Lossless Compression based on Curvelet-IWT

AvishaKhanna, Sanjeev Sharma, Nishchol Mishra
School of information technology, RGPV,Bhopal, India
avishakhanna87@gmail.com, sanjeev@rgtu.net, nishchol@rgtu.net

Abstract
The growth of data rate increases in decades. The maximum part of data is image, image required huge amount of memory for storage purpose and take more time for process of storage. Now lossless image compression is a necessary for saving memory and time. In the process of lossless image compression transform function play a vital role. In this work, we obtained better compression technique using Curvelet-IWT, resulting better peak signal to noise ratio (PSNR), Compression rate, Compression ratio. The transform function that we are using in this work is curvelet transform. In this work, Curvelet-IWT is compared with IWT, IWPT and DPCM-IWT resulting better than other techniques.

Keywords
Curvelet- IWT, IWT, IWPT and DPCM-IWT.

1. Introduction
Lossless image compression plays a crucial rule in the field of data compression. It develops very slowly and it is difficult to improve the efficiency of compression, since complete information and high fidelity are demanded. The lossless compression ratio of the conventional methods is around 2:1 and 3:1, which doesn’t satisfy practical compression needs. So, it is necessary to develop more efficient lossless image compression methods. In this paper, we will discuss various compression techniques like IWT, IWPT, DPCM-IWT. We will also discuss on various characteristics of image compression. IWT is an invertible integer-to-integer wavelet analysis algorithm [12][13]. IWT can be used in the applications where integer coefficients need to be produced for integer-encoded signals.IWT is memory-efficient, computationally faster and suitable in lossless data-compression applications. The IWT enables you to reconstruct an integer signal perfectly from the computed integer coefficients [7].IWT yields a representation which can be lossless, as it maps an integer valued sequence onto integer valued coefficients in the transformed domain; moreover, it allows for the selection of an adaptive representation, which can match the variable characteristics of image better than the IWT. Although IWPT has get very good result in the lossless compression of images, it is not perfect [6]. The range of the IWPT coefficients is bigger than the wavelet transform coefficients which affect the effect of the lossless image compression. But, if add the DPCM before the IWPT, it can decorrelate the raw image data, therefore increasing the compressibility of the image. So we can get better compress result than using the IWPT called DPCM-IWT[6].

Recently, curvelet transform has been produced. It represents edges and other singularities along curves much more efficiently than traditional transforms, i.e. using many fewer coefficients for a given accuracy of reconstruction. It requires only about 1/\sqrt{N}curvelets to represent an edge. In this work, we will use curvelet transform function to achieve better compressed image [5] [17].

2. Integer Wavelet Transform
The wavelet domain is growing up very quickly. A lot of mathematical papers and practical trials are published every month. Wavelets have been effectively utilized as a powerful tool in many diverse fields, including approximation theory, signal processing, physics, astronomy, and image processing. In one dimensional discrete wavelet transform when high pass filter is applied it results in smoothed version of the input and when low pass filter is applied, high frequency part is captured. Syntheses filters are used for reconstruction and the results of these convolutions are added. In two dimensional, firstly, one step of the one dimensional transform is applied to all rows. Then, we repeat the same for all columns. In the next step, the coefficients are proceeded that result from a convolution in both directions [12] [13]. As shown in figure 1, these steps result in four classes of coefficients: (HG and GH) reflect vertical and horizontal information, (HH) coefficients represent diagonal features of the image. At the coarsest level, we also keep low pass coefficients (LL). We can do the same decomposition on the LL quadrant up to \log_{2}(\min (height, width)). Since the discrete wavelet transform allows independent processing of the resulting components without significant perceptible interaction between them, hence it is expected to make the process of imperceptible embedding more effective. However, the used wavelet filters have floating point coefficients. Thus, when the input data consist of sequences of integers (as in
the case for images), the resulting filtered outputs no longer consist of integers, which doesn't allow perfect reconstruction of the original image [10]. However, with the introduction of Wavelet transforms that map integers to integers we are able to characterize the output completely with integers [7].

