ISSN:0975-9646

N.Bharath et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 1522-1525

Video Compression Using JPEG2000

N.Bharath[#1], V.Prakash[#2], G.Dinesh[#3], V.Dillibabu[#4], S.Kumarasamy[*5]

[#1,2,3,4] *Student*, [#5] *Assitant Professor,*
Department of Information Technology, SKP Engineering College.

Abstract **- We work on multiresolution transforms, used in compression systems of image and video in which JPEG2000 is an update to JPEG. We process the basis constructions for the wavelet transforms, and estimate some of their characteristics with those of hierarchical transforms. In the compression performance as measured by PSNR(peak signal to noise ratio), the best performance is provided by H.264, but at a higher computing problems and also JPEG2000 gives a good fidelity ratio compared to H.264 the quality then improves progressively through downloading more data bits from the source. On giving importance to the JPEG2000 and video compression using the same it's also called as "multirate multimedia streaming". In relation to visual quality, the multiresolution transforms provide an improvement more than block (single resolution) transforms.**

Index Terms- *Multiresolution Transform, Wavelet Transform, PSNR etc…*

I. INTRODUCTION

Compression of natural images (pictures) and video is quite common today; for example in Web pages we usually find images compressed with the JPEG (Joint Photographic Experts Group) codec (coder/decoder) and video compressed with several kinds of MPEG (Moving Picture Experts Group) codecs. A fundamental approach towards compression of media signals is to remove redundancy via signal prediction or linear transforms (or a combination of both), followed by a quantization (scaling and rounding to a nearest integer) and entropy coding (representing those integers with a small number of bits by exploiting their joint statistics). The scaling factor in the quantization process controls the basic tradeoff between compressed file size and decoded signal fidelity. In Fig. 1 we show a basic diagram representing the processing steps of a modern image or video compression that uses those ideas. By cascading a pixel-domain predictor with a transform operator, we mean that the prediction residuals transform is computed. The color space mapper is a first step of redundancy reduction, usually converting the pixels from an R-G-B color space to a luminance and chrominance space, such as Y-Cb-Cr (luma, blue-luma, and red-luma), with the luma and chroma images typically being encoded independently.

For video coding, pixel prediction is usually nonlinear, through motion compensation – a motion field applied to a previously-encoded frame. In image coding, most codecs do not use pixel prediction, so that a linear transform is applied directly to the image pixels. A notable exception is the new H.264 (also referred to as MPEG-4 Part 10) video codec, in which "intra" frames (those encoded independently, that is, without motion-based prediction from other frames) use pixel prediction from previously-encoded blocks within the same frame.

