Chapter 2
Color Image Watermarking Techniques Based on Magic Square and Curvelets

Abstract Four digital watermarking techniques (image watermarking using curvelet transform (WCT), watermark wavelets in curvelets of cover image (WWCT), resized watermark into curvelets of cover image (RWCT), resized watermark wavelets into curvelets of cover image (RWWCT)) based on curvelet transform are discussed in this chapter. In WCT both the embedding and extraction procedures are discussed, where the watermark is embedded into the curvelets of the color cover image. In WWCT the wavelets of watermark are obtained. These wavelets are embedded into the color image curvelets. In RWCT the watermark is resized based on magic square technique and then embedded into the color image curvelets. In RWWCT the resized watermark obtained through the magic square procedure is transformed through wavelet transformation and then the wavelets are embedded into the curvelets of the cover image. The results indicate that the embedding and extraction procedures of WWCT and RWWCT are much superior to WCT and RWCT. The regeneration of watermark image is satisfactory, but lossy.

2.1 Introduction

This chapter deals with the design, development, and implementation of the digital watermark techniques based on magic square, discrete wavelet transforms, and curvelet transform procedures for images. In the process of implementing the above techniques, it also focuses on prerequisites like magic square [1], discrete wavelet transformation [2], and curvelet transformation [3, 4]. As vision index and sensitivity of the human eye are relatively very low for blue and yellow color component [5], the watermark will be embedded in the blue component for all the techniques.

In 2009, Huang Hui-fen executed the application by using gray scale images for both host and watermark images [6]. The study obtained a new image consisting of magic square by scrambling the watermark image and embedded the same through discrete wavelet transform (DWT) technique. However, this is not tested in host color image. In 2008, Chune Zhang et al. have generated watermark image hash by

C.R.S. Rao, M.V.N.K. Prasad, *Digital Watermarking Techniques in Curvelet and Ridgelet Domain*, SpringerBriefs in Computer Science,
DOI 10.1007/978-3-319-32951-2_2

obtaining approximate scale through multi-scale curvelet transformations. The image features represented as bit sequences are then embedded into cover image [7]. Yuancheng Li discussed a novel algorithm based on gray scale images [8]. The watermark image is first scrambled using Arnold scrambling technique and then same is embedded using ridgelet transform technique. The extracted watermark is compared with original and calculated normalized correlation of approximately 0.9. Patrizio Campisi considered gray scale images for his results [9]. The watermark is embedded into the most significant edges of the host image with the help of ridgelet transform technique. The extracted watermark showed a high correlation of 100 % for JPEG formats with 0.2-bit rates.

It is proposed to develop a technique for inserting the watermark curvatures into the cover image curvatures to secure the authentication information in the watermarked image, while the watermarked image undergoes various image compression attacks. This chapter also proposkes a technique for embedding multiple copies of the watermark through the magic square procedure.

2.1.1 Magic Square Technique

An nth order magic square is arranging n^2 distinct integers in a square, such that the sum of numbers of any row, any column or any diagonal will be the same constant [1]. The magic square is a part of Indian culture from the times of Vedic days, for example the Ganesh Yantra. The 4×4 magic square of tenth century existing in Khajuraho, Parshvanath Jain temple, India, is a popular magic square [10]. Figure 2.1 represents an example of 4×4 magic square [11, 12].

Fig. 2.1 4×4 magic square

34	34	34	34	34	34
34	1	13	12	8	
34	2	14	7	11	
34	15	3	10	6	
34	16	4	5	9	

2.2 Image Watermarking Using Curvelet Transform

This section discusses two methods in which watermark can be embedded into the curvelets of the cover image. In the first method, WCT, the watermark is embedded into cover image curvelets. In the second method, WWCT, the wavelet coefficients of watermark are embedded into the curvelets of cover image [12].

2.2.1 Technique to Insert Watermark in Curvelets of Cover Image (WCT)

In this method the watermark is segmented into nonoverlapping partitions where each partition size is p×p pixels. Similarly, cover image also partitioned and curvelets are obtained for each partition. The watermark partitions are embedded into the curvelets as discussed in Sect. 2.2.1.1. Similarly, watermarked image is partitioned and then the watermark is extracted from the curvelets of the respective partitions as discussed in Sect. 2.2.1.2. The architecture of the watermark embedding procedure is shown in Fig. 2.2 and the architecture of the extraction procedure is shown in Fig. 2.3 [12].

2.2.1.1 Digital Watermark Embedding Procedure

The detailed procedure with reference to Fig. 2.2 for embedding watermark is discussed as follows.

Step 1: The third dimension or blue component of the image, CI, is partitioned into CI1 nonoverlapping blocks of p×p (for example 8×8 size) pixels in each.

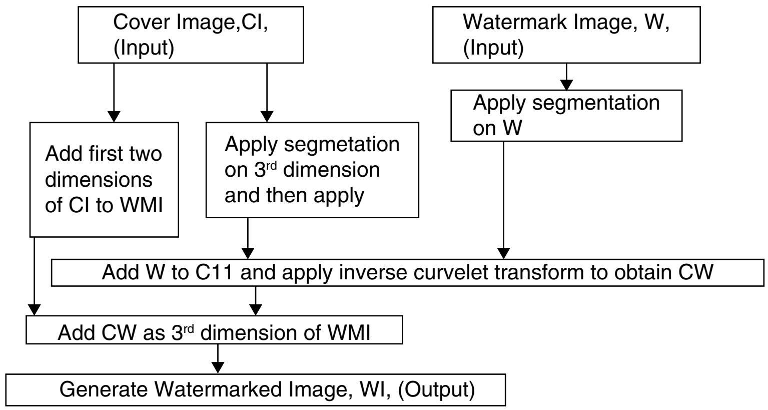

Fig. 2.2 WCT-digital watermark embedding procedure

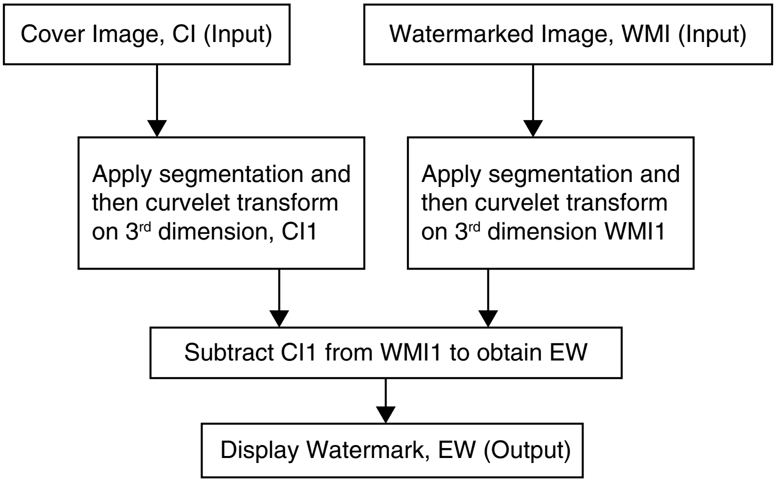

Fig. 2.3 WCT-digital watermark extraction procedure

Step 2: W is partitioned into W1 small blocks, where the size of each block is p × p pixels (for example 8 × 8 size).
Step 3: The curvelet procedure is applied to a block from CI1 to obtain curvelet coefficients, CC.
Step 4: W1 is added to the CC, using Eq. 2.1.

