International Journal of Inventive Engineering and Sciences (IJIES)
ISSN: 2319–9598, Volume-1, Issue-9, August 2013

Arbitrary-Ratio Image/Video Resizing Using Fast DCT of Composite Length for DCT-Based Transcoder

Dipika P.Chanmanwar, Priyanka S. Ghode

***Abstract*—the most popular image and video compression methods such as JPEG, MPEG 1/2/4, H.261/3/4 use transform domain techniques and in particular the Discrete Cosine Transform (DCT). One application for such images or video sequences is resizing. Resizing is extensively used to meet the requirements of a specific system, to satisfy user's interests, or to correct spatial distortions. However, a major difficulty encountered when resizing such media is the high computational complexity and the loss of quality caused by the decompression and compression.**

The purpose of this paper is to implement an arbitrary ratio image resizing scheme in the DCT domain for transcoding of the compressed images. There are several advantages in working in the DCT domain, of these advantages the one that stands out the most is the fact that images are stored in the DCT domain and therefore no initial computation is needed in order to work on the image. The downsizing process in the discrete cosine transform (DCT) domain can be implemented by truncating high-frequency coefficients, whereas the upsizing process is implemented in the DCT domain by padding zero coefficients to the high-frequency part. The implemented method combines a fast inverse and forward DCT of composite length for arbitrary-ratio upsizing or downsizing. The implemented method shows a good peak signal-to-noise ratio and less computational complexity compared with the spatial-domain and previous DCT-domain image resizing methods.

Further it will compare several methods offered by different authors for image resizing. The implemented method of arbitrary ratio image resizing improve peak signal-to-noise ratio and reduces computational complexity when compared with other existing methods.

This implemented approach of image resizing is extended for video resizing. The PSNR values of the resized video are calculated by using an existing tool. The obtained PSNR values are better when compared with other existing tools.

***Index Terms*— Arbitrary-ratio image resizing, composite length DCT, transcoder.**

I. INTRODUCTION

With the explosion in the use of digital images the need for efficient tools to manipulate those images aroused. Image resizing is a fundamental and extremely important type of image manipulation. Any tool that can perform this resizing efficiently or that can be combined easily with other image processing functions is therefore of great value.

Most images or video frames are stored in a compressed format. However, the user often requires images with different resolution for many applications. For example, it would be efficient to send a low-resolution version of an image to a remote client and then, if the client has interests in the image, the full-resolution image could be transferred later. Changing the size of the video frame would be required to convert between various digital TV standards like standard TV and HDTV, and also to fit the incoming video frame onto the user's screen. Therefore, image resizing is a fundamental and important operation of image manipulation.

Image resizing is an essential aspect in digital image processing. It is widely applied in numerous fields such as medical image processing, military applications and consumer electronics. For instance, we have to enlarge images in HDTV or medical image display, or a scale-down image will fit the minimize LCD panel in portable instruments. It is needed to ensure that images and video streams are tailored to the communication networks over which the streams travel and to the end user display devices upon which they will be presented.

Arbitrary image resizing is of great importance. Since so much multimedia material (especially images and video) is compressed using the popular block DCT framework, a resizing operation that is efficient (i.e. offers less complexity than a direct inverse transformation, spatial domain resizing, and forward transformation) and occurs in the block DCT domain is desirable. The efficiency of any image resizing algorithm is determined by two main factors: the quality of the obtained image and the computational complexity.

Image manipulation in a compressed domain is usually more efficient than that in a spatial domain in terms of computational complexity and image quality. Numerous approaches have been applied to image and video processing in the compressed domain.

Dugad and Ahuja [1] have proposed an elegant scheme for changing the image sizes in the DCT space. They have suggested a simple fast computation technique for halving and doubling of images using their low-frequency components. The principle behind the algorithms developed by Dugad and Ahuja is similar to the subband DCT computation.

In (2002) J. Mukharjee and S. Mitra proposed two modifications to the above algorithm. First they use the approximations while converting DCT coefficients from N/2-point to N–point or vice versa. Secondly, during image doubling, we directly use the DCT coefficients of the compressed image for converting it to a 16×16 block in the spatial domain by applying a 16-point IDCT. The proposed modifications considerably increase the computational overhead during image-doubling.

In (2003) H. W. Park, Y. S. Park, and S. K. Oh propose a novel approach to resize images with resizing ratio in the discrete cosine transform (DCT) domain, which exploits the

Manuscript Received on August 2013.

Dipika chanmanwar, Lecturer in Government Polytechnic, Nagpur (Maharastra), India
Priyanka Ghode, Lecturer in Government Polytechnic, Nagpur (Maharastra), India

multiplication-convolution property of DCT (the multiplication in spatial domain corresponds to the symmetric convolution in DCT domain). When an image is given in terms of its 8×8 block-DCT coefficients, its resized image is also obtained in 8×8 block-DCT coefficients. The proposed approach is computationally fast and produces visually fine images with high PSNR.

