# dFuse: An Optimized Compression Algorithm for DICOM-format Image Archive

**Suresh Jaganathan** *whosuresh@gmail.com*
*Assistant Professor*
*Department of Computer Science and Engineering*
*Sri Sivasubramania Nadar College of Engineering,*
*Chennai, Tamilnadu, India*

**Geetha Manjusha M B** *geethamtech0911@gmail.com*
*PG Scholar*
*Department of Computer Science and Engineering*
*Sri Sivasubramania Nadar College of Engineering,*
*Chennai, Tamilnadu, India*

## Abstract

Medical images are useful for knowing the details of the human body for health science or remedial reasons. DICOM is structured as a multi-part document in order to facilitate extension of these images. Additionally, DICOM defined information objects are not only for images but also for patients, studies, reports, and other data groupings. More information details in DICOM, resulted in large size, and transferring or communicating these files took lots of time. To solve this, files can be compressed and transferred. Efficient compression solutions are available and they are becoming more critical with the recent intensive growth of data and medical imaging. In order to receive the original and less sized image, we need effective compression algorithm. There are different algorithms for compression such as DCT, Haar, Daubechies which has its roots in cosine and wavelet transforms. In this paper, we propose a new compression algorithm called “dFuse”. It uses cosine based three dimensional transform to compress the DICOM files. We use the following parameters to check the efficiency of the proposed algorithm, they are i) file size, ii) PSNR, iii) compression percentage and iv) compression ratio. From the experimental results obtained, the proposed algorithm works well for compressing medical images.

**Keywords:** Medical Image, Image Compression, DICOM, Wavelets, Cosine Transforms

## 1. INTRODUCTION

Medical imaging is a regulation within the medical arena which makes use of technology to acquire images of inside the human body. These images are utilized in analytics, as training tools, and in regular healthcare. This sometimes specified as diagnostic imaging, because it is often helps doctors to diagnose easily. One kind of medical images is DICOM. DICOM not only stores the image, but also some details such as patient name, patient-id and date scanned etc. DICOM varies from other data formats, as it contain groups of information together into a data set. A DICOM data object comprises of a number of attributes, comprising items such as ID, DOB, date, name etc. Also one single attribute comprising the image pixel data. In order to maintain more details, file size increases, because of this transferring the file takes a lot of time. To avoid this difficulty, efficient compression techniques [1] are needed.

Image compression focuses [2] on the difficulty of decreasing the quantity of information that are needed to signify an image. This is used to decrease the image storage and transmission requirements. The inverse process of the image compression is called decompression and it is applied to the compressed data to get back the reconstructed image. The goal of compression is to decrease the number of bits as much as possible, while maintaining the resolution and the visual excellence of the reconstructed image as close to the original image. Image compression has many benefits. Image compression [3] provides a potential cost savings coupled with sending a smaller amount of data over switched telephone. It not only reduces storage requirements but also overall execution time. It reduces the probability of transmission errors since fewer bits are transferred. It offers a level of security against illicit monitoring. There are different compression methods available in medical imaging such as cosine [4], wavelet transforms [5].

The rest of the paper is organized as follows. Section 2 brief introduction to medical images, Section 3 describes the definitions of various compression algorithms. Section 4 narrates the proposed architecture and explains the algorithm proposed. Section 5 discusses the experimental results obtained and finally Section 6 concludes the paper.

## 2. MEDICAL IMAGES

Medical images [6, 7] are used to provide a photograph of the inner side of the body as clear as it is. These images are helpful to recognize unusual things in inner parts of the body, such as tumors, blood vessels, broken bones, and so on. The most famous types of diagnostic images are the x-ray images that use radiation to take a stationary image of a specific area of the body. Doctors can know the medical outcomes of a calculated tomography scan that can be retrieved via a commercial computer. Behind the scenes, gigantic amount of data is compressed, so that doctor observes it on a computer screen. Computed tomography scans, along with magnetic resonance imaging and positron emission tomography scans form substantial amounts of data. Data is not stockpiled on a typical hard drive and the gap required to accumulate these images in a clinical background would engage in a complete wing of the hospital and also with current electronic medical records custody laws.

