VOL. 11, NO. 17, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

VIDEO COMPRESSION ON H.264/AVC BASELINE PROFILE USING NOVEL 4×4 INTEGER TRANSFORM

Ranjan K. Senapati, G. Srikanth, K. Sujeeth, G. S. V. V. Balaji and S. J. Prasad
Department of Electronics and Communications, K L University, Vaddeswaram, Guntur district, India
E-Mail: ranjan.senapati@kluniversity.in

ABSTRACT

The H.264/AVC is the newest and present standard for video coding. The H.264 standard is developed from the same mixed composition as of before standards, but contains several new coding techniques that increase the efficiency and quality of the video compression. H.264 video coding standard specifies different methods of approaches for video compression, which are termed as profiles, aiming particular applications that are required. In this paper, we proposed a 4×4 novel Integer transform, which is derived from the basic 4×4 DCT kernel by applying signum function to the float DCT kernel. The proposed transform is applied to the base line profile of H. 264 for evaluation of the compressed video quality. We have considered Mean Structural Similarity (MSSIM) metric for evaluation. Extensive simulation conducted on various types of videos show that the proposed transforms outperform the H.264 by a wide margin. Further, the proposed 4×4 IDCT requires 25% less computation than H.264 standard and 45% less computation than Int. discrete Tchebichef transform (IDTT).

Keywords: H.264/AVC, IDCT, IDTT, mean structural similarity, video compression.

1. INTRODUCTION

H.264 format is the greatest and latest format for compression of the video [1], [13-14]. It is an upgraded version of MPEG-4 [2] and it is also called as the Advanced Video Coding/ (MPEG-4 part 10). The desired features is to deliver great quality for a compressed video in many of the applications such as streaming of video over internet, Standard television and High Definition Television, DVD format and other important services. Out of many profiles available in the market three profiles are extensively used. These are (i) Baseline profile (BP) which is initially intended for cheap hardware applications with less computational task, this type of profile is mainly applied for mobile and video conferencing applications. (ii) Main Profile (MP); Firstly it is proposed as the primary consumer friendly profile for storage and broadcasting applications, the prominence of discussed profile diminished when High or extended profile came into use for same applications. (iii) Extended or High Profile(XP): Mainly envisioned streaming of the video profile that has relatively capable for high compression and some other uses for sturdiness of system so that there will be no data losses and switching of server stream. A typical block diagram of a H.264 encoder is shown in Figure-1.

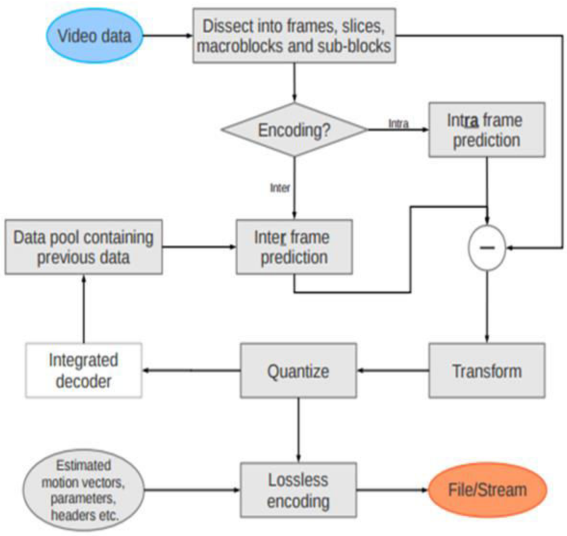

Figure-1. Block diagram of H.264 Encoder.

2. LITERATURE REVIEW AND PROBLEM FORMULATION

H.264 standard or format for video coding provides a compression gain around 2 times than earlier formats. The AVC/H.264 incorporates many innovative techniques such as hybrid predictive/transforms coding of intraframes and integer transform. The transform employed in H.264 uses integer arithmetic without any multipliers [1]. First level of transform computes only arithmetic operation which are of 16 bits in length on coefficient and factors which are used for scaling of coefficients, which produces major complexity reduction. H.264 depends heavily on prediction before transform. Preceding standard of video coding uses 8×8 DCT [3]. Many integer calculation of 8×8 DCT is reported in literature [4]-[6]. Bougel *et al.* [7] proposed 8x8 integer DCT by applying signum function to the Float DCT

ARPN Journal of Engineering and Applied Sciences

kernel. However, the transform is not satisfying the orthogonal property. This means, the same transform cannot employ for reconstruction in the starting step of decoder. To overcome this deficiency, another integer transform is proposed by Bougel *et al.* [7]. Further, this specialized feature is not valid for all orders of DCT sizes. The use of 4×4 spatial prediction in H.264 significantly reduces spatial correlation between 4×4 blocks. This motivates the choice of 4×4 transform. This minor block size leads to a noteworthy decrease in ringing objects. The integer transform used in H.264 requires add and shift operation for implementation. The scaling portion of the transform is merged into the quantizer, reducing the entire number of multiplication [8].

In this paper we proposed a novel 4×4 integer transform which is derived from the float DCT kernel after applying signum function. Interestingly, the 4×4 matrix satisfies orthogonality by just multiplying a scaling value to its transpose. That is $T^{-1}=0.25T^{t}$. The matrix only needs addition operation. No shift and multiplication operations are needed at the encoder except a scaling operation at the decoder. This leads to an overall 25% complexity reduction with respect to the int. DCT used in H.264.

3. PROPOSED 4×4 IDCT

The two dimensional DCT, of order $N \times N$, is defined as:

$$T_{DCT}(l,k) = \alpha_{lk} \sum_{c=0}^{N-1} \sum_{d=0}^{N-1} cos \frac{\pi(2c+1)l}{2N} cos \frac{\pi(2d+1)k}{2N}$$

Where,

$$\alpha_{lk} = \begin{cases} \frac{1}{\sqrt{N}}, \ for\ l,k=0 \\ \frac{2}{\sqrt{N}}, \ otherwise \end{cases} \qquad (1)$$

After applying above formulae for N=4 the matrix obtained as:

$$T_{DCT}(l,k) = \begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.6533 & 0.2706 & -0.2706 & -0.6533 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix} \qquad (2)$$

The proposed 4×4 integer DCT is obtained by applying the signum function operator to the elements of DCT obtained from (1). Therefore, it is given as:

Proposed IDCT= $sign\{T_{DCT}(l,k)\}$ (3)

where sign($\cdot$) is the signum function defined as:

$$sign\{z\}=\begin{cases} +1\ for\ z>0 \\ 0\ for\ z=0 \\ -1\ for\ z<0 \end{cases} \qquad (4)$$

Several advantages of Proposed IDCT exist. These are:

- Each and every element is either +1or -1.
- There is no operation on multiplication as well as transcendental expression.
- Contrasting with Walsh Hadamard Transform ('WHT') [7], and SDFT [11], Signed DCT [10] is essentially not be an exact integer or a power of 2.
- Proposed IDCT preserves the periodicity and spectral configuration of its initial version of DCT and preserves good energy compaction and de-correlation features.