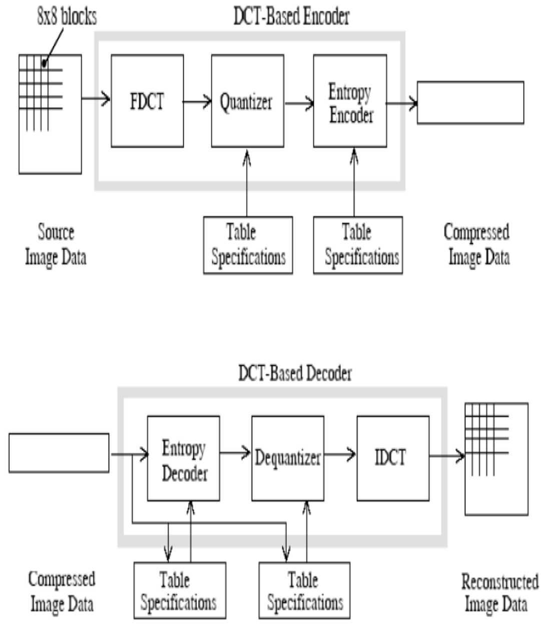

Fig 1: Block diagram for video coding of JPEG2000

II. MULTIRESOLUTION TRANSFORMS

The transform operator is not applied to the image as a whole, but rather to blocks of pixels. In codecs such as JPEG [1] or MPEG [2], the blocks have the fixed size ox 8×8, and the transform is a DCT (discrete cosine transforms). Other transforms can be used, but the DCT is fast-computable and is nearly optimal in terms of energy compaction, that is, for typical blocks the low-frequency coefficients have high magnitudes, whereas the high-frequency coefficients have low magnitudes. After quantization, many of the high-frequency coefficients are truncated to zeros, which are efficiently compressed by the entropy coder. The choice of block size is determined by a basic tradeoff: larger blocks are better for encoding flat regions, but small blocks lead to fewer ringing artifacts due to the missing high-frequencies. The sets of pixels that form the blocks can be either disjoint (non-overlapping), as in JPEG or MPEG, or overlapping, as in wavelet-based or lapped-transform-based codecs. The main disadvantage of using non-overlapping transforms is the appearance of blocking artifacts at high compression ratios. Older codecs such as JPEG and MPEG use a fixed-resolution transform, whereas modern codecs such as H.264 and JPEG2000 use multiresolution transforms.

Multiresolution signal analysis is used in many applications. In many cases, such as image coding, by multiresolution we usually mean a small set of resolutions (two to six), associated to longer block sizes for low-frequency components and shorter block sizes for high-frequency components. That works well for images, where high frequencies tend to be associated with short-duration features, such as edges and lines.

Besides representing image pixel data, another application for multiresolution transforms in video coding is in motion estimation and compensation. Typically we measure motion by cross-correlating blocks of a pair of frames and we use motion vectors to displace and interpolate pixels from the reference frame to generate predictions of pixels of the current frame. An alternative approach is to apply a complex-valued transform (wavelet or lapped) to both the reference and the current frames, and use phase measurements and phase shifting to perform motion estimation and compensation, respectively. Advantages of this transform-based approach are much finer precision in estimated motion vectors and smoother motion compensation, without blocking artifacts. One disadvantage is that it is difficult to encode images efficiently in a complex-valued transform domain.

In practice, an efficient way to obtain multiresolution signal decompositions is to apply a first transform operator to the signal, then a second transform operator to a set of low-frequency coefficients of the first transform. The low frequency coefficients of the second transform can be transformed by a third operator, and so on, up to the desired number of levels. We call this generic cascade of transform operators a *hierarchical transform*. An important case is when the transform operators are two-band decompositions and the low-frequency subbands are sent to the next transform operator, which is the well-known tree structure for a *discrete wavelet transform*. There is a vast amount of literature on wavelet image compression. Thus, in this paper we pay more attention to hierarchical transforms.

III. COMPARATIVE PERFORMANCE OF MULTIRESOLUTION IMAGE AND VIDEO CODERS

JPEG2000, H.264 and PTC (progressive transform coder). Our emphasis is on the multiresolution transforms used in those codecs, and on their performance in encoding independent frames. Of course, the compression performance of those codecs is mostly determined by their entropy coding engines that follow transform coefficient quantization, but discussing the entropy coding algorithms is outside the scope of this paper. Here we focus on the kinds of distortions that are generated at high compression rates, which depend on the choice of multiresolution transform.

JPEG2000 uses the well-known “CDF 9/7” biorthogonal wavelet filters. By relaxing the constraint that the direct (analysis) and inverse (synthesis) transform operators must use the same basis functions, biorthogonal constructions have two main advantages: first, the synthesis basis functions can have a higher degree of smoothness than the analysis ones, which is important to minimized decoded image artifacts; second, the coding gains of biorthogonal transforms are higher than those of their orthogonal counterparts.

The original pictures are 352×288-pixel rectangles from an image of the JPEG2000 test set and from an image of the Kodak test set. That picture size is referred to CIF (common intermediate format), which is one of the supported formats in H.264. For each picture, we encoded and decoded it using the JPEG, JPEG2000, PTC, and H.264 codecs, setting the quality/quantization parameters for a compression ratio of 86:1, corresponding to a bit rate of 0.28 bits/pixel. That is a relatively high ratio, which we chose so that the compression artifacts are visible, but acceptable for applications such as posting in a Web page or photo printing, especially if the entire images have over 1 million total pixels, which is quite common in today’s digital photography scenarios.