S transform is an example of wavelet transforms that map integers to integers. The result of high pass filters is smooth(s) output for an index n given in equation (1a) and the result of low pass filter is detail (d) output for an index n that is given in equation (1b). However, the sum and the difference of two integers are either both odd or both even. We can thus safely omit the last bit of the sum since it equals to the last bit of the difference. The S transform is thus reversible and its inverse is given in equations (2a) and (2b) [7].

![Image of 2D Wavelet Transform](image)

**Figure 1: A Two Dimensional Wavelet Transform**

\[
S(n) = \left\lfloor \frac{x(2n) + x(2n + 1)}{2} \right\rfloor \quad (1a)
\]

\[
d(n) = x(2n) - x(2n + 1) \quad (1b)
\]

\[
x(2n) = s(n) + \frac{d(n) + 1}{2} \quad (2a)
\]

\[
x(2n + 1) = s(n) - \frac{d(n) + 1}{2} \quad (2b)
\]

However, we need to redefine those equations in 2D in order to be applied on images and hence be useful in our implementation. In this section, we will define the construction of the 2D S-transform. (3a), (3b), (3c), and (3d) equations of an image are used for computed the 2D S-transform. Where original image is denoted by I. Original image(I) is X pixel high and Y pixel wide. Pixels located at position i and j by Iij. The transform is reversible, i.e., we can exactly recover the original image pixels from the computed transform coefficients. The inverse is given in equations (4a), (4b), (4c), and (4d). Note that the transform results in four classes of coefficients: (A) the low pass coefficients, the (H)

coefficients represent horizontal features of the image, whereas (V) and (D) reflect vertical and diagonal information respectively. During the transform we ignore any odd pixels on the borders [7].

\[
A_{ij} = \left[ (I_{2i,2j} + I_{2i+1,2j})/2 \right] \quad 3a
\]

\[
H_{ij} = \left[ (I_{2i,2j+1} - I_{2i,2j}) \right] \quad 3b
\]

\[
V_{ij} = \left[ (I_{2i+1,2j} - I_{2i,2j}) \right] \quad 3c
\]

\[
D_{ij} = \left[ (I_{2i+1,2j+1} - I_{2i,2j}) \right] \quad 3d
\]

\[
I_{2i,2j+1} = A_{ij} + \left[ (H_{ij} + 1)/2 \right] \quad 4a
\]

\[
I_{2i+1,2j} = I_{2i,2j+1} + V_{ij} - H_{ij} \quad 4b
\]

\[
I_{2i+1,2j+1} = I_{2i+1,2j} + D_{ij} - H_{ij} \quad 4c
\]

\[
I_{2i,2j} = \left[ (A_{ij} + H_{ij})/2 \right] \quad 4d
\]

Where

\[1 \leq i \leq \frac{x}{2y} \leq j \leq \frac{y}{2}\]

Note that the presented transforms are not computed using integer arithmetic. The computations are done with floating point number but the result is guaranteed to be integer because the floor function is used and hence the invertibility is preserved.[7] [8] [9].

3. Integer Wavelet Packet Transform

3.1 Wavelet transform and wavelet packets transform

In this section a review of the wavelet transform and of the filter bank scheme are given, and the wavelet packets transform are introduced. The block scheme of the single level wavelet transform is shown in Figure 2.

![Wavelet Transform Diagram](image)

**Figure 2: One Level Wavelet Transform**

The low-pass analysis filters and the high-pass ones are followed by down sampling of a factor two. At the reconstruction side, the low-pass and band-pass branches are upsamped and filtered with the synthesis filters H(z) and G(z) in order to obtain the original signal. A wavelet transform on J levels is obtained by iterating the filter bank J-1 times on the low-pass branch. The wavelet transform coefficients consist of the J high-pass and the terminal low-pass node sequences output by the
filter bank tree. Given a perfect reconstruction filter bank, the iterated scheme represents an either orthonormal or biorthogonal (non redundant) representation of the original signal. Differently from the wavelet transform, the J-level WPT are achieved by iterating the one level filter bank on both the low-pass and the high-pass branch, and then applying a pruning algorithm to select a suitable representation. An algorithm has been proposed which selects the best representation of a sequence across the entire tree based on some proper cost function, which must measure the compactness of the representation [6].