It has been confirmed that only 10% spectral modules of Proposed IDCT has 80% of the entire power of signal related with 87% power of signal in Cosine Transform of discrete type.

Applying (2) in (1) the 4×4 Proposed IDCT transform matrix is given by:

$$Proposed\ IDCT = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix} \qquad (5)$$

It can be observed Proposed IDCT matrix fulfils the orthogonality property, i.e. $IDCT^{-1} = 0.25 * IDCT^{t}$, where 't' means the transpose process. Value 0.25 is a constant factor that been taken to satisfy the orthogonal property. So we can use the identical matrix for encoding and decoding of image.

Let R be a 4×4 block of image data and S be its corresponding matrix in transformed domain. Then the forward transform operation will be

$$S= (IDCT)\ R\ (IDCT)^{t} \qquad (6)$$

Since IDCTis orthogonal, we can reconstruct the image by using the reverse transform given as:

$$R= (IDCT)^{t}\ S\ (IDCT) \qquad (7)$$

4. COMPARISON WITH OTHER INTEGER TRANSFORMS

A) Integer Discrete Tchebichef Transform (IDTT)

It exhibits interesting properties, such as high energy compaction and optimal decorrelation [12]. It has better performance in videos that has lot of details compared when compared to IDCT. IDTT can be written in matrix form as

$$Y=\Gamma \mathbf{X} \Gamma' \qquad (8)$$

$$\Gamma = \begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ -0.6708 & -0.2236 & 0.2236 & -0.6708 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ -0.2236 & 0.6708 & -0.6708 & 0.2236 \end{bmatrix} \qquad (9)$$

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

The equation derived from the transform in (9) is now factorized to differentiate the integer and float numbers.
Let the equation be

$$q(k) = \frac{p(k)}{d_k}, \qquad k = 0 \ldots\ldots\ldots 3$$

Where the above characters are defined as

$$p(k) = \begin{cases} k!\,k & for\ k = 1 \\ k!\,(k+!) & for\ k = 0,2 \\ k!\,(N-k) & for\ k = 3 \end{cases}$$

Let Q be a diagonal matrix with q(k) as the diagonal element of the kth row. Therefore

$$Q=\begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.2236 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.2236 \end{bmatrix} \tag{10}$$

Ґ=Q $\hat{\Gamma}$ (11)

$\hat{\Gamma}$ = Q^{-1}Ґ=Q$_i$Ґ (12)

Forward transform Y=x Ґ. Hence, after matrix multiplication:

Y=

$$\begin{bmatrix} q_{00}..+..q_{30} & q_{01}..+..q_{31} & q_{02}..+..q_{32} & q_{03}..+..q_{33} \\ 3(q_{00}..+..q_{30}) & q_{01}..+..q_{31} & -(q_{02}..+..q_{32}) & -3(q_{03}..+..q_{33}) \\ q_{00}..+..q_{30} & -(q_{01}..+..q_{31}) & -(q_{02}..+..q_{32}) & q_{03}..+..q_{33} \\ q_{00}..+..q_{30} & -3(q_{01}..+..q_{31}) & 3(q_{02}..+..q_{32}) & -(q_{03}..+..q_{33}) \end{bmatrix}$$

(15)

Total number of computations from (15): 60 additions and 16shift operations.

where,Q$_i$ is the diagonal matrix which contains the reciprocal of the diagonal elements of Q as it its diagonal elements are given by

$$Q_i = \begin{bmatrix} 1/q_0 & 1 & 1 & 1 \\ 1 & 1/q_1 & -1 & -1 \\ 1 & -1 & 1/q_2 & 1 \\ 1 & -1 & 1 & 1/q_3 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 4.4721 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4.4721 \end{bmatrix} \tag{13}$$

Now to derive the $\hat{\Gamma}$, substitute in (8)

$$\hat{\Gamma} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 4.4721 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4.4721 \end{bmatrix} * \begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ -0.6708 & -0.2236 & 0.2236 & -0.6708 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ -0.2236 & 0.6708 & -0.6708 & 0.2236 \end{bmatrix}$$

After matrix multiplication and simplifying it we get the following value

$$\hat{\Gamma} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -3 & -1 & -1 & 3 \\ 1 & -1 & -1 & 1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \tag{14}$$

Here the Ґ matrix represents the kernel matrix and x be example matrix. These matrices are considered so as to calculate the complexity.

a) Computational complexity of 1D-IDTT

From (14) we get that,

$$Ґ=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 1 & -1 & -3 \\ 1 & -1 & -1 & 1 \\ 1 & -3 & 3 & -1 \end{bmatrix} x=\begin{bmatrix} q_{00} & q_{01} & q_{02} & q_{03} \\ q_{10} & q_{11} & q_{12} & q_{13} \\ q_{20} & q_{21} & q_{22} & q_{23} \\ q_{30} & q_{31} & q_{32} & q_{33} \end{bmatrix}$$

Forward transform Y=x Ґ. Hence,

b) Computation by Proposed IDCT

Here the $T_{Int.\ DCT}$ matrix represents the kernel matrix and x be example matrix. These matrices are considered so as to calculate the calculation complexity

$$IDCT=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix},$$

$$x=\begin{bmatrix} q_{00} & q_{01} & q_{02} & q_{03} \\ q_{10} & q_{11} & q_{12} & q_{13} \\ q_{20} & q_{21} & q_{22} & q_{23} \\ q_{30} & q_{31} & q_{32} & q_{33} \end{bmatrix}$$. Y=x (IDCT). Hence,

ARPN Journal of Engineering and Applied Sciences

$$Y=\begin{bmatrix} q_{00}..+..q_{30} & q_{01}..+..q_{31} & q_{02}..+..q_{32} & q_{03}..+..q_{33} \\ 2(q_{00}..+..q_{30}) & q_{01}..+..q_{31} & -(q_{02}..+..q_{32}) & -2(q_{03}..+..q_{33}) \\ q_{00}..+..q_{30} & -(q_{01}..+..q_{31}) & -(q_{02}..+..q_{32}) & q_{03}..+..q_{33} \\ q_{00}..+..q_{30} & -2(q_{01}..+..q_{31}) & 2(q_{02}..+..q_{32}) & -(q_{03}..+..q_{33}) \end{bmatrix} \qquad (16)$$

Total number of computations from (16): 48 additions only

c) Computation by Int. DCT used in H.264

Here the T matrix represents the kernel matrix and x be example matrix (shown to the right side). These matrices are considered so as to calculate the calculation complexity.