$$R = W1 + CC \tag{2.1}$$

Step 5: The inverse curvelet transform procedure is applied to R and the block is added to the respective location of CC1.
Step 6: Step 5 to Step 7 are applied repeatedly till all the blocks of W1 add to respective blocks of CI1.
Step 7: Add the remaining blocks of CI1 to CC1.
Step 8: Add red and green planes of CI to resultant image, WMI, and then add CC1 to WMI to generate a watermarked image.

2.2.1.2 Digital Watermark Extraction Procedure

The procedure for extraction of watermark based on the architecture shown in Fig. 2.3 is as follows.

Step 1: The third dimension or blue component of watermarked image, WMI, is partitioned into a number of p × p (for example 8 × 8 size) pixel sized nonoverlapping blocks, WMI1.
Step 2: The third dimension or blue component of cover image, CI, is partitioned into a number of p × p pixel sized nonoverlapping blocks, CI1.

Step 3: The curvelet transformation is applied to a block from CI1 to obtain curvelet coefficients C1.
Step 4: The curvelet transformation is applied to a block from WMI1 to obtain curvelet coefficients WM1.
Step 5: Subtract the values of C1 from WM1 to obtain a block of extracting a watermark image, EWM.
Step 6: Repeat Step 5 to Step 7 for a size of given watermark (Since the watermark is embedded in continuous locations).
Step 7: Compare the extracted watermark EWM with W.

2.2.1.3 Experimental Results

A Lenna color image (Fig. 2.4a) of size 512×512 pixels is considered as the cover image, CI. A grayscale image with 64×64 pixels size is considered as watermark, W (Fig. 2.4b). The block size is considered as 8×8 pixels. The values of W are embedded with the pixel values of CI as discussed in the embedding procedure Sect. 2.2.1.1. Figure 2.5a shows the resultant watermarked image obtained through the embedding procedure. The extraction procedure discussed in Sect. 2.2.1.2 is applied on Fig. 2.5a and the watermark is extracted as shown in Fig. 2.5b and compared with the original watermark [12].

The watermark is embedded into various 24-bit standard 512×512 pixel color images, like Baboon, Pepper, and Barbara, and the respective watermarked images are generated. Table 2.1 gives the results of peak signal to noise ratio (PSNR) obtained when these watermarked images are compared with the respective cover images. From the results it is observable that the results of the WCT discussed in Sect. 2.2.1 are better than Yuancheng Li's method [8]. The results are comparable to the results obtained by Chune Zhang [7].

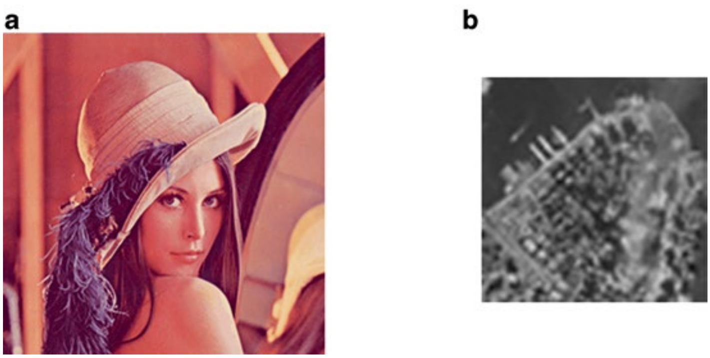

Fig. 2.4 (**a**) Lenna cover image. (**b**) Original watermark (64×64 pixels)

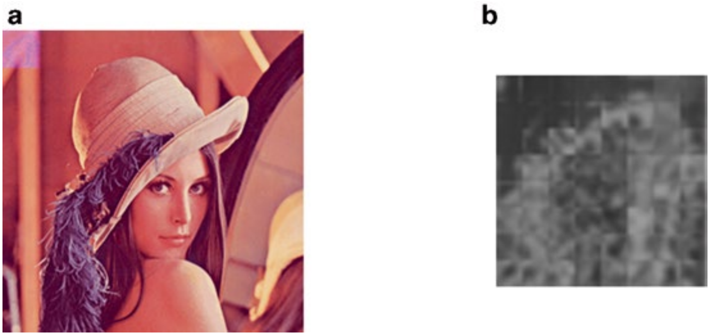

Fig. 2.5 (**a**) Watermarked Lenna BMP Image (WCT). (**b**) Extracted watermark from Fig. 2.4 (WCT)

Table 2.1 PSNR results for various cover images (WCT)

S. No.	Image name (size 512×512)	WCT	Huang Hui-fen [6]	Yuancheng Li [8]	Patrizio Campisi [9]	Chune Zhang [7]
1	Lenna	31.49	39.9	27.7	40.46	43.166
2	Baboon	31.17	–	–	37.6	–
3	Barbara	31.30	–	–	40.14	–
4	Pepper	31.20	–	–	–	–

The Lenna watermarked image (BMP image file) shown in Fig. 2.5a is compressed with GIF, JPEG, and PNG compression techniques and the resulting watermarked images are shown in Figs. 2.6, 2.8, and 2.10, respectively. Similarly the watermark is extracted from these compressed images as discussed in the procedure Sect. 2.2.1.2 and the results are shown in Figs. 2.7, 2.9, and 2.11, respectively. The technique is tested for its performance by comparing the extracted watermark with the original watermark (Fig. 2.4b) and the PSNR results are tabulated in Table 2.2. The resultant PSNR shows that the obtained watermarks are consisting of more noise [12].