Fig 2: 001.tif (uncompressed image)

Fig 3: 001.jpg

Fig 4: 001.jp2

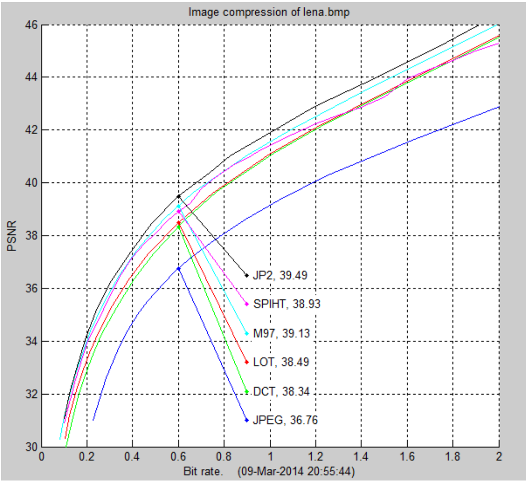

Fig 5: Plot based on compression algorithms

Above figures show the different extensions of the first frame of a video but which are converted into images where image compression takes place.

We present the PSNR (peak-signal to noise ratio, defined as the ratio in decibels between the peak values of an unsigned 8-bit pixel to the root-mean-square error in the pixel values of the decoded image) for each of the codecs, as well as the encoding times. Those times were measured in a Pentium-IV 2.4 GHz machine, after the original image files have been cached in RAM. The JPEG codec was the latest release from the Independent JPEG group. The various compression algorithm comparison of PSNR is given below,

S.NO	PSNR		BIT RATE
1	JP2	39.49	0.6
2	SPIHT	38.93	0.6
3	M97	38.93	0.6
4	LOT	38.49	0.6
5	DCT	38.34	0.6
6	JPEG	36.76	0.6

Representing various PSNR values of image 001.bmp (or) tiff

We see that the JPEG codec tends to produce a sharper appearance, but that comes at the price of excessive ringing around edges. The time-domain predictors in H.264 perform well, thus reducing the energy of the mid- and high-frequency coefficients. At the relatively high compression ratio of 86:1, most of such coefficients are quantized to zero, leading to significant blurring. User tests are usually inconclusive about the trade-off between blurriness and ringing, but there seems to be a slight preference for blurriness. When the PDF file for this paper is viewed at an increased zoom (> 200%), the blocking artifacts of JPEG are apparent, because its analysis and synthesis basis functions are not overlapping, and there is no mechanism to exploit pixel correlation across blocks. We recall that besides the pixel-domain predictor, H.264 has a nonlinear post filter that reduces blocking artifacts.

Besides quality vs. complexity there are other aspects that we have not considered. One example is partial decoding. In JPEG the entire frame has to be decoded up to the desired blocks, since encoding of a block depends on all previously-encoded blocks1. In JPEG2000 it is possible to decode only a small rectangle of an image (useful when browsing large images, e.g. those with many millions of pixels). Also, In JPEG2000 it is possible to decode a reduced resolution version of the image, for viewing at a reduced zoom factor or for fast thumbnail generation. It is clear that a reduced resolution image can be decoded by decoding only the coefficients of the second level transform, and performing only the second level inverse transform, thus generating an image that has a quarter of the size of the original image, in each dimension. Because of the relatively smooth filters of the hierarchical transform, such reduced-resolution decoding produces results almost as good as downsampling with a good filter (e.g. bicubic). Finally, JPEG2000 also generate progressive bitstreams, meaning that the quantized coefficients are encoded in bit plans, starting from the most significant bit. That way, given an encoded JPEG2000, it is possible to generate another encoded file corresponding to a higher compression ratio by simply parsing out some of the bits in the original compressed file. In other words, further compression can be performed very quickly, directly in the compressed domain, without decoding and re-encoding.