The DICOM Standards Committee survives to produce and sustain international standards for communication of biomedical diagnostics and curative information in restraints that utilize digital images and allied data. The objectives of DICOM are to accomplish companionability and to increase workflow effectiveness. DICOM is used by every medical profession that exploits images within the healthcare industry. These include dentistry, mammography, ophthalmology, endoscopy, orthopedics, radiology, cardiology, surgery, pediatrics, radiation therapy, pathology etc. DICOM also tackles the assimilation of information created by these various areas of expertise applications in the patient's e-Medical Record. This defines the network and media swapping services permitting storage and retrieval to these DICOM objects for these record systems. The compression of DICOM files has an enormous value. Compression of an image can be a solitary image or set of images. The DICOM standard has been very disinclined to accept demise algorithms in medical practice. However, the diagnostic information created by hospitals has statistically enhanced and a compression technique is desired that outcomes with larger data diminutions and so transmission speed.

## 3. DEFINITIONS

### 3.1. Cosine transforms

Cosine transform helps to detach an image into parts of differing significance based on frequency, such as higher frequency part and lower frequency part. It renovates a signal or image from the spatial domain to the frequency domain. There are different types of cosine transforms [8] such as DCT, 2DDCT and 3DDCT [9].

DCT is mainly used for changing a signal into elementary frequency components and extensively used in image compression. DCT is described as the product of a vector. It consists of $n \ x \ n$ orthogonal matrix whose rows are the basis vectors. The matrix must be orthogonal and each basis vector relates to a sinusoid of a definite frequency. The general equation for DCT is represented as:

$$F(u) = (2/N)^{\frac{1}{2}} \sum_{0}^{N-1} A(i).\cos\left[\frac{\pi.u}{2.N}(2i+1)\right] f(i) \text{ where } A(i) = \begin{cases} \frac{1}{\sqrt{2}} & if \ \epsilon = 0 \\ 1 & Otherwise \end{cases}$$

$F$ is represented as a linear combination of the basis vector. These coefficients which we get are the elements of the inverse transform, it might be observed as mirroring the amount of each frequency, there in the input $F$. The one-dimensional DCT is useful for processing only one-dimensional signals such as speech waveforms. For analysis of two-dimensional (2D) signals such as images, 2D version of the DCT is used.

2DDCT is two dimensional version of DCT. DCT is applied vertically and then horizontally. 2DDCT uses a domain data which is calculated using the below formula $\alpha = \sqrt{256/N_c N_r}$ where $Nc, Nr$ is the number of columns and rows respectively. The general equation for 2D-DCT is represented as

$$X\ (V,W)\ = \sum_{n=0}^{N_c-1} \sum_{p=0}^{N_r-1} \alpha * C_V^{2n+1}\ C_W^{2p+1},$$

where $\alpha$ is the time domain data and $C_W^{2p+1}$ are all Cosine coefficients and defined as

$$C_W^{2p+1} = \cos[((2p\ +\ 1)\ * w * 3.14)/2Nr.$$

The same definition is applied on $C_V^{2n+1}$. There are some demerits in 2DDCT.Spatial correlation of the pixels within the single 2-D block is measured and the adjacent block values are neglected. It fails to perform proficiently for binary images characterized by huge periods of invariable amplitude followed by brief periods of sharp transitions.

### 3.2. Wavelet Transforms

Different algorithms depending on wavelets have been exposed to image compression [10, 11]. Separating the smooth variations and details of the image can be done by decomposition of the image using wavelet transform. The similar extension details were being absolute for bi-orthogonal wavelets particularly for low frequency images. There are different types of wavelet transforms such as Haar and Daubechies, DWT, DTCWT.

The Haar wavelet is a certain sequence of rescaled "square-shaped" functions which together form a wavelet family or basis. Wavelet analysis is similar to Fourier analysis in that it allows a target function over an interval to be represented in terms of an orthonormal function basis.
The Haar wavelet's mother wavelet function ψ (*t*) can be described as

$$\varphi(t) = \begin{cases} 1 & 0 \le t < {}^{1}/_{2} \\ -1 & 1/2 \le t < 1 \\ 0 & \text{Otherwise} \end{cases}$$

Its scaling function φ(*t*) can be described as

$$\emptyset(t) = \begin{cases} 1\ 0 \le t < 1 \\ 0\ \text{Otherwise} \end{cases}$$