2.2.2 Technique to Insert Watermark Wavelets in Curvelets of Cover Image (WWCT)

The watermark embedding and the extraction procedures are given in Figs. 2.12 and 2.13 respectively. In this method the watermark is segmented into W1 non-overlapping partitions, where each partition size is p×p pixels. The cover image is also partitioned similarly and curvelets are obtained for every partition. The wavelet coefficients of watermark partitions are embedded into the obtained

Fig. 2.6 Watermarked Lenna image compressed by GIF format (WCT)

Fig. 2.7 Extracted watermark from Fig. 2.6 (WCT)

Fig. 2.8 Watermarked Lenna image compressed by JPEG format (WCT)

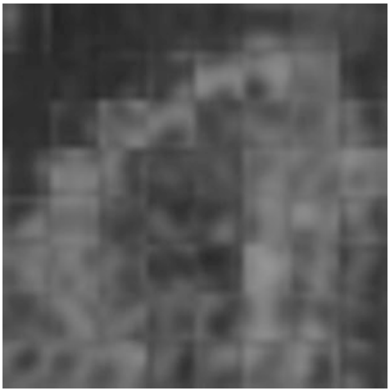

Fig. 2.9 Extracted watermark from Fig. 2.8 (WCT)

Fig. 2.10 Watermarked Lenna image compressed by PNG format (WCT)

Fig. 2.11 Extracted watermark from Fig. 2.10 (WCT)

Table 2.2 Variation of PSNR for different compression formats (WCT)

S. No.	Type of attack on 24-bit color Lenna image (size 512×512)	PSNR obtained
1	Watermark extracted from BMP image	21.53
2	Bmp converted to GIF format	21.41
3	Bmp converted to JPEG format	20.39
4	Bmp converted to PNG format	21.42

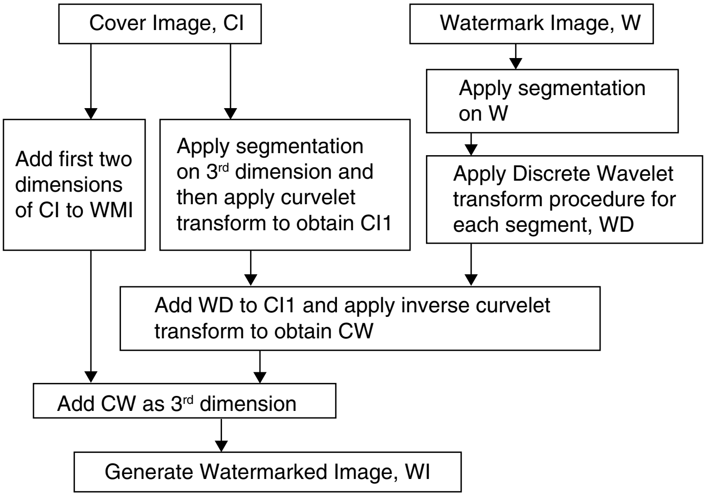

Fig. 2.12 WWCT-digital watermark embedding procedure

curvelets as discussed in Sect. 2.2.2.1. In the same way the watermarked image is partitioned into WMI1 blocks and then the watermark is extracted from the curvelets of the respective blocks as discussed in Sect. 2.2.2.2 [12].

2.2.2.1 Digital Watermark Embedding Procedure

The architecture of embedding procedure is given in Fig. 2.12 and the detailed steps for embedding watermark are as follows:

Step 1: The third dimension or blue component of the image, CI, is partitioned into a number of p×p pixel sized nonoverlapping blocks, CI1.
Step 2: W is partitioned into a number of small blocks, where the size of each block is p×p pixels, W1.
Step 3: The curvelet procedure is applied to a block from CI1 to obtain curvelet coefficients, CC.

Step 4: The discrete wavelet transform procedure is applied to a block from W1 to obtain transform coefficients, WD.
Step 5: WD is added to the CC, using Eq. 2.1.
Step 6: The inverse curvelet transform procedure is applied on R and the block is added to respective location of CC1.
Step 7: Step 3 to Step 6 are repeated till all the blocks of W1 are added to respective blocks of CI1.
Step 8: Add the remaining blocks of CI1 to CC1.
Step 9: Add red and green planes of CI to WMI and then add CC1 to WMI to generate watermarked image.

2.2.2.2 Digital Watermark Extraction Procedure

The architecture of extraction procedure is given in Fig. 2.13 and the detailed steps for extraction of the watermark are as follows:

Step 1: The third dimension or blue component of watermarked image, WMI, is partitioned into a number of p × p (for example 8 × 8 size) pixel sized nonoverlapping blocks, WMI1.
Step 2: The third dimension or blue component of cover image, CI, is partitioned into a number of p × p pixel sized nonoverlapping blocks, CI1.
Step 3: The curvelet transformation is applied to a block from CI1 to obtain curvelet coefficients C1.

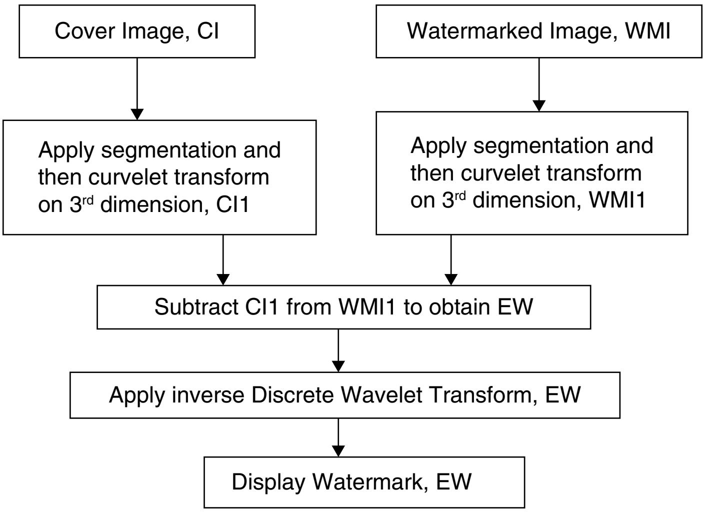

Fig. 2.13 WWCT-digital watermark extraction procedure

Step 4: The curvelet transformation is applied to a block from WMI1 to obtain curvelet coefficients WM1.
Step 5: Subtract the values of C1 from WM1 and apply inverse discrete wavelet transform procedure to obtain a block of extracting a watermark image, EWM.
Step 6: Repeat Step 3 to Step 5 for a size of given watermark (Since the watermark is embedded in continuous locations).
Step 7: Compare the extracted watermark, EWM, with W.

2.2.2.3 Experimental Results

A Lenna color image (Fig. 2.4a) of size 512×512 pixels is considered as the cover image, CI. A gray scale image with 64×64 pixels size is considered as watermark, W (Fig. 2.4b). The block size is considered as 8×8 pixels. The values of CI are embedded with content of W as given in the embedding procedure Sect. 2.2.2.1. The resultant watermarked image after manipulation is shown in Fig. 2.14. The watermark is extracted from the watermarked image (Fig. 2.14) by following the steps discussed in the extraction procedure Sect. 2.2.2.2. The extracted watermark is as shown in Fig. 2.15. The watermark is compared with the original watermark to find the performance of the procedures [12].