Y=xT

$$T=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \qquad x=\begin{bmatrix} q_{00} & q_{01} & q_{02} & q_{03} \\ q_{10} & q_{11} & q_{12} & q_{13} \\ q_{20} & q_{21} & q_{22} & q_{23} \\ q_{30} & q_{31} & q_{32} & q_{33} \end{bmatrix}$$

$$Y=\begin{bmatrix} q_{00}..+..q_{30} & q_{01}..+..q_{31} & q_{02}..+..q_{32} & q_{03}..+..q_{33} \\ 2(q_{00}..+..q_{30}) & q_{01}..+..q_{31} & -(q_{02}..+..q_{32}) & -2(q_{03}..+..q_{33}) \\ q_{00}..+..q_{30} & -(q_{01}..+..q_{31}) & -(q_{02}..+..q_{32}) & q_{03}..+..q_{33} \\ q_{00}..+..q_{30} & -2(q_{01}..+..q_{31}) & 2(q_{02}..+..q_{32}) & -(q_{03}..+..q_{33}) \end{bmatrix} \qquad (17)$$

Total number of computations from (17): 48 additions and 16 shift operations. Hence by summarizing we get that,

- The computational complexity of DTT is 18.75% more than IDCT in H.264.
- The computational complexity of Proposed Int. DCT is 25% less than IDCT used in H.264

5. SIMULATION RESULTS AND DISCUSSIONS

To predict the perceived quality of digital television and pictures, as well as other kinds of digital images and videos a method was introduced which is called Structural similarity index (SSIM). SSIM is used for measuring similarity between the two images. It is given by the following equation

$$SSIM(u,v)=\frac{(2\mu_u\mu_v+c1)(2\sigma_{uv}+c2)}{(\mu_u^2+\mu_v^2+c1)(\sigma_u^2+\sigma_v^2+c2)}$$

where μ_u is the average of u, μ_v is the average of v, σ_u^2 is the variance of u, σ_v^2 is the variance of v, σ_{uv} is the co-variance of u and v.

Mean value is given by

$$\mathrm{MSSIM}(u',v')=\frac{1}{M}\sum_{j=1}^{M}\mathrm{SSIM}(u,v) \qquad (18)$$

M is the number of windows of image

$c1=(k_1L)^2$, $c2=(k_2L)^2$ are the two variables to stabilize the division with the weak denominator; L is the dynamic range of the pixel-values. $k_1=0.01$ and $k_2=0.03$ are given by default. The performance of the algorithm is simulated on a Window XP platform having 1GB RAM capacity. The simulation plots for 'Xylophone video' and 'bouncing ball' for 50 frames along with visual qualities comparisons for different quality parameters (QP) are illustrated below.

A. Xylophone video (192x144)

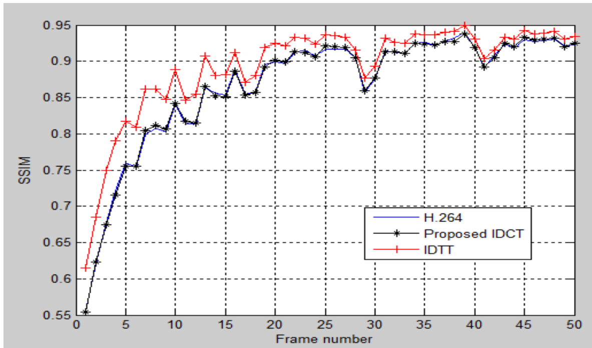

Figure-2. MSSIM comparison on H.264 baseline profile using IDTT, H.264 IDCT, and Proposed IDCT for QP=10.

Visual quality comparison

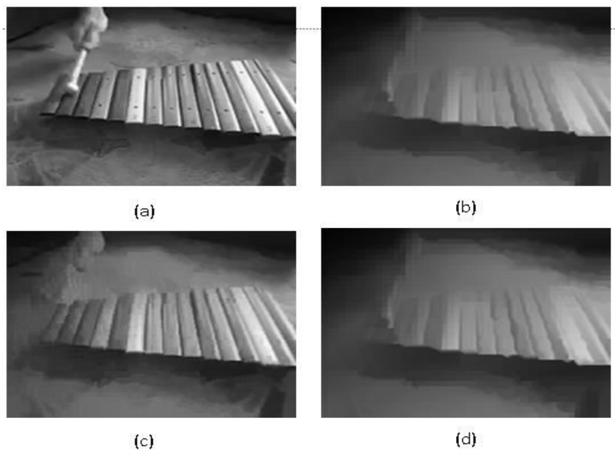

Figure-3. Original video frame no.1 shown in (a) is compressed at QP=10 using (b) H.264 (c) IDTT (d) Proposed IDCT.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

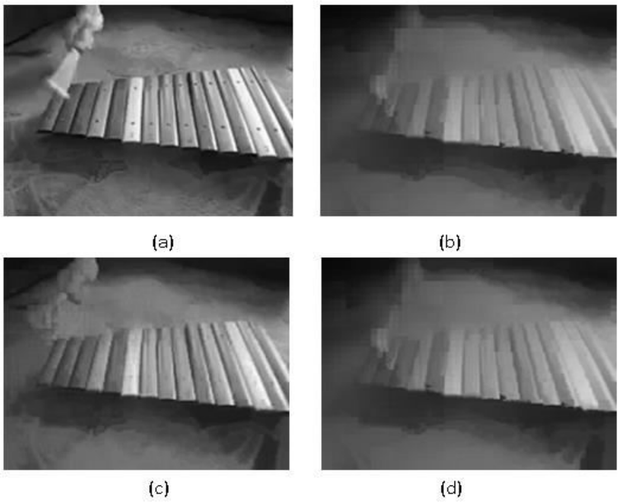

Figure-4. Original video frame no.2 shown in (a) compressed at QP=10 using (b) H.264 (c) IDTT (d) Proposed IDCT.

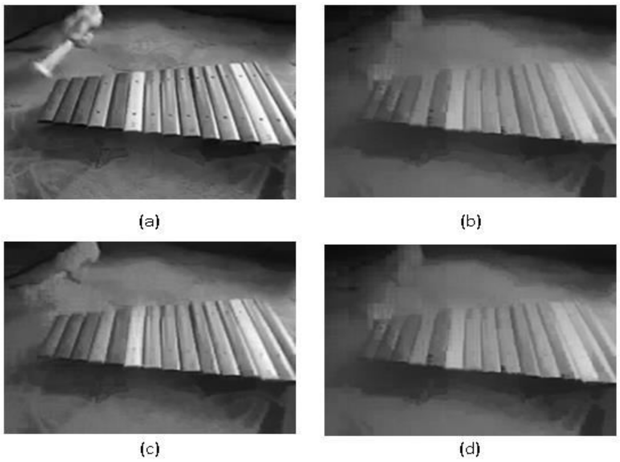

Figure-5. Original video frame no.3 shown in (a) is compressed at QP=10 using (b) IDTT (c) H.264 IDCT (d) Proposed IDCT.