Haar transform is discontinuous and does not approximate continuous signals very well. Also Haar eliminates the noise to some extent, but also it disturbs the rest of the signal. Daubechies overcomes these problems by adopting more scaling functions. Hence Daubechies produces accurate averages and differences. This makes a tremendous improvement in the capabilities of transforms. Daubechies transform consists of four scaling function coefficients and wavelets represented as shown below:

$$p_0 = \frac{1+\sqrt{3}}{4\sqrt{2}},\ p_1 = \frac{3+\sqrt{3}}{4\sqrt{2}},\ p_2\ = \frac{3-\sqrt{3}}{4\sqrt{2}},\ p_3\ = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

In each step, the wavelet transform pertain the scaling function to the input data. If the original data has $N$ values, then the scaling function applied in the wavelet transform is $N/2$ values. These values are stored in lower half of the $N$ element data input vector. The wavelet transform function coefficient values are calculated using the functions shown below:

$$q_0 = p_3,\ q_1 = -p_2\ , q_2 = p_1\ , q_3 = -p_0$$

The scaling function values are stored in upper half of the $N$ element data input vector. The wavelet and scaling functions are computed by considering the inner product of the coefficients and four data values. The scaling function is represented as shown below:

$$a_i = p_0 s_{2i} + p_1 s_{2i+1} + p_2 s_{2i+2} + p_0 s_{2i+3}$$

$$a\,[i] = p_0\, s[2i] + p_1\, s[2i+1] + p_2\, s[2i+2] + p_3\, s[2i+3]$$

The wavelet function is represented as shown below:

$$c_i = q_0 s_{2i} + q_1 s_{2i+1} + q_2 s_{2i+2} + q_0 s_{2i+3}$$

$$c[i] = q_0 s[2i] + q_1 s[2i+1] + q_2 s[2i+2] + q_3 s[2i+3]$$

## 4. ARCHITECTURE OF dFUSE

Figure 1 shows the architecture of the proposed algorithm “dFuse”. This technique is entitled as “dFuse” because “Fuse” means combination. Proposed algorithm combines the power of both spatial and temporal representations to compress the DICOM files. For the sake of compressing DICOM files, this fuse is used. Hence the technique is called “dFuse”. DICOM standard is the most required images in the field of medical imaging. DICOM files not only contain the images but also the details of the patients and other details. Hence there is a need of large amount of storage for the maintaining of these records. The intricacy in storing these images is they require additional space and memory. To avoid this difficulty there is a need of tumbling the space without damaging the quality of images. Many compression techniques are worked out to solve this problem [12, 13]. dFuse technique works well by avoiding the complexities in other algorithms. dFuse reduces the i) computational complexity and ii) storage space. Proposed algorithm has two parts encoder and decoder. Encoder part compresses the DICOM files and Decoder does the inverse operation.

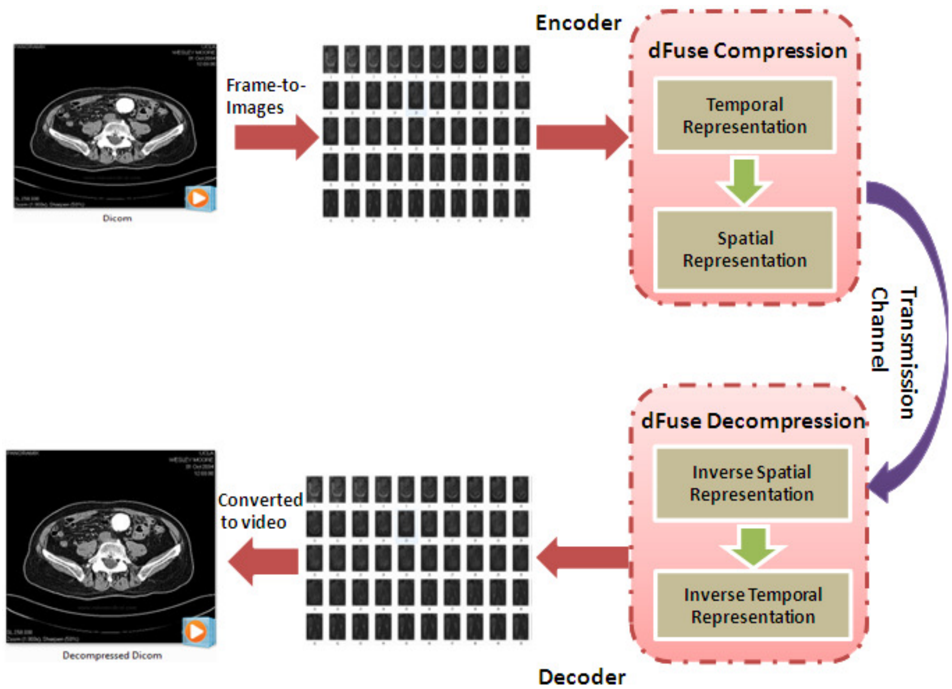
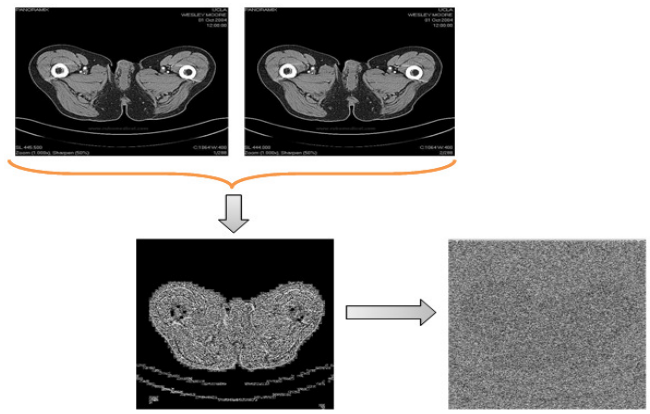