The watermark is embedded into various 24-bit standard 512×512 pixel color images and the respective watermarked images are generated. Table 2.3 gives the results of PSNR obtained when these watermarked images are compared with the

Fig. 2.14 Watermarked Lenna BMP image (WWCT)

Fig. 2.15 Extracted watermark from Fig. 2.14 (WWCT)

Table 2.3 PSNR results for various cover images (WWCT)

S. No.	Image name (size 512×512)	WWCT	Huang Hui-fen [6]	Yuancheng Li [8]	Patrizio Campisi [9]	Chune Zhang [7]
1	Lenna	42.09	39.9	27.7	40.46	43.166
2	Baboon	36.19	–	–	37.6	–
3	Barbara	39.68	–	–	40.14	–
4	Pepper	39.14	–	–	–	–

respective cover images. The results show that the performance of the technique discussed is better than the methods discussed by Huang Hui-fen [6], Yuancheng Li [8], Patrizio Campisi [9] and are comparable to the results obtained by Chune Zhang [7].

The Lenna watermarked image (BMP image file) shown in Fig. 2.14 is compressed with GIF, JPEG, and PNG compression techniques and the resulting watermarked images are shown in Figs. 2.16, 2.18, and 2.20, respectively. Similarly the watermark is extracted from these compressed images as discussed in the procedure Sect. 2.2.2.2 and the results are shown in Figs. 2.17, 2.19, and 2.21, respectively [12].

The technique is tested for its performance by comparing the extracted watermark with the original watermark (Fig. 2.4b) and the PSNR results are tabulated in Table 2.4. The resultant PSNR shows that the obtained watermarks are consisting of less noise and are in an acceptable range.

Fig. 2.16 Watermarked Lenna image compressed by GIF format (WWCT)

Fig. 2.17 Extracted watermark from Fig. 2.16 (WWCT)

Fig. 2.18 Watermarked Lenna image compressed by JPEG format (WWCT)

Fig. 2.19 Extracted watermark from Fig. 2.18 (WWCT)

Fig. 2.20 Watermarked Lenna image compressed by PNG format (WWCT)

Fig. 2.21 Extracted watermark from Fig. 2.20 (WWCT)

Table 2.4 Variation of PSNR for different compression formats (WWCT)

S. No.	Type of attack on 24-bit color Lenna image (size 512×512)	PSNR obtained
		WWCT
1	Watermark extracted from BMP image	34.65
2	Bmp converted to GIF format	33.27
3	Bmp converted to JPEG format	32.76
4	Bmp converted to PNG format	34.65

2.3 Image Watermarking Using Magic Square and Curvelet Transform

This section discusses about two methods in which the resized watermark is embedded into the curvelets of the cover image. The watermark is resized by using the magic square technique. In the first method, RWCT, one quadrant of the resized watermark is embedded into the cover image curvelets. In the second method, RWWCT, one quadrant of the resized watermark is transformed using discrete wavelet transform. The obtained wavelet coefficients are embedded into the curvelets of cover image [11, 12].

2.3.1 Magic Square

2.3.1.1 Magic Square Procedure

The following procedure is used to resize the watermark image (W) of size m×m pixels to equivalent cover image (CI) size of N×N.

Step 1: Divide the N×N with m×m to obtain k×k, the size of magic square to be generated for each pixel, i.e., for every pixel value of watermark a k×k magic square is generated.
Step 2: These k×k values will be copied to the respective positions of k×k nonoverlapping blocks.
Step 3: All the obtained magic square blocks will be positioned to obtain a watermark image IM, with size N×N.
Step 4: An adjustment value array will be generated where ever the pixel value is less than the possible magic square minimum value. For example the minimum value for any 4×4 magic square is 34 so if the pixel value is less than 34 then the magic square cannot be generated. In this case the adjustment value is generated and inserted into adjustment array, AD [11, 12].
Step 5: Store AD and IM for extraction purpose.

2.3.1.2 Application

The 8×8 magic square can be treated as replication of four 4×4 magic squares for simplicity, as shown in Fig. 2.22 [11, 12].

The elements of the magic square are generated based on the Eqs. 3.2 and 2.3. In Eq. 2.2, a(1) is representing first elements value and max is the pixel value for which the 8×8 magic square to be generated.

$$a(1) = \left(\frac{(\max - 34)}{2}\right) + 1 \tag{2.2}$$

$$a(k) = (k - 1) + 1 \tag{2.3}$$

2.3.2 *Technique to Insert Resized Watermark into Curvelets of Cover Image (RWCT)*

In this method, the resized watermark pixel values are embedded into the curvelet partitions of cover image as discussed in Sect. 2.3.2.1. The watermark is extracted from the watermarked image curvelets as discussed in Sect. 2.3.2.2. The architecture of embedding and extraction procedures is as shown in Figs. 2.23 and 2.24 respectively [12].

2.3.2.1 Digital Watermark Embedding Procedure

The steps involved in embedding resized watermark into curvelets of cover image based on the given architecture shown in Fig. 2.23 are as follows:

Step 1: The third dimension or blue component of image, CI, is partitioned into a number of $p \times p$ (for example 8×8 size) pixel sized nonoverlapping blocks, CI1.
Step 2: Apply magic square procedure discussed in Sect. 2.3.1.1 on W to obtain IM with size equal to CI.
Step 3: One quadrant of IM is partitioned into number of nonoverlapping small blocks, where the size of each block is $p \times p$ pixels, W1.
Step 4: The curvelet procedure is applied on a block from CI1 to obtain curvelet coefficients, CC.
Step 5: W1 is added to CC, using Eq. 2.4.

$$R_{i,j} = W1_{i,j} + CC_{i,j} \tag{2.4}$$

Fig. 2.22 8 × 8 Magic square format

68	68	68	68	68	68	68	68	68	68
68	1	13	12	8	8	11	6	9	
68	2	14	7	11	12	7	10	5	
68	15	3	10	6	13	14	3	4	
68	16	4	5	9	1	2	15	16	
68	8	11	6	9	1	13	12	8	
68	12	7	10	5	2	14	7	11	
68	13	14	3	4	15	3	10	6	
68	1	2	15	16	16	4	5	9	