3.2. Integer wavelet transform and integer wavelet packets transform

It is straightforward to understand that the same procedure which leads to IWT from wavelet transform can be applied to the wavelet packets transform, yielding the IWPT. The implementation follows the same scheme used for the IWT. The IWPT tree can be built iterating the single wavelet decomposition step on both the low-pass and high-pass branches, with rounding off in order to achieve the integer transforms. IWPT yields a representation which can be lossless, as it maps an integer valued sequence onto integer valued coefficients in the transformed domain; moreover, it allows for the selection of an adaptive representation, which can match the variable characteristics of images better than the IWT[1][6].


Although IWPT has get very good result in the lossless compression of images, it is not perfect. The range of the IWP coefficients is bigger than the wavelet transform compression of images, it is not perfect. The range of the wavelet coefficients is bigger than the wavelet transform [6][11]. We use these windows in the frequency domain to construct a family of analysing elements with three parameters: scale $a > 0$, location $b \in \mathbb{R}^2$ and orientation $\theta \in (0, 2\pi)$ (or $(-\pi, \pi)$ according to convenience below). At scale $a$, the family is generated by translation and rotation of a basic element $\gamma_{a00}$:

$$\gamma_{ab\theta}(x) = \gamma_{a00}(R_{\theta})(x-b),$$

Where $R_{\theta}$ is the 2-by-2 rotation matrix effecting planar rotation by $\theta$ radians. The generating element at scale $a$ is defined by going to polar Fourier coordinates $(r, \omega)$ and setting

$$\gamma_{a00}(r,\omega) = W(a,\omega) V\left(\frac{\omega}{a^2}\right), a^2, 0 < a < a_0$$

Thus the support of each $\gamma_{a00}$ is a polar ‘wedge’ defined by the support of $W$ and $V$, the radial and angular windows, applied with scale-dependent window widths in each direction. In effect, the scaling is parabolic in the polar variables $r$ and $\omega$, with $\omega$ being the ‘thin’ variable. In accord with the use of the terminology curvelet we denote families exhibiting such parabolic scaling [4, 2, 5, 3], we call this system of analysing elements curvelets. However, note that the curvelet family is not a simple affine change-of-variables acting on $\gamma_{a00}$ for $a = a_0$. We initially omit description of the transform at coarse scales. Note that these curvelets are highly oriented and they become very needle-like at fine scales. Equipped with this family of curvelets, we can define Continuous Curvelet Transform $\Gamma_c$, a function on scale/location/direction space defined by

$$\Gamma_c(a, b, \theta) = \langle \gamma_{ab\theta}, f \rangle,$$

$a < a_0$, $b \in \mathbb{R}^2$, $\theta \in (0, 2\pi)$.

Here and below, $a_0$ is a fixed number $a_0$, we call the coarsest scale for our problem. It is fixed once and for all, and must obey $a_0 < x$ for the above construction to work properly.

6. Lifting and Shifting Scheme

The lifting scheme is a technique for both designing wavelets and performing the discrete wavelet transform. Actually it is worthwhile to merge these steps and design the lifting scheme while performing the wavelet transform. This is then called the second generation wavelet transform. The basic idea of lifting is the following: If a pair of filters $(h,g)$ is complementary, that is it allows for perfect reconstruction, then for every filter $s$ the pair $(h',g')$ with $h'(z) = h(z) + s(z^2)$ allows for perfect reconstruction, too. Of course, this is also true for every pair $(h,g')$ of the form $g'(z) = g(z) + t(z^2)$ . $h(z)$.
converse is also true: If the filter banks (h, g) and (h', g') allow for perfect reconstruction, then there is a unique filter s with h'(z) = h(z) + s(z^2), g(z).[6][11]