**FIGURE 1:** dFuse Architecture

### 4.1 Spatial and Temporal Representations



a. Temporal Representation of an Image

b. Spatial Representation of an Image

**FIGURE 2:** Spatial & Temporal Representations

dFuse uses two representations for images known as spatial and temporal representation [14]. In spatial representation (Figure 2b), an image is compressed based on width and height. In this type of representation, redundancies within frames are removed. In the case of temporal representation (Figure 2a), redundancies between frames are removed. Here, when a set of frames are passed to this module, a single base frame is fixed and from which redundancies in the following frames are computed. A simple example of this is to consider every second frame as a base frame and calculate difference values between temporally related pixels between frames. This data would be much lesser in size as compared to the original data and can be exploited to achieve higher compression ratios. This would result in a large number of zero values are being computed for every second frame. This algorithm exploits redundancies that occur due to recurrent patterns in data and offers good compression for data comprising large numbers of frequently occurring patterns. There by saving memory and bandwidth.

### 4.2 dFUSE Algorithm

**Encoder Part**

Step: 1 Convert video to images

Step: 2 Extract rgb values from each image

Step: 3 Apply temporal compression

a. Consider the first two images which are more similar and consider the first image as reference image
b. Let RGB1 is the rgb values taken from the reference image and RGB2 is the rgb values taken from the another image
c. Apply the formula $RGB = RGB_1 - RGB_2$

Step: 4 Apply spatial compression [Forward dFuse]

a. Calculate time domain data $\alpha = \sqrt{256/N_f N_c N_r}$ where $N_f, N_c, N_r$ is the number of frames, columns, rows of an image.
b. X (P,Q,R) = $\sum_{m=0}^{N_f-1}\sum_{n=0}^{N_c-1}\sum_{p=0}^{N_r-1} \alpha * C_P^{2m+1} C_Q^{2n+1} C_R^{2p+1}$, where α is the time domain data and $C_R^{2p+1}$ all cosine coefficients and defined as $C_R^{2p+1} = \cos[((2p+1)*w*3.14)/2Nr]$
c. Apply the same definition to $C_P^{2m+1}$ and $C_Q^{2n+1}$

Step: 5 Archive and send it through transmission channel

**Decoder Part**

Step: 1 Apply spatial decompression [Inverse dFuse]

a. Extract the rgb values from the spatially compressed image.
b. Calculate time domain data $\alpha = \sqrt{256/N_f N_c N_r}$ where $N_f, N_c, N_r$ is the number of frames, columns, rows of an image.
c. $Y(P^{`}, Q^{`}, R^{`}) = \sum_{m=0}^{N_f-1} \sum_{n=0}^{N_c-1} \sum_{p=0}^{N_r-1} X(P, Q, R) / (\alpha * K_P^{2m+1} K_Q^{2n+1} K_R^{2p+1})$ , where α is the time domain data and $K_R^{2p+1}$ is cosine coefficients and defined as $K_R^{2p+1} = \cos[((2p + 1) * w * 3.14)/2Nr]$
d. Apply the same definition to $K_P^{2m+1}$ and $K_Q^{2n+1}$ .