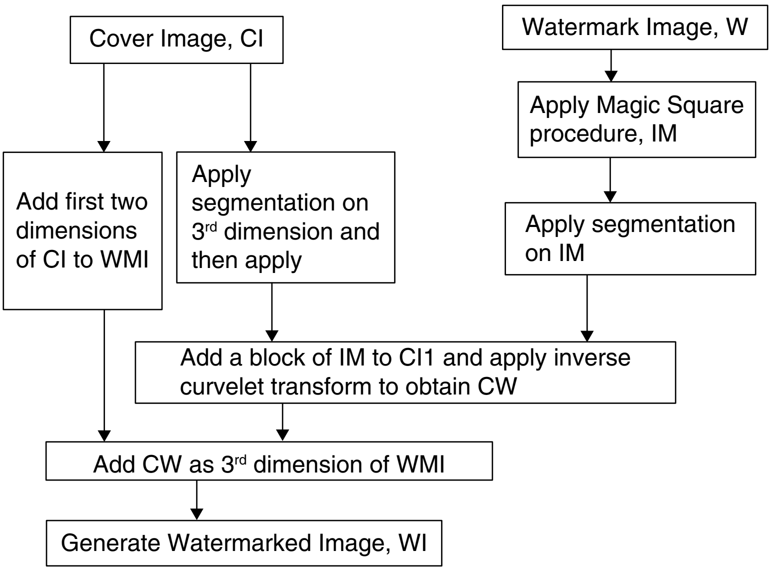

Fig. 2.23 RWCT-digital watermark embedding procedure

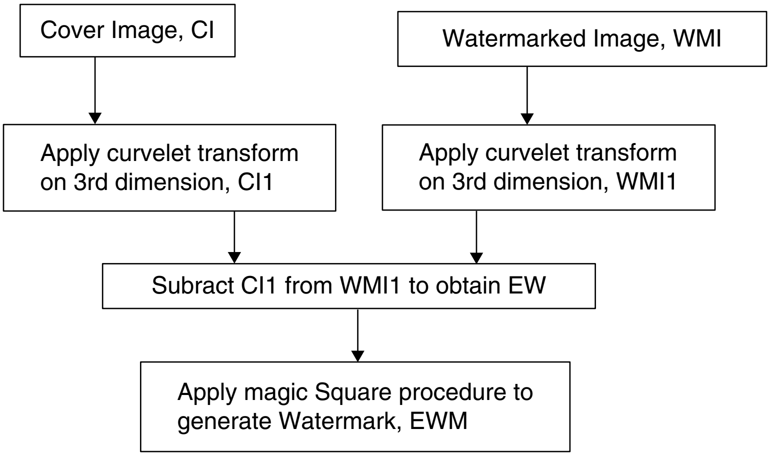

Fig. 2.24 RWCT-digital watermark extraction procedure

Step 6: The inverse curvelet transform procedure is applied on R and the block is added to respective location of CC1.
Step 7: Step 3 to Step 6 are repeated till all the blocks of W1 are added to respective blocks of CI1.
Step 8: Add the remaining blocks of CI1 to CC1.
Step 9: Add red and green planes of CI to WMI and then add CC1 to WMI to generate watermarked image.

2.3.2.2 Digital Watermark Extraction Procedure

The steps involved in extracting watermark from curvelets of watermarked image based on the given architecture shown in Fig. 2.24 are as follows:

Step 1: The third dimension or blue component of watermarked image, WMI, is partitioned into a number of p × p (for example 8 × 8 size) pixel sized nonoverlapping blocks, WMI1.
Step 2: The third dimension or blue component of cover image, CI, is partitioned into a number of p × p pixel sized nonoverlapping blocks, CI1.
Step 3: The curvelet transformation is applied to a block from CI1 to obtain curvelet coefficients C1.
Step 4: The curvelet transformation is applied to a block from WMI1 to obtain curvelet coefficients WM1.
Step 5: Subtract the values of C1 from WM1 to obtain a block of EM.
Step 6: Repeat Step 3 to Step 5 for a one quadrant (Since one quadrant of the magic square is embedded in continuous locations), EM.
Step 7: Generate the watermark based on magic square technique (the sum of all elements in any row, column, or diagonal will give the respective pixel value), EWM
Step 8: Compare the extracted watermark EWM with W.

2.3.2.3 Experimental Results

A Lenna color image (Fig. 2.4a) of size 512×512 pixels is considered as the cover image, CI. The block size is considered as 8×8 pixels. An array, IM, of a size equivalent to the size of cover image is created with all zero values initially and divided into 64 equal nonoverlapping blocks. A grayscale image with 64×64 pixels size is considered as watermark, W (Fig. 2.4b). The magic square application discussed in Sect. 2.3.1.2 is applied on each pixel of W to generate a magic square for the given pixel value. The 64 elements of the obtained magic square are inserted into all blocks of IM at the respective pixel location as considered from W. This results in resized watermark consisting of 64 blocks of the given watermark with varying intensities and the image is shown in Fig. 2.25. Figure 2.26 is obtained by scaling the Fig. 2.25 by a factor of 10. The minimum value is 34 for which a 4×4 magic square can be generated. As per the magic square procedure discussed in Sect. 2.3.1.2, for all pixels of W an adjustment array, AD, is generated [12].

The embedding procedure discussed in Sect. 2.3.2.1 is applied. The resultant watermarked image of the embedding procedure is shown in Fig. 2.27. The watermark is extracted from the watermarked image (Fig. 2.27) as discussed in the procedure Sect. 2.3.2.2 and is shown in Fig. 2.28. The result is compared with the original watermark to check the performance of the technique.

The watermark is embedded into various 24-bit standard color images and tested for the performance. Table 2.5 gives the comparison between cover images and watermarked images based on PSNR. The PSNR shows that the obtained watermarked images are noisy images. It is observable that the technique is better than Yuancheng Li's watermarking technique [8] and comparable to Chune Zhang et al. watermarking Technique [7].

The Lenna watermarked image (BMP image file) shown in Fig. 2.27 is compressed with GIF, JPEG, and PNG compression techniques and the resulting watermarked images are shown in Figs. 2.29, 2.31, and 2.33, respectively. Similarly the watermark is extracted from these compressed images as discussed in the procedure Sect. 2.3.2.2 and the results are shown in Figs. 2.30, 2.32, and 2.34, respectively [12].