Salazar and Tran (2004), proposed method used to arbitrarily rescale an image is based upon a generalization of the technique proposed in [1]. In our approach, a single mapping is constructed that implicitly involves a combined resizing and inverse transform back into the spatial domain followed by a combined resizing and forward transform into the 8 by 8 DCT domain. By combining these operations into a single mapping, complexity can be reduced over that of the method of Park et. al. Because we resize at both the inverse and forward transformations we can get any X/Y scale factor (rather than just powers of two as in the method of Dugad and Ahuja). By choosing different N-point inverse and M - point forward DCTs we can use more or less of the original image data to vary the final image quality.

Ee-Lang Tan, Woon-Seng Gan, and Meng-Tong Wong Our work is based on the framework proposed in [11]. To avoid significant degradation in resized images, we choose to perform up-sampling first followed by down-sampling to resize images. In addition, optimization is introduced to reduce the computational cost.

Resizing images by factors of $P/Q \times R/S$ are discussed in [5] and [2]. Salazar and Tran [2] proposed an arbitrary resizing algorithm in the DCT domain which is a generalization of the approach proposed by Dugad and Ahuja [1]. However, authors in [5] did not suggest the optimal length for DCT and IDCT for high PSNR. In [11], Mukherjee and Mitra have proposed to resize images with the spatial relationship of the DCT coefficients between a block and its sub-blocks [12]. Mukherjee and Mitra have avoided high computational cost by performing down-sampling by factors of $Q \times S$ followed by up-sampling by factors of $P \times R$ to resize images. However, performing down-sampling first instead of up-sampling introduces significant degradation [20].

II. IMPLEMENTED IMAGE RESIZING METHOD IN THE DCT DOMAIN

The implemented fast arbitrary-ratio image resizing method in the DCT domain produces visually fine images with high PSNR. Furthermore, this method is faster than the previous arbitrary-ratio resizing methods. The arbitrary-ratio image resizing is realized by the fast inverse and forward DCT of composite length [6], [7].

A conceptual diagram of the L/M-fold resizing is shown in Figure 3.1. For L/M-fold resizing, M consecutive 8-sample blocks are converted to L 8-sample blocks. Therefore, M 8-sample blocks are grouped as shown in Fig. (A). There it denotes 8-sample DCT coefficients of the the ith block in a group of M blocks. The total sample number of the original M blocks is 8×M. In order to perform the L/M-fold resizing, its implementation requires the number of total samples in a group of blocks should be a common multiple of L and M. If not, zero padding is required for each block.

Let L/M be an irreducible fractional number. The implemented method of L/M-fold resizing is processed with M-blocks unit, each block having a length of eight. First, the (8+q)-sample IDCT is performed for M-blocks. For this, zero padding of samples is required in advance.

The inverse transformed sequences have a total (8+q)×M of samples in a group of M blocks, which are retransformed by DCT with a length of (8+q)×(L/M). Then truncate the high-frequency coefficients except eight low-frequency coefficients, thereby reconstructing L blocks of eight samples in the DCT domain. The resized image can be obtained through this IDCT and DCT processing. No further processing is needed.

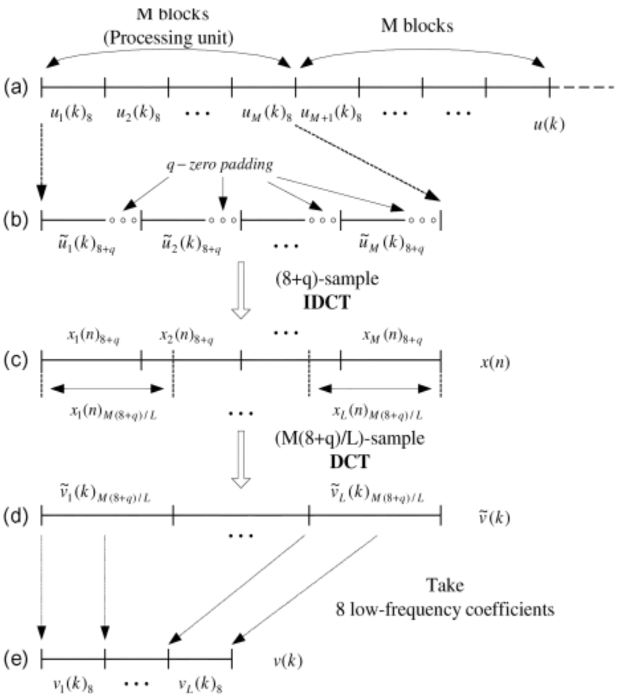

Fig. (A) Conceptual diagram of the proposed L/M-fold resizing.