The method of zero shifting is a simple and easy-to-implement technique which preserves the embedded property of the SPIHT coding. By downward scaling of the pixel values by 2^{N-1}, N being the number of bits representing the original pixel, the spatial domain magnitude becomes bipolar ranging from (-2^{N-1}) to (+2^{N-1}), with almost half of the maximum absolute spatial value. In the transform domain, the wavelet lifting scheme produces the high pass sub band by taking the weighted differences of the pixel values after prediction. Similarly, the low pass sub band is produced by taking the weighted average of the pixel values after the updating step. This low pass sub band is further decomposed iteratively for each level of decomposition to finally provide one lowest frequency band, the rest being high frequency bands. [6]

7. PROPOSED WORK

The process of image compression suffered from the problem of extraction of actual data without loss of information. In this work, we obtained better compression technique using curvelet-IWT, resulting better peak signal to noise ratio (PSNR), Compression rate, Compression ratio. In current trend, the compression of an image is performed by transform function by the decomposition of value on coefficient factor in low and high pass filter. In low pass filter, the filter generates fractional number such as 6.4. Here 0.4 is fractional data and actual data is not quantified which causes loss of data. We have proposed the integer wavelet transform (IWT) with the combination of curvelet transform. Firstly the curvelet transform is applied on an original image f -> \( \mathcal{G}f \) of function \( f(x,y) \) on \( R^2 \).

Where a is continuous scale and b is location where \( b \in R^2 \) and \( \Theta \) is orientation that shows direction where \( \Theta \in (0,2\pi) \).

\( \mathcal{G}f(a,b,\Theta) = (\gamma_{a\Theta},f) \) means project f onto \( \gamma_{a\Theta} \) called curvelet that causes sub band decomposition of an image which provides low and high frequencies values which included fractional frequencies too. Integer wavelet transform (IWT) is applied that converts fractional value into whole number. Shifting are done on the basis of low and high pass filter which produces the high pass sub band by taking the weighted differences of the pixel values after prediction. Similarly, the low pass sub band is produced by taking the weighted average of the pixel values after the updating step. This low pass sub band is further decomposed iteratively for each level of decomposition to finally provide, one lowest frequency band, the rest being high frequency bands. Curvelet IWT is compared with IWT, IWPT and DPCM-IWT resulting better than other techniques. By using Curvelet IWT technique, better compression can be obtained on the curves of images resulting better outcome. We call the new method Curvelet-IWT. The steps are as following.

1) Read original image \( f = f(x,y) \)

2) Divide the image into resolution layers. Each layer contains different frequencies. \( P_0 \) is low-pass filter \( \alpha_1 \), \( \alpha_2 \) is high pass filters.

3) Image expression can be written as :
   \[ f = p_0(p_0f) + \sum_s \alpha_s (\alpha_s f) \]

4) Ideal Low pass filter \( \Phi_0 \) with low frequencies near \( |l| \leq 1 \)

5) Ideal High Pass filter \( \Psi_2 \) with frequency near domain
   \[ |f| \in [2^s, 2^{2s} + 2] \]

The sub-band decomposition is simply applying a convolution operator:

\[ P_0 f = \Phi_0 * f, \alpha_s f = \psi_{2s} * f \]

6) Apply integer wavelet transform and convert fractional frequency value into whole number.

\[ S(n) = \frac{x(2n) + x(2n + 1)}{2} \]

\[ D(n) = x(2n) - x(2n + 1) \]

The smooth \( S(n) \) and the detail \( D(n) \) outputs are the results of the application of the high-pass and the low-pass filters respectively.