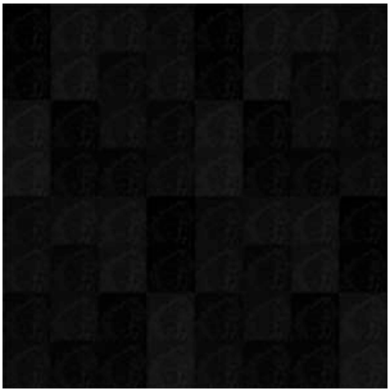

Fig. 2.25 Resized watermark (RWCT)

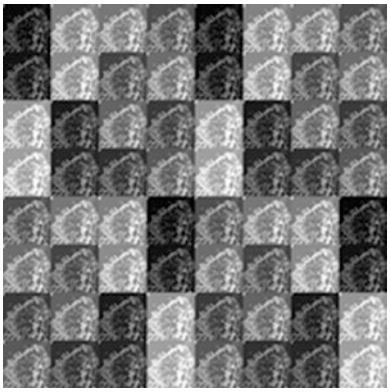

Fig. 2.26 Resized watermark scaled with 10 factor (RWCT)

Fig. 2.27 Watermarked Lenna BMP image (RWCT)

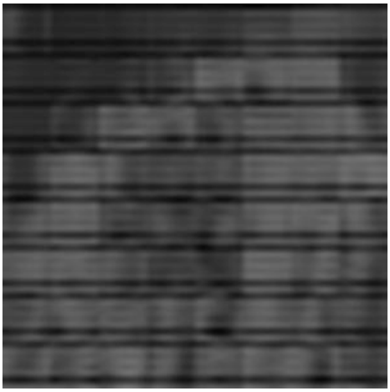

Fig. 2.28 Extracted watermark from Fig. 2.27 (RWCT)

Table 2.5 PSNR results for various cover images (RWCT)

S. No	Image name (size 512×512)	RWCT	Huang Hui-fen [6]	Yuancheng Li [8]	Patrizio Campisi [9]	Bhatnagar [13]	Chune Zhang [7]
1	Lenna	32.35	39.9	27.7	40.46	39.65	43.166
2	Baboon	31.50	–	–	37.6	–	–
3	Barbara	31.63	–	–	40.14	–	–
4	Pepper	31.52	–	–	–	–	–

Fig. 2.29 Watermarked Lenna image compressed by GIF format (RWCT)

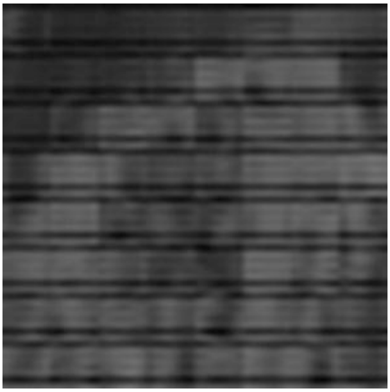

Fig. 2.30 Extracted watermark from Fig. 2.29 (RWCT)

Fig. 2.31 Watermarked Lenna image compressed by JPEG format (RWCT)

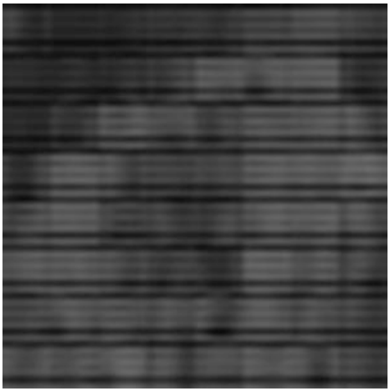

Fig. 2.32 Extracted watermark from Fig. 2.31 (RWCT)

Fig. 2.33 Watermarked Lenna image compressed by PNG format (RWCT)

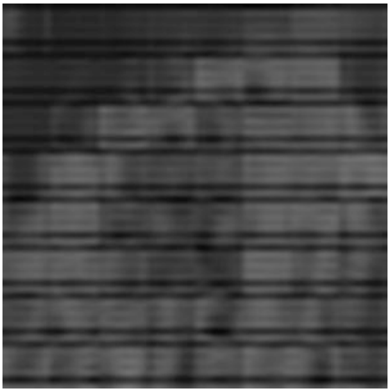

Fig. 2.34 Extracted watermark from Fig. 2.33 (RWCT)

The technique is tested for its performance by comparing the extracted watermarks with the original watermark (Fig. 2.4b) and the PSNR results are tabulated in Table 2.6. The resultant PSNR shows that the obtained watermarks are noisy. So there is a need to strengthen the technique.

2.3.3 Technique to Insert Resized Watermark Wavelets into Curvelets of Cover Image (RWWCT)

The architectures of embedding and extraction procedures of this technique are as shown in Figs. 2.35 and 2.36 respectively. In this method, the watermark is resized by using the magic square procedure as discussed in Sect. 2.3.1. One quadrant of the magic square is considered and segmented into W1 nonoverlapping partitions, where each partition size is p×p pixels, and DWT is applied to each partition. The cover image is also partitioned similarly and curvelets are calculated for every partition. The wavelets of magic square partitions are embedded into the obtained curvelets as given in the embedding procedure Sect. 2.3.1.1. In the same way the watermarked image is partitioned into WMI1 blocks and then the watermark is extracted from the curvelets of the respective blocks as discussed in Sect. 2.3.1.2 [12].

2.3.3.1 Digital Watermark Embedding Procedure

The steps involved in embedding wavelets of resized watermark into curvelets of cover image based on the given architecture shown in Fig. 2.35 are as follows:

Step 1: The third dimension or blue component of image, CI, is partitioned into a number of p×p (for example 8×8 size) pixel sized nonoverlapping blocks, CI1.

Step 2: Apply magic square procedure discussed in Sect. 2.3.1 on W to obtain IM with size equal to CI.

Table 2.6 Variation of PSNR for different compression formats (RWCT)

S. No.	Type of attack on 24-bit color Lenna image (size 512×512)	PSNR obtained RWCT
1	Watermark extracted from BMP image	15.64
2	Bmp converted to GIF format	15.65
3	Bmp converted to JPEG format	14.89
4	Bmp converted to PNG format	15.64

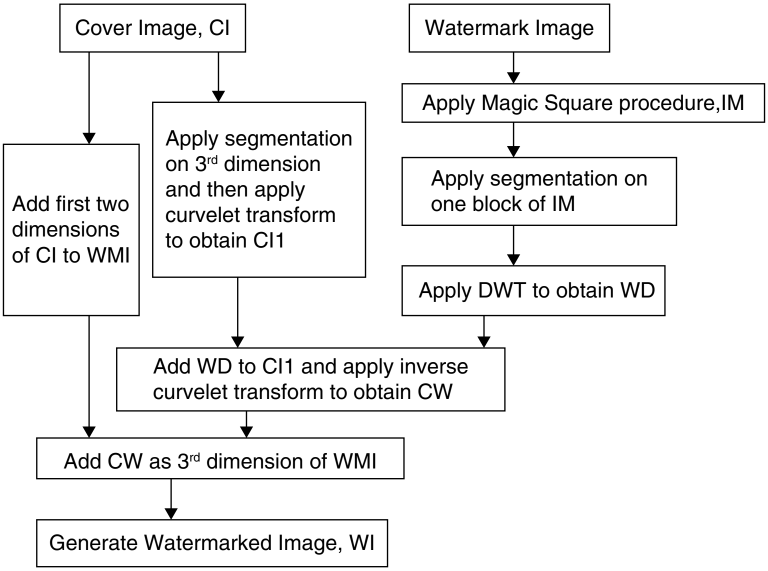

Fig. 2.35 RWWCT-digital watermark embedding procedure

Step 3: One quadrant of IM is partitioned into a number of nonoverlapping small blocks, where the size of each block is p×p pixels, W1.