IV. Further Research

A natural question that arises is there if there are better designs for hierarchical transforms, for use in multiresolution image coders. The answer is yes, and recent research efforts show promising results. For example, in the JPEG2000 codec we used only two levels of transform (wavelet), there is a loss of smoothness in the basis functions. If we were to perform another level of transformation with an LBT, the resulting loss of smoothness would lead to noticeable artifacts even at moderate compression levels. One way to increase the smoothness (or regularity) of lapped transform basis functions is to increase their length, with more than 50% overlap across blocks, as in designs based on GenLOT (generalized lapped orthogonal transform). Also, using a different implementation structure for LBTs via time-domain pre- and post-filtering, it may be possible to obtain higher regularity while still maintaining a fast computation algorithm. Also the characteristics of hierarchical and wavelet transform is to be compared for the video.

V. Conclusion

We have given a brief analysis of the multiresolution transform designs JPEG2000, and a single-resolution codec, JPEG. While PTC and H.264 use hierarchical transforms, JPEG2000 uses wavelet transforms. Although wavelet transform can potentially lead to better compression performance, hierarchical transforms can lead to faster processing times, as well as easier implementation of region decoding. Recent developments in the design of fast hierarchical transform may lead to codecs with quite similar performance to those based on wavelet transforms, but potentially more efficient implementations.

REFERENCES

1. "*Multiresolution transforms in modern image and video coding systems"* by Henrique S. Malvar Microsoft Research, One Microsoft Way, Redmond, WA 98052
2. *A Tutorial of the Wavelet Transform* Chun-Lin, Liu February 23, 2010
3. *An Evaluation of Motion JPEG 2000 for Video Archiving* Glenn Pearson and Michael Gill Lister Hill National Center for Biomedical CommunicationsNational Library of Medicine, NIH/HHS, Bethesda, MD
4. *JPEG 2000 for Long-term Preservation: JP2 as a Preservation Format* Johan van der Knijff KB/National Library of the Netherlands johan.vanderknijff@kb.nl may2011-vanderknijff
5. I. E. G. Richardson, H.264 *and MPEG-4 Video Compression*. New York: Wiley, 2003.
6. H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "*Low-complexity transform and quantization in H.264/AVC*," *IEEE Trans. Circuits Systems Video Tech.*, vol. 13, pp. 598–603, July 2003.
7. *IJG JPEG codec*, V. 6b, Mar. 1998. See http://www.ijg.org.
8. *Image Power JPEG 2000 Codec Beta Preview*, Version 1.0.0.1, Feb. 2000. See http://www.imagepower.com.
9. *H.264 Reference software version JM60*, Jan 2003. See ftp://ftp.imtc-files.org/jvt-experts/reference_software.
10. S. Oraintara, T. D. Tran, and T. Q. Nguyen, "*Regular biorthogonal linear-phase filter banks: theory, structure and application in image coding*," *IEEE Trans. Signal Processing*, vol. 51, pp. 3220–3235, Dec. 2003.
11. W. Dai and T. D. Tran, "*Regularity-constrained pre- and post-filtering for block DCT based systems*," *IEEE Trans. Signal Processing*, vol. 51, pp. 2568–2581, Oct. 2003.
12. D. S. Taubman and M. W. Marcellin, JPEG2000*: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic, Boston, MA, USA, 2002.
13. M. Rabbani and R. Joshi, *"An overview of the JPEG2000 still image compression standard*," Signal Processing: Image Communication, vol.17, no. 1, pp. 3–48, Jan. 2002.
14. *Graphics processing unit implementation of JPEG2000* for hyperspectral image compression a b Milosz Ciznicki,a Krzysztof Kurowski, a and Antonio Plaza Poznan Supercomputing and Networking Center, Noskowskiego 10, 61-704 Poznan/2012, Poland University of Extremadura, Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Avda. de la Universidad s/n. 10071 Cáceres, Spain aplaza@unex.