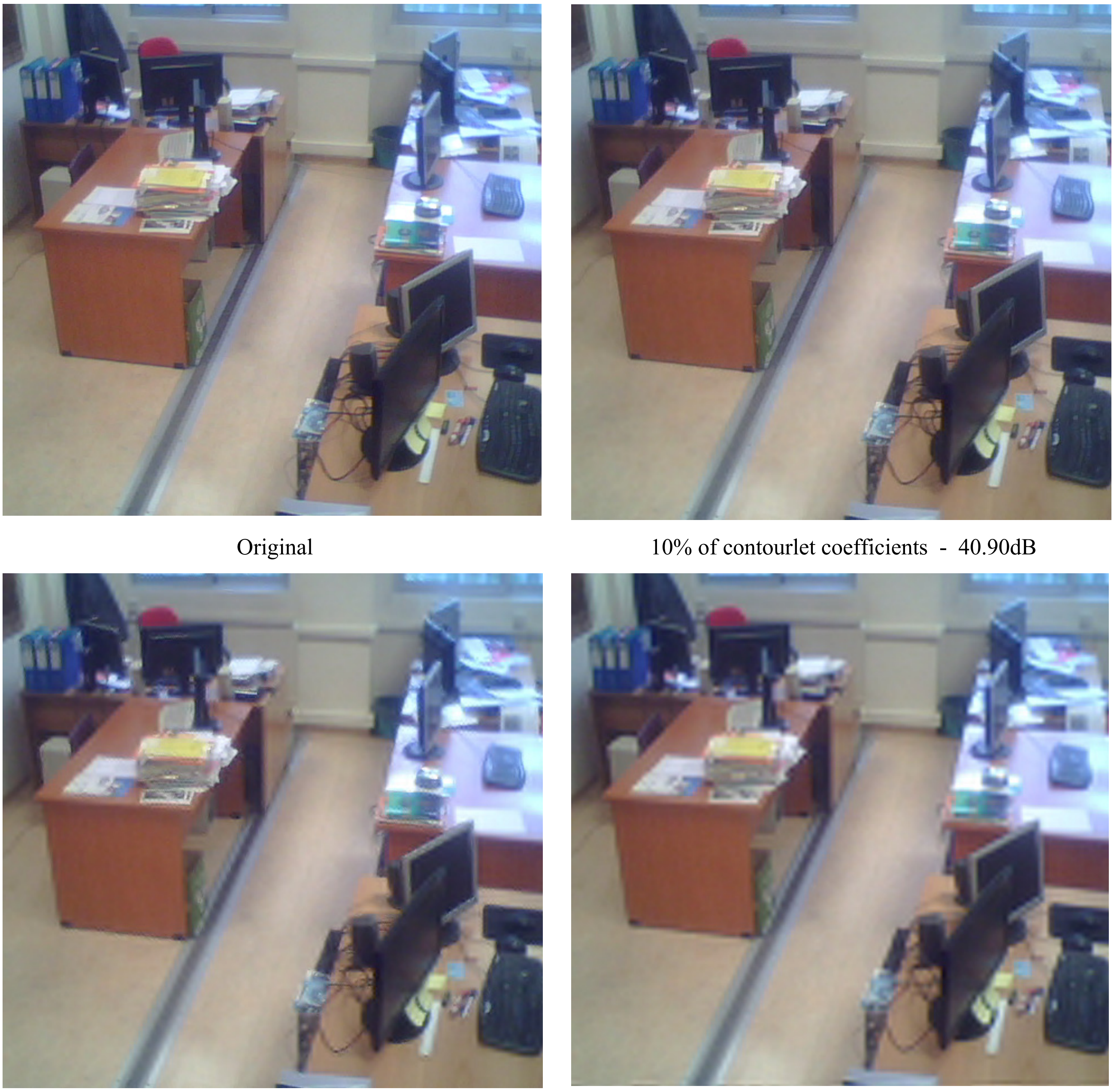
Original

5% of contourlet coefficients - 45.49dB

0.5% of contourlet coefficients - 42.38dB

0% of contourlet coefficients - 39.87dB

Figure 8. Sample frame of the encoded video-conference video for each setting. The frame has been slightly cropped to fit the figure.