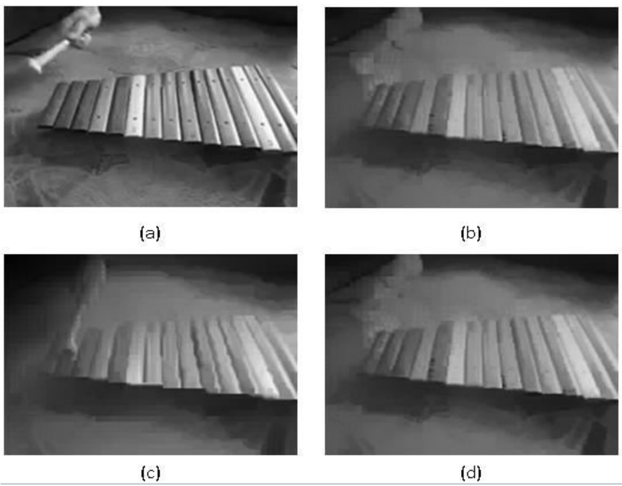

Figure-6. Original video frame no.4 shown in (a) compressed at QP=10 using (b) H.264 (c) IDTT (d) Proposed IDCT.

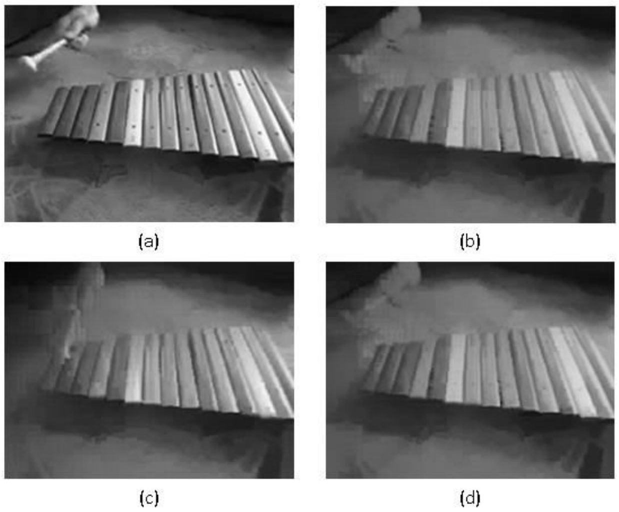

Figure-7. Original video frame no.5 shown in (a) compressed at QP=10 using (b) H.264 (c) IDTT (d) Proposed IDCT.

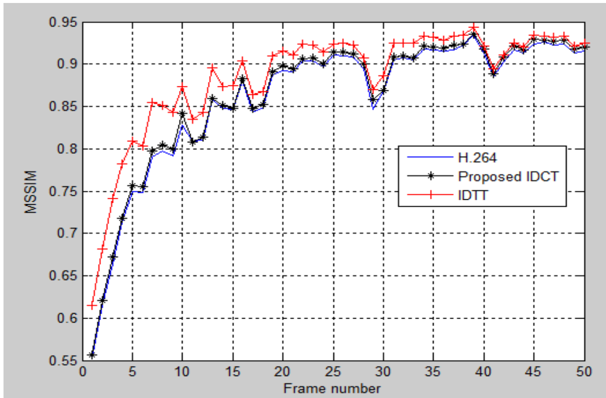

Figure-8. MSSIM comparison on H.264 baseline profile using IDTT, H.264 IDCT, and Proposed IDCT for QP=15.

Visual quality comparison

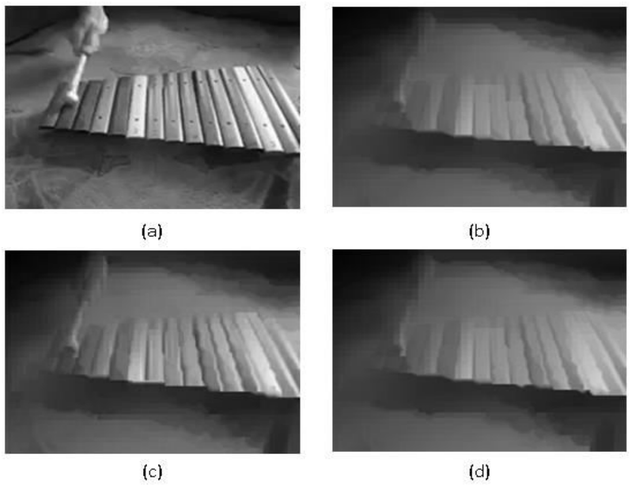

Figure-9. Original video frame no.1 shown in (a) is compressed at QP=15 using (b) H.264 (c) IDTT (d) Proposed IDCT.

VOL. 11, NO. 17, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

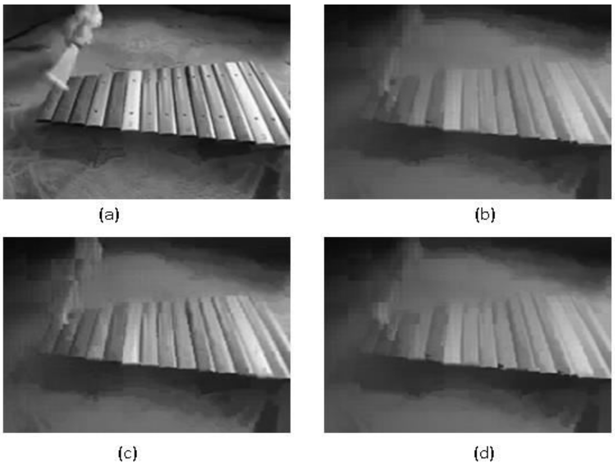

Figure-10. Original video frame no.2 shown in (a) is compressed at QP=15 using (b) H.264 (c) IDTT (d) Proposed IDCT.

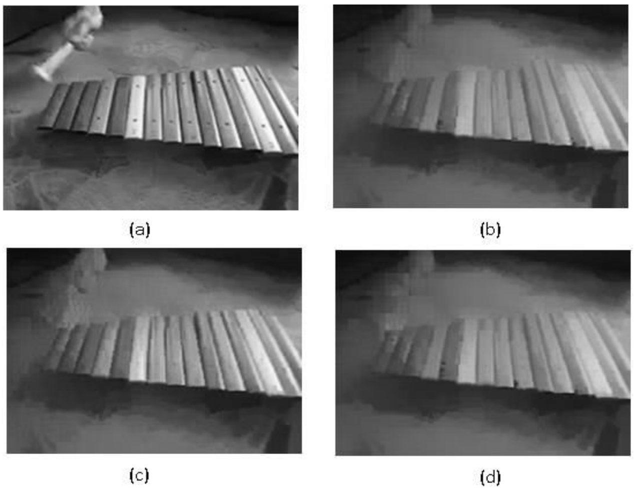

Figure-11. Original video frame no.3 shown in (a) is compressed at QP=15 using (b) H.264 (c) IDTT (d) Proposed IDCT.