The algorithm for the implemented approach is given by the following steps:

1. The determined DCT coefficients are divided into processing units of M blocks.
2. According to the resizing ratio, the numbers of padded zeros are appropriately determined and zero padding is done in DCT domain.
3. Then (8+q)-sample IDCT is performed for M blocks and in spatial domain M blocks are splitted into L blocks.
4. To be in frequency domain, (M (8+q)/L)-sample DCT is determined for L blocks.
5. Image is resized by truncating high-frequency parts.

III. VIDEO RESIZING ALGORITHM

The approach used for image resizing which is discussed in the previous sections is extended for resizing YUV videos. The images used are BMP images. So by using similarities between these two file formats the approach is extended to YUV video resizing. There are many advantages of using YUV video file format which is further explained.

YUV is a color space pixel format and a file extension for a raster graphics file (RIF). YUV files contain bitmap image data stored in the YUV format, which splits color across Y, U, and V values. It stores the brightness (luminance) as the Y value and the color (chrominance) as U and V values.

YUV video files are encoded with raw (i.e. uncompressed and header less) 4:2:0, 4:2:2 and 4:0:0 formats. Video file encoded in the YUV format stores a sequence of YUV images as a single video file may be saved in YUV 4:2:0, 4:2:2, 4:4:4 formats.

A raster image file is generally defined to be a rectangular array of regularly sampled values, known as pixels. Each pixel (picture element) has one or more numbers associated with it, generally specifying a color which the pixel should be displayed in. BMP and YUV are raster image file format. Raster graphics are the most common type of image files. They are comprised of a grid of pixels where each pixel represents an individual color within the image. Both Web graphics and digital photos are stored as raster graphics. While some raster image formats are uncompressed, most use some type of image compression. Common raster image file extensions include .BMP, .TIF, .JPG, .GIF, and PNG. Other image file categories include Vector Graphic and 3D Image files.

The primary advantage of YUV video format is that it remains compatible with black and white analog television. The Y signal is essentially the same signal that would be broadcast from a normal black and white camera (with some subtle changes), and the U and V signals can simply be ignored. When used in a color setting the subtraction process is reversed, resulting in the original RGB color space.

Another advantage is that the signal in YUV video format can be easily manipulated to discard some information in order to reduce bandwidth. The human eye actually has fairly low color resolution, the high-resolution color images being processed by the visual system by combining the high-resolution black and white image with the low-resolution color image. Standards such as NTSC reduce the amount of signal in the U and V considerably; it saves only 11% of the original blue and 30% of the original red, throwing out the rest. Since the green is already encoded in the Y signal, the resulting U and V signals are substantially smaller than they would otherwise be if the original RGB or YUV signals were sent.

For resizing YUV videos same resizing scheme is used which is used for resizing grayscale BMP images. The implemented approach for image resizing will take only Y component into consideration. For YUV video resizing implementation is done for three components. As the YUV files are header less files, while resizing the size of the video must be specified. By making some changes in the implementation the same approach is used for YUV video resizing.

The robustness of the scheme is tested by calculating PSNR values of the resized video and original video. Again the original video is resized by using an existing video resizing tool. Then PSNR values are calculated for the resized video which is resized by resizing tool and original video. These two PSNR values are compared. We are getting PSNR values better for the resized video which is resized by the proposed scheme. A higher PSNR would normally indicate that the reconstruction is of higher quality. Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB where higher is better.

IV. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

We compare PSNRs of the implemented method and the previous methods for various images obtained by 2/3-fold resizing and 3/4 -fold resizing. The PSNRs are measured between the original image and M/L-fold upsized image after L/M-fold downsizing of the original image. Table I shows the PSNRs of the 3/2-fold upsized images after 2/3-fold downsizing by using the previous and the implemented method.

The PSNR values given in table I for three images are obtained by first downsizing using 2/3 resizing ratio then upsizing by 3/2 ratio. Bilinear interpolation is a spatial domain image resizing technique. Results of the bilinear interpolation method are not satisfactory.

Table I PSNRs of Various Resizing Methods for the 3/2-fold Upsized images after 2/3-fold Downsizing.

Method(down: up)(2/3 : 3/2)	Lena	Boat	Baboon	
Bilinear /Bilinear	33.19	30.80	24.21	
Shu[3] /implemented	38.81	35.46	27.04	
Previous /Previous[4]	38.73	35.40	27.01	
Implemented /Implemented	38.81	35.46	27.04	

Table II Computational amount (Per pixel of the original image) of Various Resizing Methods for 3/2-Fold Upsizing (2<3).