Original

10% of contourlet coefficients - 40.90dB

1% of contourlet coefficients - 37.58dB

0% of contourlet coefficients - 36.29dB

Figure 9. Sample frame of the encoded surveillance video for each setting. The frame has been slightly cropped to fit the figure.

5. CONCLUSIONS AND FUTURE WORK

In this work, a low complexity algorithm for real-time video encoding optimized for video conferencing applications and surveillance cameras has been proposed. The method provides an ideal scalable video compression scheme for video conferencing content as it achieves high quality encoding and can dramatically increase compression efficiency for static regions of the image, while maintaining low complexity and can adapt to the receivers resources. One video stream can contain various resolutions of the video without the need for reencoding at the source. The receiver can select the desired

Real-Time Image and Video Processing 2012, edited by Nasser Kehtarnavaz, Matthias F. Carlsohn,
Proc. of SPIE Vol. 8437, 843704 - © 2012 SPIE - doi: 10.1117/12.924327

quality by dropping the components referring to higher quality than needed. Furthermore, the manipulation of the structural characteristics of the video through the manipulation of the contourlet transform coefficients leads to the suppression of noise induced by low-quality sensors without the need of an extra denoising or image enhancing stage. When higher compression is needed as in the case of long recordings for surveillance systems, the visual quality degradation is much more human eye friendly than with other well established video compression methods, as it introduces fuzziness and blurring instead of artificial block artifacts, providing smoother images. Additionally, the use of small GOPs makes the algorithm resistant to frame losses that can occur during transmission over IP networks. Finally, the utilization of the usually "dormant" GPU computational power lets the CPU to be utilized for other tasks, further enhancing the multitasking capacity of the system and enabling the users to take full advantage of their computational capabilities.

This work is on preliminary stage. In order to compete for compression efficiency with state of the art video compression algorithms, a highly efficient entropy encoding scheme has to be incorporated to the algorithm. The optimal tradeoff between compression rates and complexity has to be decided in order to retain the low complexity and real time characteristics of our algorithm. Also, as stated on §2.1 the creation of optimal filters for the contourlet transform is still an open research topic. Further improvement of the filters utilized would have a positive effect on the visual quality achieved by our encoding algorithm.

REFERENCES

[1] Do, M. N. and Vetterli, M., "The contourlet transform: an efficient directional multiresolution image representation," IEEE Transactions on Image Processing 14(12), 2091-2106 (2005).

[2] Liu, Z., "Minimum Distance Texture Classification of SAR Images in Contourlet Domain," Proc. International Conference on Computer Science and Software Engineering (2008).

[3] Katsigiannis, S., Keramidas, E. and Maroulis, D., "A Contourlet Transform Feature Extraction Scheme for Ultrasound Thyroid Texture Classification," Engineering Intelligent Systems 18(3/4), (2010).

[4] Liu, Z. and Xu, H., "Image denoising using Contourlet and two-dimensional Principle Component Analysis," Proc. International Conference on Image Analysis and Signal Processing (IASP 2010), 309-313 (2010).

[5] Burt, P. J. and Adelson, E. H., "The Laplacian Pyramid as a compact image code," IEEE Transactions on Communications 31(4), 532-540 (1983).

[6] Bamberger, R. H. and Smith, M. J. T., "A filter bank for the directional decomposition of images: Theory and design," IEEE Transactions on Signal Processing 40(4), 882-893 (1992).

[7] Shapiro, J. M., "Embedded image coding using zerotrees of wavelet coefficients," IEEE Transactions on Signal Processing 41(12), 3445-3462 (1993).

[8] Vetterli, M., "Multidimensional subband coding: Some theory and algorithms," Signal Processing 6(2), 97-112 (1984).

[9] Yifan, Z. and Liangzheng, X., "Contourlet-based feature extraction on texture images," Proc. International Conference on Computer Science and Software Engineering, 221-224 (2008).

[10] Cohen, A., Daubechies, I. and Feauveau, J. C., "Biorthogonal bases of compactly supported wavelets," Communications on Pure and Applied Mathematics 45(5), 485-560 (1992).

[11] Malvar, H. and Sullivan, G., "YCoCg-R: A Color Space with RGB Reversibility and Low Dynamic Range," Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Document No. JVTI014r3, (2003).

[12] Katsigiannis, S., "Acceleration of the Contourlet Transform," M.Sc. thesis, (2011).