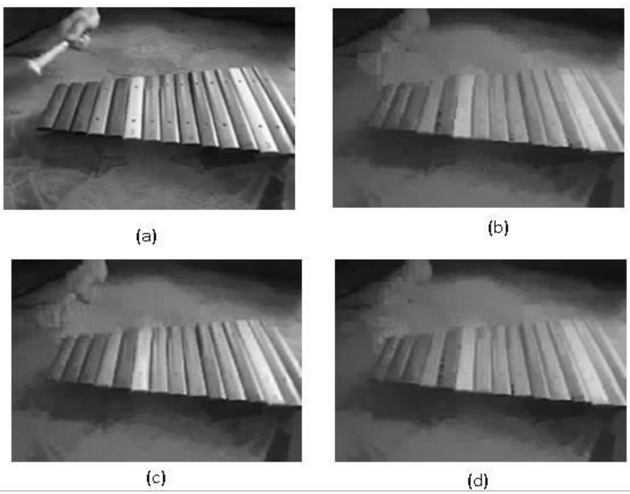

Figure-12. Original video frame no.4 shown in (a) is compressed at QP=15 using (b) H.264 (c) IDTT (d) Proposed IDCT.

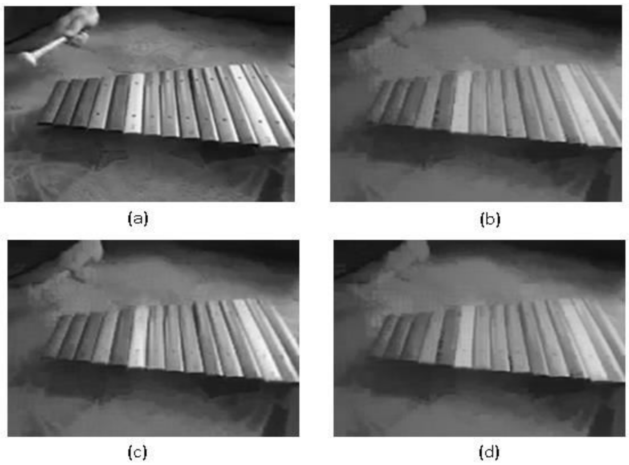

Figure-13. Original video frame no.5 shown in (a) is compressed at QP=15 using (b) H.264 (c) IDTT (d) Proposed IDCT.

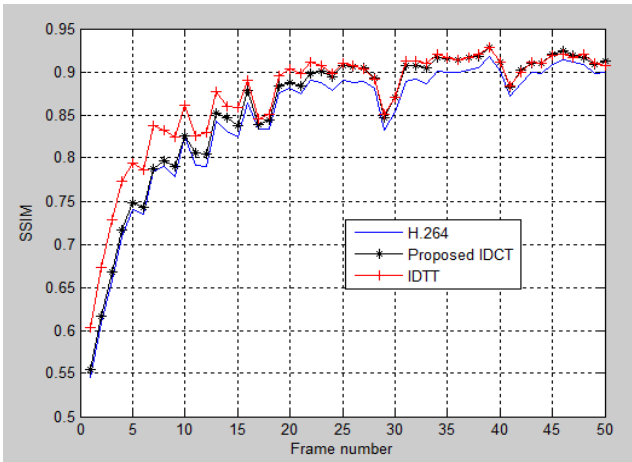

Figure-14. MSSIM comparison on H.264 baseline profile using IDTT, H.264 IDCT, and Proposed IDCT for QP=20.

Visual quality comparison

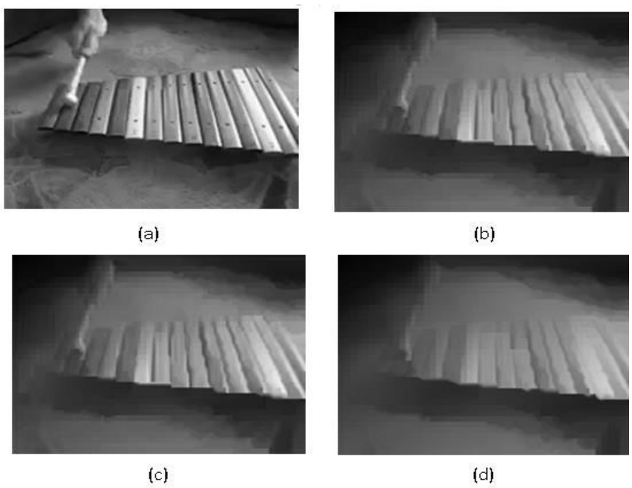

Figure-15. Original video frame no.1 shown in (a) is compressed at QP=20 using (b) H.264 (c) IDTT (d) Proposed IDCT.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

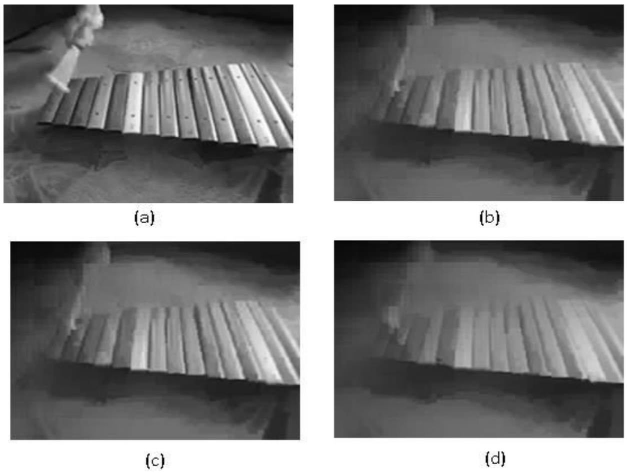

Figure-16. Original video frame no.2 shown in (a) is compressed at QP=20 using (b) H.264 (c) IDTT (d) Proposed IDCT.

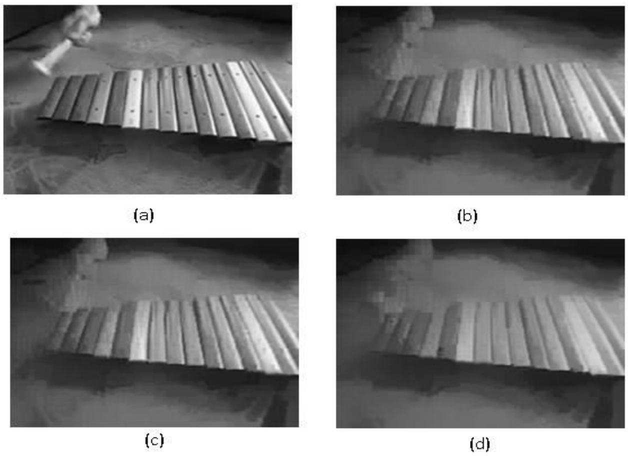

Figure-17. Original video frame no.3 shown in (a) is compressed at QP=20 using (b) H.264 (c) IDTT (d) Proposed IDCT.