Step: 2 Apply temporal decompression

a. Group the images.
b. Calculate $RGB_1 = RGB_{ref} + RGB_2$
c. Calculate $RGB_2 = RGB_1 - RGB_{ref}$
d. Reconstruct the images from rgb values.

Step: 3 Reconstruct the video from the reconstructed images.

## 5. EXPERIMENTAL RESULTS

Proposed algorithm is implemented and then checked its efficiency using these parameters, i)file size, ii) compressed ratio, iii) PSNR ratio and iv) compression percentage.

### 5.1. File Size

DICOM files of different sizes have been taken for compression. Table 1 tabulate and compare the results obtained from different sized files and various other compression algorithms. There is a large size difference between the original video and the compressed video. The video size has been reduced drastically. When compared to other compression algorithms, the proposed algorithm is most efficient. Figure 3 shows the plotted graph for the Table 1 values and from graph, we can notice the drastic reduction in video size, for e.g. 444 MB sized file is compressed to 16.8MB, which is efficient when comparing other algorithms.

| Video Size[MB] | Wavelet Transforms | | Cosine Transforms | |
|---|---|---|---|---|
| | Haar [MB] | Daubechies [MB] | 2D-DCT [MB] | dFuse [MB] |
| 0.5 | 0.11 | 0.09 | 0.08 | 0.036 |
| 12 | 0.871 | 0.6 | 0.53 | 0.44 |
| 35 | 2.38 | 1.62 | 1.51 | 0.96 |
| 41 | 3.07 | 2.56 | 2.04 | 1.15 |
| 83 | 3.81 | 2.70 | 2.58 | 1.47 |
| 216 | 14.7 | 15.6 | 13.3 | 8.11 |
| 444 | 28.12 | 24.58 | 23.46 | 16.8 |

**TABLE 1:** Comparing compression video file size for different compression algorithms

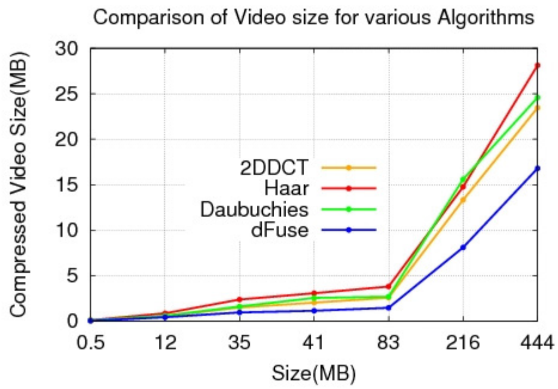



**FIGURE 3:** Comparing compression video file size for different compression algorithms

### 5.2. PSNR Calculation

Peak-signal-to-noise-ratio is used as a quality parameter for reconstruction of compression images or videos. Here signal is in the original data and the noise is in the compressed data. Calculating PSNR values is used as an estimation to human awareness for reconstructing quality of compressed data i.e. higher PSNR, high quality of video.

*PSNR Calculation:*

Step 1: Calculate Mean Square Error [MSE]

$$d(f(x,y), f'(x,y)) = ||(f(x,y) - f'(x,y)||^2$$

$$= \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j}^{n-1}\left((f(x,y) - f'(x,y)^2\right)$$, where $f(x,y)$ and $f'(x,y)$ are original and reconstructed images respectively, m and n are image size.