Step 4: The curvelet procedure is applied to a block from CI1 to obtain curvelet coefficients, CC.

Step 5: The discrete wavelet transform procedure is applied to a block from W1 to obtain transform coefficients, WD.

Step 6: WD is added to the CC, using Eq. 2.5.

$$R_{i,j} = WD_{i,j} + CC_{i,j} \tag{2.5}$$

Step 7: The inverse curvelet transform procedure is applied to R and the block is added to respective location of CC1.

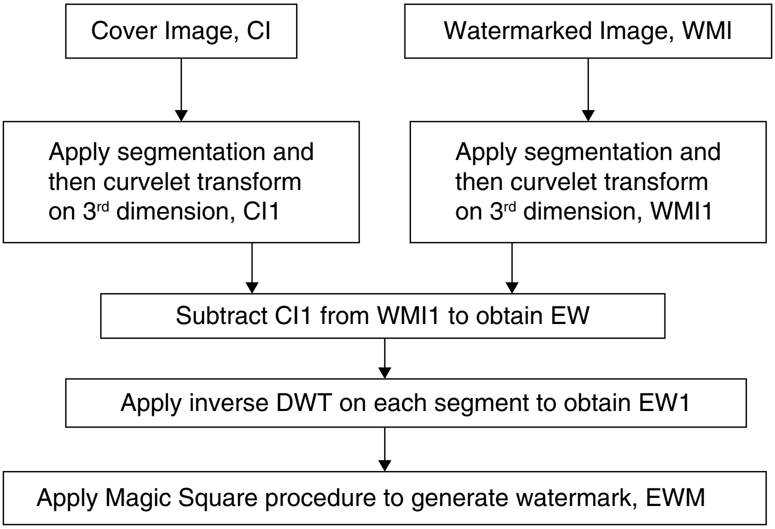

Fig. 2.36 RWWCT-digital watermark extraction procedure

Step 8: Step 3 to Step 7 are repeated till all the blocks of W1 add to respective blocks of CI1.
Step 9: Add the remaining blocks of CI1 to CC1.
Step 10: Add red and green planes of CI to WMI and then add CC1 to WMI to generate a watermarked image.

2.3.3.2 Digital Watermark Extraction Procedure

The steps involved in extracting watermark from curvelets of cover image based on the given architecture shown in Fig. 2.36 are as follows:

Step 1: The third dimension or blue component of watermarked image, WMI, is partitioned into a number of $p \times p$ (for example 8×8 size) pixel sized nonoverlapping blocks, WMI1.
Step 2: The third dimension or blue component of cover image, CI, is partitioned into a number of $p \times p$ pixel sized nonoverlapping blocks, CI1.
Step 3: The curvelet transformation is applied to a block from CI1 to obtain curvelet coefficients C1.
Step 4: The curvelet transformation is applied to a block from WMI1 to obtain curvelet coefficients WM1.
Step 5: Subtract the values of C1 from WM1 and apply inverse wavelet transform to obtain a block of EM.
Step 6: Repeat Step 3 to Step 5 for a one quadrant (Since one quadrant of the magic square is embedded in continuous locations), EM.

Step 7: Generate the watermark based on magic square technique (the sum of all elements in any row, column or diagonal will give the respective pixel value), EWM

Step 8: Compare the extracted watermark EWM with W.

2.3.3.3 Experimental Results

A Lenna color image (Fig. 2.4a) of size 512×512 pixels is considered as a cover image, CI. The block size is considered as 8×8 pixels. An array, IM, of a size equivalent to the size of cover image is created with all zero values initially and divided into 64 equal blocks. A grayscale image with 64×64 pixels size is considered as watermark, W (Fig. 2.4b). The magic square application discussed in Sect. 2.3.1.2 is applied to each pixel of W to generate a magic square for the given pixel value. The 64 elements of the obtained magic square are inserted into all blocks of IM at the respective pixel location as considered from W. This results in resized watermark consisting of 64 blocks of the given watermark with varying intensities and the image is shown in Fig. 2.25. Figure 2.26 is the complement image of Fig. 2.25. The minimum value is 34 for which a 4×4 magic square can be generated. As per the magic square procedure discussed in Sect. 2.3.1.2, for all pixels of W an adjustment array, AD, is generated [11, 12].

The embedding procedure discussed in Sect. 2.3.3.1 is applied. The resultant watermarked image of the embedding procedure is shown in Fig. 2.37. The watermark is extracted from the watermarked image (Fig. 2.37) as discussed in the procedure Sect. 2.3.3.2 and is shown in Fig. 2.38. The result is compared with the original watermark to check the performance of the technique.

The watermark is embedded into various 24-bit color images and they are compared. Table 2.7 gives the comparison between cover images and watermarked images based on PSNR. The results show that the technique is better than the techniques discussed by Huang Hui-fen [6], Yuancheng Li [8], and Patrizio Campisi [9] in the case of the Lenna cover image. The technique has shown its weakness when compared with Patrizio Campisi [9] technique in the cases of Baboon and Barbara cover images and also in the case of technique discussed by Chune Zhang [7].

The Lenna watermarked image (BMP image file) shown in Fig. 2.37 is compressed with GIF, JPEG, and PNG compression techniques and the resultant watermarked images are shown in Figs. 2.39, 2.41, and 2.43, respectively. Similarly the watermark is extracted from these compressed images as discussed in the procedure Sect. 2.3.3.2 and the results are shown in Figs. 2.40, 2.42, and 2.44, respectively [12].

The technique is tested for its performance by comparing the extracted watermark with the original watermark (Fig. 2.4b) and the PSNR results are tabulated in Table 2.8. The resultant PSNR shows that the obtained watermarks are noisy, but are in acceptable level. There is a possibility to improve the quality of extracting watermark by improving the technique.