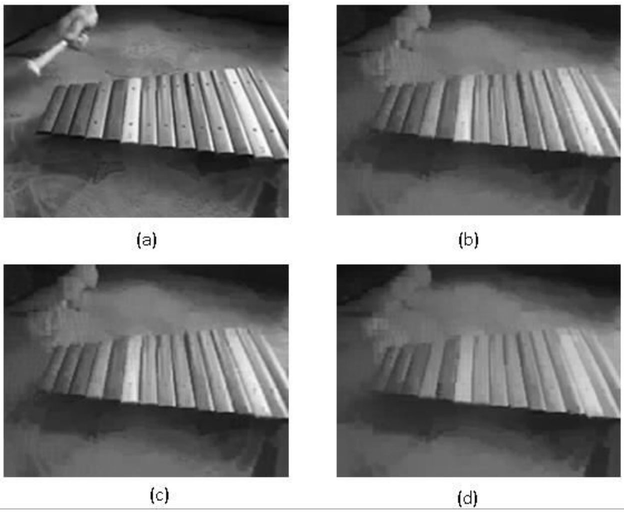

Figure-18. Original video frame no.4 shown in (a) is compressed at QP=20 using (b) H.264(c IDTT (d) Proposed IDCT.

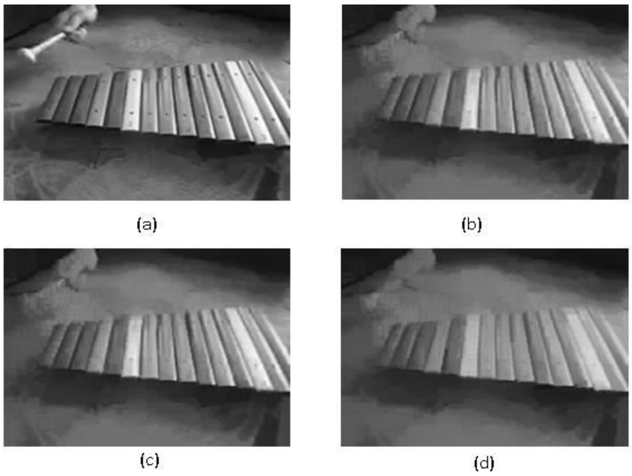

Figure-19. Original video frame no.5 shown in (a) is compressed at QP=20 using (b) H.264 (c) IDTT (d) Proposed IDCT.

B. Bouncing balls video (192x144)

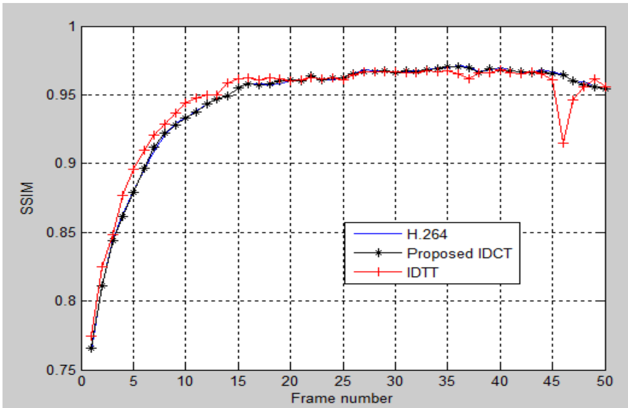

Figure-20. MSSIM comparison on H.264 baseline profile using IDTT, H.264 IDCT, and Proposed IDCT for QP=10.

Visual quality comparison

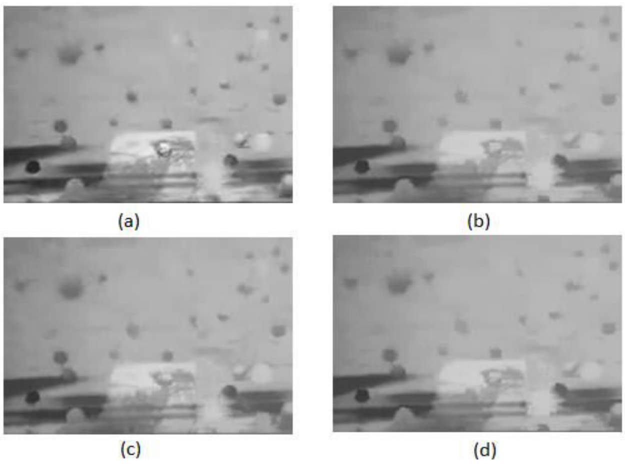

Figure-21. Original video frame no.11 shown in (a) is compressed at QP=10 using (b) H.264 (c) IDTT(d) Proposed IDCT.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

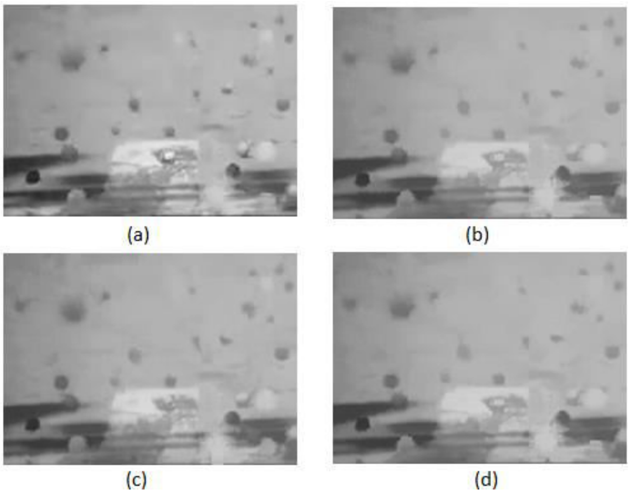

Figure-22. Original video frame no.12 shown in (a) is compressed at QP=10 using (b) H.264 (c) IDTT (d) Proposed IDCT.

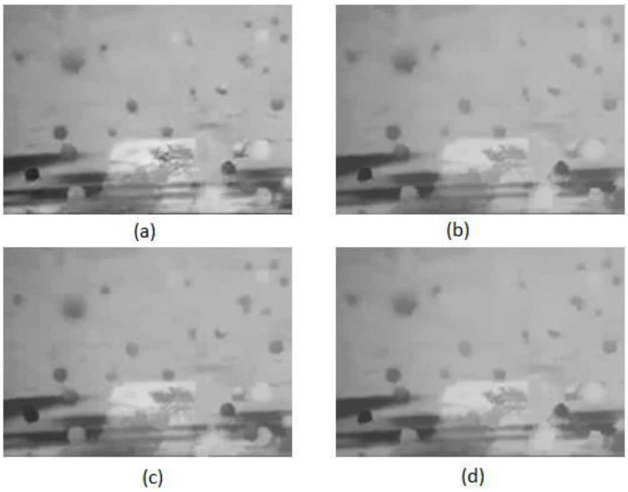

Figure-23. Original video frame no.13 shown in (a) is compressed at QP=10 using (b) H.264 (c) IDTT (d) Proposed IDCT.