Step 2: PSNR=10$\log_{10} {}^{1}/_{MSE}$

Videos of different sizes are considered and PSNR values are calculated. Table 2 shows PSNR values obtained for different comparison techniques. Figure 4 displays the graph plotted with various video sizes vs. obtained PSNR values. We can notice that the desperate increase in PSNR values which are in acceptable range and efficient when compared to other algorithms, e.g. 444 MB sized file has PSNR value of 49.998.

| Video Size[MB] | Wavelet Transforms | | Cosine Transforms | |
|---|---|---|---|---|
| | Haar [dB] | Daubechies [dB] | 2DDCT [dB] | dFuse [dB] |
| 0.5 | 20.141 | 28.785 | 43.359 | 48.136 |
| 12 | 19.542 | 27.127 | 40.349 | 49.214 |
| 35 | 19.128 | 28.992 | 41.484 | 46.412 |
| 41 | 18.194 | 27.947 | 40.441 | 49.128 |
| 83 | 15.426 | 17.389 | 34.151 | 40.361 |
| 216 | 18.114 | 20.666 | 40.888 | 49.365 |
| 444 | 21.147 | 30.845 | 44.541 | 49.998 |

**TABLE 2:** Comparing PSNR values different compression algorithms

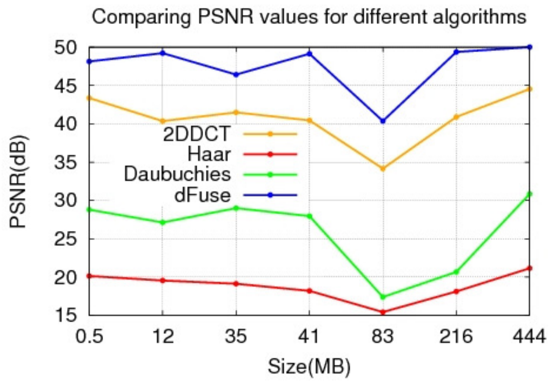



**FIGURE 4:** Comparing PSNR values different compression algorithms

### 5.3. Compression Percentage

Compression percentage is the ratio of difference between the original image and compressed image. This percentage gives how much quantity the original image is compressed. Compression percentage is calculated by using the following formula, $CP = \frac{(O_{size} - C_{size})}{O_{size}} * 100$, where $O_{size}$ is the size of original video and $C_{size}$ is the size of compressed video. Table 3 shows the obtained compression percentage for different video sizes and for various compression algorithms including the proposed one. Figure 5 illustrates the graph of obtained compression percentage for various video sizes vs. different compression algorithms. Proposed algorithm reaches maximum of 96.2 compression percentage for 444 MB sized DICOM file, denoting that the algorithm compresses well than the other algorithms which has maximum of 93 to 94 compression percentage.

| Video Size[MB] | Wavelet Transforms | | Cosine Transforms | |
|---|---|---|---|---|
| | Haar [%] | Daubechies [%] | 2DDCT [%] | dFuse [%] |
| 0.5 | 78 | 82 | 84 | 94 |
| 12 | 92.7 | 95 | 95.5 | 96.3 |
| 35 | 93.2 | 95.3 | 95.6 | 97.2 |
| 41 | 92.5 | 93.7 | 95 | 97.1 |
| 83 | 95.4 | 96.7 | 96.8 | 98.2 |
| 216 | 93.1 | 92.7 | 93.8 | 98.5 |
| 444 | 93.6 | 94.4 | 94.7 | 96.2 |

**TABLE 3:** Comparison of compression percentage for different compression algorithms

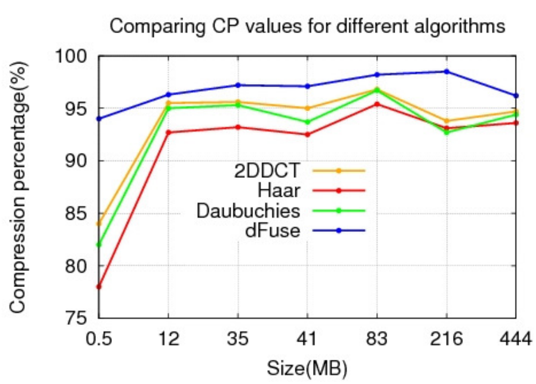



**FIGURE 5:** Comparison of compression percentage of different compression algorithms

### 5.4. Compression Ratio

Compression ratio is the ratio of original image to compressed image. This ratio gives how much quantity the image is compressed when compared to original image. Compression ratio is calculated by using the following formula, $CR = \frac{O_{size}}{C_{size}}$ , where $O_{size}$ is the size of original video and $C_{size}$ is the size of the compressed video. Table 4 illustrates the compression ratio obtained for proposed and for various other algorithms.