7) Shifting are done on the basis of low pass filter and high pass filter. On the basis of low pass filter and high pass filter we use shifting scheme that brings the spatial values in signed integer range without changing the dynamic ranges, so that the transformed coefficient calculation becomes more consistent. Shifting can be obtained by downward scaling of the pixel values by 2^{N-1}, N being the number of bits representing the original pixel, the spatial domain magnitude becomes bipolar ranging from (-2^{N-1}) to (+2^{N-1}), with almost half of the maximum absolute spatial value. The highest number bit plane is decided by the maximum absolute value of the transformed coefficients. Further, the lower absolute image values in the spatial domain (which originally ranges from 0\leq p_{0,i} \leq 2^{N-1}) are shifted to higher absolute range in the bipolar sense. The lower image values are
detailed information pertaining to the boundaries, edges, etc

8) Huffman encoding is performed on output that is mapping of the code words to the corresponding symbols will result in a compressed data Coding can be done by using following steps:
a) Decorrelate the raw image data by using some prediction function (f).
b) Apply standard Huffman Coding by using the following function:
   **Input:** Alphabet $A = \{a_1, a_2, \ldots, a_n\}$, which is the symbol alphabet of size $n$. Set $W = \{w_1, w_2, \ldots, w_n\}$, which is the set of the (positive) symbol weights (usually proportional to probabilities), i.e. $w_i$ = weight ($a_i$), $1 \leq i \leq n$.
   **Output:** Code $C(A, W) = \{c_1, c_2, \ldots, c_n\}$, which is the set of (binary) code words, where $c_i$ is the codeword for $a_i$, $1 \leq i \leq n$.

   **Constraint:** Let
   $$L(C) = \sum_{i=1}^{n} w_i \times \text{length}(c_i)$$
   be the weighted path length of code $C$. Condition: $L(C) \leq L(T)$.

9) Original image compressed.

The decoding process is the inverse of the encoding process. The processes of encoding and decoding are as Figure 3.

![Figure 3: Encoding Process](image)

### Table 1 Experimental Result

<table>
<thead>
<tr>
<th>Image (Cameraman2)</th>
<th>PSNR</th>
<th>Compression Rate</th>
<th>Compression Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWT</td>
<td>20.584</td>
<td>0.43283 bits/pixel</td>
<td>7.6497:1</td>
</tr>
<tr>
<td>IWPT</td>
<td>22.430</td>
<td>0.5456 bits/pixel</td>
<td>8.3805:1</td>
</tr>
<tr>
<td>DPCM-IWT</td>
<td>26.121</td>
<td>0.67165 bits/pixel</td>
<td>9.2359:1</td>
</tr>
<tr>
<td>Curvelet-IWT</td>
<td>29.625</td>
<td>0.7932 bits/pixel</td>
<td>11.0814:1</td>
</tr>
</tbody>
</table>

As shown in figure 5, one cannot conclude that which image has obtained better compression than other techniques, but from the Table 1 shown above, it can be easily concluded that the performance of Curvelet-IWT is better than other techniques.

8. EXPERIMENTAL RESULTS

The simulation of this paper is done in MATLAB 7.8. In this paper, we compared integer wavelet transform (IWT), integer wavelet packet transform (IWPT) and Differential Pulse Code Modulation Integer Wavelet Transform (DPCM-IWT) with Curvelet IWT. The quality of a compression method could be measured by the traditional distortion measures such as the peak signal to-noise ratio (PSNR), compression rate (CT) and compression ratio (CR). We compared the performance of integer wavelet transform and integer wavelet packet transform on two images: Cameraman2 (512 x 512). Figure 5 shows the reconstructed images of IWT, IWPT, DPCM-IWT and Curvelet-IWT.
9. CONCLUSIONS
Lossless compression of image is an important field in image processing. In this paper we have described the IWT, IWPT and DPCM-IWT and have proposed Curvelet-IWT. After testing on a large image set, we have drawn the conclusion that Curvelet-IWPT is able to achieve better performance of the corresponding real valued transforms without additional computational effort than IWT, IWPT and DPCM-IWT. However, we believe that there is more room for improvement to achieve even better performance by different choices of the elements and parameters of our model or by combining our method with the others.

10. REFERENCES