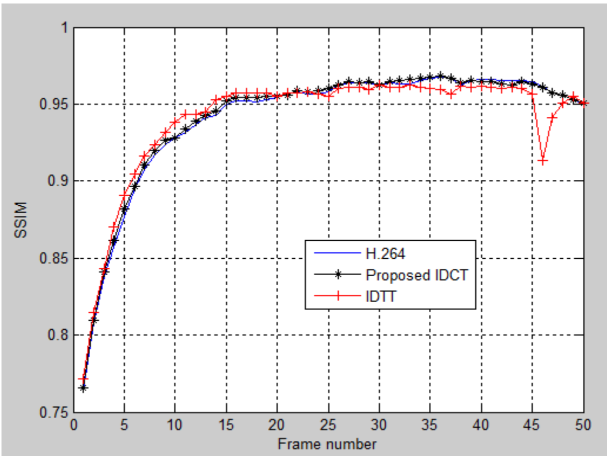

Figure-24. MSSIM comparison on H.264 baseline profile using IDTT, H.264 IDCT, and Proposed IDCT for QP=15.

Visual quality comparison

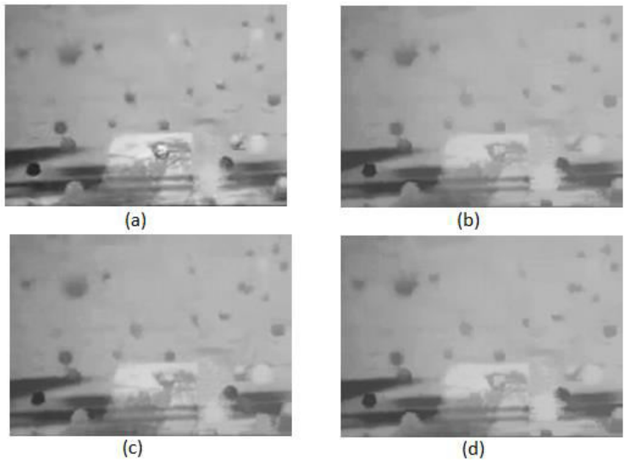

Figure-25. Original video frame no.11 shown in (a) is compressed at QP=15 using (b) H.264 (c) IDTT (d) Proposed IDCT.

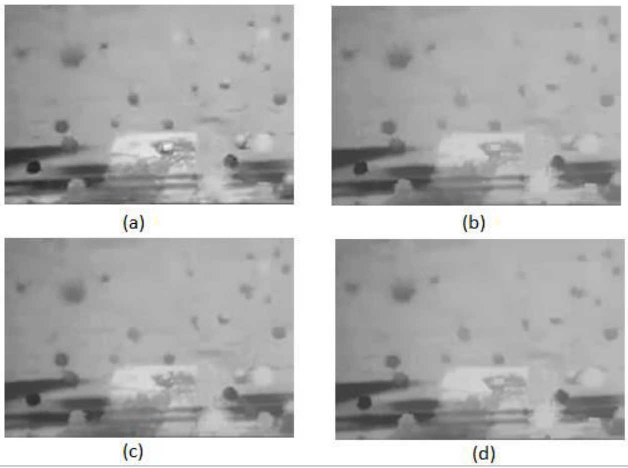

Figure-26. Original video frame no.12 shown in (a) is compressed at QP=15 using (b) H.264 (c) IDTT (d) Proposed IDCT.

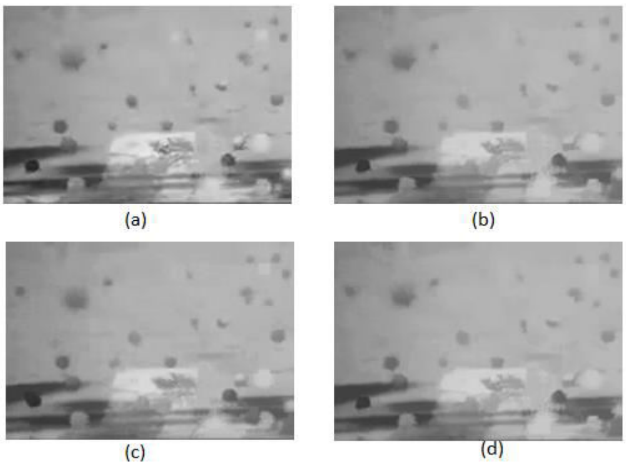

Figure-27. Original video frame no.13 shown in (a) is compressed at QP=15 using (b) H.264 (c) IDTT (d) Proposed IDCT.

VOL. 11, NO. 17, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

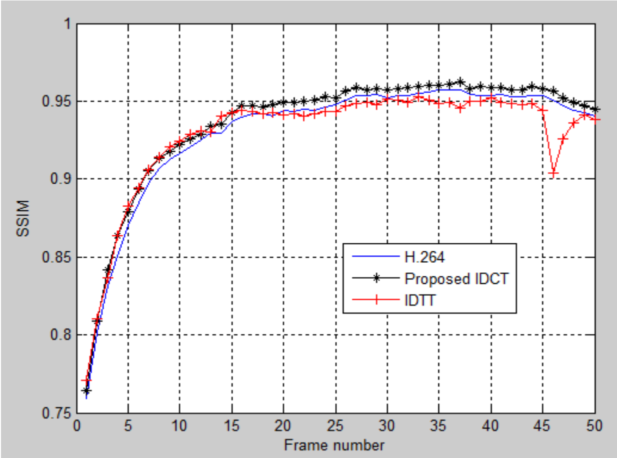

Figure-28. MSSIM comparison on H.264 baseline profile using IDTT, H.264 IDCT, and Proposed IDCT for QP=20.

Visual quality comparison

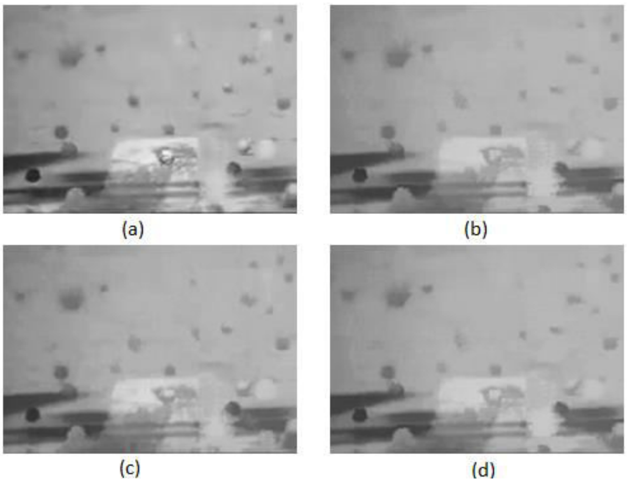

Figure-29. Original video frame no.11 shown in (a) is compressed at QP=20 using (b) H.264 (c) IDTT (d) Proposed IDCT.