| Video Size[MB] | Wavelet Transforms | | Cosine Transforms | |
|---|---|---|---|---|
| | Haar | Daubechies | 2DDCT | dFuse |
| 0.5 | 4.5:1 | 5.5:1 | 6:1 | 13.8:1 |
| 12 | 13.7:1 | 20:1 | 22.6:1 | 27.2:1 |
| 35 | 14.7:1 | 21.6:1 | 23.1:1 | 36.5:1 |
| 41 | 13.3:1 | 16:1 | 20:1 | 35.6:1 |
| 83 | 21.7:1 | 30.7:1 | 32.1:1 | 56.4:1 |
| 216 | 14.6:1 | 13.84:1 | 16.24:1 | 26.6:1 |
| 444 | 15.7:1 | 18:1 | 18.9:1 | 26.4:1 |

**TABLE 4:** Comparison of compression ratio of different compression algorithms

## 6. CONCLUSION

In this paper, we proposed a modified three dimensional discrete cosine algorithm called dFuse. Experimental results obtained from various check point, the proposed algorithm reveals good results in compression and maintains high video quality for reconstructed video. dFuse has high compression ratio and compression percentage, which saves bandwidth and solves the low bandwidth problems. Also Peak-to-Signal Noise (PSNR) ratio is high and in acceptable range when compared to other compression algorithms. Another interesting and useful feature in dFuse is it saves storage space as it reduces the video size drastically. It compresses up to 97% of a video with high quality. The proposed algorithm is created, keeping in mind, to reduce the size of DICOM-format image archives only and have not applied to other medical images such as X-Ray, Ultrasound, CRT scan and MRI scan, which will be our future work.

## 7. REFERENCES

[1] M. Nelson and J. L. Gailly, “The Data Compression Book”, 2nd ed. Newyork, M & T Books, 1996.

[2] A. C. Bovik, “The Essential Guide to Image Processing”, Academic Press, 2009.

[3] A. K. Jain, “Image data compression: A review”, In proceedings of IEEE, vol. 69, pp. 349–389, 1981.

[4] K. Rao, P. Yip, "Discrete Cosine Transform: Algorithms, Advantages, and Applications", NY Academic, 1990.

[5] Graps, A. "An introduction to wavelets", IEEE Computational Science & Engineering, vol.2, no.2, pp.50-61, Summer 1995, doi: 10.1109/99.388960, 1995.

[6] P.C. Cosman, R.M. Gray, and R.A. Olshen, “Evaluation Quality of Compressed Medical Images: SNR, Subjective Rating, and Diagnostic Accuracy”, In proceedings of the IEEE, vol. 82, No. 6 , pp. 919-932,June, 1994.

[7] D. Feng, D. Ho, H. Iida, and K. Chen, “Techniques for functional imaging,” in Medical Imaging Systems Techniques and Applications: General Anatomy, C. T. Leondes, Ed. Amsterdam, The Netherlands: Gordon and Breach, invited chapter, pp. 85–145,1997.

[8] Lodge, N.K, “Video compression techniques”, Applications of Video Compression in Broadcasting, IEE Colloquium on, 2/1 214, 2002.

[9] Ouni T, Ayedi W, Abid M, ”New low complexity DCT based video compression method ”, In Proceedings of Tele Communication ICT09, doi:10.1109/ICTEL2009.5158644, pp:202-207,2009.

[10] Gargour, C., Gabrea, M., Ramachandran, V., Lina, J.M., "A short introduction to wavelets and their applications," IEEE Circuits and Systems Magazine, vol.9, no.2, pp.57-68, doi: 10.1109/MCAS.2009.932556, 2009.

[11] S. Mallat., “A Wavelet Tour of Signal Processing”, Third Edition: The Sparse Way. Academic Press, 2008.

[12] Nadia Benamor and Najoua Essaokri Ben amara, "DICOM Image Compression By Wavelet Transform," In proceedings of IEEE Systems Man and Cybernetics (SMC), Harnmamet, Tunisia, 2002.

[13] Suapang, P., Dejhan K., Yimmun S., "Medical image compression and DICOM-format image archive”, ICCAS-SICE, pp.1945-1949, 2009.

[14] D. Ho, D. Feng, and K. Chen, “Dynamic image data compression in spatial and temporal domains: Theory and algorithm,” IEEE Trans. Inform. Technol. Biomed., vol. 1, pp. 219–228, Dec. 1997.