Method	3/2-fold upsizing	
	Multiplications	Additions
Bilinear	6.19	12.14
Previous[4]	9.79	11.94
Implemented	4.24	12.85

We also analyzed the computational amount of multiplications and additions for various resizing methods. Loeffler et al. proposed a fast 8-sample DCT algorithm [6], and fast algorithms for various radix DCTs were developed [7], [9].The method was implemented with the fast DCT and IDCT introduced in [6], [7] and [9].Table II shows the computational amount of the implemented and the previous methods for 3/2-fold upsizing.

For getting the following result a 512 x 512 BMP image is provided as an input along with resizing ratio 2:3 .The resultant image is scaled accordingly. Fig. B) is the input image and Fig. (C) is the resized image.

Fig. (B) A 512 x 512 I/P Image

Fig. (C) A 341 x 341 I/P Image

For resizing of YUV videos, a YUV video is given as an input along with resizing ratio. Each frame of the input video is resized according to the given resizing ratio. The input video is resized by the implemented approach and the existing YUV tool by the same input resizing ratio. And PSNR values are calculated for both between the resized and original video which are given subsequently:

Table III PSNR values for the video resized by YUV tool

Frame	SNR_Y	SNR_U	SNR_V
1	31.196	45.1563	47.4215
2	31.195	45.173	47.438
3	31.1921	45.1532	47.4411
4	31.1899	45.1843	47.4203
5	31.1731	45.1785	47.4378
6	31.1526	45.1912	47.4456
7	31.1643	45.197	47.4466
8	31.1859	45.198	47.4523
9	31.1922	45.194	47.4641
10	31.2007	45.1818	47.3987
11	31.2229	45.1904	47.4216
12	31.2559	45.2063	47.4515
13	31.2583	45.2771	47.5456
14	31.3253	45.4217	47.6243
15	31.4169	45.4833	47.5992

Table IV PSNR values for the video resized by proposed scheme.

Frame	SNR_Y	SNR_U	SNR_V
1	23.1476	47.0183	49.4816
2	23.1663	47.0545	49.4635
3	23.1502	47.0477	49.4962
4	23.2019	47.0622	49.498
5	23.2229	47.0575	49.5017
6	23.1902	47.069	49.4967
7	23.1014	47.0595	49.5093
8	23.2071	47.103	49.5177
9	23.1882	47.0767	49.5254
10	23.1002	47.0615	49.4792
11	23.228	47.0746	49.474
12	23.112	47.0953	49.488
13	23.0673	47.1781	49.635
14	23.1341	47.3299	49.7177
15	23.1907	47.3971	49.6847

From these tables it is clear that PSNR for the resized video which is resized by the proposed scheme is better than the PSNRs of the video resized by the YUV tool.

V. CONCLUSION

The implemented method presented a new approach to digital image resizing whereby all processing is performed in the DCT domain. This approach of image resizing has shown how to perform pixel-domain downsampling and upsampling by simple manipulation of the DCT-domain values. The resizing method is implemented through a combination of inverse and forward DCT of composite lengths. It is extended to arbitrary resizing ratio.

The proposed method produces visually fine images and improves the PSNR of the resized images. This method also demonstrates that the computational complexity of the proposed method is better than those of the previous methods. The proposed approach is extended for video resizing and improves the PSNR of the resized video.

REFERENCES

[1] R. Dugad, “A fast scheme for image size change in the compressed domain ,” IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 4, pp. 461–474, Apr. 2001.

[2] J. Mukherjee and S. K. Mitra, “Image resizing in the compressed domain using subband DCT,” IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 7, pp. 620–627, Jul. 2002.

[3] H. Shu and L. P. Chau, “An efficient arbitrary downsizing algorithm for video transcoding,” IEEE Trans. Circuits Syst. Video Technol, vol. 14, no. 6, pp. 887–891, Jun. 2004.

[4] H. W. Park, Y. S. Park, and S. K. Oh, “L/M-fold image resizing in block-DCT domain using symmetric convolution,” in IEEE Trans. Image Process., vol. 12, Sep. 2003, pp. 1016–1034.

[5] C. L. Salazar and T. D. Tran, “On resizing images in the DCT domain,” in Proc. IEEE Int. Conf. Image Processing, Singapore, Oct. 2004, pp. 2797–2800.

[6] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, “Practical fast 1-D DCT algorithms with 11 multiplications,” in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, vol. 2, May 1989, pp. 988–991.

[7] G. Bi and L. W. Yu, “DCT algorithms for composite sequence length,” IEEE Trans. Signal Process, vol. 46, no. 3, pp. 554–562, Mar. 1998.

[8] Y. S. Park and H. W. Park, “Design and analysis of an image resizing filter in the block-DCT domain,” IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 2, pp. 274–279, Feb. 2004.

[9] Y. H. Chan and W. C. Siu, “Mixed-radix discrete cosine transform,” IEEE Trans. Signal Process., vol. 41, no. 11, pp. 3157–3161, Nov. 1993.