Fig. 2.37 Watermarked Lenna BMP image (RWWCT)

Fig. 2.38 Extracted watermark from Fig. 2.37 (RWWCT)

Table 2.7 PSNR results for various cover images (RWWCT)

S. No	Image name (size 512×512)	RWWCT	Huang Hui-fen [6]	Yuancheng Li [8]	Patrizio Campisi [9]	Chune Zhang [7]
1	Lenna	42.10	39.9	27.7	40.46	43.166
2	Baboon	36.21	–	–	37.6	–
3	Barbara	39.68	–	–	40.14	–
4	Pepper	39.23	–	–	–	–

Fig. 2.39 Watermarked Lenna image compressed by GIF format (RWWCT)

Fig. 2.40 Extracted watermark from Fig. 2.39 (RWWCT)

Fig. 2.41 Watermarked Lenna image compressed by JPEG format (RWWCT)

Fig. 2.42 Extracted watermark from Fig. 2.41 (RWWCT)

Fig. 2.43 Watermarked Lenna image compressed by PNG format (RWWCT)

Fig. 2.44 Extracted watermark from Fig. 2.43 (RWWCT)

Table 2.8 Variation of PSNR for different compression formats (RWWCT)

S. No.	Type of attack on 24-bit color Lenna image (size 512×512)	PSNR obtained
1	Watermark extracted from BMP image	29.768
2	Bmp converted to GIF format	29.488
3	Bmp converted to JPEG format	29.613
4	Bmp converted to PNG format	29.768

Table 2.9 PSNR results for various cover images—Curvelet techniques

Image name (size 512× 512)	WCT	WWCT	RWCT	RWWCT	Huang Hui-fen [6]	Yuan cheng Li [8]	Patrizio Campisi [9]	Chune Zhang [7]
Lenna	31.49	42.09	32.35	42.10	39.9	27.7	40.46	43.166
Baboon	31.17	36.19	31.50	36.21	–	–	37.6	–
Barbara	31.30	39.68	31.63	39.68	–	–	40.14	–
Pepper	31.20	39.14	31.52	39.23	–	–	–	–

2.4 Summary

This chapter has presented four different methods of embedding and extraction of watermark using magic square technique and curvelet transform techniques. Table 2.9 gives the PSNR of the various watermarked image compared with the respective cover images. The current study also gives the comparative study of the extracted watermark with the original watermark (Fig. 2.4b) and the results are shown in Table 2.10. It is shown that the magic square technique can serve better in resizing the watermark image as per the requirement by spreading its brightness to create multiple copies of given watermark so as to preserve the quality of the cover image. It has also been shown that this makes the watermark to survive against compression attacks. It is observed that the technique in which the wavelet coefficients embedded in image curvelets is more efficient in retaining the watermark. The results show that the WWCT and RWWCT are better than other methods, where as WWCT is an acceptable method for preserving authentication information, i.e., for extracting less noisy watermark than other methods [12]. It is also observed that the curvelet transform techniques have demonstrated the robustness against compression attacks. But there is still scope to improve the embedding and extraction procedures, which are discussed in the next chapter.

Table 2.10 Variation of PSNR for different compression formats—Curvelet techniques

S. No.	Type of attack on 24-bit color Lenna image (size 512×512)	PSNR obtained			
		WCT	WWCT	RWCT	RWWCT
1	Watermark extracted from BMP image	21.53	34.65	15.64	29.76
2	Bmp converted to GIF format	21.41	33.27	15.65	29.48
3	Bmp converted to JPEG format	20.39	32.76	14.89	29.61
4	Bmp converted to PNG format	21.42	34.65	15.64	29.76

References

1. Tao Xie, "An evolutionary algorithm for magic squares", The 2003 congress on Evolutionary Computation, Vol. 2, 2003, pp. 906–913
2. http://www-history.mcs.st-and.ac.uk/Biographies/Daubechies.html
3. David L. Donoho & Ana Georgina Flesia, "Digital Ridgelet Transform based on True Ridge Functions", International Journal of Studies in Computational Mathematics, Vol. 10, 2003, pp. 1–30
4. Jean-Luc Starck, Emmanuel J. Candès & David L. Donoho, "The Curvelet Transform for Image Denoising", IEEE Transactions on Image Processing, Vol. 11, No. 6, 2002, pp. 670-684
5. William Symes Andrews, Lorraine Screven Frierson, Charles Albert Browne, "Magic Squares and Cubes", Chicago, IL: Open court publish company, 1908.
6. Huang Hui-fen, "Perceptual Image Watermarking Algorithm Based on Magic Squares Scrambling in DWT", 5th International Joint conference on INC, IMS, IDC, 2009, pp. 1819-1822
7. Chune Zhang, L. L. Cheng, Zhengding Qiu, and L. M. Cheng, "Multipurpose Watermarking Based on Multiscale Curvelet Transform", IEEE Transactions on Information Forensics and Security, Dec 2008, Vol. 3, No. 4, pp. 611-619.
8. Yuancheng Li, "An Image Digital Watermarking Method Based On Ridgelet and KICA", International Conference on MultiMedia and Information Technology (MMIT), 2008, pp. 345-348
9. Patrizio Campisi, Deepa Kundur & Alessandro Neri, "Robust Digital Watermarking in the Ridgelet Domain", IEEE Signal Processing Letters, Vol. 11, No. 10, 2004, pp. 826-830
10. B.A. Wandell, "Foundations of Vision", Sunderland, MA, Sinauer Associates, 1995.
11. Channapragada R. S. R. & Munaga V. N. K. Prasad, "Digital Watermarking Based on Magic Square and Ridgelet Transform Techniques", Intelligent Computing, Networking, and Informatics (International Conference on Advanced Computing, Networking and Informatics-Advances in Intelligent Systems and Computing), Vol. 243, 2014, pp. 143-161.
12. Channapragada R. S. R. & Munaga V. N. K. Prasad, "Watermarking Techniques in Curvelet Domain", International Conference on Computational Intelligence in Data Mining, Vol. 1, 2014, pp. 199-211.
13. G. Bhatnagar, B. Raman & Q.M.J. Wu, "Robust watermarking using fractional wavelet packet transform", IET Image Processing, Vol. 6, Issue. 4, 2012, pp. 386–397

Springer

http://www.springer.com/978-3-319-32950-5

Digital Watermarking Techniques in Curvelet and
Ridgelet Domain
Rao, C.R.S.; Prasad, M.V.N.K.
2016, XVIII, 92 p. 79 illus., 32 illus. in color., Softcover
ISBN: 978-3-319-32950-5