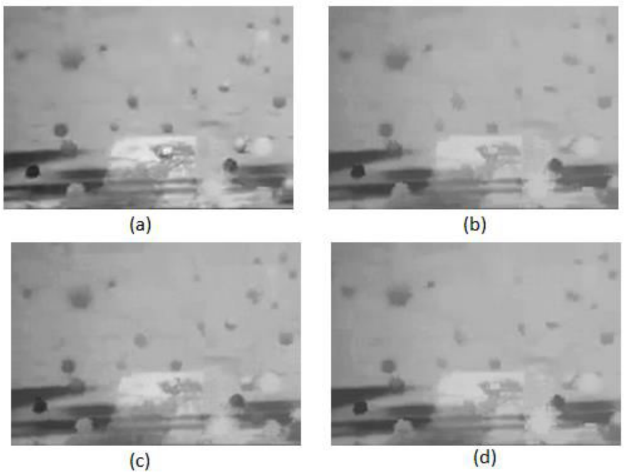

Figure-30. Original video frame no.12 shown in (a) is compressed at QP=20 using (b) H.264 (c) IDTT (d) Proposed IDCT.

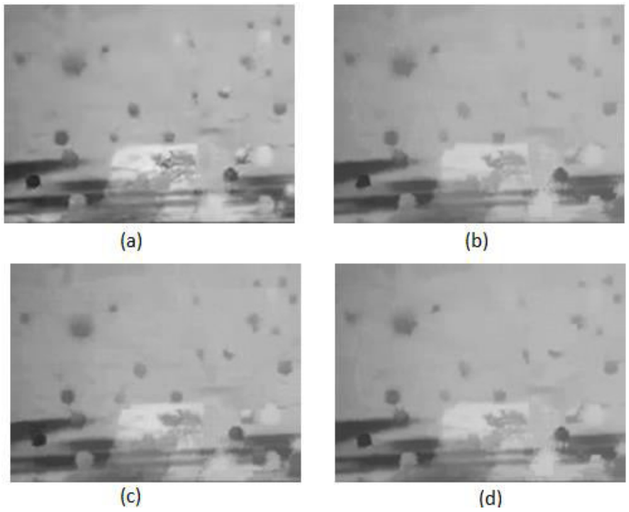

Figure-31. Original video frame no.13 shown in (a) is compressed at QP=20 using (b) H.264 (c) IDTT (d) Proposed IDCT.

6. DISCUSSIONS

From the above computational complexity comparison we have proved that the proposed IDCT need 25% less computation than Int. DCT in H.264 and 44% less than IDTT. However, the MSSIM performance of IDTT is superior for all cases at a cost of computational complexity. Simulation is carried over 50 frames. We have displayed the Figures 3-7, 9-13, 15-19 for first 5 frames for different QP. Simulation is carried over 50 frames for other video also. We have displayed the Figures 21-23, 25-27, 29-31 for frame no. 11, 12 and 13. At lower quality parameter (i.e., less compression) The MSSIM performance of proposed IDCT is slightly better in an average, which is demonstrated from Figure-2, 8 and 14 as well as Figure-20, 24, and 28. At higher QP, the proposed IDCT exhibit similar performance with H.264. As the luminance frames are more sensitive to chrominance frames, the simulations are performed on luminance frames to test the performance of the proposed IDCT.

7. CONCLUSIONS

We have proposed a new low complexity IDCT and evaluated the performance on H.264 baseline profile for different quality parameters. The prosed IDCT has significantly reduced computation than Int. DCT in H.264 (~25%) and IDTT (~45%). The video subjective quality is comparable to H.264. Therefore, our transform can be used in complexity constraint applications like surveillance cameras and remote sensor nodes (i.e., wireless sensor nodes). The future research direction is to investigate the performance in H.264 extended profile.

REFERENCES

[1] Malver H.S., Hallapuro A., Karczewicz M., Krrofsky L. 2003. Low complexity transform and quantization in H.264/AVC. IEEE Trans. on Circuits and Systems. 13(7): 598-603.

[2] Heiko Schwarz, Detlev Marpe and Thomas Wiegand. Overview of the scalable H.264/MPEG4-AVC extension. Fraunhofer Institute for Telecommunications.

[3] N. Ahmed, T. Natarajan and K. R. Rao. 1974. Discrete cosine transfom. IEEE Trans. Comput. C-23(1): 90-93.

[4] Dimitrov V., Wahid, K. and Jullien G. 2004. Multiplication free 8 × 8 2D DCT architecture using algebraic integer encoding. Electronics Lett. 40(9): 1310-1311.

[5] Lengwehasatit K. and Ortega A. 2004. Scalable variable complexities approximate forward DCT. IEEE Trans. Circuits Syst. Video Technology. 14(11): 1236-1248.

[6] Tran T.D. 2004. The BinDCT: Fast multiplierless approximation of the DCT. IEEE Signal Process. Lett. 7(6): 141-144.

[7] Bouguezel S., Ahmad M.O. and Swamy M.N.S. 2008. A low complexity 8 × 8 transform for image compression. IEEE Electronic Letter. 44(21): 1249-1250.

[8] Sujatha I, Meher P. K, Swamy M.N.S. 2008. Discrete Tchebichef Transform-A Fast 4x4 Algorithm and its Application in Image/Video Compression. IEEE Int. Symposium on Circuits and Systems (ISCAS), Seattle (Washington). pp. 260-263.

[9] R. Mukundan, S. H. Ong and P. A. Lee. 2001. Image analysis by Tchebichef moments. IEEE Trans. Image Process. 10(9): 1357-1364.

[10] Beauchamp K.G. 1984. Application of Walsh and Related Functions, Academic Press, New York.

[11] Haweel T.I. and Alhasan A.M. 1995. A simplified square wave transform for signal processing. Contemp. Math. 190: 265-272.

[12] Ranjan K. Senapati, Umesh C. Pati, and Kamala K. Mahapatra. 2011. A Image Compression Using Discrete Tchebichef Transform Algorithm. Int. conf. on CNC, pp. 1-5.

[13] Jun Zhang, Yuan Zhang, S. Chen. 2012. Complexity Constrained H.264 Video Coding in Wireless Environment. 8th Int. Conf. on WiCOM. pp. 1-4.

[14] Humberto Osslan, Sanchez Casas, K.R. Rao. 2014. The H.264 Video Coding Standard. IEEE Potentials. 33(2): 32-38.